

Structure of neighborhoods in a large social network

Alina Stoica
Orange Labs and Liafa
Paris, France
Email: stoica@liafa.jussieu.fr

Christophe Prieur
Liafa
Paris, France
Email: prieur@liafa.jussieu.fr

Abstract—We present here a method for analyzing the neighborhoods of all the vertices in a large graph. We first give an algorithm for characterizing a simple undirected graph that relies on enumeration of small induced subgraphs. We make a step further in this direction by identifying not only subgraphs but also the positions occupied by the different vertices of the graph, being thus able to compute the roles played by the vertices of the graph. We apply this method to the neighborhood of each vertex in a 2.7M vertices, 6M edges mobile phone graph. We analyze how the contacts of each person are connected to each other and the positions they occupy in the neighborhood network. Then we compare the intensity of their communications (duration and frequency) to their positions, finding that the two are not independent. We finally interpret and explain the results using social studies on phone communications.

I. INTRODUCTION

The study of social networks has changed a lot since the early pioneering works of anthropologists who decided to focus on relationships instead of individuals [1], [2], [3]. After the technical framework of social network analysis was settled in the 1970's by the combination of mathematical tools such as graph theory, algebra and statistics [4], [5], [6], the field has been again shaken with the exponential growing of the size of relational databases coming with the development of communication tools. The tremendous research activity on the structure of the World-Wide Web [7], [8] that have predated Google's PageRank algorithm [9] have given birth to a new object of study, namely *complex networks*, due to the common properties found to be shared not only by the graph of the WWW [10], [11] but also by many networks appearing in various contexts (biology, linguistics, economics and, of course, social networks) [12], [13].

There is thus a wide gap between these kinds of studies of the global structure of huge networks and qualitative studies of personal networks, sometimes built from face-to-face interviews (for a historical survey of this trend, see [14]), even though more and more such studies now take as data personal networks scraped from internet's social network services [15]. Inbetween, the classical problem of identifying roles in a (possibly quite large) network, introduced in the 1970's as one of the main tools of social network analysis [16], relies on the fact that some nodes have similar positions in the sense that they are linked to the same other nodes, which is defined as the so-called *structural equivalence* of nodes, or the more general notion of *regular equivalence*, where two nodes are equivalent if the neighbors of the two are equivalent to each

other [17].

Our work is at the intersection of these three research trends: we study the roles of nodes in the personal networks of *all* individuals of a large (thus 'complex') network. In [18] (in French), we already compared to a classical ethnographic study what can be achieved in terms of qualitative analysis with such a large-scale (2 million) collection of personal networks.

Now to address the issue of roles, we devised a method relying on a very popular data mining problem: the search for frequent subgraphs in a given (possibly large) graph. On this issue, some authors considered that frequent subgraphs are the ones that appear in a given graph (or set of graphs) more often than a chosen threshold. Some algorithms [19], [20] extend the apriori-based candidate generation-and-test approach, while others [17], [21] use a pattern-growth approach. More recently, several algorithms have been proposed for significant graph pattern mining [22], [23]. Milo *et al* [24] used another approach to find interesting patterns. They compared the frequency of subgraphs with the ones appearing in randomly generated graphs that share some properties of the network. Several methods for an efficient counting of subgraphs with a given maximal size have been proposed since. Here, we apply the method introduced by Wernicke [25] to the neighborhood of each vertex of the given network. This allows us to compute in the same time the subgraphs that appear more frequently than a chosen threshold and the ones that appear more frequently than in randomly generated networks. Unlike previously done, to our knowledge, we are also able to identify the positions that the different neighbors occupy and therefore the roles they play.

We apply this method to a large network built from mobile phone communications. Of course, there are many forms of social interactions between two people: face-to-face interactions, emails, instant messages, (fixed) telephone, 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 [26], hence the relevance of analyzing a mobile phone network in the search of understanding the underlying social network.

Different properties have been already identified in large mobile phone networks [27], [28]. Onnela *et al* [27] show with no surprise that the distributions of degree and of the duration of calls are power-laws. More strikingly they also give a definition for the strength of ties depending on the duration of calls and they analyze the connection between the strength and the connectivity or the community structure.

As in complex network studies, all of these properties are global, characterizing the structure of the mobile phone graph as a whole. Here, our aim is to identify the local structure, the way the persons contacted by a given individual (ego) are connected to each other relatively to their "importance" to ego. In order to do that, we use the frequency and the total duration of communications between ego and each of his contacts.

