



Analyzing the local structure of large social networks

Alina Stoica Beck

► To cite this version:

Alina Stoica Beck. Analyzing the local structure of large social networks. Social and Information Networks [cs.SI]. Université Paris-Diderot - Paris VII, 2010. English. NNT: . tel-00987880

HAL Id: tel-00987880

<https://theses.hal.science/tel-00987880>

Submitted on 7 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE PARIS.DIDEROT (PARIS 7)

Ecole Doctorale de Sciences Mathématiques de Paris Centre

DOCTORAT
Informatique

Alina Mihaela STOICA

Analyse de la structure locale des grands réseaux sociaux

Analyzing the local structure of large social networks

Soutenue le 12 octobre 2010 devant le jury:

Rapporteurs:

Pierluigi CRESCENZI

Universita di Firenze

Patrick GALLINARI

UPMC (LIP6)

Examineurs:

Vincent BLONDEL

MIT

Renaud LAMBIOTTE

Imperial College London

Nicolas SCHABANEL

Paris-Diderot (LIAFA)

Directeurs:

Michel HABIB

Paris-Diderot (LIAFA)

Christophe PRIEUR

Paris-Diderot (LIAFA)

Zbigniew SMOREDA

Orange Labs



Acknowledgments

First of all, I would like to thank Patrick Gallinari and Pierluigi Crescenzi for having written the reports for my dissertation. Thank you for having accepted to write them in spite of all the constraints, the short time and the month of August.

I am also grateful to Vincent Blondel, Renaud Lambiotte and Nicolas Schabanel for being part of the jury of my PhD defense.

I am indebted to my academic supervisor, Michel Habib, for having accepted to lead this PhD thesis in spite of all the special conditions and to my industrial supervisor, Zbigniew Smoreda, for having put no "company" pressure on me, allowing me to lead freely my academic research.

I would like to thank all the SENSE team in Orange Labs for having created such a nice environment, ideal for a PhD student. I spent three very pleasant years in your company, I will surely miss it. You also made me appreciate the social sciences (which was a real challenge when I started my thesis). I still understand only too little of the subject, but I am much more open to such approaches. I believe that, as a person, I have learnt a lot in your company. Special thanks to Jean-Samuel Beuscart, for being such a great fan of the "new science of networks" and, therefore, of my work. I would like to thank Maryse Piart and Noelle Delgado (from LIAFA) for their kindness and help after each one of my work trips. I also thank Frédérique Legrand for her enthusiasm for my results; it is always nice to be appreciated by your boss!

Also, a lot of thanks to all the PhD students and to all the people who joined us for lunch at 12:00 instead of 12:30 (when I am much too hungry). Among all these people, Elodie Raimond has a special place since we have been together from the beginning of the 3 years, sharing all the joys and the disappointments of the PhD student's life. You are a great friend, I hope we will keep seeing each other after having left Orange.

On a more personal note, I am grateful to my family and especially to my parents for always being there for me, even if they are more than 2,000 km away. They have been great since I decided to come to France, they have even begun to learn French! I also thank them for being such enthusiastic supporters of what I do (in their world I am a star!), although it is highly undeserved. I also thank my dear friends Roxana, Consuela, Mihai and Dan who are like a family to me. Thanks to you I have always enjoyed my life in France.

Now I want to thank the three persons without whom I could not have done this PhD thesis. First of all, I am grateful to Christophe Prieur for guiding me throughout the three years, from the moment I applied for an internship at Orange Labs to my first paper,

throughout the accomplishments and the disappointments, and even to what became my future job. Thank you for always being there when I needed your help, for encouraging me and especially for calming me down in so many moments of stress. There are a lot of things I couldn't have done without your help.

Second, I address a lot of thanks to my colleague, coauthor and friend, Thomas Couronné. Thank you for helping me discover data mining, for being such a great promoter of my results and especially for making me work. Since I began to work with you I have doubled my productivity. You are a role model for me of hard work and dynamism. I hope we will keep working together, I enjoy it very much!

And last but certainly not least, I thank you, Jérôme, for all the love and the happiness you have brought into my life.

Contents

Contents	i
1 Introduction	3
1.1 Context and motivations	3
1.2 Thesis overview and contributions	8
1.2.1 Publications	9
I Overview and survey	11
2 Basic notions	15
2.1 Graph theory concepts	15
2.2 Data mining	17
3 Complex networks	23
3.1 Complex networks properties	24
3.2 Models of networks and random generation of networks	34
3.3 Identification of patterns in complex networks	37
4 Social networks	41
4.1 Questioning and advances	43
4.1.1 Social roles	44
4.2 Egocentred analysis	46
4.3 Phone communications	48
4.4 Online activities	51
4.5 Online activities vs. offline communications	56
4.6 Applications: Marketing and services	58
II Methods and Applications	61
5 Local structure of large networks	63
5.1 Definitions	63
5.2 Efficient graph characterization	65

5.3	A method for local structure analysis	68
5.4	Algorithmic aspects	70
5.5	Applications of the method	72
5.6	Comparison to other measures	75
5.7	Chapter conclusions	77
6	From online popularity to social linkage	79
6.1	Introduction	79
6.2	Data description	80
6.3	Analysis of the online popularity	80
6.4	Social network structures	84
6.5	Chapter conclusions	88
7	An analysis of a mobile phone graph	91
7.1	Introduction	91
7.2	Data description	91
7.3	Mobile phone graph	95
7.4	Characteristic patterns	97
7.5	A characterization of ego's contacts	100
7.6	Chapter conclusions	104
8	A local structure-based clustering of nodes	107
8.1	Introduction	107
8.2	A method for nodes clustering	108
8.2.1	Pattern-frequency equivalence	108
8.2.2	The issue of the degree	111
8.2.3	Pattern-frequency clustering of nodes	113
8.3	Clusters of individuals in the mobile phone network	115
8.4	Clusters versus age and gender	118
8.4.1	Age	118
8.4.2	Gender	121
8.5	Clusters versus intensity of communication	121
8.5.1	Basic statistics	121
8.5.2	Predicting the cluster from the communications	122
8.6	A typology of customers	125
8.7	Chapter conclusions	129
III	Conclusions	131
	Bibliography	150
A	Introduction (en français)	151

B	Structure locale des grands réseaux	157
B.1	Définitions	157
B.2	Caractérisation efficace de graphe	159
B.3	Une méthode pour l'analyse de la structure locale	162
B.4	Considérations algorithmiques	165
B.5	Applications de la méthode	168
B.6	Comparaison avec d'autres mesures	169
B.7	Conclusions du chapitre	172

List of Figures

3.1	Degree distribution plot in complex networks and in real networks.	25
3.2	A power-law distribution: the in-degree in an Epinions graph.	26
3.3	A distribution with exponential cutoff and a log-normal one.	27
3.4	Hop-plot and effective diameter in an Epinions graph.	28
3.5	Clustering coefficient	29
3.6	Connected components.	30
3.7	Communities in a coauthorship network.	32
3.8	An exemple of graph.	34
3.9	Network motifs found in biological and technological networks	40
4.1	Social capital	44
4.2	An example of graph.	45
4.3	Density of Flickr and Yahoo! 360 by week	55
4.4	Degree distribution in the mobile phone graph and in the Flickr graph. . . .	57
4.5	Schematic of the two-step flow model of influence	59
5.1	The set of patterns and their positions.	64
5.2	A graph (<i>a</i>), its patterns (<i>b</i>) and the position vectors of two vertices (<i>c</i>). . .	66
5.3	Pseudocode for algorithm <i>ESU</i> that lists all size- <i>k</i> subgraphs in a graph. . .	67
5.4	Two non-isomorphic connected graphs with 6 vertices	68
5.5	A vertex, its egocentred network and its patterns.	69
5.6	Three possible positions of a neighbor and the corresponding structures. . .	69
5.7	A position of a neighbor with weight 2 and the corresponding structure . .	70
5.8	An example for the difference between centrality and position vectors. . . .	76
5.9	Two networks with the same nb. of vertices, edges and clustering coefficient. 77	
6.1	SOM of the artists depending on their popularity properties.	82
6.2	The 5 clusters	83
6.3	The patterns with at most 4 vertices and their positions.	84
6.4	The average number of edges and isolated vertices.	85
6.5	The average nb. of isolated edges, triangles and 4-cliques	86
7.1	Mean call duration depending on caller and receiver gender.	92
7.2	Average nb. of calls and SMS as a function of user's age.	93

7.3	Average call duration as a function of user's age.	94
7.4	Average number of SMS depending on caller and receiver gender and age. .	94
7.5	Distribution of degree and nb. of triangles in the phone network.	95
7.6	The set of patterns and their positions.	97
7.7	Frequent patterns: definition 1.	99
7.8	Frequent patterns: definition 2.	99
7.9	Frequent patterns: definition 3.	101
7.10	The probability of occurrence of a vertex with rank r in the position i . . .	103
8.1	The 9 patterns with at most 4 vertices and at least one edge.	110
8.2	A vertex, its egocentred network and its patterns.	110
8.3	An example of 4 egocentred networks.	112
8.4	All the possible graphs with 4 and 5 vertices.	116
8.5	The distribution of the reduced population into the 6 clusters.	117
8.6	The probability of belonging to the 6 clusters by age	119
8.7	Hierarchical clustering of ages on distributions in the 6 clusters.	120
8.8	For each cluster, the distribution in the slices of values.	123
8.9	SOM of the Mobistar customers	126
8.10	The cells occupied by each cluster.	127
8.11	The 9 profiles produced by the Kohonen SOM.	128
B.1	L'ensemble de patterns et leurs positions.	158
B.2	Un graphe (a), ses patterns (b) et deux vecteurs de position (c).	160
B.3	Pseudocode pour l'algorithme <i>ESU</i> qui énumère tous les sous-graphes. . . .	161
B.4	Deux graphes connexes non-isomorphes avec 6 sommets.	162
B.5	Un sommet, son réseau égocentré et ses patterns.	163
B.6	Trois positions possibles d'un voisin et les structures correspondantes. . . .	164
B.7	La position d'un voisin avec poids 2 et la structure correspondante.	164
B.8	Un exemple de différence entre centralité et vecteurs de position.	171
B.9	Deux réseaux avec le même nb de noeuds, de liens et coef. de clustering. . .	171

List of Tables

3.1	Degree, betweenness and closeness centrality in an example graph.	34
4.1	Basic statistics in the mobile phone graph and in the Flickr graph.	58
5.1	Equivalent notions for a vertex.	75
6.1	Dataset properties	81
7.1	Basic statistics in the mobile phone network.	96
8.1	The pattern-frequency vectors of the egocentred networks in Figure 8.3. . .	112
8.2	The distribution of the reduced and total population into the 6 clusters. . .	118
8.3	The proportion of men and women in each cluster.	121
8.4	The proportion of correct predictions in the 6 clusters.	124
8.5	The different characteristics of the individuals in the 9 profiles	128
B.1	Notions équivalentes pour un sommet.	170

Chapter 1

Introduction

1.1 Context and motivations

The main interest of our research has been in analyzing the local structure of large social networks. How is a node connected to the network? How can we analyze the whole set of nodes of the network in a reasonable time? Does the way a node is connected say anything about the person represented by the node? Is there a correlation between the structure of the network surrounding an individual and their age, gender or practices (mobile phone uses, online popularity etc.)?

So the goal of this research is to characterize individuals by analyzing the social network in which they are embedded. Such a characterization is useful for instance for service providers, for whom the knowledge of their customers is very important. It is essential to know what services customers want and how their expectations evolve so that offers or advertisement can be adjusted and sent to people who are likely to react favorably to them.

In order to obtain such a **characterization of users**, one can adopt different approaches. One can use socio-demographic data as age, gender, job, location etc. Other information that can be used, which may be even more useful and reliable than socio-demographic one, is the traces left by customers while using various services. Mobile phone providers thus know how many times a day a person makes phone-calls, how long their conversations are, with how many different people etc. In the same way, developers of online platforms can also use traces of usage. For instance on a platform of social networking and sharing of photos and videos like Flickr (www.flickr.com), users can declare each other as contacts, upload photos or videos, make them public, write comments etc. One can use this information (amount of published content, comments, number of contacts etc.) as a characterization of each person's activity on the platform. Different users can then be proposed different services depending on their uses.

Nowadays, **traces of uses** are present everywhere and are generally easy to obtain. Almost everybody has a mobile phone, an email address and more and more people use online platforms like Facebook, MySpace, Flickr, Twitter, Wikipedia, Delicious, LinkedIn etc. Some of these platforms are for social networking, others for publishing contents

(photos, videos, text etc.), for information etc. but all of them keep traces of human activity. The development of Internet, of so-called Web2.0, of communications in general but also of powerful computers being able to register, store and process large amounts of data gives thus unprecedented opportunities for human behavior analysis. Traditionally this was a field of study for sociologists, but it becomes of interest for more and more scientists, from many domains. Such databases containing traces of uses are interesting for instance for mathematicians and computer scientists, who search for relevant and tractable measures to characterize people uses, develop algorithms and software to store and efficiently process such large data etc. They are also interesting for physicists who try to discover the processes behind different activities or dynamics of people and for economists who try for instance to unfold people motivation in making choices.

Traces of uses can be analyzed from different points of view. One approach is the computation of different statistics on frequency or duration of calls in the case of mobile phone communications, or comments and published content in the case of online platforms. This gave interesting insights on the uses of different services on news groups [FSW06], wikis [HBB07], online dating communities [HEL04], question/answer forums [ZAA07, AZBA08], Youtube [CKR⁺07, MAA08] and many other platforms. Another approach, the one we adopt in this thesis, is that of analysis of the **social network** in which people are embedded. When using different services, online or offline, people connect to each other. These connections can be modeled as social networks, merely graphs where the vertices (or nodes) are the persons and the edges (or links) correspond to observed connections between them. It is important to take into consideration these connections because people aren't isolated entities, they live together, interact and influence each other. A often-confirmed phenomenon is that of "word-of-mouth" [EBK69, FS65, AD07]: when making a choice, people often talk to other people, ask for advice and are more likely to choose something if someone they trust has already chosen it. Moreover, people connecting in the same way to the others might have similar behaviors, like the same things etc. It is thus important to see, analyze and characterize people and their uses by taking into consideration the context in which they evolve, the people to which they connect, so the social networks in which they are embedded.

In sociology, the analysis of social networks hasn't appeared with the databases of traces of uses, but a lot of time before, when Internet and mobile communications didn't exist yet. Already present in the work of G. Simmel [Sim55a] (English translation) in the very beginning of the 20th century, it had a real development in the 1950s, when scholars like John A. Barnes, Elisabeth Bott, Sigfried F. Nadel studied patterns of ties between individuals [Bar54], kinship relations [Bot57] and social structure [Nad57]. Then, in the 1970s Harrison White and his students at Harvard University, among which Mark Granovetter and Barry Wellman, elaborated and popularized social network analysis. Since then, questions like strength of personal ties [Gra78], social capital [Col88, Bur92], social roles in a network [LW71, BE89] and many others keep cropping up. Traditionally, when studying social networks, sociologists used to gather data by interviews with the analyzed people. Such data is very rich, very detailed, but it takes time to obtain as one has to interview all the persons in the study. Recordings of traces of uses available nowadays offer new possibilities for social network analysis. However, one has a much less detailed

image of human activities and relations between individuals. A lot of information is not visible in the traces of uses and one cannot ask the studied people about this missing data, as in interviews. Thus, one has no idea about the type of relation between two persons: are they family, friends, colleagues, do they know each other at all? Also, one does not see all the connections between the two persons. Maybe they do not call each other by mobile phone, but have other types of contact, by line phone or e-mail etc. However, even if one does not have the same quality as in data gathered from interviews, obtaining the data is much easier, the amounts are much more important and they are about many people. The difficulty thus changes from obtaining the data to analyzing it.

As a social network is, after all, a graph, one generally uses graph theory when studying social networks. Moreover, large social networks (with, let's say, some thousands of nodes) are also **complex networks**. This is a common name for large graphs modeling relations between entities (persons, institutions, places etc.) found in real-life. A lot of excitement has surrounded the field of the analysis of complex networks since the first studies in the domain, at the end of the 1990s. What created all the excitement was the constant discovery that real-world large graphs are very different from the so-called random networks, so are not random. "Random networks" here means networks where there is no constraint for linking two nodes by an edge: any two nodes of the network can be connected by an edge with a same probability. This defines a model of random generation of networks which was introduced by Erdos and Renyi in the 1960s [ER60], thus being the first and the simplest network generation model. Probably the first paper describing differences between real-world graphs and random ones was [WS98] by Watts and Strogatz. As the graphs analyzed in this paper were different from those generated by the Erdos-Renyi model, the authors concluded that this model wasn't adapted for generation of realistic graphs. As opposed to the Erdos-Renyi model where any two nodes can be connected by a link with the same probability, in real life there is probably a reason for which two nodes become connected, there must be some factors that make a real-world graph come to life and evolve in a certain way. The authors proposed another network generation model and thus began a long series of models. Probably the most famous in this series are the ones proposed by Kleinberg [Kle00] and Barabasi and Albert [BA99], but many others exist [LKF05, KKR⁺99, KRRT99, BJN⁺02] etc.

Since these first studies, researchers have constantly noted differences between real-world graphs and random ones. Basically, no matter from which context the graph comes (sociology, biology, economy, linguistics, computer science etc.), in almost (if not) all the cases, this graph has the same properties as all the other real-world graphs, thus belonging to the group of "complex networks". We present briefly some of these properties. Complex networks have a heterogeneous distribution of the degree: most of the nodes are connected to very few others, while a small fraction of nodes are connected to a very large number of nodes. Also, most of the vertices of the graph belong to a same giant component: for most pairs of nodes, one can go from one node of the pair to the other one by following the edges of the graph. Even more, when going from the first node to the second one in the most direct way one crosses only a small number of edges, usually at most 20. And this even if the graph has several millions of nodes. Another property shared by complex networks is that of the high local density: if two nodes are connected to a common node,

there is a high probability that they are connected to each other, too. Here "high" means a lot higher than in random networks. These properties have been observed for instance in citation graphs [Red98], protein-protein interaction networks [GR03, WF01], biological neural networks [MiOO⁺01, SGS⁺02], food webs [DWM02], social networks modeling online relations [MKG⁺08, ABA03] and many others. As said before, when creating a random generation model, researchers try to identify the factors leading to the creation of links and thus to explain the formation of real-world networks. The quality of the proposed model of network generation is measured by the capacity of the model to produce networks that have (some of) the properties of real graphs.

There are several approaches for analyzing complex networks in general and social networks in particular. Generally one can place the analysis at one of the following three levels: global, intermediate or local. At the global level one takes into consideration the network as a whole and computes different properties for this set. From the previously listed properties, the computation of the giant component, of the distance between the nodes and of the distribution of the number of contacts are included in the global approach. In the intermediate approach one analyzes each node by taking into consideration the whole network. At this level one can compute for instance groups of nodes that are densely connected inside the group and sparsely connected to the other groups; this is called community detection and has been the object of many studies like [Eve80, GN02, Vir03, CMN04, BGLL08] and many others. Also at the intermediate level one can compute the "importance" of each node, usually expressed in terms of centrality (e.g. betweenness [Fre77], closeness, eigen vector [Bon87], page rank [BP98] etc.). Finally, at the local level, a widely used measure is the clustering coefficient [WS98, HK79] measuring the local density of the network. Briefly one computes how connected are to each other the nodes to which a given node is connected (as compared to the case where all these nodes are connected to each other). In this local approach the idea is to analyze each node by taking into consideration only the nodes surrounding it and not the whole network. This is the approach that we consider in this thesis.

We want to answer the following question: given a possibly large social network, describe its local structure, so the way each one of the nodes is connected to the surrounding network. This description should thus offer a characterization of the individuals belonging to a social network by taking into consideration only the structure of the social network (and not other information on the individuals). The computation of this description should take little time and memory so it can be applied to large social networks. To our knowledge, existing methods either place the analysis at the intermediate level (so they characterize the node by taking into consideration the whole network), either offer too little information (like the clustering coefficient that only counts the connections between the contacts of one node).

We propose a method to answer this question, so a method that analyzes the local structure of a given graph and describes the way each node is connected to the network. This method takes into consideration the links each node has with other nodes and the links between these nodes. We apply this method to two social networks: one modeling mobile phone communications and the other one modeling activity of MySpace users. In these networks each node corresponds to a person; when analyzing each node we call

the corresponding person *ego*. As we analyze the way *ego* is connected to the network, this analysis can be called egocentred. Our approach here is related to the analysis of egocentred networks in sociology. In this approach, one studies the personal relations a given individual (*ego*) has with other individuals. The data for such studies is obtained by interviews with *ego* who describes his relations with the other persons and, sometimes, the relations between these persons [Wel79, Wel85, Gri98, Gro05]. Here we try to adapt this approach to large social networks, where the egocentred networks are obtained by focusing on each individual and his links in the network. The egocentred networks thus obtained contain less information, are less detailed than those obtained by interviews with *ego*. The advantage however is that the networks obtained from large graphs are all built in the same way, from observed interactions, and thus are not subjective to *ego*'s opinion on his relations and especially on the relations between his contacts.

The proposed method computes a description of the way each node is connected to the surrounding network and also of how the different persons *ego* is connected to are placed in relation with each other. As it is local, this method does not need the whole social network in order to characterize one node (as opposed to intermediate methods), but merely the nodes to which *ego* is connected and the links between them. Thus, the method can be applied even if one has only fractions of a certain social network. It can be applied as well to small networks built from interviews as to large social networks. Once again, because it is local, its complexity when analyzing one *ego* is also "local" i.e. it depends only on how many contacts *ego* has in the network. This is important because it can be easily applied to large networks; to give an idea, our implementation of the method runs in 30 minutes for all the nodes in a social network with 3 million nodes and 6 million edges on a computer with standard configuration.

After having obtained a characterization of the different persons by taking into consideration the social network in which they are embedded, one can search for correlations between this description and other measures characterizing the individuals. These measures can be socio-demographic data (age, gender, job etc.) or indicators of people activity. For instance for the mobile phone network we use the intensity of communication of each person (number of calls, duration, number of SMS etc.), while for the MySpace network we use measures of online popularity. If the different parameters and the local structure of the network (obtained by applying the proposed method) are found to be correlated, then one can use the parameters in order to infer the local structure and vice-versa. This can be useful when some of the data is missing, for instance if one has the social network in which the individual is embedded but does not have the other information characterizing him. Also, one can divide the persons in the given social network into groups depending on the local structure of the network surrounding them: people connected in identical or similar ways to the network are put in the same group; people with different local structures are put into different groups. This approach is related to that of computing "roles" of nodes in a social network, where nodes occupying the same position, having the same function in the network are grouped together. Note that when searching for social roles (and so in our approach here), nodes put together in the same group are not necessarily connected to each other nor have common contacts, they are just connected in the same way to the network. The problems of dividing individuals into groups based on a prior characterization,

of research of correlations between indicators and of prediction of different parameters are frequently found in data mining. We use some well-known techniques from this domain in order to solve the different problems.

In the following section we present the structure of this thesis and its contributions.

1.2 Thesis overview and contributions

The rest of this thesis is divided into three parts.

Part I presents an overview of existing studies in the different fields of this thesis. We begin by presenting some basic notions and algorithms of graph theory and of data mining in Chapter 2. Next, we present the field of complex networks, several properties, how to compute them and the differences with random networks. Some models and algorithms for random generation of graphs are also discussed. At the end of Chapter 3 a special place is given to the problems of identifying frequent patterns and network motifs, two problems related to the approach adopted in this thesis. Chapter 4 then presents social networks and several important topics in the domain, both in small detailed social networks obtained from interviews and in large social networks modeling phone communications and online activities. We also discuss some differences between offline and online social networks by comparing a mobile phone graph to a graph obtained from activity on Flickr. This is an original work, from which a part has been published in [PSS09, SP09a]. We finish this chapter by presenting some marketing studies using social networks.

Part II is the main part of this thesis. Chapter 5 first introduces the method for characterizing the local structure of large social networks. We present the method, some algorithmic aspects and a comparison with other existing measures and methods. Part of this chapter has been published in [SP09b]. We continue in Chapter 6 with an analysis of the online popularity of artists on MySpace in relation with the social structures in which the artists are embedded. This study on MySpace popularity has been published in [SCB10]. In Chapter 7 we then begin the analysis of a social network modeling mobile phone communications. After some first statistics, we study the contacts of each person (ego) and their relative positions in the social network in relation with each other and with ego. We finish this part by Chapter 8 on clustering of individuals in the mobile phone network depending on the network structures in which they are embedded. We compare the group associated to each person with other information we have on the individuals i.e. age, gender and intensity of communication. Parts of the work presented in these last two chapters have been published in [SP09b, SSPG10].

The last part concludes this thesis and presents some possible directions for future work.

The appendix contains the French translation of the introduction and of Chapter 5, the central chapter of this thesis.

1.2.1 Publications

The research carried out during this PhD thesis led to the following publications:

International conferences with reviewing process and proceedings:

- [SCB10] Alina Stoica, Thomas Couronné, Jean-Samuel Beuscart. To be a star is not only metaphoric: from popularity to social linkage. *The 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, Washington, United States, 2010.
- [CSB10] Thomas Couronné, Alina Stoica, Jean-Samuel Beuscart. Online social network popularity evolution: an additive mixture model. *The 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Odense, Denmark, 2010.
- [SP09b] Alina Stoica, Christophe Prieur. Structure of neighborhoods in a large social network. *The 2009 IEEE International Conference on Social Computing (Social-Com)*, Vancouver, Canada, 2009.

Journals:

- [PSS09] Christophe Prieur, Alina Stoica, Zbigniew Smoreda. Extraction de réseaux égocentrés dans un (très grand) réseau social. *Bulletin de méthodologie sociologique*, number 101, 2009.

Workshop and conferences with abstract-based submission:

- [SSPG10] Alina Stoica, Zbigniew Smoreda, Christophe Prieur, Jean-Loup Guillaume. Age, Gender and Communication Networks. *NetMob, Workshop on the Analysis of Mobile Phone Networks*, Boston, United States, 2010.
- [SP09a] Alina Stoica, Christophe Prieur. Structure of ego-centered networks in very large social networks. *The XXIX International Social Network Conference (Sunbelt)*, San Diego, United States, 2009.

Part I

Overview and survey

In this part we present the different fields to which this thesis is related. We begin by reviewing several basic concepts of graph theory and data mining. Next we make a survey of existing studies on complex networks, by presenting their main properties, how to compute them and also some existing network models and random graphs generators. We continue with a survey of questioning and advances on social networks, from different points of view, going from detailed sociological approaches to analysis of large databases on phone communications and online activities.

Section 4.5, discussing several differences between an online and an offline network, is an original work.

We finish this part with a presentation of marketing studies that use social networks.

Chapter 2

Basic notions

We present here some basic graph-theory concepts and an overview of data mining algorithms.

2.1 Graph theory concepts

A *graph* $G = (V, E)$ is a set V of elements called *vertices* along with a set of so-called *edges* $E \subseteq V \times V$ connecting pairs of vertices in V . *Network* is a synonym for graph used especially in sciences like sociology or biology. We interchangeably use the terms vertex and node to refer to the elements of the set V , and similarly edge and link to refer to the elements of the set E , although vertex and edge are usually associated to the notion of graph, while node and link are associated to that of network. The graph G is *undirected* if for all $(u, v) \in E$ also $(v, u) \in E$ i.e. edges are unordered pairs of nodes. If pairs of nodes are ordered, so edges have direction, the graph is *directed*; in this case edges are usually called arcs. The graph G is *simple* if it has no multiple edges (i.e. for all $u, v \in V$ there is at most one edge connecting u to v) and no self-loops ($(v, v) \notin E$, for all $v \in V$). Throughout this document, unless specified otherwise, the considered graphs are simple and undirected. The *complement graph* of a graph $G = (V, E)$ is a graph $G' = (V', E')$ where the vertices are the same as in G (i.e. $V' = V$) and the edges are all the possible edges between vertices in V that are not present in E (i.e. $E' = \{(u, v), u, v \in V \text{ and } (u, v) \notin E\}$).

Neighborhood: A vertex $u \in V$ is a *neighbor* of the vertex $v \in V$ if and only if $(u, v) \in E$; in this case the two vertices are said to be *adjacent*. The set $N(v) = \{u \in V, (u, v) \in E\}$ represents the *neighborhood* of v , $N[v] = N(v) \cup \{v\}$ represents its closed neighborhood and $d(v) = |N(v)|$ represents its *degree*.

Paths and Connectedness: A *path* in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. A path where the first vertex in the sequence is the same as the last vertex in the sequence is called a *cycle*. The *length* of the path is the number of edges the path uses. The *distance* between two vertices u and v is the length of a shortest path from u to v . If there is no such path, the distance is infinite and the two vertices are not connected. A *connected component* is a maximal set of vertices where for every pair of vertices there is a finite path connecting

them. A graph is connected if it has exactly one connected component containing all of its vertices. The *diameter* of a graph is the largest distance found in the graph (when taking any two of its vertices). Of course this definition makes sense only for connected graphs, so one usually restricts the computation of the diameter to the largest connected component of the graph.

Graph isomorphism: Two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are *isomorphic* if and only if there exists a bijective function $\varphi : V_G \rightarrow V_H$ (called isomorphism of G and H) such that any two vertices u and v are adjacent in G if and only if $\varphi(u)$ and $\varphi(v)$ are adjacent in H . When G and H are one and the same graph, the function φ is called *automorphism* of G . The graph isomorphism is an equivalence relation on graphs so it partitions the class of graphs into equivalence classes, called isomorphism classes.

Density: The density ρ of a graph $G = (V, E)$ with at least 2 vertices is the ratio between the number of edges of the graph and the total number of possible edges: $\rho = \frac{|E|}{\binom{|V|}{2}}$.

Subgraphs: Given a graph $G = (V_G, E_G)$, a graph $H = (V_H, E_H)$ is a *subgraph* of G if $V_H \subseteq V_G$ and for all $u, v \in V_H$, if $(u, v) \in E_H$ then $(u, v) \in E_G$. H is an *induced subgraph* of G if $V_H \subseteq V_G$ and for all $u, v \in V_H$, $(u, v) \in E_H$ if and only if $(u, v) \in E_G$. As a special case, a *triangle* is a connected triplet of vertices (u, v, w) with $(u, v), (u, w), (v, w) \in E$.

Graph traversal: A graph traversal is a way of visiting all the vertices of a graph by following its edges. The most used graph traversals are the *depth-first search (DFS)* and the *breadth-first search (BFS)*. In both, one starts with a node, called the root, and explores its neighbors, their neighbors etc. until all the vertices are explored. For each node, its unexplored neighbors are called its children. In the DFS one starts with the root, then explores one child, its children, their children etc. before passing to the next child. In the BFS one starts with the root, then explores all its children, then their children etc.

Representation: Let n be the number of vertices of a graph G (i.e. $n = |V|$) and m be the number of its edges (i.e. $m = |E|$). The *adjacency matrix* of the graph G is a $n \times n$ matrix A such that $A_{i,j} = 1$ if $(i, j) \in E$ and 0 otherwise. With this encoding, testing the presence of an edge takes $\Theta(1)$ time, which is time efficient. However, running through the neighborhood of a vertex v takes $\Theta(n)$ time; moreover this representation takes $\Theta(n^2)$ space which is inefficient if the graph is sparse (i.e. $m \in o(n^2)$).

Another graph encoding, more useful in the case of large graphs, is the *adjacency list representation* where, for each vertex, one stores the (sorted) list of its neighbors. This representation needs $\Theta(m)$ space, which is efficient, and running through $N(v)$ takes $\Theta(d(v))$ time. However testing the presence of an edge (u, v) takes $\Theta(d(v))$ time ($O(\log(d(v)))$ if $N(v)$ is sorted). This encoding is nevertheless much more efficient than the previous one for large sparse graphs.

Time and space complexity: Even if this is not necessarily connected to the graph theory, we explain the three Landau notations: O , Θ and o . Given two functions f and g , one writes $f(x) \in O(g(x))$ if and only if there exists a positive real number k and a real number x_0 such that $|f(x)| \leq k|g(x)|$ for all $x > x_0$; in this case f is bounded above by g asymptotically. One writes $f(x) \in \Theta(g(x))$ if and only if there exist two positive real numbers k_1 and k_2 and a real number x_0 such that $k_1|g(x)| \leq |f(x)| \leq k_2|g(x)|$ for all $x >$

x_0 ; in this case f is bounded both above and below by g asymptotically. Finally, one writes $f(x) \in o(g(x))$ if $\forall \varepsilon > 0$ there exists a real positive number x_0 such that $|f(x)| \leq \varepsilon |g(x)|$ for all $x > x_0$; in this case f is dominated by g asymptotically.

For a useful introduction to graph theory and algorithms, see for instance [CLR01].

2.2 Data mining

Data mining is the process of extracting patterns from data. It is the application of statistical methods, data analysis and artificial intelligence to (often large) databases in order to extract meaningful information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery. We present here some useful data mining methods and several classical statistical measures. We focus our presentation on the goals of the different methods and on how they can be used, rather than the mathematical considerations (which explain how the method works and why it gives good results). For useful books on the subject, see for instance [FPSSU96, HTF01].

Data mining methods can be categorized into two sets: descriptive methods and predictive methods. In both methods, one has a database of individuals (or objects, elements etc.) which are characterized by a set of variables: for each individual, there is a value for each variable¹. In the first category of methods (descriptive) there is no favored variable; in the second case, there is one, also called the target variable (or dependent or variable to explain). Variables that can take only a few values can be seen as categories or classes; they are called categorical variables. Variables that can take any real value (maybe restricted to some interval) are called continuous variables.

Descriptive methods

Given a set of p individuals and a set of n variables characterizing them, one needs to group them in a limited number k of classes (or clusters) such that individuals with similar characteristics are grouped together. The vector of values of the n variables characterizing each individual is called *feature vector*. One has no a priori idea of the possible classes nor, sometimes, of their number. This type of problem (called **clustering**) occurs often in marketing, where companies need to divide the set of their customers in classes in order to make offers adapted to the customers' expectations and characteristics, in medicine, where patients reacting similarly to medication need to be treated in a certain way, in sociology, trade etc. There are several methods for answering this question:

- partition algorithms (k-means, density methods, Kohonen self organizing maps, relational clustering etc.),
- hierarchical methods (either agglomerative ("bottom-up") or divisive ("top-down")),
- fuzzy methods.

¹Some values might be missing; this is a special case that we do not discuss here.

There are several aspects that need to be taken into consideration when doing a clustering; often the results depend on them. First, one often needs a notion of distance between individuals: the individuals who are similar must be close to each other according to this distance. In most cases the chosen definition of distance is the Euclidean one:

$$d(u, v) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_n - v_n)^2} = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$$

where u and v are the two individuals characterized by n variables with values u_1, \dots, u_n and v_1, \dots, v_n respectively. Other possible distance are the Manhattan distance ($d(u, v) = \|u - v\|_1 = \sum_{i=1}^n |u_i - v_i|$), the angle between the corresponding vectors, the Hamming distance (which measures the minimum number of substitutions required to change one member into another) etc. Second, the number of clusters in which the population is divided must be decided. There are some methods that compute this number by themselves (e.g. the relational clustering), others where it is easy to compute it (e.g. hierarchical clustering), but also methods where this number must be given as input (e.g. k-means). This can be a problem if the given number does not correspond to the real distribution of the population. Third, the validation of the results might be difficult if one has no ideas of how the individuals should be grouped (especially if the dataset is very large). There are different methods of validation depending on the clustering algorithm. Usually, the algorithm tries to minimize the intra-cluster variance (the mean of the square distance from each individual to the center of the cluster) and to maximize the inter-cluster variance (the mean of the square distance from each cluster center to the global center). The center (or centroid) C of a cluster K is a vector representing the average of all the points in the cluster i.e. for each variable i , its value is the arithmetic mean of the values for that variable of all the points in the cluster: $C_K(i) = \frac{1}{n_K} \sum_{v \in K} v_i$ where n_K denotes the number of individuals in the cluster K , v is a point in the cluster and v_i is its value for the i -th variable.

The **k-means algorithm** assigns each point to the cluster whose center is nearest (according to the chosen distance). For creating k clusters, the algorithm works as it follows: first, it generates k random points as clusters centers (if these centers are not given as input); then, it assigns each point to the nearest cluster center and it compute the new cluster centers; it repeats the two previous steps until some convergence criterion is met. The main advantages of this algorithm are its simplicity and speed which allows it to run on large datasets. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. Also, to compute the clusters, it minimizes intra-cluster variance, but does not ensure that the result has a global minimum of variance. Therefore, when clustering a set of points, one should also perform several k-means clusterings and choose the one with the minimal variance. As the number of clusters must given as input, one should perform several clustering with different numbers k of clusters. To choose the best number of clusters, one can compute the average silhouette [KR90] of each clustering and take the one with the

highest average ². For each point and its attributed cluster, the silhouette measures how similar that point is to points in its own cluster compared to points in other clusters. This value ranges from -1 (indicating that the point has been put in the wrong cluster) to 1 (indicating that the point is very similar to the other points in its cluster). A clustering with a higher average silhouette is therefore a better clustering.

In the **Kohonen self-organizing map** [Koh90], the aim is to cluster the individuals and also to build a bi-dimensional map with n layers (a layer for each variable describing the individuals) where the individuals are placed depending on their topological proximity. The map's smallest entity is a cell, and each individual is placed in only one cell (the individual has the same position and therefore cell on all the layers); there are \sqrt{p} cells where p is the size of the population to cluster. The method has three steps. The first one is the learning. The feature vectors of the cells are randomly initialized. Then a subset of the population to model is randomly selected; for each individual in this selection the SOM finds the ("winner") cell whose feature vector is the most similar (i.e. is the closest by a given distance). The feature vector of the winner cell is updated to take into account the feature values of the individual. The feature vector of the neighbor cells are then modified to reduce the vectors gradient with the new values of the cells' feature vector. The second step of the algorithm is the processing of the global population to model: each individual is placed in the cell with the closest feature vector. Finally the last step is the clustering of the cells with, for instance, a k-means algorithm, based on the similarity of their feature vectors.

In the **hierarchical agglomerative clustering** clusters are built by progressively merging existing clusters, thus creating a hierarchy of clusters. The initial clusters are the individuals themselves. At each step of the algorithm, the two closest clusters are merged. Different definitions of distance between clusters can be used: the Euclidian distance between their centers, between all their individuals, between the two far-most individuals or, on the contrary, between the closest two, the increase in variance for the cluster being merged (Ward's criterion) etc. Each agglomeration occurs at a greater distance between clusters than the previous agglomeration, and one can decide to stop clustering either when the clusters are too far apart to be merged (distance criterion) or when there is a sufficiently small number of clusters (number criterion). As it needs to compute, several times, the distances between all the clusters, this method can be hardly applied on large data.

As opposed to the first two types of methods (partition algorithms and hierarchical methods), the fuzzy algorithms do not place each individual in only one cluster, but rather compute a probability of belonging to each one of the clusters.

Another set of descriptive methods, whose goals are quite different from those of the clustering algorithms, are the **factorial methods**. Here the idea is to project the data in a smaller number of dimensions (smaller than the n characterizing the individuals), usually 2 or 3, and thus be able to visualize it. One very popular method in this category is the **principal component analysis** [Pea01] which transforms the n possibly correlated

²This is the method proposed and implemented by the statistical tool of Matlab:
http://www.mathworks.com/access/helpdesk/help/toolbox/stats/bq_679x-18.html

variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. The new variables are linear combinations of the initial variables. By computing the values of these new variables for each individual, one has a representation of the individuals in a smaller number of variables. One can also plot them in a 2D-space represented by the first two principal components and thus have an image of the similarity between individuals.

Predictive methods

In this case there is a special variable among the n characterizing the individuals. The different methods try to estimate the value of this variable (called variable to explain or dependent or target variable) depending on the values of the other variables characterizing the individuals (called explaining or independent variables). If the target variable can have only a few values, these values are considered as classes or categories of individuals. In this case, using the explaining variables, one tries to discover the set of rules that make that each individual is given a certain class. This way, if a new individual enters the population, one can attribute him a class depending on his values for the explaining variables. This problem is called ***classification***. Another problem is the ***prediction***, where the target variable is continuous. In this case one needs to find the relation between the value of the target variable and those of the explaining variables, relation usually given by a formula. The two types of problems occur often in medicine (where one needs to predict the efficiency of medication, the probability that a patient recover), in industry (where one needs to compute the probability of occurrence of a certain phenomenon), in sociology (in order to predict the behavior of a person), in meteorology, agriculture, banking etc.

The main classification methods are:

- the decision trees,
- the linear discriminant analysis,
- the logistic regression,
- the k-nearest neighbors method,
- the methods based on neural networks: the support vector machines, the genetic algorithms, the expert systems.

The main prediction method is the linear regression.

In the classification methods, one usually uses a set of randomly chosen individuals (among the existing population) in order to learn the rules (so build a model) by which the different individuals are divided in the different classes. This is the learning set. Then one takes a set of individuals from the remaining population and test the precision of the model on them. The precision can be measured by the fraction of individuals whose real class is the same as the one predicted by the model. Nevertheless, not all methods build a model from a learning set; some methods simply attribute a class to each individual based

on some measures and not on a set of rules. For instance, the method of the **k-nearest neighbors** attributes to each individual the class of the k nearest individuals from him (according to a distance e.g. the Euclidian one). However, the choice of k , of the distance to use and the fact that the classification of each new individual requires the manipulation of a whole set of already classified individuals make this method difficult to use. One usually prefers the methods where a model is built, especially when classifying large data.

A **decision tree** is used in order to find a set of rules that associate each individual to a class. It begins by identifying the variable that divides best the individuals in the different classes such that one obtains some sub-populations, called nodes. The population of each node is then divided in other nodes based on the variable that splits best the individuals in classes. This is repeated until no division is possible or wanted. By construction, the final nodes (the leaves) contain mainly individuals of a single class. Each individual is associated to a leaf, so to a certain class, with a rather high probability when he fulfills the set of rules allowing to get from the root to that leaf. The set of rules of all the leaves represents the classification model, used to attribute classes to new individuals. This method is fast and the classification rules are easy to understand. Moreover it does not require any special conditions for the explaining variables (as for instance some probability laws or absence of collinearity). However, each level of the tree depends on the previous one, which makes that the tree might find local optimums instead of global ones.

As a prediction method, the **linear regression** estimates the value of the target variable depending on the explaining variables. More precisely it estimates the conditional expectation of the dependent variable - that is, the average value of the dependent variable when the independent variables are held fixed. Regression analysis is widely used for prediction but also to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. This method works only under several conditions: the explaining variables are continuous and linearly independent; other assumptions are also made on the sample data and on the errors of the modeling function.

We also present some useful statistical measures. The *standard deviation* σ measures the dispersion of a variable X : $\sigma_x = \sqrt{E[(X - \mu_x)^2]}$ where the operator E denotes the average or expected value and $\mu_x = E[X]$. When the variable X has N values x_1, \dots, x_N the standard deviation is $\sigma_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)^2}$. If one cannot obtain all the values taken by X for the given population, one can use a sample of the population. In this case the standard deviation is only estimated; the denominator is replaced by $N - 1$ instead of N , where N is the size of the sample. Sometimes it may be useful to *center and scale* a variable X i.e. to transform X into a new variable Z with mean zero and standard deviation one: $z_i = \frac{(x_i - \mu_x)}{\sigma_x}$ for all i from 1 to N .

The *covariance* of two variables X and Y is a measure of how much the two variables change together and is defined as $\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$. If the two variables have N values respectively x_1, \dots, x_N and y_1, \dots, y_N , the covariance is $\text{Cov}(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$.

Often one needs to measure the intensity of the relationship (or the *correlation*) be-

tween two variables X and Y . If the two *variables are continuous*, this can be done by computing the *linear correlation coefficient* (also called Pearson correlation) r_{xy} between the two variables: $r_{X,Y} = \text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_x \sigma_y} = \frac{E[(X-\mu_x)(Y-\mu_y)]}{\sigma_x \sigma_y}$. The Pearson correlation is $+1$ in the case of a perfect increasing (positive) linear relationship, -1 in the case of a perfect decreasing (negative) linear relationship, and some value between -1 and 1 in all the other cases, indicating the degree of linear dependence between the variables. As it approaches zero the correlation is weaker. The closer the coefficient is to either -1 or 1 , the stronger the correlation between the variables. If the two *variables take only a few values* (i.e. they represent classes or categories), one can verify if the two variables are independent by performing a χ^2 test (read chi-square). One can use this test to decide if the category X depends on the class Y to which the individual belongs. If one needs to measure the correlation between a *continuous variable and a categorization*, one can perform a ANOVA test. This test tells if the mean of the continuous variable is the same for the different categories. If this is true then the two variables are independent. For instance, one can use the ANOVA test in order to see if the salary (the continuous variable) is independent from the gender (the categories, male and female). However, this test says only if the means are different or not, but it does not say for which categories the means are significantly different and for which they are not. A test that can provide such information is called a *multiple comparison test*. Such tests are the Bonferroni and the Scheffé tests.

The χ^2 and the ANOVA are examples of **hypothesis tests**. Such tests are used to prove a given hypothesis H_1 . For that, one submits the opposite hypothesis H_0 to a test T that must be satisfied if H_0 is true. The idea is to show that T is not satisfied which means that H_0 is false, so H_1 is true. H_0 is called the null hypothesis while H_1 is called the alternative hypothesis. To build the test T , one associates a statistic to H_0 using the observations; this statistic must follow a theoretical law if H_0 is true. Next one measures the value v of the statistic on the given data and compares this value to the theoretical values of the law. Also one chooses a *significance level* as a threshold from which the hypothesis is rejected; usually this value is at most 0.05 . Now one computes the *p-value* which is the probability to observe such a value as v if H_0 is true. If this probability is lower than the significance level, the null hypothesis H_0 is rejected, so H_1 is accepted. On the contrary, if the *p-value* is higher than the significance level, the null hypothesis can't be rejected, so one does not know if H_1 is true.

Chapter 3

Complex networks

Informally, complex networks are modeling of large data. In many domains, sets of objects and relations between them can be modeled as graphs where the vertices are the objects and the edges correspond to relations. At the end of the 1990's, due to the exponential growth of the size of relational databases, along with the development of communication tools, researchers began to analyze graphs modeling large datasets (with at least several thousands of recordings). Although graph theory has a long tradition, the analysis of graphs modeling large datasets became a new field of study which began to develop very fast, being surrounded by a lot of excitement. This is due not just to the development of powerful computers able to store and handle such large datasets but also (and especially) to the discovery of a set of properties shared by these graphs. Large graphs (and by large we mean at least 10^5 vertices and edges) modeling datasets from numerous domains such as biology, linguistics, inter-personal communication, WWW etc. are constantly found to share several characteristics [BA99, WS98, New03]. They are therefore grouped under a common name, that of *complex networks*.

There are numerous examples of complex networks extracted from real-life phenomena. They can model for instance the presence of words in sentences, interactions between proteins, collaborations between boards of directors, traces of phone calls or online activity, mobility dynamics of people, connections by plane between airports etc. They are the object of study of many researchers, from several domains, going from computer scientists, mathematicians, physicists, to biologists, sociologists, economists etc. The interest comes from the importance of the study of such networks in understanding how nature works, how people interact, how different relations appear and evolve etc. Moreover these interactions or relations are not random, they do not appear with an equal probability between two objects or two persons, but they are triggered by different factors. This was a major discovery in the analysis of complex networks: they are not random networks. Even more, as said before, they share several non-trivial properties. Almost every large network found in nature, no matter its origin, follows a same set of characteristics. We detail these properties, along with computational issues and examples of complex networks presenting them, in Section 3.1. We then present several models for network generation in Section 3.2. We finish this discussion on complex networks by showing some techniques for frequent

patterns discovery and motifs identification in Section 3.3.

3.1 Complex networks properties

We present several properties shared by most complex networks, their values in randomly generated networks, some computational aspects and real-world examples.

In this section on complex networks properties, by randomly generated graph we mean a graph where no particular constraint is imposed (besides the number of vertices and edges): there can be an edge between each pair of vertices with the same probability. This model of graph generation was introduced by Erdos and Renyi [ER60] and is a pioneer work in the domain. The idea is very simple: we start with n nodes and we add edges such that, for each pair of nodes, an edge is added with equal probability p . This defines a set of graphs $G(n, p)$ where (n, p) are the parameters of the model. Such graphs have some interesting properties that we present in the same time as those of real-world networks.

For a graph $G = (V, E)$, let n denote its number of vertices (i.e. $n = |V|$) and m its number of edges (i.e. $m = |E|$).

Graphs randomly generated by the Erdos-Renyi model are used for comparisons with real networks: for each real graph with n vertices and m edges, one generates random graphs $G(n, p)$ with $p = \frac{2m}{n(n-1)}$, so graphs that have the same number of vertices and edges as the original one. Several characteristics are found to be shared by real-world networks but not by the randomly generated graphs. We present here each one of these characteristics, their values in several examples from real life and in random graphs but also existing methods for their computation in large graphs. Remember that we compute these properties in graphs that have typically at least 10^5 vertices and an even higher number of edges. A computation that takes $O(n^2)$ time (with n the number of vertices) is impractical for such graphs. Therefore one needs to use efficient (preferentially linear) algorithms when analyzing complex networks.

