Combining Heterogeneous Information: Contributions to the Extraction and Analysis of Feature-Rich Complex Networks

Vincent Labatut

AVIGNON UNIVERSITÉ

HABILITATION À DIRIGER DES RECHERCHES

École doctorale 536
Agroscience & Sciences

Spécialité : Informatique

Laboratoire Informatique d'Avignon

Présentée par
Vincent Labatut

Combining Heterogeneous Information

Contributions to the Extraction and Analysis of Feature-Rich Complex Networks

---

HDR soutenue publiquement le 16/06/2021 devant le jury composé de :

Céline Rouveirol, Professeur, Université Sorbonne Paris Nord, LIPN, **Présidente**
Éric Gaussier, Professeur, Université Grenoble Alpes, LIG, **Rapporteur**
Jean-Loup Guillaume, Professeur, Université de La Rochelle, L3i, **Rapporteur**
Maguelonne Teisseire, Directrice de recherche, INRAE, UMR Tetis, **Rapportrice**
Roger Guimerà, Professeur, Universitat Rovira i Virgili, ICREA, **Examinateur**

LABORATOIRE INFORMATIQUE D'AVIGNON

Contents

List of Figures

List of Tables

1 Introduction

1.1 Complex Networks

The expression *Complex networks* started to be used in the late 1990s and early 2000s to name graphs used to model real-world complex systems [4, 51, 207]. Here, the term *system* refers to a set of interacting objects, and *complex* means that they do so in a heterogeneous way, through various processes typically involving non-linearity, self-organization, and feedback loops [207]. This makes them evolve in a way that appears as neither completely regular or completely random, and results in the manifestation of so-called *emergent* behaviors at the level of the system [17]. The reductionist approach, which consists in studying the system's components in isolation in order to predict their collective behavior, is not efficient when applied to complex systems [4, 51]. This is because an emergent behavior is precisely the result of the various interactions occurring between the system's components, which consequently must be taken into account.

Graphs are mathematical objects specifically designed to represent interactions and relations, therefore they constitute a natural framework to represent complex systems. Due to the specific characteristics of these systems, these graphs themselves possess so-called *non-trivial* topological properties [4, 178], which distinguish them from regular graphs and random graphs, the two classes of graphs that were studied in the literature until then. This in turn justifies the use of a specific name for this third class of graphs: complex networks. There is no precise and extensive list of such properties, but the most frequently cited in the literature are certainly [139]: small-worldness [232], scale-freeness [18], (dis)assortative mixing [176], motif distribution [172], and community structure [179].

The seminal work of Watts and Strogatz [232] on the small-world property, and of Barabási and Albert [18] on scale-free networks, marked the beginning of a new research domain now called *Network Science*, and aiming at studying complex networks. It is a multidisciplinary field that relies largely on a number of pre-existing domains, in particular graph theory, quantitative sociology, computer science, operations research, statistical physics, and of course complex systems [17, 29, 35, 40, 178, 207]. It is worth stressing the bottom-up or data-driven origin of the Network Science field [17, 40], as it sprang from the empirical observation that complex networks possess different topological properties compared to other types of graphs. Network scientists are also interested in other problems than identifying characteristic topological properties and designing measures allowing to quantity such properties, though, including: defining models of the structure and its evolution, predicting the behavior of a network, studying some dynamic process taking place on the network, and others [178].

Network Science is mainly a *data* science, as its starting point is the modeling of real-world systems [62]. As such, its emergence is due not only to the convergence of interdisciplinary efforts, but also to the availability of the resources required to build and study large and/or numerous complex networks: computational power and access to sizable datasets [4, 17, 178]. Because of this fundamental reliance on data, information representation is a fundamental aspect of Network Science. The way the available data describing the considered system are included in the graph-based model is of the utmost importance. Yet, plain graphs are

meant only to model one type of information: the presence or absence of relationships between the object constituting the system. To handle more diverse data, it is necessary to extend this framework, which leads us to the notion of feature-rich network that is at the core of this manuscript.

1.2 Feature-Rich Networks

When building a model of some real-world system, one generally wants to include enough information so as to build a reliable representation. Graphs are no exception to this rule, and consequently researchers from various scientific domains did not wait for the emergence of Network Science to experiment with various extensions of the notion of graph. In particular, many of them appeared in the sixties, in the field of quantitative Sociology, and were later picked up (or sometimes even rediscovered) by physicists, at the emergence of Network Science [62].

Definition 1.2.1 (Plain Network) *A **plain** network is a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_I\}$ is the set of vertices, and $E \subset V^2$ is the set of edges. As the graph is undirected, we assume that the pairs of vertices constituting the edges are lexicographically ordered.*

In this manuscript, I call *plain network* the most simple type of graph, corresponding to a set of vertices connected with undirected unweighted edges (cf. Definition 1.2.1). A number of methods exist to include more information in this basic model, differing by the nature of this information. The most natural extension, which only modifies marginally this definition, is to introduce edge *directions*. This can be done by relaxing the lexicographic order constraint in Definition 1.2.1, and it allows modeling more accurately systems with asymmetric relationships. Edge *weights* are also a straightforward extension, often performed by considering a weight function or matrix that specifies the numerical value associated to each edge. This allows modeling relationships of various intensity levels, for instance. *Signed* graphs were introduced to represent systems containing two types of relationships with opposite semantics [122], such as love/hate in a social network. *Multiplex* graphs (a.k.a. multidimensional or multirelational) have been used for decades by sociologists to represent several types of relationships in the same network. Unlike signed networks, these are not mutually exclusive though, which means two vertices can be connected in more than one way, e.g. familial/friendly/professional relationships in a social network. Multimodal networks contain several types of vertices representing different categories of entities constituting the system. They often have a *multipartite* structure, i.e. relationships can be present only between entities of different types. In *spatial* (a.k.a. location-aware) networks, each vertex has a position in space, which allows them representing geometric and geographical systems. In so-called *attributed* (a.k.a. content-based) networks, each vertex is described by a set of individual fields, e.g. age, gender and occupation in a social network. *Dynamic* (a.k.a. temporal) network includes an explicit representation of time, most often under the form of chronologically ordered layers called time-slices. Finally, *hypergraphs* constitute a generalization of the notion of graph itself, allowing to define relationships between more than two vertices. As such, they can model systems that include n-ary interactions.

Due to the largely interdisciplinary nature of Network Science, it is worth stressing that the terminology used in this field is not well established: certain concepts have several names, and the same expression can be used to refer to several distinct concepts. Furthermore, some of these extensions are sometimes subsumed by more general concepts. For instance, the term *multilayer* refers to networks constituted of several relatively separate layers. As such, they include signed, multiplex, and dynamic networks. The concept of *edge-attributed* networks is even more general, as it covers all cases where the additional information can be represented as numeric or categorical labels associated to individual edges: directed, weighted, multiplex, dynamic (with fixed V), and signed networks fall into this category. Similarly, *vertex-attributed* networks subsume multipartite, attributed, spatial, and dynamic networks. Such networks are particularly interesting, as they allow leveraging not only the available *relational* information, which is already encoded in the graph structure, but also what I call the *individual* information, which is encoded in the vertex attributes. Each vertex can be described as a separate object, which is perfect to integrate traditional tabular data into the graph. More

recently, the even more general expression *feature-rich* networks, which I borrow in this manuscript, was proposed by Interdonato *et al.* [131] to refer jointly to edge- and vertex-attributed networks.

Even if a few authors have been interested in various forms of feature-rich networks right from the start of Network Science and even before, most of the activity has focused on plain networks and on their simplest extensions, in particular weighted and directed networks. Table 1.1 lists the main books and surveys dedicated to complex networks, and reveals this evolution: older bibliographic references tend to overlook certain types of advanced feature-rich networks, such as dynamic or spatial ones, reflecting a lower research activity on these topics at this time. On the contrary, more recent references tend to adopt a wider scope, surveying works that cover the whole range of feature-rich networks. It seems natural to solve a problem in its simplest form first, before tackling more general and possibly more difficult versions. As a consequence, most tools designed for complex network analysis are available for plain networks first, then their more accessible extensions. Moreover, the first methods proposed for feature-rich networks are generally straightforward generalizations of these tools. Methods specifically tailored for such networks, and possibly more efficient than the early ones, are only proposed subsequently. An other important point is that the features listed above are not mutually exclusive, and can therefore be combined in the same graph, to get for instance a directed weighted signed multiplex network. However, this also makes it more difficult to handle the extra information, which is why researchers only deal with a few features simultaneously, in practice.

Year	Reference	Dir.	Wei.	Mpr.	Spa.	Dyn.	Mpx.	Att.	Sig.	Hyp.
2002	Albert and Barabási [4]	✓	✓	✓	·	·	·	·	·	·
	Bornholdt and Schuster [36]	✓	✓	✓	·	·	·	·	✓	·
2003	Newman [178]	✓	✓	✓	·	✓	·	·	·	✓
2004	Dorogovtsev and Mendes [89]	✓	✓	✓	·	·	·	·	·	·
2005	Brandes and Erlebach [39]	✓	✓	✓	·	✓	✓	·	·	·
2006	Boccaletti *et al.* [29]	✓	✓	·	✓	·	·	·	·	·
2007	Caldarelli and Vespignani [44]	✓	✓	✓	✓	·	·	·	·	·
2010	Cohen and Havlin [60]	✓	✓	✓	✓	·	·	·	·	·
	Cui *et al.* [68]	✓	✓	✓	·	✓	·	·	·	·
	Easley and Kleinberg [91]	✓	·	✓	·	·	·	·	✓	·
	Steen [216]	✓	✓	✓	✓	·	·	·	·	·
2011	Estrada [94]	✓	✓	✓	✓	·	·	·	·	·
2013	Borgatti *et al.* [31]	✓	·	✓	✓	·	✓	·	·	·
2014	Erciyes [92]	✓	✓	✓	·	·	·	✓	·	·
	O'Malley and Onnela [183]	✓	✓	✓	✓	✓	✓	✓	·	·
2015	Barabási [17]	✓	✓	✓	·	·	·	·	·	·
	Chapela *et al.* [51]	✓	✓	✓	·	·	✓	·	·	·
	Chen *et al.* [53]	✓	✓	✓	✓	·	·	·	·	·
	Kunegis [146]	✓	✓	✓	·	✓	·	·	✓	·
	Sayama [207]	✓	✓	✓	·	·	·	·	·	✓
2016	Zweig [242]	✓	✓	✓	✓	✓	✓	✓	·	·
2019	Interdonato *et al.* [131]	✓	✓	✓	✓	✓	✓	✓	·	·
2021	Coscia [62]	✓	✓	✓	·	✓	✓	✓	·	✓

Table 1.1: List of Complex Network or Network Science surveys and books in chronological order. Each column indicates whether the reference treats a specific type of graph: directed (*Dir.*), weighted (*Wei.*), multipartite (*Mpr.*), spatial (*Spa.*), dynamic (*Dyn.*), multiplex (*Mpx.*), attributed (*Att.*), signed (*Sig.*), and hypergraphs (*Hyp.*).

In summary, even if there is still plenty of work to do when focusing only on plain networks, feature-rich networks open a whole new perspective to Network Science. On the one hand, they allow including heterogeneous information into graphs, thereby making this framework much more expressive, and allowing to design more accurate models of real-world systems. On the other hand, this brings an additional level of complexity that makes designing analysis methods more than just the generalization of pre-existing tools. Moreover, feature-rich networks are likely to bring whole new problems to the field, or problems originating from other fields. For instance, the question of identifying the most appropriate time granularity when building a dynamic network [187] is only relevant because this type of network includes an explicit

representation of time, and this problem is related to similar tasks in the context of time series analysis [58].

1.3 Organization of the Manuscript

Scope Both my Master's degree and PhD took place at an INSERM (French national institute for medical research) unit in Toulouse, in the domain of Computational Neuroscience. This laboratory was very interdisciplinary, including neurologists, psychologists, physicians, statisticians, linguists, and computer scientists. My research focused on modeling the representation and processing of cerebral information using methods related to Bayesian networks. I obtained my PhD from the Université Paul Sabatier – Toulouse III in 2003, and got hired by the Galatasaray Üniversitesi (Istanbul, Turkey) in 2005. Simultaneously to this geographic move, I also switched to a different research topic: data mining, and soon specialized in data mining on complex networks. In 2014, I was hired by Avignon Université, and started working at its computer science laboratory (LIA – *Laboratoire Informatique d'Avignon*), still on complex network analysis. A more detailed description of my background is available in Appendix A.1.

The objective of this manuscript is to summarize the work that I did on the topic of feature-rich networks since 2007, including vertex-attributed, dynamic, spatial, signed, and/or multiplex networks. Consequently, it does not describe the computational neuroscience research conducted in Toulouse [VL89, VL90, VL59, VL57, VL58, VL83, VL56, VL82, VL19, VL75]. It does not include either the data mining research on tabular data [VL55, VL47, VL17], performed when I arrived in Istanbul, or the natural language processing work conducted later on text [VL41, VL64, VL21, VL67, VL31] and speech [VL62, VL28, VL32]. Finally, it also overlooks the work realized on 'featureless' complex networks, including some methodological work on community structure detection [VL73, VL13, VL54, VL53, VL16, VL74, VL18, VL45], centrality measures [VL39, VL12, VL72, VL71], and random network generation [VL49, VL15, VL42], as well as applications to Web service composition [VL46, VL50, VL44, VL51, VL52], and various other topics [VL40, VL69]. However, in the rest of the manuscript, whenever a connection exists between these works and those discussed in this manuscript, I explain its nature.

Bibliography Bibliographic references are numbered, but I distinguish different types of such references by using several prefixes. Prefix *VL* corresponds to articles that I (co-)authored, and that are listed in Appendix A.8. The theses of undergraduate, graduate, and PhD students that I (co-)advised are prefixed with *U*, *G*, and *D*, respectively, and listed in Appendix A.3. During my research work, I produced or participated in the production of software and corpora, whose references are prefixed by *S* and *C*, respectively, and that are listed in Appendix A.7. I want to stress that, each time it is legally possible, I publish my data and source code under an open license, so that my work is reproducible and other authors can use the methods I propose. Finally, the rest of the bibliographic references do not have any prefix, and correspond to scientific articles published by other authors, as listed in the *Bibliography* section (p.148).

Structure The rest of this manuscript is divided in seven chapters gathered in three parts. I decided to present my research activity not in a chronological way, but rather thematically, by distinguishing the different types of feature-rich networks on which I worked. It is not always straightforward to make this distinction, as some of these networks stack several types of features at once, in which case I focused on the characteristics I deemed the most important. I generally do not focus on the technical details of the presented works: the interested reader is invited to consult the publications cited throughout the manuscript.

In the *first part*, I focus on my works related to vertex-attributed networks. Chapter 2 deals with community detection, and presents a comparison of vertex modules detected based on graph structure vs. attributes, using a student activity dataset collected during a ground survey. Chapter 3 introduces several works tackling two classification problems based on attributed graphs. The first is the detection of persons which are influential in real-life, based only on data describing their activity on an online medium. The second is the identification of abusive messages in online chats.

The *second part* is dedicated to dynamic networks. Chapter 4 proposes two methods leveraging sequential pattern mining to characterize the dynamics of such networks at two distinct levels. The first targets the microscopic evolution of the network, by considering vertices and their direct neighborhood. The second describes the network at a mesoscopic level, and allows characterizing and interpreting communities (groups of vertices). In Chapter 5, I present a method based on the segmentation of dynamic graphs, and aiming at generating video summaries of TV series. The dynamic network represents the characters and their interactions, and allows modeling the plot.

The *third part* is thematically wider as it covers three types of networks. Chapter 6 is dedicated to spatial networks, presenting two works revolving around the Straightness measure. I first describe the empirical results that we obtained on a collection of geometric networks, before introducing a method that I proposed to efficiently compute this spatial measure. Chapter 7 is about the partitioning of signed networks and the concept of structural balance. I present some experimental work conducted on real-world signed graphs to study existing partitioning methods, and an analysis of the space of optimal solutions of the Correlation Clustering problem itself. Chapter 8 deals with multiplex networks. I first describe a vertex centrality measure that relies on a model of opinion diffusion in a multiplex network representing users jointly interacting on several social media. I then present a graph partitioning method allowing to identify several modular structures for a single multiplex network, as well as their associated layers. The method is applied to a real-world data set of voting activity at the European Parliament.

Finally, in Chapter 9 I review the contributions of my work and discuss future perspectives for my research. Appendix A gives an exhaustive description of the research and teaching work that I conducted during and after my PhD.

Attributed Networks

2

Community Structure of Attributed Networks

2.1 Context

The *community detection* problem is an unsupervised task that consists in detecting the community structure of a network. In its simplest form, a *community structure* is a partition of the vertex set V, as described formally in Definition 2.1.1. The research activity related to community detection constitutes a large part of the articles published in the field of complex network analysis, as literally *thousands* of methods have been published since the seminal work of Girvan and Newman [112]. A number of surveys published in the last 15 years have tried to provide a unified view of this problem [103, 195, 208], but the field is now too large, and recent articles instead focus on more specialized versions of the task on specific data such as attributed networks (as mentioned above), dynamic networks [47], bipartite networks [9]; or narrow the survey to a specific type of algorithm such as local approaches [134], or nature-inspired methods [1].

Definition 2.1.1 (Community Structure) *Let $G = (V, E)$ be a network. Then a **community structure** of G is a partition of V noted $\mathcal{P} = \{P_1, \ldots, P_K\}$, where K is the number of parts. Each part P_k $(1 \leq k \leq K)$ is called a **community**.*

This is the problem that led me to start working on complex networks at Galatasaray in 2008. I was first interested in the performance assessment of community detection methods and their comparison [VL54, VL53], on plain networks. This work was conducted with Günce Orman, which I supervised as a graduate student [G13]. This led us to study the topological properties of community structures [VL74, VL42], and to propose a method to generate random graphs possessing a controlled community structure and realistic topological properties [VL99, VL49, VL15], to be used as a benchmark when evaluating community detection methods [VL18]. I also advised two undergraduate students on the topic of community detection methods [U7, U10]. Günce Orman and I noticed that existing evaluation methods were straightforward adaptations of approaches coming from cluster analysis, and just relied on community membership while completely ignoring the graph structure. We thus designed an evaluation method able to take this important information into account, through the use of community-related topological measures [VL16, VL45]. I pursued this work in two different directions. First, regarding community-related topological measures, I started a collaboration with Nicolas Dugué and Anthony Perez, during which we generalized the community-based role measures originally proposed by Guimerà and Amaral [114]. We used these measures to characterize the network position and role of so-called social capitalists, a specific kind of Twitter user [VL39, VL12, VL72, VL71]. Second, regarding the evaluation of community detection methods, I proposed on my own a modification

of existing partition-based performance measures allowing to leverage the topological properties of the communities when assessing the accuracy of community detection methods [VL73, VL13].

During this work on community detection in *plain* networks, it appeared clearly that the research in the area had focused essentially on the design and improvement of community detection methods. However, from the application perspective, detecting the community structure of a network of interest is only half the work. Indeed, this information has no value in itself: one must additionally *interpret* the community structure relative to the system modeled by the network, in order to bring some sense to it, thus allowing human understanding. In early community detection works, the studied networks were very small, which allowed to interpret the communities manually. In other words, one would study subjectively the individuals composing some community of interest, and try to identify some relevant patterns or regularities in order to reach some observations considered useful to understand the studied system [112, 179]. When the scale of the networks increased from tens to hundreds, it was still possible to consult domain experts to perform interpretation in a similar fashion [196, 202]. However, this method showed its limit on larger networks. Blondel *et al.* [27] applied their Louvain algorithm on a network of Belgian mobile phone communications, including 2.6 million vertices representing persons. Interpreting such a large network obviously requires a more automatic approach. Blondel *et al.* verified the accuracy of the top level of their hierarchical community structure by considering the homogeneity of the users in terms of spoken language. This highlights the difficulty of interpreting communities in networks of this size, and also shows that one solution can be to consider some additional information, such as vertex attributes in this case.

Yet, at this time very few researchers had addressed the problem of characterizing and interpreting the detected communities. This problem can be seen as completely independent from the method used to detect the studied community structure. The definition of the notion of community originating from social sciences underlines that vertices belonging to the same community should be relatively similar and/or share a common behavior [170]. Assessing the similarity between vertices requires describing them, which can be performed both in terms of *individual* information (i.e. personal characteristics) and *relational* information (i.e. connection to the rest of the network). Concretely, the former corresponds to vertex attributes, whereas the latter depends on the network topology. The behavior of a vertex can be described in terms of evolution of its individual and relational information. This approach allows us to take advantage of most types of information one can encode in a network (structure, directions, weights, attributes, time, etc.). Early works aiming at characterizing community structures are consistent with this perspective, as they use various topological measures to describe the vertices in plain networks. In [150], Lancichinetti *et al*. visually examined the distribution of some community-based topological measures, both at local and intermediary levels. Their goal was to understand the general shape of communities belonging to networks modeling various types of real-world systems. In [153], Leskovec *et al*. proposed to study the community structure as a whole, by considering it at various scales, thanks to a global measure called conductance. These two studies are valuable, however, from the interpretation perspective, they are limited by the fact that they consider the network as a whole. Communities are studied and characterized collectively, in order to identify trends in the whole network, or even a collection of networks.

In 2009, I started an interdisciplinary collaboration with Business Scientist Jean-Michel Balasque, at Galatasaray Üniversitesi. We organized a field survey among the Galatasaray students in order to study their buying patterns regarding certain technological goods (mobile phones and music players). This survey allowed us to gather both individual and relational information regarding this population, and thus to build an *attributed* graph representing the social network of students. We took advantage of the additional information conveyed by the vertex attributes to interpret the detected communities, a work that I describe in this chapter, and that resulted in several publications [VL98, VL48, VL23, VL24]. As mentioned before, any extra information can be leveraged to interpret community structures: I later participated in the development of a method to characterize and interpret communities in *dynamic* networks, as described in Chapter 4. Among others, this work was conducted in collaboration with Günce Orman, with whom I wrote a review about community structure characterization, based on our first-hand experience on this topic [VL22].

An *attributed network* is a graph whose vertices and/or edges are described by one or several attributes. In this manuscript, I only focus on vertex-attributed networks, so in the following I simply use the word *attributed* to

refer to this type of graphs. The formal description of this type of graph is provided in Definition 2.1.2, and Figure 2.1 shows an example. Each attribute has a specific semantic and encodes some information that could not be represented using a plain graph. For instance, the graph from Figure 2.1 exhibits the social network of a group of persons individually described by their gender, weight and height. Each attribute has a domain specifying all the values that can be used to describe some vertex. Each vertex can be characterized in terms of its attribute values, in addition to its interconnections in the network.

Definition 2.1.2 (Attributed Network) *Let $G = (V, E, \mathcal{A})$ an **attributed network**, where $V = \{v_1, \ldots, v_I\}$ is the set of vertices, $E \subset V^2$ is the set of edges, and $\mathcal{A} = \{a_1, \ldots, a_N\}$ is the set of vertex attribute functions. If the graph is undirected, we assume that edges are lexicographically ordered pairs. Each function a_n $(1 \leq n \leq N)$ is defined from V to some domain noted $Dom(a_n)$. Therefore, each vertex v_i is described by a collection of values $a_1(v_i), \ldots, a_N(v_i)$.*

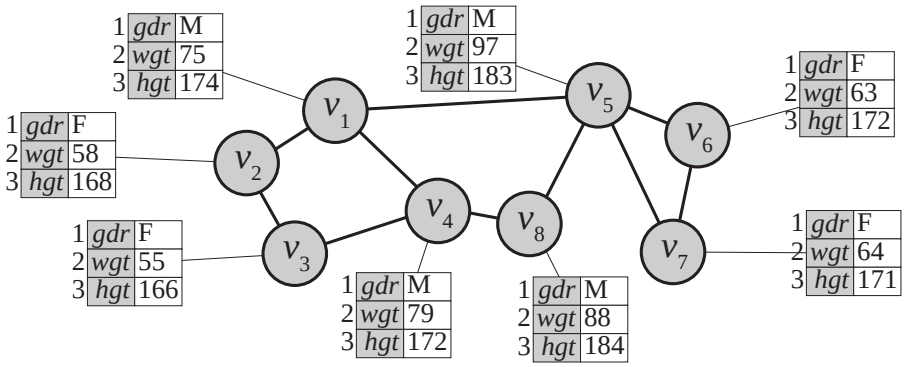

Figure 2.1: Example of random attributed network containing 8 vertices described by 3 attributes: Gender (gdr); Weight (wgt); and Height (hgt).

Wasserman and Faust [231] call vertex attributes *compositional variables*, and mention that they are the standard variables used in sociology, by opposition to *structural variables* that describe how the graph is organized. It is therefore not a surprise that attributed graphs, that combine both types of variables, have been used to model social systems at least since the 70s [231]. The research activity on this topic has increased with the development of the complex network field, as shown by the recent surveys focusing on tasks related to attributed networks, such as relational machine learning [138, 182], graph querying [230] or community detection [11, 38, 57].

The work presented in this chapter is not concerned with methods leveraging vertex attributes in the community detection process, though. These methods often rely implicitly on the assumption of *homophily*, which states that vertices with similar attributes tend to be connected [38]. One then looks for communities that are dense in terms of edges, and homogeneous in terms of attributes. However, in practice, this assumption is not necessarily true [57], or only for certain attributes, which can vary from one community to the other. Instead, the work presented here aims at studying how a community structure detected solely on the basis of the graph structure can be characterized thanks to vertex attributes. Moreover, this work also seeks to determine how much the information conveyed by the graph structure and the vertex attributes overlap, by comparing the vertex partitions obtained when considering separately both types of information. The rest of this chapter is organized as follows. In Section 2.2, I describe the field survey that we conducted and the resulting dataset, as well as the various methods used to partition it and compare both type of information. In Section 2.3, I explain the type of analysis that we performed to interpret the data. Besides comparing individual and relational information, our objective was also to review the existing tools allowing to interpret communities, and to show how they could be used on a concrete example. I conclude this chapter in Section 2.4 and discuss some related works.

2.2 Data Collection and Comparison

Generally speaking, the data collected during a survey can be considered according to two axes. I already mentioned the first one in Section 2.1: it opposes *individual* and *relational* information. The former refers to data describing only one person, whereas the latter concerns two persons, or more. On the second axis, we can distinguish three kinds of data, differing both by the nature of the information they convey and by how difficult and costly they are to obtain. First, *factual* information is the most easily accessible; it corresponds to acknowledged, generally publicly available, facts. For individual data, we can cite for example social status, gender, age, etc. For relational data, it can take the form of communication streams such as email exchanges, lists of collaborations, etc. Second, what we call *behavioral* information can either result from observations or be obtained directly by interrogating the persons of interest. For individual data, it describes how some person reacts to a given situation, whereas for relational data, the concern will be put on interactions between people, for instance by measuring the time that workers spend together in a firm. Third, *sentimental* information is related to feelings and thoughts. It is the most difficult to retrieve, since it cannot be accessed in other way than more or less direct questions, or very advanced physiological techniques [90, 185]. For individual data, it is for instance brands representations, firm image or products preferences. For relational data it corresponds to feelings (friendship, love, hate, admiration, etc.) that people have for each other. Sentimental relational data can be estimated through questions of the sociometric form, where each person is asked to list his acquaintances and to quantify the strength and orientation of their relationships [190]. This so-called sociometric approach is considered to be both the most efficient, in terms of quality of the retrieved relationships, and the most difficult to apply [56]. Extracting a social network requires relational data, which is globally more difficult and costly to gather than individual data [97]. Indeed, most available factual data focus on single persons (resumés, archives, surveys, etc.); observing interactions in a whole population obviously requires more resources than concentrating on a single individual; and making people speak about others can be an even more sensitive task than making them reveal personal details.

From this data-related difficulty regarding social networks extraction, a question arises: can the information conveyed by social networks be retrieved by other, less expensive, means? In this work, we tried to tackle this issue through the angle of group detection. We conducted a survey over a population of university students, and leveraged the collected data to partition the student population according to, on the one hand, the individual information, through cluster analysis, and on the other hand, the relational information, through community detection. The idea was to assess the agreement between the resulting student groups, in order to determine how much the two kinds of information overlap. In Section 2.2.1, I provide the context of our field study, and in Section 2.2.2 I briefly discuss the methods used to partition them and compare the resulting groups. The interested reader will find additional details in [VL98, VL48, VL23].

2.2.1 Field Survey and Data

In order to provide context to our results, I give here a brief description of the environment of our survey. At the time it was conducted, Galatasaray Üniversitesi (GSÜ) was a small Turkish public institution of about 2,000 students, located in Istanbul, near the Bosphorus. It offered a wide variety of courses (sociology, economics, international studies, management, philosophy, computer science, engineering, law, etc.) taught mainly in French. In Turkey, students enter universities after having passed a national competitive examination that determines the set of universities and departments they can choose to study in. However, GSÜ had a particular statute and could also recruit students directly from Turkish French-speaking high schools, thanks to a specific internal examination. Approximately two thirds of the students were undergraduates aiming at getting a *Lisans* diploma (equivalent to a Bachelor, or first year of Masters's degree in the EU system), the rest being Master and PhD students. Each department had a promotion of about 30 students per year. Community and cultural life was highly developed, with forty active sports clubs or cultural associations. There was a very strong feeling of belonging to a group, enhanced by the fact that the name Galatasaray also referred to a prestigious high school, a popular association football club, and various other cultural and sporting structures. After the university, very strong ties remained between GSÜ alumni, which usually helped each other professionally.

In this context, Jean-Michel Balasque and I conducted a study on the social network of the GSÜ students from that time. A university, and particularly GSÜ, can be considered as a relatively close system for students, in the sense that most of their friends also belong to it, making it an appropriate field of investigation. Our study was based on a survey taking place at several periods, in order to be able to study some of the network dynamics. The results presented in this chapter focus on the data obtained during the first phase of the overall research project, which took place during spring 2009 and involved 224 respondents, mainly at the Lisans level. This study was a very new and interesting experience for me: before that, I was used to take advantage of existing datasets, or to constitute new ones through software, like most computer scientists. Setting up such a real-world study turned out to be a difficult and demanding task.

We designed a questionnaire focusing on social and personal aspects, daily social interactions at the university, purchasing behavior, and favorite brands. Part of the required information was very personal and sensitive, so the data was anonymized. All 123 questions were designed to gather *individual* data, except one, which was dedicated to *relational* data. The individual data included the three types of information mentioned before. First, what we called *factual* information: age, gender, specialization, etc. Second, *behavioral* information, regarding the way students interact with others at GSÜ, but also their shopping habits, information sources, and buying behavior. Third, *sentimental* information, which concerns their feelings about the university, vision of their relationships with their friends, desires, hobbies, goals and favorite brands. I collectively refer to these data as the *attributes* in the rest of this chapter. For the sake of concision, I do not give an exhaustive list of all the attributes, but only a few illustrative examples in Table 2.1. The questionnaire, complete list of attributes, and anonymized raw data are all publicly available online [C10].

Data	Attribute	Type	Description
Factual	Gender	Dichotomous	Male vs. Female
	University department	Categorical	GSÜ has 22 departments
	University class	Ordinal	Current year: 6 different levels
	University grade	Real	Current grade of the student
	Entrance examination	Dichotomous	National vs. Internal
	High-School name	Categorical	193 establishments
	High-School category	Categorical	6 categories
	High-School city	Categorical	55 cities
	High-School specialization	Categorical	17 domains
	Club membership	Dichotomous	41 activities inside and outside GSÜ
Behavioral	Communication means	Categorical	13 communication means
	Social media	Ordinal	5 usage levels for 8 platforms
	Mobile phone brand	Categorical	17 brands
	Phone purchase date	Integer	Number of months since last purchase
	Digital player brand	Categorical	23 brands
	Hobby	Dichotomous	39 Yes/No propositions
Sentimental	Definition of *friend*	Dichotomous	8 Yes/No propositions
	Importance of advice	Ordinal	5 agreement levels for 3 topics
	Favorite clothing brands	Categorical	6 possible choices
	Desired items	Categorical	7 possible choices

Table 2.1: Some of the attributes obtained through the survey conducted at Galatasaray Üniversitesi.

Regarding the sole question concerned with relational data, we adopted a classic sociometric approach. It consisted in asking the students to name the peers they find the most important in their everyday life, and to quantify these relationships. We used these declarative answers to build the social network. Each vertex represents a student which either responded to the survey or was cited by a respondent (and sometimes both). Consequently, the network contains more vertices (622) than we had respondents (224), since some cited persons did not answer during this phase of the survey. Each edge is directed from the respondent towards the cited student, and has a weight corresponding to the score that the respondent associated to the relationship. The communities identified in such a network thus correspond to groups of people emotionally bound inside the university.

2.2.2 Comparison Partition

As mentioned before, in this part of the project we wanted to compare the information conveyed by the individual and relational data. To perform this comparison, we used groups detected based on each sort of data considered separately. We consequently partitioned the students using two approaches: cluster analysis over the attributes, and community detection over the network. For the community detection, we applied a selection of standard algorithms, covering a wide range of methods in order to assess their level of agreement and thereby ensure the stability of the identified community structures. The estimated community structures are well defined overall, with most algorithms reaching a modularity larger than 0.8. Using the Adjusted Rand Index (ARI) [130], we computed the similarity between these community structures, and observed little differences between the algorithms outputs. We also used a reasonable range of parameters and experimented with/without edge weights and directions, but found no significant difference either. We thus kept only the most consensual community structure for the rest of the process, which contains 22 communities, and is shown in Figure 2.2.

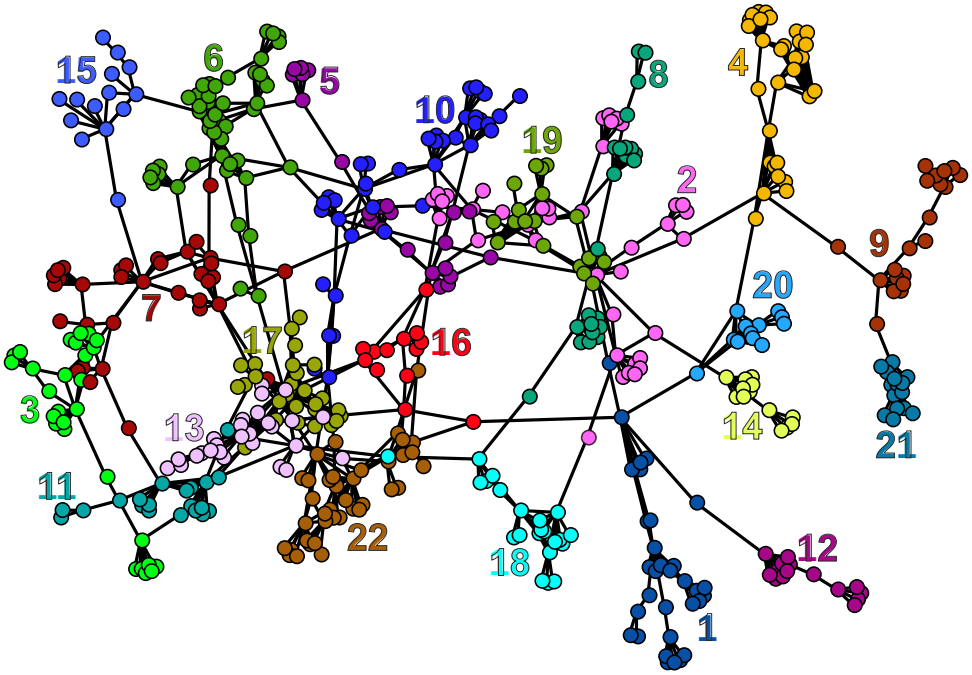

Figure 2.2: Community structure of the GSÜ social network resulting from our survey. Each one of the 22 communities is represented by a different color.

For the cluster analysis too, we applied a selection of standard algorithms. Moreover, in order to explore the discriminative power of the attributes, we performed the cluster analysis on all possible subsets of the attributes. One of the most important points for the cluster analysis is the choice of an appropriate dissimilarity function, allowing to properly compare the respondents. As shown by Table 2.1, in our case the data types of the attributes were very heterogeneous. We thus defined a composite similarity measure in order to take into account these data types, as well as the semantics of the attributes. For instance, when comparing the academic departments of two students, we considered the fact that certain departments belong to the same faculty and are therefore more similar because thematically linked.

We distinguished the clusterings depending on how much they fit the community detection in terms of ARI. For a given set of attributes, the clustering algorithms agreed even more than the community detection ones, with ARI scores above 0.9. However, when comparing clustering and community structure, most combinations of attributes led to an almost zero ARI. The best results in terms of community structure fit were obtained when performing the clustering over combinations of attributes describing the university department, class and grade, as well as the high-school. This led to ARI scores around 0.45, which can be considered as an intermediate value: for the record, the ARI upper bound is 1, which corresponds to a perfect

match. This relative similarity was somewhat supported by the number of clusters, which was around 20, and was therefore close to the 22 communities previously detected. However, it is worth stressing that the clusterings obtained with these attributes were not the best ones in terms of cluster quality, as quantified by an internal measure such as the Silhouette [204]. So, on the one hand, we were able to use individual data to identify clusters which were relatively close to (or rather: not significantly different from) the communities estimated from relational data. But on the other hand, the corresponding partitions also had a relatively poor quality when considering how well they separated the space of attributes. This indicates that, at the mesoscopic level of the groups of students, the individual attributes used during the clustering process contained at least a part of the information underlying the network structure, but that both types of data were nevertheless clearly distinct.

2.3 Attribute-Based Interpretation

In order to interpret the detected communities, we adopted two approaches: the first was based on descriptive methods and involved the use of both topological measures and vertex attributes (Section 2.3.1), whereas the second was based on predictive tools and aimed at predicting community membership based on attributes (Section 2.3.2). The interested reader will find more details on this analysis in [VL24].

2.3.1 Descriptive Methods

Two kinds of approaches can be performed to characterize the community structure. First, one can study the topology of the communities. This allows assessing the structural significance and quality of the community structure, but also starting the interpretation process, by discussing the similarities and differences observed between the communities, and by identifying vertices with specific roles. In [VL24], we leveraged a variety of topological measure for this purpose. Not only standard ones such as degree, average distance, and density; but also less known ones such as B-score to assess community significance [151], hub-dominance to measure how centralized a community is [150], as well as embeddedness [150], within-module degree and participation coefficient [115] to quantify the community-wise role of vertices. These results are described in [VL24], but in this manuscript I want to focus on the second approach, which relies on the exploitation of vertex attributes. It was guided by the structures identified during the first phase thanks to the topology of the network (communities, vertices of interest). It consisted in characterizing and discussing these structures in terms of the numerical or nominal data specific to the considered system and application domain.

Homophily The formation of communities, especially in social networks, can sometimes be explained by *homophilic* relationships, i.e. a tendency for vertices to connect with other vertices more or less similar to them, in terms of some attribute. The homophily is measured by constituting two series of attribute values for pairs of connected vertices, and computing the association between these two series. The measures proposed by Newman [177] amount to using Cohen's K and Pearson's correlation coefficient for categorical and numerical attributes, respectively. Homophily is generally processed over the whole network, but in our case we used it to characterize each community, as there was no *a priori* reason to assume they had the same attribute-related properties. Table 2.2 shows the values obtained for attributes gender (column *Gdr*) and class (*Cls*). Most communities had close to zero homophily for gender, except for a few ones for which it reached a value close to 0.5 (P_{10}, P_{13}, P_{17}). This means that students did not bond depending on their gender, except for these communities. Homophily values were more contrasted for the class attribute, with values either very close to 0 (P_3, P_9, P_{15}, P_{17}) or to 1 (P_8, P_{20}).

Dominant Attributes Communities can also be characterized by considering only the attribute values of their constituting vertices. Describing them on this basis is a general problem also occurring in classic cluster analysis [97], and there are consequently a number of existing tools for this purpose. As an example, we present in Table 2.2 some of the most characteristic attributes of our data. Of course, all communities were

Com.	#	Homophily		Class 1		Class 2		Dept 1		Hob	Phone	Plyr	Frd	Loan
		Gdr	Cls	#	Val	#	Val	#	Val	Val	Val	Val	avg	avg
P_1	32	0.24	0.30	25	P2	6	L1	7	BS	M	No	Ap	3.80	2.60
P_2	39	0.36	0.54	15	L2	12	L3	10	IE	C	No	–	3.00	1.78
P_3	28	0.12	0.00	25	L4	3	L3	12	BS	–	No	–	3.25	2.50
P_4	30	0.11	−0.04	26	P1	3	P2	6	So	M	No	Cr	2.78	1.78
P_5	23	0.19	−0.05	17	L2	–	–	15	RI	S	Sa	Ap	2.75	2.25
P_6	46	0.01	0.65	19	L2	18	L3	14	So	R	Sa	Ap	3.43	1.54
P_7	34	0.25	0.19	25	L3	9	L4	24	BS	C	Sa	Ap	3.78	2.67
P_8	23	0.17	0.74	12	P1	10	L1	9	IE	S	Sa	Ap	3.00	3.00
P_9	20	0.19	0.00	18	P1	2	P2	5	BS	–	No	Ap	2.17	1.67
P_{10}	39	0.51	0.55	31	L2	6	L1	17	BS	C	No	Ap	2.92	1.92
P_{11}	20	−0.15	0.16	17	L4	2	L3	11	IE	M	No	–	3.80	2.00
P_{12}	15	0.11	0.61	7	L1	7	L2	7	La	–	SE	Ap	4.00	1.50
P_{13}	28	0.46	0.60	14	L1	13	L2	12	Ma	M	No	So	3.64	1.64
P_{14}	13	0.00	0.56	6	L3	5	L4	7	CS	–	No	–	3.50	2.00
P_{15}	14	−0.12	0.00	13	L3	–	–	8	RI	–	–	–	2.67	3.50
P_{16}	13	−0.10	−0.03	11	L1	–	–	10	Ph	P	–	–	3.67	1.33
P_{17}	28	0.48	0.00	25	L4	3	L3	14	IE	C	No	Ap	3.11	2.22
P_{18}	22	−0.09	0.54	12	L3	–	–	–	–	–	No	–	3.71	1.17
P_{19}	20	−0.06	0.00	19	L1	–	–	9	Ma	–	No	–	3.71	1.71
P_{20}	12	−0.14	1.00	12	P1	–	–	2	–	R	No	Ap	2.14	1.71
P_{21}	15	−0.16	0.00	13	P2	2	L1	11	La	T	No	–	4.00	1.67
P_{22}	38	−0.05	−0.02	34	L1	2	P2	22	BS	C	Sa	Ap	4.00	2.00
Net.	552	0.25	0.78	124	L1	107	L2	96	BS	S	No	Ap	3.29	1.94

Table 2.2: Some of the attribute statistics obtained for the communities of Figure 2.2. See the text for details and abbreviations.

not characterized by the same attributes, which is why we selected factual, behavioral, and sentimental attributes. The *Class* columns describe the two most represented classes in each community. Sign # denotes the number of concerned students and *Val* shows the corresponding class value: Preparatory (P1–2) or Lisans (L1–4). The same goes with the *Dept* column, but for attribute department this time: Business Science (BS), Computer Science (CS), Economics (Ec), Industrial Engineering (IE), International Relations (IR), Law (La), Literature (Li), Mathematics (Ma), Philosophy (Ph), and Sociology (So). Column *Hob* shows the most popular hobby: music (M), cinema (C), sport (S), photography (P), reading (R), theater (T). Column *Phone* and *Plyr* respectively show the most widespread brands of mobile phones and digital players: Nokia (No), Samsung (Sa), Sony-Ericsson (SE), Apple (Ap), Creative (Cr), Sony (So). Columns *Frd* and *Loan* respectively indicate if students thinks they have their best friends in the university, and their inclination to take a loan. Both answers are expressed on a scale ranging from 1 (clear *no*) to 5 (clear *yes*), and the table contains average values.

For the sake of brevity, I focus my comments only on a few communities of interest. Community P_7 contains only students of third and fourth year of Lisans, but this holds for other communities too (P_3, P_{17}), so this property alone is not sufficient to characterize it. However, unlike community P_{17}, its dominating department is Business Science. Communities P_3 and P_7 can be distinguished by considering the former has no dominant hobby, and their dominant mobile phone brands are different. Students from P_{15} are more inclined to take a loan, they have the highest average score for that question (LI), which is an interesting result from the perspective or marketing. Community P_{16} contains almost exclusively first year Lisans students from the philosophy department, which is already discriminant when considering the other communities. Moreover, from an application point of view, it is interesting to notice the dominant hobby is photography and there is no dominant brand for electronic devices. Community P_{20} is very interesting because its students tend to think their best friends are not in the university (BF column): they have the lowest average score for the corresponding question. Nevertheless, this community is quite similar to others regarding hobbies and brands. This may be due to the fact that these students are in first year, often in a new city, far away from their family and high-school friends. A similar observation can be made on the communities containing a majority of first year students (e.g. P_9), and the effect tends to disappear for the communities of older students (P_{12},

P_{21}, P_{23}).

Statistical Dependence As we showed, inspecting the distribution of attribute values over communities allows detecting attributes of interest. This operation can be enhanced by a graphical representation of the network, such as illustrated by Figure 2.3 for the class attribute. It confirms our remarks regarding the relatively discriminant power of the class attribute, and the fact that it is not enough to uniquely characterize all communities. However, these somewhat subjective observations must be confirmed objectively in order to be relevant and useful. In other terms, one has to assess statistically the significance of the differences observed between the communities. For this matter, the selection of an appropriate statistical tool depends on the nature of the attribute of interest.

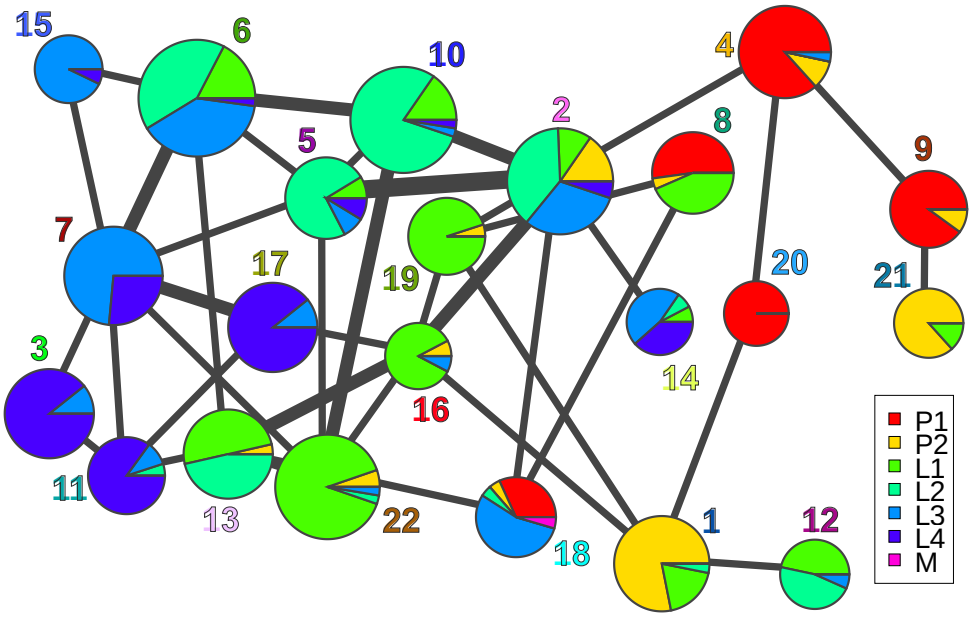

Figure 2.3: Class distribution in the community network. Each node represents a community from Figure 2.2, with matching number values and colors. The node diameters and link widths are proportional to the community sizes (expressed in number of students) and to the number of inter-community edges, respectively. Each pie chart represents the class attribute distribution in a community. Possible classes are Preparatory (P1–2), Lisans (L1–4) and Master (M).

If we want to determine whether community membership depends on some categorical attribute, then we need to assess the significance of the association between two categorical variables: the community and the attribute. The most popular test for this purpose is the well-known Pearson χ^2 test. Extensions exist to take several attributes into account, and association measures derived from this statistic (e.g. Pearson's Φ, Cramér's V) allow quantifying the *strength* of the association, by opposition to its significance. In our case, the associations between class and department on one side, and community membership on the other side, were significant ($p < 0.001$). This means that these attributes were generally good to characterize our communities, thereby confirming our previous observations.

In the case of a quantitative attribute, one can perform a classic ANOVA to test whether its average value differs significantly between communities (under the standard assumptions). Here too, extensions such as the factorial ANOVA allow considering several attributes at once. As an example, we performed an ANOVA on the sentimental attributes from Table 2.2 (best friend consideration and loan inclination). Our results showed that significant differences existed between communities for both attributes ($p < 0.05$). To identify precisely which communities differ, one has to perform a *post-hoc* test such as Tuckey's or Least Significant Difference tests. We applied the latter to our data, and it exposed several significant differences, but I limit my comments to the communities already mentioned in this section. It turns out the sentiment of having his best friend at the university was significantly lower in P_{20} compared to most other communities, especially P_{16} and P_7, so it can be considered as a characteristic of this community. Students from P_{15} were significantly

more inclined to take a loan or to delay a payment than most of the other communities, especially P_{16} and P_{20}, whose students were significantly inclined not to take a loan.

2.3.2 Predictive Methods

The descriptive tools presented in the previous section allow characterizing a community in terms of vertex attributes. This type of analysis is already interesting in terms of interpretation, but predictive methods can bring more precise models regarding the way communities are constituted. First, a model is estimated using the communities as reference groups and taking advantage of the available attributes. Its quality can be assessed in various ways, the simplest being to measure its prediction success rate on instances whose community is known. If the model is considered to fit the data well enough, it can be interpreted by considering which attributes it uses and how it combines them to estimate communities. I present here some results obtained with two families of statistical tools which can be used to build a predictive model: linear discriminant analysis (LDA) and sigmoid regression.

Linear Discriminant Analysis This model was initially designed to predict the value of a categorical variable using numerical attributes, and was later extended to categorical attributes under the name of discriminant correspondence analysis. The idea sustaining the method is close to PCA (Principal Component Analysis) and other dimension reduction methods. It consists in projecting the data in a new space maximizing the separation between the communities. The result of the projection is defined by a set of discriminant factors, corresponding to linear combinations of the input attributes. These factors are then used instead of the attributes to estimate the community of an object. Each factor can be characterized in terms of its discriminant power, and by interpreting the coefficients associated to the attributes in the corresponding function. LDA extensions exist for non-linear combinations, and for situations where the assumptions required by LDA are not met.

As an example, we tested all the numerical attributes related to our behavioral and sentimental data, which represented a total of 57 attributes. The model obtained with all attributes had 21 discriminant functions and could correctly classify 99.1% of the students. This very high rate has to be nuanced by the fact that the model included many functions, based on all 57 attributes. This can be interpreted as overfitting, and the interpretative value of this model was very weak. We processed separately the behavioral and sentimental attributes, and obtained models based on 21 functions using 31 attributes with a prediction rate of 70.5% for the former; and 21 functions using 26 attributes with a 69.8% prediction rate for the latter. These models were still large and difficult to interpret, and even removing a few attributes quickly decreased their prediction rates. These results indicate that estimating the community structure on the sole basis of the attribute values is a difficult task, as it require to consider most of the attributes.

Logistic Regression The logistic regression is able to predict the value of a dichotomous dependent variable based on numerical and dichotomous independent variables. It was extended to the prediction of categorical variables such as our communities. We applied a multinomial logit regression to the department and class attributes, which are both categorical. The model could be estimated with significantly good fit for both attributes (compared to a null model implementing the hypothesis of no influence of the attributes over the communities). The overall prediction rate was 46.8%, but varied very much depending on the community. For 4 communities (P_3, P_4, P_{17}, P_{22}), it was greater than 80% (with 89.3% as a maximum), and for 9 others (P_8, P_9, P_{11}, P_{12}, P_{14}, P_{15}, P_{19}–P_{21}) it was 0%. For the communities on which I previously focused (P_7, P_{16} and P_{20}) it was of 64.7%, 61.5% and 0%, respectively. This confirmed our previous observation: some communities can be efficiently characterized using these factual attributes, but they are not relevant for others. In marketing, this kind of information is at the origin of classic segmentation approaches. In our case, a marketing strategy based only on factual data could have very different effects depending on the targeted communities. It would certainly perform well on communities P_3, P_4, P_{17} and P_{22}, but be inefficient on communities such as P_{15}. Yet, we previously showed that this community was very attractive from a commercial point of view. The fact

that the network analysis managed to detect this community illustrates how it can be used to complement classic data analysis.

2.4 Conclusion

In this section, I described the work conducted in collaboration with business scientist Jean-Michel Balasque [VL98, VL48, VL23, VL24], which dealt with the problem of interpreting community structures in attributed graphs by leveraging the information conveyed by vertex attributes. Our first main contribution was to set up a field study in order to gather real-world data describing the social interaction and purchase habits of students from Galatasaray Üniversitesi. Our second contribution was to study how much the information conveyed by the network structure and the attributes match in term of the student groups they allow identifying. Our third contribution was a review of the methods allowing to interpret communities based on attributes, through descriptive and predictive tools. The survey that I conducted later with Günce Orman provides a more complete and recent overview of the tools available to characterize community structures [VL22]. Moreover, Günce Orman and I proposed another method to interpret community structures, but this time the additional information leveraged to perform this task is *time* instead of (and even in addition to) vertex attributes. This method thus applies to dynamic networks, and as such it is described in Chapter 4.

This Business Science collaboration is now over, but I am currently working on two other projects involving the interpretation of community structures based on attributes. Both are interdisciplinary works with historians. The first is conducted with Gaëtane Vallet and Catherine Wolff [S3, C1, VL88], and concerns the analysis of the social network of Roman emperor Trajan, whose reign spanned 98–118 AD. Each character in his entourage is described by a collection of attributes provided by historical sources, such as gender, titles, occupation, political position, and geographic origin. Figure 2.4.a displays the network with attribute *Hispanic*, which indicates whether or not a person comes from the Roman province of Hispania. We used tools similar to the ones described in Section 2.3 to answer certain historical questions, such as for instance the position of women in this network of power, or the existence of a suspected Hispanic faction in the entourage of Trajan. I come back to this work in Chapter 8, as it also involves different types of relationships, and therefore multiplex networks.

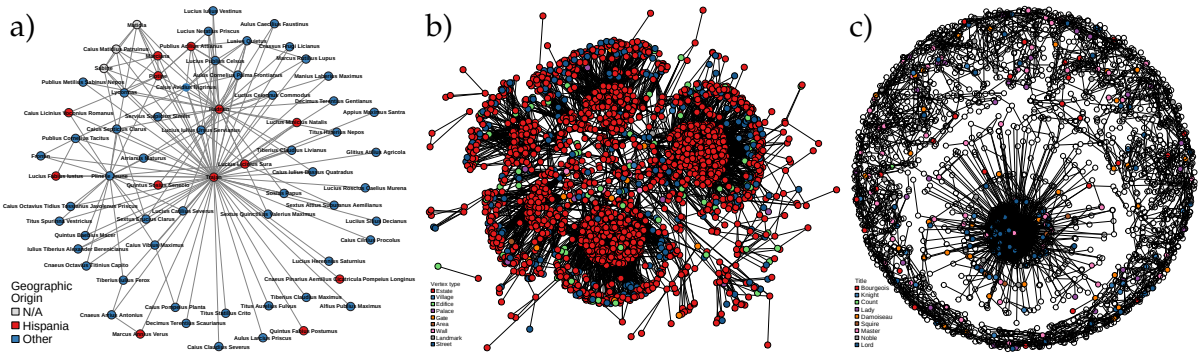

Figure 2.4: Ongoing projects related to attribute graphs. (a) Entourage of Roman emperor Trajan, with Hispanic characters in red and others in blue. (b) Confront network of medieval Avignon with the types of building in color. (c) Social network of medieval Avignon with the characters' honorific titles in color.

The second work takes place in the context of Margot Ferrand's PhD [100, S2], which is funded by the Agorantic[1] research federation (FR 3621), an interdisciplinary group of research laboratories based at Avignon Université. This is mainly a history thesis with geographic, NLP and graph aspects. It focuses on the cleaning and processing of medieval deeds concerning the then papal city of Avignon. Based on these documents, it is possible to extract various types of networks, including the so-called *confront* network (Figure 2.4.b), whose vertices model pieces of real estate described in the deeds, and edges represent their spatial proximity and/or

[1] https://agorantic.univ-avignon.fr/en/

relative spatial position. There are a number of problems to solve during the analysis of these data, many of them related to their *spatial* nature, which is why I come back to these aspects later in Chapter 6. Some of these problems concern the cross-analysis of the network structure and the attributes that describe the real estate in the deeds, e.g. type of real estate, name of the owner, and rental fee. In particular, the network can be partitioned in communities representing various areas of the city, and a number of historical questions are related to the characterization of these communities in terms of attributes: housing typology, distribution of rental fees, extent of lordships, and others. In addition, we extracted another interesting graph from the same texts: the network of social relationships between owners, and other persons mentioned in the deeds, such as witnesses and relatives (Figure 2.4.c). Its constituting vertices are also described by a variety of attributes, such as gender, name, geographic origin, titles and occupation. Here too, the idea is to partition this large network and study the resulting communities in order to identify the social groups at hand.

3

Classification and Attributed Networks

3.1 Context

Like the preceding one, this chapter is dedicated to attributed networks, however it focuses on problems and applications related to classification. This involves the classification of vertices in a given network, and the classification of whole graphs in a collection of networks. However, before turning to attributed graphs, I would like to briefly mention the work I conducted on classification problems related to *plain* graphs. First, at Galatasaray I advised Burcu Kantarcı during her Master's thesis [G11]. We characterized a collection of heterogeneous networks using a selection of topological measures, and performed an unsupervised classification of this dataset in order to detect clusters of similar networks, according to these descriptors. The idea was to generalize well-known works such as those of Watts and Strogatz [232] on small-worldness and Barabási and Albert [18] on the scale-free property. This work was presented at an international conference [VL40]. Another work concerned with unsupervised classification was the one I conducted in collaboration with Nicolas Dugué and Anthony Perez, and that I already mentioned in Chapter 2. Besides the generalization of the measures proposed by Guimerà and Amaral [114] to directed graphs, which I described in Section 2.1 as they rely on the graph community structure, the method that we proposed additionally involved the partitioning of the vertex set based on these measures. So unlike the work with Burcu Kantarcı, we wanted to classify vertices and not whole graphs, there. Our objective was to automatically detect functional community roles tailored for the data at hand, instead of using predetermined criteria as in the original method of Guimerà and Amaral [114]. We used our approach to characterize the position of a specific type of Twitter user called *social capitalist*, which has the particularity of implementing very specific strategies to increase their visibility on the social medium without providing content, thereby circumventing the criterion of relevance targeted by Twitter's ranking algorithm. This work was the object of several publications [VL39, VL12, VL72, VL71].

Let us now switch to *attributed* graphs. In the rest of this chapter, I focus on two tasks of supervised classification. The first, presented in Section 3.2, is a vertex classification task aiming at predicting the *offline* influence of Twitter users based on a number of individual and relational *online* characteristics. The second, presented in Section 3.3, seeks to detect abuse in online conversations based on the structure of these conversations. Finally, in Section 3.4 I discuss my current work and the perspectives that I identified regarding this topic.

3.2 Real-World Influence

The *Oxford Dictionary* defines influence as *"The capacity to have an effect on the character, development, or behavior of someone or something"*[2]. A number of works dealt with online influence, i.e. how much a user can exert such an ability though the considered social media [49]. Various companies even designed specific tools to measure online influence, such as Klout [198] and Kred[3]. However, very few authors have tried to assess offline influence, i.e. influence perceived in the real world, based on online activity: this was precisely the objective of the RepLab CLEF challenge in 2014 [10]. However, the methods proposed during this challenge only took advantage of the textual content of the tweets and of the information associated to the user profiles.

In the work that I present in this section, I collaborated with Jean-Valère Cossu and Nicolas Dugué in order to leverage the Twitter corpus provided for the RepLab challenge, but using simultaneously relational information taking the form of topological measures computed over the considered social network, and individual information corresponding to features extracted from the text produced by each user as well as his profile. This work was presented at an international conference [VL37] and published in an international journal [VL11]. In the following, I first describe briefly the dataset and tasks (Section 3.2.1) before turning to the methods that we proposed and applied (Section 3.2.2), and finally the main results that we obtained (Section 3.2.3).

3.2.1 Data and Tasks

The main goal of the RepLab challenge was to detect *offline* influence using *online* Twitter data. The RepLab dataset contains users manually labeled by specialists from Llorente & Cuenca[4], a leading Spanish e-Reputation firm. These users were annotated according to their perceived real-world influence, and not by considering specifically their Twitter account, although annotators only considered users with at least 1,000 followers. The annotation is binary: a user is either an *Influencer* or a *Non-Influencer*. The dataset contains a *training set* of 2,500 users, including 796 influencers, and a *testing set* of 5,900 users, including 1,563 influencers. The users are split between two mutually exclusive domains of activity: *Automotive* and *Banking*, which are balanced in terms of number of users. The dataset also includes the 600 last tweets of each user, which represents a total of 4,379,621 tweets. These tweets are written either in English or in Spanish. This dataset is publicly available online [5].

The task consisted in ranking the users in both activity domains, by decreasing order of influence. The organizers proposed a baseline consisting in ranking the users by descending number of followers. Basically, this consists in considering that the more a user has followers, the more he is expected to be influential *offline*. This baseline was directly inspired by *online* influence measurement tools. Performance was measured using the traditional *Average Precision* (AP), as described in Definition 3.2.1. It allows comparing an ordered vector (output of a submitted method) to a binary reference (manually annotated data). It was computed independently from the language, and separately for each domain. The *Mean Average Precision* (MAP), i.e. AP averaged over both domains, was used to express the overall performance.

Definition 3.2.1 (Average Precision) *For a given domain, the Average Precision (AP) is*

$$AP = \frac{1}{R} \sum_{r=1}^{R} Pre_r, \tag{3.1}$$

where R is the number of influencers in the considered domain, and Pre_r is the precision at rank r, i.e. when considering the first r users as returned by the ranking method.

[2] https://www.oxfordlearnersdictionaries.com/definition/english/influence_1
[3] https://www.home.kred/
[4] http://www.llorenteycuenca.com/
[5] http://nlp.uned.es/replab2014/

All participants of the original RepLab challenge based their ranking tools on the textual content of the tweets, and on the information contained in the user profiles. According to the official evaluation, the proposal from team UTDBRG [5] obtained the highest AP for the Automotive domain (0.721) and the best MAP (0.565). Team LyS [228] obtained the highest AP for the Banking domain (0.524). The scores obtained by team LIA [65] were 0.502 for Automative, 0.446 for Banking, and a MAP of 0.476. Note that I was not a part of this team, as I was not a member of the LIA yet. They later improved their results after the official end of the challenge [64], with scores of 0.803 (Automotive), 0.626 (Banking), and 0.714 (overall). The Followers baseline was lower than most of the submitted systems, achieving APs of 0.370 for Automotive and 0.385 for Banking, and a MAP of 0.378. The performance differences between domains were assumed to reflect a different level of difficulty for this ranking problem, as they were systematically observed for all methods.

Because the reference itself is only binary, the RepLab ordering task can alternatively be seen as a binary classification problem, consisting in deciding if a user is an influencer or not. This was not a part of the original challenge, but Ramírez-de-la-Rosa *et al.* [197] proposed a method to tackle this issue. To assess their results, they used the macro-averaged F-measure (MAF), i.e. the traditional F measure computed separately for each class, then averaged. They computed this measure for each domain separately, and averaged over both of them. Ramírez-de-la-Rosa *et al.* did not use any baseline to assess their results, but the imbalance between the influencer (31%) and non-influencer (69%) classes can be leveraged to build a baseline consisting in putting all users in the majority class (non-influencers). We call it MF-Baseline (most frequent class baseline), and it achieves a 0.50 MAF. Ramírez-de-la-Rosa *et al.* reached MAF scores of 0.696 (Automotive), 0.693 (Banking) and 0.694 (overall). Team LIA also tried to solve this extra task [64], but obtained lower results with an overall MAF of 0.40. Jean-Valère Cossu, which was a part of this team, later collaborated with Nicolas Dugué and myself in order to explore the use of structural information (in addition to the individual information already used in the LIA approach) in order to assess whether this would lead to improved results.

3.2.2 Methods

In the course of this work, we conducted a relatively comprehensive survey of the features used in the literature when performing classification tasks on data coming from social media activity, and especially the Twitter platform [VL11]. I do not list them exhaustively due to their number, but I can summarize the survey by describing the seven categories which we identified:

- *User profile*: features corresponding to fields of the user profile, such as the presence of a profile picture, an officially verified account, the URL of the personal Web page, the user's age, etc.
- *Publishing activity*: statistics describing how the user behaves on the platform, such as the number of tweets posted during the considered period, number of media sources contained in the tweets, average duration between two tweets, average number of mentions, proportion of geo-located tweets, etc.
- *Local connections*: features describing how the user is interconnected to the follower-to-followee network, with features like the incoming and outgoing degrees, PageRank centrality, size of the intersection between followers and followees, etc.
- *User interaction*: features characterizing how the user reacts to the behavior of other users, and vice-versa, such as: the proportion of retweets among the tweets published by the user, number of times a tweet published by the user was retweeted by others, number of other users' tweets marked as favorites, number of mentions by other users, etc.
- *Lexical aspects*: characteristics of the textual content produced by the user, like the size of the user's lexicon, number of hapaxes[6], number of named entities, $tf\text{–}idf$ vector, etc.
- *Stylistic traits*: non-lexical characteristics of the textual content, such as the average number of characters in a word or in a tweet, index of readability, usage of special characters, amount of hashtags and URLs, etc.
- *External data*: features describing some information retrieved from a source other than Twitter, such as the previously mentioned Klout and Kred scores, the number of hits when looking up the user's personal page on a search engine like Google, etc.

[6] In this context, a hapax is a word which is unique to a user.

In order to tackle the offline influence problem, we adopted an exploratory approach: we did not know *a priori* which of the surveyed features are relevant for the considered task, therefore we selected as many of them as possible. We discarded some features either because the necessary information was not present in the RepLab dataset and could not be retrieved online, or because it was not possible to handle them computationally. In the end, all seven categories were represented multiple times in our feature set. Note that we treated both activity domains separately, as any user appears in exactly one of them. In the following, I distinguish between *scalar* features, which describe one user through a single numerical value, and *vector* features, which represent the user through a collection of such values. To treat scalar features, we used non-linear classifiers under the form of kernelized SVMs (RBF, Polynomial and Sigmoid kernels) and logistic regression. We trained them using three distinct approaches: first with each scalar feature alone, second with all combinations of scalar features within each feature category, and third with all the scalar features at once.

The vector features are the ones allowing to characterize the textual content in a topical way (category *Lexical aspects*) through n-gram weighting. Briefly, this standard Information Retrieval approach consists in representing each sub-sequence of n words occurring in a text by its frequency (or a function thereof). This means that the text is described by a numerical vector, which is supposed to characterizes this text topically. This differs from the other features, which are scalars (i.e. unique values). In this work, we considered *unigrams*, i.e. term occurrences, and *bigrams*, i.e. term co-occurrences. Regarding the handling of the languages (French vs. Spanish), we tried two approaches: discarding this information (noted *Joint* in the rest of this section) vs. treating both languages separately (noted *Separated*).

For the unigrams, we used the standard *Term Frequency – Inverse Document Frequency* (tf–idf) approach [214], which we combined with the *Gini Purity Criterion* [223] (GPC). The *term frequency* $tf_d(t)$ of term t is simply its number of occurrences in document d. The *document frequency* $df(t)$ is the number of documents in the corpus that contain term t. The *inverse* document frequency is $idf(t) = \ln N/df(t)$, where N is the number of documents in the corpus. The Gini Purity Criterion is presented in Definition 3.2.2. The tf–idf score allows measuring the discriminant power of a term for a given document relative to a corpus, whereas the GPC indicates how much a term is spread over the document classes. In our case, these classes are *Influencer* vs. *Non-influencer*. In order to characterize a term relative to a document, we used the product $tf_d(t) \cdot idf(t) \cdot G(t)$, and relative to a class, we used $df_c(t) \cdot idf(t) \cdot G(t)$.

Definition 3.2.2 (Gini Purity Criterion) *The **purity** $G(t)$ of a term t is defined as*

$$G(t) = \sum_{c \in C} \left(\frac{df_c(t)}{df(t)} \right)^2 , \tag{3.2}$$

where C is the set of document classes, and df_c is the class-wise document frequency (number of documents in class $c \in C$ containing t) in the training set.

The general principle is then to compute one vector to represent a user, one to represent a class, and to compute the cosine similarity between them to determine to which class the user belongs. The way we obtained these vectors depended on three methodological choices, though. The first concerned the tweets that we used to describe a user: all the tweets available (noted *All*) vs. a selection of the most relevant (*Artex*). In the latter case, we extracted the 10% most informative tweets published by the user, thanks to the Artex method developed at the LIA [222]. The second methodological choice concerned what we consider to be a document. We tried two approaches: *User-as-Document* (UaD) [140] vs. *Bag-of-Tweets* (BoT). The former consisted in merging all the tweets of a user or a class into a single large document. The classification was performed by assigning a user to the most similar class, while the ranking depended on the similarity to the *Influencer* class. The latter (BoT) consisted in representing a user or class by their set of tweets. We then computed the similarity between each user BoT and each class BoT, then decided the classification outcome using a voting process. We considered two variants for this process, that constituted our third methodological choice: the first one (noted *Count*) consisted in keeping the majority class among the user's tweets, whereas

the second one (*Sum*) was based on the sum of the user's tweets similarity to the class *Influencer*. The ranking was obtained by ordering users depending on the count or sum obtained for the *Influencer* class.

For the bigrams, we first computed a co-occurrence matrix for each user or class, counting the number of times each pair of words appeared in their tweets. This is comparable to the UaD approach that we used with unigrams, as each user or class was represented by the concatenation of its tweets. Two users can be compared directly by computing the distance between their respective co-occurrence matrices. For this purpose, we computed the Euclidean distance between the linearized matrices and applied the k Nearest Neighbors method (k-NN) to separate influencers from non-influencers users by matching each user of the test collection to the k closest profiles of the training set. During this voting process, each neighbor vote was weighted using his similarity to the user of interest. The ranking was obtained by processing a score corresponding to the sum of the influential neighbors' similarities. Alternatively, we considered the co-occurrence matrix as the adjacency matrix of a co-occurrence graph, which we described through a selection of standard topological measures. This resulted in numeric vectors that we used as inputs for the same SVMs as for scalar features.

3.2.3 Results

In this section, I summarize our main results, starting with the classification task, then the ranking one. I compare our results to the best performances published in the literature. Note that the source code corresponding to this work is open source and available online [S13].

Classification The SVMs that we trained on scalar features, be it individually, by category, by combining categories, and all together, led to a performance lower than the baseline. This also held for the vector features extracted from the bigrams, which means that the centrality measures that we used to characterize the co-ocurrence networks were inefficient to find a non-linear separation of our two classes. Those results were confirmed by the logistic regressions: none of the trained predictors performed better than the most-frequent class baseline (all users as non-influencers). We also applied Random forests, which gave similar results. These classifiers usually perform very well for this type of task, so this seems to indicate that the scalar and bigram-based features simply did not convey the required discriminant information.

Feature and methods					Automotive	Banking	Average
Unigrams	User-as-Document	Separated	All		**0.833**	**0.751**	**0.792**
Unigrams	User-as-Document	Separated	Artex		0.829	0.745	0.787
Unigrams	Bag-of-Tweets	Separated	Artex	Sum	0.820	0.721	0.770
Unigrams	Bag-of-Tweets	Separated	All	Sum	0.817	0.709	0.763
Unigrams	Bag-of-Tweets	Separated	Artex	Count	0.796	0.719	0.757
Unigrams	Bag-of-Tweets	Separated	All	Count	0.786	0.702	0.744
Unigrams	User-as-Document	Joint	All		0.782	0.682	0.732
Unigrams	User-as-Document	Joint	Artex		0.773	0.672	0.722
Ramírez-de-la-Rosa *et al.* [197]					0.696	0.693	0.694
Unigrams	Bag-of-Tweets	Joint	All	Count	0.725	0.641	0.683
Unigrams	Bag-of-Tweets	Joint	All	Sum	0.725	0.641	0.683
MF-Baseline					0.500	0.500	0.500
Bigrams	Co-occurrence network				0.403	0.417	0.410

Table 3.1: Classification performances ordered by macro-averaged F-Measure. The highest scores are represented in bold. Column *Separated* vs. *Joint* indicates whether tweets were filtered based on language during the classifier training. Column *All* vs. *Artex* indicates whether the Artex methods was used to select only the most relevant tweets during training. Column *Count* vs. *Sum* denotes the voting scheme selected when using the Bag-of-Tweets model.