The paper is organized as follows. After recalling some basic definition on graphs (Section II), we detail in Section III a formal framework to address the issue of characterizing a graph in terms of a position equivalence along with an algorithm to do it. The main algorithm to characterize all the neighborhoods of a large graph is given in Section IV and in Section V we apply it to a mobile phone graph and discuss the results by comparing them to an ethnographic study on communication tools.

II. PRELIMINARIES

Let $G = (V, E)$ be a graph; V is the set of its vertices, $E \subseteq V \times V$ is the set of its edges. We define its *size* as $|V|$, the number of its vertices. Two vertices $u, v \in V$ are *adjacent* in G if $(u, v) \in E$. The graph G is *undirected* if, for all $u, v \in V$, there is no difference between (u, v) and (v, u) , it is *connected* if there exists a finite path between every two vertices and it is *simple* if there is no multiple edge and no self-loop ($(v, v) \notin E$, for all $v \in V$). For a vertex $v \in V$, we denote by $N(v) = \{u \in V, (u, v) \in E\}$ its *neighborhood*, by $N[v] = N(v) \cup \{v\}$ its closed neighborhood and by $d(v) = |N(v)|$ its *degree*. The *betweenness centrality* [29] of a vertex v is defined as $c(v) = \sum_{s, t \in V_G, v \notin \{s, t\}} \frac{\delta_v(s, t)}{\delta(s, t)}$ where $\delta(s, t)$ denotes the number of shortest paths from s to t and $\delta_v(s, t)$ denotes the number of shortest paths from s to t that pass through v .

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.

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 $(u, v) \in E_G$. For a graph G and a positive integer k , Wernicke [25] proposed the algorithm $ESU(G, k)$

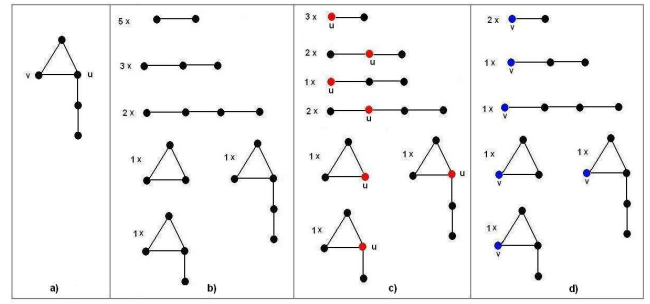


Fig. 1. A graph (a), its connected induced subgraphs (b) and the positions of the vertices u (c) and v (d).

that efficiently enumerates all the connected induced subgraphs of G with exactly k vertices.

III. COMPUTING POSITIONS OF VERTICES

This section introduces a method to characterize a graph and its vertices. Given a graph, we enumerate all its connected induced subgraphs with size at most 5 up to isomorphism. Then, for each vertex of the graph, we compute the position it occupies in each one of the found subgraphs. For example, for the graph in Figure 1 a, the number of its different induced subgraphs and the positions occupied by the vertices u and v are presented in Figure 1 b, c and d respectively.

A. Definitions

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 graph in Figure 2, 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 30 non-isomorphic graphs with at most 5 vertices and at least 1 edge (see Figure 2); we call these graphs *patterns* and we denote by \mathcal{C} their set. For a pattern $C \in \mathcal{C}$, we denote by $P(C)$ the set of its positions. There are 73 different positions in the 30 patterns; we denote by \mathcal{P} their set. We sort in ascending order the positions of a same pattern by the betweenness centrality of their vertices, then by their 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*.

We now give two definitions and a lemma that allow us to check efficiently if two graphs with at most 5 vertices are isomorphic and if two vertices in such a graph are position equivalent.