Degree distribution.

Definition. Generally in complex networks most nodes have very low degrees while there is a small fraction of nodes with very high degrees. When plotting the distribution of degrees, one obtains a curve that is very close to the axis (see Figure 3.1(a)). This is very different from the binomial degree distribution of random networks (see Figure 3.1(b)); in these random graphs the probability that a node has degree k is

$$P(\text{degree}(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}.$$

On the contrary, many real-world graphs have degree distributions with probability density functions of the form

$$p(x) = ax^{-\gamma}$$

where $p(x)$ is the probability to encounter the value x , a is a constant and γ is an exponent; distributions with such probability density functions are called *power-laws* and γ is

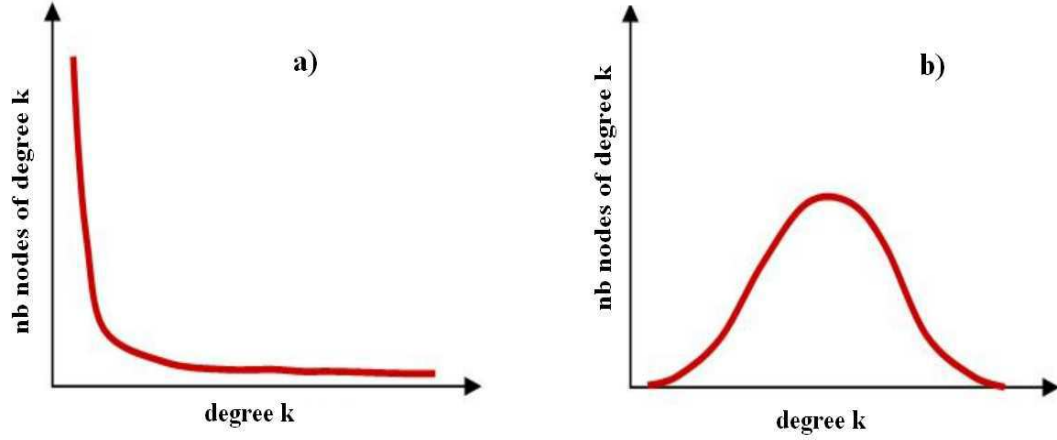


Figure 3.1: In many complex networks the degree distribution plot looks like (a), while in random networks it looks like (b).

called the power-law exponent. A power law distribution is sometimes called a *scale-free* distribution, which intuitively means that it looks the same regardless of on what scale we look at it. More precisely, there exists a function g such that $p(bx) = g(b)p(x)$ for all b (x and $p(x)$ previously defined): $g(b) = b^{-\gamma}$. The scale-free property means that when multiplying x by a scaling factor b the shape of the distribution $p(x)$ remains unchanged except for a multiplicative constant: it does not depend on the scale. When plotted in a log-log scale, a power-law distribution is a straight line (see Figure 3.2).

Computation. Computing the degree distribution of a given graph is quite easy, one needs only to find the degree of each node and then to count the number of occurrences of each degree. On the contrary, trying to match the degree distribution to a power-law is not a simple task: the power law could be only in the tail of the distribution and not over the entire distribution, estimators of the power law exponent could be biased, some required assumptions may not hold etc. There are several methods employed nowadays, like linear regressions using the plot of the data on the log-log scale (after having distributed the data in equal-sized bins or in bins with exponentially increasing size), regression using the cumulative distribution of the degree, maximum-likelihood estimators where the value of the power law exponent γ is estimated such that the likelihood that the data came from the corresponding power-law distribution is maximized, and many others. Further details on the mathematics of power-laws can be found in [Mit04, New05, CSN07].

Deviations from power-laws. There are many studies on complex networks where the degree distribution is computed and found to be skewed, with many nodes having a small degree and a small fraction of nodes having high degrees. However, this does not necessarily mean that the degree distribution is a power-law. There are several examples of real-world complex networks that present deviations from the power-law distributions; often their distributions belong to one of the two following cases: power-laws with exponential cutoffs and lognormals. For *power-laws with exponential cutoffs*, the log-log plot of the distribution looks like a power-law (so a straight line) for the lower range of values

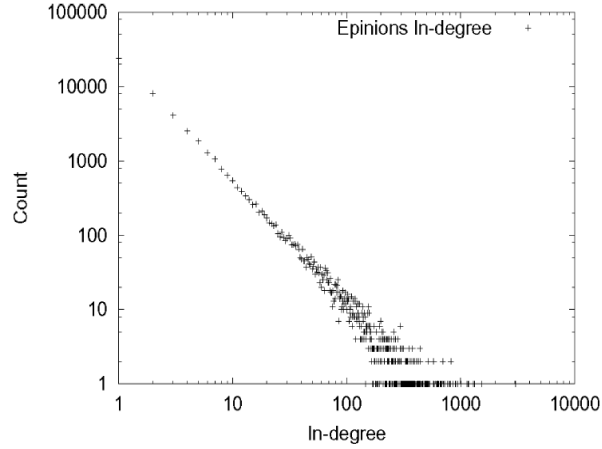


Figure 3.2: The in-degree distribution on a log-log scale for the Epinions graph (an online social network of 75,888 people and 508,960 edges [DR01]). This distribution follows a power law.

of the degree and then decays very fast for large values (see Figure 3.3(left)). Often the decay is exponential and is usually called an exponential cutoff. This distribution does not scale and is thus not asymptotically a power law; however, it does approximately scale over a finite region before the cutoff. This distribution captures limitations of size found in real world, as for example for the network of airports [ASBS00]. There is a cutoff in the possible number of nonstop destinations reachable from an airport: this might be because airports have a limited capacity to handle new edges that they end up reaching. The *lognormal distribution* is a distribution whose logarithm is a normal distribution; its plot in the log-log scale looks like a truncated parabola (see Figure 3.3(right)).

Examples of degree distributions in real-world complex networks. The degree distribution was found to be a power-law or one of the two deviations for the Internet [FFF99], the web [AH01], graphs modeling activity on online platforms [KNT06, MMG⁺07], citation graphs [Red98], protein-protein interaction networks [GR03, WF01], biological neural networks [MiOO⁺01, SGS⁺02], food webs [DWM02] and many others.

Diameter

Definition. As defined in Section 2.1, the diameter of a graph is the largest distance in the graph, where a distance is measured for each pair of nodes as the length of a shortest path between them. In other words, it is the minimum number of hops in which any node of the graph can reach any other node. This definition makes sense only for connected graphs, so one generally restricts the computation to the largest connected component. This is not a problem because in most complex networks there is a giant connected component that contains the vast majority of nodes.

Besides this classical definition of the diameter, several other terms have been used to describe the idea of distance between nodes. For instance the *effective diameter* is a measure less susceptible to outliers; it is the minimum number of hops in which some

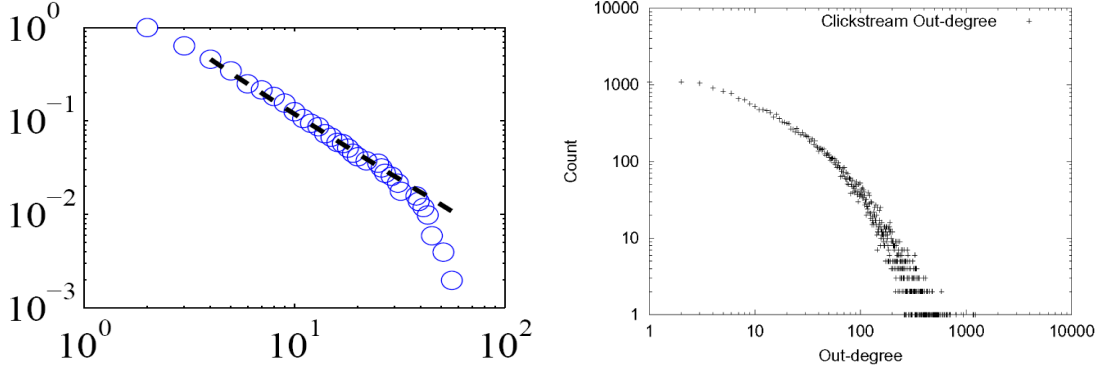


Figure 3.3: (left) The probability of the number of species per genus of mammals during the late Quaternary period [CSN07]. This distribution has an exponential cutoff. (right) The out-degree distribution of a Clickstream graph (a bipartite graph of users and the web sites they surf [MF01]). This distribution is log-normal.

fraction (e.g. 90%) of all connected pairs of nodes can reach each other [TPSF01].

Another term is that of *characteristic path length*. For each node of a graph, one starts by computing the average path length as the average distance from the node to any other node (in the same connected component). The characteristic path length is then the median value of the average path length for all the nodes [BT02]. By taking the mean value of the average path lengths for all the nodes one computes another measure, known as the *average diameter*.

A notion connected to that of diameter is the *hop-plot* [FFF99]. The hop-plot of a network is its set of pairs $(d, g(d))$ where d is a natural number and $g(d)$ is the fraction of connected node pairs whose shortest connecting path has length at most d . See Figure 3.4 for an example of the hop-plot and the effective diameter in a real-world complex network, as presented in [CF06].

Computation. Computing the distance between each pair of nodes can be done by computing first the distance from one node to every other node; this takes $\Theta(m)$ time and $\Theta(n)$ space with a breadth-first search (BFS), where n is the number of nodes of the graph and m is the number of edges. One does this for each node of the graph, so the total computation takes $\Theta(nm)$ time and $\Theta(n)$ space. The time complexity is much too high given that nm is at least 10^{10} for real-world complex networks. Faster algorithms have been proposed [AGMN92, FM91, Sei92] but they have a space complexity of $\Theta(n^2)$ which, once again, is impractical for complex networks. A common solution is to estimate the different measures. For instance for finding the hop-plot a randomized algorithm that takes $O(n+m)d$ time and $O(n)$ space, where d is the diameter of the graph, generally very small, has been proposed in [PGF02]. For the classical definition of the diameter, efficient algorithms for finding lower and upper bounds have been proposed in [MLH08, PCM10]. An estimation of the diameter is obtained in a small number of steps (often 10 steps are sufficient) where a step needs only $\Theta(m)$ time and $\Theta(n)$ space.

Examples from real-world complex networks. The diameter of many complex networks

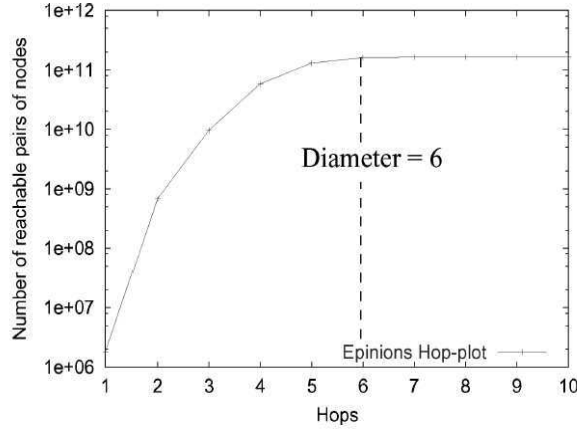


Figure 3.4: Hop-plot and effective diameter. This is the hop-plot of the Epinions graph [DR01]. We see that the number of reachable pairs of nodes flattens out at around 6 hops; thus the effective diameter of this graph is 6.

has been found to be very small compared to the graph size. The effective diameter was computed for the Internet graph in [FFF99] and was found to be around 4 for the Internet AS-level and around 12 for the Router-level. The average diameter was found to be 11.2 for the graph of the Web pages in the nd.edu domain [AJB99], 18.7 for the power grid and 3.65 for the network of actors [WS98]. Many other examples can be found in the literature; see for instance [New03] for a list of examples. This phenomenon of small diameter of complex networks, known as the "small-world" phenomenon, is rather surprising given the large size of the networks. Even more, the diameter is found to be shrinking in time [LKF05]. On the contrary, the Erdos-Renyi random networks have a diameter concentrated about $\log n / \log z$ where n is the number of nodes in the graph and z is the average degree; in this case, the diameter grows slowly as the number of nodes increases.

Clustering coefficient

Definition. The clustering coefficient can be computed for each node of a graph and, in this case, measures how densely the neighbors of the node are connected to each other, or it can be computed for the whole graph and, in this case, measures the transitivity of the graph. For a node, the clustering coefficient represents the number of links between its neighbors compared to the total possible number of links. If the node has degree $d > 1$, then its clustering coefficient is $\frac{nb_t}{\binom{d}{2}}$ where nb_t is the number of links between the neighbors of the node [WS98] (see Figure 3.5 for an exemple). Note that nb_t is the number of triangles to which the node belongs. Now, for the clustering coefficient of the graph, there are two definitions. One possibility is to compute the mean of the clustering coefficients of all the nodes with degree at least 1 of the graph. A second definition (also known as

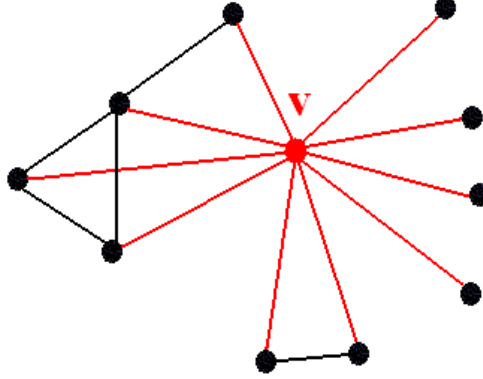


Figure 3.5: Clustering coefficient. The vertex v has 10 neighbors which are connected by 5 edges. Thus the clustering coefficient of v is $\frac{5}{\binom{10}{2}} = \frac{1}{9}$.

the *transitivity ratio* [HP57, HK79]) is

$$\frac{3 \times \text{the total number of triangles of the graph}}{\text{the number of connected triplets of the graph}}$$

where a connected triplet is formed by a central node connected to two others; the factor of 3 comes from the fact that a triangle is counted as three triplets.

Computation. For the computation of the clustering coefficient one needs to count the triangles containing a node (and repeat this for all the nodes of the graph when counting the clustering coefficient of the whole graph). The fastest algorithm for doing this relies on matrix product [IR78, CW87, AYZ97]. This is based on the observation that elements on the diagonal of A^3 (where A is the adjacency matrix of the graph) represent the number of triangles to which the nodes of the graph belong. Thus the counting of triangles can be done in $O(n^\omega)$ time where $\omega < 2.376$ is the fast matrix product exponent [CW87]. The problem of this approach is that the adjacency matrix must be stored; moreover the matrix A^2 must be computed and stored leading to a $\Theta(n^2)$ supplementary space complexity. Other solutions for the problem of counting of triangles have been proposed [Lat08, SW05]; they are slower than the previous one but require less space ($\Theta(m^{\frac{3}{2}})$ time and $\Theta(n)$ space for the first one, $\Theta(n^3)$ or $\Theta(nm)$ time and $\Theta(1)$ space for the second one), and also list the triangles (i.e. they give the 3 vertices belonging to each triangle). In the case of graphs with power-law degree distributions, the listing of triangles is faster, taking $O(mn^{\frac{1}{\alpha}})$ time and $\Theta(n)$ space where α is the exponent of the power-law. See [Lat08] for a detailed survey of algorithms for triangles computation and listing.

Examples from real-world complex networks. The clustering coefficient is found to be significantly higher in real-world complex networks than in random ones. In networks generated by the Erdos-Renyi model the clustering coefficient is equal to $\frac{z}{n}$ where z is the average node degree and n is the number of nodes. When n is large, the clustering

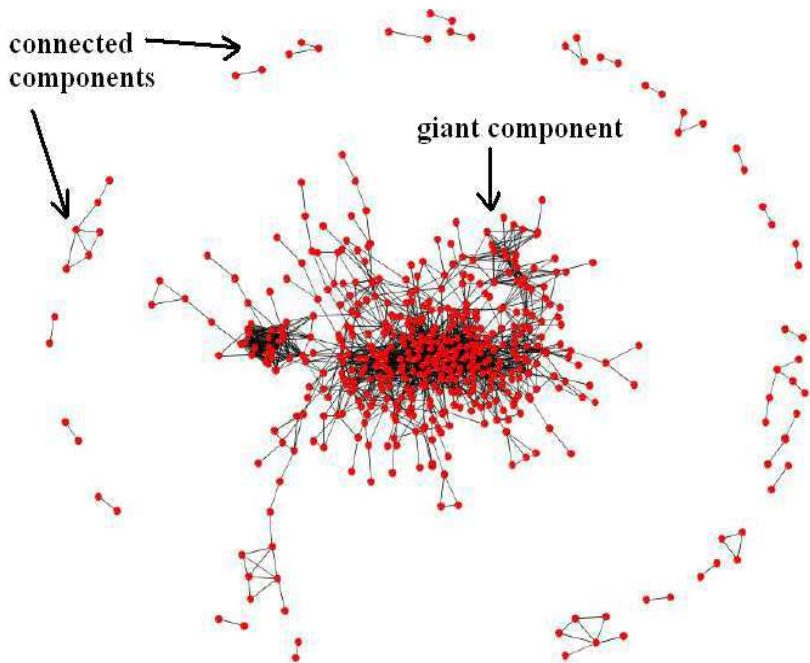


Figure 3.6: Connected components. Most of the nodes belong to a giant component and few nodes belong to small connected components.

coefficient takes very low values. On the contrary, the value of the clustering coefficient is rather high in real-world complex networks (compared to the one in random networks anyway). Thus, in [WS98] the clustering coefficient (computed as the average value of the clustering coefficients of the nodes) is found to be 0.79 in the actor network as opposed to 0.00027 in the corresponding random network and 0.08 for the power grid network as opposed to 0.005 for the random graph. Many other researchers have computed the clustering coefficient and found it to be significantly higher than in random networks in citation graphs [Red98], protein-protein interaction networks [GR03, WF01], biological neural networks [MiOO⁺01, SGS⁺02], food webs [DWM02], social networks modeling on-line relations [MKG⁺08, ABA03] and many others.

Connected components

The connected components and their sizes are computed using a graph traversal (like a breadth-first search) in $\Theta(n)$ space and $\Theta(m)$ time. In most real-world complex networks, it has been observed that most of the nodes belong to a huge connected component, often called giant component, while the rest of the nodes (if any) belong to small connected components, like in Figure 3.6. There is a giant connected component for instance in citation graphs (ArXiv and patents) [LKF05], in the autonomous systems graph [LKF05], in a web graph of 39M pages in the .uk domain [Lat08], in metabolic networks [JMBO01], food webs [DWM02], email networks [NFB02] and many others.

In random graphs, for a low value of p , there are few edges and all the connected components are small, having an exponential size distribution with finite mean size. For high values of p , the graphs have a giant component with $O(n)$ of the nodes in the graph belonging to this component (where n is the total number of nodes). The rest of the components again have an exponential size distribution with finite mean size. The changeover (called the phase transition) between these two regimes occurs at $p = 1/N$.

Communities

Communities (or modules or clusters) are groups of nodes better connected between themselves (i.e. have more links) than to the rest of the network. A large body of work has been devoted to defining and identifying communities in complex networks. There exists agglomerative methods (where nodes are grouped into hierarchies, which are grouped themselves into high-level hierarchies and so on [Eve80]), divisive methods (where, starting with the whole graph, edges are removed in a prescribed order based on a given measure, as for instance edge-betweenness [GN02]), methods based on max-flow min-cut formulations [FLG00] or on Kirchoff's laws [WH04], local methods (based on local information [Vir03]), optimization methods (based on the maximization of an objective function [CMN04]) and many others. A very efficient algorithm for extracting communities in large graphs was proposed in [BGLL08]. For a survey on community identification, see for instance [For10].

The quality of the partitions resulting from these methods is often measured by the *modularity* Q of the partition, a measure of the density of links inside communities as compared to links between communities [New04, NG04]:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where m is the number of edges of the graph, A_{ij} is the adjacency matrix, k_i is the degree of the node i , c_i is the community to which the node i belongs and $\delta(c_i, c_j)$ is the Kronecker delta symbol, equal to 1 if $c_i = c_j$ and to 0 otherwise.

As to the significance of the identified communities, it has been observed that community-like sets of nodes tend to correspond to organizational units in social networks [New06], functional modules in biological networks [RSM⁺02] and scientific disciplines in collaboration networks between scientists [GN02] (see Figure 3.7).

Centrality

The centrality is a measure of the relative importance of a node within a network. There are several definitions of centrality; here we present the most commonly used:

- the degree centrality,
- the betweenness,
- the closeness,
- the eigen vector centrality,

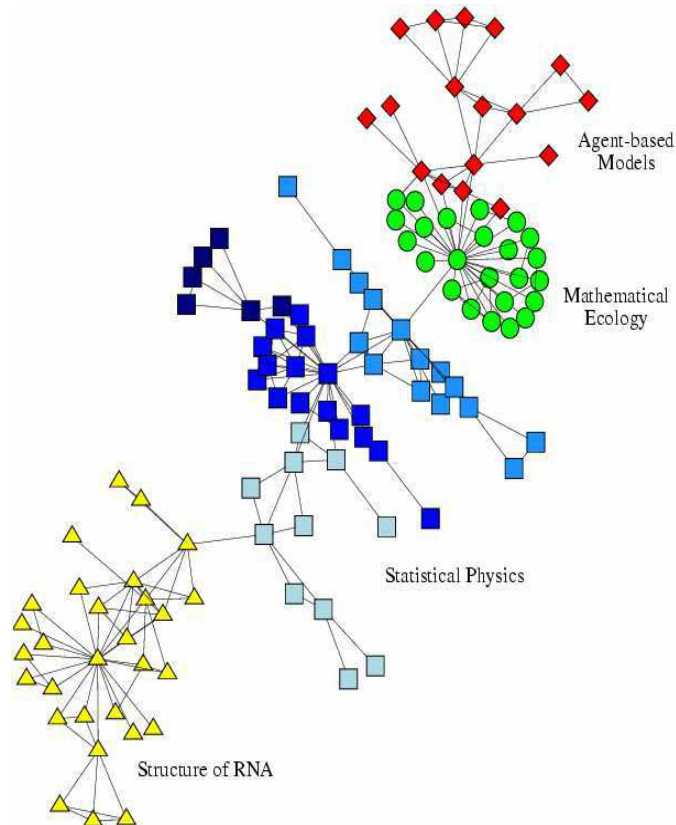


Figure 3.7: Communities. An example of a coauthorship network depicting collaborations among scientists at a private research institution [GN02]. Nodes in the network represent scientists, and a line between two of them indicates that they coauthored a paper during the period of study.

- the page rank.

The *degree centrality* is the simplest one. It is defined as the number of links a node has (the degree of the node) divided by $n - 1$ where n is the number of nodes of the graph (this is just for normalization; this way the range of values of the degree centrality is 0 to 1). Degree is often interpreted in terms of the immediate risk of a node for catching whatever is flowing through the network (such as a virus, or some information). While very simple and easy to compute, this measure does not really capture the importance of the node as some very high-degree nodes might be placed at the periphery of the network and thus be important only for a small part of the network. As explained in [Bar02], if one measures the degree centrality of nodes in the movie actors network (where two actors are connected by a link if they have acted together in a movie), the most central actors are found to be porno actors. Their importance in the movie network is however limited to the porno section and one can reasonably argue that there are other more important actors.

The *betweenness centrality* [Fre77] considers as central nodes that are placed on many shortest paths between other nodes; these nodes are important as one has to pass through them in order to travel efficiently in the different parts of the network. Thus, the betweenness centrality C_B of a vertex v is defined as

$$C_B(v) = \sum_{\substack{s \neq v \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths between s and t and $\sigma_{st}(v)$ is the number of such shortest paths that pass through v . This measure reflects better the notion of importance of a node than the previous one, but it is costly to compute (it takes $O(nm)$ time using the most efficient known algorithm [Bra01]) and thus difficult to use on large networks.

The *closeness centrality* considers as central nodes that are at a short distance from the other nodes (in the same connected component); thus the closeness of a node is the sum of the distances between this node and all the other nodes in its connected component divided by the number of nodes in the component (minus 1, as one does not take into consideration the node itself). Closeness can be regarded as a measure of how long it will take information to spread from a given node to other reachable nodes in the network. Computing the closeness means computing the shortest distance from one node to the other ones which can be done for each node in $\Theta(m)$ time and $\Theta(n)$ space with a breadth-first search (BFS). As the closeness of a node makes sense when compared to that of other nodes of the graph, one needs to compute it for (all the) other nodes of the graph, so the time complexity is multiplied by the number of nodes. An efficient randomized approximation algorithm for computing closeness centrality in weighted graphs has been proposed in [EW04]; this algorithm estimates the centrality of all vertices with high probability within a $(1 + \epsilon)$ factor, $\epsilon > 0$, in near-linear time. See Figure 3.8 for an example of graph and Table 3.1 for the values of the degree, the betweenness and the closeness centrality in this graph.

The *eigen vector centrality* [Bon87] and the *page rank* [BP98] assign relative scores to all nodes in the network based on the principle that connections to high-scoring nodes

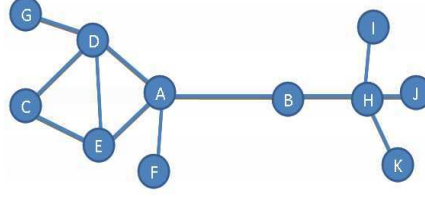


Figure 3.8: An example of graph.

Table 3.1: The degree, betweenness and closeness centrality of nodes A, B and G from Figure 3.8

node	degree	betweenness	closeness
A	4	$5 \times 5 + 4 = 29$	$1/10 \times (4 + 2 \times 3 + 3 \times 3) = 1.9$
B	2	$4 \times 6 = 24$	$1/10 \times (2 + 2 \times 6 + 2 \times 3) = 2$
G	1	0	$1/10 \times (1 + 2 \times 3 + 2 \times 3 + 4 + 3 \times 5) = 3.2$

contribute more to the score of the node in question than equal connections to low-scoring nodes.

The presented properties are some of the measures one usually computes in complex networks. These properties can be grouped in three categories depending on the level where the analysis is done. Thus there are:

global properties computed by taking into consideration the whole network; these are the degree distribution, the diameter, the connected components etc.;

local properties computed for each node, by taking into consideration the neighborhood of the node; the clustering coefficient of nodes is such a measure;

intermediate properties computed by taking into consideration the way each node is connected to the network in the context of the entire network; the identification of communities and the nodes centrality belong to this approach.

To sum up, there is a set of properties that are significantly different for real-world large graphs and for randomly generated ones. This means that edges in real graphs are not randomly created, but there are factors that influence their creation. Many researchers have tried to explain the formation of edges and, this way, the evolution of complex networks. Many models of network generation have been thus proposed. We will present some of them in the next section.

3.2 Models of networks and random generation of networks

This section presents first several *models* of network generation and then some algorithms for *random generation* of networks. First of all, it is important to distinguish between the

two approaches. In the *construction of models of networks*, the goal is, given an original network, to *explain the formation of its links*. Therefore one can generate an artificial network from one vertex to a complex object where the resulting network reproduces several properties of the original one. However, the resulting network is not randomly chosen among all the networks with those properties. That is, there may be some networks that share all the input properties but who never get generated by the model. This is not a problem since the model does not try to generate all the possible networks with the input properties but to give an explanation for the formation of links in the original network. On the contrary, the goal of *the random generation of networks* is precisely to *generate networks that are randomly (i.e. with the same probability) chosen* among all the networks that have the input properties. In this case, any network with those properties is generated with an equal probability. If the first approach proposes models in order to explain the formation of links and therefore the evolution of the network, the second approach proposes generations of networks that are then used as null models. That is, they are used as a general characterization of all the networks with the input set of properties. One can use the null model in order to see if the original network has some other properties that distinguish it from the null model (or, on the contrary, it is just one ordinary network with the input properties).

Models of networks

The simplest model of network is the Erdos-Renyi model that was discussed earlier. In this model no condition is imposed for the formation of links: any two nodes can be connected by a link with the same probability. However, the graphs generated by this model are very different from real-world complex networks. Therefore there must be a logic, a reason behind the formation of links: the links are not randomly created but generated by one or several factors.

The first model of graph generation after the Erdos-Renyi model was that proposed by Watts and Strogatz [WS98]. This model, introduced nearly 40 years after that of Erdos and Renyi, was the first one to generate graphs sharing some of the properties of real-world complex networks. In this model links do not connect random pairs of nodes, but each node is connected to k of its closest neighbors (nodes are displayed on a circle). Next, for each node u , each of its edges (u, v) is rewired with probability p to form some different edge (u, w) , where node w is chosen uniformly at random. The parameter p gives the randomness of the generated graph: when $p = 0$ the graph is completely ordered and when $p = 1$ the graph is completely random. Between the two, there is a broad region of values of p in which the clustering coefficient of the network is rather high and the average shortest path length is low.

Another model introduced just after that of Watts and Strogatz tried to explain another property of large complex-networks: the heterogeneous right skewed distribution of the degree. This model, introduced in [BA99] and known as the "preferential attachment model", contains two mechanisms: population growth and preferential attachment. The intuition behind the first mechanism is straightforward: real networks grow in time as new members join the population. The mechanism of preferential attachment, analogous to Simon's "Gibrat principle" [Sim55b] and Merton's "Matthew Effect" [Mer68], expresses the

idea that newly arriving nodes will tend to connect to already well-connected nodes rather than poorly connected ones. Specifically, Barabasi and Albert defined the probability that a new node connects to an existing node with degree d as $c \times d$ (where c is a normalizing constant). Barabasi and Albert showed that over a sufficiently long time horizon, the degree distribution of a growing network exhibiting linear preferential attachment would converge to a power-law with exponent $\gamma = 3$.

The graphs generated by the two models, however, do not exhibit all the properties of real-world complex networks. In the first model, the shape of the degree distribution is similar to that of random graphs generated by the Erdos-Renyi model. For the graphs generated by the second model, the power-law exponent of the degree distribution is fixed at $\gamma = 3$ (while many real-world graphs deviate from this value), there is exactly one connected component (while many real-world graphs have several isolated components), the average degree is constant (while the average degree of some real-world graphs increases over time [BJN⁺02, LKF05]).

Many other models have been proposed since these two initial ones. Each model tries to explain different properties observed in real-world complex networks as for instance the shrinking diameter (this is done by the *forest fire model* proposed in [LKF05]), the increasing average degree (this is done by the model proposed in [BJN⁺02]), community behavior (two models [KKR⁺99, KRRT99] try to explain this) and many others. See for instance [CF06] for a detailed presentation of existing network models.

Random generation of networks

Given a set of properties, a generator of random graphs must produce graphs that are randomly chosen among all the graphs presenting that set of properties. Usually the properties are computed in an input graph for which one needs to build null models. Several existing generators produce graphs that preserve the degree distribution of the input graph. It is the case for instance of the generator introduced in [VL05]¹ that generates simple connected graphs; this generator needs as input a set of pairs of degree and number of vertices with that degree. Another generator that preserves, for each node, its in-degree and its out-degree, was used in [MSOI⁺02]²; graphs generated this way served as null model for finding network motifs as we explain in the following section.

Sometimes one needs to generate graphs that have not just a given degree distribution but also other properties. We present here a generator introduced in [MKFV06]³ based on *dk-series*.

The algorithm introduced in [MKFV06] generates graphs that preserve the dk-series distribution of the given input graph. dk-series describe correlations amongst degrees of nodes in subgraphs of size d, for $d = 0, 1, \dots, n$. For instance, when $d = 3$ and the input graph is undirected, the 3k-distribution contains the number of connected triplets with degrees k_1, k_2, k_3 for all $k_1, k_2, k_3 \in \mathbb{N}$. The connected triplets can be triangles and 3-nodes paths, so one counts separately the triangles and the 3-nodes paths with degrees k_1, k_2, k_3 . The generated

¹Tool available at <http://fabien.viger.free.fr/liafa/generation/>

²Tool available at <http://www.weizmann.ac.il/mcb/UriAlon/>

³Tool available at http://www.sysnet.ucsd.edu/~pmahadevan/topo_research/topo.html

graphs that preserve the $3k$ -serie of the input graph will have the same number of triangles and of 3-nodes paths as the input graph; moreover the connected 3-nodes subgraphs of the generated graphs will have exactly the same combinations of degrees as the input graph.

When $d = 0$, the generated graphs have the same average degree as the input one. When $d = 1$ the degree distribution is preserved. When $d = 2$, the generated graphs have the same number of edges with degrees k_1, k_2 for all $k_1, k_2 \in \mathbb{N}$. The dk -series have two important properties: first, graphs having a dk -distribution also have the $d'k$ -distributions, with $d' < d$; second, generated graphs are more and more similar to the input graph when d increases, ending up isomorphic to it when $d = n$. Using this approach, the authors construct graphs for $d = 0, 1, 2, 3$ and demonstrate that these graphs reproduce, with increasing accuracy, important properties of measured and modeled Internet topologies. They find that the $d = 2$ case is sufficient for most practical purposes, while $d = 3$ essentially reconstructs the Internet AS- and router-level topologies.

3.3 Identification of patterns in complex networks

Frequent patterns

In numerous analysis like mining biochemical structures, program flow control study, graph comparison or compression etc., one needs to compute the number of occurrences of a graph Q as subgraph in the graphs G_1, G_2, \dots, G_n of a given database D . This is the *graph query problem*. Often one needs to solve a problem close to this one, the *frequent graph patterns problem*, where one computes all the graphs Q that are subgraph of a number of graphs in D , this number being higher than a given threshold (this number is called the support of Q). There are several algorithms for solving these problems; they can be grouped in:

- graph-theory based algorithms,
- greedy algorithms,
- algorithms using inductive logic programming.

For the *graph-theory based algorithms*, one usually follows the general principal of the Apriori algorithm introduced in [AS94] for association rule mining: in a "bottom up" approach, frequent subsets (here graphs) are extended one item at a time (a step known as candidate generation), then candidates are tested against the data. The algorithm terminates when no further successful extensions are found. For instance, *AGM*[IWM00] is an algorithm based on this idea that uses canonical codes for adjacency matrices and therefore for subgraph matching. Frequent subgraphs are generated in the bottom-up order by adding one vertex at a time (two already found frequent graphs with the same number of vertices are joined together in a candidate graph that has one more vertex). However this algorithm suffers from computational intractability when the graph becomes too large. Another algorithm, *FSG* proposed in [KK01], uses the same scheme: starting with frequent graphs with 1 and 2 nodes, it successively generates larger frequent graphs by adding one edge at a time. The algorithm expects a graph with colored edges and

nodes; however one usually analyzes graphs that are a special case, having all nodes and edges of only one color. Also, the algorithm needs to solve the graph and subgraph isomorphism problems repeatedly which is very slow and inefficient for graphs with only one color. The algorithm *GSPAN* introduced in [YH02] uses a different canonical code for the graphs Q based on depth-first search; this coding scheme gives faster results. The same canonical code is used in [YH03] for mining closed frequent graphs i.e. graphs Q that are not contained in other graphs with the same support. This algorithm, called *CloseGraph* uses an efficient scheme for generating candidate graphs based on the DFS trees of the already found graphs: new edges are added from the last discovered vertex in the DFS tree to any other vertex situated on the path from the first discovered vertex to the last discovered one in the DFS tree; new vertices are added by linking to this path.

In the *inductive logic programming approach*, first order predicates are used in the description of frequent subgraphs. The *WARMR* algorithm [DT99] uses this method; however it needs to check for equivalence of different first-order clauses which is NP-complete. The algorithm *FARMAR* [NK01] uses a weaker equivalence condition to speed up the search.

In the *greedy approach*, the graphs Q are chosen such that they minimize a given measure. For instance, the algorithm *SUBDUE* [HCD94] solves a problem related to that of finding frequent graphs, that of compressing input graphs using frequently occurring subgraphs. The subgraphs are chosen to minimize a measure called minimum description length. As in the Apriori approach, new subgraphs are found by adding new edges; the generation is stopped when no new subgraphs are found. The algorithm also allows inexact matching of subgraphs by assigning a cost to each distortion, like deletion, insertion or substitution of nodes and edges.

Network motifs

A problem related to the previous ones is that of *identifying network motifs* introduced in [MSOI⁺02]. Given a graph G one searches for *motifs* i.e. small structures that appear in G more often than in random graphs. The analysis begins with the identification of all the connected subgraphs of G with a given number of nodes; the subgraphs are directed if G is directed, else they are undirected. For each possible connected graph with that number of nodes (up to isomorphism), its number of occurrences as subgraph of G is compared to its number of occurrences in several (e.g. 1000) randomly generated graphs. The graphs are generated such that they have the same degree distribution as G i.e. the same number of nodes with a given in-degree and out-degree. Structures that appear significantly more often in G than in the randomly generated graphs are called motifs.

In [MSOI⁺02] Milo et al. identify motifs with 3 and 4 vertices in several real-world complex networks extracted from biochemistry (transcriptional gene regulation), ecology (food webs), neurobiology (neuron connectivity), and engineering (electronic circuits, World Wide Web). They find several structures to appear more often than in randomly generated networks; moreover some of these motifs are shared by several real-world networks as shown in Figure 3.9. The authors also offer some possible explanations for the occurrences as such motifs. For instance, the World Wide Web motifs may reflect a design aimed at short paths between related

pages. This similarity in motifs in the neuronal connectivity network and in the transcriptional gene regulation network may point to a fundamental similarity in the design constraints of the two types of networks. Both networks function to carry information from sensory components (sensory neurons/transcription factors regulated by biochemical signals) to effectors (motor neurons/structural genes). The feed-forward loop motif common to both types of networks may play a functional role in information processing. One possible function of this circuit is to activate output only if the input signal is persistent and to allow a rapid deactivation when the input goes off. Indeed, many of the input nodes in the neural feed-forward loops are sensory neurons, which may require this type of information processing to reject transient input fluctuations that are inherent in a variable or noisy environment.

A lot of excitement has surrounded the network motifs approach, the original paper of Milo and al. [MSOI⁺02] being cited over 1600 times as of March 2010. The analysis of network motifs has led to interesting results in the areas of protein-protein interaction prediction [AA04], hierarchical network decomposition [ILK⁺05], temporal gene expression patterns [SOMMA02] and many others. However this method has also received some criticism. First, the method assumes that matching the degree distribution of the graph in the randomly generated ones gives good null models; however the motifs found under this assumption might not be statistically frequent if one uses a better graph generator. Second, Vazquez et al. [VDS⁺04] demonstrated that global network features such as the clustering coefficient also influence local features such as the abundance of certain subgraphs. Artzy-Randrup et al. [ARFBTS04] found that certain network models (such as "preferential attachment" [BA99]) lead to a display of motifs although there is no explicit selection mechanism for local structures. Milo et al. answered this criticism in [MIK⁺04] by suggesting not only to look at the overabundance of individual subgraphs but rather at a broader picture in the form of so-called "subgraph significance profiles".

In the computation of network motifs one searches for small structures that appear more often than in random graphs, while in the computation of frequent patterns one searches for structures that appear frequently. It can be also useful to simply count the small structures and then use their distribution, as for instance for network comparison. This is the approach adopted by Przulj in [Prz06]. The author proposes a method for measuring graph similarity using "graphlet degree distributions" (which count the occurrences of the vertices of the graph in small connected non-isomorphic subgraphs). As we will see in the following chapters, this is also the approach we adopt in this thesis, although for a different goal.

In this approach and also in the identification of network motifs one needs to list (or at least to count) all the subgraphs with a given number of nodes. For the counting of subgraphs, some authors proposed algorithms for counting different type of graphs (as for instance cycles in [AYZ97] or connected undirected graphs with 4 nodes in [KKM00]) or for estimating their total number from a randomly sampled set of subgraphs [KIMA04]. For the actual listing of all the subgraphs with a given number of nodes, the most efficient algorithm to our knowledge is *ESU* which was proposed in [Wer06]. We will present this algorithm in more details in Section 5.2.





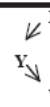
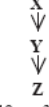
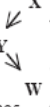


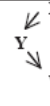
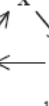

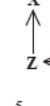


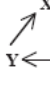
Network	Nodes	Edges	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score
Gene regulation (transcription)				Feed-forward loop		Bi-fan					
<i>E. coli</i>	424	519	40	7 ± 3	10	203	47 ± 12	13			
<i>S. cerevisiae</i> *	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons				Feed-forward loop			Bi-parallel				
<i>C. elegans</i> †	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs				Three chain		Bi-parallel					
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	13			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			
Electronic circuits (forward logic chips)				Feed-forward loop			Bi-parallel				
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)				Three-node feedback loop			Four-node feedback loop				
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web				Feedback with two mutual dyads			Uplinked mutual dyad				
nd.edu§	325,729	1.46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4 ± 4e2	15,000	1.2e6	1e4 ± 2e2	5000

Figure 3.9: Network motifs found in biological and technological networks [MSOI⁺02].

Chapter 4

Social networks

A social network is a modeling of a set of relations among a set of individuals. It can be seen as a graph where the vertices are the individuals and the edges are the relations between them. Traditionally, this has been a domain of study for sociologists and anthropologists who analyze the connection between individuals or collective behaviors and the social structures in which individuals are embedded. The idea of "social network" has been used for over a century, Georg Simmel being the first scholar to think directly in social network terms at the beginning of the twentieth century [Sim55a] (English version). His essays pointed to the nature of network size on interaction and to the likelihood of interaction in ramified, loosely-knit networks rather than groups. In the 1930s, Jacob L. Moreno pioneered the systematic recording and analysis of social interaction in small groups, especially classrooms and work groups. In 1954, John A. Barnes [Bar54] started using the term "social network" systematically to denote patterns of ties, encompassing concepts traditionally used by the public and those used by social scientists: bounded groups (e.g., tribes, families) and social categories (e.g., gender, ethnicity). The field developed with the works of Elisabeth Bott on kinship [Bot57], of Sigfried Nadel on social structure [Nad57], of Harrison White and his students at Harvard University and many others. For instance Mark Granovetter and Barry Wellman (whose principal works will be presented in the following sections) are among the former students of White who have elaborated and popularized social network analysis.

Generally, the analysis of the structure of interpersonal relations can offer insights about the persons' sociability and can explain their actions. For this type of analysis, sociologists and anthropologists usually obtain their data from interviews with the analyzed people. This data, although very detailed, can be difficult to obtain. The process of interviewing people is often long and costly and the obtained datasets rather small, with several hundreds of analyzed relations in the best of cases.

Recently, new sources of data have been used. With the development of internet, mobile communication, computer capacity, one can easily access traces of interpersonal communication such as activity on online platforms, phone calls, emails, instant messages etc. The obtained datasets are large, possibly containing millions of communications. The access to data is much easier than before but the obtained sets are less detailed as one sees

only a fraction of the interactions between people. Imagine for instance the case of mobile phone communications. While there is a recording for each mobile phone call between a certain set of people, one has no idea if those people contact each other by other means. If two persons do not call each other by mobile phone, this does not mean that they do not speak to each other at all; maybe they use the land phone, send emails or just talk to each other all the day long since they work in the same office. Also, if during interviews, one can ask details about the relation between two persons (for instance friendship, family, work etc.), here one does not have any information about this relation. However, data obtained from traces of communications has two important qualities: it is about many people and it is easy to obtain. This gives the opportunity to answer questions which previously remained unanswered: How do interpersonal relations change over time? How can we detect "abnormal" interactions (such as spam in an e-mail network)? How are items of information and viruses spread in the network? How can we identify influential people in the network?

Data obtained from traces of communication is a topic of interest for many researchers nowadays and not only sociologists and anthropologists but also computer scientists, mathematicians, physicists. The datasets are modeled as large social networks which are, after all, complex networks. Therefore all the discussion in Section 3.1 applies here. We will present several studies on large datasets obtained from traces of online activities and mobile phone communications in Sections 4.3 and 4.4. Before, we discuss some of the major findings in social network analysis before the era of large datasets in Section 4.1. Section 4.2 presents a special case of the social network analysis, the one centered around one individual, called egocentred analysis.

Remark. As said before, social network analysis has been traditionally a field of study for sociologists. Nowadays, researchers from many other domains, including computer scientists, analyze traces of inter-personal communications and thus study social networks. However, the goal is often a "sociological" one: analysis of people behavior, of their uses of different online platforms, of the mobile phone etc. Of course, it is not straightforward what to search; one has to have a good intuition and some real sociological questions in mind before starting to analyze traces of communications. Nevertheless, the core of the research is often based on observations. Thus the focus is on the interpretation of these observations rather than on the way of producing them. Many researchers do not even mention the methodology they used in order to make the observations, how long it took etc. It would be interesting to make a survey of the topics encountered in social network analysis, and especially of those regarding online platforms, and formalize them in graph theory notions, algorithms and complexity. However, this is not our goal here. We present several central questions in social network analysis and several advances on analysis of uses, as they are found in the literature. We do not intend here to formalize them, but to present them as they were published, that is focusing on observations and their interpretation, rather than the way of producing them.

4.1 Questioning and advances

The domain of social network analysis has been marked by several topics initiated by now-famous researches. One of them is Milgram's experiment [Mil67, TM69] about the *average distance* between two random persons. In this experiment participants had to reach randomly chosen individuals in the U.S.A. using a chain letter between close acquaintances. Their surprising find was that, for the chains that completed, the average length of the chain was only six in spite of the large population of individuals in the social network. While only around 29% of the chains were completed, the idea of small paths in large graphs was still a landmark find. This observation, known as "small world" or "six degrees of separation", was not explained until late 90's. It is the model proposed by Watts and Strogatz [WS98] presented in Section 3.2 that offered a first possible explanation.

Another issue that keeps cropping up in social networks analysis is that of the *strength of ties*. A person is connected to other persons by links that have different meanings and also different strengths. There are for instance friends who are sociologically closer than others, family members who are closer than work partners etc. In [Gra78], Granovetter analyzed the role of the different types of links a person has in finding a job. By interviewing people who had obtained a job in the previous five years, he observed that the persons who provided useful information about a job were rarely family or friends, but rather acquaintances who were in different occupations than the respondent. This observation is known as "the strength of weak ties". The explanation is that those to whom a person is closest (family and close friends, workmates etc.) interact with one another in numerous situations, so probably possess the same knowledge about job opportunities. Therefore they are less likely to be the sources of new information than more distant contacts. It is through the relatively weak ties of less frequent contacts and of people in different work situations that new and different information is likely to become available.

Another idea related to this one is that of *social capital* which means essentially that better connected people enjoy higher return on their efforts. An individual occupying some special location in the social network might be in a position to broker information or facilitate the work of others or be important to others in some way. This importance could be leveraged to gain some profit. However, the problem is: what does better connected mean? In general, there are two viewpoints on what generates social capital. The first one is that of *structural holes* introduced by Burt in [Bur92]. Weak connections between groups are holes in the social structure, and create an advantage for individuals whose relationships span the holes. Such individuals get lots of brokerage opportunities and can control the flow of information between groups to their benefit. The second one is that of *network closure* introduced by Coleman in [Col88]. This is the view that networks with lots of connections are the source of social capital. When the social network around an actor A is dense, it means that information flow to A is quick and usually reliable. Also, the high density means that no one around A can escape the notice of others; hence, everyone is forced to be trustworthy (or face losing reputation). Thus, it is less risky for A to trust others, and this can be beneficial to him. Although these two points might look completely opposite, Burt [Bur01] finds that they actually supplement each other. If a group has high closure but low contacts across holes, the group is cohesive but has only

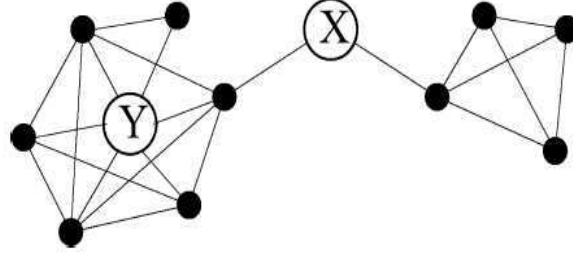


Figure 4.1: Social capital. The two concepts are illustrated by nodes X (structural holes) and Y (network closure) [CF06].

one perspective/skill. Low closure but high contacts across holes leads to disintegrated groups of diverse perspectives. Thus the best performance is achieved when both are high. Figure 4.1 (taken from [CF06] which explains very well the notion of social capital) illustrates the two concepts of social capital: the node Y benefits of network closure, as it is in the middle of a dense web, while X bridges the structural hole between the two clusters.

Another important topic in the analysis of social networks is that of identifying *social roles*.

4.1.1 Social roles

This notion refers to the position of an actor in society and it is based on the relationships that the actor in question has with other actors. Actors playing a particular social role are connected in the same way to the network. Generally nodes having the same role have to be equivalent or similar to each other by some metric. Several definitions of social roles have been proposed.

Modules. Given a graph, one can compute its modules [Gal67, HM79] and then consider that the vertices belonging to the same module have the same role.

Definition 4.1.1. Let $G = (V, E)$ be an undirected graph. A module of G is a subset of vertices $M \subseteq V$ such that for any $v \in V \setminus M$ one has either $N(v) \cap M = M$ either $N(v) \cap M = \emptyset$.

So a module is a group of vertices that are "seen" in the same way by the vertices not belonging to the module: if a vertex from the exterior of the module is linked to a vertex in the module, then it is linked to all the other vertices of the module.