The results obtained with the various methods based on unigrams were largely above the baseline though, as displayed in Table 3.1. The classification performances are shown in terms of F-measure for each domain and averaged over domains. For comparison purposes, we also reported the baseline and the results obtained by Ramírez-de-la-Rosa *et al.* [197]. Our results confirmed that the Banking domain appeared more difficult to classify than the Automotive one. Without language specific processing (*Joint* method), the Bag-of-Tweets

approach obtained state-of-the-art results, while the User-as-Document one outperformed all existing methods reported for this task, up to our knowledge. For both approaches, the performances were clearly improved when processing the languages separately (*Separated* method). This might be due to the fact that certain words were used in both languages, but in different ways.

In the case of BoT, summing the votes (*Sum*) improved the performance compared to simply counting them (*Count*). This effect was more or less marked depending on the way the languages were treated: no effect for *Joint*, strong effect for *Separated*. The domain also affected this improvement, which was much smaller for Banking than for Automotive. This could indicate that users behaved differently, in terms of how they redacted tweets, depending on their domain. This would be consistent with our assumption regarding the use of different terminology by influential users of distinct activity domains.

The tweet selection step (approach *Artex*) affected differently the BoT and UaD methods. For the former, there was an increase in performance, compared to using all available tweets (approach *All*). Moreover, this increase was noticeably higher for Banking than for Automotive, which supported our previous observation regarding editorial differences between domains. The latter method (UaD), on the contrary, was negatively affected by *Artex*. This can be explained in the following way: the tweet selection is a filter step, which reduces the noise contained in the user's Bag-of-Tweets, thus causing an increase in performance. However, the User-as-Document method already performs a relatively similar simplification, lexically speaking, so the improvement is much smaller, or can even turn into a degradation.

Ranking The results obtained for the ranking task are displayed in Table 3.2 in terms of average precision, for each domain and averaged over domains. Again, one can observe that except in a very few cases, all scores are lower for the *Banking* domain than for the *Automotive* one. Like for the classification, we included the best published performances (UTDBRG and LyS) as well as the baseline.

Feature and methods					Automotive	Banking	Average
Unigrams	User-as-Document	Separated	All		**0.803**	0.626	**0.714**
Unigrams	Bag-of-Tweets	Separated	All	Sum	0.779	0.628	0.703
Unigrams	Bag-of-Tweets	Separated	Artex	Sum	0.774	**0.633**	0.703
Unigrams	User-as-Document	Separated	Artex		0.782	0.623	0.702
Unigrams	Bag-of-Tweets	Separated	Artex	Count	0.778	0.612	0.695
Unigrams	Bag-of-Tweets	Separated	All	Count	0.762	0.592	0.677
Unigrams	User-as-Document	Joint	All		0.735	0.538	0.636
Unigrams	User-as-Document	Joint	Artex		0.722	0.547	0.634
Unigrams	Bag-of-Tweets	Joint	All	Sum	0.699	0.526	0.612
Unigrams	Bag-of-Tweets	Joint	All	Count	0.626	0.504	0.565
UTDBRG	Aleahmad *et al.* [5]				0.721	0.410	0.565
LyS	Vilares *et al.* [228]				0.602	0.524	0.563
Total number of tweets					0.332	0.449	0.385
Best combination of scalar features					0.424	0.338	0.381
RepLab Baseline					0.370	0.385	0.378
Bigrams	Co-occurrence network				0.298	0.300	0.299
Klout score					0.304	0.275	0.289

Table 3.2: Ranking performances ordered by mean average precision. The highest scores are represented in bold. Cf. Table 3.1 for a description of the columns related to the methods.

The best results based on the scalar features are presented in row *Best combination of scalar features*, and were obtained by combining categories *User activity*, *Profile fields*, *Stylistic aspects* and *External data*. However, the resulting MAP is just above the baseline, and far from the state-of-the-art approaches. We also considered each scalar feature separately. The best performance was obtained with the number of tweets posted by each user: although its MAP is just above the baseline, the performance obtained for the *Banking* domain is above UTDBRG, the previous state-of-the-art result. Thus, we may consider this feature as the new baseline for this specific domain. All others similarly processed features remained lower than the official baseline. The results obtained for the Klout score reflected very poor rankings. This is very surprising, because this

commercial product was precisely designed to measure influence in general (i.e. both on- and offline). The results obtained with the bigrams, using the direct comparison of co-occurrence networks were only slightly better than the Klout Score.

The rest of the results presented in Table 3.2 are the best that we obtained for unigrams. The Bag-of-Tweets method obtained an average state-of-the-art performance, while the User-as-Document method reached a very high MAP, even larger than the state-of-the-art, be it domain-wise (for Automotive and Banking) or in average. Compared to the classification results, the performances of the BoT and UaD methods are tighter, but the latter still dominates the former, though. Again, both methods got better results when the languages were treated separately (approach *Separated*). The BoT method still appears to perform better when using the *Sum* decision strategy (instead of *Count*).

Despite this performance reproduction point, our NLP-based methods reached higher scores than state-of-the-art works, for both classification and ranking. This indicates that typical SNA features classically used to detect spammers, social capitalists or influential Twitter users, were not very relevant to detect *offline* influencers based on *online* activity. In other terms, these typical features might be efficient to characterize influence perceived on Twitter, but not outside of it. Compared to other previous content-based methods, our approach consisting in representing a user under various forms of tweet-based bags-of-words also gave very good results. In particular, our User-as-Document method was far better than the best state-of-the-art approaches for both classification and ranking tasks. We suppose that the way a user writes his tweets is related to his offline influence, at least for the studied domains. However, our attempt to extend this occurrence-based approach to a co-occurrence-based one using graph measures did not lead to good performances.

3.3 Abuse Detection

The work described in this section is a collaboration with Richard Dufour, in the context of an industrial project involving the video game development studio *Castle Prod*[7]. It started in 2015 with Étienne Papégnies and went on with Noé Cécillon's Master thesis [G2] and ongoing PhD thesis [D2]. Castle Prod had developed an MMORPG (Massively multiplayer online role-playing game) called *SpaceOrigin*[8], a type of online game that involves, by construction, a large number of players. These players were heavily communicating in-game through various communication means, some of which required some moderation. This task was performed manually by the staff and some players, and Castle Prod wanted to at least partially automate the process. This task is traditionally formulated as a binary classification problem consisting in distinguishing between abusive and non-abusive messages among the thread constituting a conversation.

The literature provides two main approaches to solve this problem. The first relies on the *content* of the targeted messages only, generally through standard NLP and information retrieval approaches such as predefined rules [215], $tf\text{-}idf$ scores [83], lists of badwords [173], or more recently text embeddings [52] and CNNs [84]. However, it is very common for users to voluntarily obfuscate message content, in order to bypass automatic systems by making the abusive content difficult to detect [128]: we observed this behavior in the SpaceOrigin corpus. This is a major limitation of exclusively content-based methods. The second approach focuses on the context of the messages, as the reactions of other persons is generally beyond the control of the abuser. Some authors use the text surrounding the message of interest [239], others leverage information extracted from the user's profile [15], or design models of abusive users [55].

We decided to adopt a hybrid approach by combining both the textual and contextual aspects in our method. We first experimented only with an exclusively content-based approach [VL65, VL33], exploring a variety of standard features. We then turned to an exclusively contextual approach, by extracting and leveraging the conversational network of the concerned message threads [VL76, VL2, VL30, VL63, VL7]. By comparison to the work presented in Section 3.2, where the focus was on the classification of individual vertices, here we wanted to classify *whole* graphs representing conversations. Finally, we experimented with the fusion of both approaches [VL27]. One limitation of our work is that the SpaceOrigin corpus is closed, and cannot be used

[7] http://www.castleprod.com/
[8] http://www.spaceorigin.fr/

by other authors for comparison purposes. We could still share the content-less conversational graphs that we extracted from the dataset [C4], but not the textual content itself. Therefore, in the process of developing and assessing our methods, we constituted a new corpus based on Wikipedia editorial forums [C3, VL25].

In the rest of this section, I first briefly present the main approaches that we explored relative to content-based methods (Section 3.3.1). I then turn to the graph-based method, including the graph extraction process and the characterization of conversational graphs (Section 3.3.2). I explain how we performed the classification, and merged both approaches to constitute a hybrid tool leveraging simultaneously text and context (Section 3.3.3). Finally, I present the main results obtained with all these methods (Section 3.3.4).

3.3.1 Textual Content

In order to distinguish abusive messages from non-abusive ones, we experimented with a number of text-related features, some standard in NLP and some tailored to the considered task and dataset. We distinguished three main categories: low-level morphological features, higher-level lexical features, and user-related features. As in many NLP tasks, the preprocessing was also an important step of the process. I briefly describe these different aspects in this section.

Pre-Processing We experimented with two levels of pre-processing phases. In the *basic* version, we lower-cased the raw text and tokenized it using spaces and punctuation. In the *advanced* version, we performed some additional steps. First, we reverted elision. Second, we ran a deobfuscation pass by mapping hexadecimal or binary encoded text in the message back to ASCII. This is highly specific to the considered online community, because users sometimes encode part of their messages in that way. Third, we converted each URL into a sequence of tokens. The first describes whether this URL is an internal hyperlink (to a server hosting the community) or an external one. The rest are words that could possibly be extracted from the name of the web page. Fourth, we applied a standard stemming method.

Morphological Features Based on a review of the abusive messages from the SpaceOrigin dataset, we could identify some characteristics hinting at abuse. Some of these features were computed *before* the pre-processing. First, abusive messages are usually either kept short, or extremely long, which is symptomatic of a massive copy/paste. We therefore used the message length, expressed in number of characters and words. Second, abusive and obfuscated messages tend to make use of unusual characters. We consequently used a feature indicating which classes of characters appear in the message: letters, digits, punctuation, and others. Third, as uppercase letters are used to denote yelling, we defined a feature representing letter case. Fourth, abusers tend to use copy-paste a lot, which is why we used the compression ratio between the original message and its compressed version obtained with the Lempel–Ziv–Welch (LZW) algorithm [234]. Fifth, by counting the number of distinct characters in the message, we could detect the use of binary or hexadecimal obfuscation, as well as the overuse of punctuation. Sixth, when the same character appeared consecutively more than twice, we used the number of extra characters as a feature reflecting the transcription of certain oral expressions (e.g. "nooooon").

Lexical Features Our first lexical feature was the output of a Naive Bayes classifier trained upon binary Bag-of-Words (BoW) representations of the messages. Second, we used the word length, which is a component of the Automated Readability Index (ARI) [209]. The ARI measures how proficient someone is at creating text documents, as many abusive messages are not well written. Third, we considered the number of unique words in the message, as longer messages are likely to be more constructive whereas abusive ones are generally straightforward. Fourth, we used the sum of the $tf\text{–}idf$ scores over the message words. We computed such a value relative to each class taken separately. Fifth, we computed sentiment scores thanks to an augmented version of the lexicon presented in [54]. Sixth, we counted the number of bad words based on a list that we crafted ourselves. Seventh, we defined a so-called *business score* based on the mention of entities related to the game, like buildings, military units, and war terms. The rationale here is that we noticed that abusive messages tend to be strictly personal attacks with no pretense of roleplay and no mention of game jargon.

User-Related Features The first user-related feature was based on the observation that abusive comments tend to trigger a significant response from the community. It was thus related to the number of answers from unique users. Second, we investigated if the abusiveness of a user's message could be detected by considering the effect it had on the other users participating in the same conversation. To do this, we compared the writing behavior of the other users before and after the apparition of the targeted message. We modeled this behavior through a user-specific word n-gram Markov chain, constructed on the train data, and used to compute how likely some text is to have been generated by the considered user. Third, the previous feature makes sense only if the considered users have sufficient history: we defined a limit of at least 300 bigrams in order to reflect the fulfillment of this constraint.

3.3.2 Conversational Graphs

We extracted networks representing conversations between users through a textual discussion channel. They took the form of weighted graphs, in which the vertices and edges represented the users and the communication between them, respectively. An edge weight corresponded to a score estimating the intensity of the communication between both connected users. Each network was defined relatively to a *targeted message*, since the goal of this operation was to provide the features used to classify the said message.

Context Period Our first step was to select the message used to extract the conversational network. For this purpose, we defined the *context period* as a sequence of messages spanning equally before and after the targeted message. Figure 3.1.a shows an example of context period, representing each message as a vertical rectangle, including the targeted message in red. At this stage, we could create the graph vertices, each one representing a user which authored at least one of these messages.

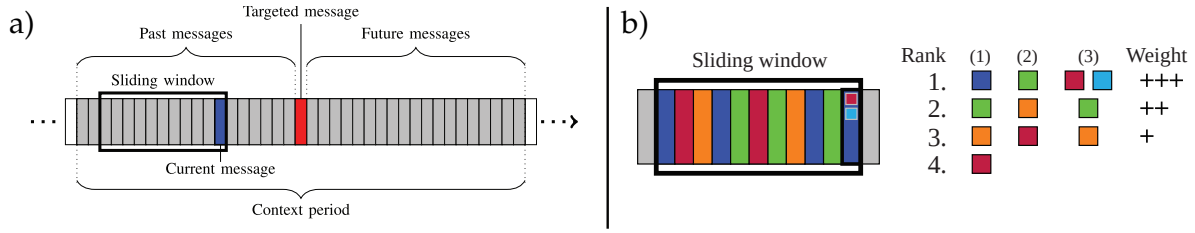

Figure 3.1: (a) Sequence of messages (represented by vertical rectangles) illustrating the various concepts used in our conversational network extraction process. (b) Example of sliding window and computation of the corresponding receivers' scores. Each color represents a specific user.

Besides the network extracted over the whole context period (before and after the targeted message), which we called the *Full* network, we also considered two additional networks. We split the period in the middle, right on the targeted message, and extracted one network over the messages published in the first half (*Past messages*), called *Before* network, and one over the other half (*Future messages*), called *After* network. Both of those smaller networks also contain the targeted message. For a prediction task, *i.e.* when using only past information to classify the targeted message, one would only be able to use the *Before* network. However, in a more general setting, all three networks (*Before*, *After* and *Full*) could be used.

Weight Assignment The second step was an iterative process aiming at adding edges to the graphs. It consisted in sliding a window over the whole context period, one message at a time, and updating the edges and their weights according to the process described next. The size of this *sliding window* was expressed in terms of number of messages, and it was fixed. It can be viewed as a focus on a part of the conversation taking place at some given time. It is shown as a thick black frame in Figure 3.1.a. We called *current message* the *last* message of the window taken at a given time (represented in blue), and *current author* the author of the current message.

Our assumption is that the current message is destined to the authors of the other messages present in the considered sliding window. Based on this hypothesis, we listed the authors of the messages currently

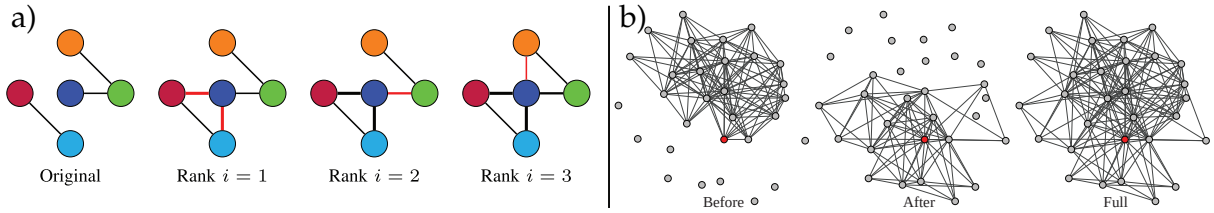

Figure 3.2: (a) Update of the edges and weights of the conversational graph corresponding to our ongoing example. The first graph displays the state before the update, and each remaining one corresponds to one rank in the user list. (b) Example of the 3 types of conversational networks extracted for a given context period. The author of the targeted message is represented in red. For readability reasons, weights and directions have been omitted.

present in the sliding window, and ordered them by their last posted message. Only the edges towards these users would be assigned weights, using a decreasing function of their rank. Figure 3.1.b displays an example of sliding window, in which the colors of the messages represent their authors. So, in this specific case, 4 different users participate in the conversation, as listed anti-chronologically in column (1). Obviously, a user is not writing to himself, so we remove the current author from the list, resulting in list (2). If a user was explicitly mentioned in a message, he was moved up the top of the list, resulting in list (3) in the figure. We could then update the graph by creating an edge between the current author and each user in the ordered list, or by increasing its weight if the edge already existed. Figure 3.2.a shows the result of this update based on our previous example from Figure 3.1.b. A real-world example of the three conversational networks obtained by applying our extraction method is shown in Figure 3.2.b, which corresponds to an abusive comment from our dataset.

Topological Measures Our graph-related classification features were all based on topological measures, according to a selection that allowed characterizing graphs in various ways. We adopted an exploratory approach and considered a large range of topological measures, focusing on the most widespread in the literature. Some of these measures can optionally handle edge directions or edge weights: we considered all practically available variants, in order to assess how informative these aspects of the graph are relatively to our classification problem.

One can distinguish topological measures in terms of scale and scope. The *scale* depends on the nature of the characterized entity: vertex, subgraph or graph. In our case, we focused only on *vertex-* and *graph-focused* measures: the former allows focusing on the author of the targeted message, whereas the latter describes the whole conversation, but we do not have any subgraph to characterize. The *scope* corresponds to the nature of the information used to characterize the entity: *microscopic* (interconnection between a vertex and its direct neighborhood), *mesoscopic* (structure of a subgraph and its direct neighborhood), and *macroscopic* (structure of the whole graph).

We processed all the features for each of the 3 types of networks (*Before*, *After*, and *Full*). The interested reader will find the exhaustive list and description of our features in [VL7].

3.3.3 Experimental Protocol

The SpaceOrigin corpus contains 4, 029, 343 messages exchanged by French-speaking players. Among them, 779 messages have been flagged by one or more users as being abusive, and subsequently confirmed as abusive by the human game moderators: they constituted our *Abuse* class. In order to keep a balanced dataset, we further extracted the same number of messages at random from the ones that have not been flagged as abusive. This constituted our *Non-abuse* class. Each message, whatever its class, was associated to its surrounding context (i.e., messages posted in the same thread).

We used the SVM classifier implemented in Sklearn under the name SVC (C-Support Vector Classification). Because of the relatively small dataset, we set-up our experiments using a 10-fold cross-validation. Each fold was balanced between the Abuse and Non-abuse classes, 70% of the dataset being used for training and 30%

for testing. We performed the training through single-threaded calculations performed on an Intel Xeon CPU E5-2620 v3s 2.5 GHz with 15 MB cache. Our source code [S5] as well as the conversational networks that we extracted from the raw data [C4] are available publicly online.

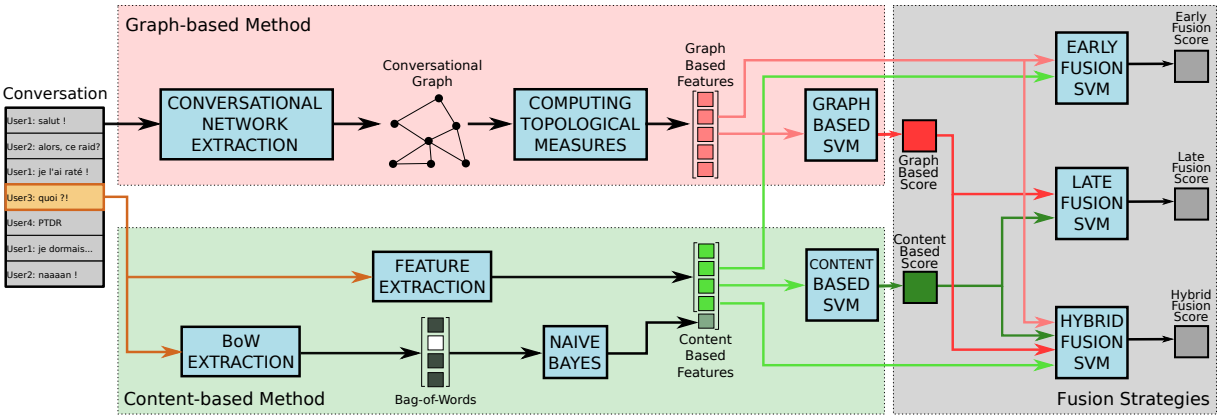

Figure 3.3: Representation of our processing pipeline. The green part corresponds to the processing of text-based feature, the red part to graph-related features, and the gray part to the fusion strategies.

First, we trained the classifier and assessed its performance separately on the text- and graph-based features, as represented in green and red in Figure 3.3, respectively. Then, based on the assumption that the content- and graph-based features convey different information, we studied how combining them affected the results. We considered three different strategies to perform this combination, as described in gray in the same figure. The first strategy followed the principle of *Early Fusion*. It consisted in constituting a global feature set containing all content- and graph-based features, then training a SVM directly using these features. The rationale here is that the classifier has access to the whole raw data, and must determine which part is relevant to the problem at hand. The second strategy was *Late Fusion*, and we proceeded in two steps. First, we trained separately two classifier to obtain two scores corresponding to the output probability of each message to be abusive given by the content- and graph-based methods, respectively. Second, we fetched these two scores to a third SVM, trained to determine if a message is abusive or not. This approach relied on the assumption that these scores contain all the information the final classifier needs, without the noise present in the raw features. Finally, the third fusion strategy can be considered as *Hybrid Fusion*, as it sought to combine both previous proposed ones. We created a feature set containing the content- and graph-based features, like with *Early Fusion*, but also both scores used in *Late Fusion*. This whole set was used to train a new SVM. The idea was to check whether the scores conveyed certain useful information present in the raw features, in which case combining scores and features should lead to better results.

3.3.4 Main Results

We performed a thorough evaluation of the effects of the parameters controlling the extraction of our features, especially those concerning the conversational graph (e.g. context period size, sliding window size). As we adopted an exploratory approach consisting in including as many features as possible in our experiments, we also conducted a correlation study to determine which one were interchangeable, and a feature study consisting in identifying the most discriminating ones. All those experiments and the corresponding results are described in detail in [VL33, VL7], and I only summarize our main results here.

Extraction Parameters Regarding the graph extraction parameters, it appeared that the sliding window size did not affect the classification results, as shown by Figure 3.4.b. On the contrary, the context period size had a clear effect, as illustrated by Figures 3.4.a and 3.4.c. We had assumed that the performance would increase until the latter reached a size corresponding roughly to the typical conversation size, then stay constant. However, the figures show that this is not the case: the performance keeps on growing (even if slowly) with the context period size. A manual investigation revealed that conversation boundaries are not well defined, and that conversion length varies much. There is consequently no typical duration for a conversation.

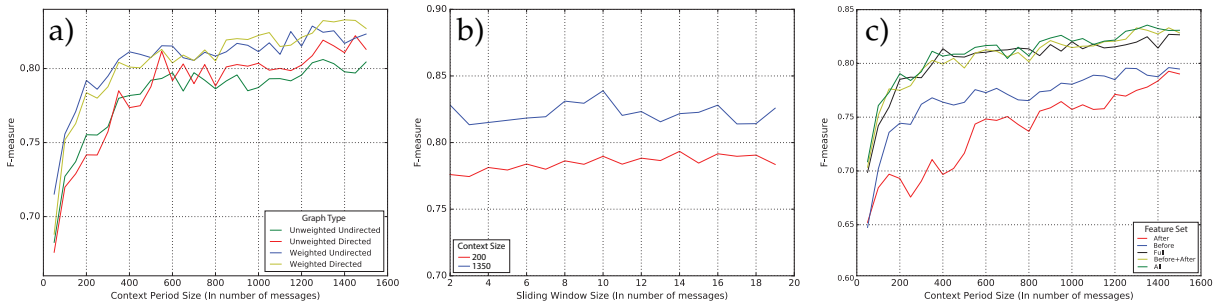

Figure 3.4: Classification performance expressed in F-measure, depending on various extraction parameters. (a) as a function of the context period size, for the (un)directed and (un)weighted feature sets. (b) as a function of the sliding window size, for 2 context period sizes (200 and 1,350 messages). (c) as a function of the context period size, for the 4 considered feature sets (*Before*, *After*, *Full*, *All*) as well as a combination of the first two (*Before+After*).

Regarding the sliding window size, Figure 3.4.a shows that the performance was slightly higher when we used more of the information encoded in graph, i.e. edge direction and weights. Moreover, weights had a stronger effect than directions. Finally, Figure 3.4.c focuses on the type of graph used to build the features: only the *Before* graph, only *After*, only *Full*, both *Before* and *After*, and all three of them. Surprisingly, our results show that what happens before the targeted message is more important than what happens after. Still, the information conveyed by these graphs is partially complementary, as the performance is significantly improved when considering them simultaneously. The performance level is similar if one considers all three graphs, or just the *Full* graph alone, which indicates that the latter still reflect the part of the conversation dynamics that is relevant for this task.

Method	Top Features	Graph	Weights	Directions	Scale
Content-Based	Naive Bayes	–	–	–	–
	tf–idf Abuse Score	–	–	–	–
	Character Capital Ratio	–	–	–	–
Graph-Based	Coreness Score	Full	–	Incoming	Graph
	PageRank Centrality	After	Unweighted	Directed	Vertex
	Strength Centrality	Full	Weighted	Outgoing	Vertex
	Vertex Count	Full	–	–	Graph
	Closeness Centrality	Before	Weighted	Outgoing	Graph
	Closeness Centrality	Before	Weighted	Outgoing	Vertex
	Authority Score	Before	Weighted	Directed	Graph
	Hub Score	Before	Unweighted	Directed	Vertex
	Reciprocity	After	–	Directed	Graph
	Closeness Centrality	After	Weighted	Undirected	Vertex
Early Fusion	Coreness Score	After	–	Outgoing	Graph
	Coreness Score	Before	–	Incoming	Graph
	Eccentricity	Before	–	Incoming	Graph
	Naive Bayes	–	–	–	–
Late Fusion	*Content-Based* TF ∪ *Graph-Based TF*	–	–	–	–
Hybrid Fusion	Graph-based output	–	–	–	–
	Content-based output	–	–	–	–
	Strength Centrality	After	Weighted	Outgoing	Vertex
	Coreness Score	Before	–	Incoming	Graph

Table 3.3: Top features identified for our 5 methods: considering content- and graph-based method separately, and using them together through our three fusion strategies.

Without Fusion In order to identify the most discriminative features, we applied an iterative method based on the *Sklearn* toolkit, which allowed us to fit a linear kernel SVM to the dataset and provide a ranking of the input features reflecting their importance in the classification process. Using this ranking, we identified the least discriminant feature, removed it from the dataset, and trained a new model with the remaining

features. The impact of this deletion is measured by the performance difference, in terms of F-measure. We reiterated this process until only one feature remained. We called *Top Features* (TF) the minimal subset of features allowing to reach 97% of the original performance (i.e. when considering the complete feature set). We applied this process to both baselines and all three fusion strategies. The *Top Features* obtained for each method are listed in Table 3.3. The last four columns precise which variants of the graph-based features are concerned. Indeed, as explained before, most of these topological measures can handle/ignore edge weights and/or edge directions, can be vertex- or graph-focused, and can be computed for each of the three types of networks (*Before*, *After* and *Full*).

There were three *Content-Based TF*. The first was the *Naive Bayes* prediction, which is not surprising as it comes from a fully fledged classifier processing binary BoWs. The second was the $tf\text{-}idf$ *score* computed over the *Abuse* class, which shows that considering term frequencies indeed improve the classification performance. The third was the *Capital Ratio* (proportion of capital letters in the comment), which is likely to be caused by abusive message tending to be shouted, and therefore written in capitals. The *Graph-Based TF* helped detecting changes in the direct neighborhood of the targeted author (Coreness, Strength), in the average distance-based vertex centrality at the level of the whole graph (Closeness), and in the general reciprocity of exchanges between users (Reciprocity). After manual investigation, it appeared that the Closeness was generally higher for the abuse class. This means that the average distance between the targeted author and the rest of the graph decreases in case of abuse. This user becomes less peripheral (or more central), and the same goes for the other users of the graph (in average). This fits in quite well with assumptions about how abuse impacts a discussion: an abuser would tend not to be peripheral in a conversation, while we can reasonably assume that the other participants will be piling on him or her and, therefore, be less peripheral themselves.

Method	**Number of Features**	**Total Runtime**	**Average Runtime**	**Precision**	**Recall**	***F*-measure**
Content-Based	29	0:00:52	0.02s	0.79	0.84	0.81
Content-Based TF	3	0:00:21	0.01s	0.76	0.83	0.79
Graph-Based	459	8:19:10	7.56s	0.90	0.88	0.89
Graph-Based TF	10	0:14:22	0.03s	0.89	0.85	0.87
Early Fusion	488	8:26:41	7.68s	0.91	0.89	0.90
Early Fusion TF	4	0:11:29	0.17s	0.89	0.87	0.88
Late Fusion	488 (2)	8:23:57	7.64s	0.94	0.92	0.93
Late Fusion TF	13	0:15:42	0.24s	0.92	0.90	0.91
Hybrid Fusion	490	8:27:01	7.68s	0.92	0.90	0.91
Hybrid Fusion TF	4	0:16:57	0.26s	0.91	0.89	0.90

Table 3.4: Comparison of the performances obtained with our 5 methods, by considering all features and only the Top Features (TF). The total runtime is expressed as *h*:*min*:*s*.

Table 3.4 presents the Precision, Recall and F-measure scores obtained on the *Abuse* class. It also shows the number of features used to perform the classification, the time required to compute the features and perform the cross validation (*Total Runtime*) and to compute one message in average (*Average Runtime*). When comparing the performance obtained with only content- and graph-based features, it appears clearly that the latter got much better performance. This is a major result, as it shows that the sole structure of the conversation is enough to efficiently detect abuses, without considering at all the content of the exchanged messages. Put differently, our graph-based method overcame a standard text-based approach, even though it is completely language-independent. The computational cost of the graph-based method is noticeably higher though, mainly because of the feature computation step. This is due to the number of features, some of which are very demanding, computationally speaking. Besides a better understanding of the dataset and classification process, one interesting use of the TF is that they allow decreasing the computational cost of the classification. Here, the drop is substantial for graph-based features (14 minutes instead of 8 hours), even if the total runtime stays largely above that of NLP methods.

With Fusion When performing the fusion, our first observation is that we got higher F-measure values compared to both baselines, independently from the fusion strategy (Table 3.4). This confirmed what we expected, i.e. that the information encoded in the interactions between the users differs from the information conveyed by the content of the messages they exchange. Moreover, this showed that both sources are at least partly complementary, since the performance increased when merging them. On a side note, the correlation between the score of the graph- and content-based classifiers was 0.56, which is consistent with these observations.

It appeared that *Late Fusion* performed better than the other fusion strategies, with an F-measure of 0.93. This is a little bit surprising: we were expecting to get superior results from the *Early Fusion*, which has direct access to a much larger number of *raw* features (488). By comparison, the *Late Fusion* only gets 2 features, which are themselves the outputs of two other classifiers. This means that the *Content-Based* and *Graph-Based* classifiers do a good work in summarizing their inputs, without loosing much of the information necessary to efficiently perform the classification task. Moreover, we assume that the *Early Fusion* classifier struggles to estimate an appropriate model when dealing with such a large number of features, whereas the *Late Fusion* one benefits from the pre-processing performed by its two predecessors, which act as if reducing the dimensionality of the data. This hints at the use of an end-to-end approach, which was not possible here due to the size of the corpus. Regarding runtime, fusion methods have the longest ones since they require to compute both content- and graph-based features.

The Top Features identified for the *Early Fusion* were either the same or very correlated (and therefore interchangeable) with the ones already discussed when not performing any fusion. The Eccentricity, in particular, was strongly correlated with the Closeness centrality. There is also the *Naive Bayes* feature which, again, shows the complementarity of the text- and graph-based features. The fact that the output of the graph-based and content-based SVM appeared among the TF of the Hybrid Fusion confirmed this, and showed that both classifier were able to retain most of the relevant information conveyed by the features.

3.4 Conclusion

In this section, I described two classification tasks conducted over attributed networks. The first was about classifying Twitter users in order to detect real-world influence, based on text content, profile characteristics, and network structure. It turned out that the topology of the network of Twitter users was not very helpful to perform this task, as the textual content was much more discriminative in this context. The second task was about automatically detecting abuse in online conversation, through message classification. We performed this task not only based on the textual content of the messages, but also on the structure of their surrounding conversation. Contrarily to the first task, this time the structural information proved to be more relevant than the textual one. In addition, we could show that both types of information are complementary in the context of this task, which is probably the main contribution of this work.

We are currently in the process on pursuing the second task, as part of Noé Cécillon's PhD [D2]. The approach we used until now relies on feature engineering, a method that has several drawbacks. First, the process of defining and selecting the features must be performed manually. Second, it is necessary to compare a large number of features in order to avoid missing any relevant information. Third, a slight change in the problem or in the dataset can invalidate all this feature engineering work. We realized that when elaborating a new corpus to test how our methods generalize to a different dataset. We constituted a collection of conversations based on the message boards used by Wikipedia editors [VL25], and obtained some performances largely below those we got on SpaceOrigin. The reasons for this are mainly that 1) the conversations are much shorter on this medium; 2) the time scale is completely different, as months can pass between two consecutive messages; 3) the messages are not posted linearly but rather form a tree. All of this drove us to turn to more automatic methods to define classification features: embeddings. These unsupervised approaches allow summarizing complex data such as text or graphs under the form of numerical vectors, which can then be used as inputs for standard data mining tools, such as classifiers. We started reviewing the main graph embedding approaches, and comparing them in terms of classification results on our abuse detection

problem [VL2]. We also have proposed certain modifications to these methods, in order to adapt them better to the problem at hand [VL76]. We are now studying various ways of combining graph and text embeddings [VL86], in particular performing the training on both type of information in order to learn a joint representation.

As of now, we treat time in a very rough way in our approach, by modeling the conversation using a *Before* and an *After* graph. Moreover, our results show that using both of these graphs is more efficient than using only one of them, which shows the importance of the temporal dimension. An obvious extension thus consists in extracting proper dynamic graphs, including several time slices, and therefore likely to better represent the dynamics of the conversation. This would require to completely redesign the features that we used until now, or to switch to a classifier able to handle sequential data, such as Recurrent Neural Networks [220]. An alternative connecting with our work on graph embeddings would consist in learning a representation of the dynamic graph that would include its temporal dimension, which would allow keeping on using standard classifiers.

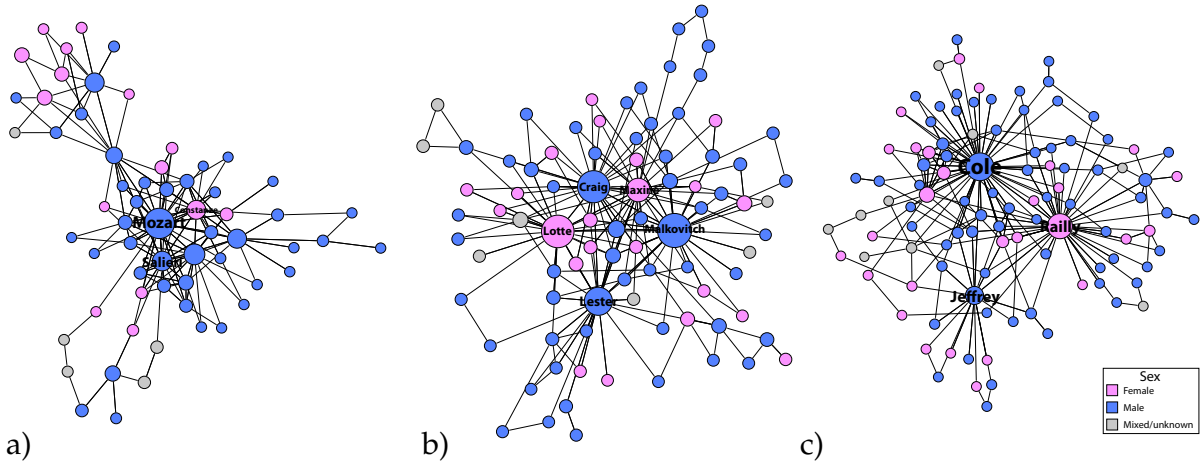

Figure 3.5: Examples of character interaction networks extracted from movie scripts: (a) *Amadeus*, (b) *Being John Malkovich*, (c) *12 Monkeys*. The displayed vertex attribute corresponds to the estimated gender.

Another extension of this work, as part of Noé Cécillon's PhD too, focuses on the extraction and classification of character interaction networks based on movie scripts (cf. Figure 3.5). Indeed, the organization of these textual documents is close enough to the linear exchange of messages that we handled when detecting abuse, so their processing requires only marginal changes. Our dataset includes not only character networks, but also metadata associated to the movies, and to the characters themselves. Using these data, we are exploring various prediction problems regarding some of the available metada, such as average grade given by viewers, box office revenues, genre, country, director, etc. This task is related to another work, conducted with Xavier Bost on the automatic summarization of TV series, which I describe in Chapter 5.

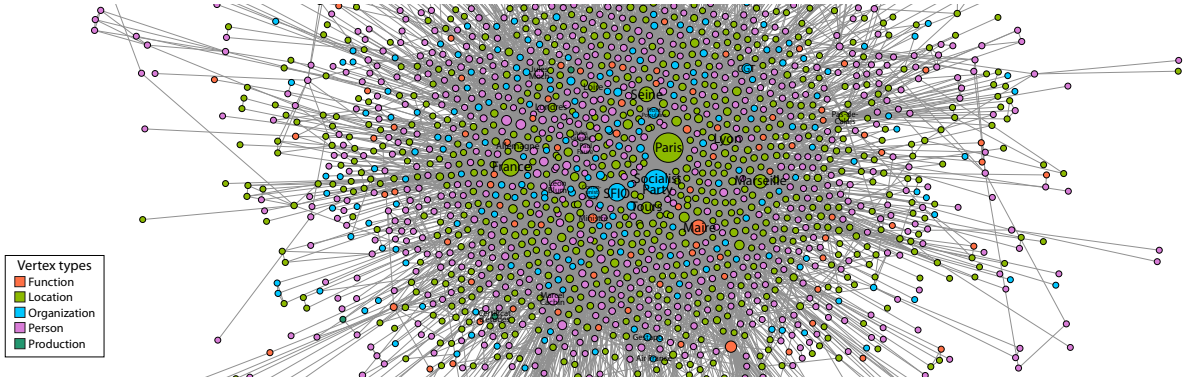

Figure 3.6: Excerpt of the network extracted from the corpus of biographical texts describing socialist members of the French parliament during the inter-war period. Vertices represent named entities, and edges correspond to co-occurrences.

The problem of extracting attributed networks from text is related to another of my ongoing projects. It aims at detecting spatio-temporal events in biographical texts and news articles, in order to identify explicit and implicit interactions between the involved historical characters and persons. Detecting these events is an *Information Extraction* [206] task that amounts to determining who did what, when, where, and possibly with/to whom. I advised several undergraduate [U11, U9] and graduate [G10, G7] students on NLP tasks related to event extraction, such as *Named Entity Recognition* [166] (NER) and *Anaphora Resolution* [192]. Some of these works resulted in publications [VL41, VL96]. This work is now conducted in the context of the Agorantic FR, in collaboration with historian Frederic Monier and political scientist Guillaume Marrel, on two types of documents: a corpus of biographical notes describing socialist members of the French parliament during the inter-war period; and journalistic articles related to certain French mayors. We extracted various types of networks from the events detected from the historical texts, as illustrated in Figure 3.6. The work on news articles aims to reconstituting the schedule of certain political public figures based on online press, in order to assess how their activity was perceived on the Web; it led to several publications [S8, VL64, VL21, VL67].

Dynamic Networks

4

Description of Dynamic Networks

4.1 Context

In its simplest form, presented in Definition 4.1.1 and illustrated in Figure 4.1, a *dynamic* network is a sequence of static graphs containing the same vertices but evolving edges [124, 168]. It is formally related to the notion of *multiplex* network studied in Chapter 8, except that the constituting graphs form a chronologically ordered sequence in the case of dynamic graphs, whereas they have no particular order in multiplex graphs. They can also be seen as a specific type of edge-attributed network (cf. Chapter 2) in which edges are described by a numerical attribute representing their timestamp. However, this perspective makes it harder to take into account the chronological ordering of the layers. Dynamic graphs allow modeling systems that evolve through time, which is the case of a number of real-world systems. On the one hand, the explicit representation of time offered by dynamic networks over static ones has multiple advantages, such as a better and eased control over the modeled system [158], or the identification of non-trivial temporal patterns [148]. On the other hand, it requires to define specific methods able to leverage this additional information, which can be a challenge and raises specific issues [124] such as the selection of the most appropriate time scale.

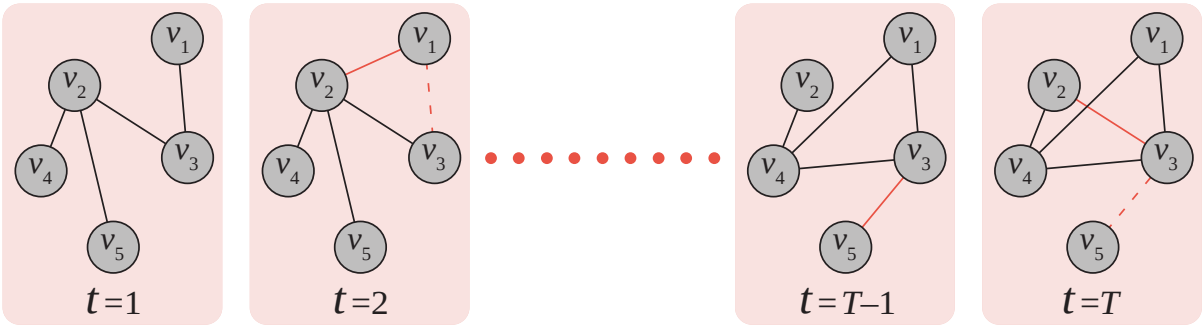

Figure 4.1: Example of dynamic network containing 5 vertices over T time slices.

Definition 4.1.1 (Dynamic Network) *Let $G = \langle G_1, \ldots, G_T \rangle$ a **dynamic network**, i.e. a sequence of T chronologically ordered graphs, which we call **time slices**. Each time slice $t \in \{1, \ldots, T\}$ is a static network $G_t = (V, E_t)$, where $V = \{v_1, \ldots, v_I\}$ is the set of vertices, and $E_t \subset V^2$ is the set of edges for slice t. Note that all*

slices have the same vertex set V, but possibly different edge sets. In case of undirected graph, we assume that edges are lexicographically ordered pairs.

Holme and Saramäki [125] mention works on dynamic networks as early as the 1960s. However, like with other feature-rich networks, the activity around this type of networks has significantly increased recently with the development of complex network analysis, as shown by the numerous recent books and surveys [3, 127, 126, 125, 158, 168]. As noted by Lambiotte [148], the interest towards dynamic networks has been supported by the availability of more and more data describing relational dynamical systems such as exchanges on social media, or mobile phone activity. More generally, they are used to model [125] social interactions based on telecommunication but also physical contacts, cell and microbiology-related systems such as interactomes (molecular interactions in a cell), distributed computing systems, biological cerebral network, groups of interacting species in ecosystems, and others.

The simplest approach when studying a dynamic graph is to aggregate all time slices to obtain a single static graph, and apply traditional tools. However, this temporal integration tends to conceal the graph dynamics. More advanced methods exist that do a better job at preserving these dynamics, that can be gathered in three distinct categories. The first is to select a tool designed for static networks, e.g. graph density [155], and apply it to each time slice separately, hence constituting one or several time series that can later be analyzed using appropriate tools. A variant consists in smoothing this process, by taking into account neighboring time slices. Several community detection methods adopt this approach, e.g. [12]. Second, some authors generalize a static tool to dynamic networks. For instance, defining a dynamic version of the concept of path allows generalizing many different path-based centrality measures, e.g. the betweenness [219]. The same goes for the concept of subgraph, e.g. motifs [13]. Finally, the third approach is to directly consider the graph as a sequence, and apply methods able to handle this type of data, e.g. by looking for co-evolution patterns Desmier *et al.* [80]. This is the type of approach I explore in this chapter.

I started working on dynamic networks at Galatasaray Üniversitesi in 2012, through a collaboration with Atay Özgövde [VL43] regarding Wireless Sensor Networks. However, this work did not result in a major methodological contribution regarding the analysis of dynamic networks, so I decided to describe it only briefly in the conclusion of this chapter (Section 4.4). Instead, I want to focus on my subsequent work on dynamic networks, that also started during my time at Galatasaray, but in the context of Günce Orman's PhD [D6], which was co-advised by Jean-François Boulicaut and myself, in collaboration with Marc Plantevit. This work led to several publications [VL38, VL14, VL70], and Günce Orman and I later continued working on the topic [VL22, VL9]. Our approach leverages methods designed for sequential pattern analysis, which is the task aiming at discovering frequent sub-sequences in sequentially structured databases [161]. Our general idea is that a dynamic network can be represented as a collection of sequences describing the evolution of its constituting parts. When applied appropriately, tools developed to study (non-graph) dynamic data can be leveraged to take advantage of such a representation.

The rest of this chapter is organized as follows. In Section 4.2, I present the concepts related to sequential pattern analysis which are used in the rest of the chapter. In Section 4.3, I describe the method that we elaborated in order to describe the dynamics of a network based on the evolution of the neighborhood of its vertices. Then, in Section 4.4, I turn to the method that we proposed to characterize dynamic community structures. I conclude in Section 4.4 with a mention of my work on WSNs and some perspectives.

4.2 Sequential Pattern Mining

In the methods proposed in this chapter, we use sequential pattern mining as a descriptive tool to find out general trends of vertex or community evolution. Here, I first shortly explain the required concepts, from the perspective of our work and how we use them.

Items roughly correspond to *discrete* descriptors characterizing the objects of interest, in our case vertices, and itemsets are therefore sets of characteristics jointly describing these objects. One can use two kinds of items, depending on the nature of the considered descriptors. If they are *Boolean*, then an item is a *single*

symbol, and its presence in an itemset means that the corresponding descriptor is true. For instance, say we want to describe weather conditions using two descriptors $wind$ and $rain$, then itemset $(rain)$ means the weather is rainy but not windy. If the descriptors are *categorical*, then an item is a couple constituted of a *symbol* and a *value*. For instance, if we distinguish several levels of rain and wind, our itemset could be $(rain{=}strong, wind{=}none)$.

Definition 4.2.1 (Item & Itemset) *Let* $\mathrm{A} = \{\alpha_1, \alpha_2, \ldots\}$ *be the set of all* ***items****. An* ***itemset*** $a \subset \mathrm{A}$ *is any subset of items from* A.

Although itemsets are sets, I represent them between parentheses, e.g. $a = (\alpha_1, \alpha_3, \alpha_4)$ because it is the standard notation in the literature.

Definition 4.2.2 (Sequence & Length) *A* ***sequence*** $\mathbf{s} = \langle s_1, \cdots, s_N \rangle$ *($N \leq T$) is a chronologically ordered list of non-empty itemsets. The* ***length*** $|\mathbf{s}| = N$ *of sequence* **s** *corresponds to the number of itemsets it contains. Two sequences are equal iff they contain the same itemsets in the same order.*

Each itemset in the sequence describes the considered object at a specific time slice t, which means that the longest a sequence can be is T, the number of time slices. It is important to stress that two itemsets can be consecutive in the sequence while not correspond to consecutive time slices: the important point is that the first to appear must be associated to a time slice *preceding* that of the second one. In other words, s_t occurs before s_{t+1} and after s_{t-1}.

Definition 4.2.3 (Sub-sequence & Super-sequence) *A sequence* $\mathbf{a} = \langle a_1, \ldots, a_M \rangle$ *is a* ***sub-sequence*** *of another sequence* $\mathbf{b} = \langle b_1, \ldots, b_N \rangle$ *iff* $\exists\, t_1, t_2, \ldots, t_M$ *such that* $1 \leq t_1 < t_2 < \ldots < t_M \leq N$ *and* $a_1 \subseteq b_{t_1}, a_2 \subseteq b_{t_2}, \ldots, a_M \subseteq b_{t_M}$. *This is noted* $\mathbf{a} \sqsubseteq \mathbf{b}$. *It is also said that* **b** *is a* ***super-sequence*** *of* **a**, *which is noted* $\mathbf{b} \sqsupseteq \mathbf{a}$.

It is worth noticing that the notions of sub- and super-sequence do not require a relation of equality between the itemsets matched in both compared sequences, but rather of inclusion. In addition to these concepts which are generic to sequential pattern mining, we defined specific notions that we used to handle networks.

Definition 4.2.4 (Vertex Sequence) *We define the* ***vertex sequence*** $\mathbf{s}_v$ *of a vertex* v *as* $\mathbf{s}_v = \langle (\alpha_{11}, \ldots, \alpha_{H1}), \ldots, (\alpha_{1T}, \ldots, \alpha_{HT}) \rangle$, *where* H *is the number of symbols available to describe each vertex. This sequence thus provides the complete description of the vertex for each time slice.*

The exact value of H, the number of available symbols, depends on the nature of the descriptors used to characterize the considered objects. For Boolean descriptors, it corresponds to $|A|$, the total number of items, as each item is a symbol. But for categorical descriptors, it is smaller, as each symbol is likely to be used in several items.

Definition 4.2.5 (Support & Supporting Vertices) *The set of* ***supporting vertices*** $S(\mathbf{s})$ *of a sequence* **s** *is defined as the set of vertices whose vertex sequence is equal to* **s**, *or is a super-sequence of* **s**: $S(\mathbf{s}) = \{v \in V : \mathbf{s}_v \sqsupseteq \mathbf{s}\}$. *The* ***support*** $\mathrm{Sup}(\mathbf{s})$ *of a sequence* **s** *is the proportion of its supporting vertices in* V: $\mathrm{Sup}(\mathbf{s}) = |S(s)|/I$.

The support is used to measure how widespread a sequence is over the data. It is also possible to restrict its scope to a part of the data using the notion of *relative* support.

Definition 4.2.6 (Relative Support & Supporting Vertices) *The set of* ***supporting vertices*** *of a sequence* **s** ***relatively*** *to a vertex subset* $X \subset V$ *is the proportion of vertices in* X *that support* **s**: $S(\mathbf{s}, X) = \{v \in X : \mathbf{s}_v \sqsupseteq \mathbf{s}\}$. *The* ***support*** *of* **s** ***relatively*** *to* X *is the proportion of vertices in* X *supporting* **s**: $\mathrm{Sup}(\mathbf{s}, X) = |S(\mathbf{s}, X)|/|X|$.

Besides the support, the *growth rate* is another measure used to assess the relevance of a sequence over a dataset.

Definition 4.2.7 (Growth Rate) *The **growth rate** of a sequence* $\mathbf{s}$ *relatively to a vertex subset* $X \subset V$ *is* $\mathrm{Gr}(s, X) = \mathrm{Sup}(\mathbf{s}, X)/\mathrm{Sup}(\mathbf{s}, \overline{X})$*, where* $\overline{X}$ *is the complement of* X *in* V*, i.e.* $X = V \setminus \overline{X}$*.*

The growth rate measures the *emergence* of $\mathbf{s}$: a value larger than 1 means that $\mathbf{s}$ is particularly frequent (i.e. emerging) in X, when compared to the rest of the network.

Definition 4.2.8 ((Closed) Frequent Sequence) *Given a minimum support threshold noted* min_{Sup}*, a **frequent sequence** (FS) is a sequence whose support is greater or equal to* min_{Sup}*. A **closed frequent sequence** (CFS) is a FS which has no super-sequence possessing the same support, or better a one.*

Sequences can be mined based on a number of different constraints, as illustrated in the sequential pattern mining literature [161]. We took advantage of the SPMF framework [105], which contains several sequential pattern mining algorithms, to detect different types of sequences. Here, without deepening this subject, we focused on frequent sequences with additional constraints on closeness and length. On the one hand, we used the *CloSpan* algorithm [236] to search for CFSs. It is an efficient algorithm able to identify long sequences in real-world data, in a practical time. On the other hand, we took advantage of the TKS algorithm [104], in which the notion of FS is expressed in terms of k most supported sequences. This algorithm allows putting constraints on the minimal/maximal length of the detected sequences. We looked for the *longest frequent sequences* (LFS) by setting the minimal length parameter to the longest value treatable by our hardware. The length of a sequence gives us an idea of the duration of a trend, but longer sequences have also a smaller support. In our case, vertices following LFSs may have interesting topological properties or a specific role in the network.

4.3 Neighborhood Evolution

The method presented in this section corresponds to a work conducted with Günce Orman and Teoman Naskali from Galatasaray Üniversitesi [VL9]. It aims at describing dynamic networks at the scale of vertices, through the characterization of their neighborhood. More precisely, our method detects specific events occurring among the groups of vertices constituting a neighborhood, for each pair of consecutive time slices, which we call *neighborhood evolution events*. For each vertex, our method outputs a sequence of categorical features corresponding to the different types of events it experienced. These features can then be analyzed with any tool capable of processing temporal categorical data, in order to extract meaningful information regarding the network dynamics. This knowledge can noticeably be used to categorize vertices, or identify trends and outliers. I first summarize the method (Section 4.3.1) before describing the main results obtained on real-world networks, using sequential pattern mining to analyze the features (Section 4.3.2). The detail of the algorithms and computations is available in [VL9].

4.3.1 Proposed Method

In order to characterize a vertex v, we first considered its *direct ego-network*, i.e. the subgraph induced by v and its direct neighbors $N(v)$, as illustrated in Figure 4.2. More precisely, we worked with what we call the *diminished* ego-network, as described in Definition 4.3.1. It is obtained by removing the ego vertex from the ego-network. In Figure 4.2, this amounts to removing v_1 and all dotted edges.

Definition 4.3.1 (Diminished Ego-network) *The **diminished ego-network** (DEN) of vertex* $v \in V$ *at time* t *(*$1 \leq t \leq T$*) is the subgraph noted* $G_t(v) = (V_t(v), E_t(v))$ *and such that:*

$$V_t(v) = N_t(v) \tag{4.1}$$

$$E_t(v) = \left\{(u_i, u_j) \in E_t : u_i, u_j \in V_t(v)\right\} \tag{4.2}$$

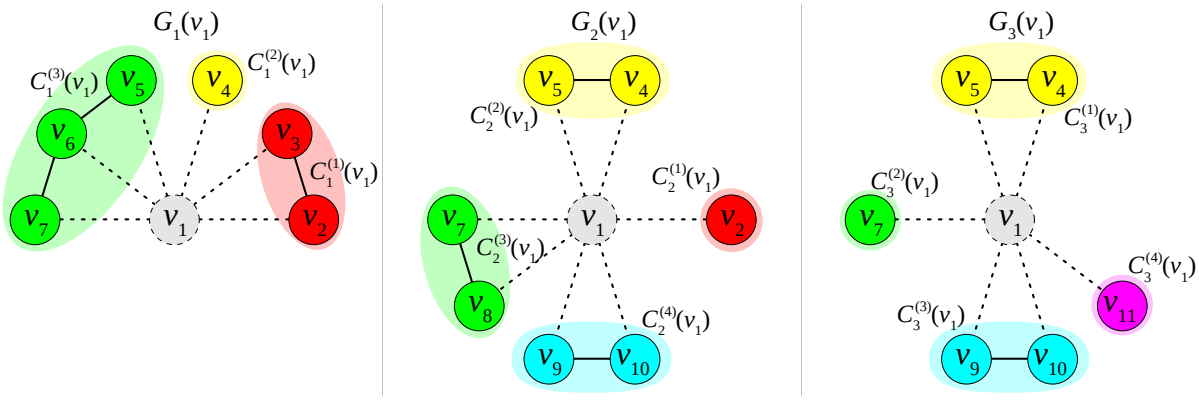

Figure 4.2: Example of direct ego-network, for vertex v_1, over 3 distinct time slices.

where $N_t(v)$ is the first order neighborhood of v at time t.

In practice, a diminished ego-network is often constituted of several components, which we call *ego-components*. They constitute a partition of the neighborhood $N_t(v)$. Each component in Figure 4.2 is represented by a different color.

Definition 4.3.2 (Ego-Component & Neighborhood Partition) *Let $G_t(v) = (V_t(v), E_t(v))$ be the diminished ego-network of $v \in V$ at time t. This graph is composed of K components (i.e. maximal connected subgraphs) called **ego-components** and noted $C_t^{(k)}(v) \subseteq N_t(v)$. The set $C_t(v) = \{C_t^{(1)}(v), \ldots, C_t^{(K)}(v)\}$ of these K ego-components constitutes what we call a **neighborhood partition**.*

We characterized the evolution of a vertex neighborhood through six so-called *evolution events*, likely to occur in the associated neighborhood partition. Note that it is possible for an ego-component to simultaneously undergo several events.

- *Birth* event: an entirely new ego-component appears from one time slice to the next one, i.e. it contains only vertices not belonging to any ego-component of the previous time slice. In Figure 4.2, this corresponds to the apparition of the cyan ego-component (v_9 and v_{10}) at $t = 2$.
- *Death* event: an ego-component completely disappears from one time slice to the next one, i.e. none of its vertices appear in any ego-component anymore. In Figure 4.2, this corresponds to the disappearance of the red ego-component (v_2) from $t = 2$.
- *Merge* event: combination of two or more ego-components into a single one, i.e. the resulting ego-component contains vertices coming from at least two distinct ego-components from the previous time-slice. In Figure 4.2, this corresponds to the fusion of the yellow ego-component (vertex v_4) and a part of the green ego-component (vertex v_5) from $t = 1$, to form the yellow ego-component at $t = 2$.
- *Split* event: an ego-component breaks down into two or more ego-components, i.e. some of its vertices belong to different ego-components in the next time slice. In Figure 4.2, this corresponds to the division of the green ego-component, whose vertex v_5 joins v_4 to form the yellow ego-component at $t = 2$, while v_7 becomes a part of a new green ego-component together with v_8.
- *Expansion* event: an ego-component integrates new vertices from one time slice to the next, i.e. it contains at least one vertex which did not belong to any ego-component in the previous time slice. In Figure 4.2, this corresponds to the green ego-component getting a new vertex v_8 at $t = 2$. Note that the same ego-component also loses two vertices (v_5 and v_6) due to other events (a merge and a contraction, respectively) occurring simultaneously.
- *Contraction* event: some vertices of an ego-component disappear from the neighborhood from one time slice to the next, i.e. at least one of its vertices from the previous time slice does not belong to any ego-component in the current one. In Figure 4.2, this corresponds to vertex v_6 (green ego-component) disappearing from the neighborhood at $t = 2$.

It is worth noticing that these events go by symmetrical pairs: *birth* vs. *death*, *merge* vs. *split*, and *expansion* vs. *contraction*.

The events we used are quite similar to those proposed in several works studying the evolution of dynamic community structures, e.g. Palla *et al.* [188] and Toyoda and Kitsuregawa [224]. However, their semantics is quite different, since in our case these events are experienced by *vertex neighborhoods*, whereas in the other works they concern *communities*. The main consequence is that in our case, it is possible for a vertex to simply disappear from the neighborhood, or appear into it. On the contrary, in the context of community structures, the whole network is considered, so vertices can only switch communities (provided the communities form a partition of V, and V is fixed). As illustrated by our example, several events can occur simultaneously. Their occurrence and frequency depends on the dynamics of the network structure in the neighborhood of the considered vertex. This is why we made the assumption that the sequence of these events can be used to characterize this vertex. The algorithm that we proposed to extract all neighborhood events has a complexity in $O(|E|^2/(TI))$, where I is the number of vertices in each time slice, and T is the number of time slices.

Once we have identified the events, each vertex can be represented as a sequence. We considered events as Boolean descriptors (an event occurs or not), which means that, at a given time slice, each vertex is described by up to 6 different items. As explained in Section 4.2, our method looks for CFSs and LFSs using algorithms CloSpan and TKS.

Depending on the homogeneity of the considered dataset, it may not be possible to find interesting sequences when considering the whole network. In this case, one can mine *subparts* of the network separately. We applied this principle using clusters identified based on the sequences of neighborhood events. For this purpose, we described each vertex by a sequence of numerical values, each one representing the number of neighborhood events it underwent at one time slice. We processed the distance between two vertices through *Dynamic Time Warping* (DTW) [111], which was designed to measure the distance between two numerical time series. Then, we performed a standard cluster analysis to identify clusters of similar sequences. We looked for both types of sequences (CFS and LFS) in each cluster separately. In order to select the most relevant ones, we leveraged the *growth rate* from Definition 4.2.7: for a subset X, this measure allows assessing how frequent a sequence is relative to the rest of the network.

4.3.2 Empirical Validation

For illustration purposes, we analyzed three real-world dynamic networks: DBLP (2,145 vertices, scientific collaborations in data mining and artificial intelligence), LastFM (1,701 vertices, social relations through musical tastes) and Enron (28,802, email exchanges). The first one was available in the literature [80], whereas we built the last two ones ourselves.

Number of Events Before looking for sequences, we examined the evolution of the number of events occurring in each network, as displayed in Figure 4.3. Only a minority of vertices go through the considered period without undergoing a single event: 99 for DBLP ($\simeq$ 4.6% of the vertices), 432 for LastFM ($\simeq$ 25%) and 153 for Enron ($\simeq$ 0.5%). It is interesting to notice that the evolution of the event counts is very different from one network to the other, which means it is a discriminant feature. However, there are also some common points: *births* are the most common events for all the networks, followed by *deaths*, whereas the rarest ones are *merges* and *splits*.

For DBLP, we observed an increase for all the event types, starting from the beginning of the considered period (1990–2012). Then, around 1994–2000, there was a sudden increase for both *births*, which can be interpreted as the apparition of new collaborations among researchers. The number of *deaths* also increased, but to a lesser extent, which means these collaborations were relatively stable and persistent. There was also an increase in *expansions*, followed by an increase in *contractions*, around time slice 2000–2006. This shows that authors tended to include new people into existing collaborations, and then possibly reduced the size of the collaborator groups. Interestingly, the periods for which the plot exhibits a sudden increase of *births* and *expansions* (1994–2000 and 1996–2002) correspond to the merger of two important Data Mining-related

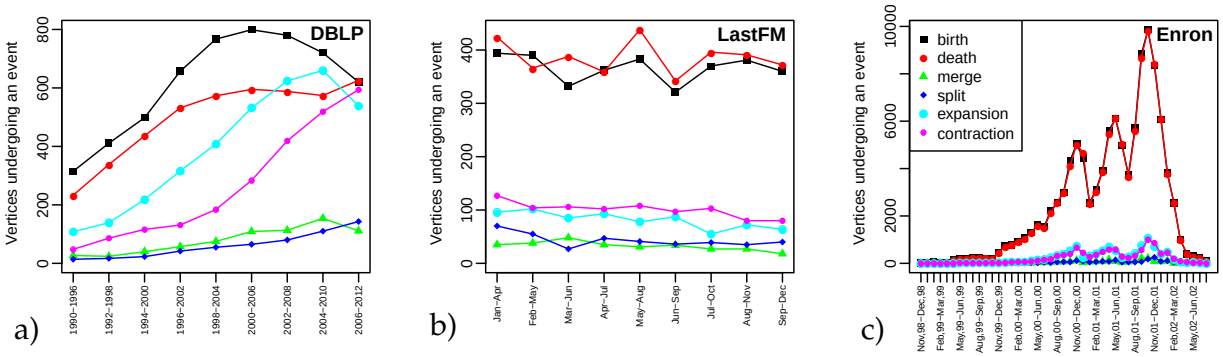

Figure 4.3: Number of vertices undergoing each type of neighborhood event, per time slice, for all three considered dynamic networks: DBLP (a); LastFM (b); and Enron (c).

conferences, ECML and PKDD (occurring in 2001) and the apparition of an important new one, ICDM (also in 2001). We believe the collaborations resulting from these academic events explain our observations regarding the neighborhood events in this network.

For LastFM, the events were relatively stable over the considered period of one year. This may be too short to uncover strong changes in this specific network, but we identified interesting pattern though, which I discuss later. Regarding Enron, symmetrical event types evolved jointly as in the other networks, but this was even more marked here. Since this is an email network, a *birth* and a *death* correspond to a conversation starting/restarting and ending/pausing, respectively. Put differently, if two persons stop sending each other emails for some time, a *death* event will occur, and if they then resume their exchange, it will be a *birth*. This is a common situation, which explains the much larger numbers of *births* and *deaths*. *Expansions* and *contractions* correspond to people joining or leaving existing conversations, which was much rarer. *Merges* and *splits* were even rarer because they happened when two conversations became one, or when some people started branching out of the conversation, which is quite unusual in email exchanges. Overall, we observed a seasonal trend, with sudden increases in June and December, which correspond to the elaboration of biannual balance sheets and activity reports.

Sequential Patterns The goal in this part is not to review exhaustively all identified sequences, but rather to show the kind of information that our method can provide to the user. Table 4.1 shows the CFSs with the highest support rates for all three networks, as well as the LFSs. It appears that the most frequent CFSs were very short (including those not represented in the table), and contained only *birth* and *death* events, which is not surprising when considering Figure 4.3. In DBLP, 95% of the vertices underwent at least one *birth* over the considered period, and the support rate was 94% for a *death*. This means that between 1990 and 2012, the overwhelming majority of authors started a completely new collaboration and ended a collaboration (not necessarily the same). For LastFM, the most frequent CFSs were the same as for DBLP, but with slightly lower support rates. In this context, this means that a large majority of users listened at least once to the same music as a friend (possibly a new one) not connected to other friends (or with different listening habits); and also stopped listening to the same music as a friend, or even ended the friendship relation. From the perspective of this system, this can be interpreted as the action of starting/stopping listening to music recommended by some other users. In the case of Enron, the $\langle (B, D)\ (B, D) \rangle$ sequence was supported by 99% of the vertices. This means that almost everyone started then stopped an email conversation with a new person or group (not necessarily the same) twice during the considered period. We already observed that births and deaths were the most common events in Figure 4.3, but the CFSs showed in addition that not only some people, but almost everyone added and dropped two ego-components during the considered period.

Let us now consider the LFSs. In the case of Enron, the longest sequence spanned 13/45 time intervals. It is a sequence of births and deaths indicating the consecutive starts/ends of email conversations. There were a number of very similar shorter sequences alternating the same events (not represented in the table). They reflect the very unstable nature of the neighborhood in this dataset, as already noticed. In DBLP, the longest sequence spanned 8/9 time intervals, alternating *births*, *expansions* and *contractions*. When considering other

Network	**Most supported Closed Frequent Sequence s**	$\lvert S(\mathbf{s})\rvert$	Sup(**s**)
DBLP	$\langle(B)\rangle$	2,041	0.950
	$\langle(D)\rangle$	2,032	0.940
LastFM	$\langle(D)\rangle$	1,530	0.900
	$\langle(B)\rangle$	1,510	0.880
Enron	$\langle(B,D)\rangle$	28,649	0.990
	$\langle(B,D),(B,D)\rangle$	28,634	0.990
Network	**Longest Frequent Sequence s**	$\lvert S(\mathbf{s})\rvert$	Sup(**s**)
DBLP	$\langle(B),(B),(E),(E),(E),(C),(C),(C)\rangle$	33	0.010
LastFM	$\langle(B),(D,M,E),(D),(D,C),(C),(C),(C)\rangle$	14	0.008
Enron	$\langle(B,D),(B),(D),(B),(D),(B),(D),(B),(D),(B),(B),(D),(B,D)$	1,266	0.004

Table 4.1: Most supported closed frequent sequences (top three rows) and longest frequent sequences (bottom three rows) with their number of supporting vertices ($\lvert S(\mathbf{s})\rvert$) and support (Sup(**s**)). Events are represented by their initial.

very similar sequences, we got a total support of 11%. It means that authors tended to start new collaborations in the beginning of the considered period, then they developed their ego-components, before finally reducing their size. It seems that once a group of collaborators was set, it tended to last, even though it evolved. As an illustration of this observation, Figure 4.4 displays an example of such a life cycle for vertex v_{98}. For clarity, only the time slices relevant to the sequence are represented. The red ego-component initially containing vertices v_{99} and v_{342} was born between $t = 3$ and 4. Then, it expanded ($t = 5$ and 6) and its size increased from 2 to 17 vertices. Later ($t = 8$ to 10), it shrunk back to 4 vertices. This case is representative of many vertices in this network. The longest sequence found in the LastFM network spanned 7 time intervals (out of 9), and was supported by 14 vertices. It is relatively similar to the sequence already discussed for DBLP. However, long sequences were rare in LastFM, and unlike for DBLP we did not not detect any shorter sequence similar to this one. Still, it is worth noticing that these 14 vertices model the most active users in this dataset. The sequence can be interpreted as follows: these users tended to listen to a relatively stable group of artists, together with their friends, which corresponded to their core favorites. But they were also open to trying new artists, without necessarily sticking to them, as shown by the fluctuations observed in the sequence. This type of interpretation is characteristic of our event-based approach and could not be done through classic topological measures.

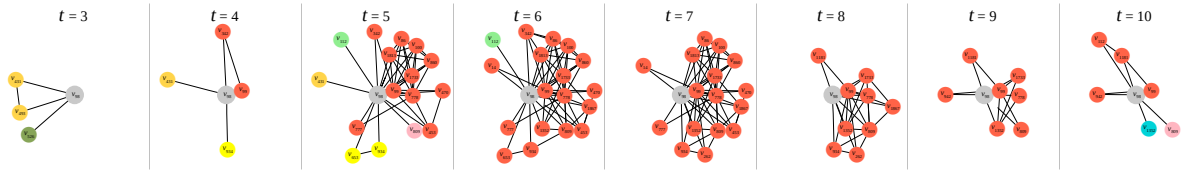

Figure 4.4: Example of a vertex (in gray) following the longest frequent sequence in DBLP. Colors correspond to ego-components, and spatial positions are fixed.

Cluster Analysis For all three networks, the cluster analysis performed on the vertex set based on their sequence similarity led to two very unbalanced clusters: the larger one contains what we call *stable vertices*, whose neighborhood did not change much over time, and the smaller one contains *active vertices*, who underwent numerous and frequent events. The CFSs detected for the larger clusters were exactly the same as the ones found for the whole graph, and they were thus not specific to these clusters, as indicated by their growth rates, which were lower than 1. The LFSs detected in the same clusters felt into two categories. On the one hand, in LastFM they concerned core vertices of the cluster and included events related to the creation/deletion of single edges. On the other hand, in DBLP and Enron, they concerned peripheral vertices and were caused by important changes in larger groups of vertices. Put differently, the second situation corresponds to vertices whose neighborhood was more unstable than for the rest of the cluster. This illustrates that LFSs can be used to characterize both the core or the periphery of such clusters.

Turning to the smaller clusters, we got much larger growth rates for all identified CFSs, which means that they were particularly typical of these clusters. For DBLP, we had a small active group of vertices whose

neighborhoods were increasing through expansions and births. For LastFM, the neighborhoods tended to expand too, but also to split. For Enron, the trend was long and included many births and deaths, as well as a few expansions and contractions. The LFSs also had a very large support, except for DBLP where it was less than half of the vertices. For Enron, it was very similar to the CFS, so it did not bring much additional information. For LastFM, we got an infinite growth rate, which means that the sequence was completely absent from the larger cluster. This sequence contains a variety of events, including *split*, *merge* and *expansion*. It is consistent with our observation from the previous subsection, in that it describes a behavior involving more neighborhood changes, compared to the larger cluster. For instance, the *split* event reflects the fact that, for a user of interest, a group of friends separated in two groups with distinct music consumption habits, while keeping certain other habits similar to the user's.

4.4 Community Characterization

The method described in this section was developed with Günce Orman, Jean-François Boulicaut, and Marc Plantevit, during Günce Orman's PhD [D6]. It was the object of several national [VL70] and international publications [VL38, VL14]. It aims at leveraging sequential pattern mining in order to characterize evolving community structures in dynamic networks.