Given a graph G and a vertex v of G , we call *neighb-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 *degrees combination* of the graph G the ascending sorted list of the neighb-degrees

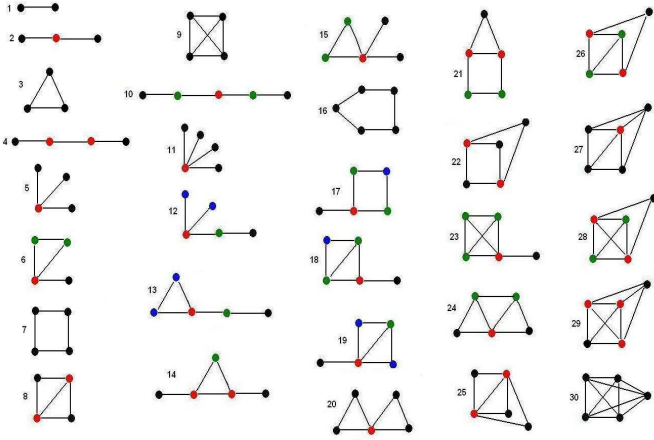


Fig. 2. The set of patterns and their positions. The order of the colors is black, blue, green and red corresponding to the ascending order of betweenness centrality and degree.

of its vertices. Note that for a graph G with n vertices and m edges one computes the neighb-degrees of all the vertices of G in $O(m)$ time, then its degrees combination in $O(n \cdot \log n)$ time.

Lemma 1: Two graphs G and H with at most 5 vertices are isomorphic if and only if their degrees combination are identical.¹ Moreover, two vertices $u, v \in V_G$ are position equivalent if and only if they have the same neighb-degree.

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

So, given a connected graph with $n \leq 5$ vertices and m edges, one can find to which pattern it corresponds (i.e. to which of the 30 graphs in Figure 2 it is isomorphic) and check if two of its vertices are position equivalent by using only the neighb-degrees of the vertices. The isomorphism class of the graph can be thus computed in time $O(m + n \cdot \log n)$ and the positions of its vertices in time $O(m)$.

B. The algorithm

We propose in Algorithm 1 a method to characterize an undirected simple graph G . First we compute the number of occurrences of the 30 patterns as induced subgraphs of G . Then we compute, for each vertex of G , its number of occurrences in the positions of the different patterns. The choice of limiting the size of the researched induced subgraphs at 5 is motivated by complexity reasons (for instance the number of non-isomorphic connected graphs with at most 6 vertices is 142).

For a graph G and a pattern $C \in \mathcal{C}$, we denote by $Sub(G, C)$ the number of occurrences of the pattern C as an induced subgraph of G . For a graph G , a position $P \in \mathcal{P}$ and a vertex $v \in V_G$, $Pos(G, P, v)$ counts the number of induced subgraphs of G that contain v in the position P .

¹This is not true for the graphs with 6 vertices: there exist two non-isomorphic graphs with 6 vertices that have the same degrees combination.

Algorithm 1 characterize. Characterizes an undirected simple graph

Input: A set of edges representing an undirected simple graph G

Output: An array `Sub` such that $Sub[C] = Sub(G, C)$ and an array `Pos` such that $Pos[v][P] = Pos(G, P, v)$

1. enumerate all the connected induced subgraphs of G of size at most 5 \Rightarrow the set S
 2. for each graph $H \in S$
 - 2.1. find the corresponding pattern C and increment $Sub[C]$
 - 2.2. for each vertex v of H , find its position P and increment $Pos[v][P]$
-

For the first part of the characterization method (line 1), we use the algorithm $ESU(G, k)$ [25] with $k \leq 5$. For the second task (line 2), we apply Lemma 1, so we compute the neighb-degrees of the vertices in each subgraph. The time complexity of this algorithm is linear in the number of connected induced subgraphs of size at most 5 of the input graph: for the first part (the enumeration), see [25]; for the second part, note that it takes a constant time to compute the corresponding pattern and the set of positions of each subgraph found in the first part. As for the space complexity, note that one doesn't need to explicitly build and store the set S , but only one subgraph at a time. When a connected induced subgraph of G of size at most 5 is found, the corresponding pattern and the positions of its vertices are computed before proceeding to the search of another subgraph.

C. Roles of vertices

When counting the number of occurrences of a vertex in the different positions, one checks the way this vertex is connected to the vertices around it. This measure is computed locally, around each vertex, and reflects **the role** a vertex plays in relation to the vertices placed at at most 5 steps from it. This is not a measure of the centrality of vertices, but of their relations with the vertices around them. Look for instance at the graph in Figure 3. The vertices x and z have degree 4, the vertex y has degree 2, while the betweenness centrality of x, y and z is 27, 28 and 24 respectively. One can note that x has an important role in this graph by connecting 4 vertices not directly linked. This is not shown by the degree nor by the betweenness centrality. By applying Algorithm 1 to the graph, one has a clearer idea about how the different vertices are connected to the vertices around them and the roles they play. For instance, it is clear that x is the center of a star with 5 vertices (it appears once in the central position of the pattern 11) 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 (it appears in the peripheral position of the pattern 11) and that it is in the center of a path (it is found 3 times in the central position of the pattern 10). As for z , one knows that it belongs to a 4-clique (it appears once in the pattern 9) and

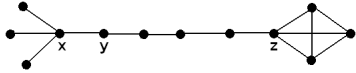


Fig. 3. An example

that it belongs to a path with at least 6 vertices.