We also present three well-known social roles definitions based on equivalence relations.

Structural equivalence [LW71]. Two nodes are considered as equivalent if and only if they have exactly the same neighbors in the graph, so they are linked to exactly the same set of nodes with (in the case of directed graphs) the arrows pointing in the same directions.

Definition 4.1.2. *Two vertices u and v of a graph G are equivalent with respect to the structural equivalence if and only if $N(u) = N(v)$.*

Thus, two structurally equivalent actors can exchange their positions without changing the network. For the graph in Figure 4.2, the structural equivalence divides the nodes into seven classes: $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E, F\}$, $\{G\}$ and $\{H, I\}$; there are only the nodes E and F , H and I respectively that are equivalent as they have exactly the same neighbors. Note that the classes of vertices defined by the structural equivalence in a graph are modules of the given graph. Structural equivalent vertices are also called *false twins* in graph theory.

Several researchers have shown that identifying nodes with identical neighborhoods does not correspond to the intuition of social roles [Sai78, JBLE01]. It is not frequent to find two persons with identical relations. There are examples of actors who play the same role without being connected to exactly the same people, but rather have similar relations with people who have themselves a same role. Two fathers, for example, will have different sets of children to whom they relate, but they might be expected to behave, in certain respects, in similar "fatherly" ways towards them. The two men occupy the same social position, that of father, even though they are not connected to the same people. There are two relations that express this idea: the automorphic equivalence and the regular equivalence.

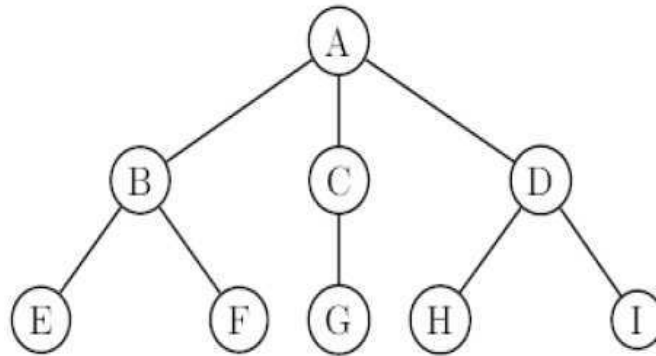


Figure 4.2: An example of graph.

Automorphic equivalence. Two nodes are considered as equivalent if one is the automorphic image of the other one.

Definition 4.1.3. *Two vertices u and v of a graph G are automorphically equivalent if there is an automorphism φ of G such that $\varphi(u) = v$.*

The idea is to consider as equivalent actors who are embedded in structures with similar inner links. Roughly, the actors' "faces" are different but the structures are identical. For the graph in Figure 4.2, the automorphic equivalence divides the nodes into five classes: $\{A\}$, $\{B, D\}$, $\{C\}$, $\{E, F, H, I\}$ and $\{G\}$.

Regular equivalence [BE89]. Two nodes are considered as equivalent if they are connected to equivalent nodes.

Definition 4.1.4. *Given a graph $G = (V, E)$, let $r : V \rightarrow \mathbb{N}$ be a role assignment for the vertices in V ; the function r can be seen as an attribution of colors to the vertices in V . Also the function r defines an equivalence relation on the vertices in V . This relation is said to be regular if and only if for all $u, v \in V$ such that $r(u) = r(v)$ one has*

$$\{r(i); i \in N(u)\} = \{r(i); i \in N(v)\}$$

So if two nodes are equivalent, the colors found in the neighborhood of one node are also found (possibly in different numbers) in the neighborhood of the other node. For the graph in Figure 4.2 one possible regular equivalence is that with the following classes: $\{A\}$, $\{B, C, D\}$ and $\{E, F, G, H, I\}$, so there are 3 colors: one for the vertex A , another one for the vertices B, C and D and another one for the vertices E, F, G, H and I .

Note that this definition is circular: to check if two vertices u and v are equivalent, one has to check if their neighbors are equivalent, and therefore if u and v are equivalent. Algorithms for computing regular equivalences have been proposed for instance in [EB93].

The three definitions presented here are often too strict for real-world data. The problem is that only few nodes are found as equivalent when using these definitions on real-world graphs. Thus the equivalence classes are much too numerous, often of the same order of magnitude as the number of vertices of the graph. The idea behind social roles is however to group nodes in a small number of clusters that can be easily used. Therefore one generally uses different heuristics like the computation of a *distance* between nodes: equivalent nodes are at distance 0 from each other and similar nodes are at a small distance. This distance can be defined for instance as a certain correlation coefficient or similarity measure between nodes. Or it can be defined using vectors characterizing the nodes; the distance between the nodes is then the (e.g. Euclidian) distance between their vectors. After having defined a certain distance between nodes one can use a clustering algorithm (as presented in Section 2.2) in order to group nodes into clusters: nodes in the same cluster are close to each other with respect to the given distance. The advantage is that one can place the number of clusters where he wants, going from a small number of clusters (if this is his goal) to a large number of clusters, where nodes in the same cluster are very similar to each other.

We presented here only some of the topics of social network analysis. For a review of social network methods see for instance [WF94].

4.2 Egocentred analysis

In the analysis of social networks, a special part is that of the study of personal relations i.e. the different relations a given individual (called *ego*) has with other individuals (often called *alters*). This is called an egocentred approach because the social relations are seen from ego's point of view. The analyzed relations are generally obtained by interviews

with ego who describes his relations with the other persons and, sometimes, the relations between these persons. Thus, ego is asked about the people he knows, with whom he interacts, about the importance of these people in his life etc. One problem here is the type of questions to ask ego (for instance he is asked to cite the persons he knows, but what does "know" mean?); this is very important since the recorded data depends entirely on it. This becomes even more complicated when ego is asked to cite the relations between his alters, as he may have a wrong impression about these relations. The advantage, however, is that the interviewer can ask whatever questions he wants, thus obtaining very detailed and meaningful data.

While the analysis of ego's relation has a long history in anthropology and sociology [RB40, Bar54, Sim55a, Bot57, Mit69, Boi74], the approach is not always a social network one in the sense that the structure of the network is not studied; rather, a given relation between two persons is analyzed in the context of the existence of other relations. Thus, the focus is predominantly on the different properties of the two individuals and of their individual relationships and not on the notion of network as structured configuration. Such analysis has been done for instance on romantic relationships [Sur88, PSE83, SEE92] or on the notion of social support [MCN97, BCP02, ATSK04].

In order to analyze personal networks using a social network approach, one needs to define ego's network (also called egocentred or personal network or personal community). One has to define a network (or a graph) so a set of nodes and a set of links. The nodes are usually ego and his alters but one can also consider their alters (so the nodes two steps away from ego) etc. The presence of the different people is mentioned by ego during the interviews. For the links, there is a link between ego and each one of his alters. The links between alters, if present, are also given by ego.

Once one has built such an egocentred network, one can study its structure: the number of nodes, links, the density, the clustering coefficient etc. The analysis of the different patterns occurring in people's personal networks is important because it can show how different networks are structured and why, what part different social and personal factors (e.g. gender, age, mobility histories, ethnicity, profession) play in this, how a person is socially integrated etc.

One of the most influential network analysts from a personal relationship perspective, Barry Wellman, consistently argued that a network approach is fundamental to understanding the character of contemporary society and the role the personal relations play within this. In a series of reports based on data collected from East York, a suburb of Toronto, in the 1970s [Wel79, Wel82, Wel85, WW90], Wellman has been particularly concerned with the ways individuals are integrated in social life. In [Wel82, Wel88] he distinguishes several configurations of relations in terms of network structure: people embedded in quite dense networks, people having several subsets of alters and also people where alters have little to do with each other. In France, egocentred analysis has been made popular by Maurizio Gribaudo who introduced the methodology of notebooks of contacts [Gri98], Michel Grossetti who studied social structures of personal relations in the Toulouse area [Gro05], Dominique Cardon and Fabien Granjon who analyzed the relation between cultural practices (media-related, recreational, communication etc.) and personal networks structure [CG05], and many others.

If in the studies presented in this section the data came from interviews, in the following ones the data comes from traces of phone communications and online activities.

4.3 Phone communications

We present here an overview of existing studies on large social networks extracted from two different environments: phone communications and online activities.

Several researchers analyzed the phone usage, the way people communicate by mobile or fixed phone in terms of duration or frequency of calls, depending on the sociological or geographical distance between people, on their gender etc.

In [SL00] Licoppe and Smoreda studied the relationship between the duration of fixed phone calls and the gender of the two persons in communication. The authors used telephone billing records on several hundreds of adult men and women during 4 months. The analysis showed that the duration of calls is correlated with receiver's gender and it is in average longer when a woman is called. Also, their in-depth conversation analysis suggested that politeness rules governing the telephone call can explain in part why it is the gender of the receiver that has the biggest effect on how the call is managed and on its overall duration. The conversations involving women tended to go through longer introductive sequences, to be more multi-thematic and digressive in nature with a corresponding lengthening and multiplication of closure sequences; meanwhile the conversations with men tend to be linear and monothematic. In the main, the callers seem to adjust their interaction style to the gender of the receiver.

The same authors analyzed the relationship between the intensity of calls and the distance between the two persons.

In [LS05] Licoppe and Smoreda, using phone calls databases and interviews focusing on the use of telephone, identified two patterns of communication: the "connected presence" and the "intermittent presence". In the first pattern of communication, the "connected presence", the two persons, socially and often also geographically close, are frequently in contact with each other, exchanging many short calls and messages. They share activities that require numerous calls for synchronization and coordination. In the second pattern, the "intermittent presence", the two persons, close friends or intimate relatives, are not able to see each other or talk very often. Their conversations are long, they give and receive news, trying to compensate for the rarity of face-to-face contacts.

These analyses have been done using fixed phone data. Nowadays, the development of mobile phones and their worldwide spread (a penetration larger than 40% worldwide and close to 100% in the developed countries) offer new possibilities of analysis. The mobile phone, a individual and ubiquitous device offering voice and text communication features, has transformed the frequency and the geography of communication as compared to older fixed phone practices. We are now virtually always accessible to others wherever we are. This offers a useful insight into individual behavior. Of course, cellular phone communications do not fully capture social exchange. A social relation is expressed through multiple interaction channels such as email, land phone communications, instant messaging, face

to face interactions, the mobile phone communications capturing only a subset of the underlying social network. However, studies on the strength of ties have shown that mobile phone is among the most intimate communication tools; a mobile phone conversation suggests a certain relation between the two individuals, given that there aren't any listings of mobile phone numbers. Moreover, people that contact each other via one communication tool tend to communicate via other ones as well [Hay05], hence the relevance of analyzing mobile phone communications in the search of understanding the underlying social network.

Behavioral data coming from telecommunication operators offers the opportunity to revisit some older research on telephone usages. For instance, using mobile phone communication data, Lambiotte et al. [LBdK⁺08] were able to test the hypothesis that the existence of a call between two persons depends on the geographical distance between them. They thus show that the probability of a mobile phone call is inversely proportional to the square of the geographical distance between the two persons.

Also, new types of analysis are possible. For instance, one can use the location of the mobile phone when the communication began (also possibly when it ended) in order to study people mobility patterns. Extensive call records of any mobile phone carrier contain even more detailed information on the spatiotemporal localization of millions of users. This is due to the fact that mobile phones, in order to place outgoing calls and to receive incoming calls, must periodically report their presence to nearby cell towers, thus registering their position in the geographical cell covered by one of the towers. The analysis of such information for a better understanding of people mobility could be of high interest for urban planning, public transportation design, traffic engineering, disease outbreak control and disaster management. Several studies try to discover the patterns of mobile phone users mobility [BDE09, GHB08, EP05] and to predict their trajectory. In this direction, Song et al. [SQBB10] measure the entropy of each individual's trajectory, thus finding a high potential predictability in user mobility.

Mobile phone communications have also been modeled as complex social networks and analyzed accordingly. In several studies the nodes of the modeling graph are the individuals communicating by mobile phone during a given period, while the links correspond to reciprocal communications: two nodes are connected by a link if there had been a least one communication between the two persons in each direction (i.e. *A* called *B* and *B* called *A*). This procedure eliminates one-way calls that suggest that the caller does not know the receiver personally.

In this approach, Onnela et al. [OSH⁺07b] used a graph modeling mobile phone communications where they computed the degree distribution. As expected, most users communicate with only a few individuals while a small minority talks with dozens. However, the degree distribution decays very fast, so the hubs (high-degree nodes) are few; this is different from the case of land lines and of emails where well-connected hubs are present. This situation is probably rooted in the fact that institutional phone numbers, corresponding to the vast majority of large hubs in the case of land lines, are absent, and in contrast with e-mail, in which a single e-mail can be sent to many recipients, resulting in well-connected hubs, a mobile phone conversation typically represents a one-to-one communication. The authors define the weight

of a link (its strength) as the total duration of mobile phone communications between the two persons; they study the relationship between the strength of ties and their betweenness centrality, finding that the two are negatively correlated. Next, the authors analyze the importance of links of different strength for the robustness of the network (actually of the largest connected component that contains 84% of the nodes). They thus find that the removal of the weak ties leads to a sudden, phase transition-driven collapse of the whole network. In contrast, the removal of the strong ties results only in the network's gradual shrinking but not its collapse. By simulating information diffusion in the network, they find that the process of diffusion changes dramatically when the strength of links is taken into consideration (as opposed to the situation where all the links are considered as having equal weight). Moreover, in contrast with the theory of the importance of weak ties in information access [Gra78], they find that both weak and strong ties have a relatively insignificant role as conduits for information ("the weakness of weak and strong ties"): the former because the small amount of on-air time offers little chance of information transfer and the latter because they are mostly confined within communities, with little access to new information. They finally conjecture that communication networks are better suited to local information processing than global information transfer.

The same authors develop the analysis of links weight in [OSH⁺07a] in mobile phone communications networks.

In [OSH⁺07a] Onnela et al. take into consideration both the duration of calls and the total number of calls between the two persons as weight of the link connecting them. Besides computing classical measures, such as different distributions, the authors define the intensity of a subgraph as the geometrical mean of the weights of its links. They count the number of fully connected subgraphs (i.e. cliques) with 2 to 10 vertices in their network and in randomly generated Erdos-Renyi networks, finding that cliques with more than 3 vertices appear a enormous number of times more often in the real graph than in the generated ones. Note, however, that it had already been observed that the number of triangles was a lot higher in real-world complex networks than in Erdos-Renyi graph, so the authors' conclusion is not surprising. When comparing the intensity of subgraphs in the real network and in a network where the links weight have been shuffled, the authors find that the real-world subgraphs have considerably higher intensities than the random ones. This shows that local organization of weights in the mobile phone graph is not random.

Several researchers analyzed the temporal dynamics of mobile phone networks e.g. the temporal stability of links. In [HRS08], Hidalgo and Rodriguez-Sickert define the persistence of a link over a set of time periods as the number of periods where the link is activated (there are reciprocal calls between the two persons during that period). They find that persistent links are more common with people with low degree and high clustering. Palla, Barabasi and Vicsek [PBV07] used mobile phone data to study the evolution of social groups. They found that large groups persist for longer times if they are capable of dynamically altering their membership, suggesting that an ability to change the group composition results in better adaptability. In contrast, the behavior of small groups displays the opposite tendency, the condition for long-term persistence being that their composition remains stable.

4.4 Online activities

Nowadays, the development of Web2.0 allows users to connect to platforms of social networking, sharing of photos, videos, blogging, where they interact, declare friendship relations and share contents, thus being active participants and not only simple visitors of sites. In this context, new digital practices have emerged for production and diffusion of contents and also for recommendation, tagging and social networking. Moreover each user is able to manage his own visibility and, throughout his profile, develop strategies for increasing the audience of his productions and therefore his popularity. Thanks to more and more precise tracking tools, he knows how many people viewed, commented, recommended, rated, and forwarded his work. These ratings, by increasing the users reflexivity about his popularity, strongly influence publishing and networking practices [Hal08], [HRW08], leading some authors to describe the Web as a huge space of competition for popularity [Waz09].

The analysis of usages and contents on these online platforms should help us anticipate users' expectations, develop recommendation systems, strengthen contents audience and growing of communities, improve segmentation and targeting of users.

Generally, one can adopt one of the three following approaches to analyze activity on online platforms:

1. analysis of usages (the way users act on these platforms),
2. analysis of the published contents (audience, diffusion etc),
3. analysis of the social networks that model the relations between individuals.

1 Usages. People connect and use the functionalities of online platforms in different ways, often with uneven frequencies ("bursts nature"). Generally, to measure users' activity, one looks at the traces left on the platforms, such as number of comments a user writes, number of photos he uploads etc. The different measures of activity are found to have a skewed distribution on news groups [FSW06], wikis [HBB07], online dating communities [HEL04], question answer forms [ZAA07, AZBA08].

In [GH06] Golder and Huberman analyze user activity on Del.icio.us, a site for recording bookmarks. Users can store bookmarks of webpages, in the same way as in browsers but with access from any computer, and can tag them with keywords. Delicious is considered "social" because, not only can one see his own bookmarks, one can also see all of every other users bookmarks. By analyzing two sets of Delicious data containing almost 20 thousand bookmarks, the authors observe that users vary greatly in the frequency and nature of their Delicious use. That is, some users use Delicious very frequently, and others less frequently. Also some users have large sets of tags, others have small sets, and there is very little correlation between the number of bookmarks a user stores and the number of tags he uses. Users' tag lists grow over time, as they discover new interests and add new tags to categorize and describe them, but the growth rates may be very different. The authors also identify different function of tags. Although a significant amount of tagging, if not all, is done for personal use rather than public

benefit, information tagged for personal use can benefit other users. In this way, Delicious functions as a recommendation system, even without explicitly providing recommendations.

When analyzing the way people make use of the functionalities of online platforms, one can for instance identify groups (or clusters) of individuals based on the measures of activity, as in [MAA08] for Youtube or [PCB⁺08] for Flickr users. All in all, the analysis of uses on online platforms gives us an idea of how different functionalities are used and can thus help developers of such platforms to improve them.

2 Contents. Many studies analyze the contents published on online platforms. For instance, some works concentrate on the success of such contents. Understanding the popularity characteristics is important because it can bring forward the latent demand created by bottlenecks in the system (e.g. poor search and recommendation engines, lack of metadata). It also greatly affects the strategies for marketing, target advertising, recommendation, and search engines.

In [CKR⁺07], the popularity of videos on Youtube (the world's largest site for publishing of videos), measured as number of views, is found not to be a perfect power-law. In the log-log plot, the distribution of the number of view is not a straight line at the two ends: the most popular and the least popular items. There are several possible explanations for this observation that contradicts Anderson's intuition of a "long tail" [And06]. For the least popular items, that receive fewer views than if the distribution was a straight line on all its length, it can be because many videos are of low interest to most users; these videos are often produced for small audience e.g. family members. Also, search or recommendation engines typically return or favor a small number of popular items [CR04, MBSA02], steering users away from unpopular ones. This way users cannot easily discover niche content because this content is not properly categorized or ranked. The most popular videos also receive fewer views than if the distribution was a straight line on all its length. A possible explanation, suggested in [GDS⁺03] for P2P downloads and adopted in [CKR⁺07] for Youtube videos, is that video content does not change (is immutable); therefore viewers are not likely to watch the same video multiple times, as they do for mutable web objects. Even the number of views of very popular items does not go past a certain limit, so there is a cutoff in the distribution.

As in the Pareto principle (or 80-20 rule), the audience is often concentrated on a minority of contents: on Youtube 10% of the top popular videos count for nearly 80% of views, while the rest 90% of the videos receive very few requests. Besides the argument that recommendation tools and search engine favor popular items, another argument is that users tend to consume the most popular contents (the hits on home pages, the subjects of buzz, the most seen contents: winner takes it all [Fra95]). As for the evolution of the popularity of Youtube videos in time, the number of views 5, 7 and 90 days after the publication of the video is found to be highly correlated to its popularity after 2 days. So a video with little audience two days after its apparition on Youtube is likely to rest unpopular forever.

The prediction of audience of contents is also addressed in [SH08], while other authors identify different patterns of success (for photos on Flickr [CMG09]) or different types of contents based on the attention they are given ([CS08]).

The two aspects, uses and contents on online platforms, are often analyzed together. This seems logical as the content is published by users and made popular by their uses. When analyzing the popularity of contents one often analyzes the strategies developed by users to build the popularity, so the uses of the tools for social networking or promotion. Also, the analysis of uses often raises questions about the success of the published contents. For instance, in the previously evoked study on the uses of tags on Delicious [GH06], the authors analyze also the popularity of URLs measured as the number of bookmarks containing them. In another study Huberman et al. [HRW08] show that productivity on Youtube is dependent of the attention received by a content: a lack of attention leads to a decline of the activity.

3 Social networks. Many of the online platforms give users the possibility to link to other users by explicitly declaring a relation (such as friendship or fan etc.) or by leaving traces on other users' profiles (such as comments). These relations between users can be modeled by graphs. The analysis of such graphs is important as it can provide characterizations of the individuals and of the links between them. For instance, one could look for individuals that have similar positions in the graph because these individuals are likely to act similarly on the platform. Moreover one can characterize individuals by using endogenous variables of the social network and also exogenous variables (such as age, gender, town, quantity of published content, quantity of comments etc.) and then measure the correlations between the two types of variables. If the two are correlated, one can predict one using the others which can be very useful if some of the information is hidden.

If data on users of online platforms is generally rich and public (personal data, declared friendship relations, comments are public on sites like MySpace, Flickr, Twitter etc.), which is rarely the case for offline data, the analysis of the relations between the users of an online platform may be difficult. First, for most of the platforms one cannot analyze the whole set of relations because this set is too big and the recording of the relations too long. Therefore one usually builds a sample of the relations present on the platform by doing a breadth-first search aspiration (see Section 2.1 for a presentation of the BFS method) of profiles: starting from some initial profiles, one follows a given relation (such as friendship declaration) and goes from profile to profile, recording the found data. The BFS crawling produces a sample with a relevant structure (good fitting of the clustering, density, and centrality values) but underestimates the in-degree and overestimates the out-degree [MMG⁺07], [KNT06]. Second, data is often noisy: many profiles may not be active (their creators never go on the profile), a user may have several profiles, the relations may not correspond to real social relations between individuals (they are artificially and maybe randomly created), the nature of the relations may be completely hidden (some of them correspond to real strong social relations, some other do not have a correspondent in the real life, but all of them have the same form on the platform: they are all friendship relations for instance).

Once we have obtained a set of data that can be modeled by a graph, we have a social network which often is also a complex network, so all the discussion in Chapter 3 applies here. One can analyze the data at three levels: global, by characterizing the network as a whole, local, by evaluating the local structure of the network, around each node, or

intermediate, by studying communities, roles of nodes etc.

For the analysis of communities, one can compute such communities by using the social network as explained in Section 3.1, or study the user defined communities. On different platforms it is possible for users to join predefined groups, so one can try to explain how and why people join these groups. It can be because they share convictions or hobbies with the people already in the group or because they have strong relations with one or several persons in the group [Sas02, SP02].

In [BHL06] Backstrom et al. try to explain the decision of joining a group and the evolution of communities by analyzing the social network in which the individuals are embedded. They use decision trees to predict and explain the decision of an individual to join a community and also how much a community grows. The explaining variables for the decision to join a community are chosen from a large set of factors such as the number of friends an individual has in a given community, but also how these friends are connected: by an edge, by a path, the average length connecting two friends, number of community members reachable from the friends etc. For the growing of the community, the explaining variables are the number of members of the community, the number of individuals with a friend in the community, the number of triangles in the community and the number of 3-paths etc. The analyses are done on two datasets: LiveJournal (a site for maintaining journals, personal and group blogs) where users can create and join communities, and DBLP (an online database of computer science publications) where conferences are used as a proxy for communities. For the decision to join a community, the number of friends already in the community plays an important role but also the connectivity of these friends: a user is more likely to join a community if his friends already there are connected to each other. One possible explanation is based on the notion of social capital: the individual knows that he will be supported by a rich local social structure if he joins. For the growth of the community, the existence of a large number of people with friends in the community is the most important factor for a significant growth.

The global properties of online networks follow the general patterns of complex networks: the degree distribution is a power law or one of its variants, the clustering coefficient significantly higher than in random networks, the diameter is small and there exists a large connected component.

In [KNT06] Kumar et al. analyze several complex networks characteristics for two large samples of Flickr (a site for photo sharing and social networking) and Yahoo!360 (a site for Yahoo! users for sharing photos or blogs among the friends of a user). This study is one of the first studies that use temporal data i.e. all the activity on the two sites is recorded during several dozens of weeks. The authors are thus able to analyze the dynamic properties of the two platforms. They begin by observing that the two networks are highly "mutual" i.e. the friendship links are often reciprocal and, as expected, the degree distributions are power-laws. One interesting result is the evolution of the density that is discovered to have, in both networks, 3 stages (see Figure 4.3): first a rapid growth, generated by an initial euphoria among a few enthusiasts who join the network and frantically invite many of their friends to join, second a decline, generated by the natural dying-out of the euphoria and third, a true organic growth when more and more people know about the network.

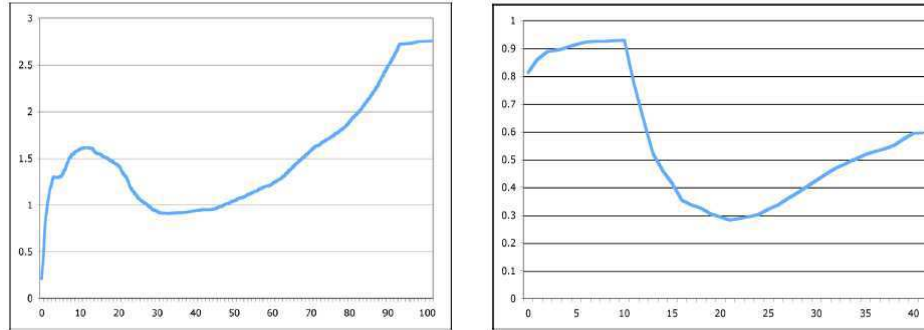


Figure 4.3: Density of Flickr and Yahoo! 360 by week [KNT06].

The authors continue their analysis by classifying the members of the two networks into one of the three groups: *singletons* (degree-zero nodes that have joined the service but have never made a connection with another user), *giant component* and *middle region*, consisting in various isolated connected components. This middle region contains about $1/3$ of the users of Flickr and about 10% of the users of Yahoo! 360. Surprisingly, these fractions remain almost constant in time, despite significant growth of the networks (for example the Flickr social network grew by a factor of over 13x during the studied period). Also, about 90% of the connected components in the middle region have a *star structure* i.e. connected components where one or two nodes (centers) have an edge to most of the other nodes in the component and a relatively large number of nodes have an edge solely to one of these centers. As for the structure of the giant component, $1/2$ of the nodes have degree one and there is a small core of highly connected vertices. The diameter is rather small but greater than 6 (suggested by the "six-degrees of separation" folklore): the average diameter is found to be 6.01 for Flickr and 8.26 Yahoo! 360, while the effective diameter is 7.61 and 10.47 respectively. The time evolution of the diameter is highly correlated to that of the density: it has a first stage of flatness, followed by a second stage where the edge density drops and the diameter grows till it reaches a peak, and a third stage, when the edge density starts increasing and the diameter starts decreasing. A similar phenomenon of shrinking diameter was observed by Leskovec et al [LKF05] in citation graphs. Finally, the authors propose a model of network evolution using a biased notion of preferential attachment. The model reproduces quite accurately the component structure of the two networks.

4.5 Online activities vs. offline communications

We compare here several characteristics of an offline social network and an online one. This section represents an original work; we have however placed it in the overview and survey part because it is here that we have presented the characteristics of online and offline networks.

The studied offline network comes from one month of mobile phone communications between 3 million persons. This dataset will be detailed in Chapter 7; it contains the communications of the clients of a same operator during a month. We model this set of communications by a simple undirected graph where the vertices are the clients and the edges are given by the presence of communications: we link two vertices by an edge if each one of the two persons called the other one at least once during the recorded month. We thus obtain a graph that has approximately 2 million vertices and 3 million edges.

The online network comes from the recordings of the activity of 1.6 million users of Flickr (www.flickr.com), a site for photo and video sharing and social networking, also during a month. We are still dealing with inter-personal communications, this time online comments instead of phone calls. On Flickr, users can put photos and videos online that the other users can see and comment. Any user can comment other users' photos or his own (we chose to filter out the comments to the own photos). As in the case of mobile phone, we model the activity of the users of Flickr by a simple undirected graph where the vertices are the users and the edges correspond to comments: two vertices are connected by an edge if each one of the two users commented at least once the other user's photos. The obtained graph has 63,000 vertices and 245,000 edges, so almost 32 times fewer vertices and 12 times fewer edges than the mobile phone network. There are several explanations for these differences.

A simple but important observation is that people interact in the two contexts (phone calls and online comments) in very different ways. First, a phone call is a synchronous communication: the two persons talk to each other, it is a live exchange. On the contrary, writing comments on Flickr is asynchronous: one just writes the comment, without necessarily waiting for an answer. Second, a mobile phone call requires some effort: the caller must have the phone number of the person he wants to call and usually he has to pay for the phone call. In contrast, writing a comment on the photo of another Flickr user is easy: one does not need a prior knowledge of the user and does not have to pay for writing comments. However, what really make the difference between mobile phone usage and writing of comments are the aim and the utility of a phone call as opposed to that of a Flickr comment: people call each other in order to synchronize, to coordinate, to give or receive news, to exchange information etc, while on Flickr, the comments are related to a photo or a video. The mobile phone is a very useful device, while writing comments may be fun, but hardly something people absolutely need in their every day life. Thus, during a day, a person is more likely to make a phone call than to write an online comment. Also a mobile phone communication indicates a certain relation between the two persons, simply by the fact that mobile phone numbers are not publicly listed. People do not call people they don't know just to comment on a certain thing.

Given these reasons, one expects, during a month, a smaller number of comments

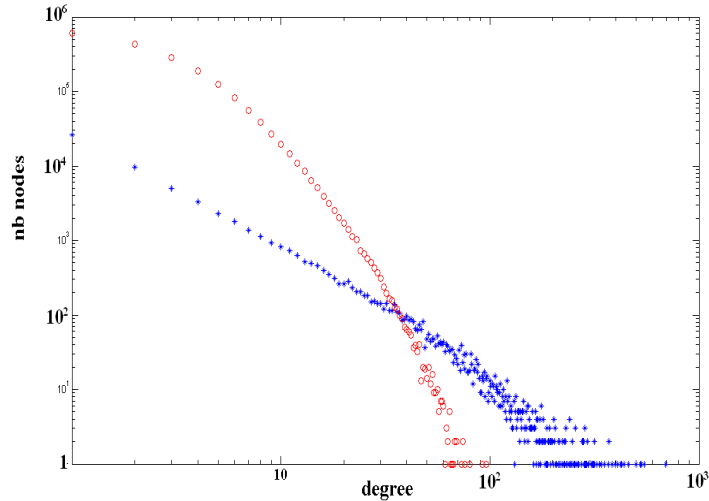


Figure 4.4: Double log scale degree distribution in the mobile phone network (red dots) and the Flickr network (blue dots).

than mobile phone calls between the same number of people. Indeed, during the followed month, the 1.6 million users of Flickr wrote approximately 4 million comments, while the 3 million mobile phone customers made approximately 150 million phone calls. All the mobile phone customers make at least one phone call and $2M$ out of the $3M$ make calls and also receive. On the contrary, most of Flickr users have an account, publish photos, but never make comments. Only 63,000 out of the $1.6M$ users give and receive comments. There are however some very active users who make a lot of comments and also receive a lot. Users' activity is much more heterogeneous on Flickr than in the mobile phone network.

The degree distributions of the two graphs are therefore completely different (see Table 4.1 and Figure 4.4): in the online network the maximal value of the degree is much higher than in the offline network, while the median is the same, so the majority of Flickr users make very few comments while there are some users that make a lot. This shows that online relations do not necessarily reflect offline, real social relations. In real life the cost of creation and keeping of relations limits their number at a certain threshold, thus introducing a cut-off in the distribution. Moreover, on online platforms, everybody is visible. A user can connect to any other user simply because all the needed contact information are on the profile. In a favorable context (great audience of the published contents, promotion etc.), some users become very popular. They become the stars of the platform, having a great number of contacts i.e. a high degree ¹. This notion of star is not present in offline contexts: one does not get phone calls from other persons just because he is popular.

To end this parenthesis on the differences between the datasets of Flickr and mobile

¹The online popularity of another online platform, MySpace, is analyzed in Chapter 6

Table 4.1: The number of nodes and links of the two networks and their average, maximal and median degree

network	# nodes	# links	avg degree	max degree	med degree
Flickr	63×10^3	245×10^3	7.8	695	2
mobile phone	2×10^6	3×10^6	3.3	96	2

phone communications, note that we defined the edges of the two graphs in the same way, by using the existence of communications. Generally one uses for online platforms the declared links (e.g. friendship links) as edges. However, many declared links are not active: the users do not contact each other, they have declared each other as friends but they haven't had any contact since. Taking into consideration only the links sustained by a certain activity allows us to filter out these cases of unused links.

4.6 Applications: Marketing and services

For a products or services provider, the knowledge of its customers is essential in order to target the audience, to propose services and publicity adapted to each user etc. To characterize the customers, several dimensions can be taken into consideration: the different socio-demographic information (such as age, gender, job, residence etc.), the uses customers make of the different services, and the social network in which they are embedded. The social network dimension is important because people do not live isolated lives, they are surrounded by other people who might influence them.

If marketing models take or not into consideration the social network aspect (although it is frequently shown that different parameters computed in the social network improve marketing models), there is one field of marketing that studies this dimension: the viral marketing. This field exploits existing social networks by encouraging customers to share product information with their friends. The motivation is that individuals are influenced by their personal relations in the decision of adopting innovations and products (this is also known as the Word-of-mouth, WOM, influence). Several researches brought this to light. For instance, sociological studies on individual choice, initiated in the 1940s' by P. Lazarsfeld team at *Bureau of Applied Social Research* at Columbia University, emphasized the influence of the network of personal relations in the decision of purchase. Engel et al. [EBK69] find that 60% of the persons asked about the choice of a car garage cited the WOM as main influence. Also, Feldman and Spencer [FS65] estimate to two-thirds the ratio of new residents of a community who used WOM for finding a doctor. Even study institutes as Harris Interactive [AD07] or BIGresearch tried to measure the importance of WOM. If the former provided a ranking of products depending on the degree of influence of WOM in the decision of consumption, the latter state that more than 90% of the interviewed persons give or receive purchase advice.

In the context of the internet, word-of-mouth advertising is not restricted to pairwise or small-group interactions between individuals. Rather, customers can share their experiences and opinions regarding a product with everyone. Quantitative marketing techniques

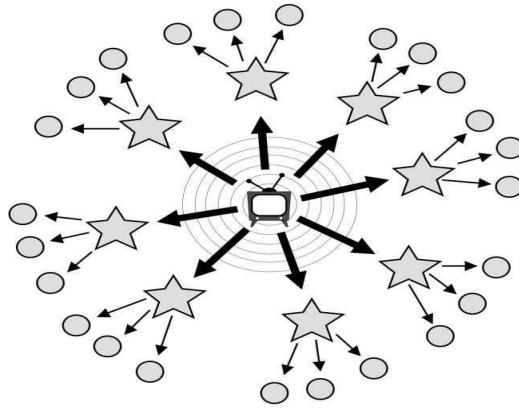


Figure 4.5: Schematic of the two-step flow model of influence [KL55].

have been proposed [Mon01] to describe product information flow online, and the rating of products and merchants has been shown to effect the likelihood of an item being bought [RZ02, CM06].

If the influence of the social network on making a decision (at least that of adopting a product or a service) is generally accepted, the existence of a group of people capable of having a greater influence than other people is still debated. Several researchers tried to identify persons with a certain position, and therefore influence, in a social network. Such people, often called "social leaders" or "influentials", would be capable to influent other people or to speed up the process of diffusion of products and services. In their *two-step flow model*, Katz and Lazarsfeld [KL55] propose the idea that there exists a small fraction of opinion leaders (stars in Figure 4.5) who act as intermediaries between the mass media and the majority of society (circles). Their influence is direct and derives from their status as individuals who are highly informed, respected, or simply "connected"; these people are capable of influencing an exceptional number of their peers. Gladwell [Gla00] sustains the concept of influentials adapted to marketing: if it is possible to find and target the influentials in a social network, then the diffusion will be extremely fast, while randomly chosen individuals will cause a slow diffusion. This hypothesis is however contradicted in [WD07]. Using a series of computer simulations of interpersonal influence processes, the authors argue that cascades of adoption do not succeed because of a few highly influential individuals influencing everyone else, but rather on account of a critical mass of easily influenced individuals influencing other easy-to-influence people. In their models, influentials have a greater than average chance of triggering this critical mass, when it exists, but only modestly greater, and usually not even proportional to the number of people they influence directly.

Part II

Methods and Applications

Chapter 5

A method for analyzing the local structure of large networks

In this chapter we present a method for analyzing the local structure of a (possibly large) network by characterizing the way each node is connected to the network. The method is designed to be applied to a given node of a network; in this case it produces a characterization of the configuration of the network surrounding the node: the structures in which the node it is embedded, the way its neighbors are placed with respect to the others and the way its links are disposed. One can apply this method to all the nodes of the network, thus obtaining a description of its local structure, or only to some of its nodes: it can be useful if one has only a fraction of the nodes of the network or if the goal is to compare some nodes to each other. Before presenting the method, we introduce some useful notions. Then we explain the method and we compare the measures produced by it to other existing indicators. We finish this chapter by making some comments on the usefulness of the method.

5.1 Definitions

Unless specified otherwise, all the considered graphs are simple and undirected.

Egocentred network. Given a graph $G = (V, E)$ and a vertex $v \in V$, we call *egocentred network* of v , denoted by $Eg(v)$, the subgraph induced in G by the neighbors of v i.e. the graph whose vertices are the neighbors of v and whose edges are the edges between these neighbors.

Patterns and positions. We call *k-patterns* all the non-isomorphic connected graphs with at most k vertices and at least 1 edge. Figure 5.1 presents the thirty 5-patterns. There are nine 4-patterns (indices 1 to 9) and three 3-patterns (indices 1 to 3). In this chapter we consider only 5-patterns that we call simply *patterns*.

Given a graph, two vertices are said to be *position equivalent* if there is an adjacency preserving permutation of the vertices of the graph such that the two vertices are interchanged (the position equivalence is actually the automorphic equivalence). A *position* is a maximal set of position equivalent vertices. For example, for each pattern in Figure 5.1,

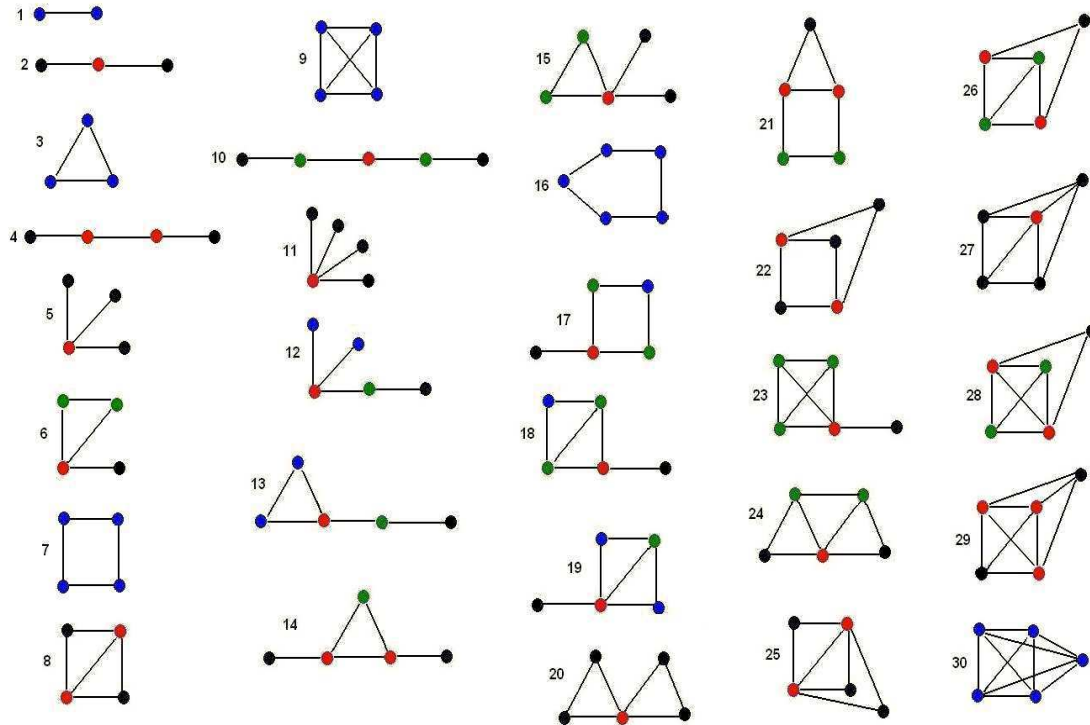


Figure 5.1: The set of patterns and their positions.

each color corresponds to a distinct position. Formally, two vertices u and v of a graph G are position equivalent if there exists an automorphism φ of G such that $\varphi(u) = v$. The positions correspond to the equivalence classes of this relation. There are 73 different positions in the 30 patterns and, as Figure 5.1 shows, a pattern has at most 4 different positions. We want to establish categories of positions so we sort the positions of a same pattern in ascending order of their betweenness centrality; for different positions having the same centrality, we sort in ascending order of the degree. We call *peripheral* the first position in this order and *central* the last one. The positions that are not central nor peripheral or are both central and peripheral are called *intermediate*. Briefly the positions colored in red are central, those colored in black are peripheral and the other ones are intermediate.

Graph characterization. Given a graph $G = (V, E)$, one can obtain a characterization of the graph by computing the occurrences of the different patterns in the graph, and of its vertices by computing the position each vertex occupies in each pattern. A pattern P is said to occur in the graph G if there exists a set of vertices $V_P \subseteq V$ such that the subgraph induced by V_P in G is isomorphic to P . Listing all the occurrences of the pattern P in the graph G means finding all the sets of vertices V_P according to the previous definition. For each occurrence of a pattern in $G = (V, E)$ one can compute in which position of the pattern the different vertices of V are placed. Thus, after having listed all the occurrences of the 30 patterns in G , one has, for each vertex $v \in V$, its number of occurrences in each one of the 73 positions (we call this the *position vector* of v). Formally, the k -*position vector* of v is a vector $Pos_k(G, v)$ that contains the number of occurrences of v in the different positions of the k -patterns: $Pos_k(G, v, i)$ counts the number of subgraphs of G with at most k vertices that contain v in the position i . As an example, Figure 5.2 represents a graph (a), the patterns it contains and their number of occurrences (b), and the number of occurrences in the different positions of two selected vertices (c) (we have noted only the positions where at least one of the two vertices is present; for all the other positions the corresponding element of the position vector is 0).

5.2 Efficient graph characterization

When characterizing a graph G as explained before, one needs to search all the induced subgraphs with a given maximal number of vertices (in our case 5), to find to which pattern each of them is isomorphic and to compute the number of occurrences of the different vertices in the different positions. All the three operations (the listing of patterns, the checking of isomorphism and the computation of positions) must be done efficiently so that one can characterize a large number of graphs in a reasonable time.

For the listing of subgraphs we use Algorithm *ESU* introduced in [Wer06]. Figure 5.3 presents this algorithm; $N_{excl}(w, V_{subgraphs})$ (line E_4) represents the set of neighbors of w which do not belong to $V_{subgraphs}$ nor have any neighbors in $V_{subgraphs}$. Basically the algorithm starts with a vertex v of G and adds neighboring vertices until a set of k vertices is obtained, hence a connected induced subgraph with k vertices. More precisely, starting with the vertex v , the algorithm repeatedly adds neighbors of v or of the already added

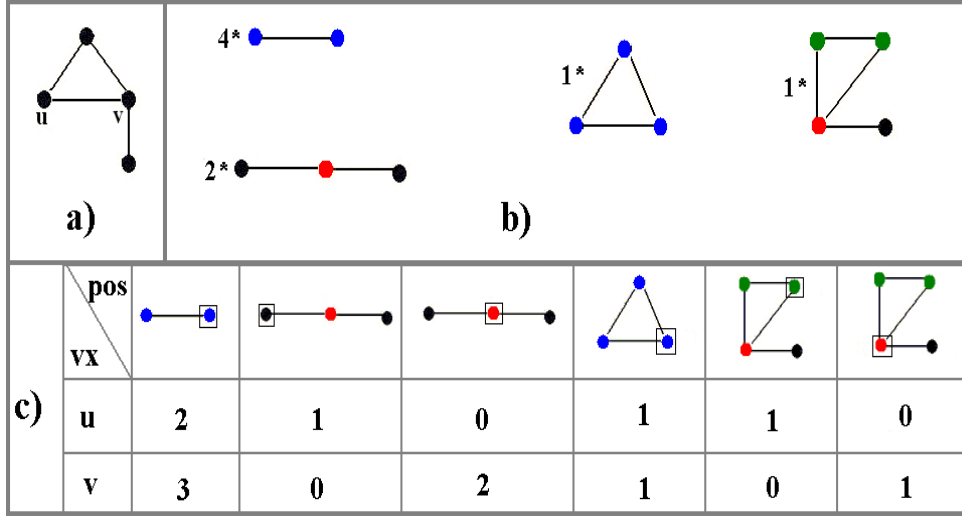


Figure 5.2: A graph (a), its patterns (b) and the position vectors of two vertices u and v (only the positions where at least one of the two vertices appears) (c).

vertices (this is the set $V_{extension}$). It is the computation of the set $V_{extension}$ that makes this algorithm efficient. To be added to this set, a vertex must satisfy two conditions: its label must be greater than that of v (the labels are simply indices from 1 to $|V_G|$) and it must have exactly one neighbor in the already added vertices. This insures the addition of each vertex exactly once. Also, as explained in [Wer06], the algorithm finds each subgraph exactly once, so one does not need to check the presence of a found subgraph in a list of already found subgraphs. To our knowledge, this is the most efficient existing algorithm for induced subgraphs listing.

Once an induced subgraph has been found, one needs to find the pattern to which it is isomorphic. For several patterns this can be done by computing the degree distribution of their vertices: patterns with different degree distributions are not isomorphic. The reverse, however, is not always true. For instance, patterns number 21 and 22 in Figure 5.1 have the same degree distributions: $(2, 2, 2, 3, 3)$. In this case one can differentiate between the two patterns by looking not only at the degrees of the vertices, but also at how vertices of different degrees are inter-connected. Thus, for pattern 21, two vertices of degree 2 are connected to each other, while the vertices of degree 2 in pattern 22 are connected only to vertices of degree 3. To take into consideration in the same time the degrees of the vertices and of their neighbors we introduce the notion of neighbor-degree.

Definition 5.2.1. *Given a graph G and a vertex v of G , we call neighbor-degree of v , denoted by $nd(v) = \sum_{u \in N[v]} d(u)$, the sum of its degree and the degrees of its neighbors. We call degree combination of the graph G the ascending sorted list of the neighbor-degrees of its vertices.*

These two notions suffice in order to check if two connected graphs with at most 5 vertices are isomorphic, as shown by the following lemma.

```

Algorithm: ENUMERATESUBGRAPHS( $G, k$ ) (ESU)
Input: A graph  $G = (V, E)$  and an integer  $1 \leq k \leq |V|$ .
Output: All size- $k$  subgraphs in  $G$ .

01 for each vertex  $v \in V$  do
02    $V_{Extension} \leftarrow \{u \in N(\{v\}) : u > v\}$ 
03   call EXTENDSUBGRAPH( $\{v\}, V_{Extension}, v$ )
04 return

EXTENDSUBGRAPH( $V_{Subgraph}, V_{Extension}, v$ )
E1 if  $|V_{Subgraph}| = k$  then output  $G[V_{Subgraph}]$  and return
E2 while  $V_{Extension} \neq \emptyset$  do
E3   Remove an arbitrarily chosen vertex  $w$  from  $V_{Extension}$ 
E4    $V'_{Extension} \leftarrow V_{Extension} \cup \{u \in N_{excl}(w, V_{Subgraph}) : u > v\}$ 
E5   call EXTENDSUBGRAPH( $V_{Subgraph} \cup \{w\}, V'_{Extension}, v$ )
E6 return

```

Figure 5.3: Pseudocode for the algorithm *ESU* which enumerates all size- k subgraphs in a given graph G [Wer06].

Lemma 5.2.2. *Two graphs G and H with at most 5 vertices are isomorphic if and only if their degree combination are identical. Moreover, two vertices $u, v \in V_G$ are position equivalent if and only if they have the same neighbor-degree.*

Proof. The proof is straightforward, it suffices to check the two statements for all the connected graphs with at most 5 vertices. \square

For the two patterns in our previous example, the degree combination of pattern 21 is (7, 7, 8, 10, 10), while that of pattern 22 is (8, 8, 8, 9, 9). Thus, the two patterns are identified as non-isomorphic. Moreover vertices of a same pattern that have distinct positions have different neighbor-degrees.