Indeed, as already mentioned in Chapter 2, many methods exist to detect communities, but very few to *interpret* them, and make sense of these vertex partitions relative to the studied system. Certain authors focus on the sole network structure [150, 153], but such characterization can be enhanced by leveraging additional features, such as vertex attributes [227], as I did in Chapter 2. In the method described here, we took advantage of the network *dynamics*. We considered the interpretation problem as completely independent from the approach used for community detection, and consisting in dealing with a sequence of vertex partitions. Our method was based on the original definition of the notion of community in social sciences, which underlines that vertices belonging to the same community should be relatively similar and/or share a common behavior. We described vertices individually, using so-called *descriptors*, which can be either topological measures or vertex attributes. The behavior of a vertex was represented by its sequence of descriptors. We characterized a community in terms of how similar the evolution of its constituting vertices is. In the following, I summarize the method (Section 4.4.1) and I provide the main results that we obtained on real-world networks (Section 4.4.2).

4.4.1 Proposed Method

Our goal is to leverage the network dynamics to characterize community structures, so we need first to define the notion of dynamic community structure.

Definition 4.4.1 (Dynamic Community Structure) *Let $G = \langle G_1, \ldots, G_T \rangle$ be a dynamic network. Then a **dynamic community structure** $\mathcal{P} = \langle \mathcal{P}_1, \ldots, \mathcal{P}_T \rangle$ is a sequence of static community structures $\mathcal{P}_t$ obtained at each time slice t ($1 \leq t \leq T$).*

Each static community structure is itself a partition of V, as introduced in Definition 2.1.1, and such that $\mathcal{P}_t = \{P_t^{(1)}, \ldots, P_t^{(K_t)}\}$, where K_t is the number of communities at time t, and $P_t^{(k)}$ is the k^{th} community at time t.

It is important to note that, as communities might split, merge or disappear, some communities represented by the same index k in two different time slices do not necessarily match. Consider the example of Figure 4.5, for instance: communities $P_1^{(2)}$ (second community at $t = 1$) and $P_2^{(2)}$ (second community at $t = 2$) are completely different. Consequently, a community is not necessarily a stable group of vertices. This is why we focus our interpretation on groups of vertices going through several times slices together, *while possibly switching communities simultaneously*. Put differently, we are interested in groups of vertices following the same community trajectory, i.e. jointly belonging successively to the same communities over time. In Figure 4.5, v_1,

v_2 and v_3 form such a group, as they stay in the first community, but v_4 and v_8 too, as they change community together.

Unlike the method that I described in Section 4.3.1, here the vertices are not described by Boolean descriptors but by *categorical* ones. Consequently, the considered items correspond to *pairs* of symbols and values, as explained in Section 4.2. Concretely each descriptor can be either a topological measure able to describe a vertex (e.g. degree, betweenness, closeness), or a vertex attribute (e.g. age, gender, occupation in a social network). Note that numerical descriptors need to be discretized. Therefore, items take the form $betweenness$=[5; 10[or $gender$=$female$. Representing vertices through such sequences allows combining individual (vertex attributes), and relational (topological measures) information, as well as the network dynamics (sequence). When we published this work, this type of representation had not been used in the literature before.

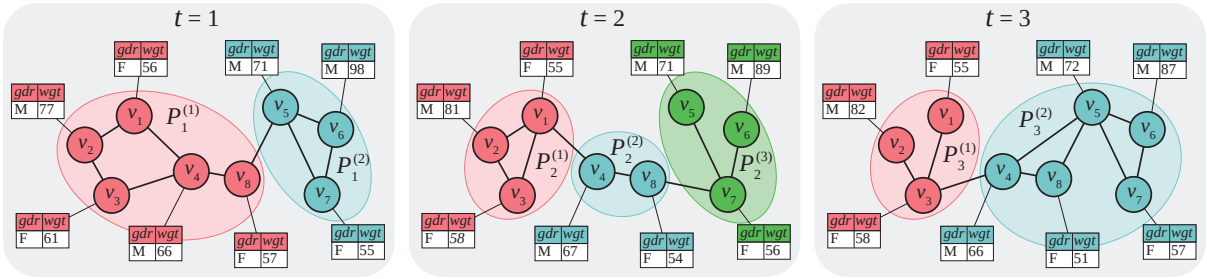

Figure 4.5: Example of attributed dynamic network, including 8 vertices described by 2 attributes (gender and body weight) over 3 time slices.

In addition, we consider community membership as an other descriptor. Figure 4.5 displays a social network containing two attributes representing the gender (gdr) and weight (wgt) of the considered persons. Suppose that we add the degree as a topological measure, as well as the community, then the vertex sequence of v_4 is $\mathbf{s}_4 = \langle(gdr$=M, wgt=66, deg=3, com=$P_1^{(1)}$), (gdr=M, wgt=67, deg=2, com=$P_2^{(2)}$), (gdr=M, wgt=66, deg=3, com=$P_3^{(3)}$)$\rangle$. We say that a sequence is *community-related* if it contains *at least* one item based on this descriptor. If it contains *only* such items, then it is called *community-focused*. If it contains none of them, it is *community-independent*. We can extract and use two types of sub-sequences from a community-related sequence.

Definition 4.4.2 (Community-wise & -less Sub-sequences) *Let* $\mathbf{s}$ *be a community-related sequence. Its* ***community-wise sub-sequence*** $\mathbf{s}^W$ *is its maximal community-focused sub-sequence. In other words, there is no other community-focused sub-sequence* $\mathbf{s}'$ *such that* $\mathbf{s}^W \sqsubset \mathbf{s}' \sqsubseteq \mathbf{s}$.

Its ***community-less sub-sequence*** $\mathbf{s}^L$ *is its maximal community-independent sub-sequence. Put differently, there is no other community-independent sub-sequence* $\mathbf{s}'$ *such that* $\mathbf{s}^L \sqsubset \mathbf{s}' \sqsubseteq \mathbf{s}$.

The community-wise sub-sequence of $\mathbf{s}_4$ from the previous example is $\mathbf{s}_4^W = \langle(com$=$P_1^{(1)}$), ($com$=$P_2^{(2)}$), ($com$=$P_3^{(3)}$)$\rangle$, whereas its community-less sub-sequence is $\mathbf{s}_4^L = \langle(gdr$=M, wgt=66, deg=3), (gdr=M, wgt=67, deg=2), (gdr=M, wgt=66, deg=3)$\rangle$.

In order to characterize dynamic community structures, we needed to solve two problems: 1) how to represent efficiently dynamic community structures; and 2) how to identify relevant information in this representation. A naive representation would consist in the set of all vertex sequences $\mathbf{s}_v$ extracted from the raw data (such as $\mathbf{s}_4$ above), and their sub-sequences. However this would not result in a compact or convenient representation. Instead, let us first consider the set Q of all possible sub-sequences of these vertex sequences: $Q = \{\mathbf{s} : \mathbf{s} \sqsubseteq \mathbf{s}_v, v \in V\}$. Among them, by focusing only on those that are community-focused (such as $\mathbf{s}_4^W$), we get all possible *community trajectories*, i.e. sequences of communities a vertex can successively belong to, in our dataset. We obtain the complete description of each such trajectory by selecting all its super-sequences in Q, which are all community-related by construction, and constitute a relatively small part of Q. In our example, that would include sequences such as $\langle(gdr$=M, deg=3, com=$P_1^{(1)}$), (wgt=67, com=$P_2^{(2)}$), (com=$P_3^{(3)}$)$\rangle$ (supported by

v_4) or $\langle (gdr{=}\text{F}, com{=}P_1^{(1)}), (gdr{=}\text{F}, wgt{=}54, com{=}P_2^{(2)}), (gdr{=}\text{F}, wgt{=}51, deg{=}2, com{=}P_3^{(3)})\rangle$ (supported by v_8).

Finally, we get the most relevant sequences for each community trajectory by first removing the community-related descriptor of its associated super-sequences in order to get community-less sub-sequences (such as $\mathbf{s}_4^L$), and then mining CFSs within them. For this purpose, as explained in Section 4.2, we used algorithm CloSpan with minimal support (min_{Sup}) and growth coefficient (min_{Gr}) thresholds. Both support and growth rate are computed relatively to the considered group of vertices. The total algorithmic complexity of our approach is in $O(rT^2I)$, where I is the number of vertices, T the number of time slices, and r the number of CFSs identified by CloSpan.

4.4.2 Empirical Validation

We evaluated our methods on two kinds of data: artificial graphs and real-world networks. For the former, we proposed LFR-DA, an extension of an existing generative model [113], in order to produce attributed dynamic graphs with controlled properties, and study how the behavior of our method is affected by these properties. However, here I focus only on the results obtained on real-world data, corresponding to the DBLP [80] and LastFM dynamic networks already studied in Section 4.3.2. We used the *Incremental Louvain* algorithm [12] to detect the evolving community structures. Regarding the descriptors, in addition to the vertex attributes present in the datasets, we computed a selection of standard topological measures: degree, local transitivity [232], eccentricity [120], betweenness [106], closeness [205], Eigenvector centrality [30], within module degree & participation coefficient [115], and embeddedness [150].

DBLP The communities detected in the DBLP network were very unstable, changing much through the considered period. They generally start smaller, they merge into larger ones around $t = 7$–9. A first look reveals that they are clearly not homogeneous in terms of descriptors. In particular, regarding the attributes (which in DBLP represent publication outlets), members of the same community often publish in several different conferences or journals. We mined CFS using parameters $min_{\text{Sup}} = 21$, $min_{\text{Gr}} = 1.00$ and $max_{seq} = 5$. To provide the reader with an order of magnitude of the results size: we obtained 1,106,108 CFSs, only 19,922 of which ($\simeq 1\%$) were community-related.

Closed Frequent Sequence s	Sup	Gr
$\langle (emb{=}\text{H}), (SDM{=}1), (btw{=}\text{H}, cls{=}\text{L}, prt{=}\text{H}), (btw{=}\text{H}, cls{=}\text{L}, prt{=}\text{H})\rangle$	40	6.86
$\langle (btw{=}\text{H}, (emb{=}\text{H}, ICDM{=}1), (cfr{=}[1;5])\rangle$	40	6.00
$\langle (emb{=}\text{H}), (TKDE{=}1, jrl{=}[1;5]), (emb{=}\text{H}, jrl{=}[1;5]), (jrl{=}[1;5]), (jrl{=}[1;5])\rangle$	40	4.00

Table 4.2: Three characteristic sequences detected for a DBLP community of interest constituted of 120 vertices. From top to bottom, they are respectively noted $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$. The topological measures appearing in the sequences are embeddedness (emb), betweenness (btw), closeness (cls), participation coefficient (prt); with value high (H) or low (L). The vertex attributes are the numbers of publications at conferences SDM and ICDM, journal TKDE, as well as the total numbers for conferences (cfr) and journals (jrl).

As an example, I focus on a community appearing at $t = 6$ (time slice 2000-2004) and containing 120 vertices. Note that, in other time slices, some of these 120 vertices belong to several different communities. We have found 3 characteristic sequences containing this community as an item. They are listed, with their support and growth rate, in Table 4.2. The evolution of the 120 vertices at $t \in \{2, 4, 6, 8, 10\}$ is represented in Figure 4.6. The vertex colors represent the sequence that they support. For the sake of readability, at each time slice, I show only the vertices belonging to the largest community involved in the sequences.

As shown in the figure, most of the vertices do not belong to the same community in the first time slices. They gather together at $t = 6$ and split again later. This community is not homogeneous around one conference or journal, since we identified 3 characteristic sequences involving different scientific platforms (SDM, ICDM and TKDE). However, all of them are related to data mining, so the community is thematically homogeneous. The topological information present in all three sequences describes vertices strongly belonging to their community and critical for information flow (high betweenness). The first sequence ($\mathbf{s}_1$) indicates that, after publishing in SDM once, the authors get connections with many other communities (high participation

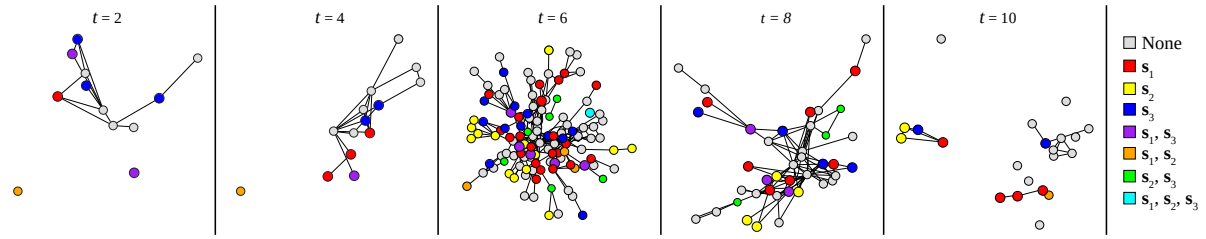

Figure 4.6: Evolution of the community of 120 vertices appearing at $t = 6$ in the DBLP data and already mentioned in Table 4.2, for time slices $t \in \{2, 4, 6, 8, 10\}$. The color of a vertex indicates which of the three characteristic sequences from Table 4.2 it supports (see legend).

coefficient). Based on the growth rates, one can observe that these sequences are not highly emergent for this community. Some investigation revealed that some other vertices exhibit similar trends, but in separate communities.

LastFM The evolution of the LastFM communities is generally smoother than for DBLP. We observed relatively large communities at $t = 1$ and 2, and communities tend to evolve most of all through small-scale events (by opposition to large-scale events such as split or merge). This network describes an evolution over a single year, while the DBLP network represents a 18 years period. So not only the systems, but also the temporal scales are different. Table 4.3 shows the characteristic sequences obtained for a community of interest containing 81 vertices and appearing at time $t = 1$. We use it to show how our method can be used to answer a concrete question regarding these data: *How do non-Jazz listeners behave?* The interested reader will find other examples of answered questions in [VL14].

Closed Frequent Sequence s	Sup	Gr
$\langle(wmd$=L, prt=L), (wmd=L, prt=L), (PF=[5;10]), (wmd=L, prt=L, TB=[5;10]), (wmd=L, prt=L)$\rangle$	15	14.97
$\langle(TB$=[5;10]), (TB=[5;10]), (FS=[1;5])$\rangle$	15	7.23
$\langle(wmd$=L, prt=L), (PF=[5;10])$\rangle$	15	6.21

Table 4.3: Three characteristic sequences detected for a LastFM community of interest constituted of 81 vertices. From top to bottom, they are respectively noted $\mathbf{s}'_1$, $\mathbf{s}'_2$ and $\mathbf{s}'_3$. The topological measures appearing in the sequences are the participation coefficient (prt) and within-module degree (wmd); with value high (H) or low (L). The vertex attributes are the numbers of times the LastFM user listens to musical artists Frank Sinatra (FS), Pink Floyd (PF), and The Beatles (TB).

The topological descriptors present in $\mathbf{s}'_1$ describe a non-hub vertex with low participation coefficient, which persists through time. Regarding the attributes, these users listen to Pink Floyd 5-10 times, then to the Beatles 5-10 times. Although the artist they listen to changes, their structural features do not change at all, and moreover both artists are not Jazz acts. Sequence $\mathbf{s}'_2$ describes an emerging vertex group, which listens to the Beatles 5-10 times for two time slices, and then to Frank Sinatra 1-5 times too. The vertices supporting these two sequences have a 50% overlap. More clearly, half of the vertices following the first sequence also follow the second. Those two sequences include two clearly non-Jazz artists (The Beatles and Pink Floyd), and one who could be considered as Jazz-relative (Frank Sinatra). So, we can suppose the users from this community are not interested in Jazz, or only remotely. We did not find any other such community of non-Jazz listeners with our method. Sequence $\mathbf{s}'_1$ indicates that they do not have many connections outside of their communities, since they have a low participation coefficient for many time slices.

We investigated the topology of this group for all the time slices, as illustrated in Figure 4.7. Circles represent vertices supporting $\mathbf{s}'_1$, and squares are their neighbors not supporting it. Unlike Figure 4.6, vertex colors do not represent the sequence they support, but rather the communities they belong to. Supporting vertices may belong to different communities when considered at different time slices. But, independently from this observation, they do not have many connections outside of their communities. They are not central at all. Mostly, they belong to a tree-like structure, holding a non-hub role.

Among the 81 vertices of this community from $t = 1$, 22 are still together at $t = 4$. The bottom sequence of Table 4.3, $\mathbf{s}'_3$, is characteristic of this specific group of vertices. The evolution of their interconnections and

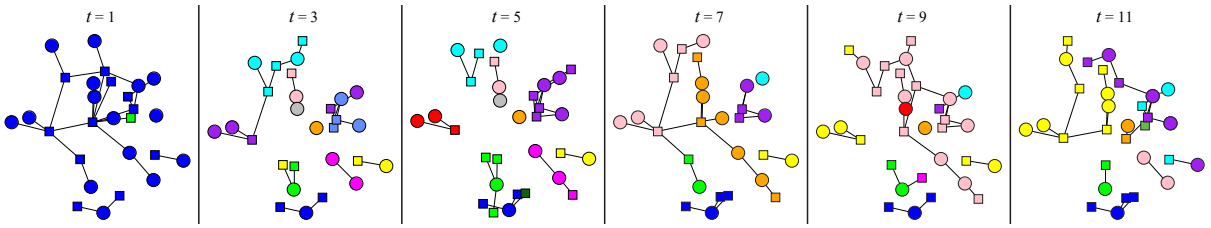

Figure 4.7: Evolution of the interconnection of the vertices supporting the top sequence ($\mathbf{s}'_1$) from Table 4.3 (circles), and of their non-supporting neighbors (squares), for $t \in \{1, 3, 5, 7, 9, 11\}$. The meaning of the vertex colors is different compared to Figure 4.6: here, each color corresponds to a specific community at a given time slice. The same color in two different time slices does not necessarily represent the same community.

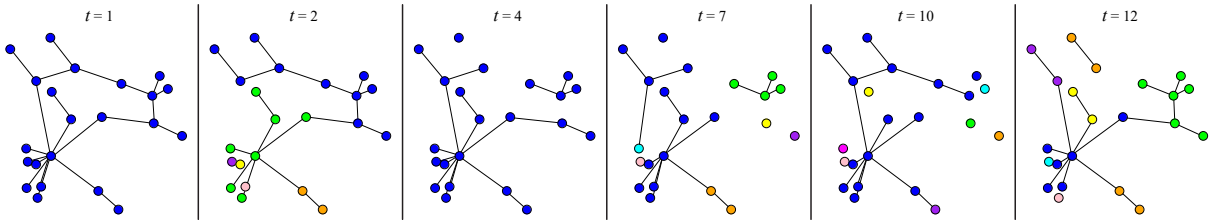

Figure 4.8: Evolution of a group of 22 vertices at $t \in \{1, 2, 4, 7, 10, 12\}$, from the LastFM data. Those vertices belong to the same community at $t = 1$ and 4. Each color corresponds to a specific community at a given time slice. The same color in two different time slices does not necessarily represent the same community.

communities is illustrated by Figure 4.8 for time slices $t \in \{1, 2, 4, 7, 10, 12\}$. The vertex colors correspond to their communities, as in Figure 4.7. Topologically, sequence $\mathbf{s}'_3$ indicates that the supporting vertices are non-hubs with low participation coefficient. Attribute-wise, these users listen to Pink Floyd 5-10 times. This trend is very similar to that of the larger community we discussed just before (sequences $\mathbf{s}'_1$ and $\mathbf{s}'_2$). This smaller sequence shows us that some vertices with similar interest keep their connections for several time slices, even if the rest of their community changes. The support obtained for $\mathbf{s}'_3$ shows that this smaller group is more homogeneous than the larger community. The same observation can be done for many other communities in our results as well. This could be interpreted as the presence of a certain hierarchical structure in the network, defined in terms of both topology and attributes (i.e. nested interests). However, a more thorough study shows that this type of hierarchy is not stable through time. For this reason, it is more accurate to consider that communities have one or several cores, corresponding to vertices supporting long sequences, i.e. evolving together for a long time. These cores can overlap and switch communities. They are joined punctually by other cores or peripheral vertices to form short-termed communities.

4.5 Conclusion and Perspectives

In this chapter, I presented two methods based on sequential pattern analysis and aiming at describing and characterizing the evolution of dynamic networks. The first method (Section 4.3) is local in the sense that it focuses on events occurring in vertex neighborhoods, whereas the second method is related to communities. Moreover, the latter can be applied to attributed dynamic networks, in which vertices are described by individual attributes, as in Chapters 2 and 3. The main contributions of this work was to show how tools originally designed to handle traditional sequential data could be adapted to the study of dynamic networks. This type of approach is not well-known by the Network Science community, and leads to results that are very different, in terms of both form and substance, to more widespread methods. There are a number of direct ways to extend this work by transposing existing variants of sequential pattern mining methods (cf. [161]) to dynamic networks, in a similar fashion. Another perspective is to adopt a null-method based model to assess the relevance of the detected patterns, and possibly restrict their number in order to alleviate the computational load. Finally, it would be interesting to assess the performance of our methods on inference tasks (by opposition to the descriptive analysis described in this chapter), such as the popular link prediction problem [167].

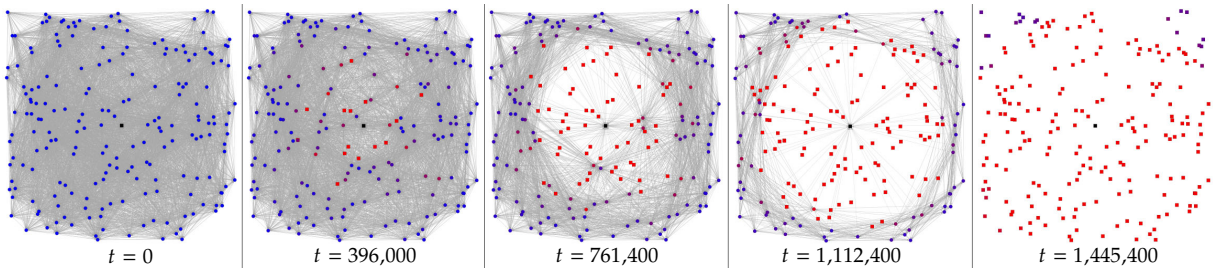

Figure 4.9: Example of the evolution of a Wireless Sensor Network. The sink is represented as a black square, whereas the rest of the vertices are sensors. Their color indicate their battery level, from full (blue) to empty (red). Dead sensors are represented by squares and alive ones by circles. Edges represent the possibility for one vertex to communicate with another.

As mentioned in the introduction of this chapter, I also want to come back briefly on the work we conducted with Atay Özgövde at Galatasaray, regarding the analysis of Wireless Sensor Networks. WSNs are communication networks constituted on the one hand of many autonomous *sensors* with very limited memory, computing power, communication reach, and battery life; and on the other hand of one or a few so-called *sinks*, possessing much larger storage capacity, computational abilities, and a power supply. Sensors gather data and send them to a sink for storage, either directly or through other sensors acting as relays when the sink is out of reach. One unique aspect of WSNs is that this is a decaying dynamic network, as sensors die one after the other when they run out of power, possibly isolating live parts of the network. Atay Özgövde and I started studying how the communication protocol affects the network overall lifetime [VL43]. For this purpose, we analyzed the evolution of a selection of standard topological measures over time, such as density, degree, assortativity, and betweenness. In doing so, we proposed a variant of the distance called the *sink-distance* which we leveraged to derive a number of distance-based measures (sink-radius, sink-betweenness).

I participate in two ongoing works aiming at analyzing and exploiting dynamic networks, and more generally dynamic data. The first takes place in the context of the interdisciplinary PhD of Noémie Févrat [D3], funded by research federation Agorantic (FR 3621) and co-advised by political scientist Guillaume Marrel and myself. It consists in studying how election laws regarding term limits affect the careers of politicians. We are in the process of constituting a large longitudinal database called the BRÉF (*Base Révisée des Élu·es de France* – Revised Database of French Representatives). It describes the elective offices occupied by French politicians, and covers terms as early as the French revolution for some types of office [S4, VL95]. Our analysis of the dynamic data relies on an approach related to sequential pattern mining, and called *Sequence Analysis* [26]. The main difference is that sequence analysis is somewhat more flexible, as it is based on an approximate comparison of sequences, and not on exact matching. Typically, one computes a similarity measure between all pairs of considered sequences, and look for classes of similar sequences.

The second work is a collaboration with historians Gaëtane Vallet and Catherine Wolff, aiming at studying the entourage of Roman emperor Trajan [S3, C1, VL88]. I already mentioned this project in Section 2.4, as the concerned data also include vertex attributes. Similarly to the previous collaboration with political scientists, these data also include some temporal information under the form of sequences of positions hold by people in Trajan's entourage, as well as sequences of missions in various parts of the empire. Here too, we use sequence analysis to classify the characters and compare them. But it will also allow answering certain historical questions related to careers under Trajan, such as the effective implementation of some form of *cursus honorum* (a compulsory standard career) at this time.

5

Segmentation of Dynamic Networks

5.1 Context

In this chapter, I present another work conducted on dynamic networks. Unlike the methods presented in Chapter 4, this is a bottom-up project aiming at developing methods able to perform a certain application. It focuses less on descriptive analysis, and more on network extraction and segmentation methods. It was conducted in collaboration with Xavier Bost during (and after) his PhD [37]. It was an interdisciplinary work funded by the Agorantic research federation (FR 3631) and involving, in addition to computer scientists, researchers of the field of Communication Studies. It was also a part of the ANR GAFES[9] project (ANR-14-CE24-0022).

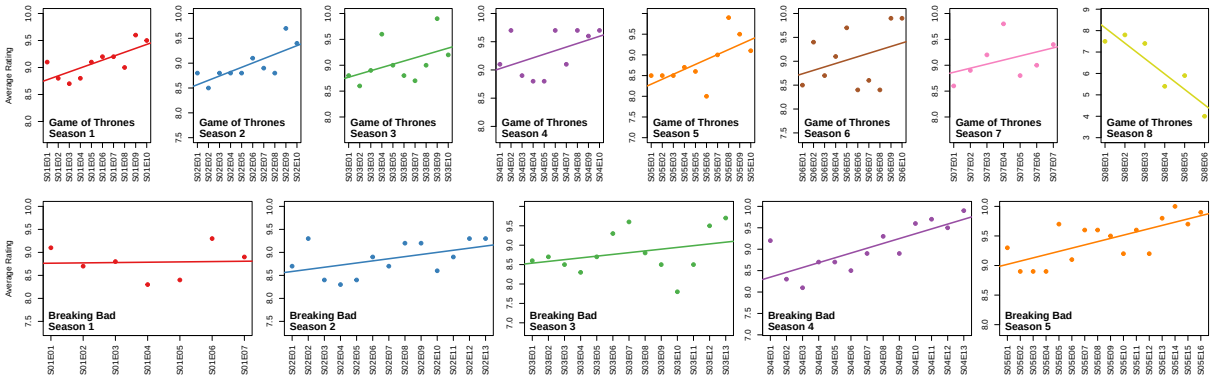

Figure 5.1: Average IMDb users' ratings for the episodes of the TV serial *Game of Throne* (top) and *Breaking Bad* (bottom), along with season trend lines.

The general topic of the thesis originates from the observation that as TV series became more popular, the way viewers consume them changed dramatically. Traditional TV series are mainly collections of one shot, narratively self-sufficient episodes, whereas modern TV series, called TV *serials* rely on continuous plots threading over dozens of successive episodes. The former were broadcast at a regular pace over a period of several months, whereas the latter are often made available to the public over a few weeks, or even one

[9] https://anr.fr/Projet-ANR-14-CE24-0022

whole season at once, and are consequently watched over a short period (a practice called *binge-watching*). As a result, viewers are largely disengaged from the plot, both cognitively and emotionally, when a new season is released. This makes it harder for the viewers to start watching a new season, possibly resulting in a disaffection for the whole TV serial. Figure 5.1 displays the average IMDb[10] (Internet Movie DataBase) ratings given to the serial *Game of Thrones* and *Breaking Bad* by their viewers. It shows a cold-start phenomenon, as the ratings start low for each season before gradually increasing as viewers get more involved.

The objective of this work was to define a method for the automatic video summarization of TV serials. The goal was not to generate teaser-type clips, but rather to provide the viewers with a summary that would get them re-immersed in the story before starting watching a new season. Video summarization methods generally rely on multimodal features specific to this medium [37]. This includes visual features, such as color and brightness, motion intensity, presence of faces; but also acoustic features, such as sound energy, presence of speech, music tempo; and finally textual features, if subtitles are available. The originally of the method discussed here is to leverage higher level information related to the plot itself. For this purpose, we modeled the story through the evolution of the dynamic network of interacting characters. I participated in this network-related part of the work, which includes two aspects that I develop in this chapter: the extraction of the dynamic character network, for which we developed a method called *Narrative Smoothing* (Section 5.2), and the exploitation of this network to generate the video summary (Section 5.3). This work has been the object of several publications [VL5, VL34, VL66, VL20, VL26, VL6].

5.2 Network Extraction

A number of authors have tried to model the plot of fictional works using the social network of their characters. Xavier Bost and I made a very thorough survey of this domain [VL6], focusing on how these networks are extracted, analyzed, and used to solve various applied problems. Most of these authors focused on literary works. Very few considered multimedia narratives, and most focused either on full length-movies or on standalone episodes of classical TV series, where character interactions are often well-structured into stable communities. These approaches consequently do not necessarily translate well when applied to *modern* TV serials. Moreover, even fewer of these works aimed at extracting *dynamic* networks, using instead a single static graph representing the whole timeline. These are likely to miss evolving and possibly parallel storylines which are typical of TV serials.

Our method aims at extracting a dynamic network of character interactions based on the conversations between them. It can handle not only standalone episodes or full-length movies, but also the complex plots of TV serials. In this case, no prior assumption can be made about a stable, static community structure that would remain unchanged in every episode and that the story would only uncover, and we have to deal with evolving relationships, possibly temporarily linked into dynamic communities. We are left with building the current state of the relationships upon the story itself, which, by focusing alternatively on different characters in successive scenes, prevents us from monitoring instantaneously the full social network underlying the plot. We thus proposed to address this problem by smoothing the sequentiality of the narrative, resulting in an instantaneous monitoring of the current state of any relation at some point of the story.

In Section 5.2.1, I describe how we identified conversational interaction between characters, and in Section 5.2.2, I explain the so-called narrative smoothing method that we proposed to handle the case of parallel subplots. Finally, in Section 5.2.3 I present the main experimental results obtained on a dataset involving three TV serials.

5.2.1 Interaction Detection

There are two main ways of considering interactions in fictional works [VL6]. The first is based on co-occurrences, and relies on the assumption that two characters interact when they are present at the same

[10] https://www.imdb.com/

place, at the same time. The second approach is conversational, as it leverages verbal exchanges between characters, and assumes that two characters interact if and only if one speak to the other. The former approach is generally considered as easier to apply (although this depends on the considered type of medium), as it only requires to identify the characters appearing at a given moment, but less reliable as characters can co-occur without interacting (and more rarely, the opposite). The latter is considered as more difficult, as it requires to detect verbal exchanges between characters, as well as who speaks and who listens. It is considered as more reliable than co-occurrences, although it is not the case in some very specific cases (e.g. mute characters). To design our method, we adopted such a conversational approach

Another important methodological point is the time scale used to infer interactions between characters. In the context of TV serials, it can be seasons, episodes, scenes, or even smaller elements such as shots. Seasons or even episodes turn out to be too wide units to provide an accurate view of the actual verbal interactions within TV serials. Because of parallel storylines, considering all the characters co-occurring in a single season or even episode as participants of the same conversation would result in many irrelevant interactions. Using the scene as a time unit is much more appropriate: it by definition a homogeneous sequence of actions occurring at the same place within a continuous period of time. The characters co-appearing in a single scene are therefore expected to speak with one another.

There are tools to automatically detect scene boundaries in videos, but even the best methods struggle to achieve a sufficient accuracy. For this reason, we performed this task manually. Similarly, automatically detecting who spoke when in a video is quite challenging: such a task, known in the speech processing field as *speaker diarization* when performed in an unsupervised way, turned out to be especially tricky when applied to TV series, often containing many speakers talking in adverse acoustic conditions (sound effects, background music, etc.). The error rates obtained with automatic tools in this context remained too high (about 50%) to serve as a reliable basis for building interaction networks. Instead, we identified the speakers manually.

Though much more relevant than larger-grained units, the scene used as a way of capturing the verbal interactions between characters may result in weak, sometimes irrelevant, interactions: if being at the same place at the same time is usually required to consider that several persons converse, it is rarely sufficient. Figure 5.2.a shows two consecutive dialogues extracted from the TV serial *House of Cards*, and belonging to the same scene. Three speakers are involved, but without any interaction between the second (*D. Blythe*) and the third (*C. Durant*) ones. The first speaker (*F. Underwood*) is talking to *D. Blythe* in the first sequence, then is moving to *C. Durant* and starts discussing with her. The resulting conversational graph naively extracted from these exchanges is shown in Figure 5.2.b. It contains a spurious edge between *D. Blythe* and *C. Durant*, which are not involved in any direct verbal interaction.

Figure 5.2: a) Two consecutive dialogue sequences within the same scene from the TV serial *House of Cards*. b) Conversational network representing this sequence, with a spurious interaction represented as a dotted edge.

Instead of globally considering the scene unit, we chose to tackle this problem by identifying the verbal interactions upon the sequence of speech turns in each scene, once manually labeled according to the corresponding speakers. In order to estimate the verbal interactions from the single sequence of utterances, we applied four basic heuristics. The first is the most general, and handles situations where the same speaker talks before and after another speaker, in which case we assume the latter talks to the former.

Definition 5.2.1 (Rule 1: Surrounded Speech Turn) *Let* $\langle \ldots, v_i, v_j, v_i, \ldots \rangle$ *be a sub-sequence of speech turns from a larger conversation. Then we consider that speaker* v_j *talks to speaker* v_i.

Note that for improved readability, each speech turn in the above sequence and the following ones is labeled according to the corresponding speaker. The second rule specifically targets the first and last utterances of each sequence, for which we do not apply the same surrounding constraints as in Rule 1.

Definition 5.2.2 (Rule 2: Starting and ending speech turns) *Let $\langle v_1, v_2, \ldots, v_{n-1}, v_n \rangle$ be a sequence of speech turns. Then we consider that v_1 talks to v_2 and that v_{n-1} talks to v_n.*

The last two rules were introduced to process ambiguous sequences of the type $\langle s_i, s_j, s_k \rangle$, where three consecutive speech turns originate in three different speakers: in such cases, the second speaker could be stated as talking to the first one as well as to the third one, or even to both of them. However, such speech turns sequences can often be disambiguated by considering speakers preceding and following the sequence.

Definition 5.2.3 (Rule 3: Local disambiguation) *Let $\langle \mathbf{s}, v_i, v_j, v_k, \mathbf{s'} \rangle$ be a sequence of speech turns, where $\mathbf{s}$ is the heading sub-sequence, $\langle v_i, v_j, v_k \rangle$ is the ambiguous sub-sequence, and $\mathbf{s'}$ is the ending sub-sequence. We distinguish 2 variants of this rule.*

***Rule 3a** applies when the middle speaker appears before the sequence, but not after, i.e. $v_j \in \mathbf{s}$ and $v_j \notin \mathbf{s'}$. We then consider that v_j is speaking with v_i rather than with v_k.*

*Symmetrically, **Rule 3b** concerns the case when the middle speaker appears after, but not before the sequence, i.e. $v_j \notin \mathbf{s}$ and $v_j \in \mathbf{s'}$. We then consider that v_j speaks with v_k instead.*

Definition 5.2.4 (Rule 4: Temporal proximity) *If the middle speaker is involved in the conversation both before and after the ambiguous sequence, i.e. $v_j \in \mathbf{s}$ and $v_j \in \mathbf{s'}$; or if v_i is involved in none of them; then we consider the ambiguous speech turn to be intended for the speaker whose utterance is temporally closer.*

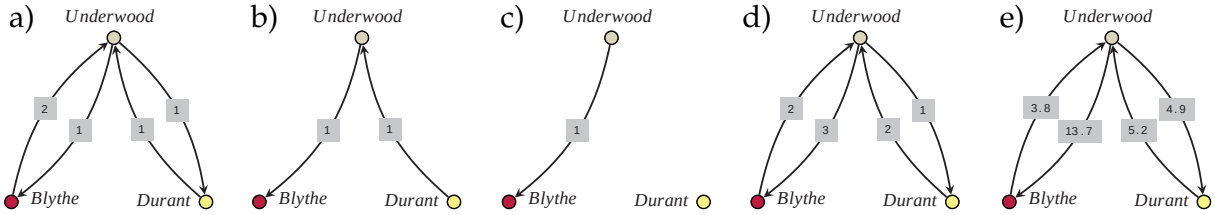

Figure 5.3: Verbal interactions estimated from the separate application of Rule 1 (a), Rule 2 (b), Rule 4 (c), and joint application of Rules 1, 2 & 4 (d, e) to the speech turn sequence shown on Figure 5.2. Weights correspond to numbers of interactions in (a–d), whereas they represent interaction durations in (e).

Figure 5.3 shows the conversational graphs obtained based on the example from Figure 5.2, when applying our rules separately or jointly. The edges are labeled with a score corresponding to the number of times each speaker talks to another one, *according to the considered rule(s)*. It is alternatively possible to use interaction duration instead of interaction count as a scoring scheme, as in Figure 5.3.e. Independently from this methodological choice, in a generic way we note $\omega(ij)$ the score obtained for edge (v_i, v_j).

5.2.2 Narrative Smoothing

As stated before, we would like to get an instantaneous measurement of the intensity of any relationship at any moment, but from the successive partial views of the underlying network that the narrative provides us. Intuitively, a particular relationship may be considered as especially important at some point of the story if the involved characters both speak frequently and a lot to each other: the time interval needed before the reactivation of the interaction in the narrative is expected to be short, and the interaction time is expected to be long whenever the relationship is active in the plot.

Before putting this idea in practice, we need first to introduce three functions, starting with a measure of the interaction between two speakers during a scene. The rule-based method presented in the previous section

allows us to estimate who speaks to whom in a given scene, and to measure this through the score ω (which can be either an interaction count or duration). Based on this, we define a function which is symmetrical relative to the speakers, and measures the total amount of interaction between them.

Definition 5.2.5 (Total Interaction) *The **total amount of interaction** $h_t(i,j)$ between speakers v_i and v_j during scene t ($1 \leq t \leq T$) is the sum of the scores ω computed for edges (v_i, v_j) and (v_j, v_i) in Section 5.2.1, i.e.*

$$h_t(i,j) = \omega_t(i,j) + \omega_t(j,i). \tag{5.1}$$

The second measure corresponds to the net balance between on one side the amount of interaction $h_\theta(i,j)$ between two characters v_i and v_j the *last* time they met, and on the other side the amount of interaction that they *have* devoted separately to other characters v_k since then.

Definition 5.2.6 (Narrative Persistence) *The **narrative persistence** $\Delta_t(i,j)$ of the interaction between speakers v_i and v_j at scene t is:*

$$\Delta_t(i,j) = h_\theta(i,j) - \sum_{t'=\theta+1}^{t} \sum_{k \neq i,j} \Big(h_{t'}(i,k) + h_{t'}(j,k) \Big), \tag{5.2}$$

where θ represents the last scene (relatively to t) during which the considered speakers have verbally interacted. If there is no speaker v_k interacting with v_i or v_j after scene θ, then we set $\Delta_t(i,j) = -\infty$.

The third measure is defined symmetrically to the narrative persistence, as the difference between on one side the amount of the interaction at the *next* meeting between the considered speakers, and the other side the amount of interaction they *will* devote separately to other characters in the meantime.

Definition 5.2.7 (Narrative Anticipation) *The **narrative anticipation** $\nabla_t(i,j)$ of the interaction between speakers v_i and v_j at scene t is:*

$$\nabla_t(i,j) = h_\theta(i,j) - \sum_{t'=t}^{\theta-1} \sum_{k \neq i,j} \Big(h_{t'}(i,k) + h_{t'}(j,k) \Big), \tag{5.3}$$

where θ represents the next scene (relatively to t) during which the speakers will interact. If there is no speaker v_k interacting with v_i or v_j before scene θ, then we set $\nabla_t(i,j) = -\infty$.

Based on these definition, we can introduce the notion of instantaneous weight, which are later used as edge weights when building the smoothed network.

Definition 5.2.8 (Instantaneous Weight) *The **instantaneous weight** $w_t(i,j)$ of edge (i,j) at scene t represents the intensity of the relationship between v_i and v_j at this time. Four possible situations can happen:*

1. *The relationship is **active** at the **current** scene. Its intensity is then the current **amount of interaction** between the speakers: $w_t(i,j) = h_t(i,j)$.*
2. *The scene occurs **before** the **first** activation of the relationship. We then use only **narrative anticipation**: $w_t(i,j) = \nabla_t(i,j)$.*
3. *The scene occurs **after** the **last** activation of the relationship. We then use only the **narrative persistence**: $w_t(i,j) = \Delta_t(i,j)$.*
4. *The scene occurs **between** two activations of the relationship. We then take the **largest** value between the **narrative persistence** and **anticipation**: $w_t(i,j) = \max(\Delta_t(i,j), \nabla_t(i,j))$.*

The first case is the simplest: each time the interaction occurs, its intensity is the total amount of interaction at the current scene. The last three cases are much trickier. Between two consecutive occurrences of the same relationship in the story, it would be tempting to consider that the relationship is still (resp. already) active if

it is recent (resp. imminent) enough at each moment considered. This corresponds to the standard approach when extracting dynamic networks: as long as the relationship is present in some temporal window, it is supposed active, and inactive as soon as no longer observed. As mentioned before, this is not appropriate in the case of TV serials. Some interacting characters may be absent from the narrative for an undefined period of time but still be linked in the underlying network, as confirmed by the fact that the last state of the relationship is generally used as a starting point when the characters are re-introduced in the story. Indeed, the temporality of the narrative should affect a relationship only when at least one of the involved characters interacts with others after and/or before the relationship is active: the relationship between two characters should only get weaker if they interact separately with others before interacting again with one another.

This is why, in order to handle the remaining cases, we need to use the previously defined narrative persistence and/or narrative anticipation. If neither of the two characters speaks to others before they interact again with one another, the last (resp. next) activation of the relation is considered as still (resp. already) fully present in the network, whatever the number of intermediate scenes the narrative introduces in-between to focus on other plot substories. Otherwise, if some character interacts with others in the meantime, then the role of the narrative persistence and/or anticipation is to decrease the instantaneous weight of the relationship.

Finally, we normalized the instantaneous weights in order to get values in $[0; 1]$. For this purpose, we selected a sigmoid function centered on zero, whose slope can be used to control how much the past and future states of a relationship affect its instantaneous weight.

5.2.3 Empirical Validation

In order to assess our method, we constituted a corpus based on three popular TV serials: *Game of Thrones*, *Breaking Bad*, and *House of Cards*. This corpus is called *Serial Speakers*; it was the object of a publication [VL26] and it is publicly available online [C5]. We first summarize our results related to the detection of the interactions between characters, before comparing narrative smoothing to more traditional methods.

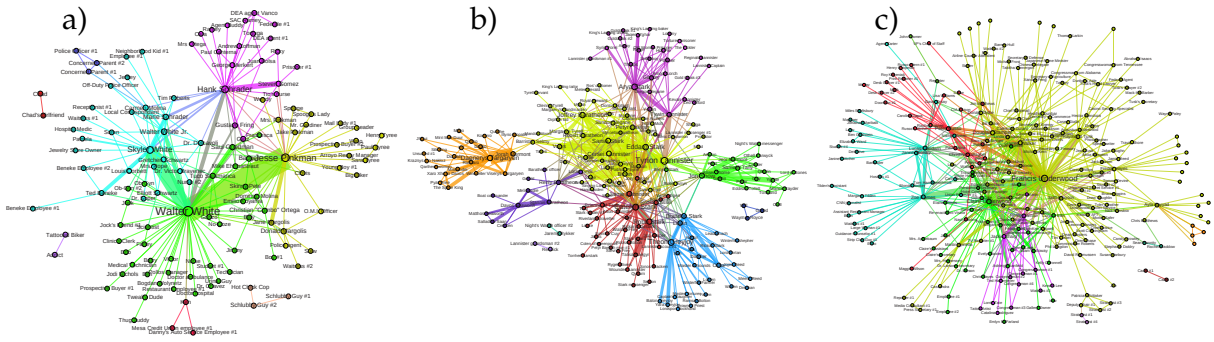

Figure 5.4: Static cumulative graphs extracted from three TV serials using narrative smoothing: *Breaking Bad* (a), *Game of Thrones* (b), and *House of Cards* (c). Colors show the detected community structure.

Conversational Interactions We evaluated our interaction extraction method in two ways. First, *directly*, by focusing on the interactions themselves; and second, *indirectly*, by considering the whole conversational network at once. In both case, we used a ground truth that we manually constituted ourselves. Regarding the direct evaluation, the character that speaks is known, so identifying a verbal interaction amounts to putting an utterance into one or several classes, each one corresponding to a possible listener. Therefore, this tasks can be viewed as a multilabel categorization problem, whose performance can be assessed using standard Information Retrieval measures [226]: *Precision*, *Recall* and *F-measure*. Regarding the indirect evaluation, we focused on the cumulative network, i.e. the static graph obtained by summing all weights over all time slices (cf. Figure 5.4), in order to ease the comparison. We adopted the method of [2], which consists in first converting the adjacency matrix of both compared graphs into two vectors by simple column concatenation; and second computing the similarity between these vectors. We used the *Jaccard Similarity* to focus only on the presence/absence of the edges, and the *Cosine Similarity* to take their weights into account.

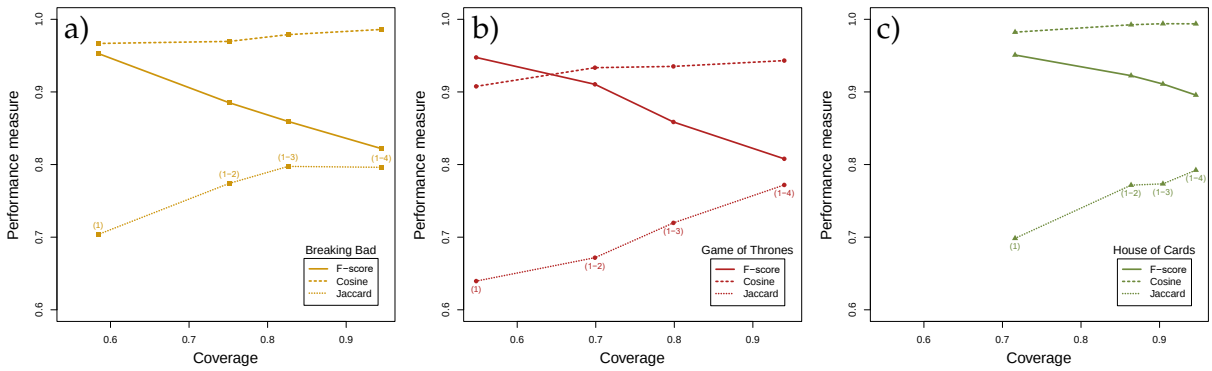

Figure 5.5: Step-by-step evaluation of the rules used for sequentially estimating verbal interactions. Each plot is dedicated to a specific TV serial: *Breaking Bad* (a), *Game of Thrones* (b), and *House of Cards* (c). The coverage (x-axis) indicates the proportion of utterances for which the rules can identify at least one listener. The performance (y-axis) is expressed according to the three measures listed in the legend and described in the text.

We evaluated our verbal interaction rules following a step-by-step process, by successively using in conjunction to the first, most robust rule, the three remaining ones. Figure 5.5 shows the changes when applying a more and more comprehensive set of rules, from the single first one, denoted *(1)*, to the whole four rules, denoted *(1–4)*. The changes are expressed in terms of *Coverage* (proportion of utterances processed by applying different subsets of rules), and performance (directly through the F-score, and indirectly through the network similarity measures). Table 5.1 displays the performance values obtained when using jointly all four rules.

TV Serial	Coverage	F-measure	Precision	Recall	Jaccard Sim.	Cosine Sim.
Breaking Bad	0.94	0.82	0.84	0.80	0.80	0.98
Game of Thrones	0.94	0.81	0.82	0.80	0.77	0.93
House of Cards	0.95	0.90	0.90	0.89	0.79	0.99

Table 5.1: Evaluation of the joint use of the four rules applied for sequentially estimating speakers interactions.

Not surprisingly, the more rules are used, the more interactions are hypothesized. The very basic first rule (*surrounded speech turn*) allows in average to hypothesize interlocutors for 62% of the spoken segments. When the whole set of rules is used, decisions are made for 94% of the utterances. The remaining utterances correspond to soliloquies or isolated utterances. More surprisingly, Figure 5.5 shows that the additional rules *(2–4)* introduce more and more mistakes when hypothesizing interlocutors at the utterance level, resulting in a decreasing F-score. But, in the meantime, the indirect evaluation measures of the resulting network improve. Such a discrepancy suggests that errors made locally when assigning each utterance to the addressed characters do not deteriorate the reliability of the resulting conversational network. Indeed, only a small proportion of the errors made at such a local level introduces irrelevant edges in the resulting cumulative network. Moreover, some errors made at the utterance-level by using more and more covering rules allow retrieving interactions that would otherwise have been missed, or improperly weighted, by using only Rule 1. Rules 2–4 tend to introduce correct interactions, but at wrong places, and finally result in more reliable conversational networks. In other words, the errors consisting in misplacing an interaction in time do not affect the cumulative network, in which time is integrated. Though basic, our four heuristics turn out to be very effective when building such cumulative conversational networks.

Narrative Smoothing In this part, we compare the network obtained through narrative smoothing with two networks extracted using the traditional sliding window approach, using 10-scene and 40-scene windows. These sizes correspond approximately to half an episode and two episodes, respectively. We start our analysis by focusing on the *protagonists* of the considered TV serials. As an illustration, I only discuss a few characters of interest here. We characterized them using the vertex *strength* (weighted generalization of the degree) which is, in our case, related to how much and how frequently a character speaks to others.

The case of Daenerys Targaryen and Tyrion Lannister, two major protagonists of *Game of Thrones*, illustrates well the limitations of time-windowing approach. Figure 5.6 displays the evolution of their strengths over

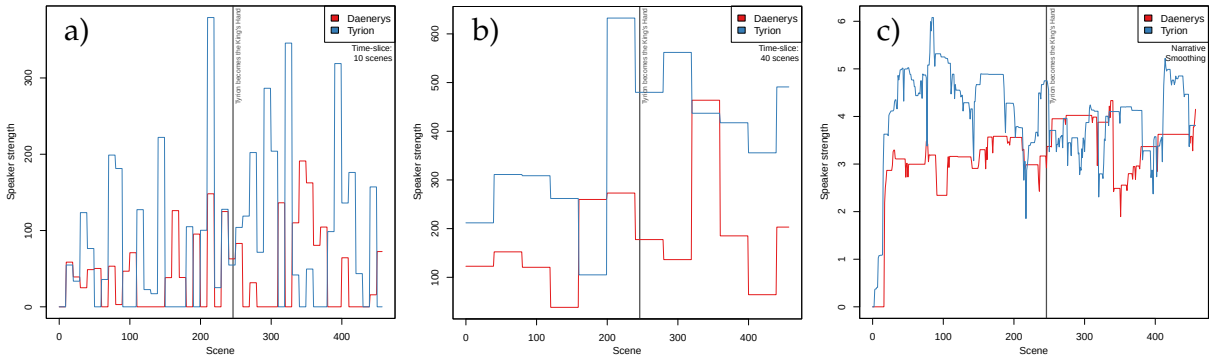

Figure 5.6: Strengths of two major characters of *Game of Thrones* plotted as functions of the chronologically ordered scenes: a) 10-scene sliding window; b) 40-scene sliding window; and c) narrative smoothing.

the first two seasons, as a function of time (expressed in terms of scenes), using the sliding window approach and narrative smoothing. The appearance of Daenerys' storyline onscreen has a relatively slow pace in these seasons and as can be seen, when the window is too narrow, this creates noisy, irrelevant measurements of her narrative importance (Figure 5.6.a). It appears very unstable because her storyline alternates with many others on the screen. A wider observation window (Figure 5.6.b) is more likely to cover successive occurrences of Daenerys in the narrative, but, unlike our narrative smoothing method, prevents from locating precisely the scenes responsible for Tyrion's current importance. For instance, a local maximum in Tyrion's strength is reached at scene 247 (Figure 5.6.a), just after a major narrative event took place – the nomination of Tyrion as the King's Counselor (represented as a vertical line in the plots). Such an event remains unnoticed when accumulating the interactions during too large time-slices (Figure 5.6.b), but is well captured by our approach (Figure 5.6.c).

Figure 5.6 also reveals an important property of narrative smoothing. Because the past (resp. future) occurrences of a particular relationship are still (resp. already) active as long as the involved characters do no interact with others in the meantime, the respective strengths of the main characters of the story appear remarkably balanced. Although Tyrion looks much more central than Daenerys in the sliding window-based networks, whatever the window size, Daenerys is nearly as central as Tyrion in the network based on our narrative smoothing method: few of her acquaintances are shown onscreen as interacting with others. On the opposite, the story focuses more frequently on Tyrion, but also on separate interactions of his usual interlocutors, weakening his instantaneous strength (especially after scene 252): the dynamic strength, as computed after applying narrative smoothing, does not reduce to a global centrality measure, but also corresponds to a more local property, that measures how exclusively a character is related to his/her social neighborhood.

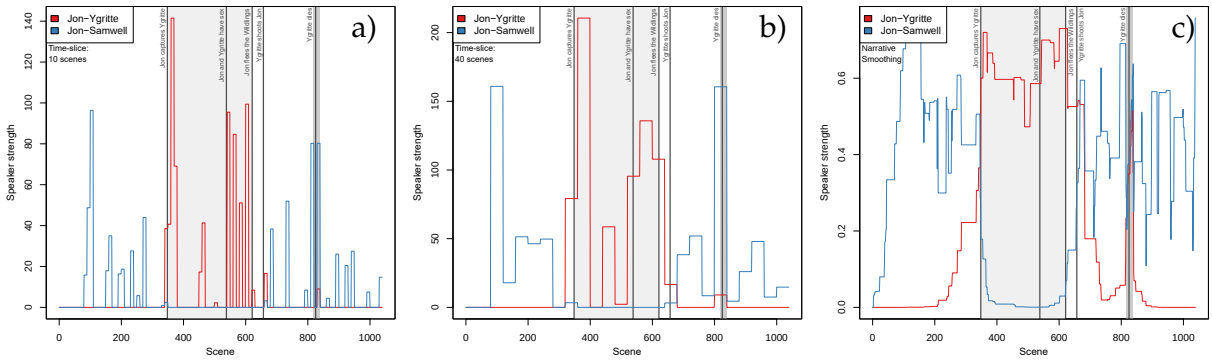

Figure 5.7: Weights of two relationships between three major characters of *Game of Thrones*, plotted as functions of the chronologically ordered scenes: a) 10-scene sliding window; b) 40-scene sliding window; and c) narrative smoothing.

We now consider relationships between pairs of characters, instead of single individuals. We characterize each relation depending on its weight. Like for the protagonists, we focus on relationships of particular interest. Figure 5.7 focuses on two relationships from *Game of Thrones*: on the one hand, the romantic relation

between Jon Snow and Ygritte, a Wildling, and on the other hand, the friendship between Jon and Samwell Tarly, who also serves in the Night's Watch. Five important events of the Jon–Ygritte relationship are marked by vertical lines: 1) their first encounter, when Jon captures Ygritte; 2) the moment when they have sex; 3) their separation, when Jon escapes the Wildlings; 4) the vengeance of Ygritte, when she shoots Jon with her bow and arrows; and 5) the death of Ygritte, during the Wildling attack of Castle Black. The first grayed area represents the duration of the romantic relationship between Jon and Ygritte, and the second is the battle of Castle Black. Samwell and Ygritte belong to two separate social groups, with which Jon alternatively interacts: Samwell when he is at Castle Black with the Night's Watch, and Ygritte when he is north of the wall with the Wildlings. This separation very clearly appears in the plot generated through narrative smoothing (Figure 5.7.c). One can distinguish periods of exclusive relationships: with Sam before the capture of Ygritte (first mark), with Ygritte during their romantic relationship (first grayed area); but also periods where Jon interacts with both, such as the Battle of Castle Black (second grayed area). Our method also allows detecting important events, corresponding to peaks of edge weight, such as the first time Ygritte and Jon have sex, or the Battle of Castle Black. By comparison, and as noted before, both methods based on time windows fail to show the continuity of these relationships, which appear to be very sporadic in Figures 5.7.a and 5.7.b. This is particularly true of the Jon–Ygritte romantic relationship, which appears as completely discontinuous. Of course, this irregularity also hides important events, which do not stand out among these large fluctuations of edge weight. Moreover, certain important events are just not associated to important weights, such as the death of Ygritte (rightmost vertical mark).

5.3 Multimodal Segmentation

In this section, I summarize the method that we proposed to leverage the social network extracted through narrative smoothing. This method is multimodal in the sense that it also takes advantage of lower level visual and acoustic features. We adopted an extractive approach, consisting in constituting the video summary by using excerpts of the TV serial. Moreover, we aimed at producing a *character-oriented* summary, i.e. the video must describe the story from the perspective of a character of interest. Our general approach was as follows. First we segmented the video in a sequence of so-called Logical Story Units (LSU). Second, we characterized each LSU using a selection of multimedia and social features. Third, based on these features, we selected the most appropriate LSUs and built the summary. I summarize these different steps here, but the interested reader can find more details in [VL5]. In Section 5.3.2, I introduce the notion of LSU and describe the features that we used to describe them. Then, in Section 5.3.3, I focus on the method that we proposed to select the most appropriate LSUs. Finally, in Section 5.3.4, I present the ground survey that we set up to assess its relevance, and our main results.

5.3.1 Segmentation of the Narrative

The first step in our method consisted in segmenting the storyline associated to a specific character into a sequence of *narrative episodes*. In any narrative, the story of a specific character usually develops sequentially and advances in stages: each narrative episode is defined as a homogeneous sequence where some event directly impacts a specific group of characters located in the same place at the same time. Though such a notion of narrative episode may be defined at different levels of granularity, such sequences are often larger than the formal divisions of books in chapters and of TV serials in episodes.

Based on the dynamic network of interacting characters produced through narrative smoothing, as explained in Section 5.2, we defined the *relationship vector* $r_t(v_i)$ of a character v_i at time t. It is constituted of the weights, ranging between 0 and 1, of his/her relationships with all the other characters during the t^{th} scene. Put differently, it is the i^{th} row (or column) of the graph adjacency matrix, which characterizes the character's neighborhood at time t. We can then compute a dissimilarity matrix $\mathbf{D}(v_i)$, where $d_{tt'}(v_i)$ is the normalized Euclidean distance between the character's relationship vectors $r_t(v_i)$ and $r_{t'}(v_i)$ at times t and t'. Because each narrative episode is defined as impacting a limited and well-identified group of interacting characters,

we assume that the relationships of a character stabilize during each narrative episode, and change whenever a new one starts.

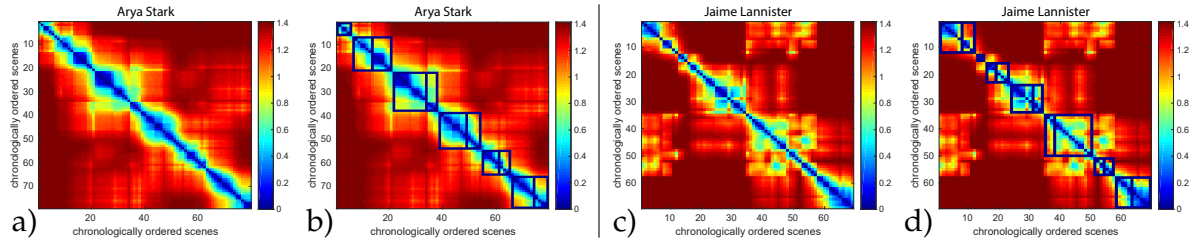

Figure 5.8: Dissimilarity matrices computed over the first five seasons of *Game of Thrones*, for two major characters: Arya Stark (a) and Jaime Lannister (c). For both characters, the second matrix (b, d) is similar, but additionally shows the narrative episodes.

Figures 5.8.a and 5.8.c show the dissimilarity matrices obtained for *Game of Throne*'s Arya Stark and Jaime Lannister, respectively. For the sake of clarity, I show only the scenes where the characters are involved, even though our narrative smoothing method can provide the character neighborhood in any scene. The diagonal blocks appearing in the matrices confirm that the characters' social environment is not continuously renewed as the storyline develops, but stabilizes for some time, before being replaced by a new social configuration. Moreover, the fact that larger blocks contain smaller ones confirms the multi-scale nature of the notion of narrative episode.

In order to formally identify these blocks corresponding to narrative episodes, we formulated the task as a variant of the set covering problem [71]. First, we set a constraint of temporal contiguity on the elements of the admissible subsets of scenes, so that to keep narrative episodes continuous over time. Second, in order to obtain a covering as close as possible to a partition, we minimized the overlap between the covering subsets instead of minimizing their number as in the standard formulation of the set covering problem. Despite this adapted objective, some relationship vectors at the boundaries between two consecutive narrative episodes may still belong to both of them: in this case, we assigned the duplicated vector to the closest relationship state. The user must set a threshold τ corresponding to the maximal admissible distance between the most covering relationship vector in each narrative episode and any other relationship vector within this narrative episode. It can be interpreted as the level of granularity desired when analyzing the story.

Figures 5.8.b and 5.8.d show the partition obtained for Arya Stark and Jaime Lannister, for a granularity level $\tau = 1.0$. Each narrative episode is represented as a box containing a vertical line. This line corresponds to the scene in which the relationship state covers at best the narrative episode. In such scenes, we call the relationship vector *representative*, as it can be regarded as conveying the typical social environment of the character within the associated narrative episodes.

5.3.2 Characterizing Logical Story Units

Extractive video summaries are generally not constituted of single shots picked from the original stream, but rather of short sequences of about 10 seconds, consisting of a few consecutive shots. Such sequences are usually selected not only because of their semantic relevance, but also because of their semantic cohesion and self-sufficiency. From a computational perspective, identifying such sequences in the video stream as potential candidates for later insertion in the final summary remains tricky. The stylistic patterns widespread among filmmakers are particularly relevant, because they are often used to emphasize the semantic consistency of these sequences. For instance, a dialogue between two persons will generally be filmed using a sequence of shots/countershots in order to keep the exchange natural, a convention called the *180-degree rule*. More generally, several sets of such recurring shots may overlap each other, resulting in possibly complex patterns well-suited for segmenting movies into consistent narrative episodes. Hanjalic *et al.* [118] call such sequences of intertwined recurring shots *Logical Story Units* (LSUs), and introduced a method for automatically extracting them. Figure 5.9 shows a sequence of five shots with one recurring shot in positions 1, 3, 5, resulting in one LSU.

Figure 5.9: Example of Logical Story Unit (LSU) with one surrounding, recurring shot in positions 1, 3, 5; extracted from *Game of Thrones*.

We used LSUs as the basic candidate units selected when building our summaries. In order to detect such LSUs, we first split the whole video into shots, which we then compared and labeled according to their similarities. Both tasks rely on image comparison, which we performed based on color distribution and the standard block-based comparison technique of Koprinska and Carrato [144]. Once identified, similar shots can be used as a basis for automatically extracting every LSU. Instead of the standard graph-based algorithm of Yeung *et al.* [238], we defined an alternative matrix-based method, which we describe in [VL5]. We then estimated the relevance of each identified LSU for possible insertion into the summary according to three criteria. The first two rely on techniques commonly used by filmmakers to tell the story, and are related to the form of the narrative. The third is related to the content of the storyline associated to the considered character.

The *shot size* represents how much of the frame is filled by the characters' faces in a shot of interest. Concretely, we sample the frames constituting the shot, detect faces using a standard method, retain the largest faces found, and express the shot size as a proportion of the frame size. To get the shot size of a LSU, we simply average over its constituting shots. We note ss_θ the shot size of the θ^{th} LRU.

The *musicality* measures how dominant music is over speech in the audio track of the LSU. Briefly, we used the method of Giannakopoulos *et al.* [110] which relies on so-called *chroma vectors* to distinguish the background music from speech. They convey the distribution of the audio signal over the twelve notes of the octave, which is different for music and speech. We note m_θ the musicality of the θ^{th} LRU.

Unlike the previous features, the *social relevance* is necessarily relative to the considered character. This measure aims at discriminating the LSUs showing some of the character's typical relationships within each narrative episode of his storyline. For this purpose, we compared the neighborhood of the character for this LSU with his/her typical neighborhood for the whole narrative episode. Concretely, we computed $sr_\theta(v_i) = \text{Sim}(r_t(v_i), \omega_\theta(v_i))$, where Sim is the cosine similarity, $r_t(v_i)$ is the character's *representative* relationship vector for the narrative episode, and $\omega_\theta(v_i)$ is the character's vector of ω scores as computed in Section 5.2.1, over the θ^{th} LSU. This $\omega_\theta(v_i)$ was therefore obtained by computing the interaction times between the character and every other character over the LSU.

We min-max normalized the average musicality so that our three features ranged between 0 and 1. We then linearly combined them into a single measure of relevance (Definition 5.3.1).

Definition 5.3.1 (Relevance Score) *The* ***relevance score*** $p_\theta(v_i)$ *of the* θ^{th} *LSU for character* v_i *is*

$$p_\theta(v_i) = \lambda_1 sr_\theta(v_i) + \lambda_2 ss_\theta + \lambda_3 m_i. \tag{5.4}$$

The λ weights must be set by the users, depending on their specific needs. On the one hand, emphasizing social relevance is expected to result in more informative summaries, able to help the user remember the plot; and on the other hand, emphasizing music and shot size is expected to result in trailer-like summaries.

5.3.3 Selecting Logical Story Units

Character-oriented summaries aim at reflecting the dynamics of a character's storyline. Once isolated by applying the segmentation method described in Section 5.3.1, each narrative episode of a specific character's storyline should be equally reflected, whatever its duration, as a major development in his/her story. The next step is to select the most relevant LRUs based on $p_\theta(v_i)$ to represent each narrative episode. Besides the

the level of granularity τ of the narrative episodes, and the weighting scheme of Equation (5.4), our algorithm requires the user to specify $T_{\max}$, the maximal duration devoted to each episode in the summary.

Each narrative episode is a sequence of candidate LSUs. Summarizing such an episode can be regarded as a task consisting in selecting a sub-sequence of these LRUs, with two joint objectives and a length constraint: the summary must not exceed the duration $T_{\max}$ and aims at containing not only relevant sequences, but also sequences that remain as diverse as possible, in order to minimize redundancy. McDonald [169] showed that such a summarization problem can be formulated as the quadratic knapsack problem presented in Definition 5.5.

Definition 5.3.2 (LSU Selection Problem) *For a given character v_i, let x_θ a binary variable set to 1 if the θ^{th} LSU is inserted in the summary of this character, and to 0 otherwise; and ℓ_θ the duration of this LSU.*

*The **LSU Selection Problem** is*

$$\begin{aligned} \max f(\mathbf{x}) &= \left(\sum_{i=1}^{n} p_\theta(v_i) x_\theta + \sum_{\theta=1}^{n} \sum_{\theta' \neq \theta} d_{\theta\theta'} x_\theta x_{\theta'} \right) \\ \textit{subject to } & \sum_{\theta=1}^{n} \ell_\theta x_\theta \leq T_{\max} \\ & x_\theta \in \{0,1\}, \theta \in \{1,\ldots,n\}. \end{aligned} \tag{5.5}$$

Such a formulation of the summarization problem can be tricky to solve exactly for large instances, even when linearizing the quadratic part of the objective function [169], and heuristic methods provide us with much more scalable, though possibly sub-optimal, resolution techniques. In [VL5], we introduced a greedy heuristic for iteratively selecting optimal LSUs, based on a generalization to the quadratic case of the greedy heuristic used to solve the linear knapsack problem in [14, 98]. For each narrative episode, a summary is built until the time duration limit T is reached, and the final character-oriented summary is made of the concatenation of all the LSUs, chronologically re-ordered, selected in every narrative episode. The total complexity of our summary extraction method is in $O(n^2)$, where n is the number of LRUs.

5.3.4 Experiments and Results

In order to assess our method for automatically generating character-oriented summaries of TV serials, we performed a large scale user study based on a real case scenario. A few weeks before the sixth season of the popular TV serial *Game of Thrones* was released, we asked a group of 187 university students and staff to answer a questionnaire in order to evaluate automatic summaries extracted from the serial with our method. The characteristics of the participants are detailed in [37, VL5]. They were asked to evaluate summaries covering the first five seasons of the serial, and centered on the storylines of five different characters to ensure generalizability: Arya Stark, Daenerys Targaryen, Jaime Lannister, Sansa Stark and Theon Greyjoy.

Generation of the Summaries We generated the summaries using three variants of our method. First, a full summary (denoted *full*), built upon the method as described in this chapter. It thus leveraged both the content and the style of the narrative. In order to keep the summary duration into reasonable boundaries, we set the granularity level to $\tau = 1.0$ and the duration devoted to each narrative episode to $T_{\max} = 25$ seconds. Second, a style-based summary (denoted *sty*), was built only upon the stylistic features, by setting the social relevance feature weight to zero. By discarding social relevance, there is no longer need for the pre-segmentation of the storyline into narrative episodes. As a consequence, the candidate LSUs are not selected among the separate subsets resulting from the segmentation step; instead they are considered as a whole single set of candidate sequences, weighted equally according to their average musicality and shot size, and finally selected by iteratively applying our selection algorithm until the resulting style-based summary has roughly the same duration as the full summary. Third, a baseline, semi-random summary (denoted *bsl*) was obtained as follows: some non-overlapping LSUs where the considered character was verbally active

were first randomly selected until reaching a duration comparable to the duration of the first two kinds of summaries; the selected LSUs were then re-ordered chronologically when inserted in the summary.

The final summaries turned out to be quite short, ranging from 1:30 to 2:50 minutes, resulting in very high compression rates: the whole story of a character during 50 one-hour episodes was summarized in about two minutes. The total time of the summary depends on the number of narrative episodes in the character's storyline: characters with fast-evolving social environments, going through more narrative episodes may therefore need longer summaries than possibly more important characters involved in fewer narrative episodes. The criterion used when building the full and style-based summaries results in summaries consisting of shorter sequences than the baseline summary: while the candidate LSUs last a bit more than 10 seconds in average, the duration of the ones selected in the *full* and *sty* summaries is very close to 5 seconds.

The computation time for dynamically generating the summaries on a personal laptop (Intel Xeon-E3-v5 CPU) was quite low (0.8–3 seconds, depending on the method and character), once LSUs, shot size, background music, along with the dynamic network of interacting characters, had been pre-computed once and for all. Finally, the three summaries may overlap, also depending on the considered methods and character. The largest overlap time was between *full* and sty (33% in average); then *sty* and bsl (8%); and finally textitfull and bsl (3%).

Survey and Performance The respondents were asked to rank the three summaries for each character, according to both criteria traditionally used for subjective evaluation: *informativeness* and *enjoyability*. The questions asked were formulated as follows: 1) *Which of these three summaries reminds you the most the character's story?*, or *Best as recap?* for short; and 2) *Which of these three summaries makes you the most want to know what happens next to the character?*, or *Best as trailer?*. The same questions were asked for their last choice, resulting in a full ranking of the three summaries. In addition, the participants were asked to motivate in a few words their ranking. The obtained answers are reported in Table 5.2.

Character	***Best as recap?***			***Best as trailer?***		
	full	***sty***	***bsl***	***full***	***sty***	***bsl***
Arya Stark	**70.9**	9.3	19.8	**57.1**	16.7	26.2
Daenerys Targaryen	**35.8**	32.8	31.3	18.2	**47.0**	34.8
Jaime Lannister	**41.5**	40.0	18.5	35.9	**43.8**	20.3
Sansa Stark	**47.7**	33.8	18.5	**58.5**	20.0	21.5
Theon Greyjoy	15.6	**45.3**	39.1	14.3	**57.1**	28.6
Average	**42.3**	32.3	25.4	36.8	**36.9**	26.3

Table 5.2: Feedback of the respondents regarding the summaries generated for the five selected characters, using the three methods: full summary (*full*), stylistic features only (*sty*), and semi-random baseline (*bsl*). Each percent indicates the proportion of respondents that preferred the concerned summary.