IV. NEIGHBORHOODS

In this section, we propose a method (Algorithm *characterize_neighborhoods*) to analyze the local structure of a large graph using the notions introduced in the previous section. For each vertex v of a given large graph G , we first compute the subgraph $Gn(v)$ induced by the neighbors of v , so we need to list the triangles containing v . The latter problem has been extensively studied in [30]; we rely on Algorithm *new-vertex-listing* proposed in this paper in order to compute, for each vertex $v \in G$, the subgraph $Gn(v)$.

We characterize the obtained neighborhood graph $Gn(v)$ using Algorithm 1, so we compute the number of occurrences of each pattern in $Gn(v)$ (the array `Sub` in Algorithm 1) and the positions occupied by the neighbors of v (the array `Pos` in Algorithm 1). Using the arrays `Sub` and `Pos` we update two global arrays S' and P' . The first one contains, for each pattern, its total number of occurrences in the neighborhood graphs of the given large graph. After having associated, using extra-data, different types to the neighbors of each vertex in the large graph, one can compute the second array: the number of occurrences in each position of the vertices of a certain type. We detail the updating of the two arrays in the next section.

Algorithm 2 *characterize_neighborhoods*. *Characterizes the neighborhood of each vertex of a large graph*

Input: An undirected simple large graph G

Output: Two arrays S' and P'

1. create an array A of $|V_G|$ integers and set them to -1
 2. for each vertex v of the graph G
 - 2.1. initialize E 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. for each vertex w in $N(u)$
 - if $A[w] = v$ then add (w, u) to E
 - 2.4. *characterize*(E)
 - 2.5. update S' and P'
-

In the next section, we apply Algorithm 2 to a large social graph. We also present the degree and triangle distributions, important parameters for the complexity of our method.

V. ANALYSIS OF A LARGE SOCIAL GRAPH

A. Description of the graph

We analyze a large graph built from a mobile phone database. The database contains a month of mobile phone

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

TABLE I

DIFFERENT MEASURES FOR THE NUMBER OF VERTICES (n) AND THE NUMBER OF EDGES (m) OF THE 2.7×10^6 NEIGHBORHOOD NETWORKS

communications (phone calls and short messages) between the clients of a same operator in a European country. We build a graph where the vertices are the clients; we connect such two vertices by an undirected edge if each of the two persons has contacted at least once the other person during the recorded month. This way we don't take into consideration the one-way contacts (calls or messages), single events in most of the cases suggesting that the two individuals don't know each other personally. We obtain a graph (that we call GM) with 2.7×10^6 vertices and 6.4×10^6 edges; 83% of its vertices and 99% of its edges belong to the same giant connected component.

For each vertex (ego) v in GM , we study the graph $Gn(v)$ induced by its neighborhood, so we analyze 2.7×10^6 graphs; we denote by $D = \{Gn(v), v \in GM\}$ this set of neighborhood graphs. As expected, most of these graphs have a small number of vertices (this number is equal to the degree of v) while only a small minority have a great number of vertices. The same statement is valid for the number of edges of each graph in D (this number is equal to the number of triangles containing v). Table I contains the minimum, maximum, median and average values of the two parameters, as well as the number of graphs in D where the value of the parameter is greater than 100. Figure 4 contains the distribution of the number of vertices and of the number of edges of the graphs in D . Only 20 graphs (i.e. $7 \times 10^{-4}\%$) have more than 100 vertices and more than 100 edges. The average of the densities of the graphs in D is the clustering coefficient of GM , equal to 0.097.

Performance. The execution of Algorithm 2 on the graph GM takes 31 minutes on a computer with a 2.8GHz processor and 4Gb RAM.