Note that for a graph G with n vertices and m edges one computes the neighbor-degrees of all the vertices of G in $O(m)$ time and $O(n)$ space (it suffices to scan all the edges in order to compute and store the degrees, then scan all the edges again to compute the neighbor-degrees), then its degree combination in $O(n \cdot \log n)$ time. For the set of patterns these quantities are constant as n and m are at most 5 and 10 respectively. Therefore one can find to which pattern a connected graph with at most 5 vertices corresponds (i.e. to which of the 30 graphs in Figure 5.1 it is isomorphic) and check if two of its vertices are position equivalent in constant time.

Note however that the lemma is not true for the connected graphs with 6 vertices. The two graphs in Figure 5.4 are not isomorphic but have the same degree combination: (7, 7, 7, 7, 10, 10).



Figure 5.4: Two non-isomorphic connected graphs with 6 vertices

5.3 A method for local structure analysis

Given a (possibly large) graph $G = (V, E)$, we want to analyze its local structure around a vertex $v \in V$ (we call this vertex *ego*). We proceed as follows – method *local_structure(v)*:

- Step 1.** Extract the egocentred network $Eg(v)$ of v i.e. the subgraph induced by the neighbors of v in G ;
- Step 2.** List the patterns of $Eg(v)$;
- Step 3.** Compute the position vectors of the vertices in $Eg(v)$.

Let us explain the three steps of the method with an example.

Step 1 and 2. In Figure 5.5(a), the black circles correspond to the neighbors of v , the black lines correspond to the edges between them and the red lines to the edges between v and its neighbors. The egocentred network $Eg(v)$ of v is represented in Figure 5.5(b) and the patterns of $Eg(v)$ in Figure 5.5(c)¹. We chose not to include v in its egocentred network because we know that it is connected to all the vertices in this graph, its presence does not bring any information. After performing the steps 1 and 2 of the method one has a rich description of the way v is connected to the graph G . For a more detailed description of the local structure of G around v one can list the patterns of a higher order (with more than 5 vertices); the patterns with 5 vertices are however a good compromise between the variety of forms and their number; even the 4-patterns provide in many cases a detailed enough picture.

Step 3. We compute the position vectors of the neighbors of v , so the number of times each neighbor appears in each one of the positions of the different patterns. Figure 5.5(d) contains the position-vectors of two neighbors of v (only the elements that are higher than 0 for at least one of the vertices; all the other elements are equal to 0). The positions occupied by the different neighbors describe the relative place of these neighbors as opposed to the other neighbors but also the links formed by v , if one looks from v 's point of view. As an example, Figure 5.6 presents the correspondence between three possible positions of a neighbor u and the structure of the graph around the edge (u, v) .

¹We have also counted the isolated vertices and edges in $Eg(v)$.

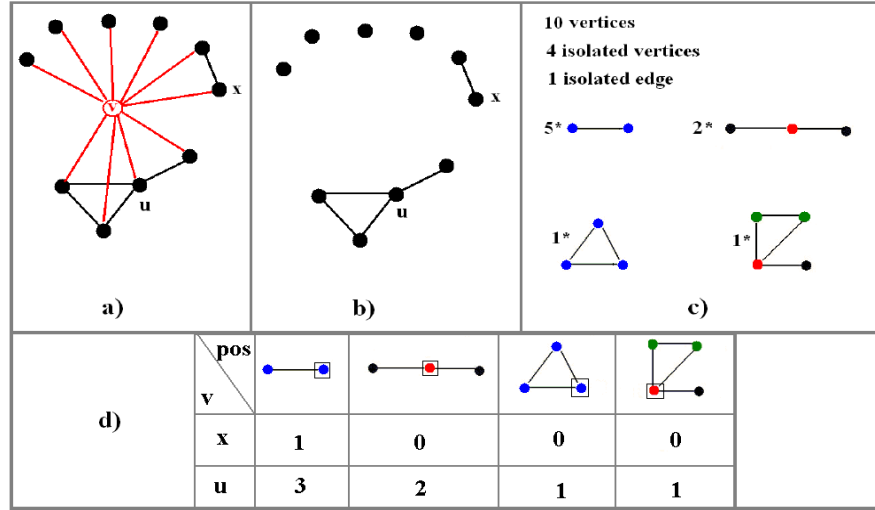


Figure 5.5: A vertex v and its neighbors (a), the egocentred network $Eg(v)$ of v (b), the patterns of $Eg(v)$ (c) and the position vectors of two neighbors of v (d) (only the positions where at least one of the two vertices appears).

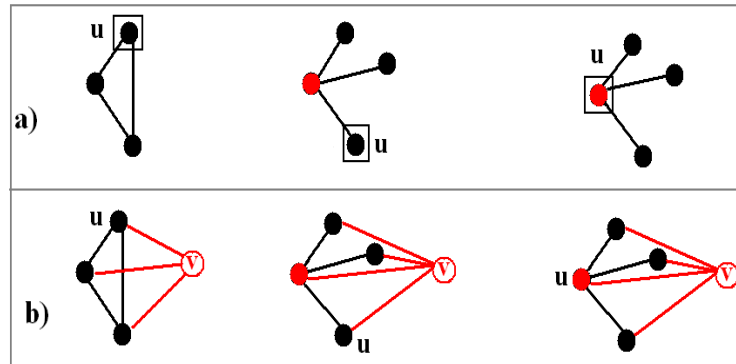


Figure 5.6: Three possible positions of the neighbor u (a) and the corresponding structures around the edge (u, v) (b).

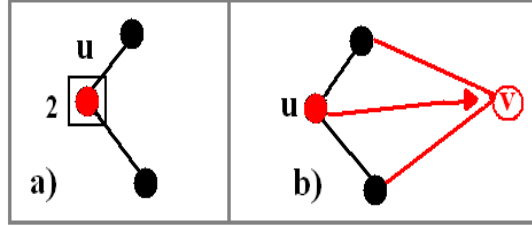


Figure 5.7: A position of the neighbor u with weight 2 (a) and the corresponding structure around the edge (u, v) (b).

If the graph G is directed, one can add this information to the description of the edges formed by v by simply adding a weight to the neighbors of v . For a node v , the weight $w_v(u)$ of a neighbor u is:

- 1 if the connection is from v to u ($v \rightarrow u$),
- 2 if the connection is from u to v ($u \rightarrow v$),
- 3 if the connection is symmetric ($v \rightarrow u$ and $u \rightarrow v$).

As an example, Figure 5.7 presents the correspondence between a possible position of a neighbor u that has weight 2 and the structure of the graph around the edge (u, v) .

The method introduced here can be used to define a relation of equivalence on the vertices of the graph G . First, each vertex can be characterized by a vector containing the number of occurrences of patterns with at most k vertices in its egocentred network. Then, one can use these vectors to identify equivalent vertices.

Definition 5.3.1. *Given a vertex v of a graph G and a positive integer k , we call k -pattern vector of v the vector containing the number of occurrences of the k -patterns (i.e. all the non-isomorphic connected graphs with at most k vertices) in the egocentred network $Eg(v)$ of v . Two vertices of the graph G are said to be k -pattern equivalent if and only if they have identical k -pattern vectors.*

5.4 Algorithmic aspects

Remember that the graph $G = (V, E)$ to which the method is applied may be large (more than 10^5 vertices and even more edges). Therefore we have to pay a particular attention at the time and space complexity of the used algorithms. First of all, we store the graph G in a adjacency list representation (see Section 2.1): for each vertex, we have the ascending sorted list of its neighbors (the vertices of V are given indices from 0 to $|V| - 1$). This representation needs $\Theta(|E|)$ space and running through $N(v)$ takes $\Theta(d(v))$ time, with $d(v)$ denoting the degree of v . Testing the presence of an edge (u, v) takes $O(\log(d(v)))$

time. For a graph $G = (V, E)$, let n denote the number of its vertices ($n = |V|$) and m the number of its edges ($m = |E|$).

Step 1. In this step we need to compute the egocentred network of the vertex $v \in V$ i.e. the subgraph induced by the neighbors of v in G . This is equivalent to listing the triangles in which v appears. For this, we rely on Algorithm *new-vertex-listing* proposed in [Lat08]. Algorithm *ComputeEgocentered* computes the egocentered network of the vertex $v \in V$.

Algorithm 1 *ComputeEgocentered.* *Computes the egocentered network of a vertex*

Input: A simple undirected graph $G = (V, E)$ and a vertex $v \in V$

Output: A simple undirected graph $Eg = (V_v, E_v)$, the egocentred network of v

1. create an array A of $|V|$ integers and set them to -1
 2. initialize V_v and E_v to the empty set
 3. for each vertex u in $N(v)$, set $A[u]$ to v
 4. for each vertex u in $N(v)$
 - 4.1 add u to V_v
 - 4.2 for each vertex w in $N(u)$ such that $w < u$
 - if $A[w] = v$ then add (w, u) to E_v
-

Algorithm *ComputeEgocentered.* One may see this algorithm as a way to use the adjacency matrix of G without explicitly storing it: when processing the vertex v , the array A is nothing but the $v - th$ line of the adjacency matrix. This array is built in $\Theta(n)$ time and space. Then one can test for any edge (u, v) in $\Theta(1)$ time and space. Since the line 4.2 is executed at most twice for each edge connecting a neighbor of v , and there are at most m such edges, we obtain that Algorithm *ComputeEgocentered* is in $O(m)$ time and $\Theta(n)$ space.

Steps 2 and 3. We want to characterize the graph $Eg(v)$, so to compute its patterns and the positions of its vertices. For simplicity of notation and because these two steps constitute a method that can be applied to any graph, not just to egocentred networks, we denote the graph $Eg(v)$ by G . First, we need to identify the connected induced subgraphs with at most 5 vertices of G , then to find the pattern to which each of these subgraphs is isomorphic and finally to compute the positions occupied by the different vertices in the found subgraphs (actually the three operations are successive: once a subgraph is found, one checks to which pattern it is isomorphic and computes the positions of the vertices, then continues the search for other subgraphs). For the first part we rely on Algorithm *ESU*(G, k) [Wer06] (see Figure 5.3) that lists the induced subgraphs of G with k vertices. For the second and the third part, we compute the neighbor-degrees and the degree combination of the found subgraph, according to Lemma 5.2.2. Algorithm *CharacterizeWithPatterns* implements the two steps.

Algorithm *CharacterizeWithPatterns.* We have slightly modified Algorithm *ESU* (Figure 5.3) in order to compute induced subgraphs with *at most* k vertices with $k \leq 5$. Also, the operation *output* $G[V_{Subgraph}]$ (line E_1 in *ESU*) is replaced by the

function *IndexPattern* that computes the pattern isomorphic with the found subgraph and the positions occupied by the different vertices. Algorithm *CharacterizeWithPatterns* has a time complexity linear in the number of patterns found in the graph G : for Algorithm *ESU* see [Wer06]; for Function *IndexPattern* note that it takes $O(m_p + n_p \times \log n_p + \log \text{nb_patterns})$ to execute, where n_p is the number of vertices in the pattern (at most 5), m_p is the number of edges (at most 10) and nb_patterns is the total number of different patterns (equal to 30 for patterns with at most 5 vertices). As all these quantities are smaller than given constants, 5, 10 and $\log 30$ respectively, one can say that *IndexPattern* has a constant time complexity and Algorithm *CharacterizeWithPatterns* is linear in the number of patterns of the graph G . As we do not dispose of a method for estimating the number of patterns of a given graph, let us note simply that the number of patterns with at most k vertices is at most n^k where n is the number of vertices of G .

Algorithm *CharacterizeLocalStructure*. We have now all the elements for writing the algorithm that characterizes the local structure of a graph $G = (V, E)$ around *each vertex* $v \in V$: Algorithm *CharacterizeLocalStructure*. This is simply the application of the two previous algorithms to all the vertices of the graph. Note however a modification: the array A is built only once for all the vertices of the graph, at the beginning of the algorithm, and then updated for each vertex. Thus the construction of A has the same time and space complexity as in Algorithm *ComputeEgocentred*: $\Theta(n)$ for both. The time complexity of Algorithm *CharacterizeLocalStructure* is thus $\Theta(n + \sum_{v \in V} (\text{nb. patterns in } Eg(v)))$ which is (at most) $O(n + \sum_{v \in V} (d(v)^5))$. As we apply this method to real-world complex networks, where most vertices have small degrees, the method is in average rather fast. In Chapter 7 we apply the method to a real-world graph with $2.7M$ vertices and $6.4M$ edges and we give an empirical complexity of our method for this graph. It takes 31 minutes for our C++ implementation of the method to execute for this graph on a computer with standard configuration, a 2.8GHz processor and 4Gb RAM.

5.5 Applications of the method

The goal of the method we introduced here is to characterize the way a node is connected to the network. It is a method for analyzing the local structure of the network that produces a characterization of each node. Its goal is not to give a ranking or ordering of nodes but merely to show how they are connected to the network. This can be useful in several situations. First, as any characterization method, it improves our knowledge of the nodes of the network. Second, the obtained characterization of nodes can be compared to other properties of the nodes: if there is a correlation, one can use one to predict the others. This is practical if some data is missing as some properties can be inferred from the other ones. Third, there are situations where a local analysis is the best way to study the problem. It is the case of data obtained independently for different persons, where the "global" network containing all the persons is unknown (as for instance in sociological studies where data on each person is obtained through individual interviews and there is no collection of the whole network). In this case one may want to study the network in which individuals are embedded, but, as there is no global network, one cannot perform

Algorithm 2 CharacterizeWithPatterns. *Characterizes an undirected simple graph*

Input: A simple undirected graph $G = (V, E)$ and a positive integer $k \leq 5$

Output: An array Pt such that $Pt[P]$ contains the nb. of occurrences of the pattern P in G , an array Ps such that $Ps[v][i] = Pos_k(G, v, i)$ (the nb. of occurrences of v in the position i)

1. set all the elements of Pt and Ps to 0
2. for each vertex $v \in V$ do
 - 2.1 $V_{extension} \leftarrow \{u \in N(v) : u > v\}$
 - 2.2 $V_{Subgraph} = \{v\}, E_{Subgraph} = \emptyset$
 - 2.3 call $ExtendSubgraph(V_{Subgraph}, E_{Subgraph}, V_{extension}, v, Pt, Ps, k)$
3. return

ExtendSubgraph

Input:

- a positive integer $k \leq 5$,
- two sets $V_{Subgraph} \subseteq V$ and $E_{Subgraph} \subseteq E$ containing the vertices and edges already added to the subgraph,
- a set of vertices $V_{extension}$ containing the vertices that can be added to the subgraph,
- a vertex v from which the construction of the subgraph has begun,
- two arrays Pt and Ps that will be updated by the procedure

1. if $|V_{Subgraph}| > k$ return
2. if $|V_{Subgraph}| > 0$ call $IndexPattern(V_{Subgraph}, E_{Subgraph}, Pt, Ps)$
3. while $V_{extension} \neq \emptyset$
 - 3.1. remove an arbitrarily chosen vertex w from $V_{extension}$
 - 3.2. $V'_{extension} = V_{extension}$
 - 3.3. $E'_{Subgraph} = E_{Subgraph}$
 - 3.4. for each $u \in N(w) : u > v$
 - if $u \in V_{Subgraph}$ add (u, w) to $E'_{Subgraph}$ //add all the edges from w to the subgraph
 - else if $u \notin N(V_{Subgraph})$ add u to $V'_{extension}$
 - 3.5. call $ExtendSubgraph(V_{Subgraph} \cup \{w\}, E'_{Subgraph}, V'_{extension}, v, Pt, Ps, k)$

IndexPattern

Input: A set of vertices $V_{Subgraph}$, a set of edges $E_{Subgraph}$ and two arrays Pt and Ps that will be updated by the procedure

1. scan the set $E_{Subgraph}$ and note each occurrence of each vertex
//thus computing the degrees of the vertices
 2. create an array D containing the degrees of the vertices
 3. for each edge $(a, b) \in E_{Subgraph}$ add $degree(b)$ to $D(a)$ and $degree(a)$ to $D(b)$
//thus computing the neighbor-degrees
 4. sort D and write it as a number
 5. find the pattern P with this number and increment $Pt(P)$
 6. for each vertex u
 - find the position i (in the pattern P) with the same neighbor-degree and increment $Ps[u][i]$
-

Algorithm 3 *CharacterizeLocalStructure* Characterizes the local structure around each vertex in a (large) graph

Input: A simple undirected graph $G = (V, E)$ and a positive integer $k \leq 5$

1. create an array A of $|V|$ integers and set them to -1
 2. for each vertex $v \in V$
 - 2.1 initialize V_v and E_v to the empty set
 - 2.2 for each vertex u in $N(v)$, set $A[u]$ to v
 - 2.3 for each vertex u in $N(v)$
 - 2.3.1 add u to V_v
 - 2.3.2 for each vertex w in $N(u)$ such that $w < u$
 - if $A[w] = v$ then add (w, u) to E_v
 - 2.4 call $\text{CharacterizeWithPatterns}((V_v, E_v), k)$
-

the classical global or intermediate network analysis.

Another situation where the study of the local structure is appropriate is for networks where nodes "importance" is local. In the opposite situation, there are networks where (some) nodes are important for the function of the whole network. Take for instance the case of the railways network of a country; in this case it is important to analyze nodes in the context of the global network: there are some nodes (railways stations) that are important for the whole network as they connect different parts of the country. In this case a local analysis is not sufficient, one needs to use measures that take into consideration the whole network. Also, in online social networks, the global perspective may be useful. In this case users are visible to the whole network: they can be seen and contacted by any other user in the network. Often there is a notion of popularity, where people try to improve their visibility and where fans can link to them. However, a local analysis may also bring important information. One can analyze for instance the links created by different persons before a certain moment in time; this is a local analysis that outputs star-fan relations (expressed by links).

A local approach is useful especially in networks where nodes importance and visibility are local. Take for instance the case of mobile phone communications. Here people cannot be contacted by everybody as mobile phone numbers are not public. And even if that was the case, people do not usually call other people just because these are known or famous. There is no measure of popularity in this network (as opposed to online platforms where different statistics on people activity and popularity are often available). People usually make phone calls because they really have something to discuss with the other person and not because they are fans of this person. In this case people a few steps away (maybe 2 suffice) from a person do not know this person; the existence of this person does not have any importance to them. For such networks characterizing nodes by looking at the whole network may not be very useful: someone with a high (say betweenness) centrality may not be more important than other persons. His presence in the network is surely important for several persons but these persons are most probably close to him in the

Table 5.1: Equivalent notions for a vertex v : in the whole graph and in the egocentred network.

graph G	egocentred network $Eg(v)$
degree of v	number of vertices
number of triangles containing v	number of edges
number of 4-cliques containing v	number of triangles

network. If this person leaves the network the vast majority of the other individuals in the network won't even notice the change. For such networks the method introduced here is more appropriate than other types of analysis taking into consideration the whole network (at least when characterizing each node).

Finally, this method can be used in order to compute a certain equivalence or similarity of vertices, notions very important for the definition of social roles played by nodes in a network. A possible relation of equivalence is the k -pattern equivalence that we have defined in Section 5.3. If one wants to compute similar vertices (instead of equivalent), one can compute a certain distance between the k -pattern vectors of the vertices (also defined in Section 5.3). We will discuss this approach and some applications in Chapter 8.

5.6 Comparison to other measures

Let us first emphasize the equivalence between several notions regarding a vertex v , in the context of the whole graph and in its egocentred network (see Table 5.1). For instance, the degree of v in the graph G corresponds to the number of vertices in the egocentred graph $Eg(v)$. Moreover the clustering coefficient of the node v is equal to the density of its egocentred network, as the number of triangles containing the node is equal to the number of edges between its neighbors, and are both equal to $\binom{d}{2}$ where d is the degree of v .

Patterns versus centrality. As presented in Section 3.1, the centrality of vertices is a measure of their importance in the network. Usually one computes the centrality of all the vertices in the graph in order to produce a ranking of vertices. There are several definitions of centrality: the degree centrality, the betweenness, the closeness, the page-rank, the eigen vector centrality etc. Besides the degree centrality (which is simply the degree of the node), all the other measures take into consideration the entire graph. As explained in the previous section, the goal of the method introduced here is to produce a local characterization of vertices. This is the main difference between our method and the different definitions of centrality: the goal is not the same. Another difference comes from the context of application of the methods: while the different measures of centrality need to have the entire network in order to compute the centrality of one node, our method needs only the neighbors of the node and the edges between them, so it can be applied only to some parts of the graph if the other parts are not known. Finally, the betweenness and closeness centrality can be hardly computed in complex networks as their time complexity

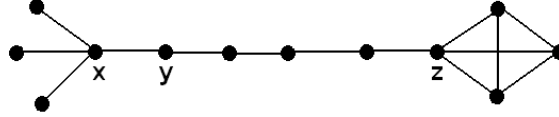


Figure 5.8: An example for the difference between centrality and position vectors.

is $O(nm)$. On the contrary, as explained earlier, our method can be easily applied to large networks.

In a different approach, one could compute the centrality of the vertices present in each egocentred network, so of the neighbors of each vertex, and compare the centralities of the different neighbors to each other. Remember that in our method we compute the k -position vector of each neighbor in order to see how the different neighbors are placed in relation with each other. The position vector is a different measure than the centrality. It reflects the relation of each one of the neighbors with the other neighbors, placed at at most 5 steps from it. It is rather a measure of how the different neighbors are placed and connected in the network than of their rank or importance. Look for instance at the graph in Figure 5.8 and suppose this is the egocentred network of some given vertex. The vertices x and z have degree 4, the vertex y has degree 2, and the betweenness centrality of x, y and z is 27, 28 and 24 respectively. While one has a ranking of the vertices (y is more central than x and x is more central than z), one does not know how these vertices are connected to the network. Even more, one can argue that it is x and not y that has a more important position in the egocentred network as it connects 4 vertices not directly linked. This is not shown by the degree nor by the betweenness centrality. By applying the method we introduced here one knows that x is the center of a star with 5 vertices and that it belongs to a path with at least 6 vertices. It is also clear that y is connected by a link to the center of a star and that it is in the center of a path. As for z , one knows that it belongs to a 4-clique and that it belongs to a path with at least 6 vertices. To sum up, the method we introduced here and the measures of centrality have different goals and are useful in different situations.

Patterns versus density and clustering coefficient. The density of the egocentred network of a vertex (or its clustering coefficient) is a first characterization of the vertex and the way it is connected to the network. For a more detailed characterization one can compute also the clustering coefficient of the egocentred network as the average of the clustering coefficient of the vertices in the egocentred network. The listing of patterns in the egocentred networks provides however a richer description of the local structure of the network than these two measures. Once again, it describes *how* the different neighbors of the vertex are disposed, in which type of structures they are embedded. For instance, imagine that the two networks in Figure 5.9 are the egocentred networks of two given vertices. These egocentred networks have the same number of vertices, of edges (so the same density) and the same clustering coefficient. These measures do not capture the

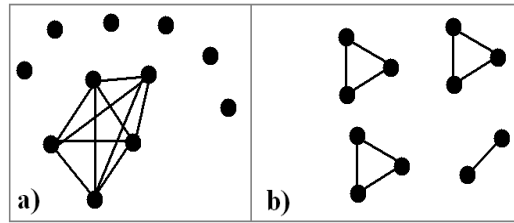


Figure 5.9: Two egocentred networks that have the same number of vertices, of edges and the same clustering coefficient.

differences between these two graphs, but the listing of patterns does.

K-pattern equivalence versus other vertex equivalences. In Section 4.1 we presented the structural, automorphic and regular equivalences, probably the most famous vertex equivalences. These notions, used in order to define social roles, are much too strict for real-world complex networks. The k-pattern equivalence that we defined in Section 5.3 is included in the structural and automorphic equivalence. This is based on the simple observations that vertices that have exactly the same neighbors in the network (so are structurally equivalent) have identical egocentred network, so identical feature vectors, and therefore are k-pattern equivalent, for all k . Also, vertices that are automorphically equivalent have isomorphic egocentred networks, so identical feature vectors and are thus k-pattern equivalent, for all k . For the two definitions, the opposite is not always true, so one can say that the k-pattern equivalence is included in the structural and automorphic equivalences. This means that the k-pattern equivalence is less strict than these two relations; however it is still not enough flexible for real-world networks. Some adaptations of the k-pattern vectors in order to compute similarity of vertices in real-world complex networks will be discussed in Chapter 8.

5.7 Chapter conclusions

We introduced in this chapter a method for analyzing the local structure of a graph around each vertex. This method provides a rich description of the way a given vertex is connected to the graph and also of the way its neighbors are placed in relation with each other. It can be applied both to small and large networks, and even to fractions of networks. In the following chapters we apply this method to two social networks, the first one modeling activity on an online platform and the second one modeling mobile phone communications. In the first case we study the relation between the popularity of users and the structure of the network in which they are embedded, while in the second case we compare the way the vertices and their neighbors are placed in the graph to other information (age, gender, intensity of communication) on the mobile phone users.

Chapter 6

From online popularity to social linkage: a case study of MySpace

6.1 Introduction

In this chapter we analyze the popularity of users' content on MySpace in relation with the social network in which the users are embedded. MySpace (www.myspace.com) is an online platform for social networking which gives signed-up users a free access to a personal space. In this space, users can present information about themselves, create a blog, publish different content, link to other users, visit their pages and write comments there. Although users can publish any kind of photos or videos, MySpace is especially known for the great number of music artists who present their musical compositions. Each user can declare his profile type as "member" or "musician". Besides being a place for publishing content, MySpace also offer its users the possibility to connect to each other. Thus, everybody can visit everybody's page and write comments there. Also each user can link to any other user by declaring friendship or best-friendship relations. These relations are not necessarily mutual: everybody can declare everybody as (best) friend, without waiting for the acceptance of the other part. The number of best friendship declarations is limited to 40, so one can consider the best friend links as stronger than the friend ones.

On the page of each user, all this information is visible: besides the published content and personal data, everybody sees how many people visited or left comments on the profile, how many users have declared him as (best) friend and how many users he has declared. Each user, thanks to these ratings on how many people viewed or commented his work, knows how popular his profile is. He can thus adjust his publishing and networking practices in order to become more popular, so he can develop strategies to increase his fame. Every user is manager of his own visibility thus transforming MySpace in a place for competition for popularity. The same situation happens on other online platforms that offer social networking tools and space for content publishing.

Several researchers have dealt with this competition for visibility and reputation on online platforms. Some of them concentrated on the success of contents while others focused on the reputation of individuals in the large social networks created by these

practices. For instance since the seminal work of Herring et al. [HKP⁺05], we know that influent bloggers are at the center of the social network, and that bloggers tend to link to bloggers of equal or superior reputation. See Section 4.4 for an overview of existing studies on online activities. While several researchers analyzed the popularity of contents or the social networks modeling online activities, few authors studied the relation between the two. Here, using a dataset of MySpace artist profiles, we try to hold together the two approaches: we study the popularity of MySpace artists in relation with the local structure of the social network surrounding them.

First, we build a popularity typology based on different measures of online popularity, using the Kohonen self organizing map technique (see Section 2.2). Second we analyze how the different artists are connected to each other using the method *local_structure* introduced in Chapter 5. We thus obtain a rich description of the structure of the network in which each node is embedded, that we confront to the online popularity of the artist. At the end, we obtain 5 distinct patterns of popularity on MySpace, described in terms of audience, recognition, and social structure.

6.2 Data description

We build a sample of the MySpace music (artistic) population based on the best friendship declaration links. After having chosen seven initial parent artists profiles among the French MySpace music top audience, a breadth-first-search crawler is employed to collect the profiles information, following the best friendship links during 3 iterations (best friend of best friend of best friend of the parents).

In order to verify that this sample is not unusual, we collect several networks varying the initial artists numbers (from 3 to 10), the parsing depth (from 2 to 4), the initial artists nationality and the collected artists via a randomized ID selection. If the total number of nodes and the music profiles proportion (in the selected population) depend on the crawling parameters, the ratio of the two is around 50%. Next, for each sample, a correlation test is applied between the followings four quantitative variables: number of comments, of friends, of profile visits (hits) and best-friendship declaration. A Mantel test (i.e. a matrix correlation test) is performed between the correlation tables; it shows that the coefficients are significantly similar, i.e. the variables of each sample are correlated in the same proportions.

As we are interested in the MySpace music profiles, we chose to remove from the data all the non-artistic individuals. The properties of the studied network sample are summarized in Table 6.1.

In the next section we cluster the artists in the sample using several popularity characteristics.

6.3 Analysis of the online popularity

We group the artists in our dataset in several clusters based on their popularity. We choose the following variables as a characterization of each artist's popularity:

Table 6.1: Dataset properties

Total number of profiles	21153
Artists profiles	13936
Total number of links	143831
Number of links between artists	83201
Reciprocal links rate (A and B have declared each other as best-friends)	40.1%
"Major" labeled artists	3422
"Indie" labeled artists	7069
"without" labeled artists	3445

- Number of visits of the profile (hits),
- Number of comments visitors have left on the profile (these first two characteristics are an indicator of the artist's *audience*),
- Number of people having declared the artist as best friend (this is a measure of the artist's *global authority*)
- Number of artists having declared him as best friend (the *artistic authority*) ,
- Fraction of the artist's best friends who have declared him as best friend (reciprocity rate, a measure of the cooperative behavior),
- Label (the artist's record label); this can be "Major", "Indie", or "Other".

The set of these six variables measured for each individual represent a feature vector characterizing the artist's popularity. As showed by Beuscart and Couronné in a previous study [BC09], the audience (expressed by the number of visits of the profile and the number of comments) and the authority (the number of artists/people having declared the artist as best friend) are the two main dimensions structuring the online popularity of artists on MySpace. Because the number of visits, comments and best-friendship declaration are heavily right-skewed, we use a log transformation instead of the value itself for these variables.

We now use the Kohonen self organizing maps (see Section 2.2 for a presentation of this clustering method) in order to group artists based on their popularity characteristics. As any clustering method, this technique uses as input feature vectors and groups together individuals with similar feature vectors while putting in separate groups individuals with different vectors.

The multi-dimensional processing of the set of individuals by the SOM provides Figure 6.1. The SOM result is a bi-dimensional map with 6 layers (a layer for each variable describing the individuals) where individuals are placed depending on their topological proximity. The map's smallest entity is a cell, and each individual is placed in only one cell (the individual has the same position on all the layers). Each cell has a feature vector (a vector of the six variables) computed from the feature vectors of the individuals in the

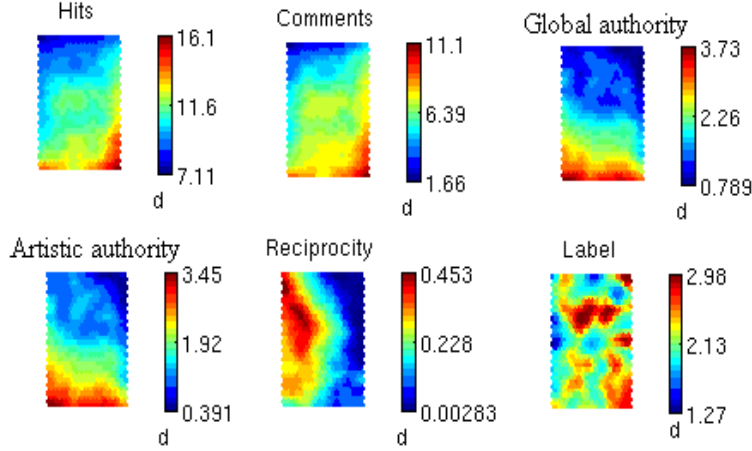


Figure 6.1: Self Organizing map of the artists depending on their popularity properties.

cell. On each layer, the color of the cell corresponds to the value of the corresponding variable for that cell. The interest of this method of clustering is the visual representation of the population for each one of the variables. Instead of the classical representation of individuals into clusters, where one does not know how the different variables contribute to individual proximity, this method provides a representation of both proximity between individuals and values of variables for the different individuals.

The obtained map appears to be structured by two independent trends: the more an artist belongs to a southern cell, the more his popularity is high, in terms of both audience and authority; and the more an artist is to the west side, the more he tends to have reciprocal links. If audience and authority are partly correlated and discriminate popular artists from anonymous, the trends are not exactly similar. Indeed, the south-western area is associated with the authoritative elites (highest artistic and global authority) and the south-eastern area is associated with the most notorious artists (highest page views and comments). If, most probably, the audience elites are not without authority and authoritative elites are not without audience, the top artists of the audience and of the authority do not overlap.

We can note that the two measures of authority (global and artistic) are correlated. The artists and the other fans create in the same way their best friendship links: the authority hierarchy follows a unique trend. Complementary, this result shows that the reciprocal links behavior is not associated with the popularity: it may be either because an authoritative artist cannot have more than 40 best friends (and therefore cannot cite everybody) or because very authoritative artists are not linking back to people who link to them (fan-star relationship). Finally we observe that the south-east area (audience elites) is associated with a strong presence of the "Major" labels.

We cluster the cells produced by the SOM using a k -means clustering. The expectation maximization algorithm is then employed to choose the best number of clusters. The population is thus distributed into 5 clusters (Figure 6.2):



Figure 6.2: The 5 clusters

Cluster1 (Cyan, population: 2732) gathers artists with a medium-to-large audience, a low authority and a weak reciprocity rate. They are mostly associated with major music labels. Our browsing of the Myspace pages of some artists in this cluster suggests that these artists, already popular offline, use their MySpace page as a display window of their music, but make very little use of the social networking tools. We may suppose that their strong audience comes from their offline popularity, but that they are not active enough to gain a strong influence on MySpace.

Cluster2 (Dark blue, pop.: 3036) gathers artists with a very strong authority, and a medium-to-high audience: these artists are not the most popular, but they are the most recommended. Most of them belong to independent labels. The qualitative browsing of their pages suggests a very intensive use of the social networking tools in order to build their online popularity. Here we find a lot of trendy groups and electronic avant-garde music, waiting for their online fame to become larger.

Cluster3 (Green, pop.: 1920) gathers artists with both a large audience and a strong authority, the MySpace elites. They have mostly major labels. Browsing their pages, we find established artists, combining traditional forms of artistic accomplishment (famous labels, presence in renowned festivals) with an active online marketing strategy.

Cluster4 (Brown, pop.: 2834) gathers artists with a very small audience and no authority. Most of their pages display very low activity, suggesting that these artists have either abandoned the page or show very little interest in online socializing practices.

Cluster5 (Orange, pop.: 2834) gathers artists with a small audience, low authority, and a strong reciprocity rate. Most of them are unsigned. On the contrary to artists from cluster 4, most of the pages we browsed are very active. These small amateur artists seem to be the ones populating the local music scenes; they are well connected to other artists from the same scene or from the same geographical area. Their small audience may not reflect their inability to reach an audience, but the small size of their musical or geographical niche.

This first part of the study provides a classification of artists based on the popularity variables. The main results are that the two dimensions of the popularity (audience and

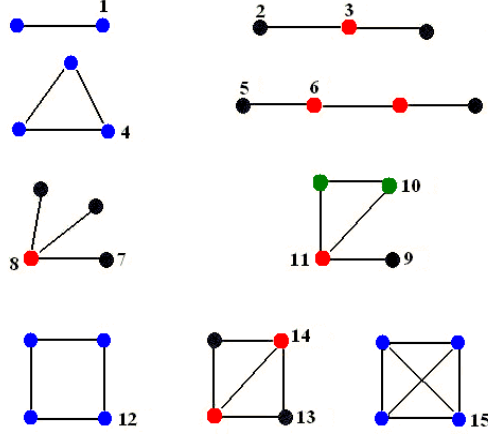


Figure 6.3: The patterns with at most 4 vertices and their positions.

authority) are correlated, but discriminate at least two elites. Moreover the best friendship links appear to have various meanings (fan - star, peers etc.). It seems relevant to study more specifically what the links distribution and network structure teach us about the best friendship significance and the artistic popularity. This is the goal of the following section.

6.4 Social network structures as a function of artists' online popularity

In this section we analyze the local structure of the social network of MySpace artists in order to see if it is different depending on the popularity cluster of the artists. We represent the sample of MySpace artists and their best-friendship declarations as a simple undirected graph where the vertices correspond to the artist profiles and the edges to the existence of a best-friendship declaration between two artists: there is an edge between the vertices u and v if u has declared v as best-friend or v has declared u as best-friend or both. The resulting graph has 13936 vertices and 65979 edges. In order to describe the local structure of the graph, around each vertex, we apply the method *local_structure* presented in Chapter 5 to all the vertices of the graph: we compute the number of occurrences of the different patterns in the egocentred network of each vertex and the positions occupied by the different neighbors in these patterns. In this chapter we use only patterns with at most 4 vertices (see Figure 6.3; we have also indexed the 15 positions in these patterns). It takes 34 seconds to run our C++ implementation of the method for all the vertices on a computer with a 2.8GHz processor and 4Gb RAM.

VERTICES. We begin by studying the structure of the graph surrounding the vertices in order to see if it differs depending on the SOM popularity cluster the vertices belong to. For this, we use the feature vectors of the vertices i.e. the number of patterns in their egocentred networks (computed by steps 1 and 2 of the method *local_structure*). We want

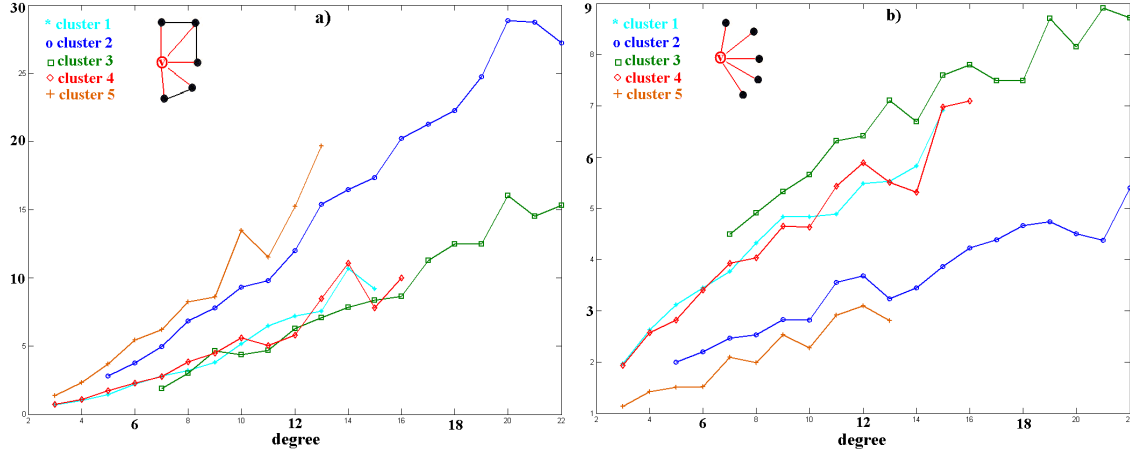


Figure 6.4: For the vertices of each cluster, the average number of edges (a) and isolated vertices (b) in the egocentred networks as a function of the degree

to compare the number of occurrences of the different patterns in the egocentred networks with respect to the popularity clusters of the vertices. As the degree distributions are not the same in the 5 clusters, one cannot simply compare the number of occurrences of the patterns; these quantities are biased by the degrees of the vertices (for instance, a vertex with a high degree probably has high values for all the patterns). Therefore, we compare the number of occurrences of patterns in the egocentred networks of the vertices with the same degree (i.e. the same number of vertices in the egocentred network). For each cluster C , each degree¹ d and each pattern P , we compute the average $FD(C, d, P)$ of the number of occurrences of the pattern P in the egocentred networks of the vertices with degree d in C . Figure 6.4 represents, for each degree d , the values of $FD(C, d, P)$ for the 5 popularity clusters; the considered pattern is the number of edges (i.e. pattern number 1, \rightarrow) in the egocentred network in Figure 6.4(a) and the number of isolated vertices in the egocentred network in Figure 6.4(b).

We observe that, for all the degrees, the vertices of the cluster 5 have the greatest number of edges in their egocentred networks, followed by those of the clusters 2, 1 and 4 and finally 3. The order is inverted for the number of isolated vertices that measures the quality of "star" of a vertex. Remember that clusters 5 and 2 are the ones on the western side of the SOM map, i.e. artists having reciprocal links, sometimes a lot of friends, but a medium to small popularity: they can be authoritative, but not with strong audience. Cluster 3, situated in the southern part of the map, contains the MySpace elite, the superstars, the popular authoritative artists. These vertices are, in terms of network structure, star centers, connecting many unlinked vertices, as Figure 6.4(b) shows.

We continue our analysis by computing, for each cluster C , each value² e of the num-

¹We take into consideration only the degrees for which there are at least 2 clusters where 1% of the nodes have that degree

²As before, we take into consideration only the values reached by at least 1% of the nodes in at least 2 clusters

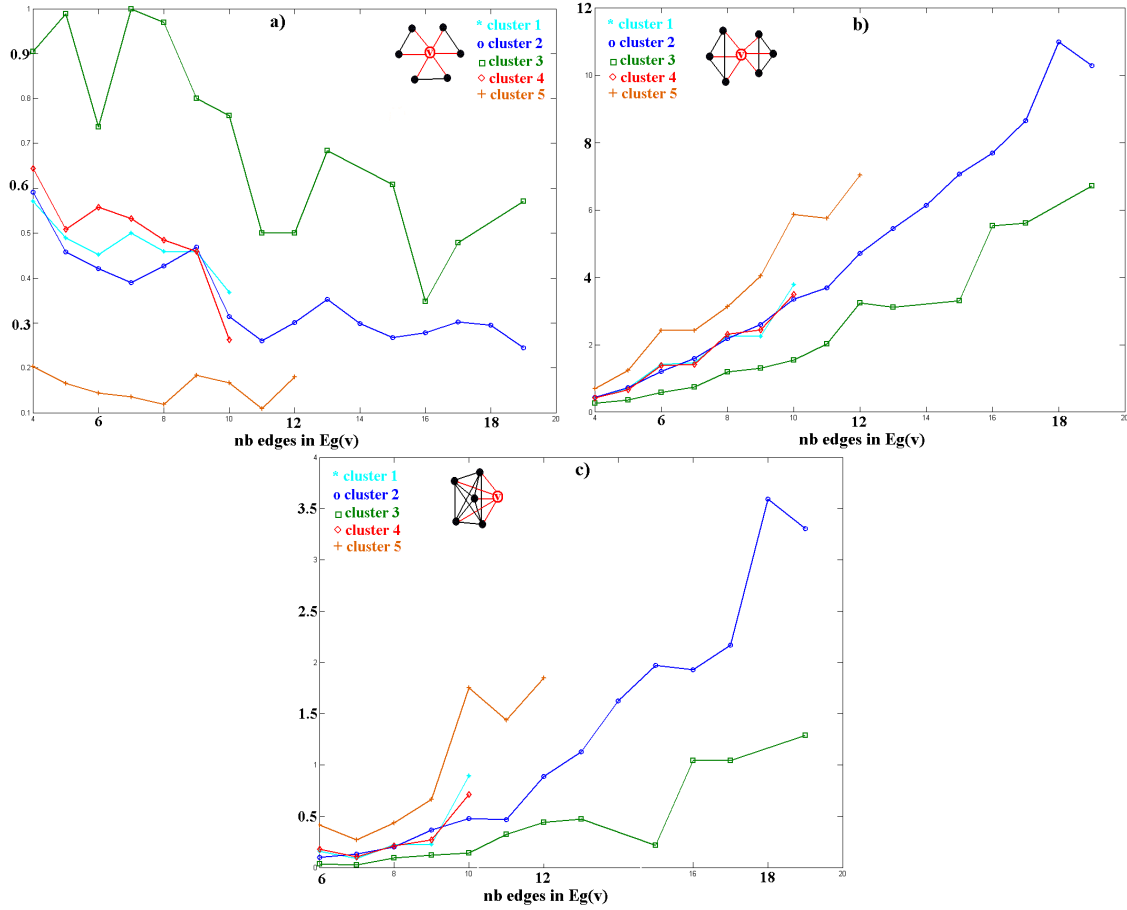


Figure 6.5: For the vertices of each cluster, the average number of isolated edges (a), triangles (b) and 4-cliques (c) in the egocentred networks as a function of the number of edges.

ber of edges in the egocentred network, and each pattern P , the average $FE(C, e, P)$ of the number of occurrences of the pattern P in the egocentred networks with e edges of the vertices in C . Figure 6.5 represents, for each value e of the number of edges in the egocentred network, the values of $FE(C, e, P)$ for the 5 popularity clusters; the considered pattern is the number of isolated edges (Figure 6.5(a)), the number of triangles (i.e. pattern number 3, \triangle , Figure 6.5(b)) and the number of 4-cliques (i.e. pattern number 9, \boxtimes , Figure 6.5(c)) in the egocentred network.

We observe that, given a value of the number of edges in the egocentred network, these edges are more likely to be found in triangles and 4-cliques for cluster 5 than for clusters 2, 1 and 4. The vertices in cluster 3 have the lowest probability to have triangles and 4-cliques in their egocentred networks. The edges between the neighbors of these vertices are often isolated (Figure 6.5(a)), confirming the character of "star" of the vertices in cluster 3.

As for the other patterns with at most 4 vertices, pattern 8 (\boxtimes) has the same order as the 4-clique \boxtimes , showing, once again, the tendency of vertices in cluster 5 to belong to dense groups and that of vertices in cluster 3 to be centers of stars. The other patterns do not present a clear order; however, for pattern 5 (\lrcorner), clusters 3 and 4 have the highest probabilities to contain this pattern in their egocentred networks and for pattern 7 (\square), it is cluster 1 that has the highest one. So, even if the number of edges in the egocentred network is the same, the structures in which these edges are placed are different for the 5 clusters, going from dense groups for the clusters 5 to sparse groups for the cluster 3.

To sum up, the social network surrounding each artist differs, depending on their popularity. The most popular artists (cluster 3) are at the center of stars; heterogeneous artists, not connected to each other, connect to these artists due to their popularity. As for artists with a medium-to-large audience, they have distinct types of insertion in the network: those in cluster 2 are inserted in dense recommendation networks, usually describing homogeneous musical universes, while those in cluster 1 belong to sparse structures. The same observation can be made for artists with a small audience: artists from cluster 5, though not very popular, are involved in dense structures, unlike artists from cluster 4 who display disconnected links. This analysis strengthens our typology, by associating types of popularity with types of insertion in the social network.

EDGES. We continue our analysis with the study of the edges formed by the vertices in the 5 popularity clusters. We want to see, for the vertices of each cluster, with which clusters they form the most of their edges and how these edges are placed in the graph. For that, we use the positions occupied by the neighbors in the egocentred network of the different vertices (i.e. the position vectors of the neighbors, computed in step 3 of the method *local_structure*). This way, we know for each neighbor u of a vertex v how many times it occurs in each one of the possible positions of the different patterns in the egocentred network of v . As the best-friendship links are directed, we add this information as weights of neighbors (as explained in Chapter 5): for a vertex v , a neighbor u has weight 1 if v has declared u as a best-friend but u hasn't, weight 2 if u has declared v but v hasn't and weight 3 if the best-friendship declaration is mutual. Also, remember that in Section 5.1 we defined three categories of positions based on their betweenness centrality and degree in the pattern: central, intermediate and peripheral. In Figure 6.3, the red positions (3, 6, 8, 11, 14) are central, the blue and the green ones (1, 4, 10, 12, 15) are intermediate and the black ones (2, 5, 7, 9, 13) are peripheral.

Let $Pos(Eg(v), u, i)$ be the number of occurrences of a neighbor u of v in the position i in the egocentred network $Eg(v)$ of v . For each cluster K we compute the probability $Pr_K(w, C, i)$ to observe a vertex with weight w of the cluster C in the position i in the egocentred networks of the vertices in K :

$$Pr_K(w, C, i) = \frac{\sum_{v \in K} \sum_{u \in Eg(v), u \in C, w} Pos(Eg(v), u, i)}{\sum_{v \in K} \sum_{u \in Eg(v)} Pos(Eg(v), u, i)}.$$

We observe that:

1. For clusters **1** and **4**, for all the 15 positions i , $Pr_{1,4}(w, C, i)$ is maximal when $C = 3$ and $w = 1$ (best-friendship links from 1 / 4 to 3). So, if one randomly picks an edge

formed by a vertex of the cluster 1 or 4, no matter the structure of the graph in which this edge is embedded, it is very probable that this edge is an out-going arc to the cluster 3. It is a star-fan relation that confirms the character of "star" of the vertices in the cluster 3 and the weak authority of the clusters 1 and 4.