The baseline summaries never obtained a majority vote, whatever the ranking criterion. Concise summaries with short, socially diverse sequences extracted from every narrative episode were globally well perceived, and turn out to be worth the slight extra computation time. For 4 out the 5 characters targeted, the full summary was selected as the most efficient recap, in some cases by far: Arya's story full summary was considered as the best recap by 70.9% of the participants. Interestingly, the both other summaries turn out to miss some major narrative episodes, and were perceived as incomplete by many participants. For 4 out of the 5 characters, the full summaries obtain higher scores when judged as recaps than when judged as trailers. This difference globally benefits the style-based summaries, more appreciated as trailers than as recaps for 4 characters out of 5. For 3 characters, such style-based summaries even obtain a majority vote according to the "trailer" criterion.

However, some of the votes were sometimes unexpected. First, the three summaries of Daenerys' storyline obtain roughly similar scores when evaluated as recaps, without any clear advantage for the full summary. Daenerys is a key-character of the serial, with the unique ability to command to dragons. Many participants focused on this aspect to assess her summaries: dragons are absent from the full summary, but can be heard in the style-based one, and happen to be present in the baseline one, completely by chance. Some

respondents used this criterion to discard the full summary, though being the only one that captured her crucial meeting with Tyrion Lannister. The scores obtained by Theon's summary were also surprising, with quite low scores for the full summary, probably penalized by a baseline summary rather semantically consistent and convincing, though somehow incomplete ("Fall of Winterfell" and "Final Reunion with Sansa" missing).

In conclusion, on the one hand, the results globally strengthened our plot modeling approach when it comes to summarizing the dynamics of a character's storyline over dozens of episodes. On the other hand, the stylistic features that we used to isolate salient sequences remained too hazardous when used on their own for capturing the whole character's storyline, but turned out to be valuable to make viewers feel like viewing what comes next.

5.4 Conclusion and Perspectives

In this chapter, I described the work conducted in the context of Xavier Bost's PhD, and which led to the design of a method able to generate extractive video summaries from TV serials. This method is based on stylistic aspects of the video itself, but more importantly for us, on a graph-based modeling of the TV serial plot. This model takes the form of a dynamic network representing the evolution of the conversational interactions between the characters. We proposed a method to extract this network, and then segment its timeline from the perspective of a character of interest. We used this segmentation to identify narrative episodes in the storyline of the character, that we leveraged to identify assess scene relevance when building the summary. The field survey that we conducted allowed us to validate empirically the relevance of the generated summaries. As mentioned before, both our source code and data are available online publicly [C5, C6]; and we published the latter as a proper corpus [VL26].

Project-wise, this work can be extended in many ways, especially on the video summary side, for instance by expanding the selection of low level features used in conjunction with our network-based social relevance. However, my personal work focuses more on the network side, on which I am currently working with Xavier Bost. First, we recently wrote a survey on the extraction and analysis of fictional networks *in general* [VL6], by expanding our scope to other works of fiction than TV serials: novels, plays, cartoons, live action and animated movies, etc. We are also advising several master students on the task of extracting such networks from novels and screenplays, through a variety of NLP methods [U5, U2] (cf. Figure 5.10). Our goal there is to explore dynamic graphs as a modeling framework for plots. As shown in our survey, a number of authors already model plots using character interaction networks, but most use *static* networks, therefore ignoring an important aspect of the plot. Besides automatic summarization, we plan to tackle various problems based on such networks, like story decomposition, classification (e.g. classifying novels by genre), and recommendation.

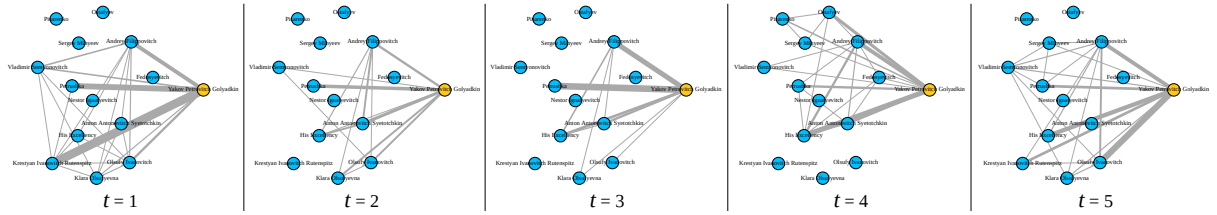

Figure 5.10: Example of dynamic network extracted from the novel *The Double*, by Fyodor Dostoevsky. Edges represent co-occurrences; and the main protagonist, Yakov Petrovitch, appears in orange.

This work on dynamic fictional character networks is related to several other aspects of my research activity mentioned in this manuscript, and I want to stress this convergence to conclude this chapter. First, from the perspective of the network extraction method, it is similar to my ongoing collaboration with Guillaume Marrel and Frederic Monnier, already mentioned in the conclusion of Chapter 3, which aims at extracting networks of interacting historical figures based on collections of biographies. The main differences are the nature of the considered texts, which are not fictional and thus present different difficulties when processed by NLP

tools; and the use of static graphs due to the lack of temporal information. Second, the work conducted with Richard Dufour in the context of Noé Cécillon's PhD [D2], also mentioned in Chapter 3, and based on the extraction and classification of conversational networks from chat messages, is connected too. The extraction method relies on speech turns, similarly to our method from Section 5.2.1; and we consider the state of the conversational graphs before and after some time of interest, which can be considered as a dynamic network, even if the most basic that can be, with only $T = 2$ time slices. We are currently in the process of extending our study in two ways: by extracting longer dynamic networks and using graph embeddings to describe them. I plan to use similar embedding methods to describe the dynamic networks obtained from works of fiction.

Spatial, Signed, and Multiplex Networks

6

Straightness of Spatial Networks

6.1 Context

In a spatial graph, vertices and edges hold a position in a metric space [19]. Figure 6.1 shows an example of random spatial network, and the concept of spatial network is presented formally in Definition 6.1.1. A spatial graph can be viewed as a specific type of attributed network (cf. Definition 2.1.2), in which each vertex is described by two numerical attributes denoting its spatial coordinates. However, in the literature they are considered apart, due to the specific semantics of these attributes. Such graphs allow modeling a variety of real-world systems whose topology is affected by spatial constraints, and they are particularly popular in quantitative geography [200, 237]. For instance, a spatial graph can be used to model the road transportation system of a city, the edges and vertices representing the streets and crossroads, respectively.

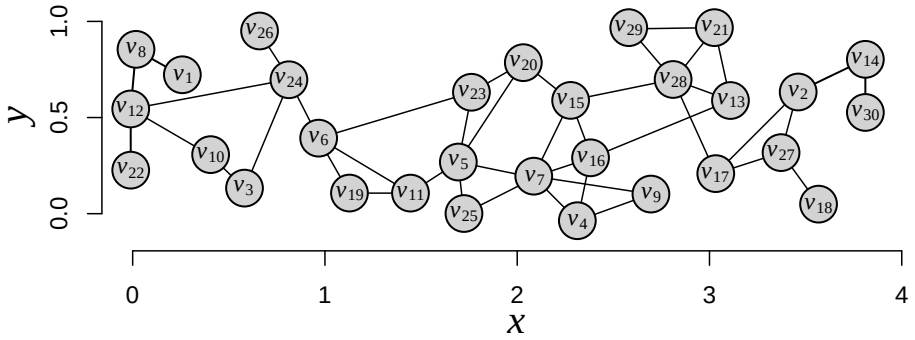

Figure 6.1: Example of spatial network containing 30 vertices uniformly distributed over $[0;4] \times [0;1]$.

Definition 6.1.1 (Spatial Network) *Let $G = (V, E)$ a **spatial network**, where $V = \{v_1, \ldots, v_I\}$ is the set of vertices, and $E \subset V^2$ is the set of edges. If the graph is undirected, we assume that edges are lexicographically ordered pairs. Each vertex $v \in V$ is characterized by a **position** $(x_v, y_v) \in \mathbb{R}^2$ in the Euclidean plane. Several distinct vertices can **coincide**, i.e. hold the same position. An **edge** (u, v) can be viewed as a straight segment connecting two vertices $u, v \in V$.*

These networks have been used as a modeling tool in Geography since the end of the 1960s, according to Barthélemy [19]. The recent development of the field of complex network analysis helped increase the activity

regarding spatial networks [96, 109]. Moreover, the interdisciplinary nature of this field eased methodological exchanges between otherwise completely separated application fields. Indeed, Geography is not the only domain needing to take spatial information into account when dealing with graphs: a number of relational systems undergo spatial constraints that affect their structure and/or functioning, such as biological neural networks, the Internet, or physically interacting social groups.

My first work on spatial networks took place in 2012 at Galatasaray Üniversitesi, in collaboration with Atay Özgövde, and dealt with Wireless Sensor Networks (WSN) [VL43]. I already mentioned it in Chapter 4, because the dynamic nature of the studied graphs was more important than their spatial nature. The analysis methods which we proposed at the time did not take the spatial aspects into account, as it was not required to solve the problem at hand. I had the occasion to work again on spatial networks when I arrived in Avignon in 2014, through a collaboration with geographer/geomatician Didier Josselin. He had set up an interdisciplinary CNRS PEPS-funded project called MoMIS[11] that included geographers, mathematicians, computer scientists, and biologists. The aim of this project was to study how certain human-built structures mimic the orb-webs woven by certain spiders. The point was not only to compare them, but also to understand the reason of this similarity, through the analysis of the properties of orb-webs and other geometrical networks. I joined the project and participated in this study [VL61, VL35], a work which I describe in Section 6.2. I also did some related work on my own, focusing on the computation of the Straightness [VL8], a measure designed to characterize spatial networks, which I present in Section 6.3. I conclude this chapter by summarizing my contributions, and describing an ongoing work on spatial networks in Section 6.4.

6.2 Study of Geometric Networks

This section summarizes the work presented in [VL61, VL35], and corresponds to my participation to the MoMIS project. I first set the general context and introduce the various types of networks considered during this project, as well as the Straightness measure that we used to characterize them (Section 6.2.1). I then turn to the main results obtained during this work, and the methods used to get them. We studied these networks by considering separately center-to-periphery routes (Section 6.2.2) and all types of routes (Section 6.2.3). Note that the source code corresponding to this work is publicly available online [S12].

6.2.1 Networks and Straightness

The MoMIS project participated in a larger scientific trend aiming at studying urban biomimetism. Its general objective was to compare, in terms of both organization and usage, the webs woven by certain types of spiders with certain man-made systems exhibiting highly similar structures. The assumption behind this project was that, on a geological time scale, spiders developed optimal strategies to forage resources available in their environment [145], and that this concerns in particular the way their webs are woven. The question is then to know whether such structures retain some of their qualities when humans adopt them to organize their artificial systems, such as transport infrastructures, energy grids, or communication nets.

We focused on orb-webs, which present the following structure. Such a network is built around a central vertex, from which a few radii originate. A spiral starts from this center and goes from radius to radius until it reaches the most peripheral point. Figure 6.2.a shows a real-world orb-web, whereas Figure 6.2.b is a graph corresponding to an idealized version of an orb-web. Radio-concentric networks, such as the one displayed in Figure 6.2.c, are built as series of concentric circles around a fixed point, and are therefore very similar to orb-webs. Both orb-webs and radio-concentric graphs are built around a central vertex at the origin of several radii. The difference between them is that in orb-webs, the spires (non-radial segments) are uninterrupted from the center to the most peripheral vertex, constituting a spiral; whereas in radio-concentric graphs, each concentric pseudo-circle is separated from the others. However, the latter is a good approximation of the former, as spires are quasi-parallel in orb-webs.

[11] MOdèles Mathématiques et Interactions Sociales – Mathematical models and Social Interactions

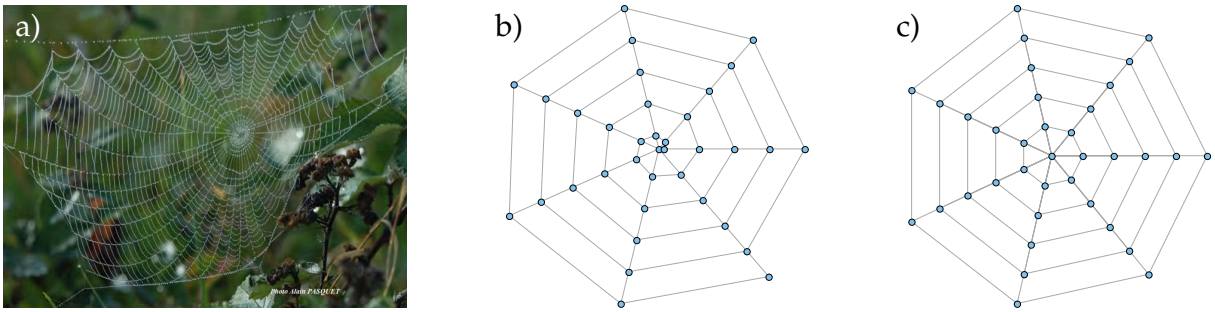

Figure 6.2: a) Actual orb-web (Source: Alain Pasquet). b) Graph representing an ideal orb-web. c) Radio-concentric graph used as an approximation of orb-webs.

Figure 6.3: a) Ancient theater of Orange, France (Source: DeAgostini). b) Suncity, Arizona (Source: Francois Gohier). c) Overhead view of the Gemasolar Thermosolar Plant, Seville, Spain (Source: Benjamin Grant).

The radio-concentric structure appears in a number of ancient and modern artificial systems, such as those displayed in Figure 6.3, but also parliamentary hemicyles, frameworks of certain buildings, irrigation systems, etc. The central vertex has generally a specific role related to control or monitoring. The analogy with orb-webs has been noticed in the literature, and the radio-concentric model has been abundantly studied, starting as early as the 1920s [189]. There was no work comparing them in a quantitative way though.

In addition to orb-webs and radio-concentric networks, we decided to also include several types of meshed graphs in our study. Rectilinear networks are based on a square mesh, as illustrated in Figure 6.4.a. They are particularly interesting, because some urban structures such as Manhattan (Figure 6.4.b) or the ancients cities of Kahun (Figure 6.4.c) and Miletus (Figure 6.4.d) are organized similarly. Graphs based on triangular, hexagonal, and octagonal meshes are not as widespread in real-world artificial systems, but we included them in our selection for matters of comparison.

Figure 6.4: a) Ideal rectilinear graph. b) Manhattan in 1840, (Source: J. Calvin Smith). c) Ancient Egyptian city of Kahun (Source: W. M. Flinders Petrie). d) Ancient Greek city of Miletus (Source: von Gerkan).

When characterizing a spatial graph, it is possible to adopt tools originally designed for plain graphs, such as the degree or transitivity. However, this amounts to completely discarding the spatial information. An alternative consists in computing measures relying on the concepts of shortest path or graph distance, but using the Euclidean distance between two connected vertices as the edge weight. This is a way of leveraging some of the spatial information while keeping a well-known tool, such as the closeness or betweenness centrality measures, for instance.

However, a number of measures have also been defined specifically to leverage spatial information [19]. In the context of this project, we focused on the *Straightness* [194, 229], which characterizes a pair of vertices. It corresponds to the ratio of the *graph* to the *Euclidean* distances between the vertices, as illustrated in Figure 6.5.a. Indeed, on a spatial graph, one can express the distance between two vertices v_1 and v_3 through two distinct measures. The first is the classic *Euclidean distance*, noted $d_E(v_1, v_3)$, and represented in dashed red in Figure 6.5.a. Based on this measure, we can define the *length of an edge* (v_1, v_2), which is the Euclidean distance $d_E(v_1, v_2)$ between its end-vertices. The *length of a path* is the sum of the lengths of its constituting segments. The *shortest path* between two vertices corresponds to the path of minimal length. The second measure is the *weighted graph distance*, or *graph distance* for short, noted $d_G(v_1, v_3)$ and represented in solid red in Figure 6.5.a. It is the length of the shortest path connecting v_1 and v_3 on the graph. Note that, since this is an Euclidean space, $d_E(v_1, v_3) \leq d_G(v_1, v_3)$.

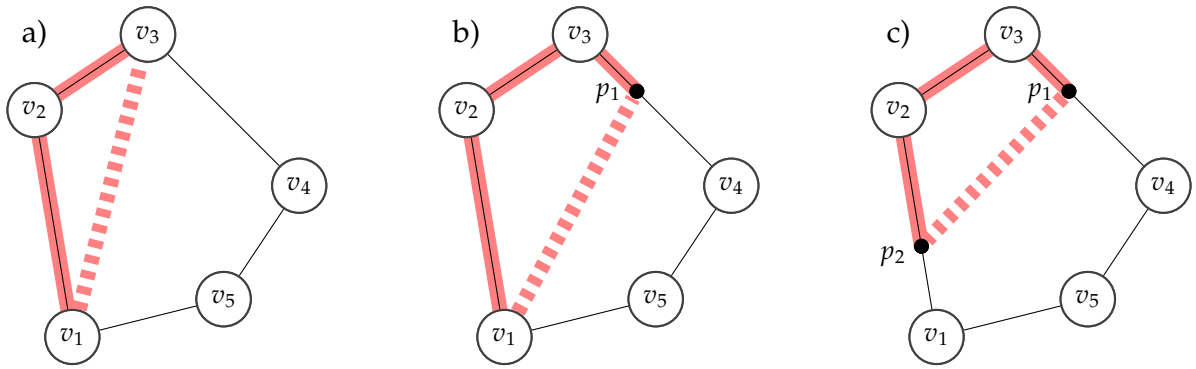

Figure 6.5: Representation of the Euclidean (dashed) and graph (solid) distances (in red) for three cases: between two vertices v_1 and v_3 (a); between a vertex v_1 and a non-vertex point p_1 (b); and between two non-vertex points p_1 and p_2 (c).

This description corresponds to a traditional use of the Straightness, in which one considers only routes going from one vertex to another. This is appropriate, for example, for graphs representing subway transportation systems: travelers can only go from one station (i.e. vertex) to another, they cannot get off or on the train anywhere else. However, there are also situations in which the route could start or end anywhere on an edge, and not necessarily exactly on a vertex. This is for instance the case, on a road network, when travelers use individual cars or simply walk: they can stop and start anywhere on a street (i.e. an edge), not necessarily at a crossroad (i.e. a vertex). In this case, one has to consider routes going from any point of the graph to any other point, even those located in-between vertices, as illustrated in Figures 6.5.b and 6.5.c. This is our case in this project, which is why we used a slightly generalized version of the Straightness, able to deal with point-to-point routes, and formalized in the Definition 6.2.1.

Definition 6.2.1 (Straightness) *Let G be a spatial graph, and P be the set of all its constituting points, including vertices ($V \subset P$). The Straightness $S(p_1, p_2)$ between two points p_1 and $p_2 \in P$ is the ratio of their Euclidean to their graph distances:*

$$S(p_1, p_2) = \begin{cases} 0 & \text{if } d_G(p_1, p_2) = +\infty, \\ 1 & \text{if } p_1 = p_2, \\ \dfrac{d_E(p_1, p_2)}{d_G(p_1, p_2)} & \text{otherwise.} \end{cases} \tag{6.1}$$

Generally speaking, since $d_E(p_1, p_2) \leq d_G(p_1, p_2)$, the maximal Straightness value is 1, which occurs when both distances are equal, i.e. when the shortest path on the graph between p_1 and p_2 is a straight line. On the contrary, it tends towards 0 when this path includes more and more detours. The ratio is not defined when $p_1 = p_2$: by convention, we set it to 1 since both distances are equal (to zero). When the points are not connected in the graph, the graph distance is conventionally $d_G(p_1, p_2) = +\infty$, and we set the Straightness to zero.

In summary, the Straightness compares the distance obtained when following the shortest path over the graph edges, to the distance as the crow flies. It quantifies how efficient the graph is in providing the most

direct path from one vertex to the other. The Straightness gets close to zero if the path is very tortuous, and it can reach one if it is completely straight. Sometimes, its reciprocal is used instead, under the names *Circuity* [184], *Directness* (though *in*directness would be more relevant) [123], *Tortuosity* [135], *Route Factor* [108] and *Detour Index* [160].

6.2.2 Center-to-Periphery Routes

We first focused on center-to-periphery routes in radio-concentric and rectilinear networks. Their straightness can be expressed analytically, as their structures are geometrically simple and symmetrical. Figure 6.6 provides an illustration of our notations and terminology. As mentioned before, we call *radius* (in red) an edge starting at the center of a radio-concentric network, and *spire* (in green) an edge connecting two consecutive radii. The shape of the network is controlled by a parameter k corresponding to the number of radii. Two consecutive radii are thus separated by an angle $\theta = 2\pi/k$. We assume that $k > 2$, because with $k = 1$ the spires are not defined, and with $k = 2$, the radii and spires coincide. In a rectilinear network (Figure 6.6.b), by analogy, we call *pseudo-radii* (in red) the edges starting at the center of the network, and *pseudo-spires* (in green) the rest of the edges. Moreover, we consider that $\theta = \pi/2$. The shape of a rectilinear network is controlled by c, the unique length of its edges. Let the center of the network O be the origin of a Cartesian coordinate system. We characterize a center-to-periphery route by an angle α, formed by the x-axis and the segment going from the network center O to the targeted peripheral point p.

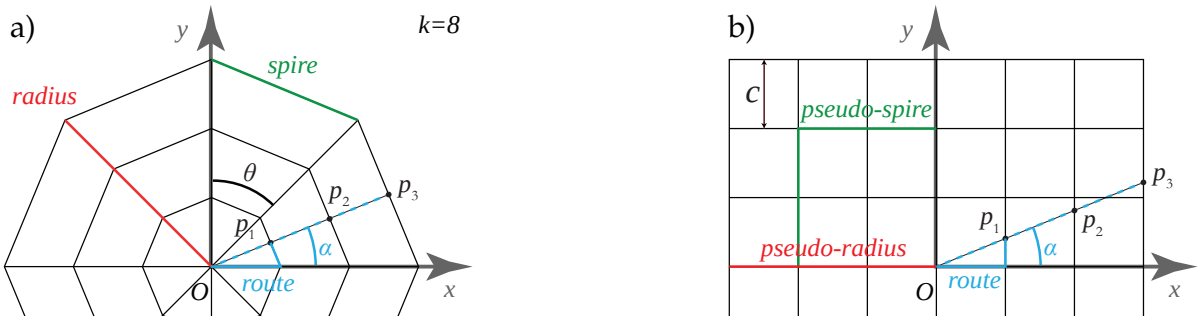

Figure 6.6: Terminology and notations used to describe the radio-concentric (a) and rectilinear (b) networks.

We took advantage of certain geometric properties of the networks to study only a subset of all possible routes, without loss of generality. First, both networks are invariant to *rotations* of angle θ, which allows us to focus on their first angular sector, i.e. the space between two consecutive radii, and the first quadrant, respectively. Second, both networks are *symmetrical* with respect to the bisector of this angle, which allows us to consider only the first *half* of the angular sector. Third, for a given α, the straightness of the route is not affected by the considered spire or pseudo-spire, due to the *homothetic* properties of the networks. Put differently, in Figure 6.6 the Straightness is the same whether we consider the route between the center O and p_1, p_2, or p_3. This allows us to focus on the spire or pseudo-spire which is the closest to the center (i.e. the one containing p_1 in our example). Theorem 6.2.1 provides the closed form of the Straightness in both types of graphs. The detail of the proofs is available in [VL35].

Theorem 6.2.1 (Center-to-Periphery Straightness) *Let O the center of the considered graph, p the destination of the considered route, and $\alpha \in [0\,;2\pi]$ the route angle. We define α', the normalized version of α, as*

$$\alpha' = \frac{\theta}{2} - \left|(\alpha \bmod \theta) - \frac{\theta}{2}\right|. \tag{6.2}$$

*In a **rectilinear** network, the Straightness between O and p depends only on α, and not on the side c. For convenience, we note it $S(\alpha)$:*

$$S(\alpha) = \frac{1}{\cos\alpha' + \sin\alpha'} \tag{6.3}$$

*In a **radio-concentric** network, it depends on both α and the number of radii k. For convenience, we note it $S_k(\alpha)$:*

$$S_k(\alpha) = \frac{1}{\cos\alpha' + \frac{\sin\alpha'}{\tan\frac{\pi-\theta}{2}} + \frac{\sin(\alpha')}{\sin\frac{\pi-\theta}{2}}}. \tag{6.4}$$

Figure 6.7 represents the Straightness as the function of α in rectilinear networks, and in radio-concentric networks for several values of k. As explained before, the maximal value is 1. On the plot, it is obtained for multiples of θ, which correspond to cases where the destination is located on a radius. The minimal value is obtained for multiples of $\theta/2$, i.e. when the destination is located in the middle of a spire. This creates a periodic pattern, whose period depends on the number of radii k. The amplitude of the pattern is also affected by k: the more radii and the closer we get to the optimal Straightness. This is due to the fact that more radii means smaller detours. The limit case corresponds to an infinite number of radii, meaning that one can go from the center to any point using a straight line, therefore resulting in a constant Straightness of 1.

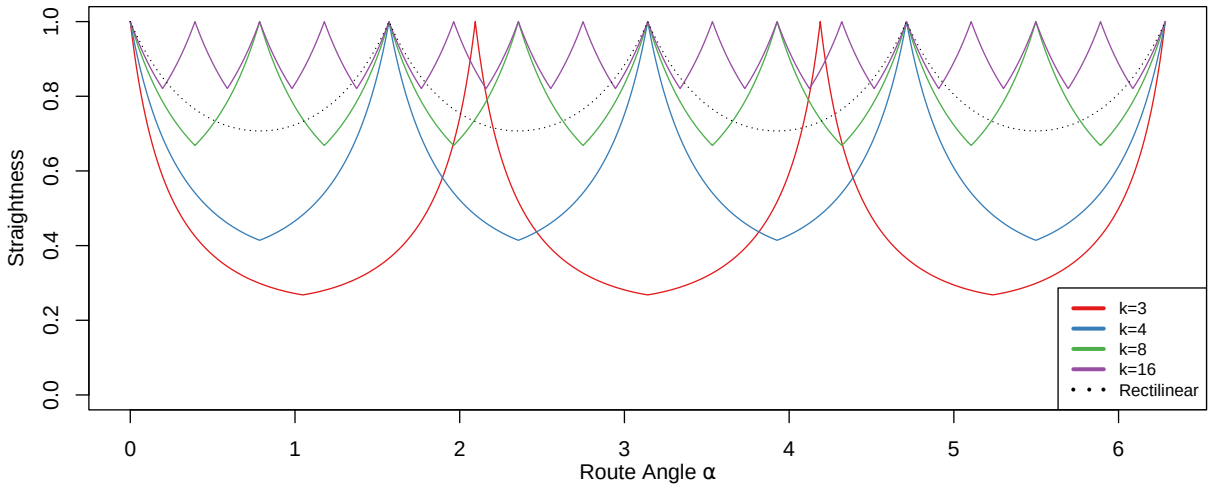

Figure 6.7: Center-to-periphery straightness in the rectilinear (dotted) and radio-concentric (solid) networks, as a function of the route angle α.

In order to compare the rectilinear and radio-concentric networks, we considered the expression $S(\alpha) - S_k(\alpha)$ and integrated it over $[0\,;2\pi]$. It turns out $k = 8$ radii are enough for the radio-concentric to reach a larger total Straightness than the rectilinear one, over all center-to-periphery routes. The number of radii in real-world orb-webs varies much depending on the spider species and habitat, but it is not uncommon for them to exceed our threshold by one order of magnitude, e.g. *Araneus diadematus*'s average 35 radii [213] or *Zilla diodia*'s more than 50 radii [241].

6.2.3 All Types of Routes

On the one hand, the results obtained analytically are robust, but on the other hand the method is difficult to generalize to all types of routes. Moreover, it relies on the geometrically regular nature of the considered networks, which does not hold when considering actual real-world structures, that can exhibit high levels of irregularities when compared to ideal models such as the radio-concentric or rectilinear networks. Instead, we dealt with this type of cases by turning to more empirical methods in order to approximate the topological measures of interest.

When considering all types of routes, it is not possible to adopt a parametric approach like we did for center-to-periphery routes with α, and to study the effect of this parameter on the Straightness. Thus, we adopted a different approach consisting in characterizing the network through the Straightness of a collection of (source,destination) pairs. As previously mentioned, in the literature authors traditionally use vertex-to-vertex Straightness to study spatial networks. Following this approach, we studied the Straightness of all pairs of vertices in the graph. However, as explained before, we wanted to consider all possible routes

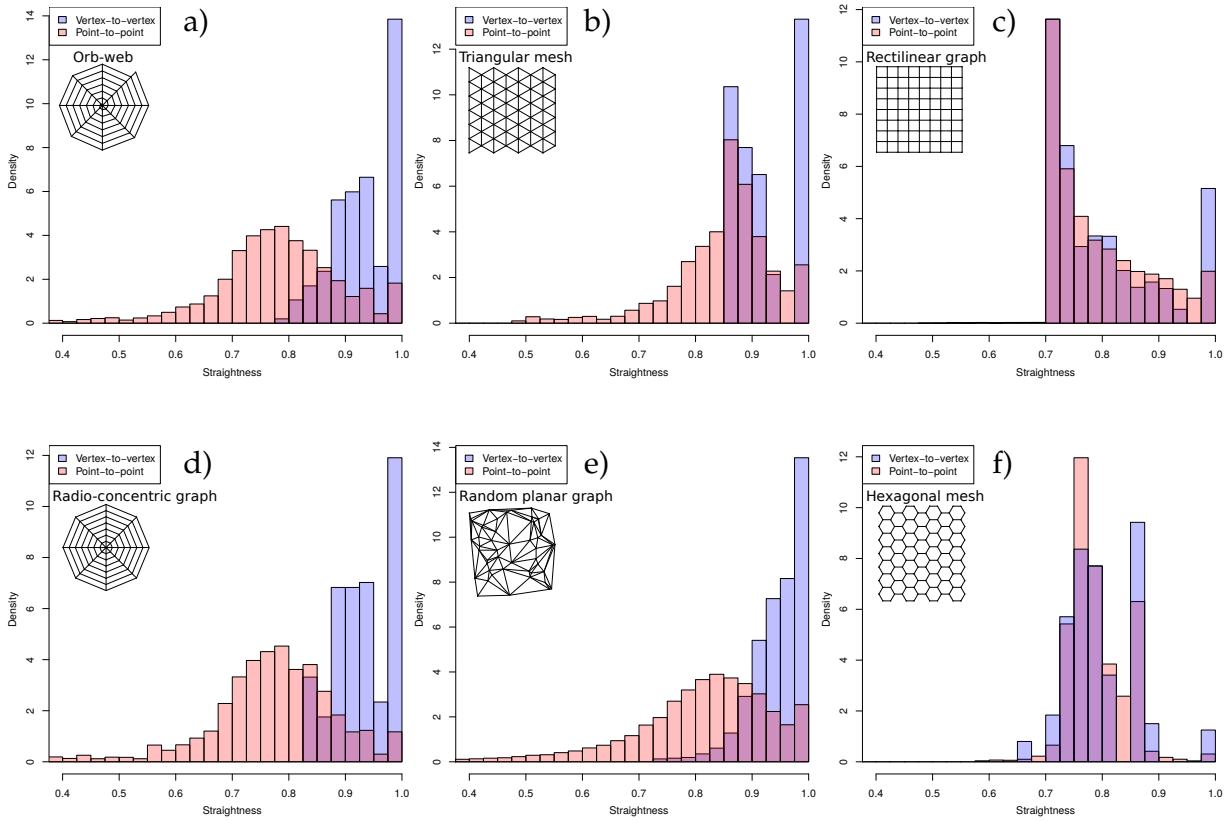

Figure 6.8: Distribution of the Straightness over each pair of vertices (blue) and points (red) in 6 types of spatial networks: orb-web (a); triangular mesh (b); rectilinear (c); radio-concentric (d); random planar (e); and hexagonal mesh (f).

on the network, including point-to-point routes. For this purpose, we proposed a method based on the discretization of the graph edges. First, additional 2-degree vertices are added on the graph edges, in a way such that no remaining edge is longer than some predefined parameter ε. This amounts to splitting the edges into shorter pieces, resulting in a modified graph $G' = (V', E')$ (with $V \subseteq V'$). Second, the Straightness is computed for each pair of vertices of this modified graph. Obviously, the smaller ε and the better the approximation. One problem with this approach is that the discretization process is likely to introduce a large number of additional vertices in the graph, resulting in a high computational cost (both in terms of time and memory). In order to alleviate some of that issue, we experimented with stochastic approaches consisting in sampling the set of vertices instead of considering them all.

Figure 6.8 shows the Straightness distributions obtained empirically for 6 types of networks. Blue bars correspond to the traditional vertex-to-vertex version, whereas red ones reflect the point-to-point version, approximated using our edge discretization process. It appears clearly that in most cases, both distributions are very different, with a point-to-point Straightness typically smaller than the vertex-to-vertex version. This backs up our assumption regarding the importance of considering point-to-point routes. The similarity between the distributions obtained for the orb-web (a) and the radio-concentric (d) graphs confirms that the latter is a good simplified model of the former. On the contrary, the rectilinear graph (c) and the hexagonal mesh (f) are very different from the orb-web, and this also holds for the octagonal mesh (not represented here for matters of space). Interestingly, both triangle-based networks (b and e) are somewhat in between these two types of distributions. The latter is relatively close to the orb-web. This random planar graph is obtained by drawing the vertex positions using uniform laws, and then connecting them using Delaunay's triangulation [79]. It can therefore be considered as a randomized version of the triangular mesh.

This Straightness-based typology constituted of 3 classes of networks also appears when considering the Straightness averaged over all pairs of vertices or points, as shown in Table 6.1. Moreover, we studied the evolution of the average Straightness as a function of the number of radii k in the orb-web and radio-concentric networks, and as a function of the side size c in rectilinear networks. In the former, the Straightness increases

Network type	Average Straightness		Average Betweenness	
	V2v	P2p	V2v	P2p
Orb-web	0.94	0.78	0.08	0.04
Radio-concentric graph	0.93	0.77	0.08	0.03
Random planar graph	0.92	0.85	0.06	0.03
Triangular mesh	0.92	0.85	0.06	0.03
Rectilinear graph	0.80	0.79	0.04	0.02
Octagonal mesh	0.80	0.79	0.05	0.04
Hexagonal mesh	0.80	0.79	0.05	0.04

Table 6.1: Average vertex-to-vertex (V2v) and point-to-point (P2p) Straightness and spatial betweenness for seven types of spatial graphs. See Figure 6.8 for some visual examples of these graphs.

with k, whereas in the latter, it stays relatively constant and thus does not depend on c. We found that orb-web and radio-concentric networks overcome rectilinear ones for $k \geq 8$, in terms of average Straightness. This confirms our theoretical result obtained for center-to-periphery routes (Section 6.2.2).

We adopted the same approaches to compute the spatial betweenness and study its distribution over the whole graph. Table 6.1 displays the average values obtained with both vertex-to-vertex and point-to-point routes. To get comparable results, we used a version of the betweenness that is normalized using the theoretical upper bound, which explains the very small values. Small differences are consequently meaningful. Briefly, the observed betweenness values also match the 3-class typology described before.

6.3 Continuous Average Straightness

The Straightness and its reciprocal are used in various ways in the literature. Besides the description of a given pair of vertices, they are also averaged to characterize a single vertex [194], a subgraph [135] or the whole graph [108, 135, 160, 229], as we did in Section 6.2.3. For a single vertex, authors consider the mean value for all pairs containing the vertex of interest. For a (sub)graph, they average over all pairs of vertices in the (sub)graph. Alternatively, on large graphs, authors average only over a subset of the vertex pairs [123, 156], primarily to limit the computational cost, but also to focus on certain parts of the graphs (e.g. daily home-office journeys in [156]).

As I explained before, the relevance of such a *discrete* average implicitly relies on the assumption that the only considered routes go from one *vertex* to another. However, there are situations such as ours, in which one wants to consider routes that can start or end anywhere on an edge, and not necessarily exactly on a vertex. In this case, averaging over pairs of vertices is likely to constitute a very rough approximation of the actual mean Straightness, as shown empirically in Section 6.2.3. One solution to solve this issue is to compute an approximate average point-to-point Straightness through the discretization process described in the same section. However, this approach can be computationally costly, and it is not clear how good of an approximation it is.

In this section, I summarize the method that I presented in [VL8] as a way to solve this issue and compute the exact point-to-point average Straightness. I adopted a continuous approach, by considering all the points constituting the graph instead of just its vertices. Instead of processing the average Straightness through a sum over pairs of vertices, I integrated the measure over the concerned edges. In other words, I proposed a different way of averaging the Straightness, while keeping the same definition of the Straightness itself. The continuous nature of my approach leads to a significantly lower computational cost, compared to a discrete approximation. My method allows the classic uses of the average Straightness, as a vertex accessibility measure or to characterize a (sub)graph, and I additionally derived several accessibility measure for edges and vertices. In Section 6.3.1, I describe the principle used to derive a closed expression of the continuous average Straightness, and define the proposed measures. The proof is too verbose for this manuscript, so it is omitted in favor of a simpler sketch of the principle applied to derive the measures. The interested reader

will find the complete proof in [VL8]. In Section 6.3.2, I present and discuss the main results obtained by computing these measures on artificial graphs and real-world road networks.

6.3.1 Proposed Measures

We define the set of the *graph points* P as the set of all points constituting the graph. It includes the vertices ($V \subset P$) as well as all the points lying on the edges. Each *point* $p \in P$ has a spatial position (x_p, y_p). As mentioned before, the approach that I proposed consists in averaging a spatial measure through integration over edges instead of summation over vertices. I perform this integration by considering a point p moving along the considered edge. For this purpose, I need to express the position of the point not in terms of its *absolute coordinates* (x_p, y_p), but rather with respect to its edge, using the notion of *relative position*, illustrated in Figure 6.9.a.

Definition 6.3.1 (Relative position) *The **position** of a point $p \in P$ **relatively** to its edge $(u, v) \in E$ is*

$$\ell_p = d_E(p, u), \tag{6.5}$$

i.e. its Euclidean distance to the first end-vertex of its edge.

The general idea is then to reformulate the spatial measure as a function of the relative positions of the concerned points, which allows integrating the measure over the edges containing these points. This generic approach could be applied to any measure, provided its reformulation is integrable. The Straightness is the ratio of the Euclidean to the graph distances between two points (cf. Definition 6.2.1), so in our case we need to reformulate both distances in terms of relative positions. For the Euclidean distance, this task is quite straightforward. For the graph distance though, the computation is more convoluted and requires leveraging the notions of *break-even point* and *break-even distance*.

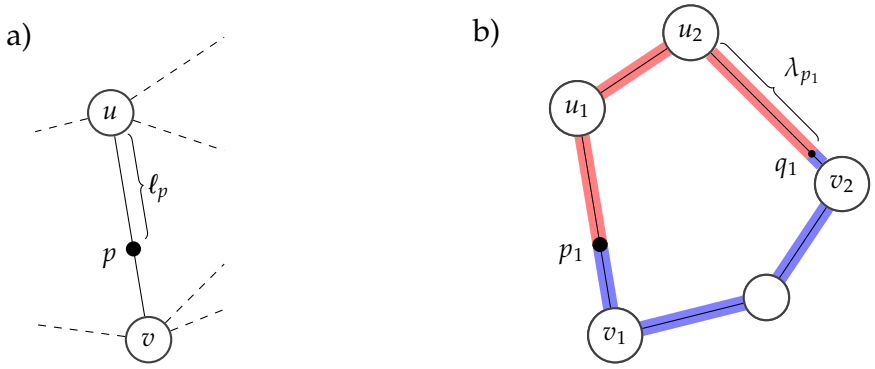

Figure 6.9: Representation of the relative position ℓ_p of a point p (a), and of the break-even point q and break-even distance λ_p of an edge (v_3, v_4) for a point p (b). By construction, the blue and red paths have the same length.

Definition 6.3.2 (Break-even point and break-even distance) *Consider a point $p \in P$ and an edge $(u, v) \in E$, such that p is either not lying on (u, v), or is one of its end-vertices u or v. The **break-even point** of (u, v) for p is the point q lying on (u, v) at an Euclidean distance λ_p from u, and such that*

$$d_G(p, u) + \lambda_p = d_G(p, v) + d_E(u, v) - \lambda_p. \tag{6.6}$$

*We call λ_p, which corresponds to the position of q relatively to its edge (i.e. $\lambda_p = \ell_q$), the **break-even distance**. Note that it is possible for the break-even point to be u or v (i.e. one of the end-vertices of the edge).*

Figure 6.9.b illustrates the notions of break-even point and break-even distance. By definition, the length of the shortest path between p_1 and the break-even point q_1 is the same whether we consider the paths going through u_2 (in red) or v_2 (in blue). The relative position ℓ_{p_1} should not be confused with the break-even

distance λ_{p_1}: the former is expressed relatively to (u_1, v_1), the edge containing the considered point p_1, whereas for the latter it is necessarily a distinct edge, here (u_2, v_2). Note that many measures are based on the notion of graph distance, for instance the closeness centrality is the reciprocal of the average graph distance between a vertex of interest and the other vertices in the graph [22]. Their reformulation would therefore also require using the notions of break-even point and distances.

Suppose that we want to compute the graph distance between p_1 and a point p_2 moving on edge (u_2, v_2) in Figure 6.9.b. The break-even point q_1 allows distinguishing two cases: if p_2 is located between u_2 and q_1, then the shortest path between p_1 and p_2 goes through u_2; whereas if p_2 is between q_1 and v_2, then this path goes through v_2. The same observation holds the other way round, to determine whether this shortest path goes through u_1 or v_1. However, as p_2 is mobile, we cannot use the corresponding break-even point, and have to leverage those of u_2 and v_2 instead (which, as the edge end-vertices, are fixed). We thus have three break-even points to consider, as represented in different colors (red, green, blue) in Figure 6.10. This amounts to a total of $2^3 = 8$ different cases to consider in general, corresponding to the height plots in the figure.

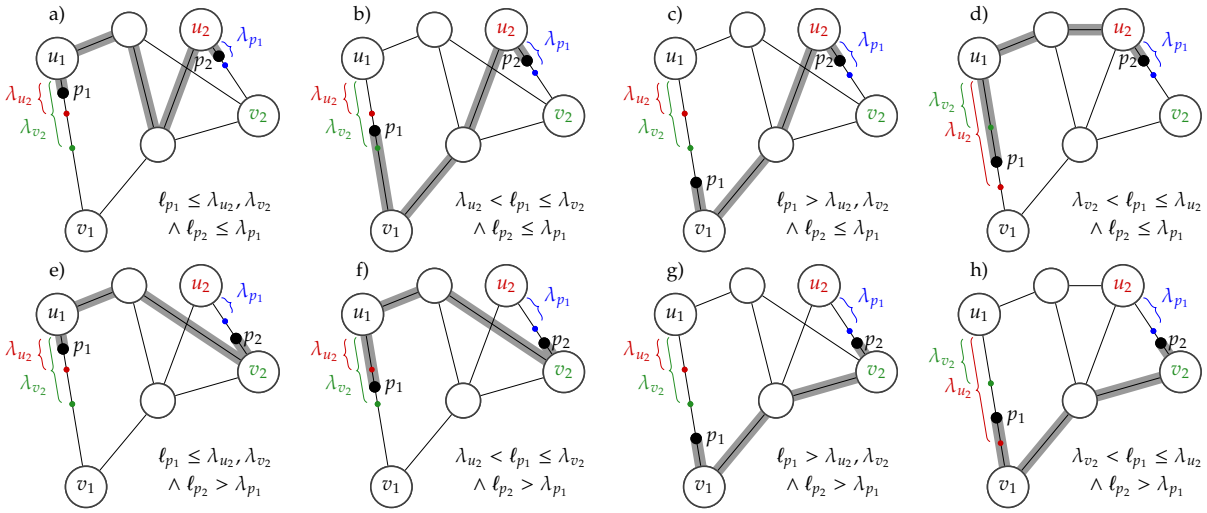

Figure 6.10: The height different possible situations of graph distance between two points p_1 and p_2. The shortest path is represented in gray.

Each case in Figure 6.10 corresponds to a slightly different expression of $d_G(p_1, p_2)$. Consequently, the expression of the Straightness $S(p_1, p_2)$ as a function of the relative positions of p_1 and p_2 also takes height different forms corresponding to these height cases[12]. For convenience, we note $S(\ell_1, \ell_2)$ this reformulation. We leveraged these expressions to define five variants of the continuous average Straightness, which I describe in the following. I start with the measures expressed relatively to a point of interest. First, the average Straightness between a point and an edge (or, more precisely: all the points lying on this edge) is obtained by integrating the Straightness over the edge of interest, and then normalizing using the edge length.

Definition 6.3.3 (Average Straightness between a point and an edge) *Let $\hat{S}_{u_2v_2}(p_1)$ be the **total** Straightness between, on one side, a point $p_1 \in P$ located at relative position ℓ_{p_1} on edge $(u_1, v_1) \in E$, and the other side all the points constituting an edge $(u_2, v_2) \in E$:*

$$\hat{S}_{u_2v_2}(p_1) = \int_0^{d_E(u_2,v_2)} S(\ell_{p_1}, \ell_{p_2}) \mathrm{d}\ell_{p_2}. \tag{6.7}$$

*The continuous **average** Straightness $S_{u_2v_2}(p_1)$ between p_1 and (u_2, v_2) is then*

$$S_{u_2v_2}(p_1) = \frac{\hat{S}_{u_2v_2}(p_1)}{d_E(u_2, v_2)}. \tag{6.8}$$

[12] See [VL8] for the exact expressions.

This measure $S_{u_2v_2}(p_1)$ is convenient to characterize the accessibility of an edge from a given point of the graph. From this result, it is straightforward to get the average Straightness between a point and the rest of the graph (i.e. all its edges), simply by summing over all the graph edges.

Definition 6.3.4 (Average Straightness between a point and the graph) *The continuous average Straightness between, on one side, a point $p_1 \in P$ located at relative position ℓ_{p_1} on edge $(u_1, v_1) \in E$, and the other side all the points constituting the graph G is*

$$S_G(p_1) = \frac{\sum_{(u_2,v_2)\in E} \hat{S}_{u_2v_2}(p_1)}{\sum_{(u_2,v_2)\in E} d_E(u_2, v_2)}. \tag{6.9}$$

This measure $S_G(p_1)$ can be considered as a global point centrality measure, representing how accessible the considered point is from the rest of a the graph.

I now focus on measures expressed relatively to an *edge* of interest, instead of a point. The average Straightness between two edges (or rather, between all their constituting points), requires integrating twice: once over each one of both considered edges.

Definition 6.3.5 (Average Straightness between two edges) *Let $\hat{S}_{u_2v_2}(u_1, v_1)$ be the **total** Straightness between, on one side, all the points constituting an edge $(u_1, v_1) \in E$, and on the other side all the points constituting an edge $(u_2, v_2) \in E$:*

$$\hat{S}_{u_2v_2}(u_1, v_1) = \int_0^{d_E(u_1,v_1)} \hat{S}_{u_2v_2}(p_1)\mathrm{d}\ell_{p_1}. \tag{6.10}$$

*The continuous **average** Straightness $S_{u_2v_2}(u_1, v_1)$ between (u_1, v_1) and (u_2, v_2) is then*

$$S_{u_2v_2}(u_1, v_1) = \begin{cases} 1, & \text{if } (u_1, v_1) = (u_2, v_2), \\ \dfrac{\hat{S}_{u_2v_2}(u_1, v_1)}{d_E(u_1, v_1) d_E(u_2, v_2)}, & \text{otherwise.} \end{cases} \tag{6.11}$$

As $\hat{S}_{u_2v_2}(u_1, v_1)$ is a double integration of the point-to-point Straightness over both considered edges, $S_{u_2v_2}(u_1, v_1)$ is symmetric with respect to these edges, i.e. $S_{u_2v_2}(u_1, v_1) = S_{u_1v_1}(u_2, v_2)$. This measure can be used to characterize the mutual accessibility of the considered edges. Based on this result, we can get the average Straightness between an edge and the rest of the graph (i.e. all its edges), by simply summing over all pairs of edges in graph.

Definition 6.3.6 (Average Straightness between an edge and the graph) *The continuous average Straightness between an edge $(u_1, v_1) \in E$ and all the points constituting the graph G is*

$$S_G(u_1, v_1) = \frac{\sum_{(u_2,v_2)\in E} \hat{S}_{u_2v_2}(u_1, v_1)}{\sum_{(u_2,v_2)\in E} d_E(u_2, v_2) d_E(u_1, v_1) - d_E(u_1, v_1)^2/2}. \tag{6.12}$$

$S_G(u_1, v_1)$ can be used as a centrality measure, but this time to describe edges in place of points as with $S_G(p_1)$. The value $S_G(u_1, v_1)$ represents the accessibility of the edge from the points constituting the whole graph.

Finally, by considering all pairs of edges in the graph, we get the average Straightness between all pairs of points constituting a graph, which can be used to characterize the whole graph at once.

Definition 6.3.7 (Average Straightness for the whole graph) *The continuous average Straightness between all pairs of points constituting the graph G is*

$$S_G(G) = \frac{\sum_{(u_1,v_1)\in E} \sum_{(u_2,v_2)\geq(u_1,v_1)} \hat{S}_{u_2v_2}(u_1,v_1)}{\sum_{(u_1,v_1)\in E} \sum_{(u_2,v_2)\geq(u_1,v_1)} d_E(u_2,v_2)d_E(u_1,v_1) - \sum_{(u_1,v_1)\in E} d_E(u_1,v_1)^2/2}, \tag{6.13}$$

where $\geq$ corresponds to the lexicographical order between edges.

In summary, I introduced 5 distinct continuous average Straightness measures, allowing to characterize various elements of a spatial graph (vertices, edges and the whole graph). As a reminder, they are described in Table 6.2 (top 5 rows), with their respective algorithmic complexity. The method used in Section 6.2.3 to approximate the point-to-point Straightness through edge discretization can be considered as a discrete approximation of our continuous average Straightness, and therefore constitutes a relevant baseline. In the following, instead of using ε (discretization step) to characterize the discretization, I prefer to use the *average edge segmentation*, noted β, which corresponds to the average number of times the original edges are split. For instance, a value of $\beta = 4$ means that an edge of the original graph G is split in 4 segments, in average, in the modified graph G'. We note $\sigma_\beta(u)$ and $\sigma_\beta(G)$ the discrete approximations of $S_G(u)$ and $S_G(G)$ obtained with an *average edge segmentation* of β, respectively. Table 6.2 includes both measures and their complexity (bottom 2 rows). The interested reader will find the detail of the complexity calculation and comparison in [VL8]. In summary, the asymptotic complexities of the continuous average measures $S_G(p)$ and $S_G(G)$ are always lower (or equivalent) than those of their discrete approximations $\sigma_\beta(u)$ and $\sigma_\beta(G)$, be it in terms of processing time or memory usage.

Measure	Description	Complexity Time	Space
$S_{uv}(p)$	Continuous average Straightness between a point p and an edge (u,v)	$O(n \log I)$	$O(I)$
$S_G(p)$	Continuous average Straightness between a point p and the graph G	$O(I^2 \log I)$	$O(I)$
$S_{u_2v_2}(u_1,v_1)$	Continuous average Straightness between two edges (u_1,v_1) and (u_2,v_2)	$O(I \log I)$	$O(I)$
$S_G(u,v)$	Continuous average Straightness between an edge (u,v) and the graph G	$O(In^2 \log I)$	$O(I)$
$S_G(G)$	Continuous average Straightness over all pairs of points constituting the graph G	$O(I^3 \log I)$	$O(I)$
$\sigma_\beta(u)$	Discrete approximation of $S_G(p)$	$O(\beta^2 I^2 \log \beta I)$	$O(\beta I)$
$\sigma_\beta(G)$	Discrete approximation of $S_G(G)$	$O(\beta^3 I^3 \log \beta I)$	$O(\beta I)$

Table 6.2: Summary of the considered measures: the 5 variants of continuous average Straightness and the 2 discrete approximations. All expressions describe the complexity for *sparse* graphs. More details in [VL8].

6.3.2 Empirical Validation

I implemented all our variants of the continuous average Straightness, as well as both discrete approximations, using the R language, and based on the R version of the igraph library [67]. The interest of this library is that it offers an easy access to graph-related operations through its R interface, while providing fast computations thanks to its underlying C implementation. Note that my source code is publicly available online [S9].

I first empirically evaluated the performance of my continuous approach in terms of computational time and memory usage, and compared it to its discrete approximations. This study was conducted on artificial graphs generated using the same process as the random planar graphs from Section 6.2.3, for various values of the average edge segmentation β, and with 10 repetitions for reliability. I focused only on $S_G(u)$ and $S_G(G)$

and their respective discrete approximations $\sigma_\beta(u)$ and $\sigma_\beta(G)$. Here, I only present and discuss $S_G(G)$ and $\sigma_\beta(G)$, as the results obtained for the other measures are qualitatively similar.

Figure 6.11.a shows the difference between the discrete approximation and the continuous average Straightness, i.e. $\sigma_\beta(G) - S_G(G)$, for each graph in the dataset, as a function of the average edge segmentation β. The horizontal dashed line materializes a zero difference. The x axis uses a logarithmic scale, which is also the case for the other plots presented in this section. My results confirm one of our observations from Section 6.2.2: the basic vertex-to-vertex average Straightness ($\beta = 0$) is a poor approximation of the point-to-point average Straightness ($S_G(G)$), as it is generally much higher. The approximation gets better when we start discretizing the edges: the values quickly decrease. However, they drop to the point where they underestimate the continuous average. Then, they get asymptotically closer to it when the average edge segmentation increases, i.e. when the edges are split into smaller and smaller pieces. The discrete approximation seems to become fairly close to the continuous average when the edges are split in 40–50 pieces or more. However, it seems hazardous to generalize this threshold to other cases.

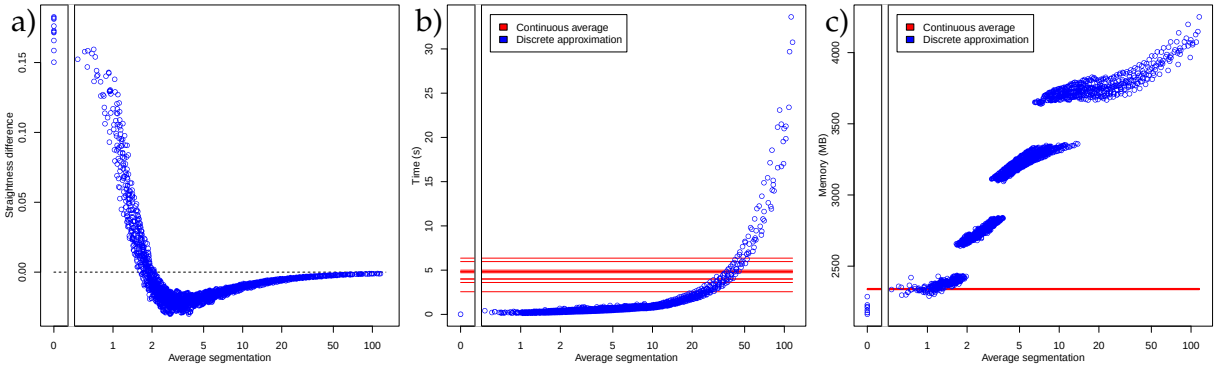

Figure 6.11: Results obtained on random planar graphs, as functions of the average edge segmentation β. (a) Difference between the discrete approximation and the continuous Straightness ($\sigma_\beta(G) - S_G(G)$). (b) Processing time and (c) memory usage required by the continuous average Straightness $S_G(G)$ (red) and its discrete approximation $\sigma_\beta(G)$ (blue).

Figure 6.11.b and 6.11.c respectively display the computational time and memory usage measured when processing the results shown in Figure 6.11.a. The blue dots are the discrete approximations, whereas the continuous average Straightness, which does not depend on β, is represented with red straight lines. Computing time is expressed in seconds and memory usage in MB. Both were obtained using a plain desktop Intel i5 3Ghz 8GB machine running Ubuntu 16.10. Both processing time and memory usage increase with β, and exceed the computational cost of the continuous average Straightness before the previously discussed threshold of $\beta \approx 40$–50. We assume that the three discontinuities present in the memory usage plot were caused by the garbage collector of the R environment, which is automatically triggered.

In [VL8], I provide a detailed analysis of the behavior of the average continuous Straightness, which I summarize here. Figure 6.12 displays the standard (vertex-to-vertex) average Straightness $\sigma_0(v)$ (a), as well as the continuous average Straightness computed for each vertex $S_G(v)$ (b), and for each edge $S_G(u, v)$ (c). One important observation is that the Straightness of an edge is affected by its length: a long edge is more likely to get a low Straightness (and vice versa), since reaching its most inner points requires a longer detour. This is illustrated, for instance, by edge (v_9, v_{12}) in Figure 6.12.c. This is not to say that only long edges have low Straightness though, as illustrated by edge (v_{17}, v_{18}) in the same graph, nor that all long edges necessarily have low Straightness. Another observation is that the Straightness of and edge and its attached vertices are not necessarily similar. For instance, on the one hand, in Figure 6.12.c, we can see that the continuous average Straightness between edge (v_9, v_{12}) and the rest of the graph is under 0.7. On the other hand, in Figure 6.12.b, the values observed for $S_G(v_9)$ and $S_G(v_{12})$ are clearly higher, reaching approximately 0.9.

When comparing the average continuous Straightness (Figure 6.12.b) to the standard vertex-to-vertex version (Figure 6.12.a), we observe that the values obtained with the latter are generally larger, which is consistent with our previous results. However, the continuous average is not just a monotonic function of its discrete counterpart. For instance, in the same figures, $\sigma(v_{15}) > \sigma(v_3)$, but $S_G(v_{15}) < S_G(v_3)$. When comparing more

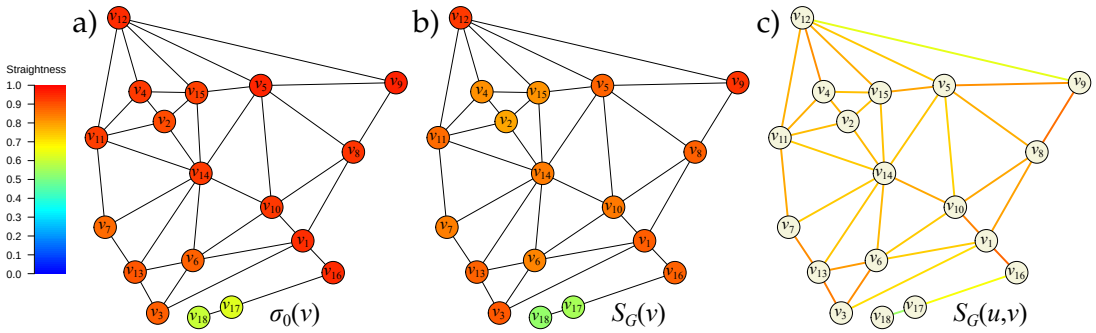

Figure 6.12: Straightness obtained for a random spatial graph: average vertex-to-vertex Straightness $\sigma_0(v)$ (a); average continuous Straightness for each vertex $S_G(v)$ (b) and edge $S_G(u,v)$ (c).

systematically the ordering of the vertices based on both measures, it appears that they largely disagree. The nature of this disagreement depends on the nature of the network, for instance it concerns the most extreme vertices in the case of the radio-concentric graph, whereas the measures agree on these vertices in the case of the random graph presented in Figure 6.12.

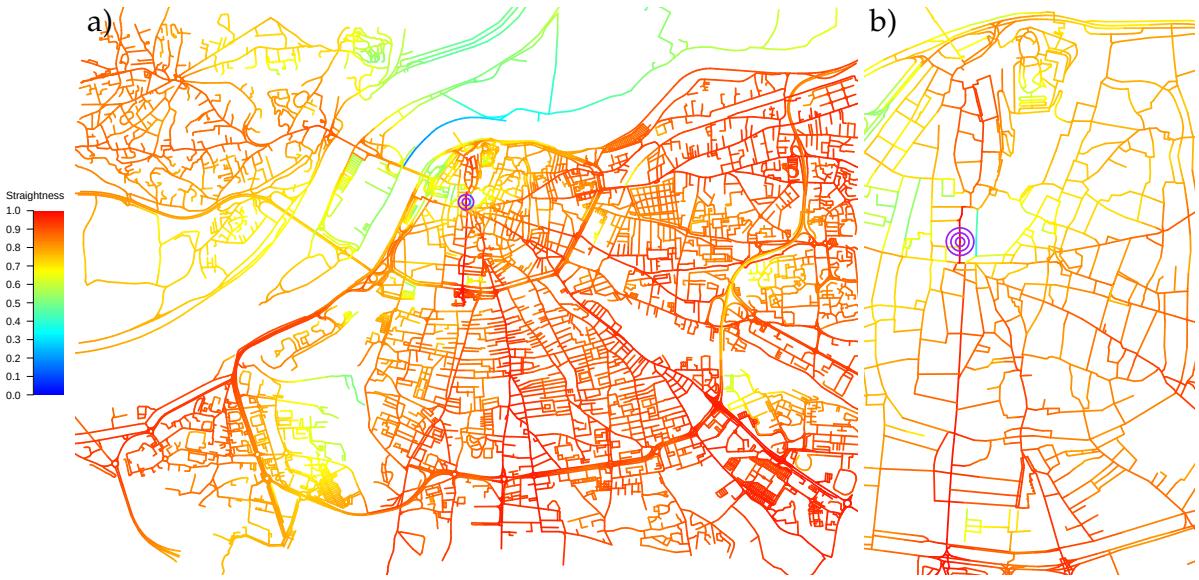

Figure 6.13: Straightness obtained for the city of Avignon, France: whole city (a) and city center (b). The color gradient corresponds to $S_{uv}(r)$, the average Straightness between a fixed vertex corresponding to the city hall (represented in purple) and each edge of the graph. Vertices are not represented to ease distinguishing the edges color.

Besides geometric and random graphs, I also used real-world road networks to assess the continuous average Straightness, based on data retrieved from the OpenStreetMap website[13]. As an example, Figure 6.13 displays a part of the city of Avignon, France. It represents $S_{uv}(r)$, the continuous average Straightness between a vertex of interest r corresponds to the city hall (in purple in the figure) and each edge of the graph. The color distribution shows that most of the city can be accessed efficiently when starting from this point, due to its radial structure. However, there are exceptions, which can be explained. There are two small yellow patches in the eastern part. Both are located near the city bypass, but not directly connected to it, which explains their low accessibility (Straightness). Moreover, the bottom one neighbors a seemingly empty zone, which is actually a freight train station. The fact that there is no road crossing the railroad in this zone also explains the observed low Straightness. The larger yellow patch on the southwestern part is not very well connected due to a similar reason, except this time the empty zone corresponds to the hight-speed train station (TGV). The lower accessibility in this area is also caused by the presence of an industrial activity sector, a water treatment plant and a recycling plant. On the western side, the accessibility is also globally

[13] https://www.openstreetmap.org

lower, because the Rhône river splits the urban area in two along a southwest to northeast axis. The blue parts correspond to a large, non-urbanized river island, which is only lightly connected to the road network. Finally, there are also traces of low Straightness in the city center, very near to the town hall. This is due to the distance effect that influence all forms of Straightness properties: on close destinations, it is more likely for the graph shortest path to be much longer than the distance as the crow flies, much more so than for distant destinations. This is particularly true for the center of Avignon, which is a medieval area with very convoluted streets, as illustrated by the right-hand plot of Figure 6.13.

6.4 Conclusion and Perspectives

In this chapter, I described my work on spatial graphs in the context of the MoMIS project. It consisted in studying the traditional Straightness of a number of spatial graphs, both theoretically and empirically. In particular, we compared orb-web and radio-concentric networks to certain geometrical and random networks. I also described the method which I proposed to perform an improved averaging of this measure, which leads to better results both computationally and qualitatively speaking. Based on this principle, I proposed five distinct measures, two of which have no match in previously existing measures. An immediate extension of the latter work is to apply the same principe of spatial integration to compute the continuous average of other spatial measures based on the concept of distance.

The project MoMIS itself is currently in standby as it did not lead to a more important funding, and consequently lost a few members. An interesting direction that I have started to explore on my own consists in approaching the problem the other way round: instead of comparing various graphs (orb-web, rectilinear, etc.) and checking their optimality according to various criteria such as their Straightness as we did, I started to design a tool able to generate a spatial graph that would be optimal according to some criteria of interest (Figure 6.14.a). Due to the complexity of the problem, I adopted a heuristic approach, which in turn necessitates solving a number of interesting geometrical problems. It is inspired from force layout methods used in graph visualization, such as Fruchterman–Reingold [107]. Starting from a random planar graph similar to that of Figure 6.8.e, one moves the vertices in order to maximize average Straightness while respecting certain constraints such as a fixed total edge length or area coverage.

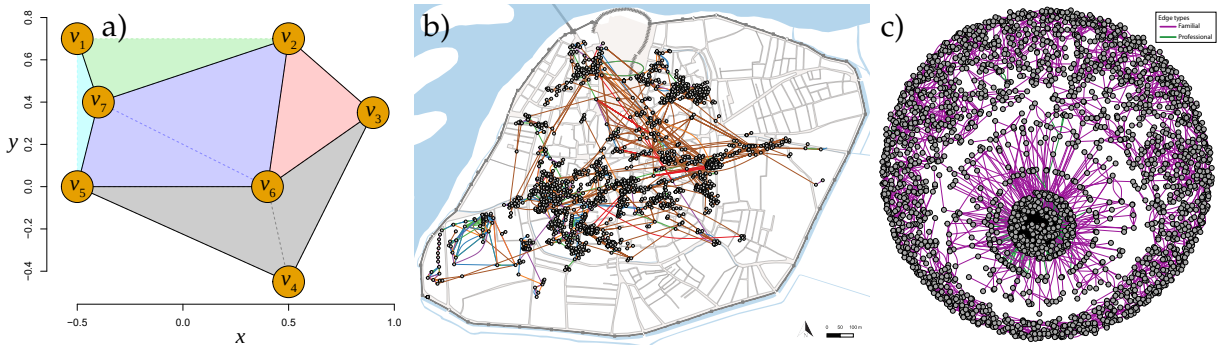

Figure 6.14: Ongoing projects related to spatial graphs: (a) One step of the method aiming at generating Straightness-wise optimal spatial networks; (b) Confront graph representing the city of Avignon, France, in medieval times; (c) Social graph of the persons owning real estate in the previous graph (pope in red).

In addition, I currently participate in an ongoing work related to spatial graphs, this time in the context of the interdisciplinary PhD of Margot Ferrand [100, S2], which I already mentioned in Section 2.4 as the studied networks also include vertex attributes. Among other objectives, this work aims at extracting a so-called *confront* network from a corpus of medieval deeds concerning the then papal city of Avignon. As illustrated by Figure 6.14.b, its vertices model pieces of real estate described in the deeds, and its edges represent their relative spatial positions as indicated in the deeds (e.g. *to the North*). The historical sources provide us with the position of a large number of pieces of real estate, so this network can be considered as spatial. One major problem here is to estimate the missing position of the rest of the real estate, based on the buildings whose position is already known, while respecting the confront constraints. An interesting research question is to

determine whether the confront network provides a good approximation of spatial distance, by comparing the Euclidean distance between pieces of real estate and their distance on this graph. As mentioned in Section 2.4, the deeds also contain information regarding the ownership and residency of the persons they mention. Therefore we can extract a spatial *social* network based on the confront network. We plan to leverage this graph to study the geographic area of influence of certain important characters, such as cardinals.

7

Partitioning of Signed Networks

7.1 Context

In a *signed* graph, each edge is labeled with a sign, either negative or positive, as described in Definition 7.1.1 and illustrated in Figure 7.1. This polarity models the nature of the relationship between the two vertices connected by the edge, and thus allows representing two types of opposed relationships in the same graph. In a way, this is similar to a two-layer multiplex network, as we have two types of relationships. However, in multiplex networks these types are independent, meaning two vertices can be connected according to several modalities. On the contrary, in signed graphs, both types are dependent, in the sense that if a positive edge exist between two vertices, there cannot be a negative one, and vice-versa. Early works actually consider graph that are both multiplex *and* signed [46]. Signed graphs can also be considered as a specific type of edge-attributed graphs, in which the edge attribute is the sign.

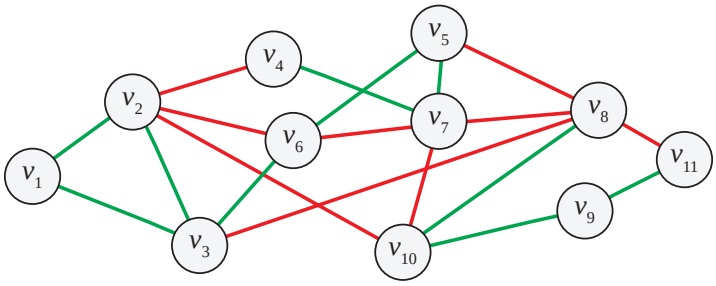

Figure 7.1: Example of random signed network containing 10 postive (in green) and 9 negative (in red) edges.

Definition 7.1.1 (Signed Network) *Let $G = (V, E, \sigma)$ a **signed network**, where $V = \{v_1, \ldots, v_I\}$ is the set of vertices, $E \subset V^2$ is the set of edges, and $\sigma : E \to \{-, +\}$ is the sign function associating a polarity to each edge. If the graph is undirected, we assume that edges are lexicographically ordered pairs. We note $E^+ \subset E$ and $E^- \subset E$ the subsets of positive and negative edges, respectively.*

This type of graphs was primarily introduced in the 40s by psychologist Heider [122], with the objective of describing sentiment relationships between people belonging to the same social group. More generally, a signed graph can be used to model any system containing two types of antithetical relationships, such as

like/dislike, for/against, similar/different, etc. After its introduction by Heider, the framework became very popular in the social sciences, and was used to model a number of social systems [46, 88, 175]. It then spread to a number of other domains, such as biology [70], finance [162], or bibliometrics [129].

When I arrived in Avignon in 2014, I started collaborating with Rosa Figueiredo, which was specialized in combinatorial optimization problems on graphs. We chose signed graphs, and more particularly their partitioning, as a common ground. Since then, I have devoted a large part of my research activity to this topic. This common work was the occasion to jointly advise several undergraduate [U6, U1, S1] and graduate [G3, G1, G9, G6] students, as well as two PhD students [D4, D1]. All this work resulted in a number of publications in national conferences [VL77, VL81, VL80, VL78, VL94, VL79, VL68], international conferences [VL29, VL60, VL36] and journals [VL3, VL1, VL4]. We also organized a symposium in 2016 (cf. Appendix A.6.1), which led to us editing a special issue in the *Journal of Interdisciplinary Methodologies and Issues in Science* [VL85] in 2017. Finally, we participated in the successful submission of the interdisciplinary ANR DeCoMaP project, but I defer its description until the manuscript conclusion (Chapter 9), as it has just started and is therefore more related to perspectives than past work. Our work was also the recipient of several smaller fundings, cf. Appendix A.6.2.

We mainly conducted two tasks, which I describe in this chapter. I first introduce the main notions related to signed graph partitioning in Section 7.2, including the original Correlation Clustering problem and its relaxed version. Then I focus on the first task, which consisted in studying the behavior of signed graph partitioning methods on real-world graphs (Section 7.3). The second task consists in characterizing and analyzing the space of optimal solutions of the Correlation Clustering problem (Section 7.4). I conclude the chapter with my ongoing work related to these topics, and its perspectives (Section 7.5).

7.2 Structural Balance and Related Notions

The concept of *structural balance* is central to the analysis of signed graphs right from the start, in the seminal work of Heider [122]. He discussed the stability of triads of connected vertices, arguing that certain states were harmonious and desirable by the modeled cognitive agents, whereas other states where imbalanced and tend to evolve towards harmonious ones, possibly by switching the sign of certain edges. As noted by Estrada [95], most subsequent works focused on this aspect of Heider's work, i.e. assumed that signed graphs tend towards balance. But he also stressed in the same article that some agents may look for instability, for various reasons (boredom, curiosity, etc.). Put differently, not all systems evolve towards a balanced state. This was later observed by various authors, as discussed by Doreian [85].