B. Characteristic patterns

We address here the problem of identifying the patterns that are "characteristic" for the set D of neighborhood graphs. There are several possible definitions for a characteristic pattern C for a set of graphs D :

- Def 1. the number of occurrences of the pattern C 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 C as induced subgraph is greater than a given threshold (this is the adopted approach in [22], [19], [20], [21]);
- Def 3. the number of occurrences of the pattern C as induced subgraph is higher for the graphs in D than for randomly generated graphs of same sizes (this approach was introduced in [24]).

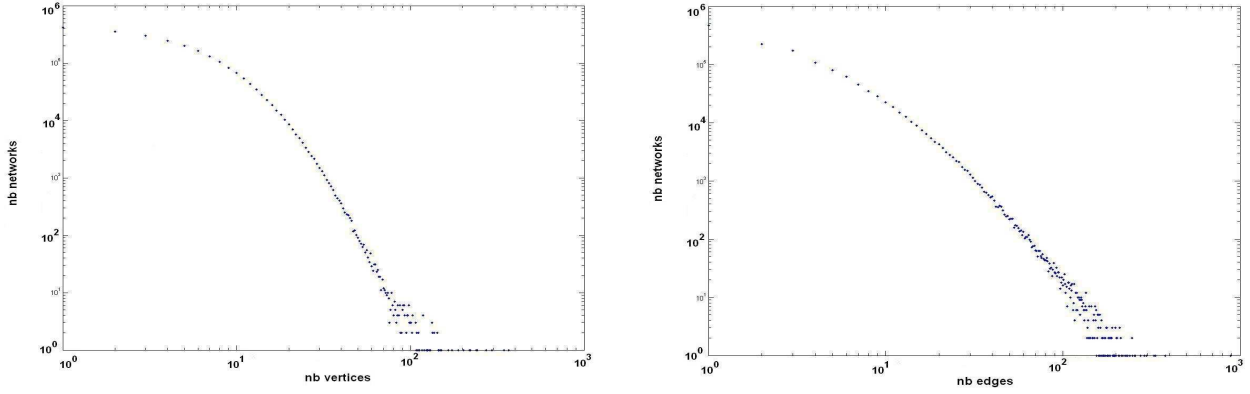


Fig. 4. The distribution of the number of vertices (a) and of the number of edges (b) of the 2.7×10^6 neighborhood networks

Def 1. We computed, for each pattern C with $k \leq 5$ vertices, the number of occurrences of C as induced subgraph 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 C . Figure 5 (left) contains 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). Of course the counting of all the occurrences of a certain pattern gives an advantage to those containing vertices of degree 1.

Def 2. Figure 5 (right) contains, for each pattern C with $k \leq 5$ vertices, the number of graphs in D that contain C 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 C . We observe that the most frequent patterns are the paths, possibly with one extra edge (added to form a star or a triangle).

Def 3. For each connected component of a graph in D we randomly generated connected graphs using the method introduced in [31]. This method computes 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. Note that all the three generations (for $d = 1, 2, 3$) preserve the degree and the clustering coefficient of the graph GM . For each pattern, we computed 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 the graphs in D and the ones with

the smallest values are characteristic for the graphs in R_d . For $d = 1$ and $d = 2$, the same patterns are identified as characteristic (see Figure 6), 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 more dense than the corresponding generated ones. So, in the neighborhood of certain vertices, several neighbors form dense clusters even though they belong to the same connected component; these clusters may correspond to the different groups of contacts of that person. Note however that the two generations don't preserve the clustering coefficients of the graphs in D . When $k = 3$, the clustering coefficient is preserved 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. Besides, the structure of neighborhoods refers not only to the distributions of the different patterns, but also to the roles played by the vertices; this question is discussed in the next section.

C. Positions of the vertices

Recall that by applying Algorithm 2 to the graph GM we computed, for each ego v in GM and each vertex u in the neighborhood $Gn(v)$ of v , the positions of u in $Gn(v)$. We analyze here, for each ego v and each vertex u in $Gn(v)$, the relation between the positions occupied by u in $Gn(v)$ and the quantity of its communications with v . Note that the positions of u are completely determined by its links with the other vertices in $Gn(v)$; the quantity of communication of u with ego v will also be relativized to the quantity of communication between ego and the other persons in his network.