2. For cluster **2**, for all the positions i , $Pr_2(w, C, i)$ is maximal when $C = 2$ and $w = 3$ (mutual best-friendship links inside the cluster). So the cluster 2, grouping artists with high (but smaller than the stars') authority and audience connects mostly to itself.
3. For cluster **3**, for all the central and intermediate positions, $Pr_3(w, C, i)$ is maximal when $C = 3$ and $w = 3$; for all the peripheral positions i.e. $i \in \{2, 5, 7, 9, 13\}$, $Pr_3(w, C, i)$ is maximal when $C = 4$ and $w = 2$ (best-friendship links from 4 to 3). So the edges formed by the vertices of cluster 3 are placed in "important" positions when they are formed inside the cluster and in peripheral positions when they are in-coming arcs. The important positions (as, for instance, position 7, the center of a star) signify that the vertices of the cluster 3 often form a central axis to which many triangles are connected i.e. many vertices, not connected to each other, connect to two linked vertices of the cluster 3. This may correspond to two popular artists of a similar music genre, where people who like the first are highly probable to like the second too.
4. For cluster **5**, for all the positions with a high degree i.e. $i \in \{4, 8, 10, 11, 14, 15\}$, $Pr_5(w, C, i)$ is maximal when $C = 2$, followed by $C = 5$, and $w = 3$ (mutual links between 2 and 5 or inside the cluster 5); for all the other positions, $Pr_5(w, C, i)$ is maximal when $C = 3$ and $w = 1$ (best-friendship links from 5 to 3). Remember that this cluster has a high reciprocity of links. The vertices here share symmetric edges especially with the vertices in cluster 2 and with themselves; these edges are often placed in dense groups (cliques, maybe with few missing edges), as the positions $\{4, 8, 10, 11, 14, 15\}$ show. We observe also a fan-star relation of the vertices in the cluster 5 towards the vertices in the cluster 3 (the other positions). The edges with cluster 3 are directed towards this cluster and are placed in peripheral or low-degree positions (for instance, the position 7 corresponds to the connection of the edge to a central axis, the position 9 to the connection to a clique etc.).

6.5 Chapter conclusions

By applying the SOM clustering method and the *local_structure* method introduced in Chapter 5 to a sample of MySpace artists, we obtained a rich description of the popularity of users. We compared two dimensions: the online popularity of the users and their connectivity in the social network.

Our approach reveals in a robust and efficient way that the best friendship links on MySpace wear various meanings, creating multiple popularity patterns. Next to unsurprising categories (clusters 3 and 4, very popular artists and unknown artists), we identify two different kinds of mid-range popularity (clusters 1 and 2), and a category of small

but socially active artists (cluster 5). We show that artists in these categories exhibit different insertions in the social network. Artists with a low authority and non reciprocal links tend to declare very popular artists as best friend thus generating a star structure. On the contrary, some mid-range and low popularity artists form small cliques with local neighbors, creating communities without stars but with triangles.

The self organizing map, providing a visual result, appears to be strongly relevant for the study of sociological multivariate data integrating non linear effects. In addition, the computation of patterns and positions of vertices in egocentred networks seems a good way to reveal the local structure of the social linkage. When put together, these methods unfold a rich and intuitive set of meaningful information.

This set of methods can be easily applied to any social network where the corresponding graph can be built and the activity of the users can be measured. An immediate transposition is feasible to the Flickr and YouTube platforms, where the popularity can be defined by the same parameters as on MySpace. Even more, the analysis can be adapted to some offline social networks as those modeling mobile phone communications, where calls frequency and duration measure users' activity.

In the following two chapters we analyze precisely a mobile phone social network, but in a different way than the study of users' popularity on MySpace. In Chapter 7 we describe the social network and some basic statistics; then we compare the positions occupied by the different neighbors of each vertex (ego) to the quantity of communication with ego. Next, the analysis we perform in Chapter 8 can be seen as going the other way around than that on MySpace: instead of clustering individuals based on their activity and then look at the social network structures, we cluster nodes based on the way they are embedded in the network and then look at the communication characteristics of the different clusters.

Chapter 7

Mobile phone uses and social network structure: an analysis of a mobile phone graph

7.1 Introduction

The last two chapters of this part are dedicated to the analysis of a social network modeling mobile phone communications. We study a database containing the recordings of one month of communications of 3 million persons. We are interested in several questions that can be grouped into 3 topics: mobile phone usage, structure of the social network and socio-demographic effects. For the mobile phone uses, we compute some statistics on frequency and duration of calls and number of SMS. We compare this information to users' age and gender. For the structure of the social network, we model the mobile phone communications set by a graph that we analyze at the local level. In this chapter, we identify characteristic patterns of the local structure of the graph. Also, we study the relative positions that the different contacts of a person occupy in his egocentred network. The next chapter is dedicated to a clustering of individuals based on the social structures in which they are embedded. We compare the obtained clusters to the other two dimensions of our data: the mobile phone usage and the socio-demographic information.

7.2 Data description

The analyzed dataset contains the recordings of the mobile phone communications of the customers of Mobistar in Belgium during the month of October 2006. Mobistar is a mobile phone operator that has approximately 30% share market in Belgium. The dataset contains several details of each mobile phone communication in the Mobistar network: the identifiers of the two persons in communication, their mobile phone operators (for the communication to be stored, at least one of the two persons must be a Mobistar customer), the type of communication (this can be call or short message SMS), the time when the communication began and its duration (in the case of a phone call). The phone

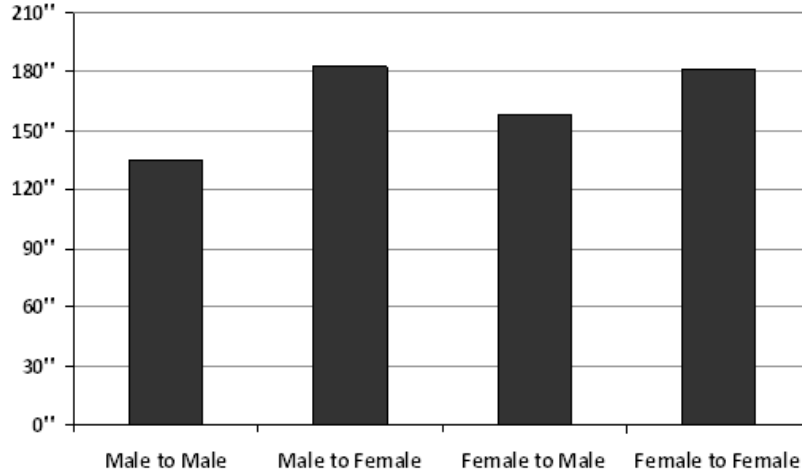


Figure 7.1: Mean call duration (in seconds) according to call initiator and receiver gender.

numbers have been hashed and each person has been given a unique identifier that does not allow finding the identity of the person. The dataset contains over 1 billion recordings involving 3.3 millions users. As we do not have the mobile phone communications between the persons not belonging to Mobistar, we keep in our analysis only the communications where the two persons are both Mobistar customers. For Mobistar customers the database contains also their age and gender. Before using this information, we compared the age and gender distribution of the mobile phone customers in our dataset (i.e. the fraction of customers of a given age and gender) to the distribution in the Belgium population. The differences between the two are very small, so there is no systematic bias in the Mobistar data as regarding these two characteristics (except for people over 55 who are underrepresented among mobile phone users).

First, we computed some statistics of mobile phone usage. The idea was to test, at a large scale, some existing results obtained from interview data. These previous observations concern gender effect on communication duration. As explained in Section 4.3, several sociological studies showed that calls were longer when a woman was called. This is because conversations with women tend to go through longer introductive and closure sequences, to be multi-thematic and digressive in nature, while conversations with man tend to be linear and monothematic. Actually, the callers seem to adjust their interaction style to the gender of the receiver. Using the mobile phone dataset, we observed the same pattern (see Figure 7.1): mobile phone calls towards a woman are, in average, longer than calls to a man, whatever caller gender is. Also, when isolating mixed-gender pairs who communicate in both directions (i.e. a man and a woman who call each other), we observe a higher average duration of calls when it is the man who calls: 171 seconds as opposed to 162 seconds when the woman calls the man.

Next, we compared mobile phone usage by age. This seems interesting as different generations of people began to use the mobile phone at a different age. As the mobile

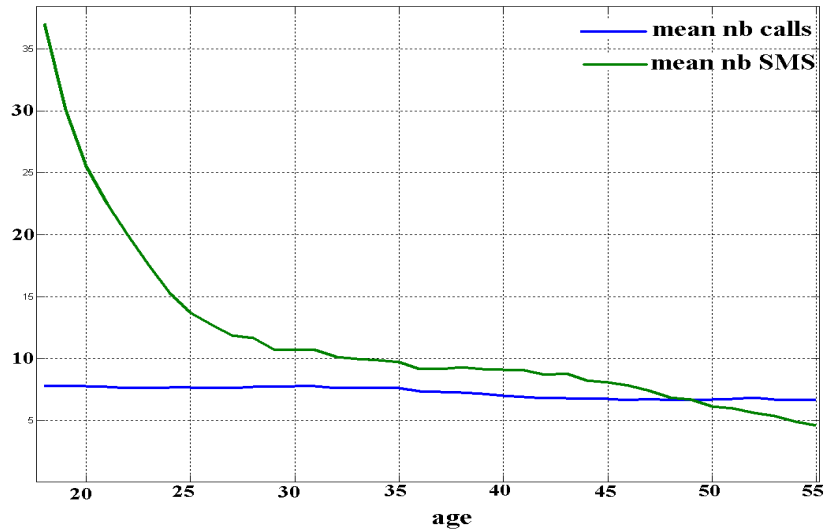


Figure 7.2: Average number of calls (blue line) and SMS (green line) as a function of phone user's age.

phone diffusion started in the mid-1990, there are only the nowadays youngest groups of population who entered in their "communication age" directly with a cell phone at hand. We thus expect a different usage of the mobile phone, especially for young people. Figure 7.2 shows the average number of out-going calls and SMS by age during the studied month, while Figure 7.3 shows the mean call duration by age. We observe no important difference in the number of calls by age. For the mean duration of a call, we observe that people from 28 to 35 have in average the longest calls (these are out-going calls, so the age is that of the caller), while people from 42 to 51 have the shortest. However, the differences are not very important, the highest mean (for the age of 28) being only with 12% higher than the lowest mean (for the age of 48). The main distinction concerns SMS usage: younger users send more SMS than older ones. In the age group 18 to 25 this tendency is really impressive: the SMS is used 4 times more frequently than a conversational exchange. Also, the SMS usage seems to be more "feminine" in general and, for the youngest part of the population (aged 18 – 25), the between-gender "texting" is particularly popular (Figure 7.4). Some authors indicate that heavy SMS use in youngster's relation with other gender is related to seduction tactics where a direct voice contact can be more "risky" for interlocutors [LY05].

While these measures represent a first analysis of the mobile phone communication data, our purpose is to study the social network modeling this data, in general, and the local structure, in particular. The remaining part of this chapter and the following one deal with the analysis of the mobile phone social network, from a local point of view, and with the correlation between local structure and intensity of communication or customers' age and gender.



Figure 7.3: Average duration of calls (in seconds) as a function of phone user's age.

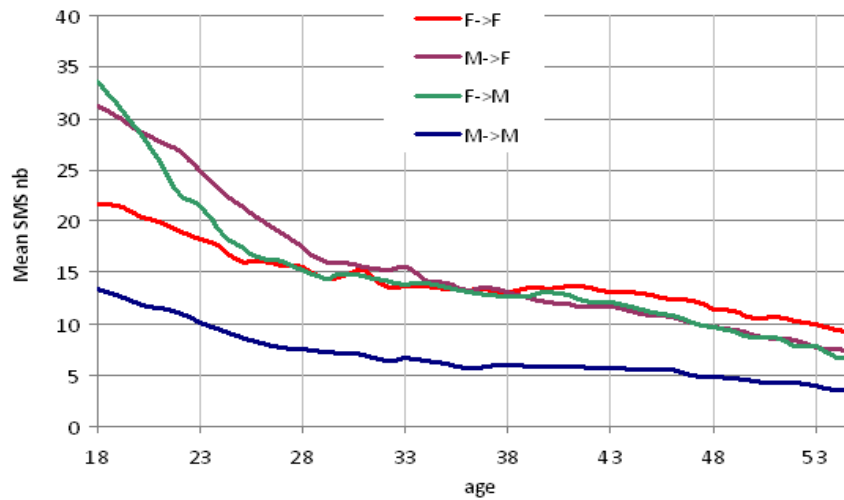


Figure 7.4: Average number of SMS from female to female (red line), female to male (green), male to female (violet) and male to male (blue).

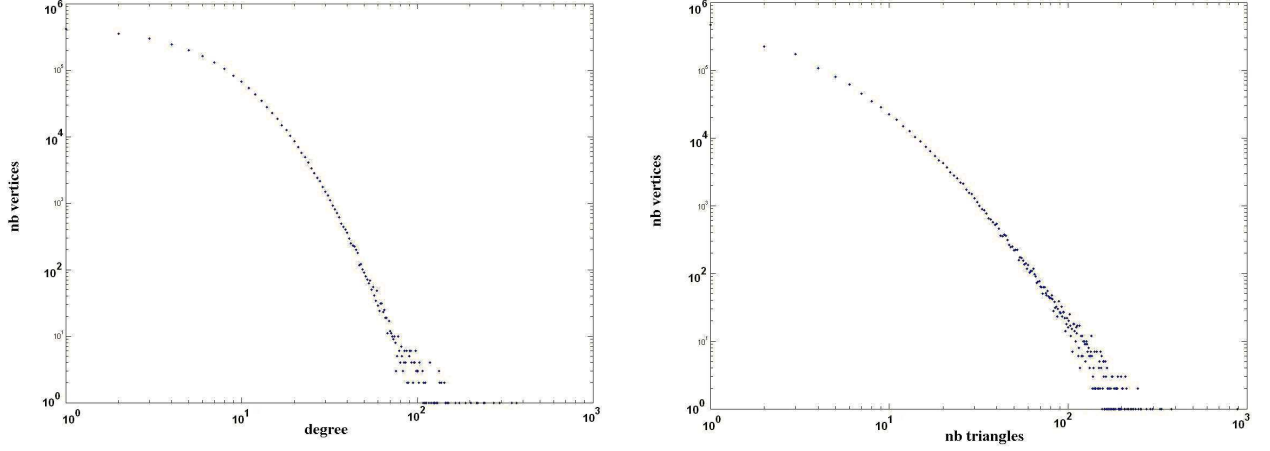


Figure 7.5: The distribution of the degree (a) and of the number of triangles (b) for the $2.7M$ vertices

7.3 Mobile phone graph

We model the mobile phone communications set by a simple undirected graph G . In this graph the vertices are the customers; we connect such two vertices by an undirected link if there had been at least one communication in each direction between the two persons during the followed period. This way we do not take into consideration the one-way contacts (calls or messages), single events in most of the cases suggesting that the two individuals do not know each other personally. We keep only the vertices with degree greater than 0, thus obtaining a graph G with 2.7×10^6 vertices and 6.4×10^6 edges. This graph shares the characteristics of complex networks. It has a giant connected component containing 83% of its vertices and 99% of the edges. As mentioned in other studies (e.g. [OSH⁺07a]), the degree distribution is very heterogeneous, with a large number of vertices having a small degree and only a small fraction having a high degree. The same statement is valid for the number of triangles containing a node (i.e. the number of edges connecting its neighbors). Only 20 vertices (i.e. $7 \times 10^{-4}\%$ of the vertices) have more than 100 neighbors connected by more than 100 edges. The distributions of the degree and of the number of triangles are presented in Figure 7.5, while Table 7.1 contains the minimum, maximum, average and median values of the two parameters. The clustering coefficient of the graph (computed as the mean value of the clustering coefficient of the vertices) is relatively high, being equal to 0.097.

In this graph, we apply the method introduced in Chapter 5 in order to analyze the local structure of the network. Remember that the method computes, for each vertex, the number of occurrences of the different patterns (Figure 7.6) in its egocentred network, and also the position vectors of its neighbors. Thus we have a description of the way the vertex is connected to the graph (given by the patterns present in its egocentred network)

parameter α	min	max	average	median	nb. networks s. t. $\alpha > 100$
degree	1	367	4.66	3	56
nb.triangles	0	887	2.28	1	560

Table 7.1: Different measures for the degree and the number of triangles containing a vertex.

and of the way its neighbors are placed in relation with each other. The mobile phone graph has $2.7M$ vertices, so the method describes $2.7M$ egocentred networks. Our C++ implementation of the method takes 31 minutes to characterize the entire set of vertices on a computer with a standard configuration: a 2.8GHz processor and 4Gb RAM.

Empirical complexity of the method. Let us discuss the complexity of our method when it is applied to the mobile phone graph G . As explained in Section 5.4, the complexity of the method depends on the number of patterns in each egocentred network. Actually, it is the enumeration of patterns and positions of vertices in each egocentred network (Algorithm *CharacterizeWithPatterns*, see Section 5.4) that depends on the number of patterns in the egocentred network. It is the complexity of this algorithm that we want to analyze. As presented in Section 5.4, the time complexity is linear in the number of patterns in the egocentred network. We do not have a method to a priori estimate the number of patterns, so let us evaluate the complexity of the algorithm a posteriori, after having computed the patterns in all the egocentred networks. For a vertex v in the mobile phone graph G , let n_v be the number of vertices in its egocentred network $Eg(v)$, m_v the number of edges and p_v the number of patterns. For all the egocentred networks in G , we have $p_v < m_v^3$, and, for 98.5% of these graphs, $p_v < m_v^2$, so for the egocentred networks of our graph G the observed time complexity of Algorithm *CharacterizeWithPatterns* is $O(m_v^2)$ in 98.5% of the cases and $O(m_v^3)$ in the rest of the cases. Given that most egocentred networks have a low number of edges, it is not very time-consuming to list all the patterns and to compute the positions occupied by the different vertices.

To finish this discussion of the empirical complexity of our method, we compared the time complexity of Algorithm *CharacterizeWithPatterns* to that of the method proposed by Kloks et al. [KKM00] that counts the induced subgraphs with exactly 4 vertices. Given that, in this method, the number of vertices of the searched subgraphs is 4, we also use Algorithm *CharacterizeWithPatterns* for listing patterns with at most 4 vertices. On the one hand, for an egocentred network with n_v vertices, the complexity of Kloks' algorithm is $O(n_v^\alpha + e^{1.69})$, where $O(n_v^\alpha)$ is the time needed to compute the square of the adjacency matrix of G . On the other hand, for each vertex v in G , the number of induced subgraphs with at most 4 vertices is smaller than $(2 \times m_v)^2$ and than $(5 \times n_v)^2$, so the time complexity of Algorithm *CharacterizeWithPatterns* is $O(n_v^2)$ for all the egocentred networks in G . Therefore, for the mobile phone graph, the time complexities of the two methods are comparable. So it is worth listing all the patterns, given that we make a step further by computing not just the number of the different patterns but also the positions occupied by the different vertices.

After applying the method to our mobile phone graph, the first analysis we perform is the computation of characteristic patterns. This analysis, related to the problems of iden-

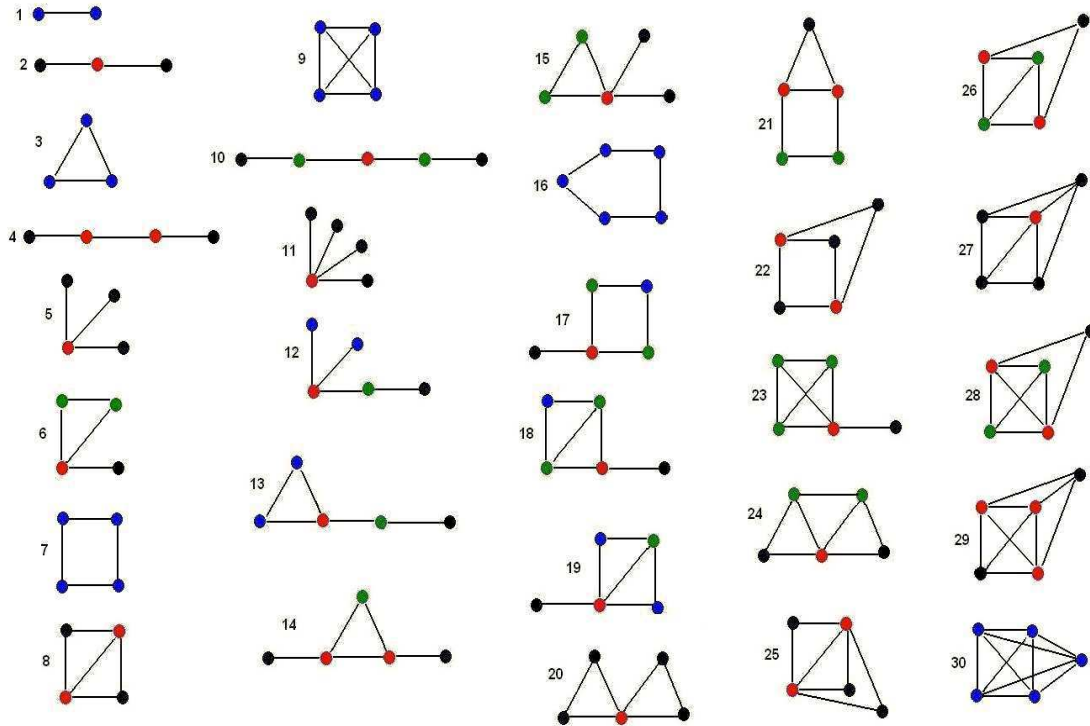


Figure 7.6: The set of patterns and their positions.

tification of network motifs and frequent patterns, is presented in Section 7.4. In a second analysis, we study the way the different neighbors are placed in the egocentred networks and we compare our observations to the intensity of mobile phone communication. This is presented in Section 7.5. Finally, in Chapter 8 we cluster individuals in the mobile phone network based on the way they are connected to the network, thus addressing the problem of identification of roles in a social network.

7.4 Characteristic patterns

When characterizing the egocentred networks of the vertices in the mobile phone graph with the method introduced in Chapter 5, we obtain the number of occurrences of each one of the patterns (Figure 7.6) in each one of the egocentred networks. This allows us to address the problem of identifying "characteristic" patterns. For this problem, several authors proposed different definitions and algorithms for computing them. As we have already counted the patterns, we are able to compute the characteristic patterns according to the different existing definitions.

Let us first denote by D the set of the egocentred networks of all the vertices in the mobile phone graph. There are several possible definitions for a characteristic pattern P for a set of graphs D :

- Def 1.** the number of occurrences of the pattern P as induced subgraph of the graphs in D is greater than a given threshold;
- Def 2.** the number of graphs in D that contain the pattern P as induced subgraph is greater than a given threshold (this is the problem of *identifying frequent patterns* that we presented in Section 3.3 and that was treated for instance in [HS, KK01, IWM00]);
- Def 3.** the number of occurrences of the pattern P as induced subgraph is higher for the graphs in D than for randomly generated graphs of same sizes (this is the problem of *identifying network motifs* that we also presented in Section 3.3 and that was introduced in [MIK⁺04]).

Definition 1. We compute, for each pattern P with $k \leq 5$ vertices, the number of occurrences of P as induced subgraph of graphs in D divided by the number of occurrences of a pattern with k vertices in D , i.e. the probability that the subgraph induced by k connected vertices of a graph in D represents the pattern P . Figure 7.7 shows the values of these probabilities for $k > 3$. We observe that the patterns that occur the most are the paths and the stars (possibly with an extra edge). Note however that the counting of all the occurrences of a certain pattern gives an advantage to those containing vertices of degree 1. For instance, in the case of 4-nodes stars \star (pattern 5 in Figure 7.6), the presence of a 6-nodes star in an egocentred network implies counting $\binom{6}{4} = 15$ occurrences of the pattern 5, \star . By this definition, some patterns are given an advantage, they occur more often simply because of the combinatory and probably not because they are characteristics for our set of egocentred networks.

It seems more plausible to count either the egocentred networks that contain a certain pattern and thus find the frequent patterns (as in definition number 2), or to refer to a null model in order to have an estimation of the expected number of occurrences of the different patterns (as in definition number 3).

Definition 2. Figure 7.8 represents, for each pattern P with $k \leq 5$ vertices, the number of graphs in D that contain P as induced subgraph divided by the number of graphs in D that contain at least one pattern with k vertices, i.e. the probability that a graph in D with at least k connected vertices contains P . We observe that the most frequent patterns are the paths, possibly with one extra edge (added to form a star or a triangle). However, it is possible that these patterns appear more often than others simply because of the degree distributions of the egocentred networks in which they are counted and not because they have a special meaning.

It thus seems a good idea to compare the number of occurrences of the different patterns to their occurrences in randomly generated graphs. This way we can see which patterns occur in our egocentred networks because there is a reason bringing vertices together, and which patterns occur often just because of the combinatory of the egocentred networks. It is the third definition that looks for patterns occurring more often than in random networks.

Definition 3. For each connected component of a graph in D we randomly generated connected graphs using the method introduced in [MKFV06]. As explained in Section

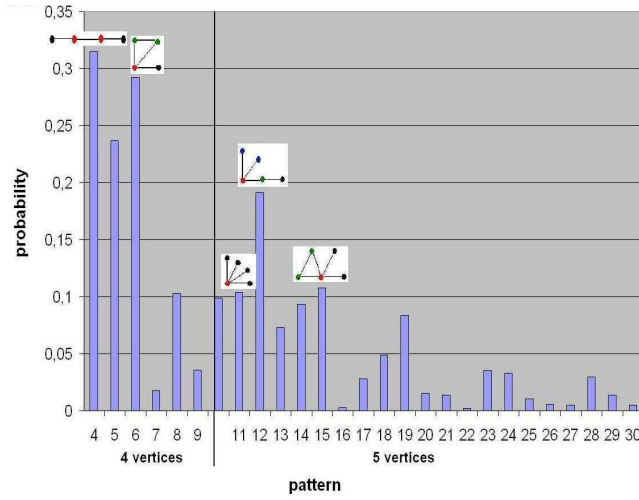


Figure 7.7: For each pattern with k vertices, the probability to be the subgraph induced by k connected vertices in D

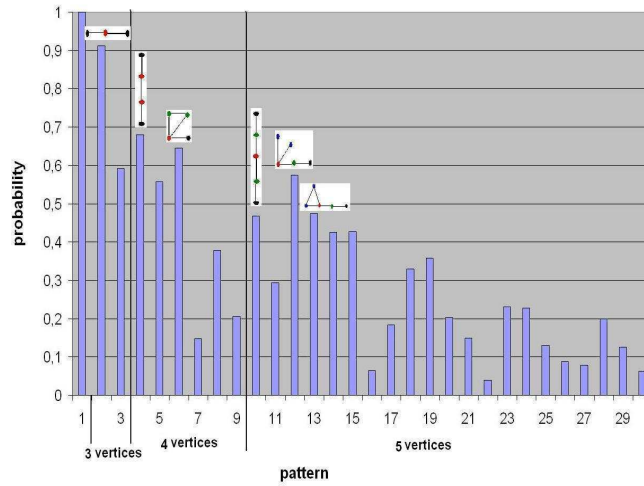


Figure 7.8: For each pattern with k vertices, the probability to occur in a graph in D that has at least k connected vertices

3.2, this method uses dK —series of probability distributions (i.e. all degree correlations within d —sized subgraphs). We built graphs for $d = 1, 2$ and 3 respectively. For $d = 1$, the generated graphs preserve the degree distribution of the original graphs, thus assuring also the same number of vertices and edges. For $d = 2$, the joint degree distribution is preserved, thus keeping also the same degree distribution. For $d = 3$, the graph generation preserves the number of triangles and wedges (i.e. chains of 3 vertices connected by 2 edges) between vertices with degrees $k_1, k_2, k_3, \forall k_1, k_2, k_3 \in \mathbb{N}$.

For each value of d , let R_d be the set of randomly generated graphs. For each pattern, we compute the ratio between its number of occurrences in the graphs in D and in the graphs in R_d . When the graphs in D are compared to the graphs in R_d , the patterns with the greatest values of the ratios are characteristic for the graphs in D and the ones with the smallest values are underrepresented. For $d = 1$ and $d = 2$, the same patterns are identified as characteristic (see Figure 7.9), with smaller values of the ratio for $d = 2$ than for $d = 1$. These patterns suggest that, although the densities of the input graphs are preserved in the generated ones, there are graphs in D that are locally denser than the corresponding generated ones. So, in the neighborhood of certain vertices, several neighbors form dense clusters; these clusters may correspond to the different groups of contacts of those persons. Note however that the two generations preserve the clustering coefficients of the graphs in D . When $k = 3$, the clustering coefficient is preserved (along with some other conditions, see Section 3.2) and the observed values of the ratio are placed between 0.99 and 1.003 for all the patterns. The generated graphs essentially reconstruct the original ones, so the $3k$ —distribution suffices in order to capture the distributions of the different patterns in the neighborhood graphs in GM . Nevertheless, this generation is very constraining for small graphs like those in D ; in many cases there is only one graph that has the $3k$ —distribution of the original one: the original one.

To sum up, computing characteristics patterns is not an easy job. Each one of the definitions has its limitations. Even if the third definition seems the most useful, the method for graph generation influences a lot the results; the characteristic patterns found by using a certain graph generation method may not appear as characteristic if one changes the method.

7.5 A characterization of ego's contacts

By applying the method *local_structure* described in Chapter 5 to each vertex (also called ego) v of the mobile phone graph, we obtain a description of how the neighbors of the vertex are placed in relation to each other. This description is a vector, called position vector, computed for each one of the neighbors u of the vertex v . It contains the number of occurrences of u in the different positions of the patterns identified in the egocentred network of v (Figure 7.6 represents the patterns and their positions; in each pattern each position has a different color). We want to see if there is a relation between the feature vectors of the different neighbors, so between their positions in the egocentred network, and the intensity of their communication with v . We thus compare the position vectors to the number of calls with v and to the total duration of calls. Note that the position vectors

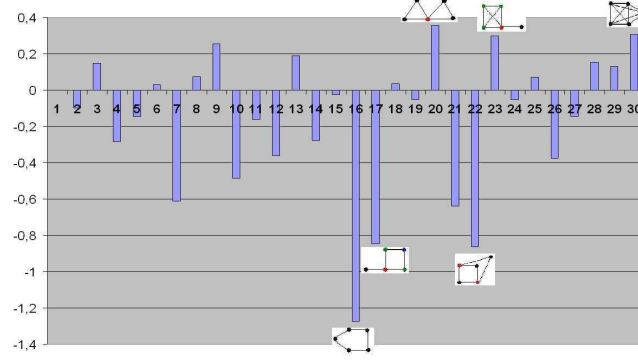


Figure 7.9: For each pattern, \log_2 of the ratio between its number of occurrences in D and in R_2

are relative quantities: they are completely conditioned by the links of each neighbor with the other neighbors. We compare these quantities to the intensity of communication that is also relativized: for each neighbor, we use the number and the total duration of calls with ego not as absolute values, but as compared with the values for the other neighbors.

The maximal number of calls

First, for each ego v , we rank its neighbors depending on the number of calls they exchanged with him: the greater the number of calls exchanged with ego, the smaller the rank (denoted by $rank_v$), such that the vertex with the greatest number of calls has rank 1 and the one with the smallest number of calls has rank $d(v)$ (i.e. the degree of v).

Let D_5 be the set of vertices (egos) in the mobile phone graph that have degree at least 5. For each ego $v \in D_5$, we study the positions occupied in its egocentred network by its neighbors with ranks 1, 2, 3 and 4 and by a randomly chosen neighbor among those with rank greater than 4, to which we give the rank 0. In order to analyze the positions of the different vertices, we answer two questions regarding the entire set D_5 :

- Q1** given a position in a pattern, which of the five ranks occupies this position the most frequently and which one the least frequently?
- Q2** given a pattern and an rank $r < 5$, in which position of the pattern the vertices with rank r appear the most frequently and in which one the least frequently?

For a rank r , let $I(r)$ be the set of all neighbors that have rank r along with the corresponding egos: $I(r) = \{(u, v) \text{ s.t. } u \text{ is a neighbor of } v, d(v) \geq 5 \text{ and } rank_v(u) = r\}$. For a position i (of all the possible positions of the different patterns), let $Pos(Eg(v), u, i)$ be the number of occurrences of the neighbor u of v in the position i in the egocentred network $Eg(v)$ of v . Also, let $Nb(r, P)$ be the total number of occurrences of a neighbor with rank r in any position of the pattern P : $Nb(r, P)$ is the sum of occurrences, for all egos $v \in D_5$, of their neighbors with rank r in the different positions of the pattern P , so $Nb(r, P) = \sum_{i \in P} \sum_{(u, v) \in I(r)} Pos(Eg(v), u, i)$. We now compute the probability that, when a vertex with rank r occurs in a position of the pattern P , this position is i :

$$Pr(r, i, P) = \begin{cases} 0 & \text{if } i \text{ is not a position of } P \\ \frac{\sum_{(u,v) \in I(r)} Pos(Eg(v), u, i)}{Nb(r, P)} & \text{otherwise.} \end{cases}$$

Figure 7.10 presents these probabilities for all the 5 ranks and all the patterns (as in Figure 7.6) with at least two positions. Remember that in Chapter 5 we classified the different positions of each pattern in 3 categories, central, intermediate and peripheral, based on their betweenness centrality and degree. Briefly the positions colored in red in Figure 7.6 are central, those colored in black are peripheral and the others one are intermediate.

Question Q1. We observe that, for all the central positions (the maximal index in each image, in the right side), the probability of occurrence in these positions of the vertices with rank 1 is greater than that of the vertices with rank 2, which is greater than that of the vertices with rank 3 etc. The opposite situation happens for the peripheral positions (the minimal index in each image, in the left side) where the randomly chosen vertex has the greatest probability of occurrence. For the intermediate positions, the vertices with the greatest probability of occurrence are generally those with ranks 2, 3 or 4.

Question Q2. We observe that the vertices with rank 1 occupy most frequently the central positions and least frequently the peripheral ones (the red curves are generally ascending or at least higher in the right side than in the left). The randomly chosen vertices occupy mostly the peripheral positions and least frequently the central ones (the black curves are generally descending), while the vertices with ranks 2, 3 and 4 have a tendency placed between these two.

So, when they appear in a pattern, the vertices with rank 1 tend to occupy the central position of the pattern; they have an important role, connecting several neighbors otherwise disconnected. The roles played by the vertices with the next three ranks are less important; they generally occupy the intermediate positions of the different patterns. The randomly chosen vertex has a marginal role, generally being connected to the vertices around it in a peripheral position. Note however that the presence of a node in the different positions is not equivalent to its centrality: even if a node is not the most central (in terms of betweenness centrality), it may occupy the central position of the different patterns. This can be shown, for instance, by looking at the egocentred networks where the neighbor with rank 4 is the most central. We compute, as before, the probabilities Pr for the vertices in these graphs. Even if the vertex with rank 4 is the most central, it has a smaller probability of occurrence in the central position of the patterns 5, 6, 12, 13, 18 than the vertices ranked 1, 2 or 3.

The maximal sum of duration of calls

We analyze, for the egocentred network of each vertex in the mobile phone graph, the position occupied by the vertex that had the greatest sum of duration of calls with ego. In 78.2% of the cases, the person that exchanged the greatest number of calls with ego (the vertices with rank 1 of the previous section) is also the person that has the greatest total duration. In the other cases, we give rank 1 to the vertex with the greatest number of calls and rank 2 to the vertex with the greatest sum of duration of calls. We also randomly

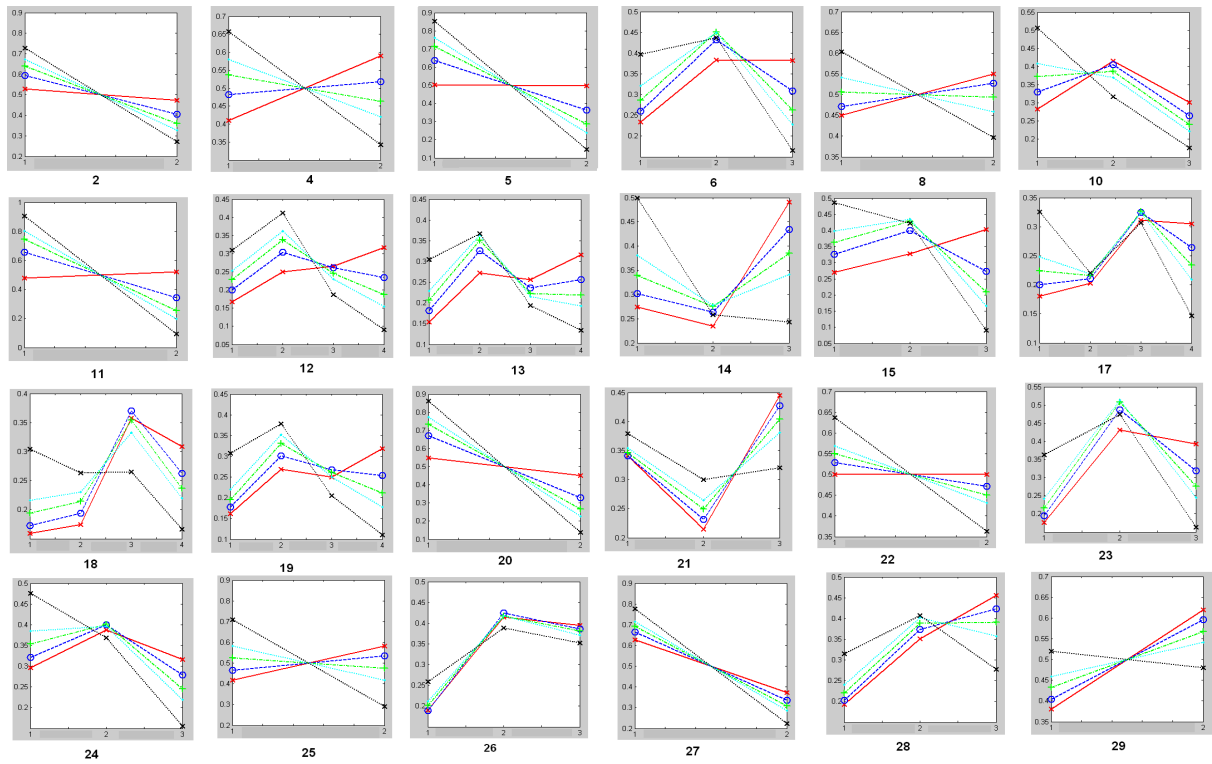


Figure 7.10: For each pattern P (each image) and each position i in P (x-axis in each image), the probability (y-axis) of occurrence of a vertex with rank r in i : rank 1—red dots, 2—blue dots, 3—green dots, 4—cyan dots, 0—black dots. In each image, the order of the positions on the x-axis corresponds to the ascending order of betweenness centrality and degree: the maximal value corresponds to the central position in the pattern, while the value 1 corresponds to the peripheral one.

choose a vertex among the other neighbors of ego. By performing a similar analysis to that of the previous section, we observe, for each pattern, that the probability of the vertices with rank 2 to occupy the central position is smaller than that of the vertices with rank 1 but higher than that of the randomly chosen vertices. The opposite situation happens for the peripheral positions. When they appear in a pattern, the vertices with rank 2 tend to occupy the intermediate positions.

Comments on the results

Our data provides us two measures of the intensity of communications between each ego and his neighbors: the frequency and the duration of calls. It seems intuitive that the person who speaks the most with ego has an important role in his network. However, when it is not the same person that has the greatest frequency of calls and the greatest duration, it is interesting to see which of the two actors has a more important role in ego's neighborhood. Using the number of occurrences in the different positions, we saw that it is the one that has the greatest frequency who has a more important role.

Remember that in Section 4.3 we presented a sociological study by Licoppe and Smoreda on phone communications [LS05]. In this study the authors, using databases of telephone calls and several interviews focusing on the use of telephone, identified two patterns of communication, the "connected presence" and the "intermittent presence". In the first one, the two persons, socially and often also geographically close, are frequently in contact with each other, exchanging many short calls and messages. They share activities that require numerous calls for synchronization and coordination, the mobile phone being especially suitable for this. It seems plausible that the persons that speak the most frequently with ego are well involved in ego's network, being well connected to other neighbors. Indeed, we saw that the actors that communicate the most with ego tend to occupy the central positions of the patterns where they appear.

In the second pattern identified by Licoppe and Smoreda, the two persons, close friends or intimate relatives, are not able to see each other or talk very often. Their conversations are long, they give and receive news, trying to compensate for the rarity of face-to-face contacts. The person that has long but rare calls with ego is probably geographically far from him, while the persons that have a great frequency of calls are generally geographically close. This hypothesis is confirmed in [LBdK⁺08], where Lambiotte et al. show that the probability of a mobile phone call between two persons is inversely proportional to the square of the geographical distance between them. Being far from ego, the person that has the greatest duration of calls but not the greatest frequency is less implied in ego's network, his role is less important. However, the duration of the calls suggests that he is sociologically close to ego, hence his more important position than that of a randomly chosen neighbor.

7.6 Chapter conclusions

In this chapter we presented an analysis of a dataset of mobile phone communications. We first computed some statistics of phone usage and then we analyzed the local structure

of the graph by using the method introduced in Chapter 5. Until now we addressed the question of computing the characteristics patterns of the egocentred networks, thus relating to some very popular problems in pattern discovery, data mining and bio-informatics. Next we analyzed the positions occupied by the neighbors of each vertex in its egocentred network, thus addressing the notion of the roles played by the different vertices in the egocentred network. When we compared the relative positions of the neighbors to the intensity of communications with ego, we found that the person who had a great frequency of calls with ego had, in average, an important position in its egocentred network. This position is generally more important than that of the person who has the greatest duration of calls with ego.

In the next chapter we group together vertices having similar egocentred networks and we confront the different groups to the quantity of communications and to the socio-demographic data.

Chapter 8

A local structure-based clustering of nodes

8.1 Introduction

Remember that, for each individual in the mobile phone network, we listed the patterns of his egocentred network. The number of occurrences of the different patterns represents a description of how each node (so each individual) is connected to the network. In this chapter, we use this description in order to group individuals into clusters: nodes are put in the same cluster because they are connected to the network in similar ways; nodes put in distinct clusters are differently embedded in the network. One can see this distribution of nodes into clusters as an identification of roles played in the network, as presented in Section 4.1.1. Without pretending to have solved the problem of identification of roles, we present a method to distribute nodes into clusters based on the local structure of the network. We use the k-pattern vectors that we have defined in Section 5.3, but in a different way than in the definition of the k-pattern equivalence (that we have also introduced in Section 5.3).

There are of course many ways of clustering nodes of a network, but the method we propose here gives quite promising results, in particular when they are confronted to other characteristics of the individuals. Indeed the probability that an individual belongs to a certain cluster depends on his age; even more, using these probabilities we are able to group together different ages, thus discovering 4 groups containing consecutive ages, corresponding to 4 life stages. The probability that a person belongs to a certain cluster also depends on his mobile phone communication intensity; moreover the intensity of communication allows us to predict with rather high accuracy the cluster a person belongs to.

We begin by presenting the method for grouping nodes into clusters based on the structure of the network in which they are embedded. We then confront the obtained clusters to age, gender and intensity of communication. We finally provide a typology of the mobile phone users in our dataset based on social network cluster, intensity of communications and socio-demographic data.

8.2 A method for nodes clustering using patterns frequency

In this section we want to group together the vertices of a given large graph G which are connected in the same way to the network. This is the problem of identification of social roles that we presented in Section 4.1.1.

8.2.1 Pattern-frequency equivalence

Generally, when computing roles of nodes in a network one defines an equivalence relation between nodes: equivalent nodes are considered to have the same role. In Section 5.3 we have defined such an equivalence relation called k -pattern equivalence. This relation is based on k -pattern vectors. We recall the two definitions here.

Definition 8.2.1. *Given a vertex v of a graph G and a positive integer k , we call k -pattern vector of v the vector containing the number of occurrences of the k -patterns (i.e. all the non-isomorphic connected graphs with at most k vertices) in the egocentred network $Eg(v)$ of v . Two vertices of the graph G are said to be k -pattern equivalent if and only if they have identical k -pattern vectors.*

Although the k -pattern equivalence is less strict than the structural and the automorphic equivalences (as explained in Section 5.6), it is still not flexible enough for real-world networks. The problem is that the equivalence classes obtained when applying the definition to large graphs are much too numerous. Here we want to group the nodes of a given large network into a small number of classes (i.e. smaller than a given constant, for instance 20). Each class should contain similar nodes in terms of network structure. It is the local structure of the network surrounding the node that should matter when attributing a node to a class, and not its degree or the fact of being connected to other nodes in the class. The interest of computing such classes is that they are very easy to use. Thus, one can measure correlations with other properties of the nodes or make predictions (e.g. predict a property when knowing the class and vice-versa).

One possible solution is to characterize each vertex of the given graph G by a vector with n components and then to define an equivalence relation of vectors (and thus obtain an equivalence relation of vertices).

Definition 8.2.2. *Given a graph $G = (V, E)$, a characterization function $f : V \rightarrow \mathbb{R}^n$ and a relation $r \in \mathbb{R}^n \times \mathbb{R}^n$, two vertices $u, v \in V$ are said to be r -equivalent if and only if one has $(f(u), f(v)) \in r$.*

If one takes for instance, for each vertex v of the graph G , $f(v)$ to be the k -pattern vector of v and r to be the identity, one has that two vertices u and v of G are r -equivalent if and only if they are k -pattern equivalent.

In order to define a r -equivalence on the vertices of a graph G , one has to give a definition of characterization function f and of relation r . Here, we base our definition of characterization function on 4-pattern vectors. As for the relation r , we define it using a clustering method that we introduce.

Characterization function. Given a large graph G , we obtain a description of each one of its vertices by analyzing its egocentred network. Thus each node is characterized by a vector, the k -pattern vector, containing the number of occurrences of the different patterns in its egocentred network. In this section we use only patterns with at most 4 vertices (Figure 8.1). They provide a detailed enough image of how the node is connected to the network while being not very numerous. We add two more elements, the number of isolated vertices and the number of isolated edges in the egocentred network, to the 4-pattern vector. We thus define a new vector characterizing each vertex, called pattern-frequency vector.

Definition 8.2.3. *Given a graph $G = (V, E)$, we call pattern-frequency function the characterization function $f : V \rightarrow \mathbb{R}^{11}$ such that for all $v \in V$ one has*

$$f(v) = (f_{iv}(v), f_{ie}(v), f_{\dashv}(v), f_{\perp}(v), f_{\triangle}(v), f_{\sqsubset}(v), f_{\lrcorner}(v), f_{\sqsupset}(v), f_{\square}(v), f_{\boxminus}(v), f_{\boxplus}(v))$$

where:

- $f_{iv}(v)$ is the number of isolated vertices in the egocentred network $Eg(v)$,
- $f_{ie}(v)$ is the number of isolated edges

and the subsequent components are the numbers of occurrences of the patterns as induced subgraphs in the egocentred network $Eg(v)$ of v :

- $f_{\dashv}(v)$, pattern 1, edges,
- $f_{\perp}(v)$, pattern 2, paths with 2 vertices,
- $f_{\triangle}(v)$, pattern 3, triangles,
- $f_{\sqsubset}(v)$, pattern 4, paths with 3 vertices,
- $f_{\lrcorner}(v)$, pattern 5, stars,
- $f_{\sqsupset}(v)$, pattern 6,
- $f_{\square}(v)$, pattern 7, chordless squares,
- $f_{\boxminus}(v)$, pattern 8, squares with one chord,
- $f_{\boxplus}(v)$, pattern 9, 4-cliques.

We call the vector $f(v)$ the pattern-frequency vector of v .

For instance, for the vertex in Figure 8.2(a), the egocentred network is represented in Figure 8.2(b) and the number of occurrences of the different patterns in Figure 8.2(c); its pattern-frequency vector is then $f(v) = (4, 1, 6, 3, 1, 2, 0, 1, 0, 0, 0)$. Note that the pattern-frequency vector of the vertex v can also be seen as a characterizing vector of its egocentred network $Eg(v)$.

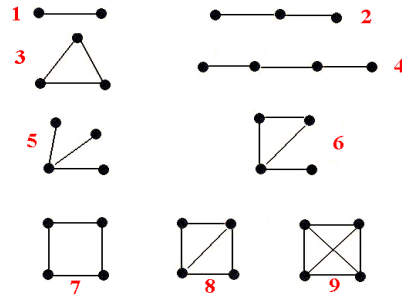


Figure 8.1: The 9 patterns with at most 4 vertices and at least one edge.

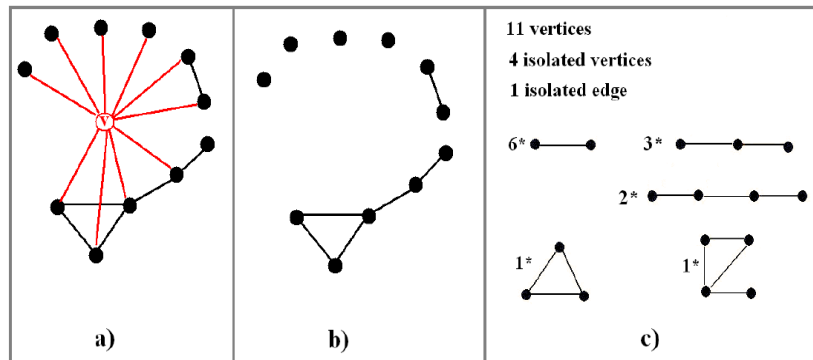


Figure 8.2: A vertex v and its neighbors (a), the egocentred network $Eg(v)$ of v (b) and the patterns of $Eg(v)$ (c).

Vector relation. We want to define a relation r on the vectors characterizing the vertices of the given graph G . We choose to define a clustering of vertices based on the pattern-frequency vectors that we have previously introduced; we call this clustering *pattern-frequency clustering*. Then, the relation r is defined such that its equivalence classes are the clusters produced by the pattern-frequency clustering.