The triads identified as *balanced* by Heider contain either three positive edges (Figure 7.2.a) or one positive and two negative edges (Figure 7.2.b); whereas those deemed unbalanced contain either three negative edges (Figure 7.2.c) or one negative and two positive edges (Figure 7.2.d). This was later interpreted in the following way [199], by considering that positive and negative edges represent friendship and enmity, respectively, and based on the assumed transitivity of these relationships. From the perspective of vertex v_1 in Figure 7.2:

- *The friend of my friend is my friend* : in Figure 7.2.a, $\sigma(v_1, v_2) = +$ and $\sigma(v_2, v_3) = +$, so we must have $\sigma(v_1, v_3) = +$. This not the case in Figure 7.2.d.
- *The enemy of my enemy is my friend* : in Figure 7.2.a, $\sigma(v_1, v_2) = -$ and $\sigma(v_2, v_3) = -$, so we must have $\sigma(v_1, v_3) = +$. This is not the case in Figure 7.2.c.

Harary later formalized this concept by defining the sign of a triad as the product of its edge signs, and stating that a triad is balanced if it is positive [46, 119]. Moreover, he generalized this idea to larger cycles, which allowed him to express the notion of balanced signed graph as described in Definition 7.2.1. This specific definition was *a posteriori* termed *Strong* Structural Balance, in order to distinguish it from subsequent variants.

Definition 7.2.1 (Strong Structural Balance) *A signed network $G = (V, E, \sigma)$ is said to be **structurally balanced** in the **strong** sense if all of its cycles of any length are positive.*

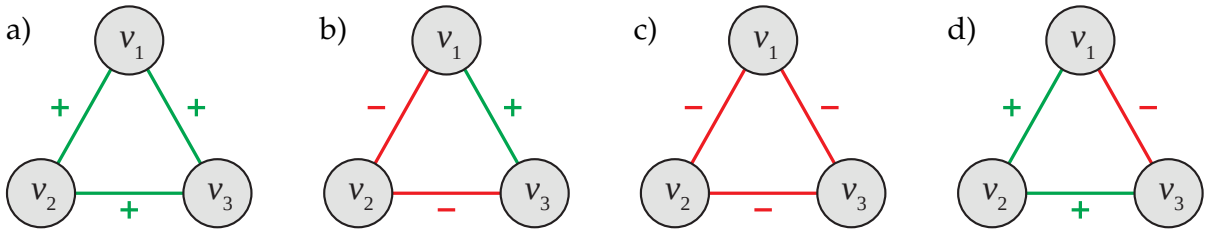

Figure 7.2: The four possible types of signed triads. The *strong* definition of structural balance considers only (a) and (b) as balanced, whereas the *weak* definition also includes (c).

Harary showed that a strongly balanced signed graph can be partitioned into two communities P_1 and P_2 (one of them possibly empty), such that all *positive* edges connect two vertices of the *same* community, and all *negative* edges connect two vertices from *different* community. In other words, each community is cohesive because it exhibits internal friendship, and both communities are antagonistic as their members are enemies. As an illustration, Figure 7.3.a and Figure 7.3.b show a strongly balanced and a strongly imbalanced graphs, respectively.

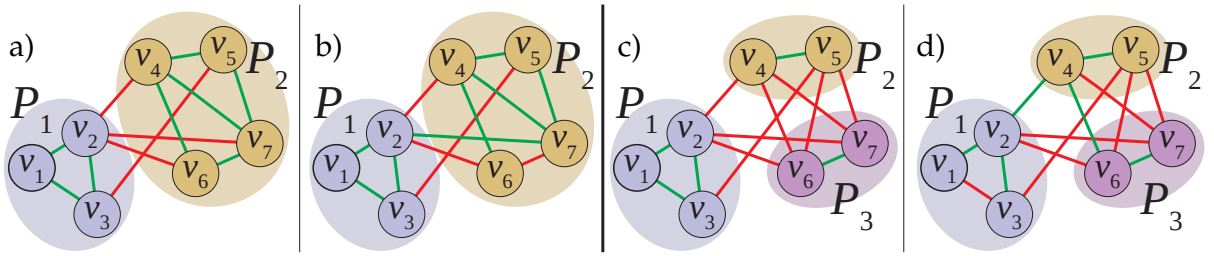

Figure 7.3: Examples of strongly and weakly balanced and imbalanced graphs: (a) strongly and weakly balanced ($I(G,\mathcal{P}) = 0$); (b) strongly and weakly imbalanced ($I(G,\mathcal{P}) = 2$); (c) strongly imbalanced but weakly balanced ($I(G,\mathcal{P}) = 0$); (d) strongly and weakly imbalanced ($I(G,\mathcal{P}) = 3$).

Davis [72] later noticed that certain social groups contain more that two hostile factions, as illustrated in Figure 7.3.c, and he generalized the notion of balance in order to handle this situation. The principle is the same: the graph is considered as balanced if one can partition V so that all *positive* edges lie *inside* the communities, and the *negative* ones are located *between* them. However, this conflicts with the fact that the all-negative triad (Figure 7.2.c) is considered as imbalanced, since each one of its vertices could now be in a different community (e.g. $\{v_2, v_4, v_6\}$ in Figure 7.3.c). For this reason, in the Weak Structural Balance, this triad is considered as balanced, resulting in Definition 7.2.2. Figure 7.3.c shows a weakly balanced graph, whereas the one in Figure 7.3.d is weakly imbalanced. The strongly balanced graph from Figure 7.3.a is also weakly balanced, and the strongly imbalanced one from Figure 7.3.b is also weakly imbalanced. However, the weakly balanced graph from Figure 7.3.b is *not* strongly balanced.

Definition 7.2.2 (Weak Structural Balance) *A signed network $G = (V, E, \sigma)$ is said to be **structurally balanced** in the **weak** sense if none of its cycles of any length contain exactly one negative edge.*

In practice, a signed network representing a real-world system is rarely perfectly balanced, though. That is why, instead of only focusing on the dichotomous problem *balanced* vs. *imbalanced*, it is generally more relevant to measure *how much* the network is imbalanced. Various measures have been proposed for this purpose. Here, I focus on the well-known *Imbalance* [86], which appears under various names in the literature, such as *Line index* [121] or *Frustration* [121]. The imbalance of a graph is based on the imbalance of the partition of this graph, as described in Definition 7.2.3.

Definition 7.2.3 (Imbalance Measure) *The **Imbalance** $I(G,\mathcal{P})$ of a partition $\mathcal{P} = \{P_1, \ldots, P_K\}$ for a signed graph $G = (V, E, \sigma)$ is the total number of **frustrated edges**, i.e. positive edges located between the communities*

and negative edges located inside the communities:

$$I(G,\mathcal{P}) = \sum_{1\leq k<k'\leq K} \Omega^+(P_k,P_{k'}) + \sum_{1\leq k\leq K} \Omega^-(P_k), \tag{7.1}$$

where $\Omega^+(P_k,P_{k'})$ denotes the number of positive edges between communities P_k and $P_{k'}$, and $\Omega^-(P_k)$ the number of negative edges inside community P_k.

If the graph is weighted, one would sum the edge weights instead of just counting them. In order to compute the imbalance of the graph $I(G)$, one needs to solve an optimization problem called *Correlation Clustering* (CC), as described in Definition 7.2.4. It was formalized by Bansal *et al.* [16], and appeared before that (under a different name) in [86]. Since both graphs from Figures 7.3.a and 7.3.c are weakly balanced, we get $I(G)$. However, the graph from Figure 7.3.b has two frustrated edges: one is positive ((v_2,v_7)) and one is negative ((v_6,v_7)), so its imbalance is $I(G)=2$. Similarly, the graph from Figure 7.3.d has three frustrated edges: two are positive ((v_2,v_4) and (v_4,v_6)) and one is negative ((v_1,v_3)), so its imbalance is $I(G)=3$.

Definition 7.2.4 (Correlation Clustering Problem (CC)) *Let $G=(V,E,\sigma)$ be a signed graph. The **Correlation Clustering** problem consists in finding a partition $\mathcal{P}$ of V minimizing the imbalance $I(G,\mathcal{P})$. We note $I(G)$ the resulting minimal value, which corresponds to the **graph imbalance**.*

Doreian and Mrvar [87] later observed that a social system can very well evolve to a state where certain members of several hostile factions start durably cooperating, or where members of the same faction definitively become enemies, arguing that the corresponding edges should not be considered as frustrated. Figure 7.4.a shows an example of such a situation including a mediator community P_3. For this purpose, they generalized the notion of weak structural balance to that of *relaxed* structural balance, through the formulation of the *Relaxed Imbalance* (RI) described in Definition 7.2.5. With the RI, the notion of frustration became something relative: an edge between two communities or inside a community is frustrated if it does not have the sign which is a majority among the other edges holding the same position. This sign can therefore be positive or negative depending on the context.

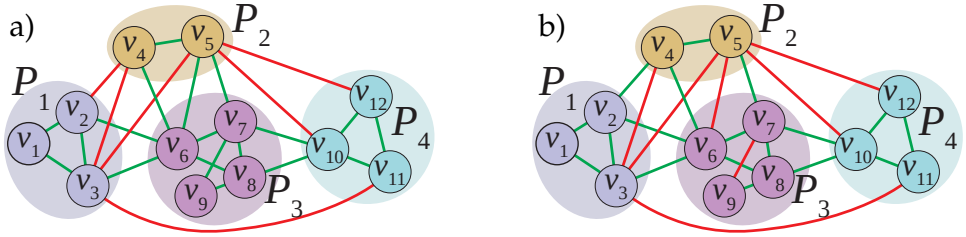

Figure 7.4: (a) relaxed balanced graph including a mediator community P_3. (b) relaxed imbalanced graph ($RI_4(G)=3$).

Definition 7.2.5 (Relaxed Imbalance Measure) *The **Relaxed Imbalance** $I(G,\mathcal{P})$ of a partition $\mathcal{P}=\{P_1,\dots,P_K\}$ for a signed graph $G=(V,E,\sigma)$ is the total number of **frustrated edges**, i.e. edges of the minority sign relative to the other edges located between two communities or inside one community:*

$$RI(G,\mathcal{P}) = \sum_{1\leq k<k'\leq K} \min\left(\Omega^-(P_k,P_{k'}),\Omega^+(P_k,P_{k'})\right) + \sum_{1\leq k\leq K} \min\left(\Omega^-(P_k),\Omega^+(P_k)\right). \tag{7.2}$$

The *Relaxed Correlation Clustering* problem (RCC) is defined similarly to the CC, as the minimization of the relaxed imbalance, with the difference that the desired number of communities K must be specified, as described in Definition 7.2.6. Both CC [16] and RCC [101] are NP-hard. Figure 7.4.a shows an example of relaxed balanced graph. The same partition would get an imbalance of $I(G,\mathcal{P})=7$, as all positive edges between the mediator community (purple) and the other community would be seen as frustrated. Solving CC on the same graph would lead to a partition where the community P_2 is merged with the mediators, with an imbalance of $I(G,\mathcal{P})=4$. Figure 7.4.b shows a relaxed imbalanced graph with $RI_4(G)=3$. There are one positive ((v_2,v_4)) and two negative ((v_5,v_6) and (v_7,v_9)) frustrated edges.

Definition 7.2.6 (Relaxed Correlation Clustering Problem (RCC)) *Let $G = (V, E, \sigma)$ be a signed graph and K an integer such as $1 \leq K \leq |V|$. The **Relaxed Correlation Clustering** problem consists in finding a K-partition $\mathcal{P}$ of V minimizing the relaxed imbalance $RI(G, \mathcal{P})$. We note $RI_K(G)$ the resulting minimal value, and it corresponds to the **relaxed graph imbalance**.*

7.3 Behavior of Signed Graph Partitioning Methods

The first task that we conducted was meant to explore the domain of signed graph partitioning problems, and the methods allowing to solve them. For this purpose, we performed a series of experiments using real-world data representing the voting activity of Members of the European Parliament (MEPs). We retrieved the raw data describing these votes from the citizen oversight website *It's Your Parliament*[14] (IYP). The IYP data describe the activity of the MEPs during the 7$^{\text{th}}$ term of the EP (2009–14). They are constituted of the votes cast by all MEPs for all roll-calls taking place during a plenary at the EP. Each roll-call is characterized by a policy domain, such as Agriculture, Budget, etc. Each MEP is characterized by their member state, national political party, and European parliamentary group. These groups are important when interpreting a partition of the MEP set, because they correspond to the political position that MEPs are supposed to hold, at least theoretically. One would expect that partitions automatically estimated based on the voting data would fit this ideological division, but as we will see, this is not necessarily the case. We extracted signed networks from the IYP data by computing an agreement score between each pair of MEPs, measuring how similarly they voted over a set of roll-calls. We experimented with various weighting schemes described in [VL36]. By using various subsets of the data based on member state, EP group, or roll-call policy domain, we were able to extract a collection of 4,150 weighted signed graphs. Our data [C7] and source code [S11] are both publicly available online.

We considered three points of interest, which I develop in the rest of this section. First, we explored whether taking negative edges into account affects the detected partition, in a real-world context (Section 7.3.1). Second, we checked whether filtering the edges with smaller weights affects the performance of partitioning methods, in weighted signed graphs (Section 7.3.2). Third, we compared the partitions identified when solving CC and RCC on the same graphs (Section 7.3.3).

7.3.1 Relevance of Negative Edges

Similarly to what I discussed in Chapter 2 regarding attributed graphs, obtaining the negative relationships can be much more difficult and costly than positive graphs, e.g. in the context of a ground survey, it is much easier to get people to name their friends than their foes. Moreover, using the negative edges during the graph analysis requires a tool specifically designed for this purpose, when many methods already exist to handle unsigned graphs. Some authors consequently questioned the relevance of using negative edges during graph partitioning. Esmailian *et al.* [93] observed on two social media datasets (Slashdot and Epinions) that, when applying an unsigned community detection method to a positive-only graph obtained by removing all negative edges, most negative edges of the original graph were falling where they were expected, i.e. between the detected communities. Moreover, they showed that the number of negative edges falling inside the communities was not significant. However, the method they used to assess this significance considered each community separately, instead of the graph as a whole; and only two datasets is not much to generalize their observation to all real-world signed networks.

As described in more details in [VL36, VL68], we consequently decided to test their assumption using the networks that we extracted from the IYP dataset. Our method was based on the comparison of graph partitions obtained through three different means:

1. Standard community detection on *positive* graphs;

[14] http://www.itsyourparliament.eu/

2. Standard community detection on the complements of *negative* graphs;
3. Solving correlation clustering on *signed* graphs.

The term *positive graph* refers to the unsigned graph obtained by removing the negative edges from a signed graph. The *complement of the negative graph* is obtained by first removing the positive edges from the signed graph, then taking the complementary of the resulting graph. This leads to an unsigned graph that encodes the information conveyed by the negative edges under a form compatible with community detection.

To partition the first two types of unsigned graphs, we used a selection of (then) state-of-the-art community detection tools: InfoMap [203], EdgeBetweenness [180], WalkTrap [193], and FastGreedy [59]. To handle the signed graph, we solved the CC problem using the *Parallel ILS algorithm* presented in [157], denoted pILS-CC here, which implements a metaheuristic approach and can handle large real-world networks. Moreover, this specific implementation is parallelized, in order to improve speed. Considering that certain of the extracted networks are very dense (especially the second type), we had to perform some minor modifications on the original algorithm, so that the processing time was acceptable.

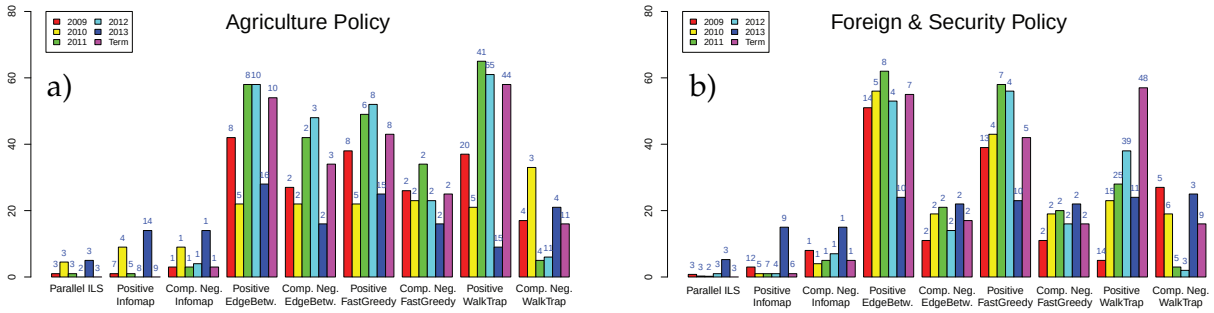

Figure 7.5: Imbalance of the partitions (bars) and numbers of detected communities (blue values), obtained through Parallel ILS (left bar group) and unsigned community detection methods (other bar groups), for each year and the whole term (see legend), processed for the *Agriculture* (a) and *Foreign & Security* (b) policy domains.

Figure 7.5 shows the Imbalance results obtained with each partitioning method, for two specific policy domains, by year and for the whole period. It also shows the number of communities detected by the methods. Note that the imbalance is expressed here in terms of *proportion* of frustrated edges. On the positive graphs, EdgeBetweenness, FastGreedy and WalkTrap get very low performance, finding a large number of communities, probably because they just rely on edge density. On the complement of the negative graph, they get much better results, although still far from both other methods. This improvement is probably caused by the fact these graphs are much denser, making it harder to find many communities. InfoMap algorithm is much better at detecting balanced partitions, reaching a much smaller imbalance on the positive graphs. On the complement of the negative graphs though, it just put all the vertices in a single community. Parallel ILS systematically obtains the lowest imbalance values. Moreover, it detects only two or three communities, which corresponds to what we were expecting *a priori* for this system. Indeed, we have no ground truth, but based on our knowledge of the data, we should get 1 community in case of unanimity, 2 if there is a clear For/Against opposition, and a very few more if some MEPs swing in-between during the considered period.

We investigated how close the partitions identified by InfoMap and Parallel ILS are, using the popular *Normalized Mutual Information* [217] (NMI) as a partition similarity measure. It turned out that they are independent, with a NMI close to zero. We concluded that, on these data, our results did not confirm the findings of Esmailian *et al.* [93] regarding the low informative value of negative edges. Taking negative edges into account leads to a lower imbalance and a different partition, containing larger communities.

7.3.2 Effect of Filtering

Based on our work from [VL36], we formulated the hypothesis that edges with small weights may constitute some form of noise, possibly making it harder both to partition the graphs, and to interpret the obtained

partitions. We tried to answer these questions empirically in [VL29], by comparing the partitions detected before and after filtering the lighter edges. We used two partitioning methods: the previously mentioned heuristic approach ILS-CC [157], and an exact method [6] that we call Ex-CC.

Figures 7.6.a and 7.6.b show the numbers of positive (in green) and negative (in red) edges for each network in the dataset, *before* and *after* filtering, respectively. On the x-axis, the networks are ordered by decreasing number of edges. The y-axis uses a logarithmic scale for the sake of readability. In average, our filtering removes 43% of the edges. However it is worth noticing that the general distribution does not change (if anything, it gets smoothed) and the proportions of positive and negative edges are also roughly preserved. This also holds when considering the weights instead of the numbers of edges (not plotted here). All the graphs are originally almost completely connected, and filtering splits only a minority of them in two (23%) or three (8%) components. But in these cases, we still keep a giant component representing 87% and 76% of the graph, respectively.

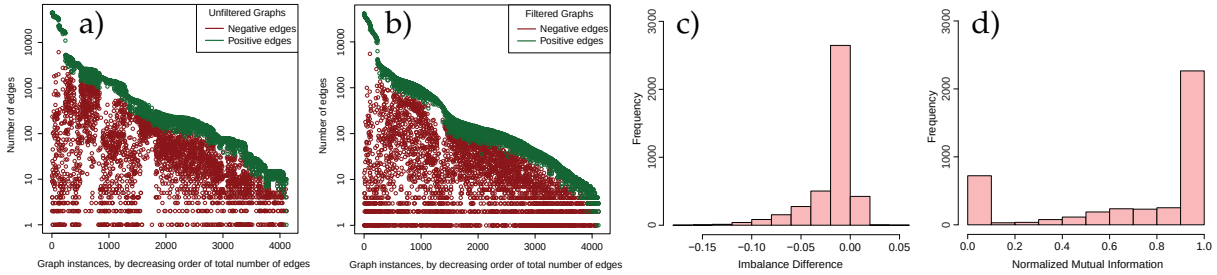

Figure 7.6: (a) Numbers of positive (green) and negative (red) edges in all unfiltered networks. (b) Same thing for filtered networks. (c) Distribution of the Imbalance difference between partitions detected by Ex-CC *with* vs. *without* filtering. (d) Distribution of the NMI when comparing the partitions detected by Ex-CC *with* vs. *without* filtering.

The filtering step tend to cause a decrease in partitioning processing time, even more so for graphs whose unfiltered partitioning requires more time. For Ex-CC, filtering cuts the processing time by 5–10% in those graphs, whereas for ILS-CC it is close to 90%. We focus only on Ex-CC to perform a reliable comparison of the detected partitions. For almost all instances, we observe an increase in the number of communities detected in the filtered graphs. Figure 7.6.c shows the distribution of NMI when assessing the similarity between the partitions obtained on the same graph before and after filtering. It is larger than 0.8 for 61% of the instances, meaning many partitions are not affected much by the filtering, despite the previously observed variation in the number of communities. However, for 20% of the instances, the NMI is lower than 0.1, which means the partitions are very different. Figure 7.6.d shows the distribution of the difference in Imbalance resulting from the filtering. Most of the instances fall on the negative side, but close to zero. This means the filtering does not affect much the quality of the partition.

In conclusion, on the one hand filtering the networks provides an improvement in terms of computational time, and on the other hand, it leads to partitions that are generally similar to those obtained without the filtering, but can occasionally be very different. The filtering step seems worthy only when dealing with large quantities of graphs as we did here, in which case the time saved may be a sufficient justification. The fact that filtering causes the partitioning methods to find more communities can be troublesome when dealing with networks containing few communities, as it may significantly increase their number, relatively speaking.

7.3.3 Comparison of CC and RCC

Another question that we studied empirically is how the partitions detected by CC and RCC differ in practice, when dealing with real-world networks, and how this affects the interpretation of these results relative to the studied system. In this manuscript, I focus on a subset of the data for matters of space and concision: the votes of the French MEPs regarding propositions related to agriculture in 2012–13. I selected the French MEPs because I know them better, which eases the interpretation, but we compare them to Italian MEPs in [VL29]. I selected this policy domain because it was potentially polarizing during this period that coincides with a

reform of the *Common Agricultural Policy* (CAP), a major budgetary item of the EU. We additionally discus roll-calls related to economy in [VL29].

Figure 7.7.a shows the similarity network obtained for the considered subset of roll-calls: vertices represent MEPs and edges represent filtered vote similarity values. The outer ring indicates the political groups, from left to right: GUE-NGL (communists, in red), G-EFA (environmentalists, in green), S&D (socialists, in pink), ALDE (liberals, in orange), EPP (conservative, in light blue), ECR (euroskeptiks, in dark blue), EFD (far-right, in purple) and NI (Non-Inscrits, i.e. far-right, in brown). Figures 7.7.c, 7.7.c and 7.7.d represent the partitions obtained with Ex-CC, Ex-RCC using $K = 3$ and Ex-RCC using $K = 4$, respectively. Each vertex represents a community, and each positive (resp. negative) edge corresponds to the set of all individual positive (resp. negative) edges between the MEPs constituting these communities. For each community, we indicate the groups of the MEPs constituting it, and its proportions of internal positive and negative edges, relatively to the total network weight. The community itself is represented by a pie chart also reflecting these proportions. Similar proportions are provided also for each edge. Finally, in both types of graphs, positive (resp. negative) edges are drawn in green (resp. red).

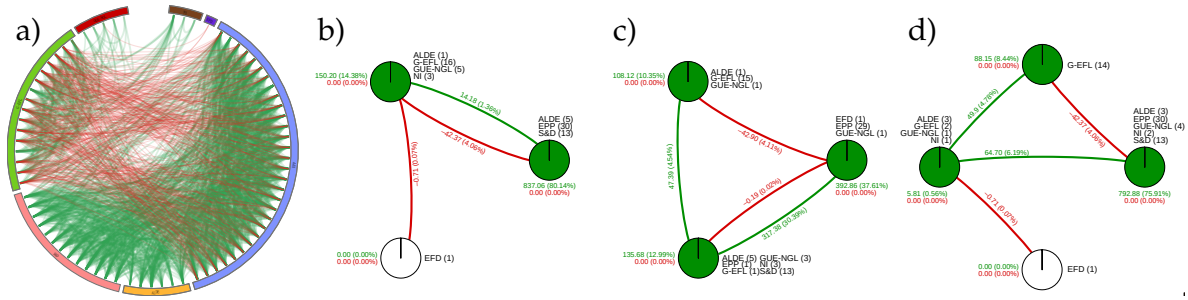

Figure 7.7: Voting similarity network between French MEPs for roll-calls related to agriculture during the year 2012-13 (a), and communities detected using Ex-CC (b) and Ex-RCC with $K = 3$ (c) and $K = 4$ (d). The outer ring colors in the similarity network match the EP political groups (see text).

The optimal CC solution contains 3 communities and has an imbalance of 14.18 (1.35% of the total weight). There are two large, negatively connected, communities of similar size: the first is largely dominated by the environmentalists (G-EFL) and also contains the radical left (GUE-NGL) and NI; whereas the second contains the center-left (S&D), right and center-right (EPP and ALDE) groups. Both communities have a majority of positive internal and negative external edges. The third community (at the bottom) contains a single vertex corresponding to an outlier: *Philippe de Villiers*, the only French member of the right-wing euroskeptic EFD group. This partition exhibits a clear left/right divide, with the exception of the three NI members, who are put together with the left/environmentalists. This divide can be explained when considering the texts voted this year, among which many concern animal rights and related matters. The fact tha one ALDE member was put with the Greens supports this, since it corresponds to *Corrine Lepage*, former Minister of the Environment in a right-wing French government. One could expect the S&D to vote similarly to the rest of the left on these topics. However, even more texts voted during this year concern the CAP, on which the center-left is more likely to side with the right. This also explains the position of NI, which is more likely to opportunistically support resolutions in favor of small family-owned farms, and therefore vote like the radical left.

The solution obtained with Ex-RCC for $K = 3$ gets a lower relaxed imbalance than with CC (0.19), which is to be expected, by definition. The partition differs from the first one in that its 3 communities have comparable sizes. Both large communities from the previous partition loose a number of members, which are gathered to form a new, intermediary group. It contains some of the radical left (GUE-NGL), the center-left (S&D), center-right (ALDE) as well as the NI group. The two other communities are the environmentalists (G-EFL with Lepage) and the rest of the right (EPP with de Villiers), respectively. This partition is interesting, because it manages to identify a community of moderate MEPs, which sometimes vote like the environmentalists, and sometimes like the right. The community graph consequently takes the form of an imbalanced (in terms of CC) triangle in which the environmentalists and the right are in opposition, whereas the moderate are positively connected to both. This type of structure could be identified only thanks to the relaxed nature of RCC, which allows here to have positive edges *between* communities.

The optimal partition identified by Ex-RCC for $K = 4$ is quite similar to the CC partition, in the sense that S&D and EPP are in the same community, and de Villiers is apart again. But NI and GUE-NGL are now part of the EPP-dominated community, and there is a fourth community formed by MEPs from almost all political groups except EPP. Due to this heterogeneity, this last community is very difficult to interpret. Yet, the partition reaches a perfect zero imbalance, which means absolutely no edge is frustrated. This highlights the fact that increasing K mechanically decreases the imbalance, but not necessarily make the partition more informative, from the application point of view. This means that the problem, as it is formulated, does not allow to completely automate the process of identifying the best partition for the end-user.

Note that I will come back to these results in Chapter 8, when partitioning the same dataset using a different method.

7.4 Solution Space of the Correlation Clustering Problem

The second task that we conducted on signed graph concerns the space of the optimal solutions of the CC problem, and more precisely the unicity of these solutions. The standard approach when partitioning a graph in the context of a given application is to find a *single* optimal solution and focus the rest of the analysis on it, as if it was the *only* optimal solution. Yet, it is possible that several, and even many, other optimal solutions exist for the considered instance (and even more so for *quasi*-optimal solutions). Moreover, these alternate solutions can be very different, in terms of how they partition the graph [43]. Figure 7.8 illustrates this on a complete unweighted signed graph. Solving the CC problem for this graph of only 7 vertices yields no less than 22 distinct *optimal* solutions. We show only a few of them to highlight how different these partitions can be. For instance, on the one hand $\mathcal{P}_a$ and $\mathcal{P}_b$ are very similar, as they are both bisections differing only in the community assignment of v_1. On the other hand, $\mathcal{P}_c$ is quite different from them: it contains an extra community obtained by separating an element from each community of the previous solutions, in addition to v_1.

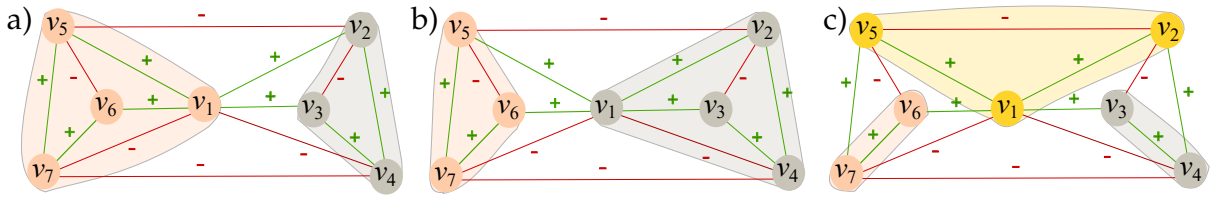

Figure 7.8: Three (out of 22) different optimal CC solutions obtained for the same 7-vertices signed graph: a) $\mathcal{P}_a = \{\{v_1,v_5,v_6,v_7\}, \{v_2,v_3,v_4\}\}$; b) $\mathcal{P}_b = \{\{v_5,v_6,v_7\}, \{v_1,v_2,v_3,v_4\}\}$; and c) $\mathcal{P}_c = \{\{v_1,v_2,v_5\}, \{v_6,v_7\}, \{v_3,v_4\}\}$. Red and green lines represent negative and positive edges, respectively. The graph is complete, but for clarity, some negative edges between communities are intentionally omitted.

Such a focus on a single solution raises several questions. First, as mentioned before, *several* optimal solutions may coexist. If so, one can wonder which network properties lead to this situation, and how many of these solutions are equally relevant to the application problem at hand. Perhaps it would be necessary to design a more appropriate version of the CC problem, in order to distinguish them, possibly based on some additional criteria related to the application context. Second, how different are these solutions? Application-wise, very similar solutions could be given the same interpretation, whereas substantially different ones might correspond to dramatically different ways of seeing the studied system. Third, when dissimilar solutions coexist for the same problem, is it possible to detect classes of similar solutions? Indeed, if such classes exist, one could need only to find one representative solution in each class, which would ease the exploration of the solution space. Fourth and finally, in case of the existence of multiple such classes, what distinguishes them from each other? Identifying these characteristic differences could provide some valuable information to understand the studied system. More generally, the answers to all these questions can drive the choice of the method used to solve CC.

In [VL3], Nejat Arınık, Rosa Figueiredo and I tackled these questions through the characterization of the space of optimal solutions associated with a collection of signed graphs. I summarize this work in the rest of

this section, starting with a brief description of our methods (Section 7.4.1), then exposing our main results (Section 7.4.2).

7.4.1 Methods

We proceeded by randomly generating a number of signed networks with various characteristics: number of vertices, number of communities, and imbalance. We then identified all their optimal solutions, and study how the number and nature of these solutions was affected by the network characteristics.

Graph Generation As a first step in the study of the solution space of the CC problem, we focused on unweighted complete signed graphs. We generated a large collection of artificial graphs using a custom random model. One advantage of using such graphs is that certain of their characteristic are not controlled, e.g. graph density, proportion of negative edges, and weight distribution, making the number of model parameters smaller, and the empirical results easier to explore.

We proposed a simple yet principled random model designed to produce complete unweighted networks with built-in modular structure. It relies on only three parameters: $|V|$ (number of vertices), K_0 (initial number of communities) and q_m (proportion of frustrated edges). First, we produce a graph containing the appropriate number of vertices, divided into K_0 approximately equal-sized communities to form a partition $\mathcal{P}_0$. We connect them with negative and positive edges, in such a way that this complete graph is perfectly balanced. Second, we introduce some imbalance into the graph, so as to match parameter q_m. For this purpose, we randomly select a pair of negative and positive edges, then switch their signs in order to make both of them frustrated. The process is repeated to other pairs of edges. On the one hand, this mechanism causes a restriction on the upper bound of q_m. But on the other hand, it allows preserving the ratio of positive to negative edges in the graph, and therefore avoids introducing another parameter in the model. It is important to note that the detected graph imbalance $I(G)$ does not necessarily match q_m, as it may be possible to find a better partition $\mathcal{P}$ than the initial $\mathcal{P}_0$ due to the introduction of random frustration.

Finding all optimal solutions is computationally costly, and constrains the graph size that we can handle to a maximum of 36 vertices. Our graph being relatively small, we only considered modular structures containing 2 to 4 communities. We generated 100 instances for each parameter set, in order to get stable results. In total, we produced a total of 22,200 instances. All these data [S7], as well as our model [C2], are publicly available online.

Analysis of the Solution Space The framework that we designed to analyze the space of optimal solutions of the CC problem was implemented by Nejat Arınık, and the source code is available online[15]. The first step of this framework is the enumeration of all distinct optimal solutions. This can be very time- and memory-consuming, so we needed an efficient method. For this purpose, we modeled the CC problem through Integer Linear Programming (ILP) and applied the problem-independent method introduced by Danna *et al.* [69] to enumerate solutions in such context. For further computational gain, we strengthened the underlying ILP model through the *cutting plane* approach, as proposed by Alès *et al.* [7]. Incidentally, we are currently working on other improvements of this enumeration step.

Once all solutions are identified for all graph instances, our second step is to perform a cluster analysis over the solution space of each instance. First, we computed the similarity between each pair of solutions identified for a given instance. As a side work, we designed a framework just to compare the available similarity measures [VL1], and select the most appropriate one for the present case: the *Variation of Information* (VI) [171]. We applied the k-medoids clustering method [137] to perform the cluster analysis, and used the *Silhouette* measure to identify the best clustering [204]. It takes a value between -1 and $+1$, where the latter represents the best possible clustering.

[15] https://github.com/CompNet/Sosocc

After the clustering, we have identified collection of K classes of solutions for each graph instance. Each one corresponds to a group of solutions that are, by construction, relatively similar. In order to assess how similar they are, we leveraged a concept that we called *core part*. The core part of a class is the maximal subset of vertices whose relative community assignment stays constant over all the solutions constituting the class. When two vertices belong to the core part, we call them *core vertices*, and they are either always in the same community, or always in different communities, for all the solutions of the class. Consequently, vertices that are always *isolated* (i.e. that constitute their own community) are core vertices, as their community assignment always differ from the rest of the core part. It is also possible to obtain an *overall core part* by proceeding similarly with all the solutions in the space (i.e. not focusing on a single class). We identified the core part of each class by adapting a method proposed by Lancichinetti and Fortunato [149] in the context of consensus clustering (cf. [VL3] for details).

7.4.2 Main Results

I review briefly our main results, concerning the size of the solution space (number of optimal solutions by instance), the classes of solutions detected in this space (i.e. the diversity of these solutions), and the nature of the core parts used to characterize these classes. I focus my analysis on the results obtained for an original partition of two communities ($K_0 = 2$), but our results obtained for more communities are similar in nature (see [VL3]).

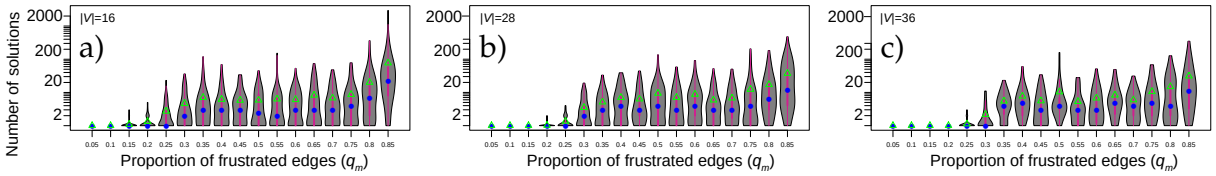

Figure 7.9: Number of solutions as a function of q_m, for $K_0 = 2$. In each violin plot, the interquartile range is shown as a purple thick line, the mean as a green triangle, and the median as a blue dot.

Number of Solutions Figure 7.9 shows the number of optimal solutions as a function of q_m, for different graph orders $|V|$ (note that the y-axis uses a logarithmic scale). Small q_m values tend to result in a unique solution: they represent 45% of the graph instances, for $K_0 = 2$. An increase in q_m, i.e. introducing more frustrated edges, leads to more solutions. This fact can be explained by considering the detected graph imbalance $I(G)$ as a function of q_m, as illustrated in Figure 7.10. We observe that when q_m increases, $I(G)$ also increases for small q_m values, but then reaches a plateau. Yet, one would expect the detected imbalance to directly depend on the number of frustrated edges originally placed in the graph. However, when q_m exceeds some threshold, this number becomes so large that it provides some form of flexibility to graph partitioning. Consequently, even if these frustrated edges are randomly distributed, it becomes possible to partition the graph into a larger number of smaller communities allowing to reach a lower imbalance than expected (though still high). In addition, this flexibility also allows finding several equally good partitions into such small communities, which leads to multiple optimal solutions. This is illustrated in Figure 7.11, which displays the number of detected communities as a function of the number of frustrated edges q_m. One can observe an increase in the number of detected communities and/or in their dispersion when q_m increases, up to a certain point.

We expected the graph order (i.e. $|V|$) to affect the number of solutions, as one could suppose that a larger graph offers more possible partitions. However, this does not seem to be the case in our results. We adopt a different angle by considering the number of solutions as a function of the *detected* imbalance $I(G)$ in Figure 7.12 (the y-axis uses a logarithmic scale). It confirms that the number of solutions tends to increase with the imbalance, whereas the graph order does not have a clear effect.

To conclude this part, our experiment reveals that it is possible to obtain many optimal solutions when solving the CC problem on certain networks. If the order of the graph does not seem to affect the number of

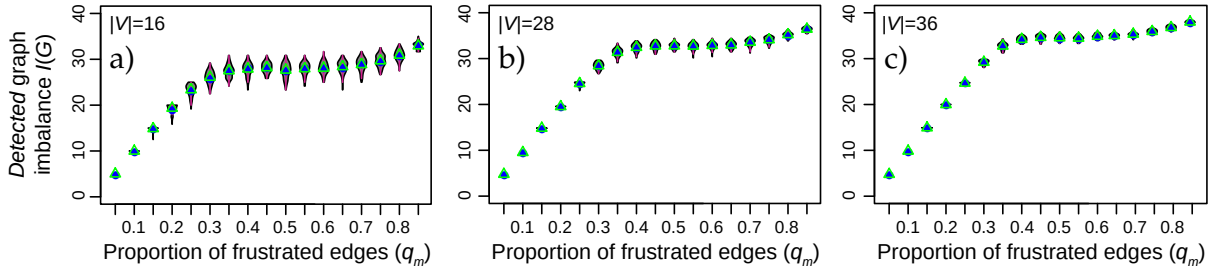

Figure 7.10: Detected graph imbalance as a function of q_m, for $K_0 = 2$.

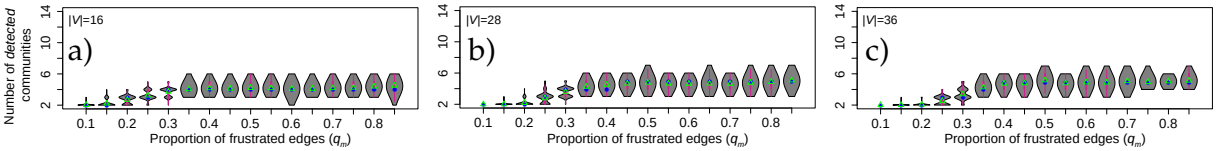

Figure 7.11: Number of detected communities as a function of q_m, for $K_0 = 2$.

solutions much, on the contrary the graph imbalance certainly plays a key role. A larger imbalance generally leads to more optimal solutions, and in addition our plots show that the associated dispersion also increases, resulting in extreme values. Thus, it certainly seems necessary to assess the multiplicity of solutions in case of relatively imbalanced networks. From a practical point of view though, certain types of real-world networks are known to have a low imbalance [154]. In this case, identifying all optimal solutions might not seem necessary. But there is no absolute guarantee to get a unique, or even few optimal solutions when the imbalance is low. For instance, we get a maximum of 55 (and an average of 4.25) solutions for $K_0 = 3$, $|V| = 16$, $I(G) = [0.10, 0.15[$, which is already quite a large number of solutions for such a low imbalance.

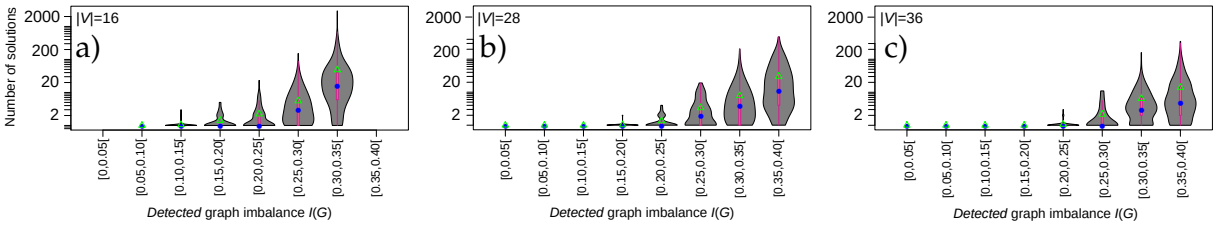

Figure 7.12: Number of solutions as a function of the detected imbalance, for $K_0 = 2$. Missing values correspond to undefined parameter values or unavailable data.

Diversity of the Solutions, and Core Parts Next, we consider the numbers of classes of solutions produced by our framework. Remember that, by construction, a class is a group of highly similar solutions. Figure 7.13 displays the proportions of cases for which there is a single solution class, as a function of the detected imbalance $I(G)$. Note that we do not include the instances for which there is only a *unique* optimal solution, as they were already discussed before. This results in the absence of certain histogram bars in the plots. It appears that our method always detects a single class for slightly imbalanced graphs, and that the number of classes increases with the imbalance. Overall, single class instances represent 66% of the cases for $K_0 = 2$.

We now turn to the characterization and comparison of the classes, through the analysis of their core parts. We express the size of a core part in terms of proportion of the graph order $|V|$. Our assumption is that, for a class to be considered as cohesive, its core part should be large enough. On the contrary, if the classes are clearly separated, the *overall* core part (processed over all solutions) should be small. Figure 7.14 shows the distribution of core part sizes as a function of the number of detected solution classes (bottom x-axis). In addition, the values are grouped using the detected imbalance $I(G)$ (top x-axis of each plot). Like before, these plots do not show cases with only a unique solution. Our first observation is that the core part size seems to increase with the number of classes, at least until it reaches a plateau. This means that the classes are more and more cohesive internally. Moreover, the dispersion also decreases when the class number

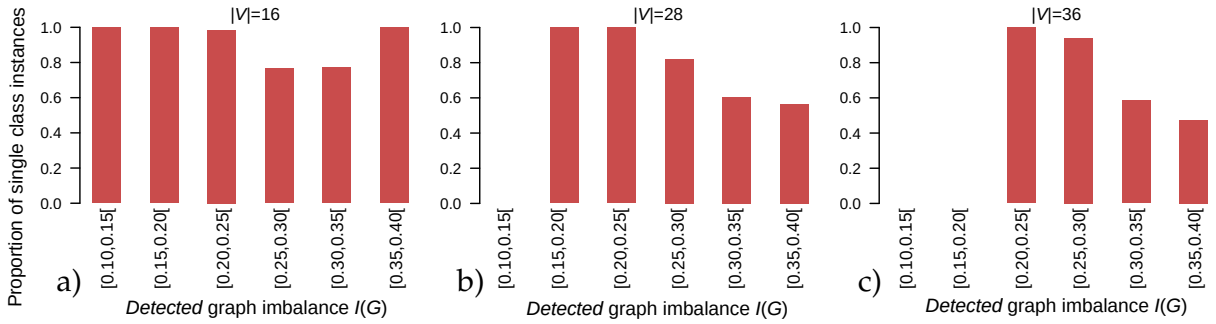

Figure 7.13: Proportion of single-class instances as a function of the detected imbalance, for $K_0 = 2$. Missing values correspond to undefined parameter values or unavailable data.

increases. In the single-class case, the core part size can be extremely small, close to zero. This indicates that, in certain cases, the cluster analysis is not conclusive: the Silhouette score is too low to conclude there are several classes, but the single class is not cohesive, and contains some sensibly different solutions. For a specific real-world application, one would need to manually consider this situation.

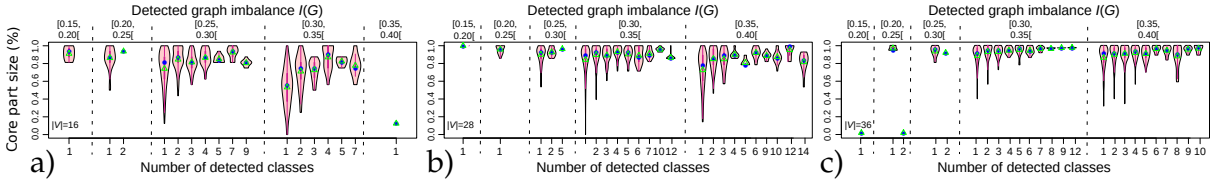

Figure 7.14: Proportion of the graph covered by the class core parts, as a function of the detected imbalance and number of classes, for $K_0 = 2$. Missing values correspond to undefined parameter values or unavailable data.

Typology of Solution Spaces In conclusion we empirically identified four different types of solution spaces. In the first, which tends to occur only in slightly imbalanced graphs, there is only one optimal solution. The second type corresponds to the case where there are multiple solutions distributed over several distinct and cohesive classes. This tends to happen for larger imbalance values. In the third type, we have a single class containing multiple solutions that are very similar, resulting in a large core. A small core means that this class is not cohesive, and corresponds to the fourth type. This typology shows that the answers to our initial questions are multiple and depend on the considered graph. Our work highlights the necessity to develop a method allowing to handle these different cases.

7.5 Conclusion

In this chapter, I summarized my work on signed graphs. This included some experimental aspects, such as testing whether negative edges and weight-based filtering affect graph partitioning, and comparing empirically the results produced by methods solving variants of the CC problem. It also included more theoretical work concerning the CC problem itself, as we studied the space of its optimal solutions. This last work is particularly interesting, as it opened a number of perspectives. First, we have just developed a better way to enumerate exact solutions of the CC problem, in order to explore the space of larger graph instances. This work is currently being reviewed [VL87]. Second, we plan to reproduce the same type of experiment, but with non-complete graphs, as well as weighted graphs, in order to study how these properties affect the solution space of CC. Such graphs are more prevalent in applications, therefore these result could be more relevant to the community.

Third, based on our knowledge of the solution space, we are in the process of designing a heuristic method allowing to enumerate optimal solutions such that each solution class is represented. This would be particularly interesting in an applications context, as this would allow the end-user to take several possibly

very different optimal partitions into account while interpreting their results. The fourth perspective is related to such heuristic, whose definition would greatly benefit from knowing the type of solution space associated to the considered graph, according to our typology (single solution; single solution class; several solution classes; multiple solutions but no classes). Indeed, this would allow estimating the number of solutions the search method must output. For this purpose, we are currently advising a group of M1 students that use machine learning to perform such predictions [U1]. The general idea is to leverage signed graph features to train classifiers and regressors to predict various characteristics of the problem space. We first used topological measures as features, before turning to graph embeddings methods. These are generic embeddings though, and we consider learning the representation and prediction task simultaneously to obtain a more relevant vector-based representation of a graph.

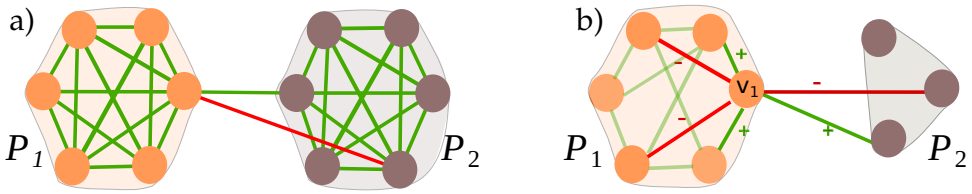

Figure 7.15: Two examples of signed graphs with two optimal solutions for CC. (a): single community containing all vertices vs. communities P_1 and P_2. (b): communities P_1 and P_2 with $v_1 \in P_1$ vs. $v_1 \in P_2$.

The fifth perspective is the exploration of problems related to signed graph partitioning, other than CC. Indeed, the fact that one can get many optimal solutions when solving CC on a real-world graph could mean that the problem is not formulated appropriately. We already experimented with RCC, the relaxed version of CC, in which the notion of balance is defined in a *looser* way. We plan to explore its solution space too, as we did for CC. But we also explore variants that are *more constrained*, such as the 2-edge connected constraint. Figure 7.15 shows two graphs for which CC has two optimal solutions. In Figure 7.15.a, the solutions are completely different: it is either a single community containing all vertices, or two communities P_1 and P_2. There are only two edges between P_1 and P_2, and so the second solution seems intuitively preferable, though. In Figure 7.15.b, the solutions are very similar, as they differ only on the membership of v_1. If one considers connections of higher order though, it seems preferable to put v_1 in P_1. The 2-edge connected constraint allows formalizing this intuition, as it requires that the vertices forming a community are connected by at least two positive paths. The ILP model that we use for CC is very suitable to the inclusion of this kind of additional constraints, and we have already started experimenting with such variants of CC [VL94].

8

Analysis of Multiplex Networks

8.1 Context

A *multiplex network*, or edge-colored multigraph in the nomenclature of Kivelä *et al.* [141], can be viewed as a *multilayer* network in which all vertices are present in each layer, and one vertex is connected to all its counterparts in all layers (the so-called *categorical* inter-layer coupling). The way vertices are interconnected within a layer can vary from layer to layer, though. Figure 8.1 shows an example of random multiplex network containing 10 vertices and 3 layers. By comparison, a regular network constituted of a single layer is called *uniplex*. The concept of multiplex network is presented formally in Definition 8.1.1.

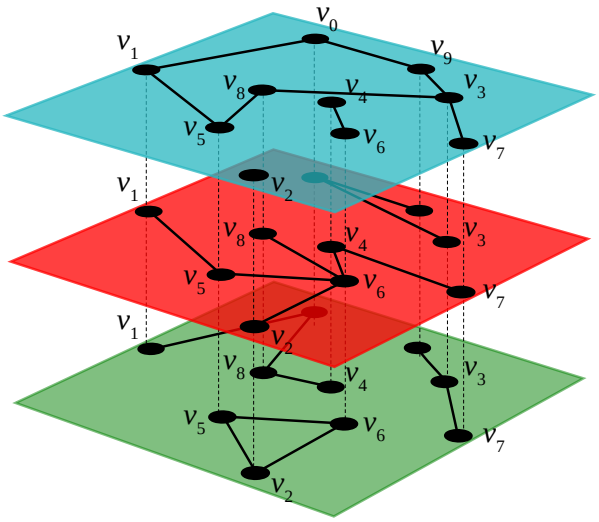

Figure 8.1: Example of multiplex network containing 10 vertices and 3 layers.

Definition 8.1.1 (Multiplex Network) *Let $G = \{G_1, \ldots, G_L\}$ a **multiplex network** constituted of L layers. Each layer $\ell \in \{1, \ldots, L\}$ is itself a uniplex network $G_\ell = (V, E_\ell)$, where $V = \{v_1, \ldots, v_I\}$ is the set of vertices, and $E_\ell \subset V^2$ is the set of edges for layer ℓ. Note that all layers have the same vertex set V, but possibly different edge sets. In case of undirected graph, we assume that edges are lexicographically ordered pairs.*

The principal interest of this type of network is that it allows modeling systems in which one can distinguish several distinct *types* of relationships or interactions between the system components. Each layer then corresponds to a specific type of relationship. For instance, the multiplex network represented in Figure 8.1 could represent three types of relationships within a group of persons: familial, friendly, and professional. Some relationships are unique to a single layer, some exist simultaneously in certain layer (e.g. working with one's spouse), and of course, some do not exist at all. Multiplex networks can be considered as rich because the type associated to each edge constitutes an additional information, compared to plain graphs. Technically, a multiplex graph can be seen as a specific type of edge-attributed network (cf. Chapter 2) in which edges are described by a categorical attribute representing the type of the relationship. However, in the literature they are considered as a class on their own, and I proceed similarly in this manuscript.

According to Wasserman and Faust [231], such networks have been used in Sociology since the 1960s. More recently, the development of the complex network analysis field has led to an increase in research activity regarding multilayer networks, and multiplex networks in particular, as shown by the numerous books and surveys published in the last few years [8, 21, 66, 82, 141]. As observed in Chapter 1 more generally for other types of feature-rich networks, there are much fewer tools to study multiplex networks than uniplex ones. This is due to the fact that the community has focused its efforts on the simpler mathematical objects at first, before turning to more advanced concepts.

Similarly, I started working on multiplex networks only after having gathered a certain experience from my work on uniplex networks. The goal of this chapter is to present this work. From a methodological perspective, I mainly dealt with two tasks related to multiplex networks. Both are the transposition to the multi-relational case of classic tasks performed on uniplex networks. First, in 2015–16 I collaborated with Alexandre Reiffers-Masson to propose a multiplex centrality measure, based on an opinion diffusion model (Section 8.2). Second, in 2019–21, Rosa Figueiredo, Nejat Arınık and myself defined a method to find multiple vertex partitions of multiplex networks (Section 8.3). This work is not related to the second and took place in the context of Nejat Arınık's PhD, which was advised by Rosa Figueiredo and myself. I conclude this chapter with a description of a more applied ongoing work on multiplex networks, and discuss possible perspectives (Section 8.4).

8.2 Opinion-Based Multiplex Centrality Measure

When identifying the central vertices in a multiplex network, it is of course necessary to take the multilayer nature of the structure into account, in order to avoid any information loss. For this purpose, various measures have been proposed in the literature. Most of them generalize widespread existing unilayer measures such as degree [20, 76, 164], betweenness [50, 163, 211], closeness [164, 212], Eigenvector [20, 76, 210], PageRank [63, 116, 181] or HITS [143]. These generalizations rely on the adaptation of unilayer concepts to the multilayer case. For instance, measures based on matrix decomposition are modified to handle tensors, whereas in others, the concept of geodesic distance is altered to take inter-layer paths into account. Several recent articles review multilayer networks and the related centrality measures [28, 76, 141].

The multiplex centrality measure that we proposed with Alexandre Reiffers in [VL10] generalizes another type of uniplex approach, based on the resolution of an optimization problem on a diffusion model. One of the first opinion diffusion models was developed by DeGroot [78]. In this model, vertices represent agents which update their opinion (a real number in $[0, 1]$) over time, by taking the average opinion of their neighbors. Some of the major extensions of this model are summarized by Jackson [132]. Later works focused not only on understanding the opinion adoption process, but also on controlling it. Recent articles [25, 34] suggest that one can use the theory of optimization and control in order to design efficient strategies for the control of opinion diffusion. In [34], the authors propose to impose a particular opinion to certain vertices in order to make the whole network adopt it too. In [25], the authors study how much a company needs to invest in a person in order to improve the adoption of some product it wants to sell to the whole network. The vertex targeted by these strategies can somehow be considered as central, and we want to explore this type of centrality in this chapter.

Our approach is based on a model representing how the individual opinion regarding a given topic of interest evolves among a group of persons. Each layer represents the influence people have on each other in a given social context or for a given social media. For instance, consider social media such as Twitter, Facebook, LinkedIn or ResearchGate. These services have different purposes, and the interconnections among the same group of users is likely to differ from one such medium to the other. This has consequences in terms of information propagation, since one can not only exchange information using a single social medium, but also receive some information through one medium and diffuse it via another one (e.g. posting a tweet as a Facebook message). A person is influenced by their neighbors, like in DeGroot models, but in all layers. Moreover, we introduce an additional influence in the model, which represents an external party able to affect each agent individually. For example, in a marketing context, this external influencer could be a firm willing to direct its communication towards certain persons in the considered social group. Our centrality measure is related to the solution of the optimization problem consisting in determining which amount of external influence to invest in each person, in order to maximize the overall opinion of the social group. Our work is not the first one connecting targeting strategies to vertex importance: in [25, 33, 34], the authors maximize opinion diffusion in a uniplex social network, by spending resources on each person. Our approach differs on two main points: first, we deal with multiplex networks; and second, we leverage the diffusion model to derive a proper centrality measure.

In this section, I give a brief description of our multiplex centrality measure (Section 8.2.1), and provide our main results obtained on real-world networks (Section 8.2.3). The interested reader will find the proofs and additional information in [VL10].

8.2.1 Stochastic Model for Opinion Dynamics

Let us consider a group of people, some of which have influence over others. This influence is conveyed through a collection of distinct social media. We assume that the opinion of a person regarding a topic of interest depends on the opinion of the people influencing her through these social media. Formally, this system can be modeled as a multiplex graph, in which the vertex set $V = \{v_1, \dots, v_I\}$ represents the persons, and each layer ℓ ($1 \leq \ell \leq L$) represents a distinct social medium. We note $W_\ell \in [0,1]^{I \times I}$ the weighted adjacency matrix corresponding to the edges of layer ℓ. We define its ij-entry $w_{ij\ell}$ as the probability that person v_j is influenced by person v_i in social media ℓ, and consequently mimics her opinion.

Let us now describe the process taking place on this network. We assume that a person $v_i \in V$ is influenced by a source external to the network, according to a Poisson point process of intensity $\lambda_i \in \mathbb{R}_+$. Moreover, a person v_i decides to mimic the opinion of one of her neighbors in layer ℓ according to a Poisson point process of intensity $\alpha_{i\ell} \in \mathbb{R}_+$. Let $k \in \mathbb{N}_+$ denote the k^{th} event (external or internal influence). We define $\Lambda := \sum_{j=1}^{I} \sum_{d=1}^{L} \lambda_j + \alpha_{jd}$. Then, $\lambda_i \Lambda^{-1}$ is the probability that the k^{th} event is individual v_i undergoing an external influence, whereas $\alpha_{i\ell}\Lambda^{-1}$ is the probability for this event to be individual v_i mimicking her neighbors' opinion in social medium ℓ (i.e. undergoing an internal influence).

For each event k, let $x_i(k) \in [0,1]$ be the opinion of person v_i. A zero value means that person v_i has no interest for the considered topic, whereas 1 represents a full interest. By modeling opinions using real values in $[0,1]$, we follow the line of work of DeGroot [78]. We call the vector $\mathbf{x}(k) := (x_1(k), \dots, x_I(k))$ the *opinion profile* of the whole social group at event k. As mentioned before, it can be updated due to the direct influence of other individuals' opinions, or due to an external influence. For each i, let $x_i(0) = x_i^0 \in [0,1]$.

Let us now detail how the opinion of a person v_i is updated. One of three scenarios are possible:

1. If at event k, person v_i receives a message (with probability $\lambda_i \Lambda^{-1}$), then her opinion is updated as follows:

$$x_i(k+1) = x_i(k)(1-\delta) + \delta \tag{8.1}$$

   where $\delta \in]0,1[$ is the *step-size*.

2. If at event k, person v_i decides to imitate her neighbors in social medium ℓ (which occurs with probability $\alpha_{i\ell}\Lambda^{-1}$), then her opinion is updated as follows:

$$x_i(k+1) = x_i(k)(1-\delta) + \delta\left(\sum_{j=1}^{I} w_{ji\ell}x_j(k)\right), \tag{8.2}$$

3. If event k does not concern person v_i (which occurs with probability $(\sum_\ell \sum_{j\neq i} \alpha_{i\ell} + \lambda_j)\Lambda^{-1}$), then her opinion is updated as follows:

$$x_i(k+1) = x_i(k)(1-\delta). \tag{8.3}$$

In this model, opinion evolution is a constant step-size stochastic approximation. Indeed, for each event k and each person v_i, $x_i(k)$ can be rewritten as follows:

$$x_i(k+1) = x_i(k) + \delta\left(Y_{i\ell}(k) - x_i(k)\right), \tag{8.4}$$

$$x_i(0) = x_i^0, \tag{8.5}$$

where

$$Y_{i\ell}(k) := \begin{cases} 1 & \text{w.p} \quad \lambda_i\Lambda^{-1}, \\ \sum_{j=1}^{I} w_{ji\ell}x_j(k) & \text{w.p} \quad \alpha_{i\ell}\Lambda^{-1}, \\ 0 & \text{w.p} \quad \Big(\sum_\ell \sum_{j\neq i} \alpha_{i\ell} + \lambda_j\Big)\Lambda^{-1}. \end{cases} \tag{8.6}$$

The previous equations suggest that for each person v_i, $x_i(k)$ is a stochastic finite difference Euler schemes of the following system of differential equations:

$$\dot{x}_i(t) = \lambda_i\Lambda^{-1} + \Lambda^{-1}\sum_{\ell=1}^{L}\alpha_{i\ell}\sum_{j=1}^{I} w_{ji\ell}x_j(t) - x_i(t), \tag{8.7}$$

$$x_i(0) = x_i^0. \tag{8.8}$$

Let $\overline{W}$ be a matrix such that the ij-entry of $\overline{W}$ is equal to $\overline{w}_{ij} = \sum_{\ell=1}^{L}\alpha_{i\ell}w_{ji\ell}$. Let W^T be the transpose of matrix W. Let A be equal to $(\overline{W} - Id_I)^{-1}$ where Id_I is the identity matrix of dimension I[2].

We are interested to compute, if it is possible, $\lim_{k\to+\infty} x_i(k)$. We use the theory of stochastic approximation [32], which highlights the relation between $\lim_{k\to+\infty} x_i(k)$ and $\lim_{t\to+\infty} x_i(t)$. The next lemma gives us the expected result.

Lemma 8.2.1 *If for all i, $\sum_{j=1}^{I}\overline{w}_{ij} < 1$, $\frac{1}{2}[\overline{W} + \overline{W}^T]$ has negative Eigenvalues and $\delta \ll 1$, then for each i,*

$$\lim_{k\to+\infty} x_i(k) \quad = \quad x_i^*, \tag{8.9}$$

where x_i^* is solution of

$$0 = \lambda_i\Lambda^{-1} + \Lambda^{-1}\sum_{\ell=1}^{L}\alpha_{i\ell}\sum_{j=1}^{I} w_{ji\ell}x_j^* - x_i^*. \tag{8.10}$$

We now have a framework that models the evolution of individual opinion occurring on a multiplex social network, and the mathematical expression of the asymptotic behavior of opinion dynamics. On this base, in the next sections, we define the multiplex opinion problem before deriving our multiplex centrality measure.

8.2.2 Derivation of the Measure

In our model, the parameter λ_i determines how much external influence person v_i receives. Our centrality measure is based on the control of this parameter. We assume that the centrality of a person in the whole multiplex network depends on the amount of external influence one should exert on her in order to increase the global opinion level of the whole social group. According to this statement, finding the most central person amounts to finding the person whose stimulation (through external influence) maximizes the total opinion. So, in order to process our centrality, we need first to solve an optimization problem.

We want our centrality to quantify how important an individual is, in term of resource allocation. For this purpose, we need to maximize the opinion profile $\mathbf{x}$ by controlling the amount of external influence $\lambda := (\lambda_1, \ldots, \lambda_I)$ spent on the individuals of the social group. We define the *Opinion Centrality* measure as follows:

Definition 8.2.1 (Opinion Centrality) *Let $U(\mathbf{x}^*(\lambda))$ the selected utility function. The **Opinion Centrality** is the vector λ^O which is the optimum of the following problem:*

$$\max_{\lambda \geq \mathbf{0}} U^R(\mathbf{x}^*(\lambda)) := \sum_{i=1}^{I} x_i^*(\lambda) \quad - \quad \underbrace{\frac{\gamma}{2} \sum_{i=1}^{I} (\lambda_i)^2}_{\text{Regularization function}} \tag{8.11}$$

where $\gamma \in \mathbb{R}_+$ is the regularization coefficient; and for each i, $x_i^*(\lambda)$ is solution of:

$$\lambda_i \Lambda^{-1} + \Lambda^{-1} \sum_{\ell=1}^{L} \alpha_{i\ell} \sum_{j=1}^{I} w_{ji\ell} x_j^*(\lambda) - x_i(\lambda) = 0; \tag{8.12}$$

and the total intensity of the external influence is constrained by a so-called budget $R > 0$:

$$\sum_{i=1}^{I} \lambda_i = R. \tag{8.13}$$

*We call this optimization problem the **Regularized Opinion Maximization Problem (OMP)**.*

Our centrality defines an index, per person, which is equal to the amount of resources that an external influencer needs to spend on each person such that he will maximize the opinion of the whole social group, regarding some topic or product of interest. Therefore, if a vertex is twice as central as another, this can be interpreted as the need to invest twice the amount of resources in this vertex in order to obtain optimal diffusion. In addition to creating a ranking measure, the opinion centrality can also be used to design a targeting strategy. The budget R is used to prevent the external influencer from targeting every person in the social group, which would lead to a degenerate situation. The regularization term allows avoiding an other degenerate situation, where the whole budget is attributed to a single person.

We are now able to compute a mathematical expression of the opinion centrality.

Lemma 8.2.2 *If for all i, $\sum_{j=1}^{I} \overline{w}_{ij} < 1$, and if γ is such that*

$$\gamma > (\Lambda R)^{-1} I^2 (\max\{a_{ij}\} - \min\{a_{ij}\}), \tag{8.14}$$

then for all j

$$\lambda_j^O = R I^{-1} + \gamma^{-1} I^{-1} \Lambda^{-1} \sum_{i=1}^{I} \sum_{j=1}^{I} a_{ij} - \gamma^{-1} \Lambda^{-1} \sum_{i=1}^{I} a_{ij}. \tag{8.15}$$

It is worth noticing that the difference between two opinion centrality values will increase the more γ and R decrease (R appears in Λ^{-1}). Despite this fact, the ranking itself is independent of γ and R.

8.2.3 Experimental Validation

To study the behavior of our measure, we consider a selection of 20 real-world networks, listed in Table 8.1. They were retrieved from Nexus[16], the network repository of the igraph library [67], as well as from M. De Domenico's Web page[17]. We selected these networks in order to get data of various sizes, ranging from 10 to 8,215 vertices, 35 to 43,129 edges, and 2 to 339 layers.

For each network, we process the opinion centrality, as well as multiplex variants of classic uniplex measures: Degree, PageRank, Eigenvector, HITS (hub and authority) and Katz centralities. The latter are computed using the software MuxViz[18] [75], which implements tensorial generalizations. Each of them is processed for each layer by taking the multiplex information into account, and an overall measure is obtained by aggregating the resulting values over all layers. The R scripts we wrote to process the opinion centrality are publicly available online [S10].

One could compare the measures simply based on the values they produce. However, centrality measures are generally used to compare vertices *within* a single network, in which case the values themselves are not relevant: one should consider their rank instead [210]. We consequently emphasize this aspect in our experimental assessment.

We first study how parameter α (which controls how much a person mimics their neighbors) affects the opinion centrality rankings, and then how different these are from those obtained for the other measures. However, the multiplex networks available online do not include this information: only the structure of the network (i.e. the W_ℓ matrices, in the notation of our model). So, we assume that all $\alpha_{i\ell}$ are equal to the same value noted $\hat{\alpha}$, and consider a wide range $\hat{\alpha} \in]0; 100]$. Our results show that when $\hat{\alpha}$ increases, the opinion centrality values increase too (without considering the budget constraint). We use Spearman's correlation to compare the opinion centrality values processed with all considered $\hat{\alpha}$, and systematically obtain a maximal correlation of 1. So, it turns out $\hat{\alpha}$ affects the opinion centrality values, but not their rank. The value of this parameter is consequently of no importance if one's objective is to rank the vertices. Note, however, that if α is provided as a part of the input data, it is likely to be heterogeneous (by opposition to the unique $\hat{\alpha}$ we used here as a substitute), and this can lead to ranking differences. In other words, α should be used when available, $\hat{\alpha}$ is only a way to model missing information.

The values obtained with the opinion measure are generally distributed relatively similarly in the studied networks. These distributions are left-skewed, i.e. most of the vertices have a higher centrality, with respect to the considered network. Moreover, they are distributed relatively homogeneously around a characteristic value, which varies depending on the network. The dispersion around the characteristic value also depends on the data. As an illustration, Figure 8.2 shows two typical distributions obtained for two networks. To ease the comparisons, the opinion centrality of a vertex (x axis) is expressed in terms of the *proportion of the budget* it receives in the optimal solution, according to our model.

We now compare the opinion measure to the other multiplex centrality measures. For each network, we process Spearman's correlation to compare the ranks obtained for the opinion centrality to those of the other measures. The results are displayed in Table 8.1. Certain measures (Eigenvector, HITS, Katz, PageRank) could not be processed for the largest networks, due to their memory cost. Missing in- and out-degree values correspond to undirected networks. The correlations vary much depending on both the network and measure of comparison, ranging from -0.97 to 0.26. Overall, we observe a mild to strong negative correlation for all considered measures. However, it is worth noticing that, for a given measure, the magnitude of the correlation varies depending on the network. This means that the opinion measure does not simply systematically reverse the rankings of the considered measure. Moreover, the correlation also varies from measure to measure for a

[16] http://nexus.igraph.org/
[17] http://deim.urv.cat/~manlio.dedomenico/data.php
[18] http://muxviz.net/

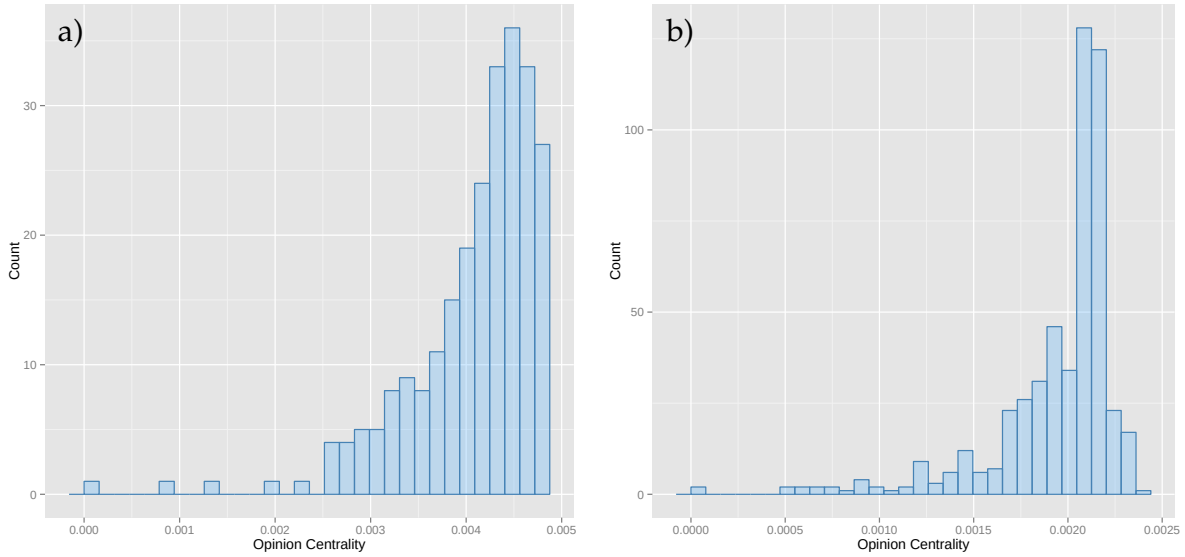

Figure 8.2: Distribution of the opinion measure for the CKM Physicians (a) and Pierre Auger collaborations (b) networks.

given network. These observations confirm empirically that the opinion measure characterizes vertices based on different criteria than the other considered centrality measures, as designed.