1) *The maximal number of calls:* First, for each ego v , we index his neighbors depending on the number of calls they exchanged with him: the greater the number of calls exchanged with ego, the smaller the index, such that the vertex with the

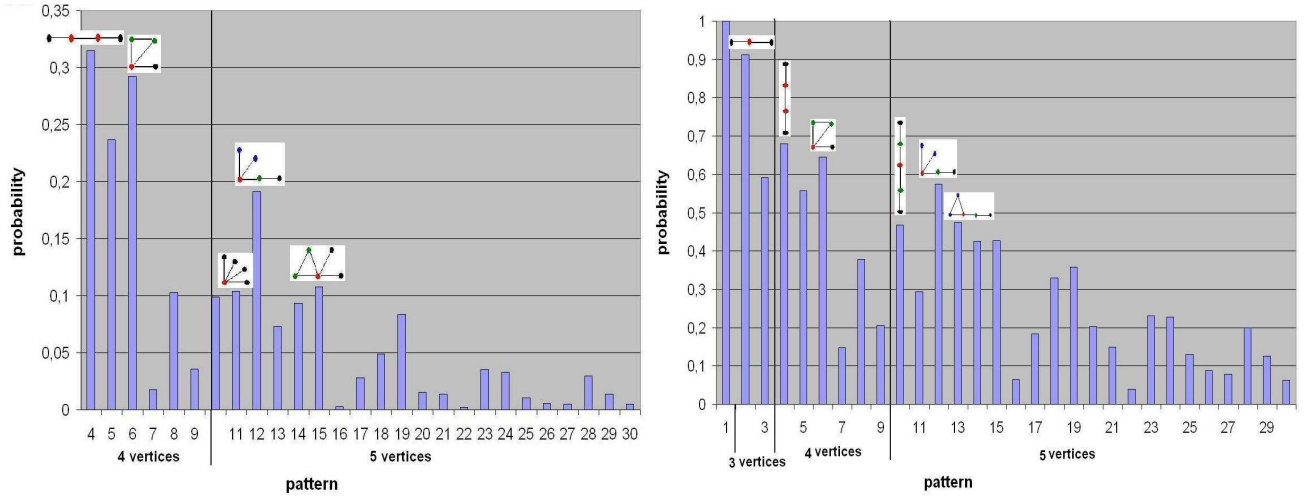


Fig. 5. For each pattern with k vertices, the probability: to be the subgraph induced by k connected vertices in D (left) and to occur in a graph in D that has at least k connected vertices (right)

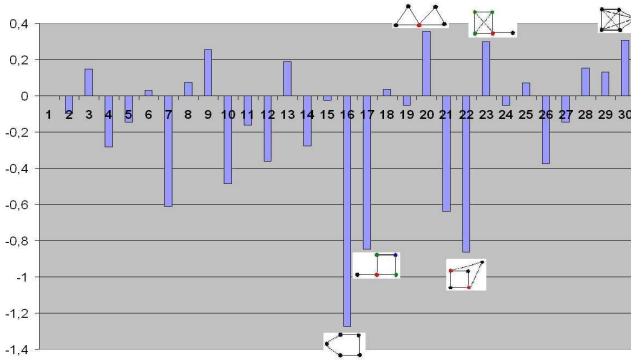


Fig. 6. For each pattern, \log_2 of the ratio between its nb of occurrences in D and in R_2

greatest number of calls has index 1 and the one with the smallest has index $d(v)$.

Let D_5 be the set of graphs in D with at least 5 vertices, i.e. the set of neighborhood graphs of the vertices in GM with degree at least 5. For each graph in D_5 , we study the positions occupied by its vertices with indices 1, 2, 3 and 4 and by a randomly chosen vertex between those with index greater than 4 to which we give the index 0. In order to do that, we answer two questions regarding the entire set D_5 :

- Q1 given a position in \mathcal{P} , which of the five indices occupies this position the most frequently and which one the least frequently?
- Q2 given a pattern $C \in \mathcal{C}$ and an index $i < 5$, in which position $p \in P(C)$ the vertices with index i appear the most frequently and in which one the least frequently?

For an index i , let $I(i)$ be the set of vertices that have index i in the graphs in D_5 along with the corresponding graphs: $I(i) = \{(u, G) \text{ s.t. } u \in V_G, G \in D_5 \text{ and } \text{index}(u) = i\}$. For an index $i < 5$ and a pattern C we count the number of

occurrences of a vertex with index i in the different positions of C : $Nb(i, C) = \sum_{p \in P(C)} \sum_{(u, G) \in I(i)} Pos(G, p, u)$. We now compute the probability that, when a vertex with index i occurs in a position of the pattern C , this position is p :

$$Pr(i, p, C) = \begin{cases} 0 & \text{if } p \notin P(C) \\ \frac{\sum_{(u, G) \in I(i)} Pos(G, p, u)}{Nb(i, C)} & \text{otherwise.} \end{cases}$$

Figure 7 contains these probabilities for all the 5 indices and all the patterns (as in Figure 2) with at least two positions.