Definition 8.2.4. *We call pattern-frequency equivalence on the vertices of a graph, the r -equivalence whose equivalence classes are the clusters built by the pattern-frequency clustering.*

So we want to define a clustering of nodes based on pattern-frequency vectors. Of course, this clustering must correspond to our main goal of grouping together vertices that are connected in a similar way to the network. We use a classical clustering method, the k-means (presented in Section 2.2). The advantage of performing a clustering to define vertex equivalence is its flexibility: one can distribute the vertices into a small number of clusters (if this is his goal) or a large number of clusters (where vertices in the same cluster are very similar to each other).

Before performing the clustering, we filter out vertices that have identical pattern-frequency vectors. These vertices are not distinguishable by using only the patterns; their egocentred networks contain exactly the same patterns in exactly the same number. By default, they belong to the same cluster. The elimination of multiple copies of the same pattern-frequency vector insures a smaller complexity of computation and also allows us to perform a finer clustering. Of course, after having clustered the remaining vertices (we call them the reduced population), we put the filtered out vertices into the clusters where the vertices with identical vectors have been already placed.

Definition 8.2.5. *Given a graph G , we call reduced population of G a maximal set of vertices of G that have distinct pattern-frequency vectors. Given a positive integer d , we denote by $Pop_d(G)$ the set of vertices in the reduced population of G that have degree d (in G).*

8.2.2 The issue of the degree

There is an important factor that must be taken into consideration before doing the clustering: the degree of vertices. It is difficult to compare the number of occurrences of patterns in egocentred networks of vertices with different degrees because these values are biased by the degree. For vertices with high degrees, the number of occurrences can have high values, too. Actually, for a vertex (ego) with degree d , a pattern with k vertices can occur at most $\binom{d}{k}$ times in its egocentred network. So, while the minimal value of the number of occurrences of a pattern is always 0, the maximal value depends on the degree of ego. Therefore, the exact values of the number of occurrences of patterns can be misleading. Look, for instance, at the four egocentred networks in Figure 8.3 (ego has been removed). Their pattern-frequency vectors are presented in Table 8.1 where one can see that the values of many variables are higher for C and D than for A and B . Even more, the networks C and D look more similar to each other than A and B , so the vectors

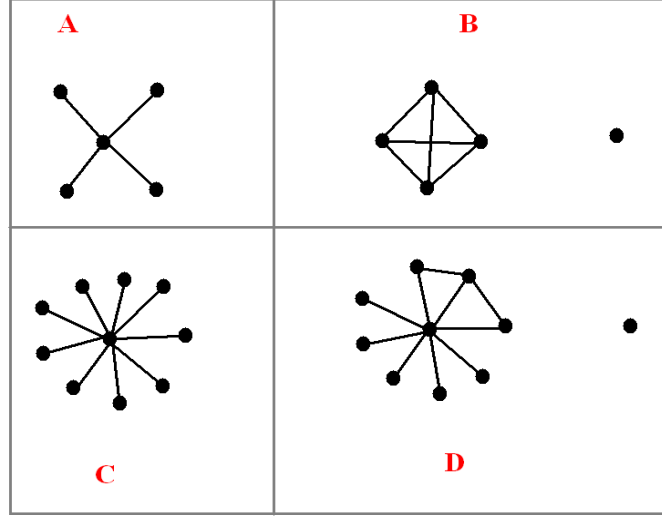


Figure 8.3: An example of 4 egocentred networks with 5 vertices (*A* and *B*) and 10 vertices (*C* and *D*) respectively (ego has been removed).

Table 8.1: The pattern-frequency vectors of the egocentred networks in Figure 8.3.

net.	f_{deg}	f_{iv}	f_{ev}	f_{\dashv}	f_{\perp}	f_{\triangle}	f_{\sqsubset}	f_{\lrcorner}	f_{\sqsupset}	f_{\sqcap}	f_{\sqcup}	f_{\boxtimes}
A	5	0	0	4	6	0	0	4	0	0	0	0
B	5	1	0	6	0	4	0	0	0	0	0	1
C	10	0	0	9	36	0	0	84	0	0	0	0
D	10	1	0	10	26	2	0	45	10	0	1	0

of *C* and *D* should be closer to each other than those of *A* and *B*. However, the Euclidian distance between the pattern-frequency vectors is 74 for *A* and *B* and 1726 for *C* and *D*.

In order to avoid the problem of the degree, we choose to perform a clustering for each degree. Thus, the distance between the vertices *C* and *D* in the previous example will be compared to the distances between other pairs of vertices of degree 10 and not to all the input vertices. If we manage to group together the vertices of each degree in a same number of clusters and to match together the clusters obtained for the different degrees, then we have that each cluster contains vertices of all the degrees. This is exactly our goal here: we want a vertex to belong to a given cluster because it has a certain type of connection to the network and not because it has a certain degree. Thus, if a vertex gets another degree during time, we can see if the type of structure in which it is connected also changes by checking if its cluster changes. It is not the difference of degree that we want to capture but the difference of structure. If we don't have exactly the same clusters for all the degrees, we cannot do this. And this is exactly what might happen if we perform a single clustering for all the degrees (and not for each degree separately): there might be clusters with no vertices of some degrees (because, for instance, there are fewer vertices of that degree).

8.2.3 Pattern-frequency clustering of nodes

We proceed as it follows:

1. for each degree, we perform several k-means clusterings (see Section 2.2 for a description of this method) on the vertices with that degree in the reduced population, using different numbers of clusters; we compute the best number of clusters;
2. we keep as final number of clusters the number indicated as best for most degrees; let this number be n_c ;
3. for each degree, we divide the vertices with that degree into n_c clusters;
4. we finally match the clusters found for the different degrees.

Let us explain the different steps.

STEP 1. Given that we base our clustering on occurrences of patterns with 4 vertices and less, we cluster only vertices with degree at least 4. For each degree, we use the k-means algorithm on modified versions of the pattern-frequency vectors of the nodes. As k-means starts by randomly picking the first centers, we perform 50 clusterings for each degree and each number of clusters and choose the clustering with the lowest intra-cluster variance. The best number of clusters is computed by comparing the average silhouette values obtained for the different numbers of clusters (see Section 2.2 for a presentation of this technique).

Let us explain why and how we modify the pattern-frequency vectors. The k-means algorithm uses a given distance between elements in order to compute the clusters; this distance is usually the Euclidian distance between the feature vectors of the elements. We need to modify the pattern-frequency vectors before computing the Euclidian distance on them. There are several reasons for that.

a) Modifying the ranges of values. Even if we focus on each degree at a time, the numbers of occurrences of the different patterns are not placed in the same ranges of values. For instance, the maximal number of occurrences of the \llcorner -pattern is generally a lot higher than the maximal number of the \boxtimes -pattern. We need to place the ranges of values of all the variables participating to the Euclidian distance between the same extreme values. This can be done for instance by centering and scaling the variables or by giving them new values, obtained from a computation of slices. It is the second solution that we adopt here.

Generally, given a group of n elements that have values a_1, a_2, \dots, a_n for a given attribute (or variable) a , one can compute k bins (or slices) such that there is a fairly equivalent number of elements whose values are placed in each bin. For that, one needs to compute $k + 1$ ascendant values (called limits) such that the first limit is the minimal value of a_i for $i \in \{1, 2, \dots, n\}$, the last limit is the maximal value of a_i and there is a fairly equivalent number of elements (i.e. $\frac{n}{k}$) whose values are placed between two consecutive limits. Now, one can use instead of the values a_1, a_2, \dots, a_n the corresponding slices: instead of the value a_i one uses the value x if a_i belongs to the x -th bin. Note that the computation of only

two bins ($k = 2$) is equivalent to the computation of the median value of the attribute a . In this case, one can use, instead of the real value a_i of the attribute, a value that is either 1 or 2 depending on a_i : if a_i is inferior to the median value, then one uses 1, otherwise 2.

This is the technique that we apply here. Instead of using the real values of the pattern-frequency vectors, we compute and use slices of values. There are several advantages in doing this. First, we eliminate the problem of comparing very different values for different patterns: now we have, for all the patterns, the same possible values. Second, the new values are established using the ranges of values, as found in the network. Thus, the number of occurrences of a given pattern in a given egocentred network can be very small comparing to the maximal possible value and, in the same time, very high comparing to its value in the other egocentred networks. We want to emphasize the fact that this value is high in *our* network, which the slices do. Thirdly, the extreme values (often difficult to handle) are simply put in the marginal slices and are no longer seen as extreme.

For each degree d and each one of the 11 components of the pattern-frequency vector, we choose 5 bins such that an equivalent number of nodes in Pop_d (the reduced population with degree d) have values in each one of the bins.

b) Using the absent patterns. By using the pattern-frequency vectors we take into consideration the presence of different structures in the egocentred networks. Besides this, it can be useful to take into consideration also the absence of different structures. Thus, two nodes are similar if they have many common patterns in their egocentred networks, but also if patterns that are not present in one are not present in the other one either. To take this information into consideration, we add to the pattern-frequency vector of each node the pattern-frequency vector of the complement graph of its egocentred network. Recall that the complement graph of a graph $G = (V, E)$ is a graph $G' = (V', E')$ where the vertices are the same as in G (i.e. $V' = V$) and the edges are all the possible edges between vertices in V that are not present in E (i.e. $E' = \{(u, v), u, v \in V \text{ and } (u, v) \notin E\}$). We thus have, for each vertex v , a vector containing the number of occurrences of patterns in the egocentred network $Eg(v)$, followed by the number of occurrences of patterns in the complement graph $Eg'(v)$ of the egocentred network. Next we replace the real values in this new vector by the corresponding slices as previously explained; we thus obtain the *extended pattern-frequency vector*.

Definition 8.2.6. *Given a vertex v of a graph G , we call extended pattern-frequency vector of v the vector with 22 components containing first the slice values of the pattern-frequency vector of v and then the slice values of the pattern-frequency vector of the complement graph $Eg'(v)$ of the egocentred network $Eg(v)$ of v .*

It is on the extended pattern-frequency vectors that we compute the Euclidian distance and we perform the k-means clustering.

STEP 3. Suppose n_c was found as best number of clusters for most degrees, so we need to divide the nodes with each degree in the reduced population into n_c clusters. We perform again 50 k-means clusterings with $k = n_c$ for each degree and we keep the clustering with the lowest intra-cluster variance.

STEP 4. We have now n_c clusters for each degree greater than 3. We need to match the clusters obtained for the different degrees so that, every node, no matter its degree, belongs to one of the n_c clusters. In order to do the matching, we compute the center (or centroid) of each cluster for each degree. Recall that the center of a cluster is the average of all the points in the cluster i.e. a vector where each component is the arithmetic mean of the values of that component for all the elements in the cluster.

We match clusters for consecutive degrees by using the centers: for each degree $d > 4$, we compute the centers of the clusters obtained for d (let C_i be the center of the i th cluster, with i from 1 to n_c) and for $d-1$ (let C'_i be the center of the i th cluster, with i from 1 to n_c) and the Euclidean distances between these centers. For each one of the clusters obtained for degree d we have to choose exactly one cluster from those obtained for degree $d-1$, and each one of these clusters must be chosen exactly once. This corresponds to a permutation of n_c elements: each cluster with index 1 to n_c obtained for degree d is given a new index, also from 1 to n_c , corresponding to the cluster for degree $d-1$ with which it is matched. We choose the permutation σ that minimizes the sum of distances between centers of matched clusters: $\sum_{i=1, \dots, n_c} \text{dist}(C_i, C'_{\sigma(i)})$. For that, let us observe that if there is a valid permutation σ such that, for all i from 1 to n_c , $\text{dist}(C_i, C'_{\sigma(i)})$ is the minimum distance between C_i and any C'_j , with j from 1 to n_c , then σ is the permutation that minimizes the sum of distances. This case may occur for many pairs of consecutive degrees, so in this case no other computation is needed. After having computed the permutation σ that minimizes the sum of distances, one has a bijective matching of clusters for the given pair of consecutive degrees. By doing this for each pair, we obtain a matching of all the clusters.

Each vertex in the reduced population thus belongs to one of the n_c clusters. We now distribute into clusters the vertices that we have previously filtered out by putting them in the clusters of the vertices with the same pattern-frequency vector.

8.3 Clusters of individuals in the mobile phone network

Using the previously described technique, we cluster the individuals in the mobile phone communication network (the same graph as in Chapter 7). The best number of clusters is found to be 6. Figure 8.4 represents the distribution into clusters of the egocentred networks of vertices with degree 4 (up) and 5 (bottom). In our graph, all the possible egocentred networks for these degrees are present; these are all the possible undirected graphs with 4 and 5 vertices respectively. For each network, we have written in red the cluster to which it belongs.

We observe that cluster 1 contains dense networks, while cluster 6 contains very sparse networks. Networks in cluster 2 seem to have a high number of stars, while those in cluster 5 have both isolated vertices and a rather dense group. For clusters 3 and 4 we can say that networks in cluster 3 are denser than those in cluster 4. These observations have been made by simply analyzing the clusters obtained for degree 4 and 5. When looking at the centers of the clusters obtained for the different degrees, we observe that, for all degree:

- the center of cluster 1 has the maximal value for the number of edges and for the

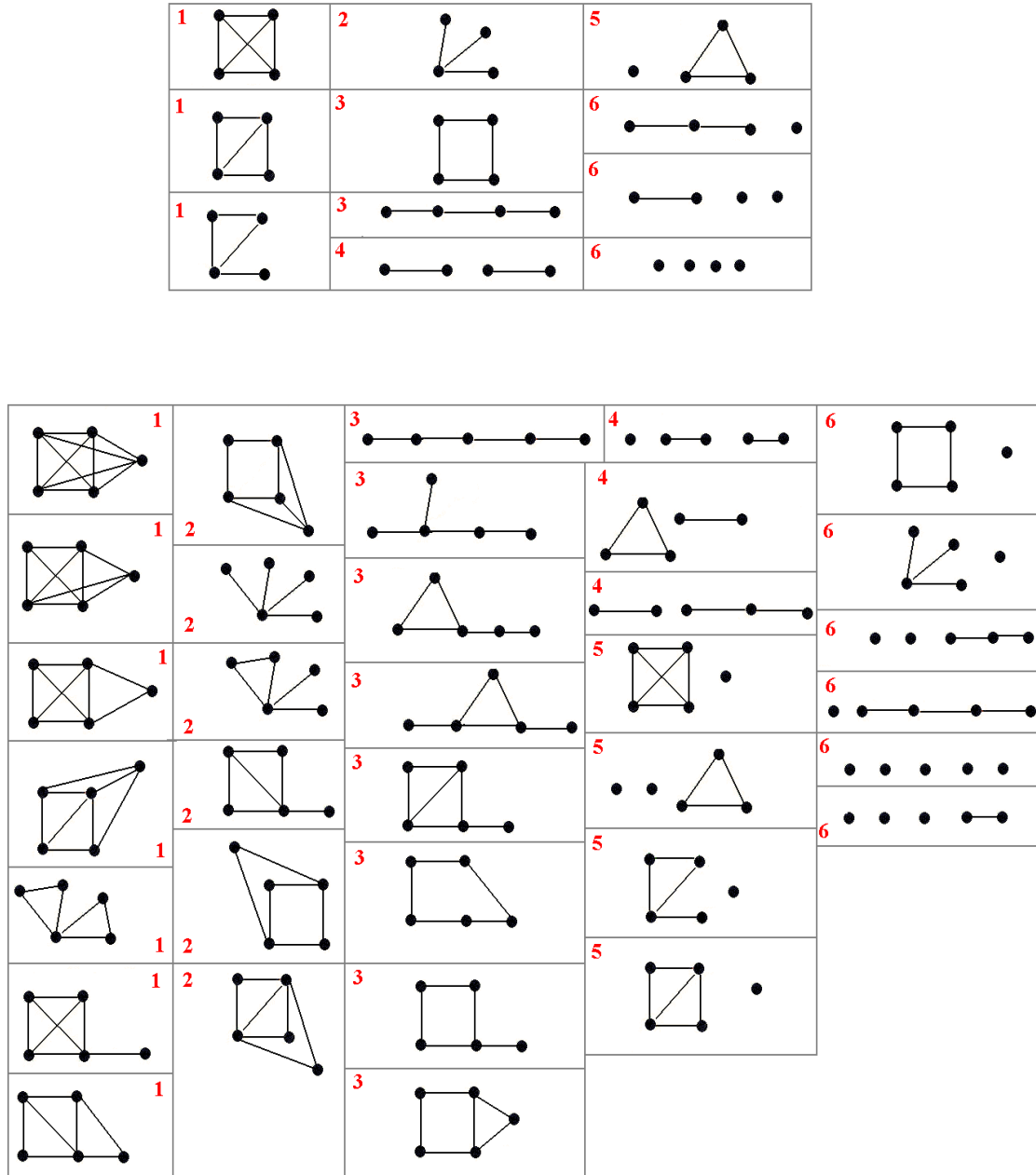


Figure 8.4: All the possible egocentred networks of vertices with degree 4 (up) and 5 (bottom) and their clusters.

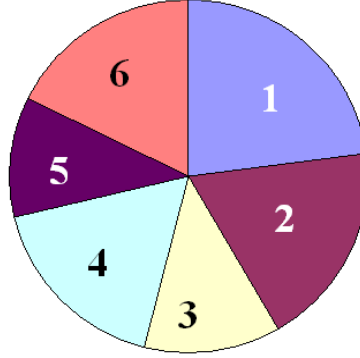


Figure 8.5: The distribution of the reduced population into the 6 clusters.

number of triangles i.e. vertices in cluster 1 have the highest average of $f_{\text{—}}$ and of f_{Δ} ;

- the opposite situation happens for cluster 6 : the center of this cluster has the minimal value for the number of edges and for the number of triangles i.e. vertices in cluster 6 have the lowest average of $f_{\text{—}}$ and of f_{Δ} ;
- from the remaining clusters, the center of cluster 5 has the maximal value for the number of isolated vertices multiplied by the number of edges i.e. vertices in cluster 5 have the highest average of $f_{iv} \times f_{\text{—}}$;
- the center of cluster 2 has the maximal value for the number of stars i.e. vertices in cluster 2 have the highest average of $f_{\text{—}}$;
- from the remaining two clusters, the center of cluster 3 has a higher value for the number of edges than the center of cluster 4 i.e. vertices in cluster 3 have a higher average of $f_{\text{—}}$ than vertices in cluster 4.

This sustains our previously made observations for degrees 4 and 5 : cluster 1 contains the densest networks, while cluster 6 contains the sparsest ones. Networks in cluster 2 have many stars, while those in cluster 5 have both isolated vertices and a dense group. Finally, networks in cluster 3 are denser than those in cluster 4.

Remember that before computing the clusters we have eliminated the multiple copies of pattern-frequency vectors. It is in this reduced population that we have computed the 6 clusters. The different resulting clusters contain fairly similar percentages of the reduced population (see Figure 8.5 and Table 8.2).

However, when reintroducing the filtered out vertices, the population is not equally divided into clusters any more. This is caused by the low local density of the graph: most vertices have very sparse egocentred networks, so the different patterns occur in their networks in small number. Thus the majority of the eliminated vertices belongs to cluster 6. After the introduction of the previously filtered out vertices, the new repartition into clusters becomes very unbalanced (Table 8.2).

Table 8.2: The distribution of the reduced and total population into the 6 clusters.

cluster	% of the reduced population	% of the total population
1	23.16	4.15
2	18.6	2.91
3	12.24	2.54
4	17.05	26.93
5	11.12	5.04
6	17.83	58.43

In the following sections we confront the identified clusters to other characteristics of the mobile phone customers.

8.4 Clusters versus age and gender

8.4.1 Age

For the mobile phone customers who have provided their birth year when subscribing to the studied operator, we want to see if there is a connection between the age of a person and his cluster. Remember that in Section 7.2 we presented some statistics on mobile phone use. There are some differences in call frequency and duration between ages, but the main distinction concerns SMS usage, the younger users sending a lot more SMS than the older ones. Here we want to see if these differences in mobile phone uses are visible in the structure of the network surrounding each person.

We compute, for each cluster k from 1 to 6 and for each age a from 18 to 55¹, the probability that a person of age a who has at least 4 contacts belongs to cluster k :

$$P(a, k) = \frac{\text{nb. persons of age } a \text{ and cluster } k}{\text{nb. persons of age } a \text{ and degree } > 3}$$

The plot of these probabilities is presented in Figure 8.6. We observe that middle age people (30 to 45) have the lowest probability of belonging to cluster 1, so generally they are not involved in dense structures. This can be seen also in the plot for cluster 6 (the cluster containing the sparsest networks), where there is a peak for 35 to 40. Younger people belong generally to clusters 2, 3 and 4 and rarely to cluster 6 (in any case, a lot less frequently than older people). The oldest people are generally placed in cluster 5: there is an increasing probability of having a densely connected group and some isolated contacts when going from 40 years old to 55.

Let us now group together the ages that have similar probabilities for the 6 clusters. We perform a hierarchical clustering on the ages using the cluster probabilities previously computed, after having centered and scaled the probabilities so that they have the same

¹18 is the minimal age to have a mobile phone subscription, while for persons of more than 55 years old, 70% of them belong to cluster 7

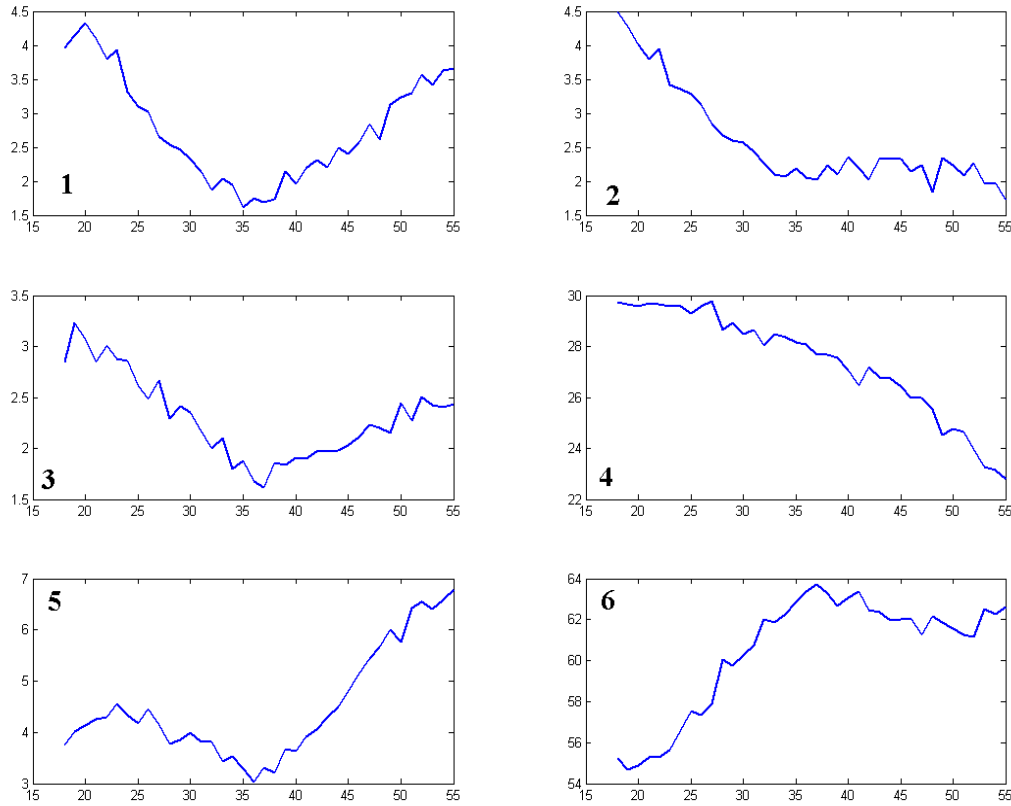


Figure 8.6: For each cluster (each image), the probability of belonging to that cluster by age (on x-axis).

mean and standard deviation for each profile. The result of this analysis is shown in Figure 8.7. We observe that there are 4 principal, homogeneous age groups similar to life stages categories: 19 – 23 (who can be associated with "students"), 24 – 27 (young people starting their active life), 28 – 48 (the age of living in couple, often with children), and 49 – 55 (people at an advanced stage of the professional life, whose children are adult or living apart). Note that this classification is based exclusively on structural characteristics of the local communication network where the degree was neutralized.

To sum up, there are some differences in the mobile phone usage and in the network structure depending on the age. Therefore a good question is: do these differences exist because with age we change our mobile phone uses or because people of different ages started using the mobile phone at a different age? As the mobile phone appeared in the 1990s', the younger persons in our database had a mobile phone from an early age, while the other persons started to use it when they were already adults. So, do the youngest people send a lot of SMS because they were used to have a mobile phone since an early age or because they are young? It would be interesting to analyze the generation effect on mobile phone uses in 50 years, when everybody would have had a mobile phone since

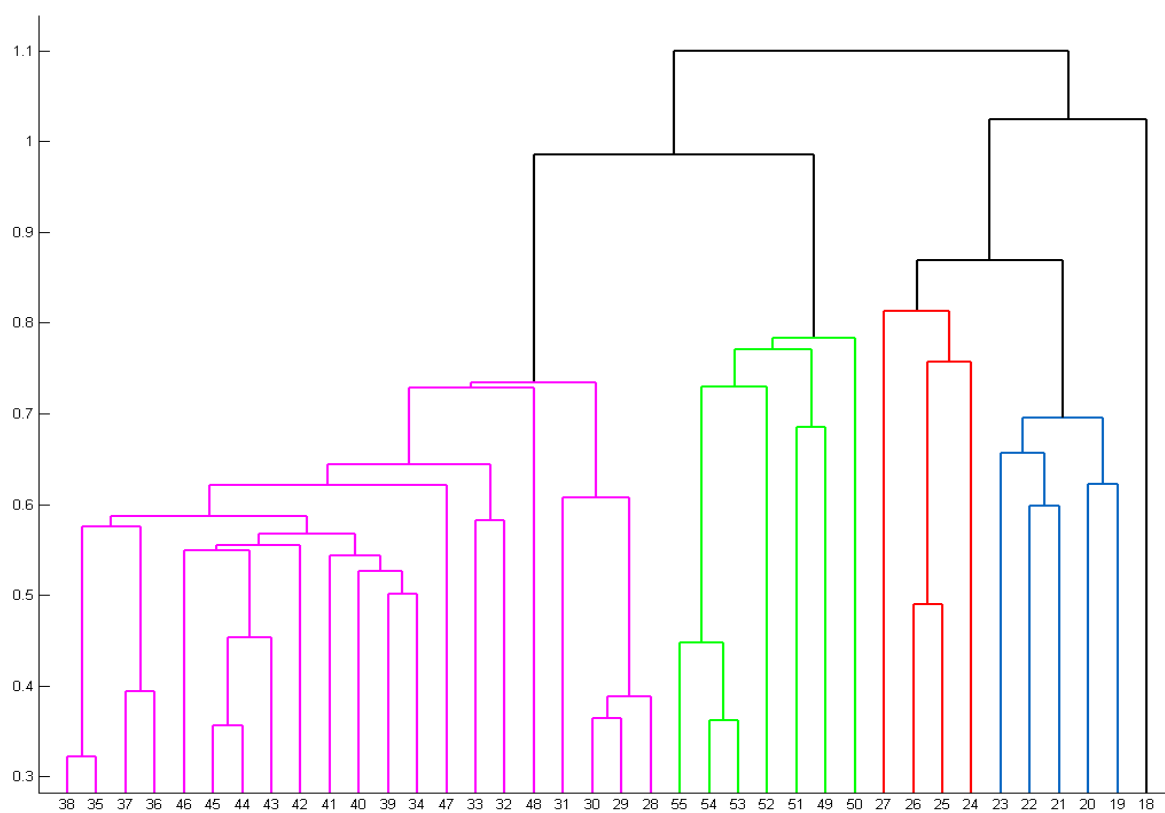


Figure 8.7: Hierarchical clustering of ages on probabilities of belonging to the 6 clusters.

Table 8.3: The proportion of men and women in each cluster.

cluster	% men	% women
1	51.35	48.65
2	48.32	51.68
3	48.68	51.32
4	47.73	52.27
5	49.98	50.02
6	48.18	51.82

a young age.

8.4.2 Gender

We compute, for each one of the 6 clusters, the probability that a person belonging to that cluster is a man. We obtain the proportions in Table 8.3. There is no important difference between the values obtained for women and men: in each cluster there are almost as many men as women. Nevertheless, a χ^2 test rejects the hypothesis that the genders and the clusters are independent (i.e. the probability that a person belongs to a given cluster is not independent from the person's gender) with $p < 0.005$. This, however, is not surprising: given the large amount of data on which the hypothesis is verified, the test tends to reject it easily.

8.5 Clusters versus intensity of communication

8.5.1 Basic statistics

We compute for each person (ego) the total number of calls he had during the followed period (both in-coming and out-going calls), the total duration of his calls and the total number of SMS (similarly, in-coming and out-going SMS). Also, we compute the average number of calls, total duration and number of SMS he had with each one of his contacts. We limit the contacts to the persons who initiated at least one communication (call or SMS) with ego and who also received at least one call or SMS from ego; these persons correspond to ego's neighbors in our graph. Besides the average values, we also compute the standard deviation for the number of calls, the duration and the number of SMS per contact. We thus have for each ego a vector with 9 variables characterizing ego's communications. We use these vectors to measure the relation between communication intensity and the previously obtained clusters.

We begin by testing, for each one of the 9 variables, the independence of the variable and the clusters by performing an ANOVA test: we test the hypothesis that the mean value of the variable is the same for the different clusters. As the distributions for the 9 components are heavily right-skewed, we use the log values instead of the real ones. The ANOVA test rejects the hypothesis of equal means for each one of the components with $p = 0$. However, the ANOVA test specifies just that the means are different (i.e. they are

not all equal) but does not say for which pairs of clusters these means are significantly different and for which they are not. In order to find this information, we perform a Bonferroni multi-comparison test for each one of the 9 variables. We thus have:

- for the total number of calls, all the means are significantly different, except for the clusters 1 and 2; the order of the mean values of the total number of calls for the 6 clusters is, from low to high: 6, 4, 5, 3, 2, 1;
- similarly, for the total duration of calls and the total number of SMS, all the means are significantly different, except for the clusters 1 and 2; in this case the order is 6, 5, 4, 3, 1, 2;
- very similar results are obtained for the other variables; the ascending order of the values is always 6, 4, 5, 3, 2, 1, maybe with an interchange of 4 and 5 and of 1 and 2; the average duration of calls per contact is the only variable for which there isn't a significant difference between the mean values for the 6 clusters.

So, for each one of the 9 components, cluster 6 has the lowest mean, followed by clusters 5 and 4 (or 4 and 5), cluster 3 and finally 2 and 1 (or 1 and 2). However, using the mean values isn't satisfying as the different variables have a right-skewed distribution. Therefore, for each variable, we compute 10 slices as we did in Section 8.2.3: we divide its spectrum of values into 10 slices or bins such that a fairly equal number of values belong to each one of the bins. Then, we compute the probability that an individual belonging to a given cluster has values in a certain bin:

$$P(\text{variable}, \text{cluster}, \text{bin}) = \frac{\#\text{individuals} \in \text{cluster s.t. value}(\text{variable}) \in \text{bin}}{\#\text{individuals} \in \text{cluster}}.$$

We plot these probabilities for the first 3 variables in Figure 8.8: the number of calls in (a), the total duration of calls in (b) and the number of SMS in (c). Each bar corresponds to a bin, going from the bin with the lowest values (dark blue) to the bin with the highest ones (dark red). For each cluster, the height of each bin represents the previously computed probability i.e. the probability that an individual in that cluster has values in that bin; the sum of heights of bins of one cluster is thus equal to 1. For the three variables, individuals in clusters 1, 2 and 3 have a greater probability to have values in the highest bins than in the lowest ones, while for cluster 6 the opposite situation happens. Cluster 4 has values especially in the intermediate bins, while cluster 5 has values both in high and low bins, but fewer in the intermediate ones.

8.5.2 Predicting the cluster from the communications

Given these differences in quantity of communications for the different clusters, we want to see if we can guess in which cluster an individual is placed given his communications. For that, we use a decision tree to unfold the relation between communication intensity and cluster and thus to predict the cluster of each individual (see Section 2.2 for an introduction to decision trees). The explanatory variables are the 9 characterizing the

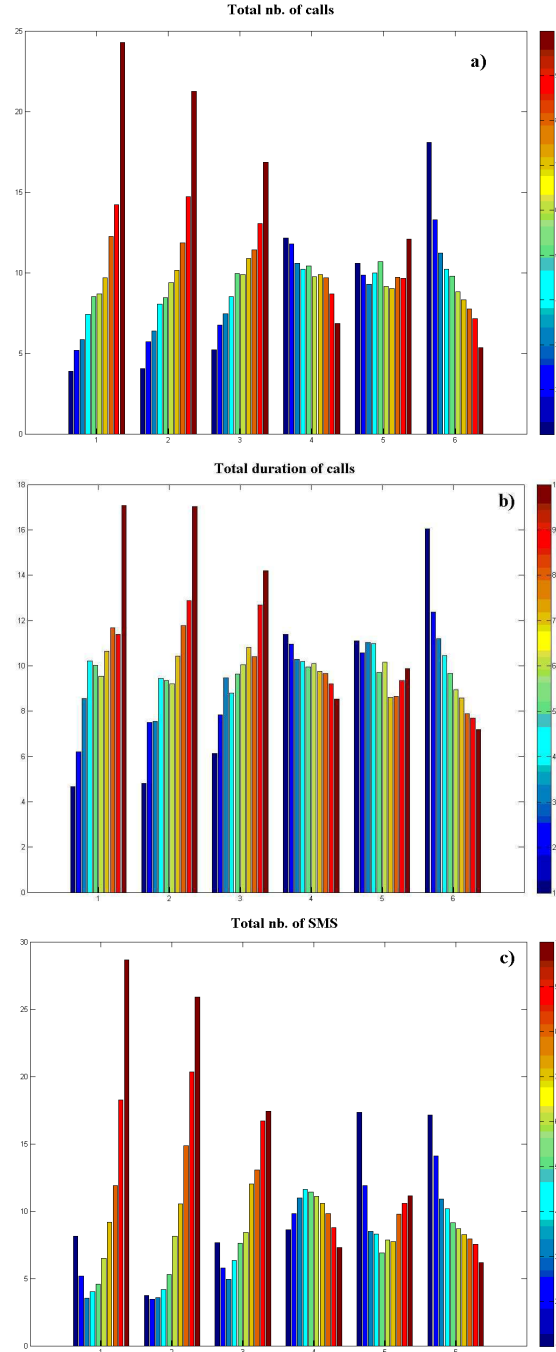


Figure 8.8: For each cluster (Ox-axis), the probability that the communications of an individual in that cluster are in a given slice of values of the number of calls (a), total duration of calls (b) and number of SMS (c).

Table 8.4: The proportion of correct predictions in the 6 clusters.

cluster	rate of success
1	31.2%
2	22.6%
3	24.3%
4	40.4%
5	51.8%
6	37.1%

communications of an individual: the number of calls, the total duration of calls, the number of SMS, the average number of calls, duration and number of SMS per contact, and the standard deviation of the number of calls, duration and number of SMS per contact. Based on the learning population, the tree learns the associations between intensity of communication and cluster; then it predicts the cluster of the individuals in the test population. If the predicted cluster is the same with the real cluster of the person, then the prediction is correct; otherwise the prediction is false. To measure the accuracy of the tree, one counts the correct predictions as compared to the size of the test population: the higher this number, the better the prediction. This number is then compared to the random prediction, where one attributes individuals into clusters randomly, with an equal probability.

Remember that the number of individuals in the 6 clusters is very uneven, with cluster 6 over-represented. If the decision tree learns and tests its rules of association on populations with such uneven distribution of clusters, it will associate everybody with cluster 6 : no matter the communication characteristics of the different persons, if everybody is put in cluster 6, the tree gives the correct class to all the individuals in cluster 6 and the wrong cluster to all the others. As the individuals in cluster 6 are much more numerous than the others, the tree has a high rate of success. We want to avoid this situation and impose to the tree to search for associations between communications and clusters. Therefore, we give it a learning population where there is an equal number of individuals belonging to each cluster; the individuals are randomly chosen from the individuals in each cluster. We do the same thing for the test population. As we want to predict 6 clusters, the rate of success of the random prediction is $\frac{100}{6} = 16.66\%$. Our decision tree has a rate of success of 34.6%, so more than twice than the random one. The rate of correct predictions in the different clusters is presented in Table 8.4.

This result shows that there is a correlation between the intensity of communication and the cluster to which an individual belongs. Even more, we are able to predict the cluster with a rather high accuracy (as compared to the random prediction) given a set of variables characterizing the communications of each person.

8.6 A typology of customers

In the previous two sections we compared the social network clusters first to customers' age and gender, and then to their communication intensity. We thus saw that the probability that an individual belongs to a given cluster is not independent from his age or communication intensity.

Here we want to take into consideration, in the same time, all the 3 dimensions characterizing the individuals: the age, the communication intensity and the social network cluster². We want to see how these characteristics are distributed in the population and also to create a typology of customers based on these 3 dimensions. We would thus obtain groups of individuals such that the persons in a same group have similar communication practices and about the same age and cluster.

We use the Kohonen self organizing map in the same way as in Chapter 6. Remember that this clustering method produces a map with several layers, one for each variable characterizing the individuals. This shows how the different variables are distributed in the population. Also, the algorithm produces cells grouping individuals with close characteristics. In a second step, the algorithm computes a clustering of the individuals. The obtained clustering will represent our typology.

We choose the following parameters to characterize the individuals:

- age; this is the socio-demographic variable;
- cluster (from 1 to 6, as obtained in the previous sections); as it takes only 6 values, this variable can be seen as a class or a label of each individual; this is the social network variable;
- communication intensity: number of calls, total duration of calls and number of SMS; these are the communication variables.

Each individual is thus characterized by a vector with 5 elements. For the communication variables, we use a log transformation instead of the values themselves as these variables are heavily right-skewed. Also, recall that the distribution of individuals into clusters is very uneven, with cluster 6 being overrepresented. As we want to measure the influence of the variable "cluster", too, we randomly choose a same number of individuals in each cluster.

The set of individuals is then processed by the Kohonen self organizing map. This algorithm does not take labels into consideration when building the map, so it builds the map using only the other variables. However, in the graphic representation of the map, it draws a layer for the labels, too. On this layer the different cells are colored depending on the labels of the individuals in the cells: the color of the cell corresponds to the label that occurs the most for the individuals in that cell.

The processing of the set of individuals by the SOM provides Figure 8.9. We observe that, unsurprisingly, the number of calls and the total duration are highly correlated, with

²We do not take into consideration the gender because its influence on the cluster is not very strong; besides, this variable takes only 2 values

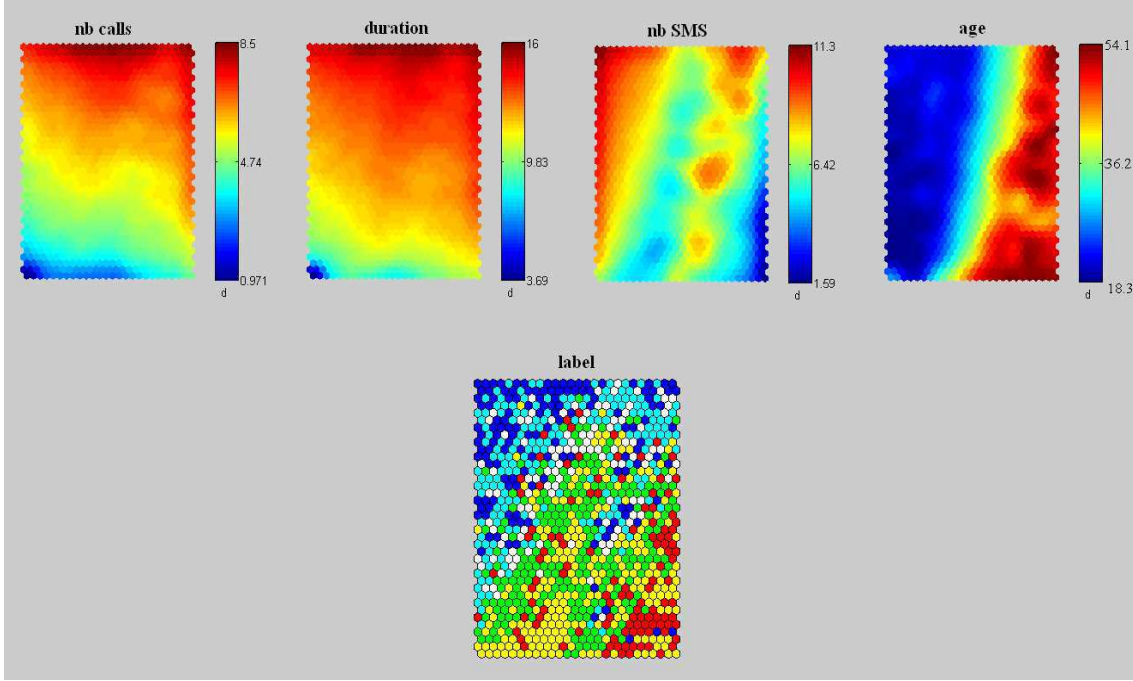


Figure 8.9: SOM results: the individuals are grouped into cells depending on their communication intensity and age; the label represents the social cluster: 1(blue), 2(cyan), 3(white), 4(green), 5(red), 6(yellow).

increasing values on the south-north axis: the individuals with the lowest number of calls and total duration are placed in the south part of the map, while those with the highest values are placed in the north part. The number of SMS, however, is not correlated to the two previous ones, its values increasing from east to west. This variable seems to be correlated to the age: the highest values of the number of SMS are in the west part, where the youngest people are placed, while the lowest values are placed in the east part, where the oldest persons are placed. All these observations sustain our previous ones, presented in Section 7.2: there is no influence of the age on the call frequency and duration, but there is a high influence on the number of SMS.

Let us now analyze the distribution of the variable "cluster" in the different cells, so the last image in Figure 8.9. Figure 8.10 shows the same distribution, cluster by cluster. Thus, each image in this figure corresponds to a cluster: the red cells contain mostly individuals of the given cluster, while the white cells contain mostly individuals of other clusters. Recall that the different clusters are not taken into consideration when building the map; the cells are colored depending on the clusters of the people present in the cell, after all the computations. We observe that clusters 1, 2 and 3 are present especially in the north-west side of the map, while clusters 4, 5 and 6 are placed especially in the south-east side. Most of the cells labeled cluster 1 contain individuals with very high

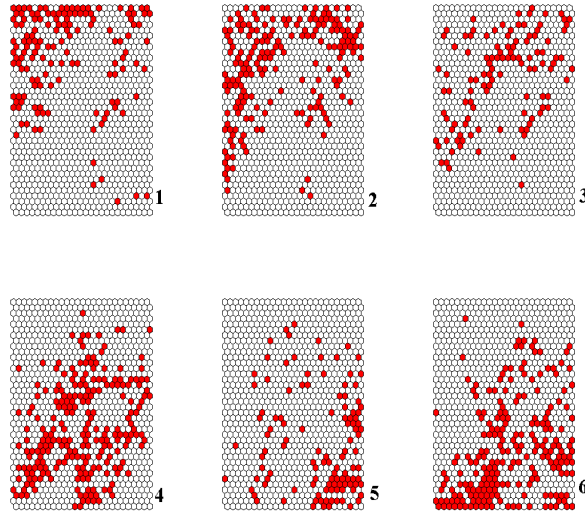


Figure 8.10: For each cluster (each image), the cells where the cluster is in the majority (the red cells).

number of SMS or very high number of calls and total duration (dark red cells in the first 3 layers). Cluster 2 is generally associated with cells containing individuals with a high number of SMS or a high number of calls and total duration (orange to red cells in the first 3 layers). Clusters 3 and 4 are generally present in cells where the individuals have a medium number of calls, total duration and number of SMS. Cluster 6 is especially placed in the south-east part of the area, where there are individuals with low numbers of calls, total duration and number of SMS (the blue cells in the first 3 layers). There seems to be no clear relation between the label of the cell and the average age of the persons in the cell, except for cluster 5 which is present especially in the cells containing the oldest people (dark red cells in the fourth layer).

As in Chapter 6, we cluster the cells using the k-means algorithm. We thus obtain 9 profiles, as showed in Figure 8.11. We present the different characteristics of the people with each profile in Table 8.5. This result represents a typology of individuals based on their age, communication intensity and social network cluster.

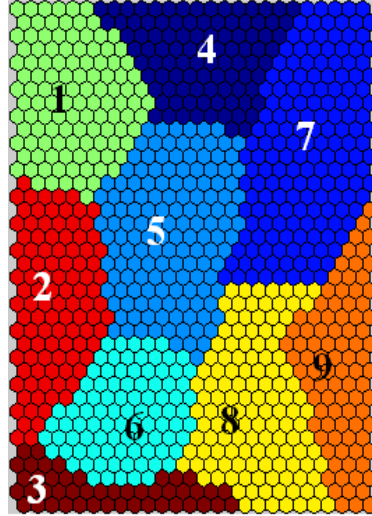


Figure 8.11: The 9 profiles produced by the Kohonen SOM.

Table 8.5: The different characteristics of the individuals in the 9 profiles produced by the SOM.

profile	age	nb. calls & duration	nb. SMS	most represented cluster(s)
1 green	youngest	high	very high	1(45%), 2(41%)
2 red	youngest	medium	high	2(38%), 3(20%)
3 brown	youngest-middle	very low	low	6(70%)
4 dark blue	youngest-middle	very high	medium	1(31%), 2(31%)
5 light blue	youngest-middle	medium-high	low	4(39%), 6(24%)
6 cyan	youngest-middle	low	low	4(45%), 6(43%)
7 blue	oldest	high	high	2(29%), 1(19%)
8 yellow	oldest	low	low	4(34%), 6(29%)
9 orange	oldest	low	very low	5(42%), 6(35%)

8.7 Chapter conclusions

In this chapter we continued the analysis of the mobile phone graph with a clustering of nodes, thus relating to the problem of identification of roles in a network. In this problem often encountered in social network analysis, one wants to group together the nodes of the network that are connected in similar ways to the network. There are however several questions that make this problem difficult to solve: What is a good characterization of the way a node is connected to the network? What does "similar connections" mean? Can the solution be applied to large graphs? How can one check the relevance of the different groups of nodes? In which conditions can one say that there is no better way of grouping the nodes?

We have made several choices in order to answer the different questions. First, we have characterized the way a node is connected to the network by counting the patterns present in its egocentred network; we have stored the number of occurrences of the different patterns in a pattern-frequency vector characterizing the node. Second, we have considered that nodes connected in a similar way to the network have close pattern-frequency vectors; here "close" is defined with respect to a set of transformations made on the pattern-frequency vectors. We have thus proposed a method for nodes clustering that groups together vertices that are embedded in similar egocentred networks. The clustering is done efficiently, so the method can be applied to large graphs. As said before, we have made several choices in order to answer the different questions. The proposed method gives promising results when applied to our real-world graph. As always, in this kind of methods, the solution validation is a delicate problem, but the results we have obtained for our large social network sustain the relevance of our method.

We have applied the proposed method to the mobile phone graph described in the previous chapter. This graph models one-month mobile phone communications between the 3 million customers of Mobistar. The clusters produced by the method can be seen as a segmentation of the set of customers based on their social network insertions. We have compared the different clusters to the other information we had on the individuals (age, gender and communication intensity), showing that the different parameters characterizing the individuals are not independent. Thus, the probability that a node belongs to a given cluster is not independent from the age, gender or mobile phone use of the person represented by the node. These results confirm the soundness of our method, even though, as always, many concurrent clusterings for various purposes may as well be relevant.

Part III

Conclusions

The main goal of our research was to characterize the individuals connected in a social network by analyzing the local structure of the network. For that, we proposed a method that describes the way a node (corresponding to an individual) is embedded in the network. This method provides a characterization of the individual and also of the relative positions occupied by the neighbors of the node in its egocentred network (which can also be seen as a description of how the links formed by a node are embedded in the network). Our method is related to the analysis of egocentred networks in sociology and to the local approach in the study of complex networks. As it takes into consideration only the surrounding network when analyzing one node, it can be applied to small networks, to fractions of networks (one does not need the entire network when analyzing one node) and also to large networks; this is due to its rather small complexity, depending only on the number of neighbors of the node. Although in this thesis we applied the method only to social networks, it can be applied in the same way to any other graph, no matter its origin.