To get a better insight of the opinion measure, We consider the vertices individually. Figure 8.3.a represents the difference in ranking between the opinion centrality and the Authority measure, on the CKM Phys. network. The right plot is built similarly, but focuses on the out-degree in the Lazega network. Both plots are very typical of what we observe for other data and centrality measures. For a given centrality measure (here: the Authority and the out-degree), the vertices are ordered on the x axis by increasing centrality values, whereas the y axis represent how the vertex ranking changes from the considered measure to the opinion measure. On the extremes, the opinion measure tends to demote the vertices the most central according to the other measures, whereas it promotes the least central ones. Those all become moderately central, according to the opinion measure. On the contrary, certain vertices previously with intermediate ranking are placed among the most or least central vertices by the opinion measure. By comparison, the same plot

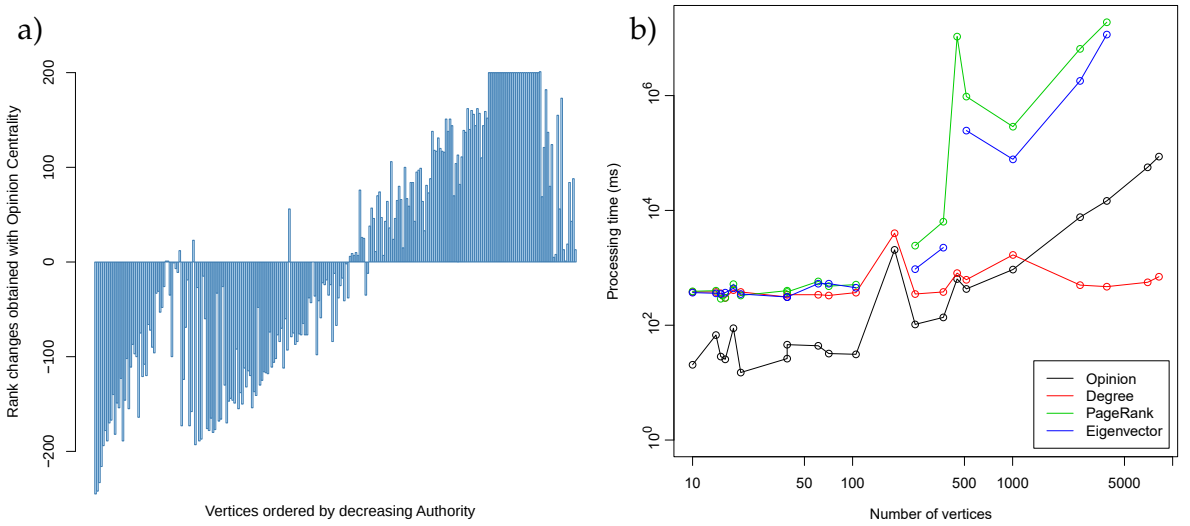

Figure 8.3: a) Rank difference between the Opinion centrality and the Authority measure, for each individual vertex in the CKM Physicians network. Each bar corresponds to a vertex, and its height matches the rank difference. b) Processing times of the main considered measures, as functions of the numbers of vertices.

Network	Degree			Eigen-	HITS		Katz	Page-
	All	In	Out	Vector	Auth.	Hub		Rank
Aarhus CS [163]	−0.81	–	–	−0.94	−0.74	−0.74	−0.74	−0.74
Arabidopsis [74]	−0.76	−0.45	−0.65	–	–	–	–	–
C. Elegans [74]	−0.38	−0.72	0.13	−0.96	–	–	–	–
CKM Phys. [61]	−0.80	−0.96	−0.17	−0.86	–	0.02	−0.49	−0.15
Drosophila [74]	−0.73	−0.89	−0.36	–	–	–	–	–
EU-Air [45]	−0.95	–	–	−0.96	–	–	–	–
FAO Trade [74]	−0.41	−1.00	0.02	–	–	–	–	–
Hepatitus C [75]	−0.18	0.01	−0.01	−0.57	–	−0.17	−0.49	–
Human-HIV1 [75]	−0.48	−0.51	−0.12	−0.72	−0.30	−0.11	0.26	−0.12
Kapferer1 [136]	−0.90	−0.91	−0.86	−0.98	−0.78	−0.68	−0.77	−0.66
Kapferer2 [136]	−0.90	−0.91	−0.87	−0.95	−0.83	−0.76	−0.82	−0.75
Knoke [142]	−0.68	−0.72	−0.34	−0.76	−0.77	−0.16	−0.89	−0.10
Lazega [152]	−0.83	−0.93	−0.50	−0.94	−0.90	−0.48	−0.95	−0.48
London [77]	−0.72	–	–	−0.77	−0.03	−0.03	−0.03	−0.06
Padgett [42]	−0.93	–	–	−0.94	−0.89	−0.89	−0.89	−0.89
Pierre Auger [73]	−0.46	–	–	−0.73	–	−0.40	−0.40	−0.47
Rattus GPI [74]	−0.50	−0.73	−0.08	−0.88	–	–	–	–
Roethlisberger [201]	−0.60	−0.66	−0.59	−0.89	−0.79	−0.73	−0.82	−0.77
Sampson [41]	−0.71	−0.93	0.14	−0.82	−0.82	0.00	−0.96	−0.06
Thurmann [221]	−0.73	−0.97	−0.70	−0.96	−0.82	−0.71	−0.78	–
Wolfe Primates	−0.50	−0.59	−0.45	−0.44	−0.38	−0.36	−0.38	−0.36

Table 8.1: Spearman's correlation between the opinion measure and the other considered multiplex centrality measures.

built to compare two alternative multiplex measure typically leads to a flatter figure, with smaller ranking differences, especially regarding the most and least central vertices.

The ranking differences observed between the opinion centrality and the other multiplex measures are due to the optimization problem it is based upon. Indeed, in this problem, it can be necessary to externally stimulate certain vertices which do not have a particularly high degree, or have no neighbors with a particularly high degree. For instance, a leaf vertex whose unique edge is directed towards the rest of the network will not be reachable from another vertex. So, if one wants to influence the opinion of the whole network, it is worth acting directly on this vertex. This kind of vertex is typically considered by measures such as the Degree or Eigenvector centralities as *not* central. This highlights the fact the semantics of the opinion measure is clearly unlike that of the other measures considered here.

Finally, we compare the computational costs of the measures. During the processing of the opinion centrality, the most expensive operation is the inversion of an $I \times I$ matrix (I being the number of vertices by layer). Therefore the time complexity associated to the opinion centrality is in $O(I^3)$. We do not have access to the algorithmic complexity for the other multiplex measures considered in this chapter, so we compare the measures empirically. Figure 8.3.b displays the processing times obtained when computing our opinion centrality, as well as the multiplex versions of the degree, PageRank and Eigenvector measures. We used a plain desktop PC (i5 3.00GHz quadcore processor with 16GB RAM). Note that both axes use a logarithmic scale. We did not include all measures for matters of readability, and because in- and out-degree behave like degree, whereas Hub, Authority and Katz performances are located in between Eigenvector and PageRank. The processing time for the degree is quite stable, as expected from this purely local measure. For the Eigenvector and PageRank measures, it increases exponentially with the number of vertices. The opinion measure also undergoes a very fast increase, but clearly slower than Eigenvector and PageRank. For the largest network, it is a matter of minutes. In terms of memory usage, the opinion centrality is also less expensive, as illustrated by the fact we could not process the Eigenvector, Authority, Hub, Katz and PageRank measures for the largest networks considered in this study, due to memory limits. Finally, we did not detect any effect of the opinion centrality parameters on its processing time.

8.3 Multiple Partitioning of Multiplex Networks

Together with vertex centrality measures, graph partitioning methods (also called community detection methods) are probably the most widespread tools when performing the descriptive analysis of a complex network [102]. Like already mentioned in Chapter 2, this task is traditionally defined informally, as the identification of mutually exclusive groups of vertices exhibiting strong internal cohesion but loosely connected to each other [103].

The literature contains 3 main approaches to partition multiplex graphs [141]: 1) merge the layers and apply a traditional partitioning method to the resulting uniplex graph, such as in [23]; 2) apply a uniplex method separately to each layer and merge the resulting partitions, such as in [24]; and 3) use a method specifically designed for multilayer graphs, which partitions the set of all vertices over all layers, such as in [174]. All three approaches are based on the assumption that one is looking for a *single* partition. In both first approaches, each cluster of the partition contains all instances of the same vertex over all layers: it can be considered as a consensual partition, fitting all layers at once. In the latter approach, a community does not necessarily span all layers (two instances of the same vertex can belong to different communities).

The work presented in this section was realized during the PhD of Nejat Arınık [D4], who was co-advised by Rosa Figueiredo and myself. It was the object of several oral communications [VL81, VL60, VL80, VL78], and the interested reader will find the technical details and a thorough discussion of our results in [VL4].

In this instance, we wanted to study the voting activity of the Members of the European Parliament (MEPs), based on the IYP dataset introduced in Chapter 7. We were interested not only in detecting groups of MEPs which would be cohesive in terms of votes, but also in identifying the different typical *voting behavior patterns* of the European Parliament (EP), i.e. the characteristic ways in which the MEP set is divided by these votes. The publicly available raw data are basically a table showing how each MEP voted at each roll-call. A convenient way of representing this type of system is to extract a vote similarity network for each roll-call, in which vertices represent MEPs, and (possibly weighted) edges represent how similarly two MEPs vote. Partitioning such a graph allows identifying the opposed factions. When dealing with several roll-calls as we do, the standard approach consists in averaging the edge weights over the series of roll-calls (e.g. [225, 233]), in order to get a single network representing the whole period. Looking for factions in this network amounts to applying Type 1 of the multiplex partitioning typology presented above (i.e.layer merge). This is what we did in Chapter 7. However, this averaging leads to some information loss due to the temporal integration performed on the raw data. We showed in the work described in Chapter 7 that this approach does not allow identifying clearly interpretable voting patterns for these data.

This is the reason why we proposed a new partitioning method, aiming at identifying not only the factions associated to each roll-call, but also the sets of roll-calls exhibiting similar voting patterns. For this purpose, we decided to keep the uniplex networks representing the roll-calls *separated*, in order to form a proper *multiplex* network. We then partition each layer separately, as in Type 2 of our multiplex partitioning typology. Afterwards, however, instead of merging all the resulting partitions, we identify groups of similar partitions which we then merge separately. In the end, for a single multiplex network, we get a *set* of partitions, instead of a *single* partition as in the traditional approaches mentioned before. Each one is assumed to represent a typical voting pattern of the EP. In this section, I briefly describe the method that we proposed (Section 8.3.1), before turning to the main results that we obtained on the EP dataset (Section 8.3.2).

8.3.1 Description of the Partitioning Method

I have to handle two distinct types of partitions, so I need to clarify my terminology first. I keep on calling *community structure* a partition of V obtained for a given layer G_ℓ. I reserve the term *clustering* to refer to a partition of the set of all community structures (i.e. when considering all layers). The subsets of community structures constituting a clustering are simply called *clusters*. Finally, the *characteristic community structure* of a cluster is a community structure that is representative of the cluster.

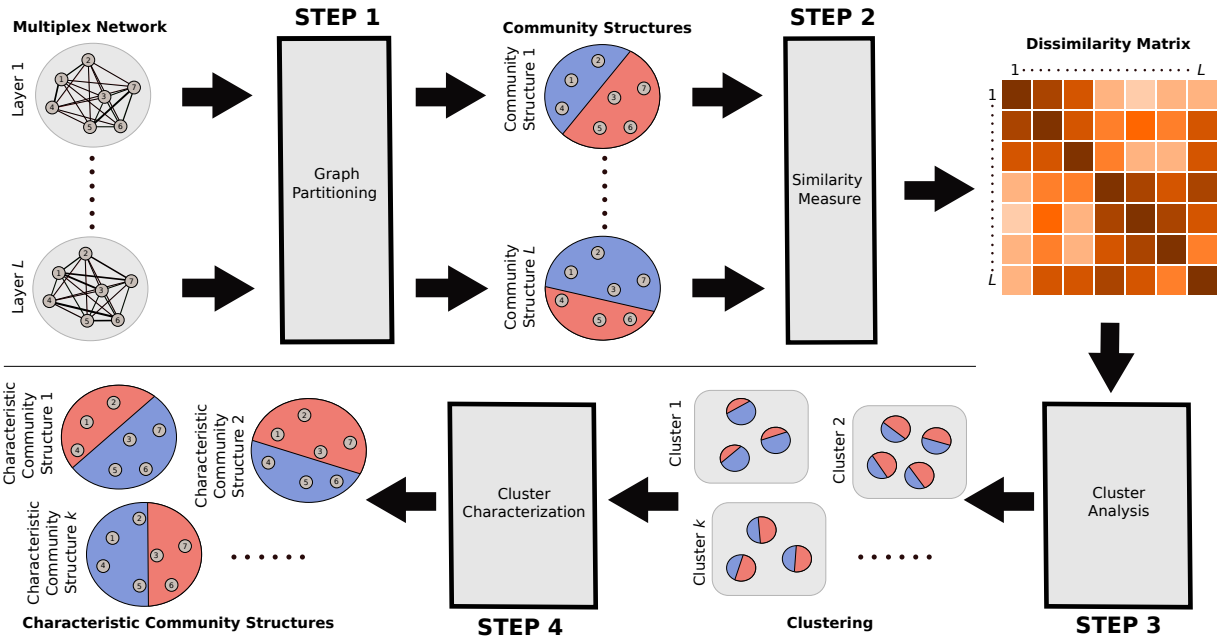

Figure 8.4: General workflow of the proposed partitioning method.

The goal of our method is to identify simultaneously the main types of community structures and the specific layers that match them. To this aim, we propose a four-stepped method, summarized in Figure 8.4.

Step1: Identifying the patterns The first step consists in separately partitioning the L layers of our multiplex network, in order to get as many community structures. For this purpose, any community detection method can be applied, as along as it outputs proper partitions of the vertex set V.

Step 2: Computing the Similarity Values The second step consists in computing the similarity between the community structures obtained at the preceding step. A number of measures have been defined to compare such partitions, each one possessing a specific behavior. Selecting the most appropriate measure is a challenge due to the difficulty to compare them efficiently and in a way that is relevant to application at hand. As a side-work, we have proposed a whole framework for this sole purpose in [VL1] (I do not describe it in this manuscript, as it is not directly related to feature-rich networks). In the context of the analysis of the EP data, we identified the Purity [165] as the most appropriate measure. It considers the maximal overlap between the clusters of the considered partitions. This is an asymmetric measure, i.e. its value can change if we switch the partitions, so it is customary to use its harmonic mean. Based on the selected measure, we compute a similarity value for each pair of community structures, and then build a dissimilarity matrix summarizing these comparisons.

Step 3: Performing the Clustering We now want to gather similar community structures together. For this purpose, we apply the k-medoids clustering method [137] to the previously obtained dissimilarity matrix. It is similar to the well-known k-means algorithm in the sense that it partitions the dataset in k clusters, while minimizing the distance between the members of each cluster and some center of the cluster. The difference is that in k-means, this center is an average value, whereas in k-medoids it is one of the actual data point from the dataset. It is generally used in place of k-means when one cannot perform the required average operation, which is our case (we cannot straightforwardly process an average community structure).

This method requires us to specify parameter k, but we do not know it in advance. In this situation, the standard approach is to use all possible values of k, from 2 to ℓ (where ℓ is the number of patterns), and assess the quality of the $\ell - 1$ resulting clusterings through some internal criterion such as the *Silhouette* measure [204]. In theory, the k value associated to the highest criterion value is the best candidate. However,

in practice, one possibly has to consider application-related factors to make his choice. For example, marginal improvements of the Silhouette are sometimes caused by the creation of singleton clusters, which generally do not bring much relevant information in terms of interpretation of the clustering. It is therefore necessary to study qualitatively how the clusters evolve with k to make an informed choice.

Step 4: Computing the Characteristic Community Structures We now have k clusters, each one containing a certain number of possibly different but supposedly very similar community structures. For each cluster, we want to compute a characteristic community structure representing the whole cluster, such that these small differences are smoothed. For this purpose, we use a similarity network-based approach, inspired by the work of Lancichinetti & Fortunato [149].

Based on a collection of partitions of the same set, they derive a consensual partition by first extracting a weighted similarity network, and then performing community detection in this network. The resulting communities correspond to consensual clusters, and the community structure is the consensual partition. Their network is built as follow: each node represents an element of the partitioned set, and the weight of a link is the proportion of partitions in which both connected nodes belong to the same part.

We experimented with this approach, and found out we obtain better results by using the following *signed* version. The weights are now the difference between the proportion of partitions putting both vertices in the same part, and the proportion of partitions putting them in different ones. We use the Ex-CC method (see Chapter 7 for more details) to partition the resulting signed graph, and finally obtain a partition corresponding to the characteristic community structure of the considered cluster.

8.3.2 Main Results

As mentioned before, we applied our partitioning method to the IYP dataset describing the voting activities of MEPs, which we already presented in Chapter 7. Our data [C7] and source code [S6] are both publicly available online. Like before, I focus on a subset of the data for matters of space and concision: the votes of the French MEPs regarding propositions related to agriculture in 2012–13. More information regarding the application and a more detailed interpretation and discussion of our results can be found in [VL4]. Another important point is that the vote similarity networks that we extract are *signed*. Our multiple partitioning method from Section 8.3.1 is generic enough, so the only specifics is that Step 1 must rely on a graph partitioning method able to handle signed graph, such as the ones we discuss in Chapter 7.

Figure 8.5 displays our main results from [VL4], as well as from our preceding work [VL29] already presented in Chapter 7, which we used as a baseline. The former are based on the multiple partitioning method described in Section 8.3.1, whereas the latter come from the traditional approach consisting in merging all layers before detecting the communities on the resulting single integrated graph. Each plot is organized similarly, and shall be read from the center to the periphery. The negative and positive edges are drawn at the center, in red and green, respectively. Next, the inner colored ring represents the vertices (MEPs), and these colors correspond to the detected communities (i.e. partition of the MEPs). If a MEP was often absent, he is ignored and appears in white. Finally, the outer ring shows the European political groups to which the MEPs belong. They are ordered according to the political spectrum, from left to right: GUE-NGL (communists, in red), G-EFA (environmentalists, in green), S&D (socialists, in pink), ALDE (liberals, in orange), EPP (conservative, in light blue), ECR (euroskeptiks, in dark blue), EFD (far-right, in purple) and NI (Non-Inscrits, i.e. far-right, in brown).

In this figure, Plot 8.5.a shows the names of the French MEPs for the considered period. Each one of the next 5 plots corresponds to one of the clusters identified at Step 3 of our method, and exhibits the characteristic community structure obtained for this cluster at Step 4. Plots 8.5.g and 8.5.h show the results obtained when applying the traditional layer-merging method, and using two distinct signed graph partition method to detect communities in the resulting integrated graph: CC (Correlation Clustering) and RCC (Relaxed Correlation Clustering), which are discussed in Chapter 7.

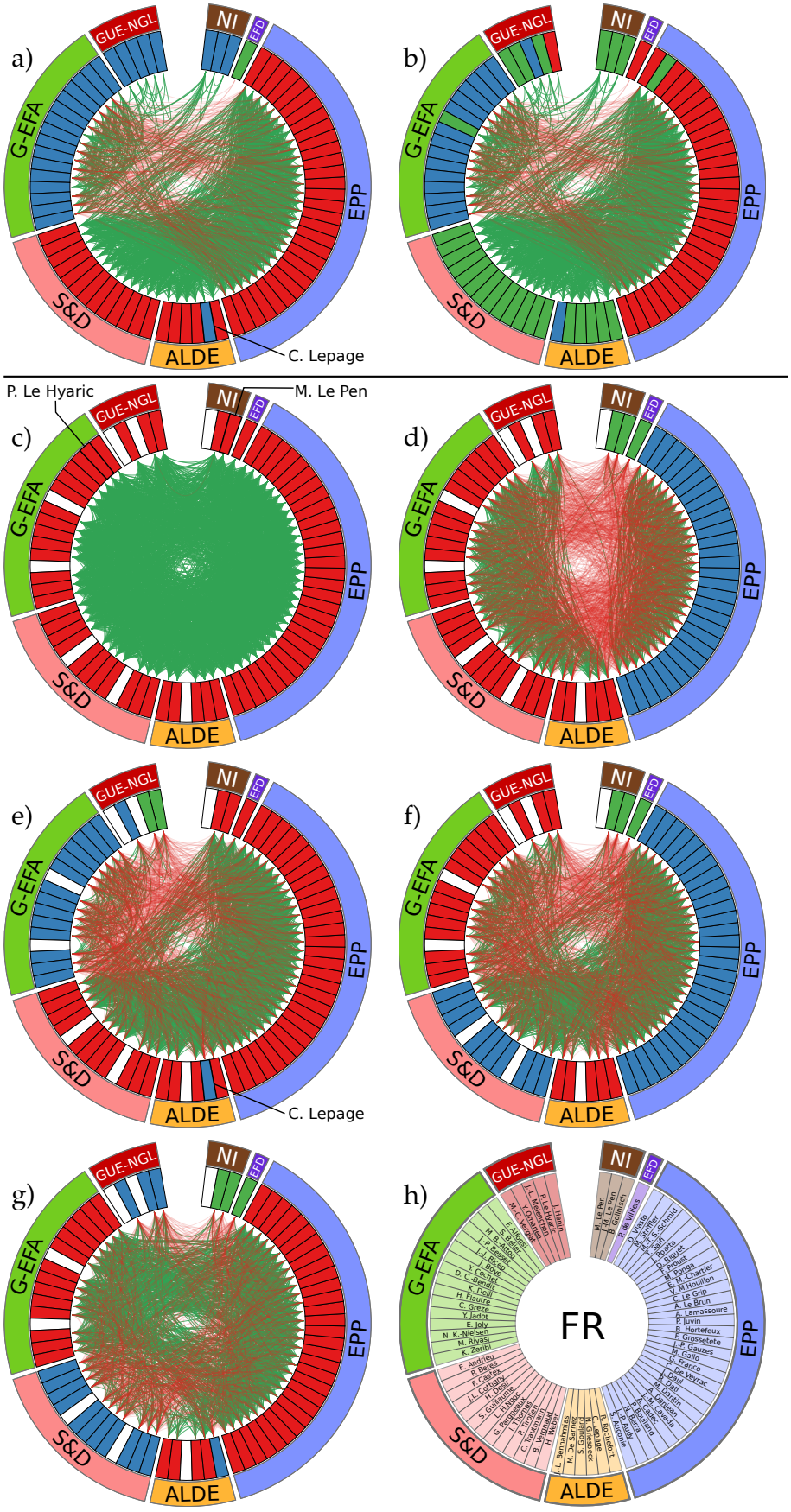

Figure 8.5: Communities detected for the French MEPs on agriculture-related roll-calls from 2012-13. Red and green lines at the center represent negative and positive edges, respectively (red edges are drawn on top of green ones in order to improve readability). Around the edges, each MEP is represented by a colored tile, whose color corresponds to the MEP's detected community. The outer ring represents the political groups at the EP. The top two plots show the baseline results, obtained in [VL29] on the same data, but through temporal integration, when solving CC (a) and RCC (b) (see Chapter 7). The bottom plots show the characteristic community structures obtained for the 5 clusters detected in [VL4] with $k = 5$. The last plot (h) shows the MEPs' names.

Baseline Plots 8.5.a and 8.5.b show that the integrated network is highly polarized, as it contains many negative edges and results in good partitions in terms of structural balance. As discussed in Chapter 7, solving CC and RCC on this network allowed identifying two antagonistic factions respectively led by the environmentalists (G-EFA) and the right conservatives (EPP), joined by some other groups or individual MEPs. The position of S&D and ALDE is interesting, because they belong to the right-wing faction according to CC, whereas they hold an intermediate position with RCC. When using this traditional approach, it was not possible to provide a solid explanation for this observation, or to discover in which context this happens, due to the temporal integration. But one can assume that these groups were sometimes voting like the left-wing faction, and sometimes like the right-wing one. One of the reasons why we proposed our multiple partitioning method was to check the validity of this assumption.

Unanimity The characteristic community structure corresponding to the first cluster detected by our multiple partitioning method is shown in Plot 8.5.c. It corresponds to a situation of unanimity, and represents the largest cluster with 100/232 of the roll-calls (43% of the layers), so we can assume that it represents the regular voting behavior in the considered voting context. The emergence of such a high level of agreement is completely hidden when considering only the integrated network. Traditional approaches studying voting networks usually discard unanimity situations as a preprocessing, but our method allows treating them just like any other voting patterns. All the other clusters correspond to community structures containing varying antagonistic factions.

EPP vs. the Rest Plot 8.5.d displays the characteristic community structure associated to the second cluster, which represents 34/232 roll-calls (15%). It opposes the right-wing conservative group (EPP) to the rest of the MEPs, while both Euroskeptic groups (EFD and NI) abstain. An examination of the content of the corresponding legislative documents, as well as of certain positioning documents produced by the EP groups, such as election manifestos and public letters, reveals that this voting behavior corresponds to EPP trying to block radical changes related to the CAP. This concerns in particular the capping of direct payments to farmers, a basic income conditioned on the implementation of certain EU rules, that constitutes a consequent budgetary item. For instance, the first amendment proposed by S&D, which matches the characteristic pattern, aims at setting a threshold of 200 k€ in order to decrease the support granted to large agriculture structures without affecting small- and middle-sized businesses. This change is first rejected by EPP, but a compromise is later found by raising the cap to 300 k€. The corresponding roll-call belongs to the next cluster, which therefore exhibits a much different voting pattern.

Environmentalists vs. the Rest The third cluster contains 74/232 roll-calls (32%) and its characteristic community structure is shown in Plot 8.5.e. The main difference with the previous one is that S&D and ALDE now side with EPP, leaving G-EFA alone. This is quite similar to the voting pattern identified through CC on the integrated network (Plot 8.5.a), except for the NI group and a few MEPs. In particular, Corinne Lepage appears in both cases as an environmentalist member of ALDE, as she is placed in the G-EFA faction. The roll-calls composing this cluster are mainly associated to amendments related to environmental aspects of agriculture, and most are proposed by G-EFA, sometimes in collaboration with C. Lepage.

Obviously, the singular position of G-EFA in this cluster is caused by its systematic opposition to the other groups on the texts associated to this cluster. However, one can distinguish two different situations. On the one hand, G-EFA tables amendments to enhance and complete the social and/or environmental regulations proposed in the amended text, and then vote in their favor. For instance, a legislative text was presented to enforce crop diversification: G-EFA proposed to add crop rotation in order to reduce the use of chemical fertilizer, but the other groups disagreed. On the other hand, G-EFA opposes amendments considered as not environmentally and/or socially progressive enough. For instance, they voted against an amendment proposed to trigger milk quotas only in case of severe market imbalance, as they considers that those quotas help avoiding overproduction, whereas all the other groups supported this amendment.

S&D/EPP vs. the Rest Plot 8.5.f represents the characteristic community structure of the fourth cluster, which corresponds to 18/232 roll-calls (8%). It contains a community formed by the far-left, environmentalist and liberal groups (GUE-NGL, G-EFA, ALDE), vs. another community containing the socialists and conservatives (S&D, EPP), while both Euroskeptical groups form an abstentionist faction. This constitutes a new type of pattern, different from all the others met until now, including in the baseline. In particular, it is is worth noticing that S&D and ALDE do not belong to the same community. Thus, if these groups alternatively side with left- and right-wing groups, as already assumed based on the baseline (and as illustrated before), our method shows that they do not always do so simultaneously.

It is worth stressing that groups may behave similarly, but for different reasons. This is not apparent in the identified community structures, and requires a deeper analysis of the documents voted during the concerned roll-calls. This is well illustrated by the issue of the gradual elimination of export refunds. These are subsidies granted for certain products (e.g. cereals, rice, sugar, etc.) that are exported outside the EU, and aiming at enabling EU exporters to better compete on world markets. ALDE opposed them because it wants to liberalize agriculture as much as possible and sees export subsidies as a cause of unfair competition. G-EFA was also against them, but because it considers that they do not favor fair producer prices and do not encourage product quality. S&D also criticized the continuation of export subsidies, but voted to keep them because eliminating them would weaken the EU's hand in worldwide trade negotiations. EPP was not against eliminating the export subsidies in general, but they want to keep them in case of crisis.

Unholy Alliance Finally, Plot 8.5.g shows the characteristic community structure of the fifth cluster, with the opposition between on the one hand a community gathering environmentalists and right-wing liberals and conservatives (G-EFA, ALDE, EPP), and on the other hand a community composed of the far-left and socialist groups (GUE-NGL and S&D), while the Euroskeptics abstain once again. These communities are surprising from a political standpoint, as they exhibit a somewhat unholy alliance between environmentalists and conservatives, whose views generally clash for agriculture matters. But the voting pattern is also surprising when considering the baseline, as it does not detect this alliance at all. The cluster contains only 6 roll-calls (2%), which shows that this situation does not happen often. As before, examining the concerned documents and debates reveals that these groups voted similarly in this specific context, but for very different reasons (detailed in [VL4]).

Concluding Comments The results obtained with our multiple partitioning method confirm in a more objective way the assumption we made based on the RCC community structure (Plot 8.5.b), and according to which S&D and ALDE sometimes vote like the left-wing groups (as in Plot 8.5.d) and sometimes like the right-wing ones (Plot 8.5.e). Our method additionally identifies the documents for which the EP adopts these two patterns: it turns out most of them are amendments to the same legislative propositions, in both of these clusters. But our method also shows that these two groups vote differently in a number of occasions (Plots 8.5.f and 8.5.g), a fact overlooked when using the traditional approach. Finally, it allows identifying the *Unholy Alliance* voting pattern, which had completely been overlooked by the traditional approach.

8.4 Conclusion and Perspectives

The goal of this chapter was to describe my works related to the analysis of multiplex networks. I first presented the outcome of my collaboration with Alexandre Reiffers: a vertex centrality measure based on a diffusion model. It was inspired by the way opinion can propagate when users interact simultaneously through several social media. Through an empirical assessment of the measure, I showed that its behavior was different from other multiplex centrality measures. In particular, high degree vertices are not necessarily considered as central, and low degree vertices can be seen as central if they allow a better control of the opinion diffusion.

Second, I summarized a multiple partitioning method for multiplex graphs, designed with Rosa Figueiredo and Nejat Arınık during his PhD. The opinion measure was designed in a top-down way as it can be viewed

as the multiplex generalization of a uniplex measure, whereas the partitioning method is the result of a bottom-up work, as it aims to solve a specific application problem related to European politics. Its particularity is that it finds both a partition of V, the vertex set, but also of the graph layers, such that layers associated to similar vertex partitions are grouped together.

My work on multiplex graphs continues, on different applications and data. There is first the collaboration with historians and geographers mentioned in Sections 2.4 and 6.4, aiming at studying various types of graphs extracted from a corpus of medieval deeds from Avignon, when it was the papal city. One of these graphs is a multiplex network of social interactions between real state owners and related persons, such as family and witnesses (cf. Figure 6.14.c). We can distinguish several types of interactions in this social network (familial, professional, ecclesiastic), which makes it a multiplex network. We plan to study the redundancy between the various layers, not only in terms of edge distribution, but also in terms of community structure. Along the graphs extracted from our corpus of deeds, we can also mention the so-called confront network, whose vertices are pieces of real estate and edges represent spatial proximity. We can connect this confront graph to the social network, as the deeds precisely aim at describing ownership. Once the social and confront networks are connected in this way, the whole can be considered as a multilayer network (not multiplex, as the vertices do not match), which we plan to study. In particular, an interesting aspect will be to compare the distance between people on the confront layer, based on their residency, and on the social layer, in order to assess whether the two are correlated.

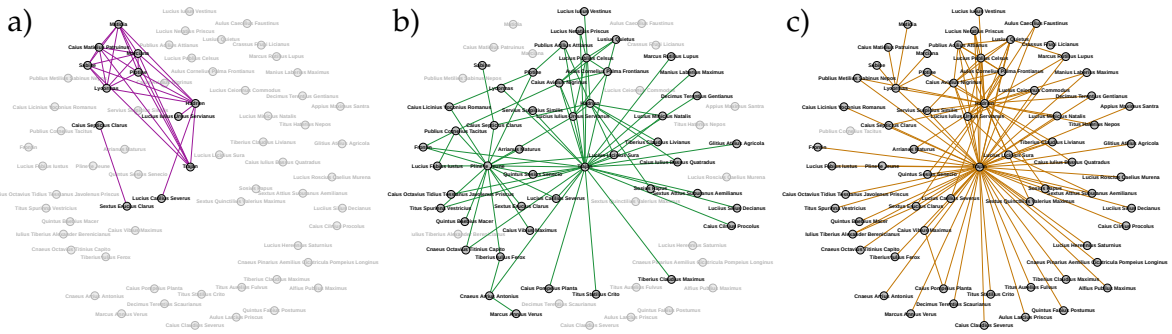

Figure 8.6: Multiplex social network describing the entourage of Roman emperor Trajan. Each plot represents a type of relationship: familial (a), friendly (b), and professional (c).

Finally, I also participate in another ongoing multidisciplinary work that deals with applied aspects of multiplex network analysis, and that I already mentioned in Sections 2.4 and 4.5. This collaboration with historians Gaëtane Vallet and Catherine Wolff aims at studying the entourage of Roman emperor Trajan. A first purely descriptive part is currently under review [VL88], based on a multiplex network extracted from historical sources and containing three layers shown in Figure 8.6. It consists in characterizing the position of the persons in Trajan's entourage according to several centrality measures, and discussing the historical relevance of the results. There is no methodological contribution here, from the perspective of complex network analysis, but we show how network analysis tools can be leveraged to answer concrete historical questions, regarding for instance the nature of the relation between Trajan and the Senate. The rest of this project will focus more on graph partitioning, as well as the possible integration of signed edges in certain layers. The joint exploitation of the multiplex and signed aspects of the graph will be the occasion to propose new analysis methods.

9

Conclusion and Perspectives

In this manuscript, I summarized the work that I conducted in the domain of feature-rich network extraction and analysis, since my PhD. Most of this work was the outcome of several collaborations that started through the years.

The first part of the manuscript concerned vertex-attributed networks, i.e. graphs whose vertices are described by additional attributes. I presented the ground survey that we conducted at Galatasaray Üniversitesi on a population of students, and our comparative study of the groups identified using the resulting attributes vs. the identified social relations. I also described the classification work based on attributed networks conducted at the LIA, regarding two tasks: the prediction of offline influence based on social media using Twitter data, and the automatic detection of abusive messages in online chats using the content of the exchanged messages and the structure of the conversations.

The second part of the manuscript was dedicated to dynamic networks, i.e. multilayer networks in which each layer represents a step in the evolution of the network. I first explained how, when I was in Istanbul, we leveraged sequential pattern mining to design two methods allowing to characterize the evolution of dynamic networks: the first was local and focused on events occurring at the level of vertex neighborhoods, whereas the second aimed at describing whole communities. I then presented a work conducted in Avignon, and aiming at generating video summaries of TV series. The method that we proposed largely relies on the use of a dynamic network of character interactions in order to perform plot modeling.

Finally, the third part of the manuscript gathered works dealing with spatial, signed, and multiplex networks, all conducted in Avignon. Regarding spatial networks, I presented the experimental study that we did on the behavior of the Straightness measure in various types of geometric networks, and more particularly orb-webs. I also explained the method that I proposed to compute this measure more efficiently. Regarding signed graphs, I described the work we did around the correlation clustering problem, the most widespread signed graph partitioning problem. Besides some experimental work aiming at assessing the behavior of methods solving this problem in various complex, we also studied the problem itself, and characterized its space of optimal solution, thereby opening a number of perspectives. Regarding the multiplex networks, I presented two methods. The first is a centrality measure based on a model of opinion diffusion in multiple social media. The second is a partitioning method allowing to detect several typical partitions of the vertex set, and to identify the network layers for which they are the most relevant.

In the conclusion of each chapter of the manuscript, I have already identified the ongoing works related to the chapter's topic, as well as the direct perspectives. Therefore, I will not discuss them again here. Instead, I will explain how I would like to channel my research activity from a more general point of view. This involves developing methods allowing to handle heterogeneous data coming from various sources (Section 9.1) and applying them in collaboration with researchers from various fields in order to find interesting problems to solve (Section 9.2). Finally, I present the recently started DeCoMaP ANR project, which perfectly illustrates both of these aspects (Section 9.3).

9.1 Multimodality

By multimodality, I mean here taking advantage of several types of data simultaneously, in order to get a better model or to improve the results on the task at hand. Multimodality was at the core of my PhD topic, which consisted in proposing a framework tailored to model large-scale networks of anatomically connected cerebral regions, and to validate these models by comparison with neuroimaging data obtained on real subjects. We chose to represent the activity at the level of a cerebral region using two complementary modalities: one for the amplitude of this activation, and the other for its semantics, corresponding to the pattern of neurons activated at a lower anatomical level. The framework that I proposed therefore had to be able to handle and combine both aspects of the cerebral information.

When my research interests turned to data mining and complex network analysis, after my PhD, and particularly when I arrived at Galatasaray, I realized that not only are graphs great for a number of tasks such as visualization, descriptive analysis and prediction, but they are also a very good modeling tool when one wants to deal with multimodal data. Indeed, by construction they allow representing relational information (i.e. relationships and interactions), and extending them to include individual information (i.e. characterizing individual vertices or edges) is quite straightforward. However, it is not so easy to generalize to such feature-rich graphs the various tools designed to analyze plain graphs. To my opinion, this is what makes this research direction very interesting and promising: there is much to do regarding the design of methods able to take into account the specifics of certain types of features. This is even more the case when considering the joint inclusion of several such types (e.g. spatial dynamic multiplex network). Most of my ongoing works, listed throughout this manuscript, use graphs that include several types of features.

Another very interesting aspect of multimodality and graphs is that including more information in our models is likely to reveal new problems that were not relevant when only dealing with plain graphs. For instance, using dynamic networks rises the question of identifying an appropriate time scale to define the duration of a network time slice. The relational information can be leveraged to answer this question, which is otherwise irrelevant to static graphs. Another point that I would like to explore is the informative value of the additional modalities. Extracting feature-rich networks generally requires more effort, in one way or another: is this additional cost worth it? This is a point on which I already worked, e.g. I explored whether using negative edges improved signed graph partitioning (see Chapter 7), and whether considering content improved performance on a supervised classification task of conversational networks (see Chapter 3).

From the methodological point of view, recent developments in graph embeddings and representation learning offer very promising perspectives to integrate graphs and other modalities. Graph embeddings are vector-based representations of graphs or parts of graphs (vertices, edges, subgraphs) obtained either through predefined computations often performed on the adjacency matrix, such as taking the spectrum of the graph Laplacian [147], or automatically learned, generally through neural approaches [117]. The latter are generally direct adaptations of Natural Language Processing (NLP) methods [159] (at least the earliest ones). The primary interest of such representations is to allow applying classic data mining tools *directly*, without any adaptation to graphs and their constraints. For instance, instead of applying a community detection method, which was specifically designed to partition graphs, one could compute first a graph embedding, and then perform a traditional cluster analysis, to get similar results [218]. But graph embeddings can also be leveraged to integrate several modalities, by learning a joint representation involving all these modalities at once, e.g. time and attributes in [235]. There is much interesting work to do here, to propose appropriates representations of various combinations of modalities. Another important aspect is that such representations can be trained either in a supervised way, relative to a task of interest, or in an unsupervised (or even self-supervised) way, in order to obtain a generic, task-independent representation. The former generally get better results, but the latter are trained once and for all. Recent NLP methods such as BERT [81] are currently being adapted to graphs in order to explore this idea [240].

9.2 Interdisciplinarity

I worked in an interdisciplinary context during my PhD, and since then I have always tried to conduct a part of my research activity in collaboration with researchers from other fields. As I mentioned in the introduction (Chapter 1), the laboratory in which I did my PhD included scholars from most fields related to cognition: neurologists, psychologists, physicians, statisticians, linguists, etc. I directly interacted with them, especially neuro-psychologists (see for instance [VL19]), and I really enjoyed these collaborations, as they involved acquiring knowledge from neighboring fields, and opened new perspectives to my work. This experience also taught me that interdisciplinary work is not easy though, and requires patience and efforts. When I arrived in Istanbul, I pursued such collaborations with colleagues from Political Science and Business Science. The wider gap between our disciplines made it harder to communicate, but the work was certainly not less interesting, on the contrary.

I was hired by Avignon Université in part thanks to my experience and partiality for interdisciplinarity. The university created a federation of laboratories called Agorantic[19] (FR 3621) in 2012, in order to promote works between researchers from different fields. Since then, this federation helped to start a number of collaborations between, among others, human science researchers on one side and computer scientists or mathematicians on the other side. Through this means, when I arrived in Avignon, I started working with geographers (Chapter 6) and political scientists (e.g. [VL67]). I want to keep on conducting such works in the future, for the following reasons.

First, collaborating with other fields is a good way to get access to new applications, and therefore renew the interest of my research work in general. From this perspective, Agorantic is a great tool which I want to keep on supporting. Other perspectives, more connected to my PhD work, also include the possibility to collaborate through the *Institute of Language, Communication, and the Brain*[20] (ILCB), another group of laboratories that includes the LIA. It focuses on cognitive aspects of natural language. Second, I invested a lot of time through the years, to get the skills necessary to properly work in an interdisciplinary context, communicate with scientists from other fields, acquire some of the foreign concepts necessary to this communication, take into account my misconceptions and those of people from other domains. This investment is now paying off, and makes my interdisciplinary work more efficient. Third, I am convinced that interdisciplinary work is a great way to get access to completely new problems, otherwise difficult or even impossible to come up with while staying in one's own domain. For instance, the needs and requirements of social scientists often necessitate to take into account aspects that would otherwise be discarded by computer scientists, because regarded as unimportant or too complicated to handle. The other way round, methods developed specifically for a given problem are likely to be useful to solve other problems in completely different application settings. Fourth and finally, I think that complex problems require interdisciplinarity: it is generally not possible to get a full grasp of the full situation using a single perspective. Involving several disciplines is more likely to provide complementary views, thereby helping to solve the problem.

More concretely, I plan to keep on exploring interdisciplinary collaborations with social science researchers through problems related to NLP, information retrieval (IR) and graph models. Graphs are a great medium to conduct such collaboration, as they are already used by many social scientists, who are familiar with a number of related concepts. Even when it is not the case, graphs constitute an intuitive visual support, that can be sufficient to get a good understanding of the data, provided they are appropriately interpreted. Many social sciences are interested in relationships, under one form or another, and the data handled by social scientists can often be leveraged to extract a variety of interesting graphs. Translating the problematics of social scientists into graph-based concepts is often a good source of interesting computer science problems to solve, as illustrated by the various ongoing project that I mentioned in the chapter conclusions of this manuscript. A large part of the data handled by social scientists take the form of text, and processing large corpora requires a minimum of automation. This makes NLP and IR tools particularly important to extract relevant information from big datasets. Several NLP tasks such as coreference resolution are not satisfactory solved yet, especially for the French language. I plan to work on these aspects, by leveraging recent NLP

[19] https://agorantic.univ-avignon.fr/en/
[20] https://www.ilcb.fr/

representation methods that I already mentioned in Section 9.1, such as (Camem)BERT and its variants, which are particularly promising [133]. This part of my ongoing and future work is directly connected to the project described in the conclusion of Chapter 3 regarding the extraction of attributed networks from biographical notices and journal articles.

9.3 DeCoMaP ANR Project

DeCoMaP (*Détection de la Corruption dans les Marchés Publics* – Detecting corruption in public procurement markets) is a research project funded by the ANR (French research agency) that started recently, and that I contributed to design (cf. Appendix A.6.2). It illustrates particularly well the directions that I want to explore in my future research activity, as described in Sections 9.1 and 9.2. The expression *Public Procurement* refers to governments' purchasing activities of goods, services, and construction of public works, which comprises large shares of government budgets (e.g. 10% of the GDP in France[21]). Yet, public procurement processes are open to the use of public resources for different interests other than the public good, involving corrupt transfers between state officials and private sector firms [48]. The World Bank has estimated that roughly USD 1.5 trillion in public contract awards are influenced by corruption [191], and the volume of bribes exchanging hands for public sector procurement alone is about USD 200 billion per year. In order to fight this type of corruption, France implemented a European directive enforcing the publication of open data describing the whole public contracting process. According to the OECD, such good practices help exposing governmental misconduct, and ease judicial enforcement as well as media and citizen monitoring [186]. The goal of the DeCoMaP project is to leverage these open data to design automatic methods able to perform corruption and fraud prediction.

The project is clearly interdisciplinary, as it involves researchers from Computer Science, Economy, Law, and Political Science. It relies on some preliminary work conducted by Pierre-Henri Morand (Economy), Rosa Figueiredo (Computer Science) and myself, in the framework of Agorantic. Moreover, it aims at producing a practical tool, that will be used by NGOs and governmental agencies interested in fighting fraud and corruption. The main data source is the BOAMP[22] (*Bulletin Officiel des Annonces des Marchés Publics*, the French official publisher for public procurement markets), which gives us access to a very large collection of invitation to tender and the result of the bidding process, provides us with several modalities. First, each entity in this dataset is described by a number of fields corresponding to what I called individual information in this manuscript, and that can be modeled as vertex attributes in a graph. Second, each completed bid allows establishing a commercial relationship between a supplier and a client entities, which constitutes a form of relational information (or in other words, the structure of the graph). These relationships can be characterized in various ways in terms of how healthy they are, from the point of view of fraud and corruption [99]. We plan to model this using signed weights, which brings two additional modalities. Each call for bids and tender decision has a precise date in this dataset, therefore time constitutes the fifth modality. Most entities involved in the considered commercial exchanges are private companies or public institutions, which can be localized in a French city, thereby providing a spatial dimension constituting the sixth modality. In addition, during the second stage of DeCoMaP, we plan to expand our BOAMP-based database using other sources, in particular the BRÉF. As I explained in Chapter 4, this database that I participated to constitute for another project is a comprehensive description of the elected offices occupied by French representatives. Our objective there is to add individuals to the model, as they can help identifying hidden connections between public and private organizations (e.g. a municipal councelor being the director of some local company providing services to the same municipality). Merging such secondary sources into our dataset would add another modality to the network, making it multipartite.

In the course of DeCoMaP, we plan to design graph partitioning and link prediction methods able to use some, or even all of the modalities that I listed. We will adopt two general approaches: define graph-specific methods by combining, generalizing and extending existing approaches; and experiment with vector representations of these graphs, in order to leverage traditional data mining tools. An important part of the project is to

[21] https://www.economie.gouv.fr/daj/observatoire-economique-commande-publique
[22] https://www.boamp.fr/

provide guidelines aiming at improving the quality of open public datasets from the perspective of fraud detection. For this reason, the results provided by our tools will have to be explainable, which constitutes a very interesting and challenging constraint.

Appendix

A

Curriculum Vitæ

A.1 General Information

Here is a brief summary of my university career. I started my undergraduate studies at the *Université de Pau et des Pays de l'Adour*, where I did a *DEUG* (entry level diploma) in Mathematics and Computer Science (1995–1997), and a *Licence* in Computer Science (BSc, 1997–1998). I then moved to Toulouse for my graduate studies which took place at the *Université Paul Sabatier –– Toulouse III*, and did a *Maîtrise* in Computer Science (1998–1999), a *DEA* in Artificial Intelligence (MSc) (1999-2000), and a PhD in Computer Science and Artificial Intelligence (2000–2003).

My research topic during both DEA and PhD concerned Computational Neurosciences, and focused on modeling the representation and processing of cerebral information. My PhD was funded for three years by an *Allocation Ministérielle* (grant from the French Ministry of Education), I and was also a *Moniteur* (junior lecturer) at the same university in Toulouse.

The end of my PhD was funded through a position of part-time *ATER* (lecturer), which I hold for two years (2003–2005). This second year gave me the opportunity to switch to a different research topic: Data Mining. Moreover, I took advantage of these two years to launch a startup with a few other PhD students.

In 2005, after my second year as an *ATER*, I was hired as an Assistant Professor in Computer Science by the Galatasaray Üniversitesi, located in Istanbul (Turkey). This is a Turkish-French public university funded by both countries, with classes taking place in Turkish, French and English. My work was similar to that of a *Maître de Conférences* in the French university system, with the important difference that my teaching load was 280 hours/year (approximately four classes by semester), instead of 192 hours/year in France.

When I arrived at Galatasaray, I went on with my new research topic, specializing further by focusing on data mining *in complex networks*, which is my current main domain of activity. Due to this topical change and my significant teaching load, which included the creation of nine distinct classes from scratch, I did not publish anything from 2005 to 2008.

In 2014, I was hired by *Avignon Univerité* (then *Université d'Avignon et des Pays de Vaucluse*), as an Associate Professor in Computer Science. Research-wise, I joined the *Laboratoire Informatique d'Avignon* (LIA, the Computer Science laboratory of *Avignon Université*). Since then, I have been teaching Computer Science-related classes, mainly in the Computer Science diplomas awarded by the *CERI* (Computer Science department of *Avignon Université*), but also in the Geomatics programme of the Geography department and the Digital Management programme of the Law & Economics department.

A.1.1 Professional Experience

2014–present *Avignon, France*	**Associate Professor**, Avignon Université
2005–2014 *Istanbul, Turkey*	**Assistant Professor**, Galatasaray Üniversitesi French-speaking university depending on both Turkish and French states
2003–2006 *Toulouse, France*	**Technical consultant and co-founder**, Personnalité Numérique SAS Internet start-up, developed a remote/portable virtual hard-drive
2003–2005 *Toulouse, France*	**Lecturer**, Université Paul-Sabatier Part-time position
2000–2003 *Toulouse, France*	**Junior Lecturer**, Université Paul-Sabatier

A.1.2 Education

2000–2003
Toulouse, France

Doctor of Philosophy (PhD), Université Paul-Sabatier – Toulouse III

- Funded by a a French national research fellowship
- Specialty Field: Computer Science & Artificial Intelligence
- Thesis: 'Réseaux causaux probabilistes à grande échelle : un nouveau formalisme pour la modélisation du traitement de l'information cérébrale (Large scale probabilistic causal networks: a new formalism for the modeling of cerebral information processing)' [VL89]
- Defense: Toulouse, 18 December 2003
- Jury :
  - *President*: Claudette Cayrol, PR, Université Paul-Sabatier, Toulouse
  - *Reviewer*: Salem Benferhat, PR, Université d'Artois, Lens
  - *Reviewer*: José Miguel Bernardo, PR, Universitat València, Spain
  - *Examiner*: Emmanuel Guigon, CR CNRS, Université Pierre & Marie Curie, Paris
  - *Examiner*: Henri Prade, DR CNRS, Université Paul-Sabatier, Toulouse
  - *Advisor*: Josette Pastor, IR INSERM, HDR, Unité 455, Toulouse
  - *Guest*: Pierre Celsis, DR INSERM, Unité 455, Toulouse

1999–2000
Toulouse, France

Diplôme d'études approfondies / Master of Science (DEA / MSc), Université Paul-Sabatier – Toulouse III

- Speciality Field: Artificial Intelligence
- Thesis: 'Réseaux causaux : quelle approche pour le cerveau ? (Causal networks: Which approach to model the brain?)' [VL90]
- Advisor: Josette Pastor, IR INSERM, HDR, Unité 455, Toulouse

1998–1999
Toulouse

Maîtrise, Université Paul-Sabatier – Toulouse III

- Specialty Field: Computer Science

1997–1998
Pau, France

Licence / Bachelor of Science (BSc), Université de Pau et des pays de l'Adour

- Specialty Field: Computer Science

1995–1997
Pau, France

Diplôme d'études universitaires générales (DEUG), Université de Pau et des pays de l'Adour

- Specialty Field: MIAS (Mathematics, Computer Science & Physics)

A.2 Teaching Activity

This section summarizes my teaching activity. I first briefly describe the classes I taught and the teaching-related tasks that I performed at the three institutions for which I worked (Sections A.2.1, A.2.2, and A.2.3), before providing a synthetic view of my teaching work (Section A.2.4).

A.2.1 In Toulouse

I started working as a lecturer from the beginning of my PhD, not only at my university (Université Paul Sabatier – Section A.2.1.1), but also in a professional institute related to this university (IUP TMM – Section A.2.1.1) and focusing on medical technologies and methods. Teaching at both institutions allowed me to work in very different contexts, regarding both students and topics.

A.2.1.1 Université Paul-Sabatier – Toulouse III

At Université Paul-Sabatier, I first taught to the first year students. Due to the size of the university, the pedagogical team was large, with several tens of persons. I started with *Caml Programming* lab sessions (TP) for to Computer Science and Mathematics students (DEUG MIAS, then CIMP). I could then teach the tutorials (TD) associated to this class, then the tutorials of second year corresponding to the follow-up class, *Advanced Caml Programming*.

During my second year of PhD, I gave tutorials (TD) and lab sessions (TP) of *Pascal Programming* to first year Biology students (DEUG SVT). Computer Science students are generally interested by this type of class, but it is not necessarily the case for biology students. Therefore, I had to adjust my pedagogical approach to this public characterized by different motivations. On the same note I also gave *Algorithms* tutorials (TD) third year engineering students (Licence SDI – Sciences de l'ingénieur).

Besides programming and algorithms, I also gave tutorials in *Logic & Boolean Algebra* to first year Mathematics and Computer Science students (Licence CIMP). Finally, I took part in the teaching of *Productivity Suite Use* to first year Mathematics, Computer Science, and Biology students (Licences CIMP & SVT).

For all these classes, the pedagogical material was provided by the professors in charge, in order to get a relatively uniform teaching over the large team of lecturers. The lecturers giving the lab sessions and tutorials were in charge of their evaluation though, which required writing the examination subjects.

A.2.1.2 IUP TMM – Techniques et Méthodes Médicales

The teaching context was very different at the IUP TMM, in which I started teaching in my first year of PhD. First, promotions were very small, with only a few tens of students, which could all fit in the same room. Moreover, the goal of this training course was to become an engineer specialized in medical-related problems. Therefore, Computer Science was not the central part of the curriculum, but it still had an important role due to the development of bioinformatics, medical imaging, and all applications related storing and mining medical data.

I gave lab sessions on several different topics: *C Programming* (second year), *Productivity Suite Use* (third year), *Data Base Management* (third year), and *Artificial Intelligence* (third year). Starting from my third year of PhD, I also gave the *Artificial Intelligence* lectures. For all the lab sessions, I was in charge of the examinations, and for the lectures I was in charge of organizing the whole class. This allowed me to gather some experience regarding class preparation, student assessment, but also public speaking and student interaction.

A.2.2 In Istanbul

The second part of my career took place in Istanbul, at the Galatasaray Üniversitesi. It is a Turkish public institution, founded in 1992 through a bilateral agreement with the French state. France participates in the funding of the university, and in return teaching is mainly done in French. Indeed, Turkey has a tradition of bilingual teaching in highschool, resulting in a sufficient number of potential French-speaking students to supply a small university. Galatasaray Üniversitesi includes an engineering faculty, whose first two years are organized like French preparatory schools. I taught in the Computer Engineering and Industrial Engineering departments of this faculty, as well as in the Mathematics departments, which was created later. I was hired mainly for two reasons: 1) set up certain classes that were missing from the Computer Science Lisans programme (a four-year diploma similar to a Bachelor of Science); and 2) help starting a new Master's degree programme in Computer Science.

A.2.2.1 Classes Taught

When I arrived, I took up some existing classes: *Introduction to Algorithms & C Programming* (second year), *Advanced Algorithms & C Programming* (second year), and *Operating Systems* (third year). I started updating both former classes in collaboration with the mathematics professor that were in charge of them until then. I completely redesigned the third class, because the teacher previously in charge had left the university. I also created several new classes from scratch: *Artificial Intelligence* (fourth year), *Data Mining* (fifth year) and *Web Programming* (fifth year). My teaching duties were changed later, which led me to create a *Distributed Systems* class from scratch (fourth year). I also had to set up an *Introduction to Operating Systems* class (second year), and I participated in more general an *Introduction to Computer Science* class (first year). Finally, I later

could create a new master class directly connected to my research topics, and focusing on *Complex Network Analysis*.

For all these classes, I was in charge of creating the pedagogical material (assignments, lab sessions, tutorials, lecture notes, examinations), which required a significant amount of work. For algorithms and C programming, in particular, I produced a number of documents in collaboration with my colleague Damien Berthet, which we later collected in a series of three books now publicly available on HAL [VL91–VL93]. We had developed a pedagogical approach based on the introduction of algorithmic concepts through graphics-oriented tasks. For instance, the notion of recursive function can be presented using fractal drawing. In practice, we used the SDL[23] (Simple DirectMedia Layer) library, which is open-source and compatible with the C langage.

I also proposed various assignments when this was relevant to the concerned class. For the *Artificial Intelligence* class, in particular, I set up a project based on the classic Bomberman[24] video game. I first implemented a Java version of this game [S17], under an open-source license and containing an API designed to allow the implementation of intelligence agents controlling the game players. Groups of two or three students had to implement their own agent. At the end of the term, these were fighting each other during a public tournament, in order to assess and compare them. The recreational aspect of the topic and the presence of this tournament allowed me to organize this assignment during seven consecutive years. Videos showing the popularity of this project are available on my YouTube channel[25].

A.2.2.2 Teaching Context

Galatasaray students were very different from the students I had taught to in France. Promotions were small, with only around thirty students, all selected through a national examination. Although they all were French-speaking, their linguistic skills were heterogeneous, and language is a significant hurdle for which I had to adjust my pedagogical method, for instance by using a different linguistic register and by speaking slower. Master classes were open to students coming from all Turkish universities, most of them not speaking French, which is why they were given in English.

I was in charge of all the classes that I gave in Galatasaray, including lectures, lab sessions, assignments and examinations. I was completely free to define their content and the way the examinations were conducted. Therefore, I could acquire more experience in class design and management. I also had to interact with a very different administrative and academical system, cultural differences among students and colleagues, and different work methods. Another important difference with France is the amount of work, as I contractually had to give 280 effective hours of class by academic year. Moreover, the examination system required to organize up to six different tests for a given class. With at least three classes by term, this means 36 tests to write and correct for a single year.

A.2.2.3 Administrative Duties

Galatasaray students could apply for various funding schemes to go study in France. Some were funded by the French state, whereas others depended on French companies installed in Turkey. I took part in the internal process consisting in selecting the students the university would propose for this various fellowships. This included participating in several committees formed with other professors, auditioning students applying to these fellowships, and most of all helping the students from my faculty to constitute their application file. This last part required to advise them regarding their curriculum choices, explain them the administrative aspects of the procedure, assist them in the elaboration of their professional project, and help them contacting the heads of their targeted master's degree.

Another aspect of my teaching-related administrative work at Galatasaray was to set up and maintain a course management system. I first installed it to deal with the needs of my own classes, but it was soon

[23] https://www.libsdl.org/
[24] https://en.wikipedia.org/wiki/Bomberman
[25] http://www.youtube.com/watch?v=oFOLmUG_M2g

adopted by the colleagues of my department, then those of my faculty, and finally other professors from the rest of the university, especially the very large FLE department (French as a Foreign Language). After a few years, I was awarded a partial time-release in order to make this platform, as well as my involvement in its administration, more official.

A.2.3 In Avignon

A.2.3.1 Classes taught

Starting from 2014, I have been teaching at Avignon Université, almost only at the Computer Science Department (CERI – Centre d'Enseignement et de Recherche en Informatique), and mainly at the master level. In the Software Engineering specialty (ILSEN – Ingénierie du logiciel pour la société numérique), I participated in lab sessions from classes related to programming and algorithms: *Object Modeling & UML* (third year), *Application Servers* (fourth year). I am in charge of the *Programming Project* class (third year). I also have been giving lectures and lab sessions for classes more related to my own research topics: *Data Mining* and *Information Retrieval & Indecation* (both in fifth year). When the Artificial Intelligence specialty was created, I started an *Unsupervised Learning* class (fourth year).

In the Computer Network specialty (RISM – Réseaux informatiques et systèmes mobiles), I was at first in charge of a class also related to my research: *Complex Network Analysis* (fifth year). This specialty then took a more security-oriented orientation (SICOM – Systèmes Informatiques Communicants : réseaux, services et sécurité), and I participated to the *Advanced Security* class (fifth year).

I also punctually intervene in other diplomas than the Computer Science ones. I give a short class every year to students from the Geomatics Master (Géoter – Géomatique et conduite de projets territoriaux). It is an introduction to the analysis of complex networks focusing on *Centrality & Community Detection*). I also give a class introducing *Complex Networks Analysis* every few years to PhD students from the Science Faculty. Finally, I give a *Business Information Retrieval* class in the *Digital Gouvernance* Master (fourth year).

A.2.3.2 Teaching Context

In Avignon, the promotion size ranges from 150 to 80 students depending on the considered year, so the number of professors involved in a class is larger than what I experienced in Istanbul but larger than in Toulouse. The audience corresponds to local students, Avignon Université being a community-based institution. Unlike Galatasaray Üniversitesi, students are not selected, so their skills are more heterogeneous. Students from the Computer Science department are interested in the most concrete aspects of Computer Science, in particular programming, and including theoretical elements in the class curriculum is sometimes a challenge. Some of the classes I give (*Unsupervised Learning*, *Information Retrieval*, *Data Mining*, *Complex Network Analysis*) lend themselves to this, and it is very rewarding to manage to get a positive response from the students on these aspects.

The classes I give in other faculties (Geography, Economics-Law) require to adapt my teaching methods to a very different audience not able to program, and sometimes even not at ease with computers. It is about finding an appropriate balance between being too technical and loosing the students' attention, and being to superficial to provide them with interesting and useful content. This type of class also allowed me to learn how to better use application aspects in the class, and even to build a class depending on the application domain, such as business information retrieval.

A.2.3.3 Administrative Duties

Since my arrival at Avignon Université, I have been participating in many administrative activities related too teaching, including: tutoring third year students during their short internships, advising fourth year students during their year-long project, advising fifth year students during their end of studies work, tutoring students in part-time training, participating in pedagogical committees and juries.

The Computer Science department receives thousands of applications from foreign students, through the CampusFrance service. I take my part in the processing of these files. From 2015 to 2017, I was in charge of the University Diploma in Computer Science (DU d'informatique). Since 2019, I have been the head of the newly created Artificial Intelligence Master's degree.

A.2.4 Synthetic View

To summarize my teaching career, I would stress that I worked in very different contexts in terms of class topic (ranging from practical to theoretical classes), academic level (from first to fifth year, and even doctoral students), language (French, French-speaking and English-speaking students), diploma (computer scientists, mathematicians, biologists, physicists, engineers, geographers), scale (promotions ranging from a few students to several hundreds).

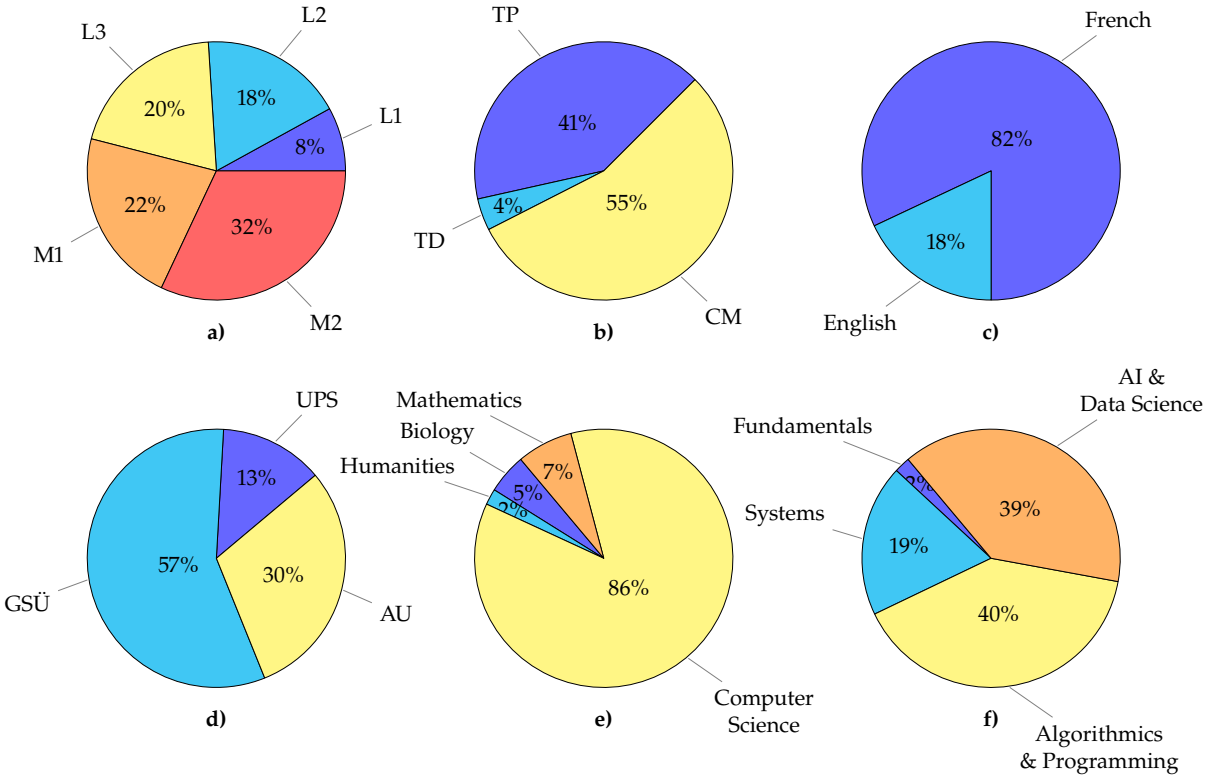

Figure A.1: Distribution of the teaching hours depending on: a) the academic year; b) the type of class; c) the teaching language; d) the institution; e) the student diploma; and f) the class topic.

Section A.1 lists all the classes that I taught in the three different institutions I worked for, with the number of taught hours. As is customary in the French university system, I distinguish between lab sessions (TP, *Travaux pratiques*), tutorials (TD, *travaux dirigés*) and lectures (CM, *cours magistraux*). The last column of the table corresponds to the equivalent total time expressed in terms of tutorial hours. The standard yearly amount of teaching time in the French system is 192 TD hours. Note that the standard at Galatasaray Üniversitesi was 280 *effective* hours per year at the time I was working there.

Section A.1 shows the distribution of my teaching hours according to various criteria: *academic year* (L1, L2, L3, M1, M2); *type of class* (TP, TD, CM); *teaching language* (French vs. English); *institution* (Université Paul Sabatier, Galatasaray Üniversitesi, Avignon Université); student domain (Computer Science, Mathematics, Biology & Medicine, Social Sciences); and class topic (Fundamental topic, Systems, Algorithms & Programming, Artificial Intelligence & Data Science). I have taught at all levels from L1 to M2, with a prevalence of M1 and M2 classes, as I was hired by Galataray University precisely to set up several master classes. The number of TP and CM hours are quite balance and I did only few TD hours, as there was no TD hours at Galatasaray Üniversitesi. I did a significant amount of teaching in English, mainly during the master classes at Galatasaray Üniversitesi. Due to the much larger yearly duty at Galatasaray Üniversitesi, as well as the time I spent there, this institution corresponds to the majority of my teaching hours. Finally, due to my research domain, I taught a lot of classes related to Artificial Intelligence and Data Mining.

Period	Class	Diploma	Year	TP	TD	CM	Equ.
2000-01	Advanced productivity suite use	IUP TMM	L3	20			13
2000-02	Database management	IUP TMM	L3	40			27
2000-03	Introduction to C programming	IUP TMM	L2	60			40
2000-04	Artificial intelligence	IUP TMM	L3	48		4	38
2000-05	Introduction to Caml programming	Licence CIMP	L1	156	63		167
2001-02	Introduction to Pascal programming	Licence SVT	L1	12	18		26
2003-04	Introduction to computer science	Licence CIMP	L1	8			5
2003-04	Introduction to productivity suite use	Licences CIMP & SVT	L1	24			16
2004-05	Logic and Boolean algebra	Licence CIMP	L1		20		20
2004-05	Introduction to algorithms	Licence SDI	L2		10		10
2004-05	Advanced Caml programming	Licence CIMP	L2		24		24
2005-14	Algorithms & C programming	Lisans Info & Maths.	L2	84		252	434
2005-12	Web programming	Master Informatique	M2			252	378
2005-12	Operating systems	Lisans Informatique	L3			294	441
2005-13	Advanced algorithms and programming	Lisans Info & Maths.	L2	56		224	373
2005-14	Artificial intelligence	Lisans Informatique	M1			378	567
2005-14	Data mining	Master Informatique	M2			378	567
2007-12	Distributed systems	Lisans Informatique	M1	140		140	303
2008-09	Introduction to operating systems	Lisans Informatique	L2			21	32
2011-14	Introduction to computer science	Lisans Informatique	L1			6	9
2012-14	Complex network analysis	Master Informatique	M2			84	126
2014-18	Complex network analysis	Master RISM	M2	75		60	165
2014-18	Application servers	Master ILSEN	M1	120			120
2014-18	Object modeling & UML	Licence Informatique	L3	105			105
2014-21	Business intelligence	Master ILSEN	M2	189			189
2014-21	Indexing and information retrieval	Master ILSEN	M2	84		84	210
2014-21	Programming project	Licence Informatique	L3	304			304
2014-21	Introduction to complex networks	Master GCPT	M1	10		24	46
2018-19	Advanced security	Master SICOM	M2	10			11
2018-21	Business information retrieval	Master GN	M1	18		18	45
2018-21	Challenges for the digital society	Master Informatique	M2	43		3	48
2019-21	Unsupervised learning	Master IA	M1	24		24	60
			Total	1,539	135	2,214	3,888
			Yearly average	77	7	111	195

Table A.1: Summary of the classes that I have given: lab session hours (*TP*, travaux pratiques), tutorial hours (*TD*, travaux dirigés) and lecture hours (*CM*, cours magistraux). Column *Equ.* corresponds to the equivalent in tutorial hours, as is customary in the French university system. The three parts of the table describe my teaching activity in Toulouse (Université Paul Sabatier — Toulouse III), Istanbul (Galatasaray Üniversitesi), and Avignon (Avignon Université), respectively.

A.3 Supervision

This section lists the students that I supervised or co-supervised during and after my PhD. *Undergraduate students* (Section A.3.1) include students in fourth year, i.e. which corresponds to the first year of Master's degree in the EU system (M1) and the last year of Lisans diploma in the Turkish system. I did not include

third year projects (L3 in EU) or below. *Graduate students* (Section A.3.2) include students in fifth year, second year of Master's degree in the EU system (M2) and in Turkey too, as well as students possessing a Master's degree but which were not PhD candidates. Finally, *Doctoral students* (Section A.3.3) include past and current PhD students.

A.3.1 Undergraduate Students

I advised the end of studies project of 7 individual undergraduate students at Galatasaray Üniversity (the diploma is called *Lisans* in the Turkish system, and corresponds to an M1 level in the EU) and 6 groups at Avignon Université (M1), where it takes the form of a collective work including 2 to 4 students at once.

[U1] Alexandre Faure, Laurent Pereira da Silva Quintas, and Virgile Sucal. 'Classification de graphes signés (Classifying signed graphs)'. Co-advised with R. Figueiredo & N. Arınık (LIA). Master 1 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2021 (cited on pages 83, 95, 134).

[U2] Tewis Lemaire and Baptiste Quay. 'Extraction de réseaux sociaux fictionnels à partir de romans (Extraction of fictional social networks from novels)'. Co-advised with X. Bost (Orkis). Master 1 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2021 (cited on page 63).

[U3] Vincent De Germiny, Maximilien Hugot, and Alexandre Martin. 'Réseaux politiques et dynamiques spatiales (Political networks and spatial dynamics)'. Co-advised with R. Figueiredo (LIA) & A. Grunin (CIHAM – Histoire, Archéologie, Littératures des mondes chrétiens et musulmans médiévaux). Master 1 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2019.