Question Q1. We observe that, for all the central positions (the right side of each image), the probability of occurrence of the vertices with index 1 is greater than that of the vertices with index 2, which is greater than that of the vertices with index 3 etc. The opposite situation happens for the peripheral positions (the left side of each image) 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 index 2, 3 or 4.

Question Q2. We observe that the vertices with index 1 occupy most frequently the central positions and least frequently the peripheral ones (the red curves are generally ascending). 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 indices 2, 3 and 4 have a tendency placed between these two.

So, when they appear in a pattern, the vertices with index 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 indices 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 counting the positions occupied by a node is not equivalent

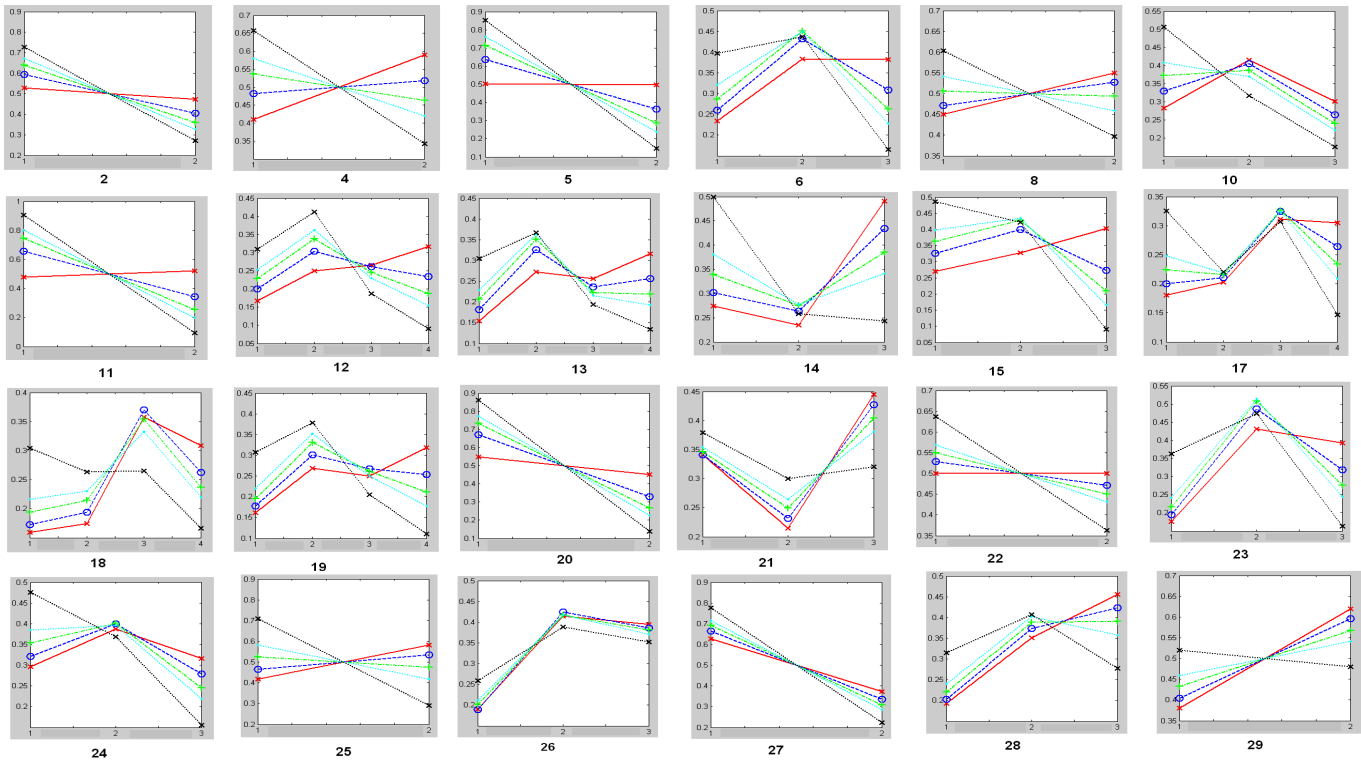


Fig. 7. For each pattern C and each position $p \in P(C)$ (x-axis), the probability (y-axis) of occurrence of a vertex with index i in p : index 1—red dots, 2—blue dots, 3—green dots, 4—cyan dots, 0—black dots. In each image, the x-axis goes from peripheral (left) to central positions (right).