We applied the method we introduced to two large social networks, one modeling online activity on MySpace (a platform for social networking and video publishing), the other one modeling mobile phone communications. In the first case we were interested in analyzing the online popularity of artists on MySpace. We first grouped individuals into clusters using their popularity characteristics (mainly their online audience and authority), thus obtaining 5 clusters. Besides two unsurprising categories (very popular artists and unknown artists), we identified two different clusters of medium popularity and a category of small but socially active artists. Next we compared the obtained clusters to the local structure of the network surrounding each node, so we analyzed the popularity of artists in relation with the structure of the network in which they are embedded. We thus showed that artists in different categories exhibit different insertions in the social network. On the one hand, artists with a low authority and non reciprocal links tend to declare very popular artists as best friends thus generating a star structure. On the other hand, some medium and low popularity artists with many reciprocal links form cliques with their neighbors, thus creating dense communities, without stars but with triangles. Our research on MySpace belongs to the analysis of popularity on online networks, where researchers try to discover how fame is built, what strategies users employ, how they adapt their publishing and networking practices in order to be popular. There are many studies on this competition for online popularity, but they focus either on published content and its popularity, either on the structure of the social network embedding the users. Here we tried to hold together the two approaches, so to make the connection between fame and social linkage. The same kind of analysis, using the methods we employed here, can be done on other online platforms where the popularity can be measured and the social network can be built. An immediate transposition can be imagined for Flickr and Youtube for instance. In the same way, one can also study offline networks for which there are recordings of users' activity, as for instance a mobile phone communications network. It is such a network that we analyzed next, but in a different approach.

We used the list of one-month communications between 3 million mobile phone users. We were interested in three aspects that we tried to compare: social-demographic data (users' age and gender), communication intensity (for each couple of persons, their num-

ber of calls, duration of calls and number of SMS) and social network structure. First we confirmed, using these large amounts of data, some existing sociological theories on communication duration depending on receiver's gender and on young people's tendency to send SMS. Next, by applying the method introduced previously, we analyzed the local structure of the social network modeling the set of mobile phone communications. The results of our method gave us the possibility to test several definitions of characteristic patterns, thus relating to two popular problems in data mining and bioinformatics: the frequent patterns discovery and the network motifs identification. Next, we analyzed the positions occupied by the neighbors of each node (ego) and we compared them to the quantity of communication with ego. We thus saw that the person that speaks the most with ego has an important position in his egocentred network. If this result seems intuitive, it isn't necessarily straightforward if there is the person who speaks the most as number of calls or the one who speaks the most as total duration of calls who has the most important position. In our dataset, it is the person with the highest frequency of calls who has a more important place in the egocentred network; this result is sustained by an existing sociological study on patterns of communication.

In our opinion, the next logical step of our analysis was to group together nodes with similar egocentred networks, so connected in the same way to the network. This is the problem of identification of roles in a network. Without pretending to have solved this problem, we proposed a method for grouping the nodes of a large network that we applied to the mobile phone social network. One of the main problems when trying to identify the roles played by the different nodes of the network is the results validation. In small networks one can simply look at the different nodes and decide if the attributed roles correspond to the structure of the network surrounding each node. Of course one cannot do this in large networks. This is why we cannot pretend having solved the problem of identifying social roles. We simply proposed a way to cluster nodes depending on the local structure of the network; there may be other clusterings with more satisfying attributions of roles of nodes. However, the results obtained when applying the method to the mobile phone social network are quite promising. We compared the 6 clusters of mobile phone users identified by the method to the two other dimensions characterizing the individuals: the socio-demographic data and the intensity of communication. A first observation is that belonging to a certain cluster is not independent from users' age, gender and intensity of communication. Even more, by using the distribution of persons of different ages into clusters, we were able to identify 4 homogeneous age groups, corresponding to life stages. And this using only the way the nodes are connected to the network, independently from the number of neighbors of each node. Next, we were able to predict with a rather high probability the cluster of each person using his communication intensity, thus showing that local structure and communication intensity are correlated. Persons embedded in dense structures seem to communicate more by mobile phone than persons belonging to sparse networks. These results make us believe in the relevance of our method for nodes clustering. This method can be easily applied to any large network; it will cluster nodes depending on the structure of the network surrounding them. It is important to precise that our method groups together nodes that are connected in the same way when comparing to the other nodes of the network and not to a theoretical situation or to nodes

of other networks. Thus a node belongs to a certain cluster because it is similar to the other nodes of the cluster and different from the other nodes of the network, of this precise network. The clustering of nodes depends entirely on the structure of the given network, as it is in this network that we want to find groups of nodes.

These are the main conclusions of our work during this thesis. A lot of extensions and improvements can be imagined; we present some of them in the following section.

Further work

As said before, the set of methods we used here can be easily applied to other (social) networks. Using the local structures in which nodes are embedded, it would be interesting for instance to compute clusters of individuals in social networks that are denser than the mobile phone one but sparser than the MySpace one. In the mobile phone network, many links are missing: maybe two persons contact each other by different means, but not by mobile phone. Thus, in our graph, we see the two persons as not connected and we analyze them in consequence, although they do connect, but by means that are not visible to us. On the contrary, in social networks modeling online activities, there are many links that do not correspond to a real, social relation between the two persons. As we saw in the study on MySpace popularity, people connect to other people they do not know, just because they are popular, creating thus a fan-star structure. Such social networks are therefore denser than the "real" social network where each link corresponds to a social relation between the two persons. For the mobile phone graph, we search for clusters of individuals in a network that is sparser than the real one, while for the online networks, we search in a graph that is denser than the real network. It would be thus interesting to analyze a graph with a density between the two.

In another perspective, the method itself could be improved. For instance one could characterize the way each node is connected to the network by analyzing the network at at most 2 (or more) steps from the node i.e. the network formed by ego's neighbors, their neighbors and the links between all these nodes. However, the computation and results complexity might increase a lot. Also one would have to deal with the distinction between direct neighbors and distance-2 neighbors. Maybe an easier way to take into consideration the distance-2 neighborhood is to analyze in more details how different individuals (with different local characteristics) connect to each other. One can see the global network as the union of many egocentred networks that partially overlap. It would be interesting to see how and why they overlap, for which type of egocentred network, in which proportion etc.

Another improvement can be done by taking into consideration the weight of links when computing patterns. For instance, for the mobile phone social network, one can put a weight on the links using the frequency of calls and their duration or the frequency of SMS. Then, instead of characterizing a node by the number of patterns present in its egocentred network, one could characterize it using the number of weighted patterns. The weight of the pattern can be for instance the following couple: average weight of links in the pattern, standard deviation of weight of links in the pattern. One has thus an idea of

the quantity of information that flows in the pattern and also of its distribution (balanced or not). Characterizing nodes by counting weighted patterns would offer a more detailed description of how each node is connected to the network.

Also one could give a weight or simply a color to the nodes of the network (instead of the links). This can be done at the global level, by coloring each node of the network depending on some statistics, at the local level, by coloring each node in the egocentred network depending on its relation with ego, or at the intermediate level, for instance by coloring the nodes depending on the community they belong to. Then, instead of simply counting patterns, one could count patterns with different combinations of colors on their nodes. Maybe such an approach would provide a better definition of characteristic patterns.

Another possible direction is to take into consideration the temporal dynamics of the network. For instance one could try to predict the cluster to which a node will belong in a second network (obtained some time after the first network) by using the cluster of the node (and maybe other characteristics) in the first network. Or one could predict some events (like the formation or the deletion of links) by using the way the nodes are embedded into the network. Maybe the fact of having many stars or triangles etc. in the egocentred network says something about the capacity of a node of adding new links or of losing existing ones. One could also describe how the different nodes evolve in time by computing their cluster in different snapshots of the same network. From this description one could compute patterns of evolution or see if the way of changing the cluster is related to other information on the nodes. For instance in the mobile phone social network one can see if the individuals remain in the same cluster from one period to the other, if they change, how they change and how their evolution is related to their age or gender. In an even more precise approach one could analyze the local structure of the network dynamically i.e. when each event happens (as for instance the formation or deletion of links).

Bibliography

- [AA04] Istvan Albert and Reka Albert. Conserved network motifs allow protein-protein interaction prediction. *Bioinformatics*, 20(18):3346–3352, 2004.
- [ABA03] Lada A. Adamic, Orkut Buyukkokten, and Eytan Adar. A social network caught in the web. *First Monday*, 8(6), 2003.
- [AD07] B. Bassett et J. Hoskins Allsop D. Word-of-mouth research: principles and applications. *Journal of Advertising Research*, 34:398–411, 2007.
- [AGMN92] Noga Alon, Zvi Galil, Oded Margalit, and Moni Naor. Witnesses for boolean matrix multiplication and for shortest paths. In *FOCS*, pages 417–426. IEEE, 1992.
- [AH01] Lada A. Adamic and Bernardo A. Huberman. The web’s hidden order. *Communications of the ACM*, 44(9):55–60, September 2001.
- [AJB99] R. Albert, H. Jeong, and A. L. Barabási. Diameter of the world-wide web. *Nature (London)*, 401(6749):130, 1999.
- [And06] D. Anderson. *The Long Tail: How the Future of Business is Selling Less of More*. Hyperion Books; New York edition, 2006.
- [ARFBTS04] Y. Artzy-Randrup, S.J. Fleishman, N. Ben-Tal, and L. Stone. Comment on ”network motifs: Simple building blocks of complex networks” and ”superfamilies of evolved and designed networks”. *Science*, 305(5687):1107, August 2004.
- [AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.
- [ASBS00] L. A. N. Amaral, A. Scala, M. Barthélemy, and H. E. Stanley. Classes of small-world networks. In *Proceeding of the National Academy of Sciences*, 2000.
- [ATSK04] M.J Aartsen, T.G. Van Tilburg, C.H.M Smits, and C.P.M. Knipscheer. A longitudinal study on the impact of physical and cognitive decline on the personal network in old age. *Journal of Social and Personal Relationships*, 21:249–266, 2004.

- [AYZ97] Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
- [AZBA08] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 665–674, New York, NY, USA, 2008. ACM.
- [BA99] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- [Bar54] John A. Barnes. Class and committees in a Norwegian island parish. *Human Relations*, 7:39–58, 1954.
- [Bar02] Albert-László Barabási. *Linked: The New Science of Networks*. Perseus Books Group, 2002.
- [BC09] J.S Beuscart and T. Couronne. The distribution of online reputation. In *ICWSM'09*, May 2009.
- [BCP02] K.K. Bost, M.J. Cox, and C. Payne. Structural and supportive changes in couples' family and friendship networks across the transition to parenthood. *Journal of Marriage and Family*, 64:517–531, 2002.
- [BDE09] Murat Ali Bayir, Murat Demirbas, and Nathan Eagle. Discovering spatiotemporal mobility profiles of cellphone users. In *WOWMOM*, pages 1–9. IEEE, 2009.
- [BE89] S.P. Borgatti and M.G. Everett. The class of all regular equivalences: Algebraic structure and computation. *Social Networks*, 11(1):65–88, March 1989.
- [BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, (10):P10008 (12pp), 2008.
- [BHKL06] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM Press.
- [BJN⁺02] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3-4):590 – 614, 2002.
- [Boi74] J. Boissevain. *Friends of Friends, Networks, Manipulators and Coalitions*. Basil Blackwell, Oxford, 1974.

- [Bon87] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.
- [Bot57] Elizabeth Bott. *Family and Social Network*. Tavistock, London, 1957.
- [BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
- [Bra01] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.
- [BT02] Tian Bu and Donald F. Towsley. On distinguishing between internet power law topology generators. In *INFOCOM*, 2002.
- [Bur92] R. Burt. *Structural Holes. The Social Structure of Competition*. Cambridge, Harvard University Press, 1992.
- [Bur01] Ronald S. Burt. Structural holes versus network closure as social capital. In Nan Lin, Karen S. Cook, and Ronald S. Burt, editors, *Social Capital: Theory and Research*, pages 31–56. Aldine de Gruyter, New York, 2001.
- [CF06] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys (CSUR)*, 38(1), 2006.
- [CG05] Dominique Cardon and Fabien Granjon. Social networks and cultural practices a case study of young avid screen users in france. *Social Networks*, 27:301–315, 2005.
- [CKR⁺07] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world’s largest user generated content video system. In *IMC’07*, 2007.
- [CLR01] T. H. Cormen, C. E. Leiserson, and T. L. Rivest. *Introduction to Algorithms*. The MIT Press, 2001.
- [CM06] Judith A. Chevalier and Dina Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, August 2006.
- [CMG09] Meeyoung Cha, Alan Mislove, and P. Krishna Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *WWW*, pages 721–730, 2009.
- [CMN04] Aaron Clauset, Cristopher Moore, and M.E.J. Newman. Finding community structure in very large networks. *Physical Review E*, 70(6), 2004.
- [Col88] J.S. Coleman. Social capital in the creation of human capital. *American journal of sociology*, 94(S1):95, 1988.

- [CR04] Junghoo Cho and Sourashis Roy. Impact of search engines on page popularity. In *Proceedings of the 13th conference on World Wide Web*, 2004.
- [CS08] Riley Crane and D. Sornette. Quality, and junk videos on youtube: Separating content from noise in an information-rich environment. In *The 2008 AAAI Spring Symposium*, 2008.
- [CSB10] T. Couronné, A. Stoica, and J.S. Beuscart. Online social network popularity evolution: an additive mixture model. In *The 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2010.
- [CSN07] Aaron Clauset, Cosma R. Shalizi, and M.E.J. Newman. Power-law distributions in empirical data. *SIAM Reviews*, June 2007.
- [CW87] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. In *STOC*, pages 1–6. ACM, 1987.
- [DR01] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001.
- [DT99] Luc Dehaspe and Hannu Toivonen. Discovery of frequent DATALOG patterns. *Data Mining and Knowledge Discovery*, 3(1):7–36, 1999.
- [DWM02] Jennifer A. Dunne, Richard J. Williams, and Neo D. Martinez. Food-web structure and network theory: The role of connectance and size. *PNAS*, 99(20):12917–12922, 2002.
- [EB93] M.G. Everett and S.P. Borgatti. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361–376, 1993.
- [EBK69] J. Engel, R. Blackwell, and R. Kegerreis. How information is used to adopt an innovation. *Journal of Advertising Research*, 9:3–8, 1969.
- [EP05] Nathan Eagle and Alex Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 4(2):p28 – 34, 2005.
- [ER60] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Science*, pages 17–61, 1960.
- [Eve80] Brian Everitt. *Cluster Analysis*. Halsted Press, London; New York, second edition, 1980.
- [EW04] David Eppstein and Joseph Wang. Fast approximation of centrality. *Journal of Graph Algorithms Applications*, 8(1):39–45, 2004.

- [FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, New York, NY, USA, 1999.
- [FLG00] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Sixth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2000)*, pages 150–160. ACM, 2000.
- [FM91] Toms Feder and Rajeev Motwani. Clique partitions, graph compression, and speeding-up algorithms. In *STOC*, pages 123–133. ACM, 1991.
- [For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.
- [FPSSU96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.
- [Fra95] Robert H. Frank. *The winner-take-all society*. Free Press, New York, 1995.
- [Fre77] Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, 1977.
- [FS65] S. Feldman and M. Spencer. The effect of personal influence in the selection of consumer services. In *Proceedings of the Fall Conference of the American Marketing Association*, 1965.
- [FSW06] Danyel Fisher, Marc Smith, and Howard T. Welser. You are who you talk to: Detecting roles in usenet newsgroups. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.
- [Gal67] T. Gallai. Transitiv orientierbarbare graphen. *Acta Mathematica Hungarica*, 18(1-2):25–66, 1967.
- [GDS⁺03] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 314–329, New York, NY, USA, 2003. ACM.
- [GH06] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.
- [GHB08] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

- [Gla00] Malcolm Gladwell. *The tipping point: how little things can make a big difference*. Little Brown, Boston, 1st edition, 2000.
- [GN02] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*, 2002.
- [GR03] D.S. Goldberg and F.P. Roth. Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences*, 100(8):4372–4376, 2003.
- [Gra78] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, pages 1360–1380, 1978.
- [Gri98] Maurizio Gribaudo. *Espaces, temporalités, stratifications. Exercices sur les réseaux sociaux*. EHESS, Paris, 1998.
- [Gro05] Michel Grossetti. Where do social relations come from?: A study of personal networks in the Toulouse area of France. *Social Networks*, 27(4):289 – 300, 2005.
- [Hal08] A. Halavais. Do dugg diggers digg diligently. In *AOIR’08*, 2008.
- [Hay05] Caroline Haythornthwaite. Social networks and internet connectivity effects. *Information, Communication and Society*, 8(2):125–147, June 2005.
- [HBB07] T. Holloway, M. Bozicevic, and K. Borner. Analyzing and visualizing the semantic coverage of wikipedia and its authors. *Complexity*, 12(3):30–40, 2007.
- [HCD94] Lawrence B. Holder, Diane J. Cook, and Surnjani Djoko. Substructure discovery in the subdue system. In *KDD Workshop*, pages 169–180, 1994.
- [HEL04] Petter Holme, Christofer R. Edling, and Fredrik Liljeros. Structure and time-evolution of an internet dating community. *Social Networks*, 26(2):155–174, 2004.
- [HK79] Frank Harary and Helene J. Kimmel. Matrix measures for transitivity and balance. *Journal of Mathematical Sociology*, 6:199–210, 1979.
- [HKP⁺05] Susan C. Herring, Inna Kouper, John C. Paolillo, Lois Ann Scheidt, Michael Tyworth, Peter Welsch, Elijah Wright, and Ning Yu. Conversations in the blogosphere: An analysis ”from the bottom up”. In *HICSS’05*, 2005.
- [HM79] Michel Habib and M. C. Maurer. On the x-join decomposition for undirected graphs. *Discrete Applied Mathematics*, 3:198–207, 1979.
- [HP57] Frank Harary and Herbert H. Paper. Toward a general calculus of phonemic distribution. *Language : Journal of the Linguistic Society of America*, 33:143–169, 1957.

- [HRS08] Cesar A. Hidalgo and C. Rodriguez-Sickert. The dynamics of a mobile phone network. *Physica A: Statistical Mechanics and its Applications*, 387(12):3017–3024, May 2008.
- [HRW08] Bernardo A. Huberman, Daniel M. Romero, and Fang Wu. Crowdsourcing, attention and productivity. *CoRR*, abs/0809.3030, 2008.
- [HS] H. He and A.K. Singh. Graphrank: Statistical modeling and mining of significant subgraphs in the feature space. In *ICDM '06*.
- [HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [ILK⁺05] Shalev Itzkovitz, Reuven Levitt, Nadav Kashtan, Ron Milo, Michael Itzkovitz, and Uri Alon. Coarse-graining and self-dissimilarity of complex networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 71(1), 2005.
- [IR78] Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.
- [IWM00] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 13–23, London, UK, 2000. Springer-Verlag.
- [JBLE01] Jeffrey C. Johnson, Stephen P. Borgatti, Joseph J. Luczkovich, and Martin G. Everett. Network role analysis in the study of food webs: An application of regular role coloration. *Journal of Social Structure*, 2(3), 2001.
- [JMBO01] H. Jeong, S.P. Mason, A.-L. Barabási, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411, 2001.
- [KIMA04] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.
- [KK01] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 313–320, Washington, DC, USA, 2001. IEEE Computer Society.
- [KKM00] Ton Kloks, Dieter Kratsch, and Haiko Mller. Finding and counting small induced subgraphs efficiently. *Inf. Process. Lett.*, 74(3-4):115–121, 2000.
- [KKR⁺99] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: measurements, models and methods. In *Proceedings of the 5th Annual International Computing and Combinatorics Conference (COCOON)*, volume 1627 of *Lecture Notes in Computer Science*, pages 1–18, Tokyo, Japan, 1999. Springer.

- [KL55] Elihu Katz and Paul Lazarsfeld. *Personal Influence: The Part Played by People in the Flow of Mass Communications*. Glencoe:the Free Press, 1955.
- [Kle00] Jon Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170, New York, NY, USA, 2000. ACM.
- [KNT06] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *KDD '06*, August 2006.
- [Koh90] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, 1990.
- [KR90] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data An Introduction to Cluster Analysis*. Wiley Interscience, New York, 1990.
- [KRRT99] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. In *Proceedings of the 25th VLDB Conference*. 1999.
- [Lat08] Matthieu Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theor. Comput. Sci.*, 407(1-3):458–473, 2008.
- [LBdK⁺08] R. Lambiotte, V.D. Blondel, C. de Kerchove, E. Huens, C. Prieur, Z. Smoreda, and P. Van Dooren. Geographical dispersal of mobile communication networks. *Physica A*, 387(21):5317–5325, September 2008.
- [LKF05] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, New York, NY, USA, 2005. ACM Press.
- [LS05] Christian Licoppe and Zbigniew Smoreda. Are social networks technologically embedded? How networks are changing today with changes in communication technology. *Social Networks*, 27(4):317–335, October 2005.
- [LW71] F. Lorrain and H. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.
- [LY05] R. Ling and B. Yttri. *Control, emancipation and status: The mobile telephone in the teen's parental and peer group control relationships*. Oxford, 2005.
- [MAA08] Marcelo Maia, Jussara Almeida, and Virgílio Almeida. Identifying user behavior in online social networks. In *SocialNets '08: Proceedings of the 1st workshop on Social network systems*, pages 1–6, New York, NY, USA, 2008. ACM.

- [MBSA02] Stefano Mossa, Marc Barthelemy, Eugene H. Stanley, and Luis A. Amaral. Truncation of power law behavior in scale-free network models due to information filtering. *Physical Review Letters*, 88(13), 2002.
- [MCN97] D. Morgan, P. Carder, and M. Neal. Are Some Relationships more Useful than Others? The Value of Similar Others in the Networks of Recent Widows. *Journal of Social and Personal Relationships*, 14(6):745–759, 1997.
- [Mer68] Robert K. Merton. The Matthew Effect in Science. *Science*, 159(3810):56–63, 1968.
- [MF01] Alan L. Montgomery and Christos Faloutsos. Identifying web browsing trends and patterns. *IEEE Computer*, 34(7):94–95, 2001.
- [MIK⁺04] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, and Uri Alon. Response to Comment on "Network Motifs: Simple Building Blocks of Complex Networks" and "Superfamilies of Evolved and Designed Networks". *Science*, 305(5687):1107d–, 2004.
- [Mil67] Stanley Milgram. The small world problem. *Psychology Today*, 1:61, 1967.
- [MiOO⁺01] Satoru Morita, Ken ichi Oshio, Yuko Osana, Yasuhiro Funabashi, Kotaro Oka, and Kiyoshi Kawamura. Geometrical structure of the neuronal network of *Caenorhabditis elegans*. *Physica A: Statistical Mechanics and its Applications*, 298(3-4):553–561, September 2001.
- [Mit69] J. Clyde Mitchell. *Social networks in urban situations: Analysis of personal relationships in central African towns*. Manchester University Press, Manchester, 1969.
- [Mit04] Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251, 2004.
- [MKFV06] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. *SIGCOMM Comput. Commun. Rev.*, 36(4):135–146, 2006.
- [MKG⁺08] Alan Mislove, Hema Swetha Koppula, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Growth of the flickr social network. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 25–30, New York, NY, USA, 2008. ACM.
- [MLH08] Clmence Magnien, Matthieu Latapy, and Michel Habib. Fast computation of empirically tight bounds for the diameter of massive graphs. *ACM Journal of Experimental Algorithmics*, 13, 2008.
- [MMG⁺07] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC'07*, October 2007.

- [Mon01] Alan L. Montgomery. Applying quantitative marketing techniques to the internet. *Interfaces*, 30:90–108, 2001.
- [MSOI⁺02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, October 2002.
- [Nad57] SF Nadel. *The Theory of Social Structure*. Cohen and West, London, 1957.
- [New03] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 167(45), 2003.
- [New04] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, 2004.
- [New05] M.E.J. Newman. Power laws, pareto distributions and zipf’s law. *Contemporary Physics*, 46:323–351, 2005.
- [New06] M.E.J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582, 2006.
- [NFB02] M.E.J. Newman, Stephanie Forrest, and Justin Balthrop. Email networks and the spread of computer viruses. *Phys. Rev. E*, 66(3):035101, September 2002.
- [NG04] M.E.J. Newman and M Girvan. Finding and evaluating community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(2):026113.1–15, 2004.
- [NK01] Siegfried Nijssen and Joost Kok. Faster association rules for multiple relations. In *IJCAI’01: Proceedings of the 17th international joint conference on Artificial intelligence*, pages 891–896, 2001.
- [OSH⁺07a] J. P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, A.M. de Menezes, K. Kaski, A.L. Barabási, and J. Kertész. Analysis of a large-scale weighted network of one-to-one human communication. *New Journal of Physics*, 9(6):179+, June 2007.
- [OSH⁺07b] J. P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A. L. Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, May 2007.
- [PBV07] Gergely Palla, Albert-Laszlo Barabasi, and Tamas Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, April 2007.

- [PCB⁺08] Christophe Prieur, Dominique Cardon, Jean-Samuel Beuscart, Nicolas Pissard, and Pascal Pons. The strength of weak cooperation: A case study on flickr. *CoRR*, abs/0802.2317, 2008.
- [PCM10] Claudio Imbrenda Leonardo Lanzi Pierluigi Crescenzi, Roberto Grossi and Andrea Marino. Finding the Diameter in Large Graphs: Experimentally turning a lower bound into an upper bound. In *18th Annual European Symposium on Algorithms*, 2010.
- [Pea01] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Phil. Mag.*, 6(2):559–572, 1901.
- [PGF02] Christopher R. Palmer, Phillip B. Gibbons, and Christos Faloutsos. ANF: a fast and scalable tool for data mining in massive graphs. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, New York, NY, USA, 2002. ACM Press.
- [Prz06] Natasa Przulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):177–183, 2006.
- [PSE83] M.R. Parks, C.M. Stan, and L.L. Eggert. Romantic involvement and social network. *Social Psychology Quarterly*, 46(2):116–131, 1983.
- [PSS09] C. Prieur, A. Stoica, and Z. Smoerda. Extraction de réseaux égocentrés dans un (très grand) réseau social. *Bulletin de méthodologie sociologique*, (101):5–27, 2009.
- [RB40] A.R. Radcliffe-Brown. On social structure. *Journal of the Royal Anthropological Institute*, 70:1–12, 1940.
- [Red98] S. Redner. How popular is your paper? An empirical study of the citation distribution. *European Physical Journal B*, 4:131–134, 1998.
- [RSM⁺02] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555, August 2002.
- [RZ02] Paul Resnick and Richard Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay’s reputation system. In Michael R. Baye, editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Elsevier Science, 2002.
- [Sai78] L. Sailer. Structural equivalence: Meaning and definition, computation and application. *Social Networks*, 4:117–145, 1978.
- [Sas02] K. Sassenberg. Common bond and common identity groups on the internet: Attachment and normative behavior in on-topic and off-topic chats. *Group Dynamics*, 6(1):27–37, 2002.

- [SCB10] A. Stoica, T. Couronné, and J.S. Beuscart. To be a star is not only metaphoric: from popularity to social linkage. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*. AAAI, 2010.
- [SEE92] R. Simon, D. Eder, and C. Evans. The development of feeling norms underlying romantic love among adolescent females. *Social Psychology Quarterly*, 55:29–46, 1992.
- [Sei92] Raimund Seidel. On the all-pairs-shortest-path problem. In *STOC*, pages 745–749. ACM, 1992.
- [SGS⁺02] O. Shefi, I. Golding, R. Segev, E. Ben-Jacob, and A. Ayali. Morphological characterization of in vitro neuronal networks. *Physical Review E*, 2002.
- [SH08] Gábor Szabó and Bernardo A. Huberman. Predicting the popularity of online content. *CoRR*, abs/0811.0405, 2008.
- [Sim55a] G. Simmel. *Conflict and the Web of Group Affiliations*. Free Press, New York, 1955.
- [Sim55b] H. Simon. On a class of skew distribution functions. *Biometrika*, 42(3-4):425–440, 1955.
- [SL00] Zbigniew Smoreda and Christian Licoppe. Gender-specific use of the domestic telephone. *Social Psychology Quarterly*, 63:238–252, 2000.
- [SOMMA02] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:1061–1036, 2002.
- [SP02] K. Sassenberg and T. Postmes. Cognitive and strategic processes in small groups: Effects of anonymity of the self and anonymity of the group on social influence. *British Journal of Social Psychology*, 41:463–480, 2002.
- [SP09a] A. Stoica and C. Prieur. Structure of ego-centered networks in very large social networks. In *The XXIX International Social Network Conference (Sunbelt)*, 2009.
- [SP09b] A. Stoica and C. Prieur. Structure of neighborhoods in a large social network. In *Proceedings of the 2009 IEEE International Conference on Social Computing (SocialCom)*, pages 26–33. IEEE Computer Society, 2009.
- [SQBB10] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Laszlo Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

- [SSPG10] A. Stoica, Z. Smoreda, C. Prieur, and J.L. Guillaume. Age, gender and communication networks. In *NetMob, Workshop on the Analysis of Mobile Phone Networks*, 2010.
- [Sur88] C.A Surra. *The Effects of the Interactive Network on Developing Relationships*. Newbury Park, CA: Sage Publications, 1988.
- [SW05] Thomas Schank and Dorothea Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. volume 3503 of *Lecture Notes in Computer Science*, pages 606–609. Springer, 2005.
- [TM69] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425 – 443, 1969.
- [TPSF01] L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the internet topology. In *Global Internet*, San Antonio, Texas, USA, November 2001. IEEE CS Press.
- [VDS⁺04] A. Vazquez, R. Dobrin, D. Sergi, J.P. Eckmann, Z.N. Oltvai, and A.-L. Barabasi. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(52):17940–17945, December 2004.
- [Vir03] Satu Virtanen. Clustering the chilean web. In *LA-WEB '03: Proceedings of the First Conference on Latin American Web Congress*, Washington, DC, USA, 2003. IEEE Computer Society.
- [VL05] Fabien Viger and Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In Lusheng Wang, editor, *COCOON*, volume 3595 of *Lecture Notes in Computer Science*, pages 440–449. Springer, 2005.
- [Waz09] Bill Wazik. *And Then There's This. How Stories live and die in viral culture*. Viking, New York edition, 2009.
- [WD07] D. J. Watts and P. S. Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34:441–458, 2007.
- [Wel79] Barry Wellman. The community question: the intimate networks of east yorkers. *American Journal of Sociology*, 84:1201–1231, 1979.
- [Wel82] Barry Wellman. *Studying Personal Communities*. Sage, Beverly Hills, 1982.
- [Wel85] Barry Wellman. *Domestic Work, Paid Work and Net Work*. Sage, London, 1985.

- [Wel88] Barry Wellman. Structural analysis: from method and metaphor to theory and substance. In Barry Wellman and Stephen D. Berkowitz, editors, *Social structures: a network approach*, pages 19–61. Cambridge University Press, Cambridge, 1988.
- [Wer06] Sebastian Wernicke. Efficient detection of network motifs. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 3(4):347–359, 2006.
- [WF94] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [WF01] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proc Biol Sci*, 268(1478):1803–1810, 2001.
- [WH04] Fang Wu and Bernardo A. Huberman. Finding communities in linear time: A physics approach. *European Physical Journal B*, 38:331–338, 2004.
- [WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small world networks. *Nature*, 393(4):440–442, June 1998.
- [WW90] Barry Wellman and S. Wortley. Different strokes from different folks: Community ties and social support. *American Journal of Sociology*, 96(3):558–588, 1990.
- [YH02] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 721, Washington, DC, USA, 2002. IEEE Computer Society.
- [YH03] Xifeng Yan and Jiawei Han. Closegraph: mining closed frequent graph patterns. In *Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 286–295. ACM, 2003.
- [ZAA07] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 221–230, New York, NY, USA, 2007. ACM.

Appendix A

Introduction (en français)

Contexte et motivations

La principale motivation de notre recherche a été l'analyse de la structure locale des grands réseaux sociaux. Comment un noeud est-il connecté au réseau ? Comment peut-on analyser la totalité des noeuds en temps raisonnable ? Est-ce que la façon dont le noeud est connecté au réseau nous donne des informations sur la personne représentée par le noeud ? Est-ce qu'il y a une corrélation entre la structure du réseau autour d'un individu et son âge, sexe ou usages (du téléphone mobile, des plateformes sociales en ligne etc.) ?

Donc le but de notre recherche est de caractériser des individus en analysant le réseau social dans lequel ils sont connectés. Une telle caractérisation est utile par exemple pour les fournisseurs de services, pour lesquels la connaissance de leurs clients est très importante. Il leur est essentiel de savoir quels sont les services que les clients souhaitent avoir et comment leurs attentes évoluent pour que les offres et la publicité soient adaptées et envoyées aux personnes susceptibles d'y répondre favorablement.

Pour obtenir une telle **caractérisation des utilisateurs**, on peut adopter plusieurs approches. On peut utiliser des données sociodémographiques comme l'âge, le sexe, le métier, la position géographique etc. D'autres informations peuvent être exploitées, qui peuvent s'avérer encore plus profitables et fiables que les données sociodémographiques : ce sont les traces laissées par les clients en utilisant différents services. Les opérateurs de téléphonie mobile savent ainsi combien de fois par jour une personne effectue des appels téléphoniques, quelles sont les durées de ses conversations, avec combien de personnes elle communique etc. De la même façon, les créateurs de plateformes en ligne peuvent aussi utiliser des traces d'usage. Par exemple sur une plateforme de réseau social et de partage de photos et vidéos comme Flickr (www.flickr.com), les utilisateurs peuvent se déclarer les uns les autres comme contacts, peuvent enregistrer et publier des photos et des vidéos, peuvent écrire des commentaires etc. On peut utiliser ces informations (quantité de contenu publié, commentaires, nombre de contacts etc.) comme une caractérisation de l'activité de chaque personne sur la plateforme. Ensuite on peut proposer aux différents utilisateurs des services spécifiques à leurs usages.

Aujourd'hui, les **traces d'usage** sont présentes partout et sont généralement faciles

d'obtenir. Presque tout le monde a un téléphone portable, une adresse e-mail et de plus en plus de personnes utilisent des plateformes en ligne comme Facebook, MySpace, Flickr, Twitter, Wikipedia, Delicious, LinkedIn etc. Unes de ces plateformes sont dédiées au réseau social, d'autres à la publication de contenus (photos, vidéos, textes etc.), à l'information etc. mais toutes gardent des traces d'activité humaine. Le développement d'Internet, de "Web2.0", des communications en général mais aussi d'ordinateurs puissants capables d'enregistrer, mémoriser et traiter des gros volumes de données offrent des possibilités sans précédent pour l'analyse du comportement humain. Traditionnellement ceci a été le champ d'étude des sociologues, mais de plus en plus de chercheurs, de nombreux domaines, s'y intéressent. De telles bases de données contenant des traces de communications intéressent par exemples des mathématiciens et des informaticiens qui cherchent des mesures pertinentes pour caractériser les usages, développent des algorithmes et des logiciels pour traiter efficacement les gros volumes de données etc. Elles intéressent aussi des physiciens qui essaient de découvrir les processus derrière les différentes activités ou dynamiques des gens ou des économistes qui essaient par exemple de dévoiler les motivations des individus dans la prise de décisions.

Les traces d'usages peuvent être analysées de plusieurs points de vue. Une approche possible est de calculer différentes statistiques sur la fréquence ou la durée des appels dans le cas des communications par téléphone mobile, les commentaires et les contenus publiés dans le cas des plateformes en ligne etc. Cette approche a donné des résultats intéressants sur l'usage des différents services des groupes d'information [FSW06], wikis [HBB07], communautés de rencontres en ligne [HEL04], forums de questions/ réponses [ZAA07, AZBA08], Youtube [CKR⁺07, MAA08] et beaucoup d'autres plateformes. Une autre approche, que nous adoptons dans cette thèse, consiste dans l'analyse du **réseau social** connectant les individus. En utilisant les différents services, en ligne ou hors ligne, les gens se connectent les uns aux autres. Ces connections peuvent être modélisées comme des réseaux sociaux, simplement des graphes où les noeuds sont les personnes et les liens correspondent à des connections observées entre eux. Il est important de prendre en considération ces connections car les individus ne sont pas des entités isolées, ils vivent ensemble, interagissent et s'influencent les uns les autres. Un phénomène souvent confirmé c'est celui de "bouche-à-oreille" ("word-of-mouth") [EBK69, FS65, AD07] : avant de prendre une décision, les gens parlent souvent avec d'autres gens, demandent leur conseil et sont plus susceptible de choisir un produit si une personne à laquelle ils font confiance l'a déjà choisi. De plus, il est possible que les individus se connectant de la même façon aux autres aient des comportements similaires, aiment les mêmes choses etc. Il est donc important de voir, analyser et caractériser les gens et leurs usages en prenant en considération le contexte dans lequel ils évoluent, les gens auxquels ils se connectent, donc le réseau social dans lequel ils sont intégrés.

En sociologie, l'analyse des réseaux sociaux n'a pas apparue avec les bases de données sur les traces d'usages, mais beaucoup de temps auparavant, quand Internet et les communications mobiles n'existaient pas encore. Déjà présente dans les travaux de G. Simmel [Sim55a] (traduction anglaise) au tout début du 20ème siècle, elle s'est beaucoup développée dans les années 1950 quand des chercheurs comme John A. Barnes, Elisabeth Bott, Sigfried F. Nadel ont étudié des types de liens entre des individus [Bar54], des relation de parenté

[Bot57] et des structures sociales [Nad57]. Ensuite, dans les années 1970 Harrison White et ses étudiants à l'université Harvard, parmi lesquels Mark Granovetter et Barry Wellman, ont développé et rendu populaire l'analyse des réseaux sociaux. Depuis, des questionnements comme la force des liens interpersonnels [Gra78], le capital social [Col88, Bur92], les rôles sociaux [LW71, BE89] et beaucoup d'autres reviennent souvent. Traditionnellement, dans l'analyse des réseaux sociaux, les sociologues recensaient leurs données par des entretiens avec les individus étudiés. Les données ainsi obtenues sont très riches, très détaillées, mais leur collecte prend du temps car on doit interviewer toutes les personnes de l'étude. Les traces d'usages disponibles aujourd'hui offrent des nouvelles possibilités pour l'analyse des réseaux sociaux. Néanmoins, on a une image beaucoup moins détaillée des activités humaines et des relations entre les individus. Beaucoup d'informations ne sont pas visibles dans les traces d'usage et, par rapport à l'entretien, on ne peut pas poser des questions sur les informations manquantes aux gens étudiés. Le type de relation entre deux personnes observées n'est ainsi pas connu : sont-elles amies, collègues, famille, se connaissent-elles ? Aussi, on ne voit pas toutes les connections entre les deux personnes. Peut-être elles ne se contactent pas par téléphone mobile mais ont d'autres types de contact, par téléphone fixe, e-mail etc. Toutefois, même si les données ne sont pas aussi détaillées que celles obtenues par entretien, la collecte des données est beaucoup plus simple, les volumes sont beaucoup plus importants et ils concernent beaucoup de gens. La difficulté change ainsi de la collecte de données à leur traitement.

Comme un réseau social est un graphe, on utilise généralement la théorie des graphes quand on étudie des réseaux sociaux. De plus, les grands réseaux sociaux (avec au moins quelques milliers de noeuds) sont aussi des **graphes de terrain** (ou grands réseaux d'interaction, en anglais complex networks). C'est le nom commun donné aux graphes modélisant des relations entre entités (personnes, institutions, endroits etc.) existantes dans la vraie vie. L'analyse des graphes de terrain a été l'objet d'un grand intérêt depuis les premières études dans le domaine, à la fin des années 1990. Ce qui a généré tout l'intérêt a été la découverte récurrente que les grands réseaux modélisant des relations réelles sont très différents des réseaux aléatoires, donc ils ne sont pas aléatoires. Le terme "réseaux aléatoires" fait référence ici à des réseaux où il n'y a aucune contrainte pour relier deux noeuds par un lien : chaque paire de noeuds peut être connectée par un lien avec la même probabilité. Ceci définit un modèle de génération aléatoire de réseaux introduit par Erdos et Renyi dans les années 1960 [ER60], étant ainsi le premier et le plus simple modèle de génération. Le probable premier article décrivant des différences entre des réseaux réels et aléatoires a été [WS98] par Watts et Strogatz. Comme les graphes étudiés dans cet article étaient très différents de ceux générés par le modèle Erdos-Renyi, les auteurs ont conclu que ce modèle n'était pas adapté pour la génération de graphes réalistes. Par rapport au modèle Erdos-Renyi où n'importe quels deux noeuds peuvent être connectés par un lien avec la même probabilité, dans la vraie vie il y a probablement une raison pour laquelle deux noeuds deviennent connectés, il doit y avoir des facteurs qui font qu'un graphe apparaît et évolue d'une certaine façon. Les auteurs ont proposé un nouveau modèle de génération et ainsi a commencé une longue série de modèles. Les plus connus dans cette série sont ceux proposés par Kleinberg [Kle00] et Barabasi et Albert [BA99], mais beaucoup d'autres existent [LKF05, KKR⁺99, KRRT99, BJN⁺02] etc.

Depuis ces premières études, les chercheurs ont constamment observé des différences entre les réseaux réels et ceux aléatoires. Essentiellement, peu importe le contexte duquel le graphe provient (sociologie, biologie, économie, linguistique, informatique etc.), dans presque (si n'est-ce que) tous les cas, le graphe a les mêmes propriétés que tous les autres graphes modélisant des relations réelles, appartenant ainsi au groupe de "graphes de terrain". Nous présentons brièvement quelques unes de ces propriétés ici : la plupart des noeuds sont connectés à très peu de noeuds, tandis qu'une petite fraction de noeuds est connectée à un grand nombre de noeuds. Aussi, la plupart des noeuds appartiennent à la même composante géante : pour la plupart des paires de noeuds, on peut se déplacer d'un noeud à l'autre en suivant les liens du graphe. De plus, en allant du premier noeud au deuxième de la façon la plus directe, on traverse seulement un petit nombre de liens, habituellement inférieur à 20. Et ceci même si le graphe a plusieurs millions de noeuds. Une autre propriété partagée par les graphes de terrain est celle de la grande densité locale : si deux noeuds sont reliés à un noeud commun, il y a une forte probabilité qu'ils soient reliés entre eux aussi. Ici "forte" signifie beaucoup plus forte que dans des réseaux aléatoires. Ces propriétés ont été observées par exemple dans des graphes de citation [Red98], d'interactions de protéines [GR03, WF01], dans des réseaux neuraux biologiques [MiOO⁺01, SGS⁺02], chaînes alimentaires [DWM02], réseaux sociaux modélisant des relations en ligne [MKG⁺08, ABA03] et beaucoup d'autres. Comme présenté auparavant, en développant un modèle de génération aléatoire, les chercheurs essaient d'identifier les facteurs qui amènent à la création des liens et ainsi expliquer la formation des réseaux réels. La qualité du modèle de génération proposé est mesurée par la capacité du modèle de produire des réseaux qui partagent les propriétés des réseaux réels.

Il y a plusieurs approches dans l'analyse des graphes de terrain en général et des réseaux sociaux en particulier. Généralement l'analyse se place à un des trois niveaux suivants : global, intermédiaire ou local. Au niveau global on prend en considération le réseau dans sa totalité et on calcule des différentes propriétés de cet ensemble. Parmi les propriétés présentées antérieurement, le calcul de la composante géante, de la distance entre les noeuds et de la distribution du nombre de contacts appartiennent à cette approche. Dans l'approche au niveau intermédiaire, on analyse chaque noeud en prenant en considération le réseau global. A ce niveau on peut calculer par exemple des groupes de noeuds qui sont densément connectés à l'intérieur du groupe et peu connectés aux autres groupes ; cela s'appelle détection de communautés et a fait l'objet de nombreuses études comme [Eve80, GN02, Vir03, CMN04, BGLL08] et beaucoup d'autres. Aussi au niveau intermédiaire on peut calculer "l'importance" de chaque noeud, habituellement exprimée en termes de centralité (e.g. betweenness [Fre77], closeness, vecteur propre [Bon87], page rank [BP98] etc.). Finalement, au niveau local, une mesure largement utilisée est le coefficient de clustering [WS98, HK79] qui mesure la densité locale du réseau. Brièvement, on calcule dans quelle mesure les noeuds auxquels un noeud donné est connecté sont connectés entre eux (par rapport au cas où tous ces noeuds sont connectés entre eux). Dans cette approche locale l'idée est d'analyser chaque noeud en prenant en considération seulement les noeuds qui l'entourent et pas le réseau global. C'est l'approche que nous adoptons dans cette thèse.

Nous nous proposons de répondre à la question suivante : étant donné un réseau so-

cial (potentiellement grand), décrire sa structure locale, donc la façon dont chacun de ses noeuds est connecté au réseau environnant. Cette description devrait représenter une caractérisation des individus appartenant au réseau social en prenant en considération seulement la structure du réseau (et pas d'autres informations sur les individus). Le calcul de cette description devrait prendre peu de temps et de mémoire pour qu'il puisse être appliqué à des grands réseaux sociaux. A notre connaissance, les méthodes existantes soit placent l'analyse au niveau intermédiaire (donc elles caractérisent le noeud en prenant en considération tout le réseau), soit offrent trop peu d'informations (comme le coefficient de clustering qui simplement compte les liens entre les contacts d'un noeud).

Nous proposons une méthode pour répondre à cette question, donc une méthode qui analyse la structure locale d'un graphe donné et qui décrit la façon dont chaque noeud est connecté au réseau. Cette méthode prend en considération les liens que chaque noeud a avec d'autres noeuds et les liens entre ces noeuds. Nous appliquons cette méthode à deux réseaux sociaux : un modélisant des communications par téléphone mobile et un autre modélisant des activités sur MySpace. Dans ces réseaux chaque noeud correspond à une personne ; quand nous analysons une personne nous appelons celle-ci *ego*. Comme nous analysons la façon dont ego est connecté au réseau, cette analyse peut être appelée égocentré. Notre approche ici est liée à l'analyse de réseaux égocentrés retrouvée en sociologie. Dans ce cas, on étudie les relations personnelles qu'un individu donné (ego) a avec d'autres individus. Les données pour des telles études sont obtenues par des entretiens avec ego qui décrit ses relations avec les autres personnes et, parfois, les relations entre ces personnes [Wel79, Wel85, Gri98, Gro05]. Ici nous essayons d'adapter cette approche à des grands réseaux sociaux, où les réseaux égocentrés sont obtenus en se fixant sur chaque individu et ses liens dans le réseau. Les réseaux égocentrés ainsi obtenus contiennent moins d'informations, sont moins détaillés que ceux obtenus par des entretiens avec ego. L'avantage toutefois est que les réseaux obtenus à partir de grands graphes sont tous construits de la même façon, en utilisant des interactions observées, et ainsi ne sont pas subjectifs à l'opinion d'ego sur ses relations et surtout sur ceux de ses contacts.

La méthode proposée calcule une description de la façon dont chaque noeud est connecté au réseau environnant et aussi de la façon dont les différentes personnes auxquelles ego est connecté sont placées les unes par rapport aux autres. Comme l'approche est locale, la méthode n'a pas besoin de tout le réseau social pour caractériser un noeud (par rapport aux méthodes intermédiaires), mais seulement des noeuds auxquels ego est connecté et des liens entre eux. Ainsi, la méthode peut être appliquée même si on a juste des fractions d'un certain réseau social. Elle peut être appliquée aussi bien à des petits réseaux obtenus par des entretiens qu'à des grands réseaux sociaux. Encore une fois, parce qu'elle est locale, sa complexité dans l'analyse d'un ego est aussi "locale" i.e. elle dépend seulement du nombre de contacts d'ego dans le réseau. Cela est important parce qu'elle peut facilement être appliquée à des grands réseaux ; pour donner une idée, notre implémentation de la méthode s'exécute en 30 minutes pour tous les noeuds d'un réseau social avec 3 millions de noeuds et 6 millions de liens sur un ordinateur de configuration standard.

Après avoir obtenu une caractérisation des différentes personnes en prenant en considération seulement le réseau social les incluant, on peut chercher des corrélations entre cette description et d'autres mesures caractérisant les individus. Ces mesures peuvent

être des informations sociodémographiques (âge, sexe, métier etc.) ou des indicateurs de l'activité des individus. Par exemple pour le téléphone mobile nous utilisons l'intensité des communications de chaque personne (nombre d'appels, durée, nombre de SMS etc.), tandis que pour le réseau MySpace nous utilisons des mesures de popularité en ligne. Si les différents paramètres et la structure locale du réseau (obtenue en appliquant la méthode proposée) sont corrélés, alors on peut utiliser les paramètres pour déduire la structure locale et vice-versa. Cela peut être utile quand il y a des données manquantes, par exemple si on a le réseau social dans lequel l'individu est intégré sans avoir les autres informations le caractérisant. On peut aussi distribuer les personnes du réseau social donné dans des groupes en fonction de la structure du réseau les entourant : les individus connectés au réseau des façons identiques ou similaires sont mis dans le même groupe ; les individus avec des structures locales différentes sont mis dans des groupes différents. Cette approche est liée au calcul de "rôles" des noeuds d'un réseau social, où les noeuds occupant la même position, ayant la même fonction dans le réseau sont regroupés. Notons que dans la recherche de rôles sociaux (et dans notre approche ici), les noeuds mis ensemble dans le même groupe ne sont pas forcément liés les uns aux autres et n'ont pas forcément de contact commun, ils sont juste connectés de la même manière au réseau. Le problème de la distribution d'individus dans des groupes en s'appuyant sur une caractérisation préalable, de la recherche de corrélations entre des indicateurs et de la prédiction des différents paramètres sont souvent rencontrés dans la fouille de données (data mining). Nous utilisons quelques techniques bien connues de ce domaine pour résoudre les différents problèmes.