[U4] Julien Boge, Julien Delvaux, and Adrien Sartori. 'Réseaux et pouvoir dans l'Europe du premier Moyen-Âge (Networks and power in first Middle-Age Europe)'. Co-advised with R. Figueiredo (LIA) & A. Grunin (CIHAM – Histoire, Archéologie, Littératures des mondes chrétiens et musulmans médiévaux). Master 1 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2018.

[U5] Axel Clerici, Alexandra Moshina, and Gaëtan Schmidt. 'Extraction de réseaux sociaux fictionnels à partir de textes (Extraction of fictional social networks from text)'. Master 1 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2018 (cited on page 63).

[U6] Hajar Ezzraidy, Ayoub Hajjar, and Jihan Menioui. 'Traitement des réseaux signés (Processing of signed networks)'. Co-advised with R. Figueiredo (LIA). Master 1 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2016 (cited on page 83).

[U7] Bekir Çınar. 'Détection de communautés par propagation du label pour la plateforme Gephi (Community detection through Label Propagation for the Gephi framework)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2013 (cited on page 7).

[U8] Banu Erdem. 'Analyse et extraction de données à partir du site du Parlement Européen (Analysis and extraction of data from the European Parliament website)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2013 (cited on page 136).

[U9] Hatice Burcu Küpelioğlu. 'Exploitation de la syntaxe HTML pour la reconnaissance d'entités nommées (Leveraging HTML syntax for the recognition of named entities)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2012 (cited on pages 34, 136, 138).

[U10] Burcu Kantarcı. 'Mise en place d'une Plateforme pour la détection de communautés dans les réseaux complexes (Setting up a platform for the detection of communities in complex networks)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2011 (cited on pages 7, 136).

[U11] Yasa Bolkar Akbulut. 'Reconnaissance d'entités nommées pour l'extraction automatique d'un réseau social à partir de Wikipedia (Named entity recongnition for the automatic extraction of a social network based on Wikipedia)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2011 (cited on pages 34, 136).

[U12] Cihan Aksoy and Koray Mançuhan. 'Annotation automatique de descriptions de services Web (Automatic annotation of web service descriptions)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2010 (cited on pages 137, 138).

[U13] Nadin Kökciyan. 'Classification de services Web par des méthodes de clustering (Classification of Web services through clustering methods)'. Lisans Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2009 (cited on page 137).

A.3.2 Graduate Students

The graduate student projects I advised include 1 DESS thesis (*Diplôme d'Études Supérieures Spécialisées*, some sort of professional Master's degree) in Toulouse, 5 MSc theses in Istanbul and 7 M2 theses in Avignon. I was the only advisor in Toulouse and Istanbul, whereas this is generally a group work in Avignon, which allowed me to start collaborating with various colleagues. I also list here 2 pre-doctoral internships realized by Mendonça [G9] and Arınık [G3] right after their graduation and just before they started their PhD.

[G1] Thomas Bringer. 'Extraction et analyse de graphes signés pour détecter la corruption dans les marchés publics (Extraction and analysis of signed graphs to detect corruption in public procurement)'. Co-advised with R. Figueiredo (LIA). Master 2 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2019 (cited on page 83).

[G2] Noé Cécillon. 'Exploration de caractéristiques d'embeddings de graphes pour la détection de messages abusifs (Exploring graph embedding characteristics for the detection of abusive messages)'. Co-advised with R. Dufour (LIA). Master 2 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2019 (cited on page 25).

[G3] Nejat Arınık. 'Modélisation par graphes signés pour détecter les situations de corruption et de collusion dans les marchés publics (Signed graph modeling for the detection of corruption and collusion in public procurement)'. Co-advised with R. Figueiredo (LIA) & P.-H. Morand (LBNC). Pre-doctoral Internship. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2017 (cited on pages 83, 126).

[G4] Chadine Ganouni. 'Capturer les criminels ! Une approche combinant les jeux à somme nulle et l'apprentissage en ligne (Capture the felons! An approach combining zero-sum games and on-line learning)'. Co-advised with A. Reiffers-Masson (INRIA) & Y. Hayel (LIA). Master 2 Thesis. Avignon, FR: Avignon Université, Centre d'Enseignement et de Recherche en Informatique (CERI), 2016.

[G5] Abir Hadda. 'Résolution automatique d'anaphores grammaticales (Automatic resolution of grammatical anaphoras)'. Co-advised with L. A. Cabrera Diego (LIA) & J.-M. Torres (LIA). Master 2 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2016 (cited on page 136).

[G6] Vitor dos Santos Ponciano. 'Graph optimization problems related with social network balance'. Co-advised with R. Figueiredo (LIA) & P.-H. Morand (LBNC – Laboratoire Biens, Normes, Contrats). Master 2 Thesis. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2016 (cited on pages 83, 135).

[G7] Sabrine Ayachi. 'Extraction de réseaux sociaux à partir de notices biographiques (Extraction of social networks based on biographical articles)'. Co-advised with G. Marrel (LBNC – Laboratoire Biens, Normes, Contrats) & F. Monier (CNE – Centre Norbert Elias). Master 2 Thesis. Avignon, FR: Université d'Avignon et des Pays de Vaucluse, Centre d'Enseignement et de Recherche en Informatique (CERI), 2015 (cited on pages 34, 136).

[G8] Adonis Baazaoui. 'Trust networks based on social networks'. Co-advised with R. El-Azouzi (LIA). Master 2 Thesis. Avignon, FR: Université d'Avignon et des Pays de Vaucluse, Centre d'Enseignement et de Recherche en Informatique (CERI), 2015.

[G9] Israel Mendonça. 'Signed Graph Optimization Applied to Community Detection and Related Problems'. Co-advised with R. Figueiredo (LIA) & P. Michelon (LIA). Pre-doctoral Internship. Avignon, FR: Université d'Avignon, Centre d'Enseignement et de Recherche en Informatique (CERI), 2015 (cited on pages 83, 126, 135, 138).

[G10] Samet Atdağ. 'Comparison and Combination of Named Entity Recognition Tools Applied to Biographical Texts'. MSc Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2013 (cited on pages 34, 136, 138).

[G11] Burcu Kantarcı. 'Classification of Complex Networks in Terms of Topological Properties'. MSc Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2013 (cited on page 19).

[G12] Cihan Aksoy. 'Fully Automatic Annotation of Web Service Descriptions'. MSc Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2013 (cited on pages 137, 138).

[G13] Günce Keziban Orman. 'Community Detection in Complex Networks'. MSc Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2010 (cited on pages 7, 136).

[G14] Barış Aksoy. 'Cluster Analysis of Decompression Illness'. Co-advised with M. Egit (Galatasaray Üniversitesi). MSc Thesis. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2009.

[G15] Sylvain Gadave. 'Aide à la construction de modèles en neurosciences (Helping to design neuroscience models)'. DESS Thesis. Toulouse, FR: Université Paul-Sabatier - Toulouse III, IUP ISI, DESS Ingénierie des Systèmes Informatiques, 2005 (cited on page 137).

A.3.3 Doctoral Students

I have co-advised two completed PhDs, including one in Istanbul (Orman [D6]) and one in Avignon (Gresse [D5]). I am part of the advising team of 4 ongoing PhDs: 2 are funded directly by the French Ministry of Education (Arınık [D4], Cécillon [D2]), 1 by the DeCoMaP ANR project[26] (Potin [D1]), and 1 is an interdisciplinary thesis funded by the Agorantic Research Federation[27] (Févrat [D3]).

[D1] Lucas Potin. 'Complex Graph Analysis for the Detection of Corruption in Public Procurements'. Co-advised with C. Largeron (Laboratoire Hubert Curien) & R. Figueiredo (LIA). PhD Thesis. Avignon, FR: Avignon Université, Expected in 2024 (cited on pages 83, 127).

[D2] Noé Cécillon. 'Combinaison du contenu et de la structure par representation learning : application à l'analyse de documents textuels (Combining content and structure through representation learning: application to the analysis of textual documents)'. Co-advised with G. Linarès (LIA) & R. Dufour (LIA). PhD Thesis. Avignon, FR: Université d'Avignon, Expected in 2022 (cited on pages 25, 32, 64, 127, 134, 137).

[D3] Noémie Févrat. 'Réélection ou rotation ? Reconstitution automatique de trajectoires politiques et réformes de la représentation démocratique (Relection or rotation? Automatically retrieval of professional political trajectories, and democratic representation reforms)'. Co-advised with G. Marrel (LBNC – Laboratoire Biens, Normes, Contrats). PhD Thesis. Avignon, FR: Université d'Avignon, Expected in 2022 (cited on pages 49, 127).

[D4] Nejat Arınık. 'Solving Graph partitioning problems related to structural balance'. Co-advised with R. Figueiredo (LIA) & R. El-Azouzi (LIA). PhD Thesis. Avignon, FR: Avignon Université, 2021 (cited on pages 83, 104, 127, 134, 135, 138).

[26] https://anr.fr/Project-ANR-19-CE38-0004
[27] https://agorantic.univ-avignon.fr/en/

[D5] Adrien Gresse. 'L'art de la voix : Caractériser l'information vocale dans un choix artistique (The art of voice: Characterizing vocal information in artistic choices)'. Co-advised with J.-F. Bonastre (LIA) & R. Dufour (LIA). PhD Thesis. Avignon, FR: Avignon Université, 2020. ⟨tel-02938152⟩ (cited on page 127).

[D6] Günce Keziban Orman. 'Contribution to the interpretation of evolving communities in complex networks: Application to the study of social interactions'. Co-advised with J.-F. Boulicaut (INSA Lyon). PhD Thesis. Istanbul, TR & Lyon, FR: Galatasaray Üniversitesi & INSA Lyon, 2014. ⟨tel-01081028⟩ (cited on pages 37, 44, 127).

A.4 Review Work

A.4.1 Conferences

A.4.1.1 International Conferences

- Member of the Program Committee, *2nd International Workshop on Modeling and Mining Social-Media-Driven Complex Networks* (Soc2Net), IEEE / ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM), The Hague, NL, 2020.
- Member of the Review Committee, *International Workshop on Modeling and Mining Social-Media-driven Complex Networks* (Soc2Net), International AAAI Conference on Web and Social Medial (ICWSM), Munich, DE, 2019.
- Member of the Program Committee, *1st International workshop on Online Social Networks and Media: Network Properties*, The Web Conf (WWW), Lyon, FR, 2018.
- Member of the Program Committee, *International workshop on Mining Attributed Networks* (MATNet), The Web Conf (WWW), Lyon, FR, 2018.
- Member of the Technical Program Committee, *22ème IEEE Symposium on Computers and Communications* (ISCC), Heraklion, GR, 2017.
- Member of the Review Committee, *20th IEEE Symposium on Computers and Communications* (ISCC), Larnaca, CY, 2015.
- Member of the Review Committee, *13th European Control Conference* (ECC), Strasbourg, FR, 2014.
- Member of the International Program Committee, *1st International Conference on Digital Information and Communication Technology and its Applications* (DICTAP), Dijon, FR, 2011.
- Member of the International Program Committee, *2nd International Conference on Networked Digital Technologies* (NDT), Prague, CZ, 2010.
- Member of the International Program Committee, *1st International Conference on Networked Digital Technologies* (NDT), Ostrava, CZ, 2009.

A.4.1.2 National Conferences

- Member of the Program Committee, *11ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques* (MARAMI), Montpellier, FR, 2020.
- Member of the Program Committee, *57ème colloque de l'ASRDLF*, Avignon, FR, 2020–2021.
- Member of the Program Committee, *Atelier Démonstrations* (DémoEGC), 18ème Conférence internationale sur l'extraction et la gestion des connaissances (EGC), Paris, FR, 2018.
- Member of the Program Committee, 18ème Conférence internationale sur l'extraction et la gestion des connaissances (EGC), Paris, FR, 2018.
- Member of the Program Committee, *9ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques* (MARAMI), Avignon, FR, 2018.
- Member of the Program Committee, *Atelier Démonstrations* (DémoEGC), 17ème Conférence internationale sur l'extraction et la gestion des connaissances (EGC), Grenoble, FR, 2017.

- Member of the Program Committee, *8ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques* (MARAMI), La Rochelle, FR, 2017.
- Member of the Program Committee, *Atelier Données participatives et sociales* (DPS), 16ème Conférence internationale sur l'extraction et la gestion des connaissances (EGC), Reims, FR, 2016.
- Member of the Program Committee, *Atelier Démonstrations* (DémoEGC), 16ème Conférence internationale sur l'extraction et la gestion des connaissances (EGC), Reims, FR, 2016.
- Member of the Program Committee, *7ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques* (MARAMI), Cergy, FR, 2016.
- Member of the Program Committee, *1er Atelier Réseaux sociaux et Intelligence Artificielle* (ARIA), 9ème Plate-forme Intelligence Artificielle (PFIA), Rennes, FR, 2015.
- Member of the review committee, *Atelier Démonstrations* (DémoEGC), 15ème Conférence internationale sur l'extraction et la gestion des connaissances (EGC), Luxembourg, LU, 2015.
- Member of the review committee, *6ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques* (MARAMI), Nîmes, FR, 2015.

A.4.2 Journals

A.4.2.1 Editor

I have been an editor at Complexity from 2018 to 2020, and I have guest-edited two special issues of the Journal of Interdisciplinary Methodologies and Issues in Science.

- Member of the editorial committee, *Complexity*, Wiley-Hindawi, 2018–2020.
- Guest editor, *Journal of Interdisciplinary Methodologies and Issues in Science* (JIMIS), Special issue on the Analysis of networks and graphs, Épisciences (CNRS), 2019.
- Guest co-editor, *Journal of Interdisciplinary Methodologies and Issues in Science* (JIMIS), Special issue on Graphs and Social Systems, Épisciences (CNRS), 2017.

A.4.2.2 Reviewer

- *EPJ Data Science*, Springer, 2021.
- *PLoS ONE*, Public Library of Science, 2015–2017, 2020.
- *Computer Communications*, Elsevier, 2020.
- *Frontiers in Cell and Developmental Biology*, Frontiers, 2020.
- *Frontiers in Physics*, Frontiers, 2020.
- *IEEE Transactions on Knowledge and Data Engineering* (TKDE), IEEE, 2013, 2018–2020.
- *Information Sciences* (Inform. Sciences), Elsevier, 2016, 2018, 2020.
- *Entropy*, MDPI, 2020.
- *Journal of Business Research*, Elsevier, 2019, 2020.
- *IEEE/ACM Transactions on Networking* (TNET), IEEE/ACM, 2016, 2018–2020.
- *Symmetry*, MDPI, 2019.
- *Online Social Networks and Media*, Elsevier, 2019.
- *Future Generation Computer Systems*, Elsevier, 2019.
- *Processes*, MDPI, 2019.
- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE, 2018.
- *Social Networks Analysis and Mining* (SNAM), Springer, 2013–2016, 2018.
- *Applied Sciences*, MDPI, 2018.
- *WIREs Data Mining and Knowledge Discovery*, Wiley, 2018.
- *Journal of Complex Networks* (J. Comp. Net.), Oxford University Press, 2017.
- *Algorithms*, MDPI, 2017.
- *Computational Intelligence* (Comput. Intell.), Wiley, 2015–2016.
- *International Journal of Information Technology and Management* (IJITM), InderScience, 2016.

- *Complex Adaptive Systems Modeling* (CASM), Springer, 2016.
- *International Journal of Computational Science and Engineering* (IJCSE), InderScience, 2016.
- *Revue d'Intelligence Artificielle* (RIA), Lavoisier, 2016.
- *International Journal of Social Network Mining* (IJSNM), InderScience, 2013, 2015.
- *Psychometrika*, Springer, 2015.
- *International Journal of Information and Communication Technology* (IJICT), InderScience, 2014.
- *Physica A: Statistical Mechanics and its Applications* (Phys. A), Elsevier, 2014.
- *IEEE Wireless Communication Letters* (IEEE Wireless Comm. Lett.), IEEE, 2014.

A.4.3 PhD defense committees

- Co-advisor for Adrien Gresse, "L'art de la voix : Caractériser l'information vocale dans un choix artistique" (The art of the voice: Characterizing vocal information when performing an artistic choice), Laboratoire Informatique d'Avignon (LIA), Avignon Université, Advisors: Jean-François Bonastre, Richard Dufour & Vincent Labatut, 06/02/2020. ⟨tel-02938152⟩
- Examiner for Vinh Loc Dao, "Characterizing community detection algorithms and modules in large scale complex networks", Institut Mines-Télécom Atlantique (IMT Atlantique), Brest (FR), Advisors: Philippe Lenca & Cécile Bothorel, 17/12/2018. ⟨tel-02121358⟩
- Guest for Maël Camu, "Détection de communautés orientée sommet pour réseaux mobiles opportunistes sociaux" (Vertex-centred community detection for opportunistic mobile social networks), Laboratoire d'informatique de Paris 6 (LIP6), Université Pierre et Marie Curie, Equipe Learning, Fuzzy and Intelligent systems (LFI), Paris (FR), Advisors: Marie-Jeanne Lesot, Adrien Revault d'Allonnes & Marcin Detyniecki, 20/12/2017. ⟨tel-01745380⟩
- Examiner for Xavier Bost, "A storytelling machine? Automatic video summarization: the case of TV series", Laboratoire Informatique d'Avignon (LIA), Université d'Avignon et des Pays de Vaucluse, Avignon (FR), Advisors: Georges Linarès, Serigne Gueye & Damien Malinas, 23/11/2016. ⟨tel-01402549⟩
- Guest for Jean-Valère Cossu, "Analyse de l'image de marque sur le Web 2.0" (Brand Image on the Web 2.0), Laboratoire Informatique d'Avignon (LIA), Université d'Avignon et des Pays de Vaucluse, Avignon (FR), Advisors: Marc El-Bèze, Juan-Manuel Torres & Éric SanJuan, 16/12/2015. ⟨tel-01291032⟩
- Examiner for Adrien Guille, "Diffusion de l'information dans les médias sociaux : modélisation et analyse" (Information Diffusion in Social Medias: Modeling and Analysis), Laboratoire Entrepôts, Représentation & Ingénierie des Connaissances (ERIC), Université Lyon 2, Lyon (FR), Advisors: Djamel Zighed & Cécile Favre, 25/11/2014. ⟨tel-01100255⟩
- Co-advisor for Günce Keziban Orman, "Contribution to the interpretation of evolving communities in complex networks: Application to the study of social interactions", INSA Lyon, Lyon (FR), Advisors: Jean-François Boulicaut & Vincent Labatut, 16/07/2014. ⟨tel-01081028⟩

A.4.4 Research agencies

- Evaluation of research project proposals for the Swiss Academy of Technical Sciences (SATW), *Partenariat Hubert Curien/Germaine de Staël* (PHC/GdS) program, 2019.
- Evaluation of research project proposals for the Agence nationale pour la recherche (ANR), *Collaborative Research* (PRC) and *Young Researchers* (JCJC) programs , 2017.
- Evaluation of research project proposals for the Netherlands Organisation for Scientific Research (NWO), *Innovational Research Incentives Scheme* (*Veni scheme*), 2016.
- Evaluation of research project proposals for the Agence nationale pour la recherche (ANR), *Young Researchers program* (JCJC), 2016.

A.5 Seminars

This section lists the invited speeches that I have given since the end of my PhD.

- "Résumé automatique de séries TV" (Automatic summarization of TV series). 20èmes Rencontres enseignants-chercheurs LIA/LMA, Avignon, France, 12/03/2020.
- "Modélisation par graphes signés pour détecter les situations de conflits" (Signed graph-based models to detect conflictual situations). Digital Humanities Day, organized by the UMR Ciham & the FR Agorantic, Avignon, France, 27/05/2019.
- "Playing around with CLEF collections, Elastic Search and Kibana". CLEF/ARIA Master Class, with R. Dufour & L. Ermakova, CLEF 2018, Avignon, France, 14/09/2018.
- "Données ouvertes et détection de corruption dans les marchés publics" (Open data and detection of corruption in public procurements). Series of seminars on the digital world, Université pour Tous, Vaison-la-romaine, France, 09/01/2018.
- "Modèles à base de graphes signés pour la détection de corruption dans les marchés publics" (Signed Graph-based Models for the Detection of Corruption in Public Procurements). Seminar of the Living Lab, Villa Créative Supramuros, Université d'Avignon et des Pays du Vaucluse (UAPV), Avignon, France, 11/07/2017.
- "Modélisation par graphes signés" (Modeling through Signed Graphs). Seminar of the Agorantic Research Federation, Université d'Avignon et des Pays du Vaucluse (UAPV), Avignon, France, 07/12/2016.
- "Complex Networks & Community Detection". École Doctorale Agrosciences & Sciences (ED 536), Université d'Avignon et des Pays du Vaucluse (UAPV), Avignon, France, 15/06/2016.
- "Complex Networks: Presentation, main results and open problems". Maore Team, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), Université de Montpellier, Montpellier, France, 09/05/2016.
- "Rôle communautaire d'un nœud" (Community role of a vertex). *Travail d'Espace* seminars, UMR Espace, Aix-Marseille Université (AMU), Aix-en-Provence, France, 26/05/2015.
- "Natural Language and Complex Networks". *Apprentissage automatique et sciences du langage* seminars, Laboratoire Parole et Langage (LPL), Aix-Marseille Université (AMU), Aix-en-Provence, France, 15/05/2015.
- "Analyse de réseaux complexes" (Complex networks analysis). ComplexNetworks Team, Laboratoire d'informatique de l'Université Pierre et Marie Curie – Paris 6 (LIP6), Paris, France, 12/04/2013.
- "Analyse de réseaux complexes" (Complex networks analysis). A3 Team, Laboratoire d'informatique de l'Université Paris Nord – Paris 13 (LIPN), Paris, France, 11/04/2013.
- "Community detection in complex networks: problems, methods and applications". Workshop on Graph Theory and Applications III, Boğaziçi University, Istanbul, Turkey, 11/10/2012.
- "Complex Networks Analysis". Galatasaray Üniversitesi, Mathematics Department, Istanbul, Turkey, 20/04/2012.
- "Détection et interprétation de communautés dans les réseaux complexes" (Detection and interpretation of communities in complex networks). IRIT-UT1 Seminars, University of Toulouse 1 Capitole, Toulouse, France, 24/01/2012.
- "Réseaux complexes : une introduction" (Complex Networks: an Introduction). Galatasaray Üniversitesi, Computer Science Department, Istanbul, Turkey, 21/12/2011.
- "Complex Networks: Methods and Application to a Few Case Studies". SosLab Team, Boğaziçi University, Istanbul, Turkey, 20/06/2011.
- "Réseaux causaux probabilistes à grande échelle" (Large-scale causal probabilistic networks). Cortex Team, Loria, Nancy, France, 29/03/2004.

- Roundtable host for the session "Apports des connaissances expertes en IA" (Contribution of expert knowledge in AI), 1st conference on AI in Région Sud, Avignon, FR, 28/11/2019. Participants: Elena Cabrio, Serena Villata, & Frédéric Boucharat.

A.6 Research Management

This section describes the part of my activity loosely related to the management of research, including the organization of conferences (Section A.6.1), the funding of research projects (Section A.6.2), and research-related duties (Section A.6.3).

A.6.1 Conference Organization

- Local chair, International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Avignon, FR, 2019.
- Head of the organizing committee, Modèles & Analyse des Réseaux : Approches Mathématiques & Informatiques (MARAMI), Avignon, FR, 2018.
- Local chair, Conference and Labs of the Evaluation Forum (CLEF), Avignon, FR, 2018.
- Co-head of the organizing committee, Journée Graphes et Systèmes Sociaux (JGSS), Avignon, FR, 2016.

A.6.2 Funded Projects

A.6.2.1 ANR Projects

- Project *Détection de la corruption dans les marchés publics* (DeCoMaP – Detection of Corruption in Public Procurement), ANR-19-CE38-0004, 2019-2023, budget: 300k€, coordinator: Pierre-Henri Morand (LBNC – Laboratoire Biens, Normes, Contrats), LIA supervisor: **Vincent Labatut**.
- Project *Gallerie des festivals* (GaFes – Gallery of Festivals), ANR-14-CE24-0022, 2014-2019, budget: 800k€, coordinator: Georges Linarès (LIA).

A.6.2.2 Other National Funding

- PGMO (Programme Gaspard Monge pour l'Optimization et la recherche opérationnelle – Gaspard Monge Programme for Optimization and Operations Research), Project *Clustering and path problems on signed social networks*, 2019-21, budget: 6k€, coordinator: Rosa Figueiredo (LIA).
- PGMO (Programme Gaspard Monge pour l'Optimization et la recherche opérationnelle – Gaspard Monge Programme for Optimization and Operations Research), Project *Exploiting Antagonistic Relations in Signed Networks under the Structural Balance Hypothesis*, 2017-19, budget: 8k€, coordinator: Rosa Figueiredo (LIA).
- CNRS PEPS (Projets Exploratoires Premier Soutien – Exploratory Projects) MoMIS (Modèles mathématiques et Interactions Sociales – Mathematical models and Social Interactions), Project *Urbi&Orbi*, 2015-16, budget: 5k€, coordinator: Didier Josselin (UMR Espace).
- PGMO (Programme Gaspard Monge pour l'Optimization et la recherche opérationnelle – Gaspard Monge Programme for Optimization and Operations Research), Project *Signed graph optimization applied to community detection and related problems*, 2015-17, budget: 6k€ coordinator: Rosa Figueiredo (LIA).

A.6.2.3 Binational Funding

- PHP Pessoa (Hubert Curien French-Portuguese binational programme), Project *Combinatorial Optimization Methods for a Social Network Application*, 2018-19, budget: 5k€, coordinator: Rosa Figueiredo (LIA) & Cristina Requejo (DMAT).

A.6.2.4 Regional Funding

- EJD PACA (Provence-Alpes-Côte d'Azur doctoral programme), Project *Alert : Modèles d'utilisateurs pour la supervision automatique des réseaux sociaux* (User models for the automatic supervision of social medias), 2015-18, coordinator: Georges Linarès (LIA), industrial partner: Nectar de Code.

A.6.2.5 Local Funding

- Projets d'Excellence (Avignon University Funding), Project *DE2T : Live-tweet politique et riposte-party – les communautés 2.0 lors du débat présidentiel de l'entre deux-tours de 2017* (Political live-tweet and riposte-party – communities 2.0 during the in-between-two-rounds presidential debate), budget: 8k€, 2017-18, coordinator: Mickael Rouvier (LIA).
- FR Agorantic, Projet *CoCoMa : Modélisation par graphes signés pour détecter les situations de corruption et de collusion dans les marchés publics* (CoCoMa: Signed graph modeling for the detection of corruption and collusion in public procurement), budget: 6k€, 2016-17, coordinator: **Vincent Labatut**.
- FR Agorantic, Project *Optimisation et équilibre structurel dans les graphes signés : une application pour comprendre les marchés publics* (Optimization and structural balance in signed graphs: an application to understand public procurement), budget: 3k€, 2015-16, coordinator: Rosa Figueiredo (LIA).
- Pierre Bergé grant, Project *Recommandation Automatique de Voix pour la Création Multimédia* (Automatic voice recommendation for multimedia creation), one PhD student funded by the Fondation Universitaire d'Avignon, 2015-18, coordinator: Jean-François Bonastre (LIA).
- FR Agorantic, Project *RÉSOPO : Extraction de réseaux sociaux implicites à partir de notices biographiques d'acteurs politiques d'hier et d'aujourd'hui* (Extraction of implicit social networks based on past and current biographical notices), budget: 3k€, 2014-15, coordinator: **Vincent Labatut** (LIA).
- LIA (Laboratory internal project), Project *Signed Graph Optimization Applied to Community Detection and Related Problems*, 2014-15, budget: 15k€, coordinator: **Vincent Labatut** (LIA).

A.6.3 Research Duties

- 2020–present: Contact at the LIA for the Carnot Institute on Cognition.
- 2016–present: Member of the LIA Laboratory Council (Conseil de laboratoire): I was elected to this position in 2016 and 2020.
- 2016–present: Co-head of the *Politic(s), transparency et ethic* research axis at the Agorantic research federation: I was nominated at this position in 2016.

A.7 Dissemination

This section lists the software which I produced or helped produce (Section A.7.1), as the well as the corpora and datasets that I elaborated or help elaborating during my research (Section A.7.2). Note that I only focus on the resources for which I *actively* contributed. Every time it is legally possible, I publish the resources that I produce or help producing under an open license, and make them publicly available online. The URL hosting these resources are indicated in this section. This section also lists the general public papers written about my research (Section A.7.3).

A.7.1 Software

[S1] *SignedCentrality – Centrality measures for signed networks*
- Date: 2020–present
- Description: Centrality measures and embeddings for signed networks
- Authors: Alexandre Faure, Laurent Pereira da Silva Quintas, Virgile Sucal, Nejat Arınık, and **Vincent Labatut**
- Programming language: Python
- Repository: `https://github.com/CompNet/SignedCentrality`
- Used in: [U1]
- cited on page 83.

[S2] *MedievalAvignon – Analysis of social and spatial networks in medieval Avignon*
- Date: 2019–present
- Description: Extraction and analysis of networks based on medieval real estate data
- Authors: **Vincent Labatut**
- Programming language: Java, R
- Repository: `https://github.com/CompNet/MedievalAvignon`
- cited on pages 17, 80.

[S3] *TrajanNet – Social network of emperor Trajan*
- Date: 2019–2020
- Description: Extraction and analysis of social networks centered around Roman emperor Trajan, see also the dataset [C1]
- Authors: **Vincent Labatut**
- Programming language: R
- Repository: `https://github.com/CompNet/TrajanNet`
- Used in: [VL88]
- cited on pages 17, 49, 137.

[S4] *BréfInit – Extraction of political data*
- Date: 2019–2020
- Description: Software written to build the first version of the BRÉF database
- Authors: **Vincent Labatut**
- Programming language: R
- Repository: `https://github.com/CompNet/BrefInit`
- Used in: [VL95]
- cited on page 49.

[S5] *Alert – Abuse Detection in Online Conversations*
- Date: 2018–present
- Description: Detection of abusive message by combining content- and graph-based features
- Authors: Noé Cécillon, **Vincent Labatut**, and Richard Dufour
- Programming language: Python, R
- Repository: `https://github.com/CompNet/Alert`
- Used in: [VL2, VL27, VL76, D2]
- cited on page 29.

[S6] *MultiNetVotes – Multiple Partitioning of Multiplex Signed Networks*
- Date: 2018–2019
- Description: Extraction and analysis of vote-based multiplex networks
- Authors: Nejat Arınık and **Vincent Labatut**
- Programming language: R
- Repository: `https://github.com/CompNet/MultiNetVotes` ⟨`hal-02180966`⟩
- Used in: [VL4, VL78, D4]
- cited on page 106.

[S7] *SignedBenchmark – Random generation of signed graphs*

▹ Date: 2017–2018
▹ Description: Produces signed graphs with controlled structural balance, cf. the corresponding dataset [C2]
▹ Authors: **Vincent Labatut**
▹ Programming language: R
▹ Repository: `https://github.com/CompNet/SignedBenchmark`
▹ Used in: [VL3]
▹ cited on pages 91, 137.

[S8] *TranspoloSearch – Web-based information extraction for political science*
▹ Date: 2015–2018
▹ Description: Reconstructing the calendar of elected representatives based on online press
▹ Authors: **Vincent Labatut**
▹ Programming language: Java
▹ Repository: `https://github.com/CompNet/TranspoloSearch` ⟨`hal-02178524`⟩
▹ Used in: [VL64, VL67]
▹ cited on page 34.

[S9] *Continuous Straightness – Topological measure to describe spatial graphs*
▹ Date: 2016
▹ Description: Continuous approach to compute the straightness measure
▹ Authors: **Vincent Labatut**
▹ Programming language: R
▹ Repository: `https://github.com/CompNet/SpatialMeasures` ⟨`hal-02180337`⟩
▹ Used in: [VL8]
▹ cited on page 77.

[S10] *Opinion Centrality – Opinion-based centrality measure for multiplex networks*
▹ Date: 2015–2016
▹ Description: Centrality measure for multiplex networks
▹ Authors: Alexandre Reiffers and **Vincent Labatut**
▹ Programming language: R
▹ Repository: `https://github.com/CompNet/MultiplexCentrality` ⟨`hal-02180368`⟩
▹ Used in: [VL10]
▹ cited on page 101.

[S11] *NetVotes – Extraction and partition of voting networks*
▹ Date: 2014–2018
▹ Description: Extraction and analysis of vote-based similarity networks, see also the dataset [C7]
▹ Authors: Israel Mendonça, Nejat Arınık, and **Vincent Labatut**
▹ Programming language: R
▹ Repository: `https://github.com/CompNet/NetVotes` ⟨`hal-02180177`⟩
▹ Used in: [VL29, VL36, VL68, G6, G9, D4]
▹ cited on pages 86, 138.

[S12] *SpiderNet – Generation and analysis of spatial graphs*
▹ Date: 2014–2016
▹ Description: Comparative study of orb weaver's networks and urban networks
▹ Authors: **Vincent Labatut**
▹ Programming language: R
▹ Repository: `https://github.com/geomatique/SpiderNet`
▹ Used in: [VL35, VL61]
▹ cited on page 67.

[S13] *Twitter Influence – Detecting offline influence through Twitter activity*
▹ Date: 2014–2015
▹ Description: Detection of Twitter users which are influential in the real-world
▹ Authors: Jean-Valère Cossu, Nicolas Dugué, and **Vincent Labatut**

- ▹ Programming language: Java, Perl, R
- ▹ Repository: https://github.com/CompNet/Influence ⟨hal-02179513⟩
- ▹ Used in: [VL11, VL37]
- ▹ cited on page 23.

[S14] *TopoMeasures – Topological Measures for Community Detection Assessment*
- ▹ Date: 2012–2016
- ▹ Description: Performance assessment measures for community detection methods
- ▹ Authors: **Vincent Labatut**
- ▹ Programming language: R
- ▹ Repository: https://github.com/CompNet/TopoMeasures ⟨hal-02179362⟩
- ▹ Used in: [VL13, VL73].

[S15] *SocCap – Social Capitalists & Community Roles*
- ▹ Date: 2013–2014
- ▹ Description: Identification of social capitalists and their community roles
- ▹ Authors: Nicolas Dugué, **Vincent Labatut**, and Anthony Perez
- ▹ Programming language: R, C++, C, Java
- ▹ Repository: https://github.com/CompNet/Orleans ⟨hal-02178682⟩
- ▹ Used in: [VL12, VL39, VL71, VL72].

[S16] *Nerwip – Named Entity Extraction for Wikipedia Pages*
- ▹ Date: 2011–2014
- ▹ Description: Assessment, combination and comparison of NER tools. See also the corpus [C8]
- ▹ Authors: **Vincent Labatut**, Yasa Bolkar Akbulut, Hatice Burcu Küpelioğlu, and Samet Atdağ
- ▹ Programming language: Java
- ▹ Repository: https://github.com/CompNet/nerwip ⟨hal-02178479⟩
- ▹ Used in: [VL41, VL96, U9, U11, G5, G7, G10]
- ▹ cited on page 138.

[S17] *TBB – Total Boum Boum*
- ▹ Date: 2008–2014
- ▹ Description: Clone of the Bomberman game, including an API to develop intelligent agents
- ▹ Authors: **Vincent Labatut**
- ▹ Programming language: Java
- ▹ Repository: https://github.com/vlabatut/totalboumboum
- ▹ Used in: Artificial Intelligence classes
- ▹ cited on page 121.

[S18] *Ganetto – Galatasaray Network Toolbox*
- ▹ Date: 2009–2013
- ▹ Description: Collection of scripts for the generation and analysis of complex networks
- ▹ Authors: **Vincent Labatut**, Günce Keziban Orman, and Burcu Kantarcı
- ▹ Programming language: R
- ▹ Repository: https://github.com/CompNet/Ganetto ⟨hal-02177739⟩
- ▹ Used in: [VL13, VL15, VL16, VL18, VL42, VL45, VL49, VL53, VL54, VL73, VL74, U10, G13].

[S19] *Jepa – Java European Parliament API*
- ▹ Date: 2008–2013
- ▹ Description: Extraction of data from the European Parliament official website
- ▹ Authors: **Vincent Labatut** and Banu Erdem
- ▹ Programming language: Java
- ▹ Repository: ⟨hal-02177397⟩
- ▹ Used in: [U8].

[S20] *G+P – Google+ Parser*
- ▹ Date: 2011–2012
- ▹ Description: Extraction of data from the (now discontinued) Google+ social media
- ▹ Authors: **Vincent Labatut**

- ▹ Programming language: Java
- ▹ Repository: https://github.com/CompNet/GooglePlusParser ⟨hal-02177387⟩.

[S21] *Mataws – Multimodal Automatic Tool for the Annotation of Web Services*
- ▹ Date: 2010–2012
- ▹ Description: Automatic annotation of Web service descriptions, see also the dataset [C9]
- ▹ Authors: Cihan Aksoy, **Vincent Labatut**, and Koray Mançuhan
- ▹ Programming language: Java
- ▹ Repository: https://github.com/CompNet/mataws ⟨hal-01937857⟩
- ▹ Used in: [VL46, VL97, U12, U13, G12]
- ▹ cited on page 138.

[S22] *Rage – Réseaux à grande échelle*
- ▹ Date: 2000–2003
- ▹ Description: Implementation of the modeling framework designed during my PhD
- ▹ Authors: **Vincent Labatut**
- ▹ Programming language: Java
- ▹ Repository: ⟨hal-02177293⟩
- ▹ Used in: [VL19, VL56–VL59, VL75, VL82, VL83, VL89, VL90, G15].

A.7.2 Corpora

[C1] *Trajan's Networks – Entourage of emperor Trajan*
- ▹ Date: 2019–2020
- ▹ Description: Description of the entourage of Roman emperor Trajan, see also the software [S3]
- ▹ Authors: Gaëtane Vallet and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.13143104
- ▹ Used in: [VL88]
- ▹ cited on pages 17, 49, 134.

[C2] *Benchmark of signed networks*
- ▹ Date: 2019–2020
- ▹ Description: Collection of artificial unweighted complete signed graphs, see also the model [S7]
- ▹ Authors: Nejat Arınık, **Vincent Labatut**, and Rosa Figueiredo
- ▹ Repository: 10.6084/m9.figshare.8233340
- ▹ Used in: [VL3]
- ▹ cited on pages 91, 135.

[C3] *WAC – Wikipedia Abusive Conversations*
- ▹ Date: 2019–2020
- ▹ Description: Corpus of abusive conversations between Wikipedia editors
- ▹ Authors: Noé Cécillon, Richard Dufour, and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.11299118
- ▹ Used in: [VL25, D2]
- ▹ cited on page 26.

[C4] *SpaceOrigin Data – Conversational network corpus*
- ▹ Date: 2016–2020
- ▹ Description: Conversational networks extracted from a collection of chat messages
- ▹ Authors: Étienne Papégnies, Noé Cécillon, Richard Dufour, and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.7442273
- ▹ Used in: [VL2, VL7, VL27, VL30, VL63, VL76, D2]
- ▹ cited on pages 26, 29.

[C5] *Serial Speakers – Collection of annotated TV serials*
- ▹ Date: 2015–2020
- ▹ Description: Multimodal annotation of TV series

- ▹ Authors: Xavier Bost and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.3471839
- ▹ Used in: [VL26]
- ▹ cited on pages 55, 63.

[C6] *Narrative Smoothing – Fictional graphs extracted through narrative smoothing*
- ▹ Date: 2015–2016
- ▹ Description: Conversational networks extracted from TV serials
- ▹ Authors: Xavier Bost and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.2199646
- ▹ Used in: [VL5, VL6, VL20, VL34, VL66]
- ▹ cited on page 63.

[C7] *NetVotes Data – Graphs extracted from the European Parliament votes using our NetVotes software*
- ▹ Date: 2014–2016
- ▹ Description: Vote similarity networks from the European Parliament, see also the software [S11]
- ▹ Authors: **Vincent Labatut**, Israel Mendonça, and Nejat Arınık
- ▹ Repository: 10.6084/m9.figshare.1456268, 10.6084/m9.figshare.5785833
- ▹ Used in: [VL4, VL29, VL36, VL68, G9, D4]
- ▹ cited on pages 86, 106, 135.

[C8] *Nerwip Corpus – Annotated Wikipedia biographical articles*
- ▹ Date: 2011–2014
- ▹ Description: Articles annotated for named entities, see also the software [S16]
- ▹ Authors: Hatice Burcu Küpelioğlu, Samet Atdağ, and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.1289791
- ▹ Used in: [VL41, VL96, U9, G10]
- ▹ cited on page 136.

[C9] *Mataws Corpus – Annotated Web service descriptions*
- ▹ Date: 2010–2012
- ▹ Description: Web service descriptions annotated with our tool Mataws [S21]
- ▹ Authors: Cihan Aksoy and **Vincent Labatut**
- ▹ Repository: 10.6084/m9.figshare.1289755
- ▹ Used in: [VL46, VL97, U12, G12]
- ▹ cited on page 137.

[C10] *Galatanet Dataset – Social network of the Galatasaray University students*
- ▹ Date: 2009–2010
- ▹ Description: Data collected during a student survey
- ▹ Authors: **Vincent Labatut** and Jean-Michel Balasque
- ▹ Repository: 10.6084/m9.figshare.1289732
- ▹ Used in: [VL23, VL24, VL48, VL98]
- ▹ cited on page 11.

A.7.3 Outreach

Several outreach media have presented some work to which I contributed:

- ▶ Researchers develop a new system to detect abuse in online communities, **TechXplore**, 13/02/2019, for [VL7].
- ▶ The Emerging Threat from Twitter's Social Capitalists, **MIT Technology Review**, 02/07/2014, for [VL39].

A.8 Publications

This section is an exhaustive list of my publications. All the full texts are available in their official version (when applicable) by clicking on the title hyperlink in the publication entry. The preprint versions are all available through my personal page on HAL[28] (preprint repository of the CNRS, the French center for scientific research), or directly by clicking on the HAL hyperlink in the bibliographic entry.

A.8.1 International Peer-reviewed Journals

[VL1] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Characterizing external measures for the assessment of cluster analysis and community detection'. In: *IEEE Access* 9 (2021), pp. 20255–20276. DOI: 10.1109/access.2021.3054621. ⟨hal-03124118⟩ (cited on pages 83, 91, 105).

[VL2] Noé Cécillon, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Graph embeddings for Abusive Language Detection'. In: *Springer Nature Computer Science* 2 (2021), p. 37. DOI: 10.1007/s42979-020-00413-7. ⟨hal-03042171⟩ (cited on pages 25, 33, 134, 137).

[VL3] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Multiplicity and Diversity: Analyzing the Optimal Solution Space of the Correlation Clustering Problem on Complete Signed Graphs'. In: *Journal of Complex Networks* 8.6 (2020), cnaa025. DOI: 10.1093/comnet/cnaa025. ⟨hal-02994011⟩ (cited on pages 83, 90, 92, 135, 137).

[VL4] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Multiple partitioning of multiplex signed networks: Application to European parliament votes'. In: *Social Networks* 60 (2020), pp. 83–102. DOI: 10.1016/j.socnet.2019.02.001. ⟨hal-02082574⟩ (cited on pages 83, 104, 106, 107, 109, 134, 138).

[VL5] Xavier Bost, Serigne Gueye, **Vincent Labatut**, Martha Larson, Georges Linarès, Damien Malinas, and Raphaël Roth. 'Remembering Winter Was Coming: Character-oriented Video Summaries of TV Series'. In: *Multimedia Tools and Applications* 78.24 (2019), pp. 35373–35399. DOI: 10.1007/s11042-019-07969-4. ⟨hal-02278188⟩ (cited on pages 51, 58, 60, 61, 138).

[VL6] **Vincent Labatut** and Xavier Bost. 'Extraction and Analysis of Fictional Character Networks: A Survey'. In: *ACM Computing Surveys* 52.5 (2019), p. 89. DOI: 10.1145/3344548. ⟨hal-02173918⟩ (cited on pages 51, 63, 138).

[VL7] Étienne Papégnies, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Conversational Networks for Automatic Online Moderation'. In: *IEEE Transactions on Computational Social Systems (TCSS)* 6.1 (2019), pp. 38–55. DOI: 10.1109/TCSS.2018.2887240. ⟨hal-01999546⟩ (cited on pages 25, 28, 29, 137, 138).

[VL8] **Vincent Labatut**. 'Continuous Average Straightness in Spatial Graphs'. In: *Journal of Complex Networks* 6.2 (2018), pp. 269–296. DOI: 10.1093/comnet/cnx033. ⟨hal-01571212⟩ (cited on pages 67, 73–75, 77, 78, 135).

[VL9] Günce Keziban Orman, **Vincent Labatut**, and Ahmet Teoman Naskali. 'Exploring the Evolution of Node Neighborhoods in Dynamic Networks'. In: *Physica A: Statistical Mechanics and its Applications* 482 (2017), pp. 375–391. DOI: 10.1016/j.physa.2017.04.084. ⟨hal-01512952⟩ (cited on pages 37, 39).

[VL10] Alexandre Reiffers and **Vincent Labatut**. 'Opinion-based centrality in multiplex networks: A convex optimization approach'. In: *Network Science* 5.2 (2017), pp. 213–234. DOI: 10.1017/nws.2017.7. ⟨hal-01486629⟩ (cited on pages 97, 98, 135).

[VL11] Jean-Valère Cossu, **Vincent Labatut**, and Nicolas Dugué. 'A Review of Features for the Discrimination of Twitter Users: Application to the Prediction of Offline Influence'. In: *Social Network Analysis and Mining (SNAM)* 6 (2016), p. 25. DOI: 10.1007/s13278-016-0329-x. ⟨hal-01203171⟩ (cited on pages 20, 21, 136).

[28] https://cv.archives-ouvertes.fr/vlabatut

[VL12] Nicolas Dugué, **Vincent Labatut**, and Anthony Perez. 'A Community Role Approach To Assess Social Capitalists Visibility in the Twitter Network'. In: *Social Network Analysis and Mining (SNAM)* 5 (2015), p. 26. DOI: 10.1007/s13278-015-0266-0. ⟨hal-01163741⟩ (cited on pages 4, 7, 19, 136).

[VL13] **Vincent Labatut**. 'Generalized Measures for the Evaluation of Community Detection Methods'. In: *International Journal of Social Network Mining (IJSNM)* 2.1 (2015), pp. 44–63. DOI: 10.1504/IJSNM.2015.069776. ⟨hal-00802923⟩ (cited on pages 4, 8, 136).

[VL14] Günce Keziban Orman, **Vincent Labatut**, Marc Plantevit, and Jean-François Boulicaut. 'Interpreting communities based on the evolution of a dynamic attributed network'. In: *Social Network Analysis and Mining (SNAM)* 5 (2015), p. 20. DOI: 10.1007/s13278-015-0262-4. ⟨hal-01163778⟩ (cited on pages 37, 44, 47).

[VL15] Günce Keziban Orman, **Vincent Labatut**, and Hocine Cherifi. 'Towards realistic artificial benchmark for community detection algorithms evaluation'. In: *International Journal of Web Based Communities* 9.3 (2013), pp. 349–370. DOI: 10.1504/IJWBC.2013.054908. ⟨hal-00840261⟩ (cited on pages 4, 7, 136).

[VL16] Günce Keziban Orman, **Vincent Labatut**, and Hocine Cherifi. 'Comparative Evaluation of Community Detection Algorithms: A Topological Approach'. In: *Journal of Statistical Mechanics: Theory and Experiment* 08 (2012), P08001. DOI: 10.1088/1742-5468/2012/08/P08001. ⟨hal-00710659⟩ (cited on pages 4, 7, 136).

[VL17] **Vincent Labatut** and Hocine Cherifi. 'Evaluation of Performance Measures for Classifiers Comparison'. In: *Ubiquitous Computing and Communication Journal* 6 (2011), pp. 21–34. ⟨hal-00653071⟩ (cited on page 4).

[VL18] Günce Keziban Orman, **Vincent Labatut**, and Hocine Cherifi. 'On Accuracy of Community Structure Discovery Algorithms'. In: *Journal of Convergence Information Technology* 6.11 (2011), pp. 283–292. DOI: 10.4156/jcit.vol6.issue11.32. ⟨hal-00653084⟩ (cited on pages 4, 7, 136).

[VL19] **Vincent Labatut**, Josette Pastor, Serge Ruff, Jean-François Démonet, and Pierre Celsis. 'Cerebral modeling and dynamic Bayesian networks'. In: *Artificial Intelligence in Medicine* 30.2 (2004), pp. 119–139. DOI: 10.1016/S0933-3657(03)00042-3. ⟨hal-00634307⟩ (cited on pages 4, 113, 137).

A.8.2 International Peer-reviewed Book Chapters

[VL20] Xavier Bost, **Vincent Labatut**, Serigne Gueye, and Georges Linarès. 'Extraction and analysis of dynamic conversational networks from TV series'. In: *Social Network Analysis and Mining*. Ed. by M. Kaya, J. Kawash, S. Khoury, and M. Y. Day. Lecture Notes in Social Networks. Springer, 2018. Chap. 3, pp. 55–84. DOI: 10.1007/978-3-319-78196-9_3. ⟨hal-01543938⟩ (cited on pages 51, 138).

[VL21] Guillaume Marrel and **Vincent Labatut**. 'La visibilité politique en ligne de la maire de Paris - Contribution à la mesure de l'écho Web-médiatique d'Anne Hidalgo'. In: *Big Data et visibilité en ligne - Un enjeu pluridisciplinaire de l'économie numérique*. Presses des Mines, 2018, pp. 271–286. ⟨hal-03173422⟩. URL: https://www.pressesdesmines.com/produit/big-data-et-visibilite-en-ligne/ (cited on pages 4, 34).

[VL22] **Vincent Labatut** and Günce Keziban Orman. 'Community Structure Characterization'. In: *Encyclopedia of Social Network Analysis and Mining*. Ed. by R. Alhajj and J. Rokne. Springer, 2017. DOI: 10.1007/978-1-4614-7163-9_110151-1. ⟨hal-01525440⟩ (cited on pages 8, 17, 37).

[VL23] **Vincent Labatut** and Jean-Michel Balasque. 'Informative Value of Individual and Relational Data Compared Through Business-Oriented Community Detection'. In: *The Influence of Technology on Social Network Analysis and Mining*. Ed. by T. Özyer, J. Rokne, G. Wagner, and A. H. P. Reuser. Lecture Notes in Social Networks. Springer, 2013. Chap. 6, pp. 303–330. DOI: 10.1007/978-3-7091-1346-2_13. ⟨hal-00633650⟩ (cited on pages 8, 10, 17, 138).

[VL24] **Vincent Labatut** and Jean-Michel Balasque. 'Detection and Interpretation of Communities in Complex Networks: Methods and Practical Application'. In: *Computational Social Networks: Tools, Perspectives and Applications*. Ed. by A. Abraham and A.-E. Hassanien. Springer, 2012. Chap. 4, pp. 81–113. DOI: 10.1007/978-1-4471-4048-1_4. ⟨hal-00633653⟩ (cited on pages 8, 13, 17, 138).

A.8.3 International Peer-reviewed Conferences With Proceedings

[VL25] Noé Cécillon, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'WAC: A Corpus of Wikipedia Conversations for Online Abuse Detection'. In: *12th Language Resources and Evaluation Conference (LREC)*. Marseille, FR, 2020, pp. 1375–1383. ⟨hal-02497514⟩ (cited on pages 26, 32, 137).

[VL26] Xavier Bost, **Vincent Labatut**, and Georges Linarès. 'Serial Speakers: a Dataset of TV Series'. In: *12th Language Resources and Evaluation Conference (LREC)*. Marseille, FR, 2020, pp. 4249–4257. ⟨hal-02477736⟩ (cited on pages 51, 55, 63, 138).

[VL27] Noé Cécillon, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Abusive Language Detection in Online Conversations by Combining Content- and Graph-based Features'. In: *International Workshop on Modeling and Mining Socia-Media Driven Complex Networks (Soc2Net)*. Vol. 2. Frontiers in Big Data 8. Munich, DE, 2019. DOI: 10.3389/fdata.2019.00008. ⟨hal-02130205⟩ (cited on pages 25, 134, 137).

[VL28] Adrien Gresse, Mathias Quillot, Richard Dufour, **Vincent Labatut**, and Jean-Francois Bonastre. 'Similarity metric based on Siamese neural networks for voice casting'. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Brighton, UK: IEEE Publishing, 2019, pp. 6585–6589. DOI: 10.1109/ICASSP.2019.8683178. ⟨hal-02004762⟩ (cited on page 4).

[VL29] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Signed Graph Analysis for the Interpretation of Voting Behavior'. In: *International Conference on Knowledge Technologies and Data-driven Business (i-KNOW) - International Workshop on Social Network Analysis and Digital Humanities (SnanDig)*. Vol. 2025. CEUR Workshop Proceedings. Graz, AT, 2017. ⟨hal-01583133⟩ (cited on pages 83, 88, 89, 106, 107, 135, 138).

[VL30] Étienne Papegnies, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Graph-based Features for Automatic Online Abuse Detection'. In: *5th International Conference on Statistical Language and Speech Processing (SLSP)*. Vol. 10583. Lecture Notes in Artificial Intelligence. Le Mans, FR: Springer, 2017, pp. 70–81. DOI: 10.1007/978-3-319-68456-7_6. ⟨hal-01571639⟩ (cited on pages 25, 137).

[VL31] Mathias Quillot, Cassandre Ollivier, Richard Dufour, and **Vincent Labatut**. 'Exploring Temporal Analysis of Tweet Content from Cultural Events'. In: *5th International Conference on Statistical Language and Speech Processing (SLSP)*. Vol. 10583. Lecture Notes in Artificial Intelligence. Le Mans, FR: Springer, 2017, pp. 82–93. DOI: 10.1007/978-3-319-68456-7_7. ⟨hal-01580578⟩ (cited on page 4).

[VL32] Adrien Gresse, Mickael Rouvier, Richard Dufour, **Vincent Labatut**, and Jean-Francois Bonastre. 'Acoustic Pairing of Original and Dubbed Voices in the Context of Video Game Localization'. In: *Interspeech*. Stockholm, SE, 2017, pp. 2839–2843. DOI: 10.21437/Interspeech.2017-1311. ⟨hal-01572151⟩ (cited on page 4).

[VL33] Étienne Papegnies, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Impact Of Content Features For Automatic Online Abuse Detection'. In: *18th International Conference on Computational Linguistics and Intelligent Text Processing (CICling)*. Vol. 10762. Lecture Notes in Artificial Intelligence. Budapest, HU: Springer, 2017, pp. 404–419. DOI: 10.1007/978-3-319-77116-8_30. ⟨hal-01505502⟩ (cited on pages 25, 29).

[VL34] Xavier Bost, **Vincent Labatut**, Serigne Gueye, and Georges Linarès. 'Narrative smoothing: dynamic conversational network for the analysis of TV Series plots'. In: *2nd International Workshop on Dynamics in Networks (DyNo/ASONAM)*. San Francisco, US: IEEE Publishing, 2016, pp. 1111–1118. DOI: 10.1109/ASONAM.2016.7752379. ⟨hal-01276708⟩ (cited on pages 51, 138).

[VL35] Didier Josselin, **Vincent Labatut**, and Dieter Mitsche. 'Straightness of rectilinear vs. radio-concentric networks: modeling, simulation and comparison'. In: *7th Annual Symposium on Simulation for Architecture and Urban Design (SimAUD)*. Ed. by R. Attar, A. Chronis, S. Hanna, and M. Turrin. London, UK, 2016, pp. 95–102. ⟨hal-01367824⟩ (cited on pages 67, 70, 135).

[VL36] Israel Mendonça, Rosa Figueiredo, **Vincent Labatut**, and Philippe Michelon. 'Relevance of Negative Links in Graph Partitioning: A Case Study Using Votes From the European Parliament'. In: *2nd European Network Intelligence Conference (ENIC)*. Karlskrona, SE: IEEE Publishing, 2015, pp. 122–129. DOI: 10.1109/ENIC.2015.25. ⟨hal-01176090⟩ (cited on pages 83, 86, 87, 135, 138).

[VL37] Jean-Valère Cossu, Nicolas Dugué, and **Vincent Labatut**. 'Detecting Real-World Influence Through Twitter'. In: *2nd European Network Intelligence Conference (ENIC)*. Karlskrona, SE: IEEE Publishing, 2015, pp. 83–90. DOI: 10.1109/ENIC.2015.20. ⟨hal-01164453⟩ (cited on pages 20, 136).

[VL38] Günce Keziban Orman, **Vincent Labatut**, Marc Plantevit, and Jean-François Boulicaut. 'A Method for Characterizing Communities in Dynamic Attributed Complex Networks'. In: *IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM)*. Beijing, CN: IEEE Publishing, 2014, pp. 481–484. DOI: 10.1109/ASONAM.2014.6921629. ⟨hal-01011913⟩ (cited on pages 37, 44).

[VL39] Nicolas Dugué, **Vincent Labatut**, and Anthony Perez. 'Identifying the Community Roles of Social Capitalists in the Twitter Network'. In: *IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM)*. Beijing, CN: IEEE Publishing, 2014, pp. 371–374. DOI: 10.1109/ASONAM.2014.6921612. ⟨hal-01011910⟩ (cited on pages 4, 7, 19, 136, 138).

[VL40] Burcu Kantarcı and **Vincent Labatut**. 'Classification of Complex Networks Based on Topological Properties'. In: *3rd Conference on Social Computing and its Applications (SCA)*. Karlsruhe, DE: IEEE Publishing, 2013, pp. 297–304. DOI: 10.1109/CGC.2013.54. ⟨hal-00940688⟩ (cited on pages 4, 19).

[VL41] Samet Atdağ and **Vincent Labatut**. 'A Comparison of Named Entity Recognition Tools Applied to Biographical Texts'. In: *2nd International Conference on Systems and Computer Science (ICSCS)*. Lille, FR: IEEE Publishing, 2013, pp. 228–233. DOI: 10.1109/IcConSCS.2013.6632052. ⟨hal-00849797⟩ (cited on pages 4, 34, 136, 138).

[VL42] Günce Keziban Orman, **Vincent Labatut**, and Hocine Cherifi. 'An Empirical Study of the Relation Between Community Structure and Transitivity'. In: *3rd Workshop on Complex Networks (CompleNet)*. Vol. 424. Studies in Computational Intelligence. Melbourne, US: Springer, 2012, pp. 99–110. DOI: 10.1007/978-3-642-30287-9_11. ⟨hal-00717707⟩ (cited on pages 4, 7, 136).

[VL43] **Vincent Labatut** and Atay Özgövde. 'Topological measures for the analysis of wireless sensor networks'. In: *3rd International Conference on Ambient Systems, Networks and Technologies (ANT)*. Vol. 10. Procedia Computer Science. Niagara Falls, CA: Elsevier, 2012, pp. 397–404. DOI: 10.1016/j.procs.2012.06.052. ⟨hal-00723724⟩ (cited on pages 37, 49, 67).

[VL44] Chantal Cherifi, **Vincent Labatut**, and Jean-François Santucci. 'On Flexible Web Services Composition Networks'. In: *1st International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. Vol. 166. Communications in Computer and Information Science. Dijon, FR: Springer, 2011, pp. 45–59. DOI: 10.1007/978-3-642-21984-9_5. ⟨hal-00620565⟩ (cited on page 4).

[VL45] Günce Keziban Orman, **Vincent Labatut**, and Hocine Cherifi. 'Qualitative Comparison of Community Detection Algorithms'. In: *1st International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. Vol. 167. Communications in Computer and Information Science. Dijon, FR: Springer, 2011, pp. 265–279. DOI: 10.1007/978-3-642-22027-2_23. ⟨hal-00611385⟩ (cited on pages 4, 7, 136).

[VL46] Cihan Aksoy, **Vincent Labatut**, Chantal Cherifi, and Jean-François Santucci. 'MATAWS: A Multimodal Approach for Automatic WS Semantic Annotation'. In: *3rd International Conference on Networked Digital Technologies (NDT)*. Vol. 136. Communications in Computer and Information Science. Macau, CN: Springer, 2011, pp. 319–333. DOI: 10.1007/978-3-642-22185-9_27. ⟨hal-00620566⟩ (cited on pages 4, 137, 138).

[VL47] **Vincent Labatut** and Hocine Cherifi. 'Accuracy Measures for the Comparison of Classifiers'. In: *5th International Conference on Information Technology (ICIT)*. Amman, JO, 2011, 5p. ⟨hal-00611319⟩ (cited on page 4).

[VL48] **Vincent Labatut** and Jean-Michel Balasque. 'Business-oriented Analysis of a Social Network of University Students'. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Odense, DK: IEEE Publishing, 2010, pp. 25–32. DOI: 10.1109/ASONAM.2010.15. ⟨hal-00633643⟩ (cited on pages 8, 10, 17, 138).

[VL49] Günce Keziban Orman and **Vincent Labatut**. 'The Effect of Network Realism on Community Detection Algorithms'. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Odense, DK: IEEE Publishing, 2010, pp. 301–305. DOI: 10.1109/ASONAM.2010.70. ⟨hal-00633641⟩ (cited on pages 4, 7, 136).

[VL50] Chantal Cherifi, **Vincent Labatut**, and Jean-François Santucci. 'Benefits of Semantics on Web Service Composition from a Complex Network Perspective'. In: *2nd International Conference on Networked Digital Technologies (NDT)*. Vol. 88. Communications in Computer and Information Science. Prague, CZ: Springer, 2010, pp. 80–90. DOI: 10.1007/978-3-642-14306-9_9. ⟨hal-00620559⟩ (cited on page 4).

[VL51] Chantal Cherifi, **Vincent Labatut**, and Jean-François Santucci. 'Topological Properties of Web Services Similarity Networks'. In: *6th International Conference on Computer Science and Information Systems (ICCSIS)*. Athens, GR: Athens Institute for Education and Research (ATINER), 2010, 12p. ⟨hal-00620552⟩ (cited on page 4).

[VL52] Chantal Cherifi, **Vincent Labatut**, and Jean-François Santucci. 'Web Services Dependency Networks Analysis'. In: *2nd International Conference on New Media and Interactivity (NMIC)*. Istanbul, TR, 2010, 9p. ⟨hal-00620541⟩ (cited on page 4).

[VL53] Günce Keziban Orman and **Vincent Labatut**. 'A Comparison of Community Detection Algorithms on Artificial Networks'. In: *12th International Conference on Discovery Science (DS)*. Vol. 5808. Lecture Notes in Artificial Intelligence. Porto, PT: Springer, 2009, pp. 242–256. DOI: 10.1007/978-3-642-04747-3_20. ⟨hal-00633640⟩ (cited on pages 4, 7, 136).

[VL54] Günce Keziban Orman and **Vincent Labatut**. 'Relative Evaluation of Partition Algorithms for Complex Networks'. In: *1st International Conference on Networked Digital Technologies (NDT)*. Ostrava, CZ: IEEE Publishing, 2009, pp. 20–25. DOI: 10.1109/NDT.2009.5272078. ⟨hal-00633624⟩ (cited on pages 4, 7, 136).

[VL55] Barış Aksoy, **Vincent Labatut**, Murat Egi, Tamer Özyiğit, Petar Denoble, Costantino Balestra, Richard Vann, and Alessandro Marroni. 'Association rules of DCI patient clusters and reliability of clustering analysis'. In: *International Conference on Diving Hyperbaric Medicine*. Aberdeen, UK: European Underwater and Barometrical Society, 2009, 6p. ⟨hal-00633616⟩ (cited on page 4).

[VL56] **Vincent Labatut**, Josette Pastor, and Serge Ruff. 'Dynamic Bayesian modeling of the cerebral activity'. In: *18th International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, MX: AAAI Press, 2003, pp. 169–174. DOI: 10.1.1.106.8036. ⟨hal-00634306⟩ (cited on pages 4, 137).

[VL57] **Vincent Labatut** and Josette Pastor. 'Dynamic Bayesian Networks for Integrated Neural Computation'. In: *1st International IEEE-EMBS Conference on Neural Engineering*. Capri, IT: Omnipress, 2003, pp. 537–540. DOI: 10.1109/CNE.2003.1196882. ⟨hal-00634305⟩ (cited on pages 4, 137).

[VL58] **Vincent Labatut** and Josette Pastor. 'Modeling the cerebral activity with dynamic probabilistic networks'. In: *5th International Conference on Simulations in Biomedicine (BioMedicine)*. Ljubljana, SI: WIT Press, 2003, pp. 459–468. DOI: 10.2495/BIO030451. ⟨hal-00634299⟩ (cited on pages 4, 137).

[VL59] **Vincent Labatut** and Josette Pastor. 'Bayesian Modeling of Cerebral Information Processing'. In: *Bayesian Models in Medicine - 8th European conference on Artificial Intelligence in Medicine (AIME)*. Cascais, PT, 2001, pp. 41–46. ⟨hal-00634303⟩ (cited on pages 4, 137).

A.8.4 International Peer-reviewed Conferences Without Proceedings

[VL60] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Study of the European Parliament votes through the multiple partitioning of signed multiplex networks'. In: *29th European Conference on Operational Research (EURO)*. Valencia, ES, 2018. ⟨hal-01939888⟩ (cited on pages 83, 104).

A.8.5 National Peer-reviewed Journals

[VL61] Didier Josselin, Sonia Chekir, Alain Pasquet, **Vincent Labatut**, Yvan Capowiez, Christophe Mazzia, Yezekael Hayel, Adrien Lammoglia, Cyrille Genre-Grandpierre, Dieter Mitsche, and Frédéric Patras. 'Modélisation, simulation et analyse de propriétés de réseaux orbitèles (Modeling, simulation and analysis of the properties of orb weaver's networks)'. In: *Revue Internationale de Géomatique / International Journal of Geomatics and Spatial Analysis* 25.4 (2015), pp. 515–536. DOI: 10.3166/RIG.25.515-536. ⟨hal-01249881⟩ (cited on pages 67, 135).

A.8.6 National Peer-reviewed Conferences With Proceedings

[VL62] Adrien Gresse, Richard Dufour, **Vincent Labatut**, Mickaël Rouvier, and Jean-François Bonastre. 'Mesure de similarité fondée sur des réseaux de neurones siamois pour le doublage de voix (Similarity measure based on siamese neural networks for voice dubbing)'. In: *XXXII^{èmes} Journées d'Études sur la Parole (JEP)*. Aix-en-Provence, FR, 2018, pp. 10–18. DOI: 10.21437/JEP.2018-2. ⟨hal-01819198⟩ (cited on page 4).

[VL63] Étienne Papégnies, Richard Dufour, **Vincent Labatut**, and Georges Linarès. 'Détection de messages abusifs au moyen de réseaux conversationnels (Detection of abusive messages based on conversational networks)'. In: *8ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. La Rochelle, FR, 2017, 12p. ⟨hal-01614279⟩ (cited on pages 25, 137).

[VL64] **Vincent Labatut** and Guillaume Marrel. 'La visibilité politique en ligne : Contribution à la mesure de l'e-reputation politique d'un maire urbain (Online political visibility: contribution to the measure of the political e-reputation of an urban mayor)'. In: *Big Data et visibilité en ligne : Un enjeu pluridisciplinaire de l'économie numérique*. Fort-de-France, FR, 2017, pp. 271–286. ⟨hal-01904352⟩ (cited on pages 4, 34, 135).

[VL65] Étienne Papegnies, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Detection of abusive messages in an on-line community'. In: *14ème Conférence en Recherche d'Information et Applications (CORIA)*. Marseille, FR, 2017, pp. 153–168. DOI: 10.24348/coria.2017.16. ⟨hal-01505017⟩ (cited on page 25).

[VL66] Xavier Bost, **Vincent Labatut**, Serigne Gueye, and Georges Linarès. 'Extraction de réseaux dynamiques conversationnels par lissage narratif (Extraction of dynamical conversational networks through narrative smoothing)'. In: *7ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Cergy-Pontoise, FR, 2016, 12p. ⟨hal-01385215⟩ (cited on pages 51, 138).

[VL67] Guillaume Marrel, **Vincent Labatut**, and Marc El Bèze. 'Le Web comme miroir du travail politique quotidien ? Reconstituer l'écho médiatique en ligne des événements d'un agenda d'élu (The Web as a mirror of the daily political work? Identifying the online media coverage of the events constituting the agenda of an elected politician)'. In: *13ème Congrès de l'Association Française de Science Politique (AFSP)*. Aix-en-Provence, FR, 2015, p. 25. ⟨hal-01904338⟩ (cited on pages 4, 34, 113, 135).

[VL68] Israel Mendonça, Rosa Figueiredo, **Vincent Labatut**, and Phillipe Michelon. 'Informative Value of Negative Links for Graph Partitioning, with an application to European Parliament Votes'. In: *6ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Nîmes, FR, 2015, 12p. ⟨hal-02055158⟩ (cited on pages 83, 86, 135, 138).

[VL69] **Vincent Labatut**. 'Étude de l'omniprésence des propriétés petit-monde et sans-échelle (Study of the ubiquity of the small-world and scale-free properties)'. In: *5ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Paris, FR, 2014, 12p. ⟨hal-01075680⟩ (cited on page 4).

[VL70] Günce Keziban Orman, **Vincent Labatut**, Marc Plantevit, and Jean-François Boulicaut. 'Une méthode pour caractériser les communautés des réseaux dynamiques à attributs (A method to characterize communities in attributed dynamic networks)'. In: *14ème Conférence Extraction et Gestion des Connaissances (EGC)*. Rennes, FR, 2014, pp. 101–112. ⟨hal-00918181⟩ (cited on pages 37, 44).

[VL71] Nicolas Dugué, **Vincent Labatut**, and Anthony Perez. 'Identification de rôles communautaires dans des réseaux orientés, appliquée à Twitter (Identification of community roles in directed networks, applied to Twitter)'. In: *14ème Conférence Extraction et Gestion des Connaissances (EGC)*. Rennes, FR, 2014, pp. 125–130. ⟨hal-00918175⟩ (cited on pages 4, 7, 19, 136).

[VL72] Nicolas Dugué, **Vincent Labatut**, and Anthony Perez. 'Rôle communautaire des capitalistes sociaux dans Twitter (Community role of social capitalists in Twitter)'. In: *4ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Saint-Étienne, FR, 2013, 12p. ⟨hal-00859779⟩ (cited on pages 4, 7, 19, 136).

[VL73] **Vincent Labatut**. 'Une nouvelle mesure pour l'évaluation des méthodes de détection de communautés (A new measure for the evaluation of community detection methods)'. In: *3ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Villetaneuse, FR, 2012, 12p. ⟨hal-00743888⟩ (cited on pages 4, 8, 136).

[VL74] Günce Keziban Orman, **Vincent Labatut**, and Hocine Cherifi. 'Relation entre transitivité et structure de communauté dans les réseaux complexes (Relation between transitivity and community structure in complex networks)'. In: *2ème Journée thématique Fouille de grands graphes (JFGG)*. Grenoble, FR, 2011, 4p. ⟨hal-01112256⟩ (cited on pages 4, 7, 136).

[VL75] Josette Pastor and **Vincent Labatut**. 'Modélisation de la structure fonctionnelle cérébrale (Modeling the cerebral functional structure)'. In: *10èmes Journées Neurosciences & Sciences de l'Ingénieur (NSI)*. Dinar, FR, 2000, pp. 93–94. ⟨hal-01176094⟩ (cited on pages 4, 137).

A.8.7 National Peer-reviewed Conferences Without Proceedings

[VL76] Noé Cécillon, **Vincent Labatut**, Richard Dufour, and Georges Linarès. 'Tuning Graph2vec with Node Labels for Abuse Detection in Online Conversations'. In: *11ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Montpellier, FR, 2020. ⟨hal-02993571⟩ (cited on pages 25, 33, 134, 137).

[VL77] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Characterizing measures for the assessment of cluster analysis and community detection'. In: *11ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Montpellier, FR, 2020. ⟨hal-02993542⟩ (cited on page 83).

[VL78] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Multiple Partitioning of Multiplex Signed Networks'. In: *21ème Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF)*. Montpellier, FR, 2020. ⟨hal-02428300⟩ (cited on pages 83, 104, 134).

[VL79] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Multiple Optimal Solutions but Single Search: A Study of the Correlation Clustering Problem'. In: *20ème Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF)*. Le Havre, FR, 2019. ⟨hal-02051683⟩ (cited on page 83).

[VL80] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Exploiting Antagonistic Relations in Signed Graphs under the Structural Balance Hypothesis'. In: *PGMO Days*. Paris, FR, 2018. ⟨hal-01939893⟩ (cited on pages 83, 104).

[VL81] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Analysis of Roll-Calls in the European Parliament by Multiple Partitioning of Multiplex Signed Networks'. In: *9ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques (MARAMI)*. Avignon, FR, 2018. ⟨hal-01933679⟩ (cited on pages 83, 104).

A.8.8 Local Conferences Without Proceedings

[VL82] **Vincent Labatut**, Josette Pastor, and Serge Ruff. 'Le cerveau humain : un réseau causal fonctionnel ? (The human brain: a functional causal network?)' In: *1er Colloque des doctorants de l'École Doctorale Informatique et Télécommunications (EDIT)*. Toulouse, FR: Université Paul Sabatier - Toulouse III, 2002, pp. 34–39. ⟨hal-01176101⟩ (cited on pages 4, 137).

[VL83] **Vincent Labatut**, Josette Pastor, Marc Lafon, and Pierre Celsis. 'Modèles intégrés du traitement de l'information cérébrale (Integrated models of cerebral information processing)'. In: *Journées INSERM Midi-Pyrénées*. Toulouse, FR: INSERM, 2001, 2p. ⟨hal-01176104⟩ (cited on pages 4, 137).

A.8.9 Editorials

[VL84] **Vincent Labatut**. 'Introduction to the Special Issue on Graph and Network Analysis'. In: *Journal of Interdisciplinary Methodologies and Issues in Science (JIMIS)* 2 (2019). DOI: 10.18713/JIMIS-180719-5-6. ⟨hal-02368152⟩.

[VL85] **Vincent Labatut** and Rosa Figueiredo. 'Introduction to the special issue on Graphs & Social Systems'. In: *Journal of Interdisciplinary Methodologies and Issues in Science (JIMIS)* 2 (2017). DOI: 10.18713/JIMIS-300617-2-0. ⟨hal-01560655⟩ (cited on page 83).

A.8.10 Submitted Articles

[VL86] Noé Cécillon, Richard Dufour, and **Vincent Labatut**. 'Approche multimodale par plongements de texte et de graphes pour la détection de messages abusifs (Text and graph multimodal embedding approach for the detection of abusive messages)'. In: *Submitted to Traitement Automatique des Langues* (2021) (cited on page 33).

[VL87] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. 'Local Search-based Enumeration of Optimal Solution Space for Correlation Clustering Problem'. In: *Submitted to the Journal of Global Optimization* (2021) (cited on page 94).

[VL88] Gaëtane Vallet, **Vincent Labatut**, and Catherine Wolff. 'Personnages centraux et capital social dans l'entourage de Trajan de 98 à 118 apr. J.-C. (Central characters and social capital in the entourage of Trajan from 98 to 118 AD)'. In: *Submitted to the Journal of Historical Network Research* (2020) (cited on pages 17, 49, 110, 134, 137).

A.8.11 Theses

[VL89] **Vincent Labatut**. 'Réseaux causaux probabilistes à grande échelle : un nouveau formalisme pour la modélisation du traitement de l'information cérébrale (Large scale probabilistic causal networks: a new formalism for the modeling of cerebral information processing)'. PhD Thesis. Toulouse, FR: Université Paul-Sabatier - Toulouse III, École Doctorale Représentation des connaissances et formalisation du raisonnement (RCFR), 2003. ⟨tel-00005190⟩ (cited on pages 4, 119, 137).

[VL90] **Vincent Labatut**. 'Réseaux causaux : quelle approche pour le cerveau ? (Causal networks: Which approach to model the brain?)' DEA Thesis. Toulouse, FR: Université Paul-Sabatier - Toulouse III, Formation Doctorale Représentation des connaissances et formalisation du raisonnement (RCFR), 2000 (cited on pages 4, 119, 137).

A.8.12 Lecture Notes

[VL91] Damien Berthet and **Vincent Labatut**. *Supports de cours 2005–2014 (Lecture Notes 2005–2014)*. Vol. 1. Algorithmique & programmation en langage C (Algorithms & C Programming). Galatasaray Üniversitesi, 2014. ⟨cel-01176119⟩ (cited on page 121).

[VL92] Damien Berthet and **Vincent Labatut**. *Sujets de travaux pratiques 2005–2014 (Lab Session Questions 2005–2014)*. Vol. 2. Algorithmique & programmation en langage C (Algorithms & C Programming). Galatasaray Üniversitesi, 2014. ⟨cel-01176120⟩ (cited on page 121).

[VL93] Damien Berthet and **Vincent Labatut**. *Corrigés de travaux pratiques 2005–2014 (Lab Session Corrections 2005–2014)*. Vol. 3. Algorithmique & programmation en langage C (Algorithms & C Programming). Galatasaray Üniversitesi, 2014. ⟨cel-01176121⟩ (cited on page 121).

A.8.13 Reports and Unpublished Documents

[VL94] Nejat Arınık, Rosa Figueiredo, and **Vincent Labatut**. *2-Edge Connected Balanced Subgraphs for Correlation Clustering Problem*. Research Report. Avignon, FR: Avignon Université, 2020. ⟨hal-02428305⟩ (cited on pages 83, 95).

[VL95] **Vincent Labatut**, Noémie Février, and Guillaume Marrel. *BRÉF – Base de données Révisée des Élu·es de France (Revised database of the representatives elected in France)*. Technical Report. Avignon, FR: Avignon Université, 2020. ⟨hal-02886580⟩ (cited on pages 49, 134).

[VL96] **Vincent Labatut**. *Improved Named Entity Recognition Through SVM-based Combination*. Research Report. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2013, 12p. ⟨hal-01322867⟩ (cited on pages 34, 136, 138).

[VL97] Cihan Aksoy and **Vincent Labatut**. *A Fully Automatic Approach to the Semantic Annotation of Web Service Descriptions*. Research Report. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2012, 21p. ⟨hal-01112241⟩ (cited on pages 137, 138).

[VL98] Jean-Michel Balasque and **Vincent Labatut**. *Valeur informative supplémentaire apportée par l'analyse des réseaux sociaux: premiers enseignements d'une étude au sein d'une population étudiante (Supplementary informational value produced through social network analysis: first results of a study carried over a student population)*. Research Report. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering & Faculty of Management, 2010. ⟨hal-01863315⟩ (cited on pages 8, 10, 17, 138).

[VL99] Günce Keziban Orman and **Vincent Labatut**. *A Modification to Improve the Realism of Networks Generated with the LFR Model*. Research Report. Istanbul, TR: Galatasaray Üniversitesi, Faculty of Computer Engineering, 2010. ⟨hal-01863318⟩ (cited on page 7).

Bibliography

A

[1] D. A. Abduljabbar, S. Z. Mohd Hashim, and R. Sallehuddin. 'Nature-inspired optimization algorithms for community detection in complex networks: a review and future trends'. In: *Telecommunication Systems* 74 (2020), pp. 225–252. DOI: 10.1007/s11235-019-00636-x (cited on page 7).

[2] A. Agarwal, A. Kotalwar, and O. Rambow. 'Automatic Extraction of Social Networks from Literary Text: A Case Study on Alice in Wonderland'. In: *International Joint Conference on Natural Language Processing*. 2013, pp. 1202–1208. URL: http://www.aclweb.org/website/old_anthology/I/I13/I13-1171.pdf (cited on page 55).

[3] C. Aggarwal and K. Subbian. 'Evolutionary Network Analysis: A Survey'. In: *ACM Computing Surveys* 47.1 (2014), p. 10. DOI: 10.1145/2601412 (cited on page 37).

[4] R. Albert and A.-L. Barabási. 'Statistical mechanics of complex networks'. In: *Reviews of Modern Physics* 74.1 (2002), pp. 47–96. DOI: 10.1103/RevModPhys.74.47 (cited on pages 1, 3).

[5] A. Aleahmad, P. Karisani, M. Rahgozar, and F. Oroumchian. 'University of Tehran at RepLab 2014'. In: *Working Notes for CLEF 2014 Conference*. Vol. 1180. CEUR Workshop Proceedings. 2014, pp. 1528–1536. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-AleahmadEt2014.pdf (cited on pages 21, 24).

[6] Z. Alès and A. Knippel. 'An extended edge-representative formulation for the K-partitioning problem'. In: *Electronic Notes in Discrete Mathematics* 52 (2016), pp. 333–342. DOI: 10.1016/j.endm.2016.03.044 (cited on page 88).

[7] Z. Alès, A. Knippel, and A. Pauchet. 'Polyhedral combinatorics of the K-partitioning problem with representative variables'. In: *Discrete Applied Mathematics* 211 (2016), pp. 1–14. DOI: 10.1016/j.dam.2016.04.002 (cited on page 91).

[8] A. Aleta and Y. Moreno. 'Multilayer Networks in a Nutshell'. In: *Annual Review of Condensed Matter Physics* 10 (2019), pp. 45–62. DOI: 10.1146/annurev-conmatphys-031218-013259 (cited on page 97).

[9] T. Alzahrani and K. J. Horadam. 'Community Detection in Bipartite Networks: Algorithms and Case studies'. In: *Complex Systems and Networks*. Understanding Complex Systems. Springer, 2016, pp. 25–50. DOI: 10.1007/978-3-662-47824-0_2 (cited on page 7).

[10] E. Amigó, J. Carrillo-de-Albornoz, I. Chugur, A. Corujo, J. Gonzalo, E. Meij, M. de Rijke, and D. Spina. 'Overview of RepLab 2014: Author Profiling and Reputation Dimensions for Online Reputation Management'. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Vol. 8685. Lecture Notes in Computer Science. Springer, 2014, pp. 307–322. DOI: 10.1007/978-3-319-11382-1_24 (cited on page 20).

[11] M. Atzmueller, S. Günnemann, and A. Zimmermann. 'Mining communities and their descriptions on attributed graphs: a survey'. In: *Data Mining and Knowledge Discovery* in press (2021). DOI: 10.1007/s10618-021-00741-z (cited on page 9).

[12] T. Aynaud and J.-L. Guillaume. 'Static community detection algorithms for evolving networks'. In: *8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*. 2010, pp. 508–514. URL: http://ieeexplore.ieee.org/document/5520221/ (cited on pages 37, 46).

B

[13] P. Bajardi, A. Barrat, F. Natale, L. Savini, and L. V. Colizza. 'Dynamical patterns of cattle trade movements'. In: *PLoS ONE* 6.5 (2011), e19869. DOI: 10.1371/journal.pone.0019869 (cited on page 37).

[14] E. Balas E.and Zemel. 'An algorithm for large zero-one knapsack problems'. In: *Operations Research* 28.5 (1980), pp. 1130–1154. DOI: 10.1287/opre.28.5.1130 (cited on page 61).

[15] K. Balci and A. A. Salah. 'Automatic analysis and identification of verbal aggression and abusive behaviors for online social games'. In: *Computers in Human Behavior* 53 (2015), pp. 517–526. DOI: 10.1016/j.chb.2014.10.025 (cited on page 25).

[16] N. Bansal, A. Blum, and S. Chawla. 'Correlation Clustering'. In: *43rd Annual IEEE Symposium on Foundations of Computer Science*. 2002, pp. 238–247. DOI: 10.1109/SFCS.2002.1181947 (cited on page 85).

[17] A.-L. Barabási. *Network Science*. Cambridge University Press, 2015. URL: http://barabasi.com/networksciencebook/ (cited on pages 1, 3).

[18] A.-L. Barabási and R. Albert. 'Emergence of scaling in random networks'. In: *Science* 286.5439 (1999), p. 509. DOI: 10.1126/science.286.5439.509 (cited on pages 1, 19).

[19] M. Barthélemy. 'Spatial networks'. In: *Physics Reports* 499.1-3 (2011), pp. 1–101. DOI: 10.1016/j.physrep.2010.11.002 (cited on pages 66, 69).

[20] F. Battiston, V. Nicosia, and V. Latora. 'Structural measures for multiplex networks'. In: *Physical Review E* 89.3 (2014), p. 032804. DOI: 10.1103/PhysRevE.89.032804 (cited on page 97).

[21] S. Battiston, G. Caldarelli, and A. Garas, eds. *Multiplex and Multilevel Networks*. Oxford University Press, 2018. DOI: 10.1093/oso/9780198809456.001.0001 (cited on page 97).

[22] A. Bavelas. 'Communication patterns in task-oriented groups'. In: *Journal of the Acoustical Society of America* 22.6 (1950), pp. 725–730. DOI: 10.1121/1.1906679 (cited on page 75).

[23] M. Berlingerio, M. Coscia, and F. Giannotti. 'Finding redundant and complementary communities in multidimensional networks'. In: *Conference on Information and Knowledge Management*. 2011. DOI: 10.1145/2063576.2063921 (cited on page 104).

[24] M. Berlingerio, F. Pinelli, and F. Calabrese. 'ABACUS: frequent pAttern mining-BAsed Community discovery in mUltidimensional networkS'. In: *Data Mining and Knowledge Discovery* 27.3 (2013), pp. 294–320. DOI: 10.1007/s10618-013-0331-0 (cited on page 104).

[25] K. Bimpikis, A. Ozdaglar, and E. Yildiz. 'Competitive Targeted Advertising over Networks'. In: *Operations Research* 64.3 (2016), pp. 705–720. DOI: 10.1287/opre.2015.1430 (cited on pages 97, 98).

[26] P. Blanchard, F. Bühlmann, and J.-A. Gauthier, eds. *Advances in Sequence Analysis: Theory, Method, Applications*. Vol. 2. Life Course Research and Social Policies. Springer, 2014. DOI: 10.1007/978-3-319-04969-4 (cited on page 49).

[27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. 'Fast unfolding of communities in large networks'. In: *Journal of Statistical Mechanics* 2008.10 (2008), P10008. DOI: 10.1088/1742-5468/2008/10/P10008 (cited on page 8).

[28] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, I. Romance M. Sendiña-Nadal, Z. Wang, and M. Zanin. 'The structure and dynamics of multilayer networks'. In: *Physics Reports* 544.1 (2014), pp. 1–122. DOI: 10.1016/j.physrep.2014.07.001 (cited on page 97).

[29] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. 'Complex networks: structure and dynamics'. In: *Physics Reports* 424.4-5 (2006), pp. 175–308. DOI: 10.1016/j.physrep.2005.10.009 (cited on pages 1, 3).

[30] P. F. Bonacich. 'Power and centrality: A family of measures'. In: *American Journal of Sociology* 92.5 (1987), pp. 1170–1182. DOI: 10.1086/228631 (cited on page 46).

[31] S. P. Borgatti, M. G. Everett, and J. C. Johnson. *Analyzing Social Networks*. Sage, 2013. URL: https://uk.sagepub.com/en-gb/eur/analyzing-social-networks/book255068 (cited on page 3).

[32] V. S. Borkar. *Stochastic approximation - A Dynamical Systems Viewpoint*. Vol. 48. Texts and Readings in Mathematics. Hindustan Book Agency, 2008. DOI: 10.1007/978-93-86279-38-5 (cited on page 99).

[33] V. S. Borkar and A. Karnik. 'Controlled gossip'. In: *49th Annual Allerton Conference on Communication, Control, and Computing*. IEEE. 2011, pp. 707–711. DOI: 10.1109/Allerton.2011.6120237 (cited on page 98).

[34] V. S. Borkar, J. Nair, and N. Sanketh. 'Manufacturing consent'. In: *48th Annual Allerton Conference on Communication, Control, and Computing*. 2010, pp. 1550–1555. DOI: 10.1109/TAC.2014.2349591 (cited on pages 97, 98).

[35] K. Börner, S. Sanyal, and A. Vespignani. 'Network science'. In: *Annual Review of Information Science and Technology* 41.1 (2007), pp. 537–607. DOI: 10.1002/aris.2007.1440410119 (cited on page 1).

[36] S. Bornholdt and H. G. Schuster. *Handbook of Graphs and Networks - From the Genome to the Internet*. Wiley-VCH, 2002. URL: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-3527606335.html (cited on page 3).

[37] X. Bost. 'A storytelling machine ? Automatic video summarization: the case of TV series'. PhD Thesis. Université d'Avignon et des Pays de Vaucluse, Laboratoire Informatique d'Avignon, 2016. ⟨tel-01637270⟩. URL: https://tel.archives-ouvertes.fr/tel-01637270 (cited on pages 50, 51, 61).

[38] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova. 'Clustering attributed graphs: models, measures and methods'. In: *Network Science* 3.3 (2015), pp. 408–444. DOI: 10.1017/nws.2015.9 (cited on page 9).

[39] U. Brandes and T. Erlebach, eds. *Network Analysis: Methodological Foundations*. Vol. 3418. Lecture Notes in Computer Science. Springer, 2005. DOI: 10.1007/b106453 (cited on page 3).

[40] U. Brandes, G. Robins, A. McRanie, and S. Wasserman. 'What is network science?' In: *Network Science* 1.1 (2013), pp. 1–15. DOI: 10.1017/nws.2013.2 (cited on page 1).

[41] R. Breiger, S. Boorman, and P. Arabie. 'An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling'. In: *Journal of Mathematical Psychology* 12.3 (1975), pp. 328–383. DOI: 10.1016/0022-2496(75)90028-0 (cited on page 103).

[42] R. Breiger and P. Pattison. 'Cumulated social roles: The duality of persons and their algebras'. In: *Social Networks* 8.3 (1986), pp. 215–256. DOI: 10.1016/0378-8733(86)90006-7 (cited on page 103).

[43] M. Brusco, P. Doreian, A. Mrvar, and D. Steinley. 'Two Algorithms for Relaxed Structural Balance Partitioning: Linking Theory, Models, and Data to Understand Social Network Phenomena'. In: *Sociological Methods & Research* 40.1 (2011), pp. 57–87. DOI: 10.1177/0049124110384947 (cited on page 90).

C

[44] G. Caldarelli and A. Vespignani. *Large Scale Structure and Dynamics of Complex Networks - From Information Technology to Finance and Natural Science*. World Scientific, 2007. DOI: 10.1142/6455 (cited on page 3).

[45] A. Cardillo, J. Gómez-Gardeñes, M. Zanin, M. Romance, D. Papo, F. del Pozo, and S. Boccaletti. 'Emergence of network features from multiplexity'. In: *Scientific Reports* 3 (2013), p. 1344. DOI: 10.1038/srep01344 (cited on page 103).

[46] D. Cartwright and F. Harary. 'Structural balance: A generalization of Heider's theory'. In: *Psychological Review* 63 (1956), pp. 277–293. DOI: 10.1037/h0046049 (cited on pages 82, 83).

[47] R. Cazabet and G. Rossetti. 'Challenges in Community Discovery on Temporal Networks'. In: *Temporal Network Theory*. Computational Social Sciences. Springer, 2019. Chap. 10, pp. 181–197. DOI: 10.1007/978-3-030-23495-9_10 (cited on page 7).

[48] E. Çeviker Gürakar. *Politics of Favoritism in Public Procurement in Turkey: Reconfigurations of Dependency Networks in the AKP Era*. Palgrave Macmillan, 2016. DOI: 10.1057/978-1-137-59185-2 (cited on page 114).

[49] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. 'Measuring User Influence in Twitter: The Million Follower Fallacy'. In: *International AAAI Conference on Web and Social Media*. Vol. 4. 2010, pp. 10–17. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14033 (cited on page 20).

[50] T. Chakraborty and R. Narayanam. 'Cross-layer betweenness centrality in multiplex networks with applications'. In: *32nd IEEE International Conference on Data Engineering*. 2016, pp. 397–408. DOI: 10.1109/ICDE.2016.7498257 (cited on page 97).

[51] V. Chapela, R. Criado, S. Moral, and M. Romance. 'Mathematical Foundations: Complex Networks and Graphs (A Review)'. In: *Intentional Risk Management through Complex Networks Analysis*. SpringerBriefs in Optimization. Springer, 2015. Chap. 2, pp. 9–36. DOI: 10.1007/978-3-319-26423-3_2 (cited on pages 1, 3).

[52] V. S. Chavan and S. S. Shylaja. 'Machine learning approach for detection of cyber-aggressive comments by peers on social media network'. In: *International Conference on Advances in Computing, Communications and Informatics*. 2015, pp. 2354–2358. DOI: 10.1109/ICACCI.2015.7275970 (cited on page 25).

[53] G. Chen, X. Wang, and X. Li. *Fundamentals of Complex Networks: Models, Structures and Dynamics*. Wiley, 2015. DOI: 10.1002/9781118718124 (cited on page 3).

[54] Y. Chen and S. Skiena. 'Building Sentiment Lexicons for All Major Languages'. In: *52nd Annual Meeting of the Association for Computational Linguistics*. 2014, pp. 383–389. DOI: 10.3115/v1/p14-2063 (cited on page 26).

[55] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec. 'Antisocial Behavior in Online Discussion Communities'. In: *9th International AAAI Conference on Web and Social Media*. 2015, pp. 61–70. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10469 (cited on page 25).

[56] B. Chollet. 'L'analyse des réseaux personnels dans les organisations : quelles données utiliser ?' In: *Finance Contrôle Stratégie* 11.1 (2008), pp. 105–130. URL: http://www.sietmanagement.fr/wp-content/uploads/2016/04/Chollet1.pdf (cited on page 10).

[57] P. Chunaev. 'Community detection in node-attributed social networks: a survey'. In: *Computer Science Review* 37 (2020), p. 100286. DOI: 10.1016/j.cosrev.2020.100286 (cited on page 9).

[58] A. Clauset and N. Eagle. 'Persistence and periodicity in a dynamic proximity network'. In: *DIMACS Workshop on Computational Methods for Dynamic Interaction Networks*. 2007. URL: http://arxiv.org/abs/1211.7343 (cited on page 4).

[59] A. Clauset, M. E. J. Newman, and C. Moore. 'Finding community structure in very large networks'. In: *Physical Review E* 70.6 (2004), p. 066111. DOI: 10.1103/PhysRevE.70.066111 (cited on page 87).

[60] R. Cohen and S. Havlin. *Complex Networks - Structure, Robustness and Function*. Cambridge University Press, 2010. URL: https://www.cambridge.org/gb/academic/subjects/physics/statistical-physics/complex-networks-structure-robustness-and-function (cited on page 3).

[61] J. Coleman, E. Katz, and H. Menzel. 'The Diffusion of an Innovation Among Physicians'. In: *Sociometry* 20.4 (1957), pp. 253–270. DOI: 10.2307/2785979 (cited on page 103).

[62] M. Coscia. *The Atlas for the Aspiring Network Scientist*. Michele Coscia, 2021. URL: https://www.networkatlas.eu/ (cited on pages 1–3).

[63] M. Coscia, G. Rossetti, D. Pennacchioli, D. Ceccarelli, and F. Giannotti. 'You know Because I Know: A multidimensional network approach to human resources problem'. In: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2013, pp. 434–441. DOI: 10.1145/2492517.2492537 (cited on page 97).

[64] J.-V. Cossu, E. Ferreira, K. Janod, J. Gaillard, and M. El-Bèze. 'NLP-Based Classifiers to Generalize Expert Assessments in E-Reputation'. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Lecture Notes in Computer Science. Springer, 2015, pp. 340–351. DOI: 10.1007/978-3-319-24027-5_37 (cited on page 21).

[65] J.-V. Cossu, J. Killian, E. Ferreira, J. Gaillard, and M. El-Bèze. 'LIA@Replab 2014'. In: *Working Notes for CLEF 2014 Conference*. Vol. 1180. CEUR Workshop Proceedings. 2014, pp. 1458–1467. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-CossuEt2014.pdf (cited on page 21).

[66] E. Cozzo, G. F. Arruda, F. A. Rodrigues, and Y. Moreno. *Multiplex Networks - Basic Formalism and Structural Properties*. SpringerBriefs in Complexity. Springer, 2018. DOI: 10.1007/978-3-319-92255-3 (cited on page 97).

[67] G. Csárdi and T. Nepusz. 'The igraph software package for complex network research'. In: *InterJournal* Complex Systems (2006), p. 1695. URL: http://www.interjournal.org/manuscript_abstract.php?361100992 (cited on pages 77, 101).

[68] L. Cui, S. Kumara, and R. Albert. 'Complex Networks: An Engineering View'. In: *IEEE Circuits and Systems Magazine* 10.3 (2010), pp. 10–25. DOI: 10.1109/MCAS.2010.937883 (cited on page 3).

D

[69] E. Danna, M. Fenelon, Z. Gu, and R. Wunderling. 'Generating Multiple Solutions for Mixed Integer Programming Problems'. In: *International Conference on Integer Programming and Combinatorial Optimization*. Vol. 4513. Lecture Notes in Computer Science. Springer, 2007, pp. 280–294. DOI: 10.1007/978-3-540-72792-7_22 (cited on page 91).

[70] B. DasGupta, G. A. Enciso, E. Sontag, and Y. Zhang. 'Algorithmic and complexity results for decompositions of biological networks into monotone subsystems'. In: *Biosystems* 9.1 (2007), pp. 161–178. DOI: 10.1016/j.biosystems.2006.08.001 (cited on page 83).

[71] M. S. Daskin. *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 2011. DOI: 10.1002/9781118032343 (cited on page 59).

[72] J. A. Davis. 'Clustering and structural balance in graphs'. In: *Human Relations* 20.2 (1967), pp. 181–187. DOI: 10.1177/001872676702000207 (cited on page 84).

[73] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall. 'Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems'. In: *Physical Review X* 5.1 (1 2015), p. 011027. DOI: 10.1103/PhysRevX.5.011027 (cited on page 103).

[74] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora. 'Structural reducibility of multilayer networks'. In: *Nature Communications* 6 (2015), p. 6864. DOI: 10.1038/ncomms7864 (cited on page 103).

[75] M. De Domenico, M. A. Porter, and A. Arenas. 'MuxViz: a tool for multilayer analysis and visualization of networks'. In: *Journal of Complex Networks* 3.2 (2 2015), pp. 159–176. DOI: 10.1093/comnet/cnu038 (cited on pages 101, 103).

[76] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas. 'Mathematical Formulation of Multilayer Networks'. In: *Physical Review X* 3.4 (2013), p. 041022. DOI: 10.1103/PhysRevX.3.041022 (cited on page 97).

[77] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas. 'Navigability of interconnected networks under random failures'. In: *Proceedings of the National Academy of Sciences* 11.23 (2014), pp. 8351–8356. DOI: 10.1073/pnas.1318469111 (cited on page 103).

[78] M. H. DeGroot. 'Reaching a consensus'. In: *Journal of the American Statistical Association* 69.345 (1974), pp. 118–121. DOI: 10.1080/01621459.1974.10480137 (cited on pages 97, 98).

[79] B. Delaunay. 'Sur la sphère vide. À la mémoire de Georges Voronoï'. In: *Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et naturelles* 6 (1934), pp. 793–800. URL: http://mi.mathnet.ru/eng/izv/y1934/i6/p793 (cited on page 72).

[80] E. Desmier, M. Plantevit, C. Robardet, and J-F. Boulicaut. 'Cohesive Co-Evolution Patterns in Dynamic Attributed Graphs'. In: *International Conference on Discovery Science*. Vol. 7569. Lecture Notes in Artificial Intelligence. 2012, pp. 110–124. DOI: 10.1007/978-3-642-33492-4_11 (cited on pages 37, 41, 46).

[81] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Conference of the North American Chapter of the Association for Computational Linguistics*. 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423 (cited on page 112).

[82] M. E. Dickison, M. Magnani, and L. Rossi. *Multilayer Social Networks*. Cambridge University Press, 2016. URL: http://multilayer.it.uu.se/book.html (cited on page 97).

[83] K. Dinakar, R. Reichart, and H. Lieberman. 'Modeling the detection of Textual Cyberbullying'. In: *5th International AAAI Conference on Weblogs and Social Media / Workshop on the Social Mobile Web*. 2011, pp. 11–17. URL: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/3841 (cited on page 25).

[84] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati. 'Hate speech detection with comment embeddings'. In: *24th international conference on world wide web*. 2015, pp. 29–30. DOI: 10.1145/2740908.2742760 (cited on page 25).

[85] P. Doreian. 'Reflections on Studying Signed Networks'. In: *Journal of Interdisciplinary Methodologies and Issues in Science* 2 (2017), pp. 2.1–2.14. DOI: 10.18713/JIMIS-170117-2-1 (cited on page 83).

[86] P. Doreian and A. Mrvar. 'A partitioning approach to structural balance'. In: *Social Networks* 18.2 (1996), pp. 149–168. DOI: 10.1016/0378-8733(95)00259-6 (cited on pages 84, 85).

[87] P. Doreian and A. Mrvar. 'Partitioning signed social networks'. In: *Social Networks* 31.1 (2009), pp. 1–11. DOI: 10.1016/j.socnet.2008.08.001 (cited on page 85).

[88] P. Doreian and A. Mrvar. 'Structural Balance and Signed International Relations'. In: *Journal of Social Structure* 16 (2015), pp. 1–49. URL: https://www.cmu.edu/joss/content/articles/volume16/DoreianMrvar.pdf (cited on page 83).

[89] S. N. Dorogovtsev and J. F. F. Mendes. 'The shortest path to complex networks'. In: *arXiv* cond-mat.stat-mech (2004), p. 0404593. URL: http://arxiv.org/abs/cond-mat/0404593 (cited on page 3).

[90] O. Droulers and B. Roullet. 'Emergence du neuromarketing : apports et perspectives pour les praticients et les chercheurs'. In: *Décisions Marketing* 46 . (2007), pp. 9–22. URL: https://www.editions-ems.fr/revues/decisions-marketing/articlerevue/130-emergence-du-neuromarketing-apports-et-perspectives-pour-les-praticiens-et-les-chercheurs.html (cited on page 10).

E

[91] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. URL: https://www.cs.cornell.edu/home/kleinber/networks-book/ (cited on page 3).

[92] K. Erciyes. *Complex Networks: An Algorithmic Perspective*. CRC Press, 2014. URL: http://dl.acm.org/citation.cfm?id=2678067 (cited on page 3).

[93] P. Esmailian, S. E. Abtahi, and M. Jalili. 'Mesoscopic analysis of online social networks: The role of negative ties'. In: *Physical Review E* 90.4 (2014), p. 042817. DOI: 10.1103/PhysRevE.90.042817 (cited on pages 86, 87).

[94] E. Estrada. *The Structure of complex networks: Theory and applications*. Oxford University Press, 2011. URL: http://strathprints.strath.ac.uk/34153/ (cited on page 3).

[95] E. Estrada. 'Rethinking structural balance in signed social networks'. In: *Discrete Applied Mathematics* 268 (2019), pp. 70–90. DOI: 10.1016/j.dam.2019.04.019 (cited on page 83).

[96] A. J. Evans. 'Complex spatial networks in application'. In: *Complexity* 16.2 (2010), pp. 11–19. DOI: 10.1002/cplx.20339 (cited on page 67).

[97] Y. Evrard, B. Pras, and E. Roux. *Market : Études et recherches en Marketing*. Dunod, 2000. URL: https://www.dunod.com/entreprise-economie/market-fondements-et-methodes-recherches-en-marketing (cited on pages 10, 13).

F

[98] D. Fayard and G. Plateau. 'An algorithm for the solution of the 0–1 knapsack problem'. In: *Computing* 28.3 (1982), pp. 269–287. DOI: 10.1007/BF02241754 (cited on page 61).

[99] M. Fazekas and I. J. Tóth. 'From corruption to state capture: A new analytical framework with empirical applications from Hungary'. In: *Political Research Quarterly* 69.2 (2016), pp. 320–334. DOI: 10.1177/1065912916639137 (cited on page 114).

[100] M. Ferrand. 'Genèse et transformations d'un espace urbain médiéval : étude géo-historique de la ville d'Avignon (XIIIe-XVe siècle)'. PhD Thesis. Avignon Université, 2021 (expected). URL: http://theses.fr/s185892 (cited on pages 17, 80).

[101] R. Figueiredo and G. Moura. 'Mixed integer programming formulations for clustering problems related to structural balance'. In: *Social Networks* 35.4 (2013), pp. 639–651. DOI: 10.1016/j.socnet.2013.09.002 (cited on page 85).

[102] L. da Fontoura Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. 'Characterization of complex networks: A survey of measurements'. In: *Advances in Physics* 56.1 (2007), pp. 167–242. DOI: 10.1080/00018730601170527 (cited on page 104).

[103] S. Fortunato. 'Community detection in graphs'. In: *Physics Reports* 486.3-5 (2010), pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002 (cited on pages 7, 104).

[104] P. Fournier-Viger, A. Gomariz, T. Gueniche, E. Mwamikazi, and R. Thomas. 'TKS: Efficient Mining of Top-K Sequential Patterns'. In: *International Conference on Advanced Data Mining and Applications*. Vol. 8346. Lecture Notes in Computer Science. 2013, pp. 109–120. DOI: 10.1007/978-3-642-53914-5_10 (cited on page 39).

[105] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng. 'SPMF: A Java Open-Source Pattern Mining Library'. In: *Journal of Machine Learning Research* 15 (2014), pp. 3569–3573. URL: https://jmlr.org/papers/volume15/fournierviger14a/fournierviger14a.pdf (cited on page 39).

[106] L. C. Freeman. 'Centrality in Social Networks I: Conceptual Clarification'. In: *Social Networks* 1.3 (1978), pp. 215–239. DOI: 10.1016/0378-8733(78)90021-7 (cited on page 46).

[107] T. M. J. Fruchterman and E. M. Reingold. 'Graph drawing by force-directed placement'. In: *Software: Practice and Experience* 21.11 (1991), pp. 1129–1164. DOI: 10.1002/spe.4380211102 (cited on page 80).

G

[108] M. T. Gastner and M. E. J. Newman. 'Shape and efficiency in spatial distribution networks'. In: *Journal of Statistical Mechanics* 01.1 (2006), P01015. DOI: 10.1088/1742-5468/2006/01/P01015 (cited on pages 70, 73).

[109] M. T. Gastner and M. E. J. Newman. 'The spatial structure of networks'. In: *European Physical Journal B* 49.2 (2006), pp. 247–252. DOI: 10.1140/epjb/e2006-00046-8 (cited on page 67).

[110] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis. 'Music tracking in audio streams from movies'. In: *10th IEEE Workshop on Multimedia Signal Processing*. 2008, pp. 950–955. DOI: 10.1109/MMSP.2008.4665211 (cited on page 60).

[111] T. Giorgino. 'Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package'. In: *Journal of Statistical Software* 31.1 (2009), pp. 1–24. DOI: 10.18637/jss.v031.i07 (cited on page 41).

[112] M. Girvan and M. E. J. Newman. 'Community structure in social and biological networks'. In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799 (cited on pages 7, 8).

[113] D. Greene, D. Doyle, and P. Cunningham. 'Tracking the Evolution of Communities in Dynamic Social Networks'. In: *International Conference on Advances in Social Networks Analysis and Mining*. 2010, pp. 176–183. DOI: 10.1109/ASONAM.2010.17 (cited on page 46).

[114] R. Guimerà and L. A. N Amaral. 'Functional cartography of complex metabolic networks'. In: *Nature* 433 (2005), pp. 895–900. DOI: 10.1038/nature03288 (cited on pages 7, 19).

[115] R. Guimerà and L. A. N. Amaral. 'Cartography of complex networks: modules and universal roles'. In: *Journal of Statistical Mechanics* 2005.2 (2005), P02001. DOI: 10.1088/1742-5468/2005/02/P02001 (cited on pages 13, 46).

H

[116] A. Halu, R. J. Mondragón, P. Panzarasa, and G. Bianconi. 'Multiplex PageRank'. In: *PLoS ONE* 8.10 (2013), e78293. DOI: 10.1371/journal.pone.0078293 (cited on page 97).

[117] W. L. Hamilton. *Graph Representation Learning*. Vol. 46. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2020. DOI: 10.2200/s01045ed1v01y202009aim046 (cited on page 112).

[118] A. Hanjalic, R. L. Lagendijk, and J. Biemond. 'Automated high-level movie segmentation for advanced video-retrieval systems'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 9.4 (1999), pp. 580–588. DOI: 10.1109/76.767124 (cited on page 59).

[119] F. Harary. 'On the notion of balance of a signed graph'. In: *Michigan Mathematical Journal* 2.2 (1953), pp. 143–146. DOI: 10.1307/mmj/1028989917 (cited on pages 83, 84).

[120] F. Harary. *Graph Theory*. Addison-Wesley, 1969. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/705364.pdf (cited on page 46).

[121] F. Harary, R. Z. Norman, and D. Cartwright. *Structural models: An introduction to the theory of directed graphs*. John Wiley, 1965. URL: https://books.google.fr/books/about/Structural_Models.html?id=DbY9AAAAIAAJ (cited on page 84).

[122] F. Heider. 'Attitudes and cognitive organization'. In: *Journal of Psychology* 21.1 (1946), pp. 107–112. DOI: 10.1080/00223980.1946.9917275 (cited on pages 2, 82, 83).

[123] P. M. Hess. 'Measures of Connectivity'. In: *Places* 11 (1997), pp. 58–65. URL: https://escholarship.org/uc/item/9599t9f1 (cited on pages 70, 73).

[124] P. Holme. 'Modern temporal network theory: a colloquium'. In: *European Physical Journal B* 88.9 (2015). DOI: 10.1140/epjb/e2015-60657-4 (cited on page 36).

[125] P. Holme and J. Saramäki. 'Temporal networks'. In: *Physics Reports* 519.3 (2012), pp. 97–125. DOI: 10.1016/j.physrep.2012.03.001 (cited on page 37).

[126] P. Holme and J. Saramäki. *Temporal Networks*. Understanding Complex Systems. Springer, 2013. DOI: 10.1007/978-3-642-36461-7 (cited on page 37).

[127] P. Holme and J. Saramäki, eds. *Temporal Network Theory*. Computational Social Sciences. Springer, 2019. DOI: 10.1007/978-3-030-23495-9 (cited on page 37).

[128] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 'Deceiving Google's Perspective API Built for Detecting Toxic Comments'. In: *arXiv* cs.LG (2017), p. 1702.08138. URL: https://arxiv.org/abs/1702.08138 (cited on page 25).

[129] Z. Huang and Y. Qiu. 'A multiple-perspective approach to constructing and aggregating Citation Semantic Link Network'. In: *Future Generation Computer Systems* 26.3 (2010), pp. 400–407. DOI: 10.1016/j.future.2009.07.006 (cited on page 83).

[130] L. Hubert and P. Arabie. 'Comparing partitions'. In: *Journal of Classification* 2.1 (1985), pp. 193–218. DOI: 10.1007/BF01908075 (cited on page 12).

I

[131] R. Interdonato, M. Atzmueller, S. Gaito, R. Kanawati, C. Largeron, and A. Sala. 'Feature-rich networks: going beyond complex network topologies'. In: *Applied Network Science* 4 (2019), p. 4. DOI: 10.1007/s41109-019-0111-x (cited on page 3).

J

[132] M. O. Jackson. *Social and economic networks*. Vol. 3. Princeton University Press Princeton, 2008. URL: https://press.princeton.edu/books/paperback/9780691148205/social-and-economic-networks (cited on page 97).

[133] M. Joshi, O. Levy, D. S. Weld, and L. Zettlemoyer. 'BERT for Coreference Resolution: Baselines and Analysis'. In: *Conference on Empirical Methods in Natural Language Processing / 9th International Joint Conference on Natural Language Processing*. 2019, pp. 5803–5808. DOI: 10.18653/v1/D19-1588 (cited on page 114).

K

[134] R. Kanawati. 'Seed-Centric Approaches for Community Detection in Complex Networks'. In: *International Conference on Social Computing and Social Media*. Vol. 8531. Lecture Notes in Computer Science. 2014, pp. 197–208. DOI: 10.1007/978-3-319-07632-4_19 (cited on page 7).

[135] A. R. Kansal and S. Torquato. 'Globally and locally minimal weight spanning tree networks'. In: *Physica A* 301 (1-4 2001), pp. 601–619. DOI: 10.1016/S0378-4371(01)00430-7 (cited on pages 70, 73).

[136] B. Kapferer. *Strategy and transaction in an African factory*. Manchester, UK: Manchester University Press, 1972. URL: https://books.google.fr/books?id=fcXnAAAAIAAJ (cited on page 103).

[137] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, 1990. DOI: 10.1002/9780470316801 (cited on pages 91, 105).

[138] H. Khosravi and B. Bina. 'A Survey on Statistical Relational Learning'. In: *Canadian Conference on Artificial Intelligence*. Vol. 6085. Lecture Notes in Artificial Intelligence. 2010, pp. 256–268. DOI: 10.1007/978-3-642-13059-5_25 (cited on page 9).

[139] J. Kim and T. Wilhelm. 'What is a complex graph?' In: *Physica A* 387.11 (2008), pp. 2637–2652. DOI: 10.1016/j.physa.2008.01.015 (cited on page 1).

[140] Y.-M. Kim, J. Velcin, S. Bonnevay, and M.-A. Rizoiu. 'Temporal Multinomial Mixture for Instance-Oriented Evolutionary Clustering'. In: *European Conference on Information Retrieval*. Lecture Notes in Computer Science. Springer, 2015, pp. 593–604. DOI: 10.1007/978-3-319-16354-3_66 (cited on page 22).

[141] M. Kivelä, A. Arenas, M. Barthélemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. 'Multilayer Networks'. In: *Journal of Complex Networks* 2.3 (2014), pp. 203–271. DOI: 10.1093/comnet/cnu016 (cited on pages 96, 97, 104).

[142] D. Knoke and J. Wood. *Organized for action: Commitment in voluntary associations*. New Brunswick, NJ, USA: Rutgers University Press, 1981. URL: https://books.google.fr/books?vid=ISBN0813509114 (cited on page 103).

[143] T. Kolda and B. W. Bader. 'The TOPHITS model for higher-order web link analysis'. In: *SIAM Data Mining Conference Workshop on Link Analysis, Counterterrorism and Security*. 2006. URL: http://www.siam.org/meetings/sdm06/workproceed/Link%20Analysis/21Tamara_Kolda_SIAMLACS.pdf (cited on page 97).

[144] I. Koprinska and S. Carrato. 'Temporal video segmentation: A survey'. In: *Signal Processing: Image Communication* 16.5 (2001), pp. 477–500. DOI: 10.1016/S0923-5965(00)00011-4 (cited on page 60).

[145] J. R. Krebs and N. B. Davies, eds. *Behavioural Ecology: An Evolutionary Approach*. Wiley-Blackwell, 1993. URL: https://www.wiley.com/en-us/Behavioural+Ecology:+An+Evolutionary+Approach,+4th+Edition-p-9781444313628 (cited on page 67).

[146] J. Kunegis. *Handbook of Network Analysis*. Tech. rep. KONECT - The Koblenz Network Collection, 2015. URL: http://konect.uni-koblenz.de/downloads/konect-handbook.pdf (cited on page 3).

[147] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. de Luca, and S. Albayrak. 'Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization'. In: *SIAM International Conference on Data Mining*. 2010, pp. 559–570. DOI: 10.1137/1.9781611972801.49 (cited on page 112).

L

[148] R. Lambiotte. 'Rich gets simpler'. In: *Proceedings of the National Academy of Sciences* 113.36 (2016), pp. 9961–9962. DOI: 10.1073/pnas.1612364113 (cited on pages 36, 37).

[149] A. Lancichinetti and S. Fortunato. 'Consensus clustering in complex networks'. In: *Scientific Reports* 2 (2012), p. 336. DOI: 10.1038/srep00336 (cited on pages 92, 106).

[150] A. Lancichinetti, M. Kivelä, J. Saramäki, and S. Fortunato. 'Characterizing the Community Structure of Complex Networks'. In: *PLoS ONE* 5.8 (2010), e11976. DOI: 10.1371/journal.pone.0011976 (cited on pages 8, 13, 44, 46).

[151] A. Lancichinetti, F. Radicchi, and J. J. Ramasco. 'Statistical significance of communities in networks'. In: *Physical Review E* 81.4 (2010), p. 046110. DOI: 10.1103/PhysRevE.81.046110 (cited on page 13).

[152] E. Lazega. *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*. Oxford, UK: Oxford University Press, 2001. DOI: 10.1093/acprof:oso/9780199242726.001.0001 (cited on page 103).

[153] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. 'Microscopic evolution of social networks'. In: *International Conference on Knowledge Discovery and Data Mining*. ACM, 2008, pp. 462–470. DOI: 10.1145/1401890.1401948 (cited on pages 8, 44).

[154] J. Leskovec, D. Huttenlocher, and J. Kleinberg. 'Signed Networks in Social Media'. In: *ACM SIGCHI Conference on Human Factors in Computing Systems*. 2010, pp. 1361–1370. DOI: 10.1145/1753326.1753532 (cited on page 93).

[155] J. Leskovec, J. Kleinberg, and C. Faloutsos. 'Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations'. In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2005, pp. 177–187. DOI: 10.1145/1081870.1081893 (cited on page 37).

[156] D. Levinson and A. El-Geneidy. 'The minimum circuity frontier and the journey to work'. In: *Regional Science and Urban Economics* 39 (6 2009), pp. 732–738. DOI: 10.1016/j.regsciurbeco.2009.07.003 (cited on page 73).

[157] M. Levorato, L. Drummond, Y. Frota, and R. Figueiredo. 'An ILS algorithm to evaluate structural balance in signed social networks'. In: *30th Annual ACM Symposium on Applied Computing*. 2015, pp. 1117–1122. DOI: 10.1145/2695664.2695689 (cited on pages 87, 88).

[158] A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang, and A.-L. Barabási. 'The fundamental advantages of temporal networks'. In: *Science* 358.6366 (2017), pp. 1042–1046. DOI: 10.1126/science.aai7488 (cited on pages 36, 37).

[159] Z. Liu, Y. Lin, and M. Sun. *Representation Learning for Natural Language Processing*. Springer, 2020. DOI: 10.1007/978-981-15-5573-2 (cited on page 112).

[160] R. Louf, P. Jensen, and M. Barthelemy. 'Emergence of hierarchy in cost-driven growth of spatial networks'. In: *Proceedings of the National Academy of Sciences of the USA* 110 (22 2013), pp. 8824–8829. DOI: 10.1073/pnas.1222441110 (cited on pages 70, 73).

M

[161] N. R. Mabroukeh and C. I. Ezeife. 'A taxonomy of sequential pattern mining algorithms'. In: *ACM Computing Surveys* 43.1 (2010), pp. 1–41. DOI: 10.1145/1824795.1824798 (cited on pages 37, 39, 48).

[162] M. MacMahon and D. Garlaschelli. 'Community detection for correlation matrices'. In: *Physical Review X* 5.2 (2015), p. 021006. DOI: 10.1103/PhysRevX.5.021006 (cited on page 83).

[163] M. Magnani, B. Micenkova, and L. Rossi. 'Combinatorial analysis of multiple networks'. In: *arXiv* cs.SI (2013), p. 1303.4986. URL: http://arxiv.org/abs/1303.4986 (cited on pages 97, 103).

[164] M. Magnani and L. Rossi. 'The ML-Model for Multi-layer Social Networks'. In: *International Conference on Advances in Social Networks Analysis and Mining*. 2011, pp. 5–12. DOI: 10.1109/ASONAM.2011.114 (cited on page 97).

[165] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511809071 (cited on page 105).

[166] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís. 'Named Entity Recognition: Fallacies, challenges and opportunities'. In: *Computer Standards and Interfaces* 35.5 (2013), pp. 482–489. DOI: 10.1016/j.csi.2012.09.004 (cited on page 34).

[167] V. Martínez, F. Berzal, and J.-C. Cubero. 'A Survey of Link Prediction in Complex Networks'. In: *ACM Computing Surveys* 49.4 (2017), p. 69. DOI: 10.1145/3012704 (cited on page 48).

[168] N. Masuda and R. Lambiotte. *A Guide to Temporal Networks*. World Scientific, 2016. DOI: 10.1142/9781786341150 (cited on pages 36, 37).

[169] R. McDonald. 'A Study of Global Inference Algorithms in Multi-document Summarization'. In: *European Conference on Information Retrieval*. Vol. 4425. Lecture Notes in Computer Science. Springer, 2007, pp. 557–564. DOI: 10.1007/978-3-540-71496-5_51 (cited on page 61).

[170] D. W. McMillan and D. M. Chavis. 'Sense of community: A definition and theory'. In: *Journal of Community Psychology* 14.1 (1986), pp. 6–23. DOI: 10.1002/1520-6629(198601)14:1<6::AID-JCOP2290140103>3.0.CO;2-I (cited on page 8).

[171] M. Meilă. 'Comparing Clusterings by the Variation of Information'. In: *16th Annual Conference on Learning Theory / 7th Kernel Workshop*. Vol. 2777. Lecture Notes in Computer Science. Springer, 2003, pp. 173–187. DOI: 10.1007/978-3-540-45167-9_14 (cited on page 91).

[172] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 'Network motifs: simple building blocks of complex networks'. In: *Science* 298.5594 (2002), pp. 824–827. DOI: 10.1126/science.298.5594.824 (cited on page 1).

[173] H. Mubarak, K. Darwish, and W. Magdy. 'Abusive Language Detection on Arabic Social Media'. In: *1st Workshop on Abusive Language Online*. 2017, pp. 52–56. URL: http://www.aclweb.org/anthology/W/W17/W17-30.pdf (cited on page 25).

[174] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. 'Community Structure in Time-Dependent, Multiscale, and Multiplex Networks'. In: *Science* 328.5980 (2010), pp. 876–878. DOI: 10.1126/science.1184819 (cited on page 104).

N

[175] Z. P. Neal. 'A sign of the times? Weak and strong polarization in the U.S. Congress, 1973–2016'. In: *Social Networks* 60 (2020), pp. 103–112. DOI: 10.1016/j.socnet.2018.07.007 (cited on page 83).

[176] M. E. J. Newman. 'Assortative Mixing in Networks'. In: *Physical Review Letters* 89.20 (2002), p. 208701. DOI: 10.1103/PhysRevLett.89.208701 (cited on page 1).

[177] M. E. J. Newman. 'Mixing patterns in networks'. In: *Physical Review E* 67 (2003), p. 026126. DOI: 10.1103/PhysRevE.67.026126 (cited on page 13).

[178] M. E. J. Newman. 'The structure and function of complex networks'. In: *SIAM Review* 45 (2003), pp. 167–256. DOI: 10.1137/S003614450342480 (cited on pages 1, 3).

[179] M. E. J. Newman. 'Modularity and community structure in networks'. In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582. DOI: 10.1073/pnas.0601602103 (cited on pages 1, 8).

[180] M. E. J. Newman and M. Girvan. 'Finding and evaluating community structure in networks'. In: *Physical Review E* 69.2 (2004), p. 026113. DOI: 10.1103/PhysRevE.69.026113 (cited on page 87).

[181] M. K. Ng, X. Li, and Y. Ye. 'MultiRank: Co-Ranking for Objects and Relations in Multi-Relational Data'. In: *17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 1217–1225. DOI: 10.1145/2020408.2020594 (cited on page 97).

[182] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. 'A Review of Relational Machine Learning for Knowledge Graphs'. In: *Proceedings of the IEEE* 104.1 (2016), pp. 11–33. DOI: 10.1109/JPROC.2015.2483592 (cited on page 9).

O

[183] A. J. O'Malley and J.-P. Onnela. 'Topics in social network analysis and network science'. In: *arXiv* physics.soc-ph (2014), p. 1404.0067. URL: http://arxiv.org/abs/1404.0067 (cited on page 3).

[184] S. O'Sullivan and J. Morrall. 'Walking Distances to and from Light-Rail Transit Stations'. In: *Journal of the Transportation Research Board* 1538 (1996), pp. 19–26. DOI: 10.3141/1538-03 (cited on page 70).

[185] R. Ohme, D. Reykowska, D. Wiener, and A. Choromanska. 'Application of frontal EEG asymmetry to advertising research'. In: *Journal of Economic Psychology* 31.5 (2010), pp. 785–793. DOI: 10.1016/j.joep.2010.03.008 (cited on page 10).

[186] Organisation for Economic Co-operation and Development. *Compendium of good practices on the use of open data for Anti-corruption: Towards data-driven public sector integrity and civic auditing*. Tech. rep. OECD, 2017. URL: https://www.oecd.org/gov/digital-government/g20-oecd-compendium.pdf (cited on page 114).

[187] G. K. Orman, N. Türe, S. Balcisoy, and H. A. Boz. 'Finding proper time intervals for dynamic network extraction'. In: *Journal of Statistical Mechanics* 2021 (2021), p. 033414. DOI: 10.1088/1742-5468/abed45 (cited on page 3).

P

[188] G. Palla, A.-L. Barabási, and T. Vicsek. 'Quantifying social group evolution'. In: *Nature* 446.7136 (2007), pp. 664–667. DOI: 10.1038/nature05670 (cited on page 41).

[189] R. E. Park and E. W. Burgess. 'The Growth of the City: An Introduction to a Research Project'. In: *The City*. University of Chicago Press, 1925, pp. 47–62. DOI: 10.7208/9780226636641 (cited on page 68).

[190] P. Parlebas. *Sociométrie, réseaux et communication*. Presses Universitaires de France, 1992. URL: https://www.puf.com/content/Sociom%C3%A9trie_r%C3%A9seaux_et_communication (cited on page 10).

[191] N. Passas. *Corruption in the procurement process/outsourcing government functions: Issues, case studies, implications*. Tech. rep. Institute for Fraud Prevention, Northeastern University, 2007. URL: https://pdf4pro.com/view/corruption-in-the-procurement-process-1b90ae.html (cited on page 114).

[192] M. Poesio, R. Stuckardt, and Y. Versley. *Anaphora Resolution: Algorithms, Resources, and Applications*. Theory and Applications of Natural Language Processing. Springer, 2016. DOI: 10.1007/978-3-662-47909-4 (cited on page 34).

[193] P. Pons and M. Latapy. 'Computing communities in large networks using random walks'. In: *International Symposium on Computer and Information Sciences*. Vol. 3733. Lecture Notes in Computer Science. 2005, pp. 284–293. DOI: 10.1007/11569596_31 (cited on page 87).

[194] S. Porta, P. Crucitti, and V. Latora. 'The Network Analysis of Urban Streets: A Primal Approach'. In: *Environment and Planning B* 33 (5 2006), pp. 705–725. DOI: 10.1068/b32045 (cited on pages 69, 73).

[195] M. A. Porter, J.-P. Onnela, and J. P. Mucha. 'Communities in Networks'. In: *Notices of the American Mathematical Society* 56.9 (2009), p. 1082. URL: http://www.ams.org/notices/200909/rtx090901082p.pdf (cited on page 7).

R

[196] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. 'Defining and identifying communities in networks'. In: *Proceedings of the National Academy of Sciences* 101.9 (2004), pp. 2658–2663. DOI: 10.1073/pnas.0400054101 (cited on page 8).

[197] G. Ramírez-de-la-Rosa, E. Villatoro-Tello, H. Jiménez-Salazar, and C. Sánchez-Sánchez. 'Towards Automatic Detection of User Influence in Twitter by Means of Stylistic and Behavioral Features'. In: *Mexican International Conference on Artificial Intelligence*. Vol. 8856. Lecture Notes in Computer Science. Springer, 2014, pp. 245–256. DOI: 10.1007/978-3-319-13647-9_23 (cited on pages 21, 23).

[198] A. Rao, N. Spasojevic, Z. Li, and T. Dsouza. 'Klout Score: Measuring influence across multiple social networks'. In: *IEEE International Conference on Big Data*. 2015, pp. 2282–2289. DOI: 10.1109/bigdata.2015.7364017 (cited on page 20).

[199] A. Rapoport. 'Mathematical models of social interaction'. In: *Handbook of Mathematical Psychology*. Vol. 2. John Wiley & Sons, 1963. Chap. 14, pp. 493–580. URL: https://archive.org/details/handbookofmathem017893mbp (cited on page 83).

[200] A. Reggiani and P. Nijkamp, eds. *Complexity and Spatial Networks - In Search of Simplicity*. Advances in Spatial Science. Springer, 2009. DOI: 10.1007/978-3-642-01554-0 (cited on page 66).

[201] F. Roethlisberger and W. Dickson. *Management and the worker*. Cambridge, UK: Cambridge University Press, 1939. URL: https://psycnet.apa.org/record/1940-00509-000 (cited on page 103).

[202] M. Rosvall and C. T. Bergstrom. 'An information-theoretic framework for resolving community structure in complex networks'. In: *Proceedings of the National Academy of Sciences* 104.18 (2007), pp. 7327–7331. DOI: 10.1073/pnas.0611034104 (cited on page 8).

[203] M. Rosvall and C. T. Bergstrom. 'Maps of random walks on complex networks reveal community structure'. In: *Proceedings of the National Academy of Sciences* 105.4 (2008), p. 1118. DOI: 10.1073/pnas.0706851105 (cited on page 87).

[204] P. J. Rousseeuw. 'Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis'. In: *Journal of Computational and Applied Mathematics* 20.1 (1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7 (cited on pages 13, 91, 105).

S

[205] G. Sabidussi. 'The centrality index of a graph'. In: *Psychometrika* 31.4 (1966), pp. 581–603. DOI: 10.1007/BF02289527 (cited on page 46).

[206] S. Sarawagi. 'Information extraction'. In: *Foundations and Trends in Databases* 1.3 (2008), pp. 261–377. DOI: 10.1561/1900000003 (cited on page 34).

[207] H. Sayama. *Introduction to the Modeling and Analysis of Complex Systems*. OpenSUNY, 2015. URL: http://textbooks.opensuny.org/introduction-to-the-modeling-and-analysis-of-complex-systems/ (cited on pages 1, 3).

[208] S. E. Schaeffer. 'Graph clustering'. In: *Computer Science Review* 1.1 (2007), pp. 27–64. DOI: 10.1016/j.cosrev.2007.05.001 (cited on page 7).

[209] R. J. Senter and E. A. Smith. *Automated Readability Index*. Technical Report AMRL-TR-6620. Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, 1967. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/667273.pdf (cited on page 26).

[210] L. Solá, M. Romance, R. Criado, J. Flores, A. García del Amo, and S. Boccaletti. 'Eigenvector centrality of nodes in multiplex networks'. In: *Chaos* 23.3 (2013), p. 033131. DOI: 10.1063/1.4818544 (cited on pages 97, 101).

[211] A. Solé-Ribalta, M. de Domenico, S. Gómez, and A. Arenas. 'Centrality rankings in multiplex networks'. In: *ACM conference on Web science*. 2014, pp. 149–155. DOI: 10.1145/2615569.2615687 (cited on page 97).

[212] A. Solé-Ribalta, M. de Domenico, S. Gómez, and A. Arenas. 'Random walk centrality in interconnected multilayer networks'. In: *Physica D* 323-324 (2016), pp. 73–79. DOI: 10.1016/j.physd.2016.01.002 (cited on page 97).

[213] A. Soler and R. Zaera. 'The secondary frame in spider orb webs: the detail that makes the difference'. In: *Scientific Reports* 6.1 (2016), p. 31265. DOI: 10.1038/srep31265 (cited on page 71).

[214] K. Sparck Jones. 'A statistical interpretation of term specificity and its application in retrieval'. In: *Journal of Documentation* 28.1 (1972), pp. 11–21. DOI: 10.1108/eb026526 (cited on page 22).

[215] E. Spertus. 'Smokey: Automatic recognition of hostile messages'. In: *14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*. 1997, pp. 1058–1065. URL: http://dl.acm.org/citation.cfm?id=1867616 (cited on page 25).

[216] M. van Steen. *Graph Theory and Complex Networks: An introduction*. Maarten van Steen, 2010. URL: http://www.distributed-systems.net/gtcn/index.php/book-request (cited on page 3).

[217] A. Strehl and J. Ghosh. 'Cluster ensembles: A knowledge reuse framework for combining multiple partitions'. In: *Journal of Machine Learning Research* 3 (2002), pp. 583–617. DOI: 10.1162/153244303321897735 (cited on page 87).

T

[218] A. Tandon, A. Albeshri, V. Thayananthan, W. Alhalabi, F. Radicchi, and S. Fortunato. 'Community detection in networks using graph embeddings'. In: *Physical Review E* 103.2 (2021), p. 022316. DOI: 10.1103/PhysRevE.103.022316 (cited on page 112).

[219] J. Tang, M. Musolesi, C. Mascolo, V. Latora, and V. Nicosia. 'Analysing information flows and key mediators through temporal centrality metrics'. In: *3rd ACM EuroSys Workshop on Social Networks Systems*. 2010. DOI: 10.1145/1852658.1852661 (cited on page 37).

[220] A. Tealab. 'Time series forecasting using artificial neural networks methodologies: A systematic review'. In: *Future Computing and Informatics Journal* 3.2 (2018), pp. 334–340. DOI: 10.1016/j.fcij.2018.10.003 (cited on page 33).

[221] B. Thurman. 'In the office: Networks and coalitions'. In: *Social Networks* 2.1 (1979), pp. 47–63. DOI: 10.1016/0378-8733(79)90010-8 (cited on page 103).

[222] J.-M. Torres-Moreno. 'Artex is AnotheR TEXt summarizer'. In: *arXiv* cs.IR (2012), p. 1210.3312. URL: https://arxiv.org/abs/1210.3312 (cited on page 22).

[223] J.-M. Torres-Moreno, M. El-Bèze, P. Bellot, and F. Bechet. 'Opinion detection as a topic classification problem'. In: *Textual Information Access: Statistical Models*. John Wiley & Son, 2012. Chap. 9, pp. 337–368. DOI: 10.1002/9781118562796.ch9 (cited on page 22).

[224] M. Toyoda and M. Kitsuregawa. 'Extracting evolution of web communities from a series of web archives'. In: *14th ACM conference on Hypertext and hypermedia*. 2003, pp. 28–37. DOI: 10.1145/900051.900059 (cited on page 41).

[225] V. A. Traag, G. Krings, and P. van Dooren. 'Significant Scales in Community Structure'. In: *Scientific Reports* 3 (2013), p. 2930. DOI: 10.1038/srep02930 (cited on page 104).

[226] G. Tsoumakas and I. Katakis. 'Multi-Label Classification: An Overview'. In: *International Journal of Data Warehousing and Mining* 3.3 (2007), pp. 1–13. DOI: 10.4018/jdwm.2007070101 (cited on page 55).

[227] M. Tumminello, S. Micciché, F. Lillo, J. Varho, J. Piilo, and R. N. Mantegna. 'Community characterization of heterogeneous complex systems'. In: *Journal of Statistical Mechanics* 2011.1 (2011), P01019. DOI: 10.1088/1742-5468/2011/01/P01019 (cited on page 44).

V

[228] D. Vilares, M. Hermo, M. A. Alonso, C. Gómez-Rodríguez, and J. Vilares. 'LyS at CLEF RepLab 2014: Creating the State of the Art in Author Influence Ranking and Reputation Classification on Twitter'. In: *Working Notes for CLEF 2014 Conference*. Vol. 1180. CEUR Workshop Proceedings. 2014, pp. 1468–1478. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-VilaresEt2014.pdf (cited on pages 21, 24).

[229] I. Vragović, E. Louis, and A. Díaz-Guilera. 'Efficiency of informational transfer in regular and complex networks'. In: *Physical Review E* 71 (3 2005), p. 036122. DOI: 10.1103/PhysRevE.71.036122 (cited on pages 69, 73).

W

[230] Y. Wang, Y. Li, J. Fan, C. Ye, and M. Chai. 'A survey of typical attributed graph queries'. In: *World Wide Web* 24.1 (2020), pp. 297–346. DOI: 10.1007/s11280-020-00849-0 (cited on page 9).

[231] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Vol. 8. Structural Analysis in the Social Sciences. Cambridge University Press, 1994. URL: http://www.cambridge.org/zw/academic/subjects/sociology/sociology-general-interest/social-network-analysis-methods-and-applications (cited on pages 9, 97).

[232] D. J. Watts and S. H. Strogatz. 'Collective dynamics of 'small-world' networks'. In: *Nature* 393.6684 (1998), pp. 440–442. DOI: 10.1038/30918 (cited on pages 1, 19, 46).

[233] A. S. Waugh, L. Pei, J. H. Fowler, P. J. Mucha, and M. A. Porter. 'Party Polarization in Congress: A Network Science Approach'. In: *arXiv* physics.soc-ph (2009), p. 0907.3509. URL: http://arxiv.org/abs/0907.3509 (cited on page 104).

[234] T. A. Welch. 'A Technique for High-Performance Data Compression'. In: *Computer* 17.6 (1984), pp. 8–19. DOI: 10.1109/mc.1984.1659158 (cited on page 26).

X

[235] Z. Xu, Z. Ou, Q. Su, J. Yu, X. Quan, and Z. Lin. 'Embedding Dynamic Attributed Networks by Modeling the Evolution Processes'. In: *28th International Conference on Computational Linguistics*. 2020, pp. 6809–6819. DOI: 10.18653/v1/2020.coling-main.600 (cited on page 112).

Y

[236] X. Yan, J. Han, and R. Afshar. 'CloSpan: Mining Closed Sequential Patterns in Large Datasets'. In: *SIAM International Conference on Data Mining*. 2003, pp. 166–177. DOI: 10.1137/1.9781611972733.15 (cited on page 39).

[237] X. Ye and X. Liu, eds. *Cities as Spatial and Social Networks*. Human Dynamics in Smart Cities. Springer, 2019. DOI: 10.1007/978-3-319-95351-9 (cited on page 66).

[238] M. Yeung, B.-L. Yeo, and B. Liu. 'Segmentation of Video by Clustering and Graph Analysis'. In: *Computer Vision and Image Understanding* 71.1 (1998), pp. 94–109. DOI: 10.1006/cviu.1997.0628 (cited on page 60).

[239] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards. 'Detection of harassment on Web 2.0'. In: *WWW Workshop: Content Analysis in the Web 2.0*. 2009, pp. 1–7. URL: http://www.cse.lehigh.edu/~brian/pubs/2009/CAW2/ (cited on page 25).

Z

[240] J. Zhang, H. Zhang, C. Xia, and L. Sun. 'Graph-Bert: Only Attention is Needed for Learning Graph Representations'. In: *arXiv* cs.LG (2021), p. 2001.05140. URL: https://arxiv.org/abs/2001.05140 (cited on page 112).

[241] S. Zschokke. 'Radius construction and structure in the orb-web of Zilla diodia (Araneidae)'. In: *Journal of Comparative Physiology A* 186.10 (2000), pp. 999–1005. DOI: 10.1007/s003590000155 (cited on page 71).

[242] K. A. Zweig. *Network Analysis Literacy: A Practical Approach to the Analysis of Networks*. Lecture Notes in Social Networks. Springer, 2016. DOI: 10.1007/978-3-7091-0741-6 (cited on page 3).

Résumé

Le concept de *Réseau Complexe* (ou *Graphe de Terrain*) est généralement utilisé dans la littérature pour faire référence à un graphe représentant un système complexe du monde réel. Cela confère à ces graphes des propriétés topologiques qualifiées de non-triviales, qui les distinguent des graphes réguliers et aléatoires. Les plus connues sont les propriétés petit-monde et sans-échelle, dont la découverte a marqué le début d'un nouveau domaine de recherche appelé *Science des Réseaux*, et visant à étudier les réseaux complexes. Il s'agit d'un champ multidisciplinaire reposant sur de nombreux domaines pré-existants, en particulier la théorie des graphes, la sociologie quantitative, l'informatique, la recherche opérationnelle, la physique statistique, et bien sûr l'étude des systèmes complexes.

Ce point de départ basé sur la modélisation de systèmes réels fait de la science des réseaux une science des *données*. En tant que telle, son émergence est due non seulement à la convergence de travaux interdisciplinaires, mais aussi à la disponibilité des ressources nécessaires à la construction et à l'étude de larges collections de grands réseaux complexes, à savoir : une puissance computationnelle suffisante et l'accès à des données appropriées. En raison de cette dépendance aux données, la représentation de l'information est une problématique fondamentale de la science des réseaux. La façon dont les données décrivant le système considéré sont intégrées au graphe constituant son modèle est de la plus grande importance. Or, un graphe basique ne permet de modéliser qu'un seul type d'information : la présence ou l'absence de relation entre les objets constituant le système. Pour exploiter des données plus diverses, il est nécessaire d'étendre ce type de modèle, ce qui nous amène à la notion de *Réseau Enrichi*, qui est au cœur de cette thèse.

Dans ce manuscrit, je résume le travail que j'ai mené sur le sujet de ces réseaux enrichis. Dans la *première partie*, je considère les réseaux dont les sommets sont décrits par des attributs. Le Chapitre 2 s'intéresse à la détection de communautés, et présente une comparaison de sous-groupes de sommets identifiés sur la base de la structure du graphe et sur celle des attributs de ces nœuds. Le Chapitre 3 traite de deux problèmes de classification : le premier consiste à détecter des personnes influentes dans la vie réelle, en exploitant seulement des données décrivant leur activité sur un média social ; et le second porte sur l'identification de messages abusifs dans des salons de discussion en ligne.

La *deuxième partie* est dédiée aux réseaux dynamiques. Le Chapitre 4 propose deux méthodes tirant parti de méthodes de fouille de motifs séquentiels pour caractériser la dynamique de ces réseaux à deux niveaux. La première vise à les décrire au niveau microscopique (sommets), et la seconde au niveau mésoscopique (communautés). Dans le Chapitre 5, je présente une méthode basée sur la segmentation de graphes dynamiques, et visant à générer des résumés vidéo de séries télévisées.

La *troisième partie* couvre trois types de réseaux enrichis. Le Chapitre 6 est dédié aux réseaux spatiaux, et présente deux travaux portant sur la mesure de Rectitude. Le sujet du Chapitre 7 est le partitionnement de graphes signés et le concept d'équilibre structural. Le Chapitre 8 porte sur les réseaux multiplexes. J'y décris une mesure de centralité basée sur un modèle de diffusion d'opinion à travers plusieurs média sociaux, et une méthode de partitionnement de graphes permettant d'identifier plusieurs structures modulaires au sein du même réseau multiplexe.

Abstract

The concept of *Complex Network* is generally used in the literature to refer to a graph representing a real-world complex system. This confers such graphs so-called non-trivial topological properties that distinguish them from regular and random graphs. Among them, the most widely known are small-worldness and scale-freeness, whose study marked the beginning of a new research domain now called *Network Science*, and aiming at studying complex networks. It is a multidisciplinary field that relies largely on a number of pre-existing domains, in particular graph theory, quantitative sociology, computer science, operations research, statistical physics, and of course complex systems.

Network Science is mainly a *data* science, as its starting point is the modeling of real-world systems. As such, its emergence is due not only to the convergence of interdisciplinary efforts, but also to the availability of the resources required to build and study large and/or numerous complex networks: computational power and access to sizable datasets. Because of this fundamental reliance on data, information representation is a fundamental aspect of Network Science. The way the available data describing the considered system are included in the graph-based model is of the utmost importance. Yet, plain graphs are meant only to model one type of information: the presence or absence of relationships between the object constituting the system. To handle more diverse data, it is necessary to extend this framework, which leads us to the notion of *Feature-Rich* network that is at the core of this thesis.

In this manuscript, I summarize the research that I conducted on the topic of feature-rich networks. In the *first part*, I focus on vertex-attributed networks. Chapter 2 deals with community detection, and presents a comparison of vertex modules detected based on graph structure vs. attributes, using a student activity dataset collected during a ground survey. Chapter 3 tackles two classification problems based on attributed graphs. The first is the detection of persons which are influential in real-life, based only on data describing their activity on an online medium. The second is the identification of abusive messages in online chats.

The *second part* is dedicated to dynamic networks. Chapter 4 proposes two methods leveraging sequential pattern mining to characterize the dynamics of such networks at two distinct levels. The first targets the microscopic evolution of the network, whereas the second describes the network at a mesoscopic level. In Chapter 5, I present a method based on the segmentation of dynamic graphs, and aiming at generating video summaries of TV series.

The *third part* covers three types of networks. Chapter 6 is dedicated to spatial networks, presenting two works revolving around the Straightness measure. Chapter 7 is about the partitioning of signed networks and the concept of structural balance. Chapter 8 deals with multiplex networks. I describe a vertex centrality measure relying on a model of opinion diffusion in multiplex networks, and a graph partitioning method allowing to identify several modular structures for a single multiplex network.

AVIGNON UNIVERSITÉ