Le chapitre suivant représente la traduction française du chapitre central de cette thèse, celui décrivant la méthode proposée.

Appendix B

Une méthode pour l'analyse de la structure locale des grands réseaux

Dans ce chapitre nous présentons une méthode pour l'analyse de la structure locale des réseaux (éventuellement grands) en caractérisant la façon dont chaque noeud est connecté au réseau. La méthode est conçue pour être appliquée à un noeud donné du réseau ; dans ce cas elle produit une caractérisation de la configuration du réseau entourant le noeud : les structures dans lesquelles le noeud est intégré, la manière dont ses voisins sont placés les uns par rapport aux autres et la façon dont ses liens sont disposés. On peut appliquer cette méthode à tous les noeuds du réseau, obtenant ainsi une description de sa structure locale, ou seulement à quelques uns de ces noeuds : cela peut être utile si l'on a juste une fraction des noeuds du réseau ou si le but est de comparer quelques noeuds entre eux. Avant de présenter la méthode, nous introduisons quelques notions utiles. Ensuite nous expliquons la méthode et nous comparons les mesures qu'elle produit à d'autres indicateurs existants. Nous terminons ce chapitre en faisant quelques commentaires sur l'utilité de la méthode.

B.1 Définitions

Sauf précisé différemment, tous les graphes considérés sont simples et non-dirigés.

Réseau égocentré. Etant donné un graphe $G = (V, E)$ et un sommet $v \in V$, nous appelons *réseau égocentré* de v , noté $Eg(v)$, le sous-graphe induit dans G par les voisins de v i.e. le graphe dont les sommets sont les voisins de v et les liens sont les liens entre ces voisins.

Patterns et positions. Nous appelons *k-patterns* tous les graphes connexes non-isomorphes avec au plus k sommets et au moins 1 lien. Figure B.1 présente les trente 5-patterns. Il y a neuf 4-patterns (numéros 1 à 9) et trois 3-patterns (numéro 1 à 3). Dans ce chapitre nous prenons en considération seulement les 5-patterns que nous appelons simplement *patterns*.

Deux sommets d'un graphe donné sont position-équivalents s'il existe une permutation des sommets du graphe telle que l'adjacence est respectée et les deux sommets sont échangés (la position-équivalence est en fait l'équivalence automorphique). Une *position*

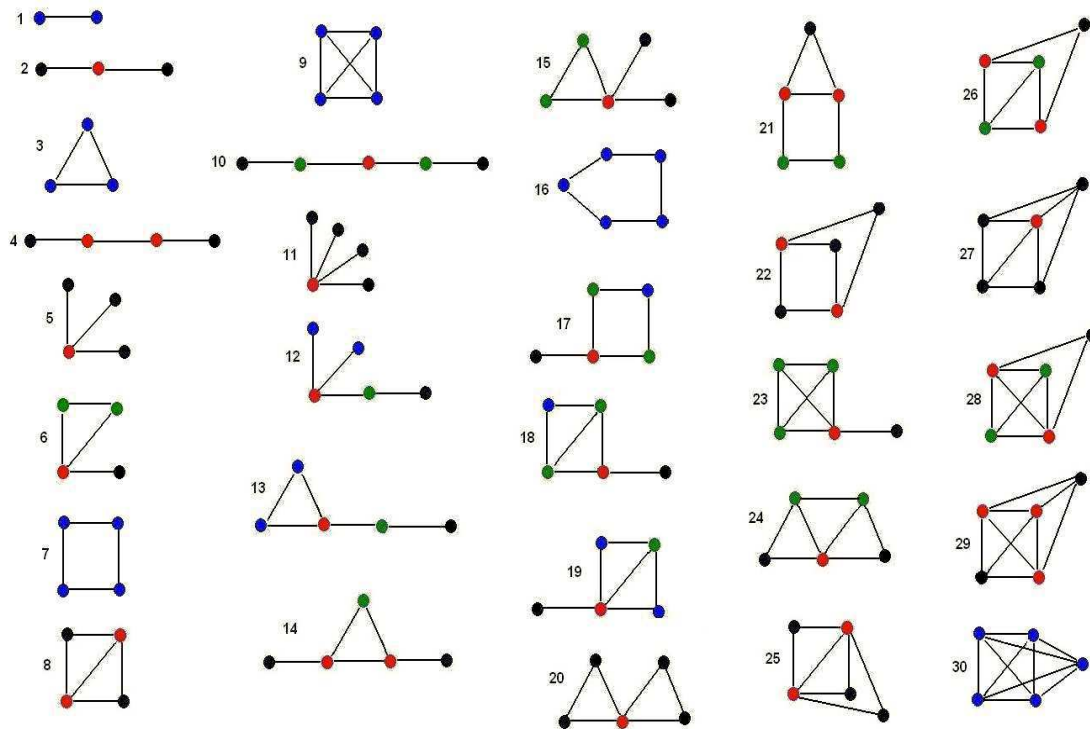


Figure B.1 – L'ensemble de patterns et leurs positions.

est un ensemble maximal de sommets position-équivalents. Par exemple, pour chaque pattern de la Figure B.1, chaque couleur correspond à une position distincte. Formellement, deux sommets u et v d'un graphe G sont position-équivalents s'il existe un automorphisme φ de G tel que $\varphi(u) = v$. Les positions correspondent aux classes d'équivalence de cette relation. Il y a 73 positions différentes dans les 30 patterns et, comme la Figure B.1 montre, un pattern a au plus 4 positions différentes. Nous voulons établir des catégories de positions donc nous trions les positions d'un même pattern en ordre croissant de leur centralité *betweenness*; pour des positions ayant la même centralité, nous trions en ordre croissant du degré. Nous appelons *périphérique* la première position dans cet ordre et *centrale* la dernière. Les positions qui ne sont ni centrales ni périphériques ou qui sont à la fois centrales et périphériques sont appelées *intermédiaires*. Brièvement, les positions colorées en rouge sont centrales, celles colorées en noir sont périphériques et les autres sont intermédiaires.

Caractérisation de graphes. Etant donné un graphe $G = (V, E)$, on peut obtenir une caractérisation de G en comptant les apparitions des différents patterns dans le graphe, et une caractérisation de ses sommets en comptant les positions que chaque sommet occupe dans chaque pattern. Un pattern P apparaît dans le graphe G s'il existe un ensemble de sommets $V_P \subseteq V$ tel que le sous-graphe induit par V_P dans G est isomorphe à P . Enumérer toutes les apparitions du pattern P dans le graphe G signifie trouver tous les ensembles V_P respectant la définition précédente. Pour chaque apparition d'un pattern dans $G = (V, E)$ on peut calculer dans quelle position du pattern se trouvent les différents sommets de V . Ainsi, après avoir énuméré toutes les apparitions des 30 patterns dans G , on a, pour chaque sommet $v \in V$, son nombre d'apparitions dans chacune de 73 positions (on appelle cela le *vecteur de position* de v). Formellement, le k -vecteur de position de v est un tableau $Pos_k(G, v)$ qui contient le nombre d'apparitions de v dans les différentes positions des k -patterns : $Pos_k(G, v, i)$ compte les sous-graphes de G avec au plus k sommets qui contiennent v dans la position i . Par exemple, la Figure B.2 représente un graphe (a), les patterns qu'il contient (b), et le nombre d'apparitions de deux sommets choisis dans les différentes positions (c) (nous avons noté seulement les positions où au moins un des deux sommets est présent ; pour toutes les autres positions l'élément correspondant dans le vecteur de position est 0.)

B.2 Caractérisation efficace de graphe

Quand on caractérise un graphe comme expliqué précédemment, on a besoin de chercher tous les sous-graphes induits avec un nombre maximal de sommets donné (dans notre cas 5), de trouver à quel pattern chacun d'eux est isomorphe et de calculer le nombre d'apparitions des différents sommets dans les différentes positions. Les trois opérations (l'énumération de patterns, la vérification de l'isomorphisme et le calcul de positions) doivent être faites efficacement pour pouvoir caractériser un grand nombre de graphes en temps raisonnable.

Pour l'énumération de sous-graphes on utilise l'Algorithme *ESU* introduit dans [Wer06]. La Figure B.3 présente cet algorithme ; $N_{excl}(w, V_{subgraphs})$ (ligne E_4) représente l'ensemble

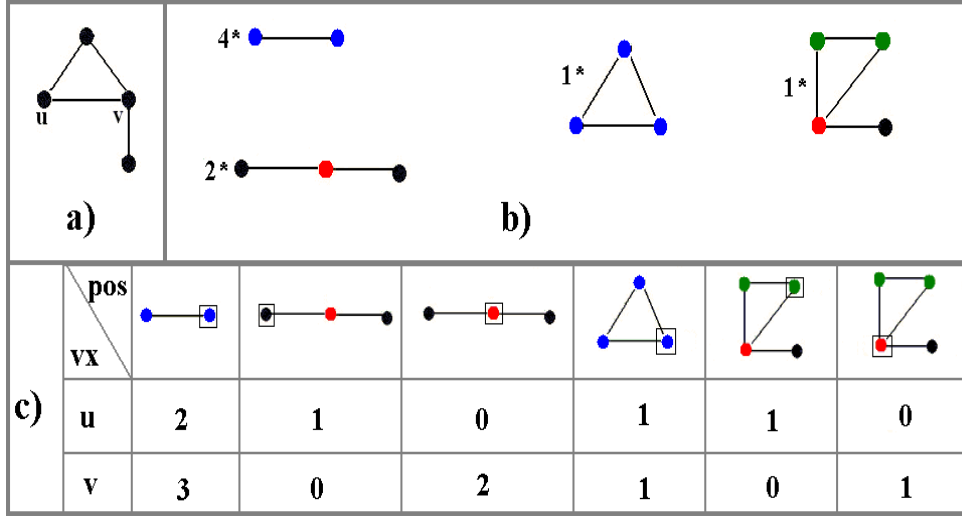


Figure B.2 – Un graphe (a), ses patterns (b) et les vecteurs de position des sommets u et v (seulement les positions où au moins un des deux sommets est présent) (c).

de voisins de w qui n'appartiennent pas à $V_{subgraphs}$ et n'ont pas de voisin dans $V_{subgraphs}$. Essentiellement, cet algorithme commence avec un sommet v de G et ajoute des sommets voisins jusqu'à l'obtention d'un ensemble de k sommets, donc d'un sous-graphe connexe induit avec k sommets. Plus précisément, commençant par le sommet v , l'algorithme ajoute répétitivement des voisins de v ou des sommets déjà ajoutés ($V_{extension}$ est l'ensemble de sommets qui peuvent être ajoutés). C'est le calcul de l'ensemble $V_{extension}$ qui rend cet algorithme efficace. Pour être ajouté à cet ensemble, un sommet doit satisfaire deux conditions : son étiquette doit être supérieure à celle de v (les étiquettes sont simplement des numéros de 1 à $|V_G|$) et il doit avoir exactement un voisin dans les sommets déjà ajoutés. Cela assure l'ajout de chaque sommet exactement une fois. Aussi, comme expliqué dans [Wer06], l'algorithme trouve chaque sous-graphe exactement une fois, dont on n'a pas besoin de vérifier la présence d'un sous-graphe trouvé dans la liste de graphes déjà identifiés. A notre connaissance, cet algorithme est le plus efficace algorithme existant pour l'énumération de sous-graphes induits.

Une fois avoir trouvé un sous-graphe induit, on a besoin de trouver le pattern auquel il est isomorphe. Pour plusieurs patterns cela peut être fait en calculant la distribution de degré de leurs sommets : les patterns avec des distributions de degré différentes ne sont pas isomorphes. Néanmoins la réciproque n'est pas toujours vraie. Par exemple, les patterns numéro 21 et 22 de la Figure B.1 ont la même distribution de degré (2, 2, 2, 3, 3). Dans ce cas on peut différencier les deux patterns en regardant non seulement les degrés des sommets, mais aussi comment les sommets de différents degrés sont interconnectés. Ainsi, pour le pattern 21, deux sommets de degré 2 sont liés l'un à l'autre, tandis que les sommets de degré 2 du pattern 22 sont connectés seulement à des sommets de degré 3. Pour prendre en considération en même temps les degrés des sommets et des leurs voisins, nous introduisons la notion de voisin-degré (en anglais neighbor-degree).

```

Algorithm: ENUMERATESUBGRAPHS( $G, k$ ) (ESU)
Input: A graph  $G = (V, E)$  and an integer  $1 \leq k \leq |V|$ .
Output: All size- $k$  subgraphs in  $G$ .

01 for each vertex  $v \in V$  do
02    $V_{Extension} \leftarrow \{u \in N(\{v\}) : u > v\}$ 
03   call EXTENDSUBGRAPH( $\{v\}, V_{Extension}, v$ )
04 return

EXTENDSUBGRAPH( $V_{Subgraph}, V_{Extension}, v$ )
E1 if  $|V_{Subgraph}| = k$  then output  $G[V_{Subgraph}]$  and return
E2 while  $V_{Extension} \neq \emptyset$  do
E3   Remove an arbitrarily chosen vertex  $w$  from  $V_{Extension}$ 
E4    $V'_{Extension} \leftarrow V_{Extension} \cup \{u \in N_{excl}(w, V_{Subgraph}) : u > v\}$ 
E5   call EXTENDSUBGRAPH( $V_{Subgraph} \cup \{w\}, V'_{Extension}, v$ )
E6 return

```

Figure B.3 – Pseudocode pour l'algorithme *ESU* qui énumère tous les sous-graphes avec k sommets dans un graphe donné G [Wer06].

Definition B.2.1. *Etant donné un graphe G et un sommet v de G , nous appelons voisin-degré de v , noté $nd(v) = \sum_{u \in N[v]} d(u)$, la somme de son degré et ceux de ses voisins. Nous appelons combinaison de degrés du graphe G la liste triée en ordre croissant des voisin-degrés de ses sommets.*

Ces deux notions suffisent pour vérifier si deux graphes connexes avec au plus 5 sommets sont isomorphes, comme le montre le lemme suivant.

Lemma B.2.2. *Deux graphes connexes G et H avec au plus 5 sommets sont isomorphes si et seulement si leurs combinaisons de degrés sont identiques. De plus, deux sommets $u, v \in V_G$ sont position-équivalents si et seulement s'ils ont le même voisin-degré.*

Proof. La démonstration est directe, il suffit de vérifier l'affirmation pour tous les graphes connexes avec au plus 5 sommets. \square

Pour les deux patterns de notre exemple précédent, la combinaison de degré du pattern 21 est (7, 7, 8, 10, 10), tandis que celle du pattern 22 est (8, 8, 8, 9, 9). Ainsi, les deux patterns sont trouvés comme non-isomorphes. De plus, les sommets du même pattern qui ont des positions différentes ont des voisin-degrés distincts.

Remarquons que pour un graphe G avec n sommets et m liens on calcule les voisin-degrés de tous les sommets de G en temps $O(m)$ et espace $O(n)$ (il suffit de parcourir tous les liens pour calculer et mémoriser tous les degrés, ensuite parcourir tous les liens



Figure B.4 – Deux graphes connexes non-isomorphes avec 6 sommets.

de nouveau pour calculer les voisin-degrés), ensuite sa combinaison de degré en temps $O(n \cdot \log n)$. Pour l'ensemble de patterns ces quantités sont constantes comme n et m sont au plus 5, respectivement 10. Donc on peut trouver à quel pattern un graphe connexe avec au plus 5 sommets correspond (i.e. auquel des 30 graphes de la Figure B.1 il est isomorphe) et vérifier si deux de ses sommets sont équivalents en temps constant.

Toutefois le lemme n'est pas valable pour les graphes connexes avec 6 sommets. Les deux graphes de la Figure B.4 ne sont pas isomorphes mais ont la même combinaison de degrés : (7, 7, 7, 7, 10, 10).

B.3 Une méthode pour l'analyse de la structure locale

Etant donné un graphe (éventuellement grand) $G = (V, E)$, nous nous proposons d'analyser sa structure locale autour d'un sommet $v \in V$ (nous appelons ce sommet *ego*). Nous procédons comme il suit – méthode ***structure_locale(v)*** :

Etape 1. Extraire le réseau égocentré $Eg(v)$ de v i.e. le sous-graphe induit par les voisins de v dans G ;

Etape 2. Enumérer les patterns de $Eg(v)$;

Etape 3. Calculer les vecteurs de position des sommets de $Eg(v)$.

Nous expliquons les trois étapes de la méthode avec un exemple.

Etapes 1 et 2. Dans la Figure B.5(a), les cercles noirs correspondent aux voisins de v , les traits noirs correspondent aux liens entre eux et les traits rouges aux liens entre v et ses voisins. Le réseau égocentré $Eg(v)$ de v est représenté dans la Figure B.5(b) et les patterns de $Eg(v)$ dans la Figure B.5(c) ¹. Nous choisissons de ne pas inclure v dans son réseau égocentré parce que nous savons qu'il est connecté à tous les sommets de ce graphe, sa présence n'apporte aucune information. Après avoir effectué les deux premiers pas de la méthode, on a une description riche de la façon dont v est connecté au graphe G . Pour une description plus détaillée de la structure locale de G on peut énumérer les patterns d'un plus grand ordre (avec plus de 5 sommets) ; les patterns avec 5 sommets

¹ Nous avons aussi compté les sommets et les liens isolés de $Eg(v)$.

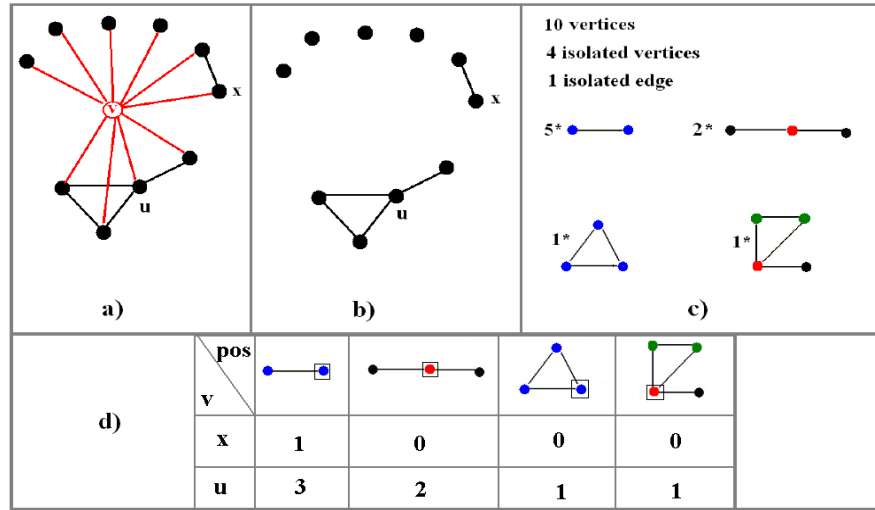


Figure B.5 – Un sommet v et ses voisins (a), le réseau égocentré $Eg(v)$ de v (b), les patterns de $Eg(v)$ (c) et les vecteurs de position de deux voisins de v (d) (seulement les positions où au moins un des deux sommets est présent).

représentent toutefois un bon compromis entre la variété des formes et leur nombre ; même les 4-patterns offrent dans beaucoup de cas une image suffisamment détaillée.

Etape 3. Nous calculons les vecteurs de position des voisins de v , donc le nombre de fois chaque voisin apparaît dans chacune des positions des différents patterns. La Figure B.5(d) présente les vecteurs de positions de deux voisins de v (seulement les éléments qui sont supérieurs à 0 pour au moins un des sommets ; tous les autres éléments sont égaux à 0). Les positions occupées par les différents voisins décrivent la place relative de ces voisins par rapport aux autres voisins mais aussi les liens formés par v , si on regarde du point de vue de v . Par exemple, la Figure B.6 présente la correspondance entre trois positions possibles d'un voisin u et la structure du graphe autour du lien (u, v) .

Si le graphe G est dirigé, on peut ajouter cette information à la description des liens formés par v en donnant simplement un poids aux voisins de v . Pour un noeud v , le poids $w_v(u)$ d'un voisin u est :

- 1 si la connexion est de v à u ($v \rightarrow u$),
- 2 si la connexion est de u à v ($u \rightarrow v$),
- 3 si la connexion est symétrique ($v \rightarrow u$ et $u \rightarrow v$).

Comme exemple, la Figure B.7 présente la correspondance entre une position possible d'un voisin u qui a poids 2 et la structure du graphe autour du lien (u, v) .

La méthode introduite ici peut être utilisée pour définir une relation d'équivalence sur les sommets du graphe G . D'abord, chaque sommet peut être caractérisé par un vecteur

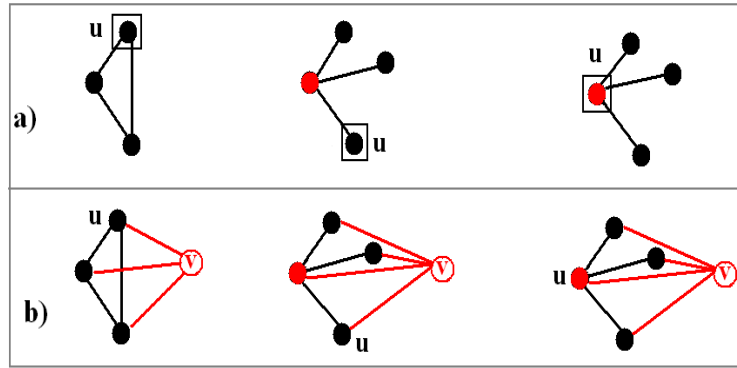


Figure B.6 – Trois positions possibles du voisin u (a) et les structures correspondantes autour du lien (u, v) (b).

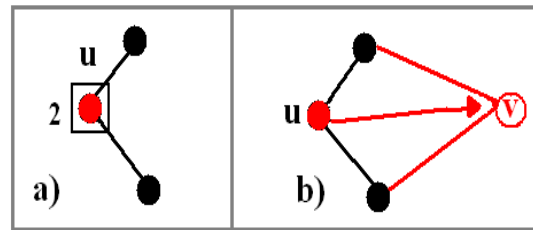


Figure B.7 – La position du voisin u avec poids 2 (a) et la structure correspondante autour du lien (u, v) (b).

contenant le nombre d'apparitions de patterns avec au plus k sommets dans son réseau égocentré. Ensuite, on peut utiliser ces vecteurs pour identifier des sommets équivalents.

Definition B.3.1. *Etant donné un sommet v d'un graphe G et un nombre entier positif k , nous appelons k -pattern vecteur de v le tableau contenant le nombre d'apparitions des k -patterns (i.e. tous les graphes connexes non-isomorphes avec au plus k sommets) dans le réseau égocentré $Eg(v)$ de v . Deux sommets du graphe G sont k -pattern équivalents si et seulement s'ils ont des k -pattern vecteurs identiques.*

B.4 Considérations algorithmiques

Nous rappelons que le graphe $G = (V, E)$ auquel la méthode est appliquée peut être grand (plus que 10^5 sommets et encore plus de liens). Par conséquent on doit faire attention à la complexité temps et espace des algorithmes utilisés. Premièrement, nous mémorisons le graphe G dans la représentation liste d'adjacence (voir la Section 2.1) : pour chaque sommet, on a la liste de ses voisins triée en ordre croissant (les sommets de V sont numérotés de 0 à $|V| - 1$). Cette représentation nécessite espace $\Theta(|E|)$ et le parcourt de $N(v)$ prends $\Theta(d(v))$ temps, où $d(v)$ représente le degré de v . Le test de la présence d'un lien (u, v) prends $O(\log(d(v)))$ temps. Pour un graphe $G = (V, E)$, soit n le nombre de ses sommets ($n = |V|$) et m le nombre de ses liens ($m = |E|$).

Etape 1. Dans cette étape nous avons besoin de calculer le réseau égocentré d'un sommet $v \in V$ i.e. le sous-graphe induit par les voisins de v dans G . Cela est équivalent à l'énumération des triangles contenant v . Pour cela, nous nous appuyons sur l'algorithme *new-vertex-listing* proposé dans [Lat08]. L'algorithme *ComputeEgocentered* calcule le réseau égocentré d'un sommet $v \in V$.

Algorithm 4 *ComputeEgocentered*. *Calcule le réseau égocentré d'un sommet*

Entrée : Un graphe $G = (V, E)$ simple non-dirigé et un sommet $v \in V$

Sortie : Un graphe $Eg = (V_v, E_v)$ simple non-dirigé, le réseau égocentré de v

1. créer un tableau A de $|V|$ nb. entiers initialisés à -1
 2. initialiser V_v et E_v à l'ensemble vide
 3. pour chaque sommet $u \in N(v)$, mettre $A[u]$ égal à v
 4. pour chaque sommet $u \in N(v)$
 - 4.1 ajouter u à V_v
 - 4.2 pour chaque sommet $w \in N(u)$ tel que $w < u$
 - si $A[w] = v$ alors ajouter (w, u) à E_v
-

L'algorithme *ComputeEgocentered*. On peut voir cet algorithme comme une façon d'utiliser la matrice d'adjacence de G sans la mémoriser explicitement : quand on traite un sommet v , le tableau A n'est rien d'autre que la v -ième ligne de la matrice d'adjacence. Ce tableau est construit en $\Theta(n)$ temps et espace. Ensuite on peut vérifier la présence

d'un lien (u, v) en $\Theta(1)$ temps et espace. Comme la ligne 4.2 est exécutée au plus deux fois pour chaque lien connectant un voisin de v , et il y a au plus m tels liens, on obtient que l'algorithme *ComputeEgocentered* a une complexité temps de $O(m)$ et espace de $\Theta(n)$.

Étapes 2 et 3. Nous voulons caractériser le graphe $Eg(v)$, donc nous calculons ses patterns et les positions de ses sommets. Pour simplifier les notations et parce que ces deux étapes constituent une méthode qui peut être appliquée à tout graphe, pas juste des réseaux égocentrés, nous notons le graphe $Eg(v)$ par G . D'abord, nous avons besoin d'identifier les sous-graphes connexes induits avec au plus 5 sommets de G , ensuite de trouver le pattern auquel chacun de ces graphes est isomorphe et finalement de calculer les positions occupées par les différents sommets dans le sous-graphe identifié (en fait les trois opérations sont successives : une fois avoir trouvé le sous-graphe, on vérifie à quel pattern il est isomorphe et on calcule les positions des sommets, ensuite on continue la recherche d'autres sous-graphes). Pour la première partie nous nous appuyons sur l'Algorithme *ESU*(G, k) [Wer06] (voir la Figure B.3) qui énumère les sous-graphes induits de G avec k sommets. Pour la deuxième et la troisième partie, nous calculons les voisin-degrés et la combinaison de degrés du sous-graphe trouvé, conformément au lemme B.2.2. L'Algorithme *CharacterizeWithPatterns* implémente les trois étapes.

L'Algorithme *CharacterizeWithPatterns*. Nous avons légèrement modifié l'Algorithme *ESU* (Figure B.3) pour calculer les sous-graphes induits avec *au plus* k sommets où $k \leq 5$. Aussi, l'opération *output* $G[V_{Subgraph}]$ (ligne E_1 dans *ESU*) est remplacée par la fonction *IndexPattern* qui calcule le pattern isomorphe au sous-graphe trouvé et les positions occupées par les différents sommets. L'Algorithme *CharacterizeWithPatterns* a une complexité temps linéaire dans le nombre de patterns trouvés dans le graphe G : pour l'Algorithme *ESU* voir [Wer06] ; pour la fonction *IndexPattern* remarquer que son exécution prend $O(m_p + n_p \times \log n_p + \log \text{nb_patterns})$, où n_p est le nombre de sommets dans le pattern (au plus 5), m_p est le nombre de liens (au plus 10) et nb_patterns est le nombre total de patterns différents (égal à 30 pour les patterns avec au plus 5 sommets). Comme toutes ces quantités sont inférieures à des constantes données, 5, 10 et $\log 30$ respectivement, on peut dire que *IndexPattern* a une complexité temps constante et l'Algorithme *CharacterizeWithPatterns* est linéaire dans le nombre de patterns du graphe G . Comme nous n'avons pas de méthode pour estimer le nombre de patterns d'un graphe donné, nous remarquerons simplement que le nombre de patterns avec au plus k sommets est au plus n^k où n est le nombre de sommets de G .

L'Algorithme *CharacterizeLocalStructure*. Nous avons maintenant tous les éléments pour écrire l'algorithme qui caractérise la structure locale du graphe $G = (V, E)$ autour de *chaque* sommet $v \in V$: l'Algorithme *CharacterizeLocalStructure*. Celui-ci est juste l'application des deux algorithmes précédents à tous les sommets du graphe. Remarquons toutefois une modification : le tableau A est construit une seule fois pour tous les sommets du graphe, au début de l'algorithme, et ensuite mis-à-jour pour chaque sommet. Ainsi la construction de A a la même complexité temps et espace que dans l'Algorithme *ComputeEgocentred* : $\Theta(n)$ pour les deux. La complexité temps de l'Algorithme *CharacterizeLocalStructure* est ainsi $\Theta(n + \sum_{v \in V} (\text{nb. patterns dans } Eg(v)))$ qui est (au plus) $O(n + \sum_{v \in V} (d(v)^5))$. Etant donné que nous appliquons la méthode à des grands réseaux réels, où la plupart de sommets a un degré faible, la méthode est en moyenne très rapide.

Algorithm 5 CharacterizeWithPatterns. *Caractérise un graphe simple non-dirigé*

Entrée : Un graphe simple non-dirigé $G = (V, E)$ et un nombre entier positif $k \leq 5$

Sortie : Un tableau Pt tel que $Pt[P]$ contient le nb. d'occurrences du pattern P dans G ,
un tableau Ps tel que $Ps[v][i] = Pos_k(G, v, i)$ (le nb. d'occurrences de v dans la position i)

1. mettre tous les éléments de Pt et Ps à 0
2. pour chaque sommet $v \in V$ faire
 - 2.1 $V_{extension} \leftarrow \{u \in N(v) : u > v\}$
 - 2.2 $V_{Subgraph} = \{v\}, E_{Subgraph} = \emptyset$
 - 2.3 appeler $ExtendSubgraph(V_{Subgraph}, E_{Subgraph}, V_{Extension}, v, Pt, Ps, k)$
3. retourner

ExtendSubgraph

Entrée :

- un nombre entier positif $k \leq 5$,
- deux ensembles $V_{Subgraph} \subseteq V$ et $E_{Subgraph} \subseteq E$ contenant les sommets et les liens déjà ajoutés au sous-graphe,
- un ensemble de sommets $V_{extension}$ contenant les sommets qui peuvent être ajoutés au sous-graphe,
- un sommet v où la construction du graphe a commencé,
- deux tableaux Pt et Ps qui seront mis-à-jour par la procédure

1. si $|V_{Subgraph}| > k$ retourner
2. si $|V_{Subgraph}| > 0$ appeler $IndexPattern(V_{Subgraph}, E_{Subgraph}, Pt, Ps)$
3. tant que $V_{Extension} \neq \emptyset$
 - 3.1. prendre un sommet w choisi aléatoirement dans $V_{Extension}$
 - 3.2. $V'_{Extension} = V_{Extension}$
 - 3.3. $E'_{Subgraph} = E_{Subgraph}$
 - 3.4. pour chaque $u \in N(w) : u > v$
 - si $u \in V_{Subgraph}$ ajouter (u, w) à $E'_{Subgraph}$ //ajouter tous les liens de w vers le sous-graphe
 - sinon si $u \notin N(V_{Subgraph})$ ajouter u à $V'_{Extension}$
 - 3.5. appeler $ExtendSubgraph(V_{Subgraph} \cup \{w\}, E'_{Subgraph}, V'_{Extension}, v, Pt, Ps, k)$

IndexPattern

Entrée : Un ensemble de sommets $V_{Subgraph}$, un ensemble de liens $E_{Subgraph}$ et deux tableaux Pt et Ps qui seront mis-à-jour par la procédure

1. parcourir l'ensemble $E_{Subgraph}$ et noter chaque occurrence de chaque sommet
//ainsi calculant les degrés des sommets
2. créer un tableau D contenant les degrés des sommets
3. pour chaque lien $(a, b) \in E_{Subgraph}$ ajouter $degré(b)$ à $D(a)$ et $degré(a)$ à $D(b)$
// ainsi calculant les voisin-degrés
4. trier D et l'écrire comme un nombre
5. trouver le pattern P avec ce numéro et incrémenter $Pt(P)$
6. pour chaque sommet u
 - trouver la position i (dans le pattern P) avec le même voisin-degré et incrémenter $Ps[u][i]$

Dans le Chapitre 7 nous appliquons la méthode à un graphe réel avec $2.7M$ sommets et $6.4M$ liens et nous donnons une complexité empirique de notre méthode pour ce graphe-là. L'exécution de notre implémentation C++ de la méthode prend 31 minutes sur un ordinateur de configuration standard avec un processeur de 2.8GHz et 4Go RAM.

Algorithm 6 *CharacterizeLocalStructure.* Caractérise la structure locale autour de chaque sommet dans un (grand) graphe

Entrée : Un graphe simple non-dirigé $G = (V, E)$ et un nombre entier positif $k \leq 5$

1. créer un tableau A de $|V|$ nombres entiers et les mettre à -1
 2. pour chaque sommet $v \in V$
 - 2.1 initialiser V_v et E_v à l'ensemble vide
 - 2.2 pour chaque sommet u dans $N(v)$, mettre $A[u]$ à v
 - 2.3 pour chaque sommet u dans $N(v)$
 - 2.3.1 ajouter u à V_v
 - 2.3.2 pour chaque sommet w dans $N(u)$ tel que $w < u$
si $A[w] = v$ alors ajouter (w, u) à E_v
 - 2.4 appeler $\text{CharacterizeWithPatterns}((V_v, E_v), k)$
-

B.5 Applications de la méthode

Le but de la méthode que nous avons présentée ici est de caractériser la façon dont un sommet est connecté au réseau. C'est une méthode pour l'analyse de la structure locale du réseau qui produit une caractérisation de chaque sommet. Son but n'est pas de donner un classement ou un ordre de sommets, mais simplement de montrer comment ils sont connectés au réseau. Cela peut être utile dans plusieurs situations. D'abord, comme n'importe quelle méthode de caractérisation, il améliore notre connaissance des sommets du réseau. Deuxièmement, la caractérisation des sommets obtenue peut être comparée à d'autres propriétés des sommets : s'il y a une corrélation, on peut utiliser l'une pour prévoir les autres. Cela est pratique quand il y a des données manquantes parce que quelques unes des propriétés peuvent être déduites des autres. Troisièmement, il y a des situations où une analyse locale est la meilleure façon d'étudier le problème. C'est le cas des données obtenues indépendamment pour des personnes différentes, où le réseau "global" contenant toutes les personnes est inconnu (comme par exemple dans le cas des études sociologiques où les données sur chaque personne sont obtenues par des entretiens individuels et il n'y a aucune collection du réseau entier). Dans ce cas on peut vouloir étudier le réseau dans lequel les individus sont inclus, mais, comme il n'y a aucun réseau global, on ne peut pas faire une analyse de réseau globale ou intermédiaire classique.

Une autre situation où l'étude de la structure locale est appropriée ce sont les réseaux où "l'importance" des noeuds est locale. Dans la situation opposée, il y a des réseaux où certains noeuds sont importants pour le fonctionnement du réseau entier. Prenons par

exemple le cas du réseau de chemins de fer d'un pays ; dans ce cas il est important d'analyser les noeuds dans le contexte du réseau global : il y a quelques noeuds (des stations de train) qui sont importants pour le réseau entier parce qu'ils connectent les différentes parties du pays. Dans ce cas une analyse locale n'est pas suffisante, on a besoin d'utiliser des mesures qui prennent en considération le réseau entier. Aussi, pour les réseaux sociaux en ligne, la perspective globale peut être utile. Dans ces cas, les utilisateurs sont visible dans le réseau entier : ils peuvent être vus et contactés par n'importe quel autre utilisateur du réseau. Souvent il y a une notion de popularité, où les gens essaient d'améliorer leur visibilité et où les supporteurs peuvent se connecter à eux. Cependant, une analyse locale peut aussi apporter des informations importantes. On peut analyser par exemple les liaisons créées par des personnes différentes avant un certain moment dans le temps ; celle-ci est une analyse locale qui relève les relations star-fan (exprimées par des liens).

Une approche locale est utile surtout dans des réseaux où l'importance et la visibilité des noeuds sont locales. Prenons par exemple le cas des communications par téléphone portable. Ici les gens ne peuvent pas être contactés par n'importe qui étant donné que les numéros de téléphone portable ne sont pas publics. Et même si c'était le cas, d'habitude les gens n'appellent pas d'autres gens juste parce que ceux-ci sont connus. Il n'y a aucune mesure de popularité dans ce réseau (par rapport aux plateformes en ligne où des différentes statistiques à propos de l'activité des gens et de leur popularité sont souvent disponibles). Les gens ont d'habitude des appels téléphoniques parce qu'ils ont vraiment quelque chose à discuter avec l'autre personne et pas parce qu'ils sont les supporteurs de cette personne. Dans ce cas les gens à quelques pas (peut-être 2 suffisent) d'une personne ne connaissent pas cette personne ; l'existence de cette personne n'a aucune importance pour eux. Pour des tels réseaux la caractérisation des noeuds en prenant en considération le réseau entier peut ne pas être très utile : quelqu'un avec une grande (disons *betweenness*) centralité peut être moins important que d'autres personnes. Sa présence dans le réseau est sûrement importante pour plusieurs personnes mais ces personnes sont le plus probablement près de lui dans le réseau. Si cette personne quitte le réseau, la grande majorité des individus dans le réseau ne remarquera même pas le changement. Pour des tels réseaux la méthode présentée ici est plus appropriée que d'autres types d'analyse prenant en considération le réseau entier (au moins quand on caractérise un noeud donné).

Finalement, cette méthode peut être utilisée pour calculer une certaine équivalence ou similarité des sommets, des notions très importantes pour la définition de rôles sociaux joués par les noeuds d'un réseau. Une relation d'équivalence possible est la *k*-pattern équivalence que nous avons définie dans la Section B.3. Si l'on veut calculer des sommets similaires (au lieu d'équivalents), on peut calculer une certaine distance entre les *k*-pattern vecteurs des sommets (aussi définis dans la Section B.3). Nous discuterons cette approche et quelques applications dans le Chapitre 8.

B.6 Comparaison avec d'autres mesures

Soulignons d'abord l'équivalence entre plusieurs notions quant à un sommet v , dans le contexte du graphe entier et dans son réseau égocentré (voir le Tableau B.1). Par exemple,

Table B.1 – Notions équivalentes pour un sommet v : dans le graphe total et dans le réseau égocentré.

graphe G	réseau égocentré $Eg(v)$
degré de v	nombre de sommets
nombre de triangles contenant v	nombre de liens
nombre de cliques-4 contenant v	nombre de triangles

le degré de v dans le graphe G correspond au nombre de sommets dans le réseau égocentré $Eg(v)$. De plus, le coefficient de clustering du noeud v est égal à la densité de son réseau égocentré, comme le nombre de triangles contenant le noeud est égal au nombre de liens entre ses voisins, et tous les deux sont égaux à $\binom{d}{2}$ où d est le degré de v .

Patterns versus centralité. Comme nous présentons dans la Section 3.1, la centralité des sommets est une mesure de leur importance dans le réseau. D’habitude on calcule la centralité de tous les sommets du graphe pour produire un classement des sommets. Il y a plusieurs définitions de centralité : la centralité de degré, la betweenness, la closeness, le page-rank, la centralité vecteur propre etc. Hormis la centralité de degré (qui est simplement le degré du noeud), toutes les autres mesures prennent en considération le graphe entier. Comme expliqué dans la section précédente, le but de la méthode présentée ici est de produire une caractérisation locale des sommets. C’est la principale différence entre notre méthode et les différentes définitions de centralité : le but n’est pas le même. Une autre différence vient du contexte d’application des méthodes : tandis que les différentes mesures de centralité doivent avoir le réseau entier pour calculer la centralité d’un noeud, notre méthode a besoin seulement des voisins du noeud et des liens entre eux, donc elle peut être appliquée seulement à quelques parties du graphe si l’on ne connaît pas les autres. Finalement, les centralités betweenness et closeness peuvent être difficilement calculées dans des grands réseaux comme leur complexité temps est $O(nm)$. Au contraire, comme expliqué plus tôt, notre méthode peut être facilement appliquée à des grands réseaux.

Dans une approche différente, on pourrait calculer la centralité des sommets présents dans chaque réseau égocentré, donc celle des voisins de chaque noeud, et comparer entre elles les centralités des différents voisins. Nous rappelons que dans notre méthode nous calculons le k -vecteur de position de chaque voisin pour voir comment les différents voisins sont placés les uns par rapport aux autres. Le vecteur de position est une mesure différente de la centralité. Il reflète la relation de chacun des voisins avec les autres voisins, placés à au plus 5 pas de lui. C’est plutôt une mesure de la façon dont les différents voisins sont placés et connectés dans le réseau que de leur rang ou importance. Regardons par exemple le graphe dans la Figure B.8 et supposons que c’est le réseau égocentré d’un certain noeud. Les sommets x et z ont degré 4, le sommet y a degré 2 et la centralité betweenness de x, y et z est 27, 28 et 24 respectivement. Si l’on a un classement des sommets (y est plus central que x et x est plus central que z), on ne sait pas comment ces noeuds sont connectés au réseau. De plus, on pourrait affirmer que c’est x et pas y qui a une position plus importante dans le réseau égocentré parce qu’il connecte 4 sommets non-reliés directement. Cela n’est

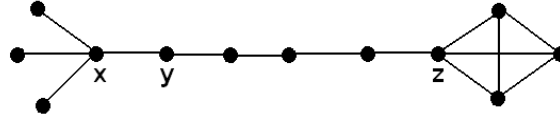


Figure B.8 – Un exemple de différence entre centralité et vecteurs de position.

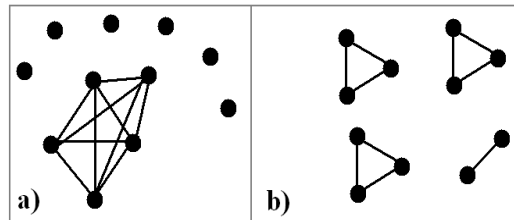


Figure B.9 – Deux réseaux égocentrés qui ont le même nombre de sommets, de liens et le même coefficient de clustering.

montré ni par le degré, ni par la centralité betweenness. En appliquant la méthode que nous avons présentée ici, on sait que x est le centre d'une étoile avec 5 sommets et qu'il appartient à un chemin avec au moins 6 sommets. Il est aussi clair que y est connecté par un lien au centre d'une étoile et qu'il est dans le centre d'un chemin. Quant à z , on sait qu'il appartient à une clique-4 et à un chemin avec au moins 6 sommets. Pour résumer, la méthode que nous avons présentée ici et les mesures de centralité ont des buts différents et sont utiles dans des situations différentes.

Patterns versus densité et coefficient de clustering. La densité du réseau égocentré d'un sommet (ou son coefficient de clustering) est une première caractérisation du sommet et de la façon dont il est connecté au réseau. Pour une caractérisation plus détaillée on peut calculer aussi le coefficient de clustering du réseau égocentré défini comme la moyenne du coefficient de clustering des sommets du réseau égocentré. L'énumération de patterns dans les réseaux égocentrés fournit cependant une description plus riche de la structure locale du réseau que ces deux mesures. Encore une fois, elle décrit *comment* les différents voisins du sommet sont disposés, dans quel type de structures ils sont intégrés. Par exemple, imaginons que les deux réseaux dans la Figure B.9 sont les réseaux égocentrés de deux sommets donnés. Ces réseaux égocentrés ont le même nombre de noeuds, de liens et le même coefficient de clustering. Ces mesures ne capturent pas les différences entre ces deux graphes, tandis que l'énumération de patterns si.

K-pattern équivalence versus d'autres équivalences de sommets. Dans la Section 4.1 nous présentons les équivalences structurelle, automorphique et régulière, probablement les plus connues équivalences de sommets. Ces notions, utilisées pour définir des rôles sociaux, sont trop strictes pour des grands réseaux réels. La k-pattern équivalence que

nous avons définie dans la Section B.3 est incluse dans l'équivalence structurelle et automorphique. Cela s'appuie sur les observations simples que les sommets qui ont exactement les mêmes voisins dans le réseau (donc sont structurellement équivalents) ont des réseaux égocentrés identiques, donc des vecteurs de paramètres (en anglais *feature vectors*) identiques et sont donc k -pattern équivalents, pour tout k . Aussi, les sommets automorphiquement équivalents ont des réseaux égocentrés isomorphes, donc des vecteurs de paramètres identiques et sont ainsi k -pattern équivalents, pour tout k . Pour les deux définitions, la réciproque n'est pas toujours vraie, donc on peut dire que la k -pattern équivalence est incluse dans les équivalences structurelle et automorphique. Cela signifie que la k -pattern équivalence est moins stricte que ces deux relations ; cependant elle n'est toujours pas assez flexible pour des réseaux réels. Quelques adaptations des k -pattern vecteurs pour calculer la similarité des sommets dans des graphes de terrain seront discutées dans le Chapitre 8.

B.7 Conclusions du chapitre

Nous avons présenté dans ce chapitre une méthode pour l'analyse de la structure locale d'un graphe autour de chaque sommet. Cette méthode fournit une description riche de la façon dont un noeud donné est connecté au graphe et aussi de la façon dont ses voisins sont placés les uns par rapport aux autres. Elle peut être appliquée aussi bien à des petits réseaux qu'à des grands et même à des fractions de réseaux. Dans les chapitres suivants nous appliquons cette méthode à deux réseaux sociaux, le premier modélisant l'activité sur une plateforme en ligne et le deuxième modélisant des communications par téléphone portable. Dans le premier cas nous étudions la relation entre la popularité d'utilisateurs et la structure du réseau dans lequel ils sont intégrés, tandis que dans le deuxième cas nous comparons la façon dont les sommets et leurs voisins sont placés dans le graphe à d'autres informations (âge, sexe, intensité de communication) sur les utilisateurs de téléphone portable.

Index

A

adjacency
 adjacency list, 38, 92
 adjacency matrix, 38, 51, 59, 93
automorphism, 38, 87

B

breadth-first search, 38, 52, 55, 75, 102

C

centrality, 53, 97
 betweenness, 55, 87, 97
 closeness, 55, 97
 degree centrality, 55, 97
 eigen vector, 55, 97
 page rank, 55, 97
classification, 42
cluster
 center, 40, 138
clustering, 39, 130, 137
 hierarchical clustering, 41, 142
clustering coefficient, 50, 57, 97, 98, 117
community, 53, 76
complexity, 38, 118
connected component, 37, 52, 77, 117
correlation, 43, 102
 Pearson correlation, 43
covariance, 43
cycle, 37

D

decision tree, 43, 147
degree, 37, 50, 55
 degree combination, 88, 93
 degree distribution, 46, 60, 71, 76, 79,
 117
 neighbor-degree, 88, 93

density, 38, 76, 97, 98
depth-first search, 38, 60
diameter, 38, 48, 77
 average diameter, 49, 77
 effective diameter, 48, 77
distance, 37, 65
 Euclidian distance, 40, 133
 Manhattan distance, 40

E

ego, 68, 123, 145
egocentred network, 85, 106, 117, 131
equivalence
 automorphic equivalence, 67, 85, 99
 k-pattern equivalence, 97, 99, 130
 pattern-frequency equivalence, 133
 position equivalence, 85
 regular equivalence, 68, 99
 structural equivalence, 66, 99
Erdos-Renyi model, 46, 57

F

feature vector, 39

G

graph, 37
 complement graph, 37, 137
 directed graph, 37
 random graph, 46, 58, 60, 72
 simple graph, 37

H

hop-plot, 49

I

isomorphism, 38, 59

K

k-means, 40, 137, 138, 152
 k-nearest neighbors, 42
 k-pattern vector, 92, 97, 130
 Kohonen SOM, 41, 103, 149

L

linear regression, 43
 lognormal distribution, 48

N

neighbor, 37, 85
 neighbor-degree, 88, 93
 neighborhood, 37
 network
 network closure, 65
 network model, 57
 network motifs, 60, 120

P

p-value, 44, 145
 path, 37, 55, 57
 characteristic path length, 49
 path length, 37
 patterns, 85, 106, 117
 frequent patterns, 59, 120
 k-patterns, 85
 PCA, 41
 position, 85
 central, 87, 109, 124
 intermediate, 87, 109, 124
 peripheral, 87, 109, 124
 position equivalence, 85
 position vector, 87, 109, 117
 power-law, 46, 51, 58, 76
 exponential cutoff, 47
 preferential attachment, 57, 77

R

roles, 66, 99, 130

S

small world, 50, 65
 social capital, 65, 76
 standard deviation, 43, 135

strength of ties, 65
 structural holes, 65
 subgraph, 38, 59, 60
 induced subgraph, 38, 85

T

test
 χ^2 test, 44, 145
 ANOVA test, 44, 145
 hypothesis test, 44
 Mantel test, 102
 multiple comparison test, 44, 146
 transitivity ratio, 51
 triangle, 38, 50, 58, 93, 117, 131