to measuring its centrality: even if a node is not the most central (in terms of betweenness centrality), it may occupy the central position of some patterns. This can be shown, for instance, by looking at the graphs in D_5 where the vertex with index 4 is the most central. We compute, as before, the probabilities Pr for the vertices in these graphs. Even if the vertex with index 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 with index 1, 2 or 3.

2) *The maximal sum of duration of calls:* We analyze, for the network of each ego, 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 index 1 of the previous section) is also the person that has the greatest total duration. In the other cases, we gave index 1 to the vertex with the greatest number of calls and index 2 to the vertex with the greatest sum of duration of calls. We also randomly chose 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 index 2 to occupy the central position is smaller than that of the vertices with index 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 index 2 tend to occupy the intermediate positions.

D. 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 that 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 that has a more important role.

In [32], Licoppe and Smoreda analyzed the relation between social networks, exchanges between actors and communication tools using databases of telephone calls, Internet traffic and several interviews focusing on the use of telephone. They identified a first pattern of communication, that of "connected presence", where 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, which confirms the soundness of our method.

The second pattern identified by Licoppe and Smoreda is

that of "intermittent presence", where 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 [33], 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. It would be interesting to study in more details the position of this actor in ego's network, the way it is connected to the other neighbors. The computation of its number of occurrences in the positions of the different patterns seems a good way to do that.

VI. CONCLUSIONS AND PERSPECTIVES

We presented in this paper a method for analyzing the local structure of large graphs that we applied to a $2.7M$ vertices, $6M$ edges mobile phone graph. In the neighborhood of each vertex (called ego) of the graph, we listed all the patterns and we identified the characteristic ones. Then we addressed the notion of roles in a graph. Using the positions occupied by each vertex, we wanted to find how they connect to each other, how they are placed and how important they are in the local structures they form with the other vertices. Our goal was not to define a new measure for the centrality of vertices, but rather to give a method for analyzing in detail the way the vertices connect to the rest of the network. As a next step, we intend to use the method introduced here to find vertices that occupy similar positions in the network. Thus we would be able to group together vertices that connect in the same way to the network. The partition of vertices into classes based on their positions could then be confronted with other properties of the actors that are not a consequence of the network (for instance age, gender, job etc.).

It would also be interesting to apply this method to other kinds of large graphs, for instance to online communities where the neighborhoods are denser but the connections between people weaker.

REFERENCES

- [1] E. Bott, *Family and Social Network*. London: Tavistock, 1957.
- [2] J. Boissevain, *Friends of Friends, Networks, Manipulators and Coalitions*. Oxford: Basil Blackwell, 1974.
- [3] F. Barth, *Political leadership among the Swat Pathans*. London: Athlone Press, 1959.
- [4] F. Lorrain and H. White, "Structural equivalence of individuals in social networks," *Journal of Mathematical Sociology*, vol. 1, pp. 49–80, 1971.
- [5] J. Scott, *Social Network Analysis*. London: Sage, 1992.
- [6] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

- [7] D. Gibson, J. M. Kleinberg, and P. Raghavan, "Inferring web communities from link topology," in *ACM Conference on Hypertext and Hypermedia*, 1998, pp. 225–234.
- [8] K. Efe, V. Raghavan, C. H. Chu, A. L. Broadwater, L. Bolelli, and S. Ertekin, "The shape of the Web and its implications for searching the Web," 2000.
- [9] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.
- [10] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, 1999.
- [11] D. Watts and S. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, 1998.
- [12] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 167, no. 45, 2003.
- [13] S. Bornholdt and H. G. Schuster, Eds., *Handbook of Graphs and Networks*. Wiley-Vch, 2003.
- [14] B. Wellman, "An egocentric network tale," *Social Networks*, vol. 15, pp. 423–436, 1993.
- [15] B. Hogan, *The Handbook of Online Research Methods*. Sage: Thousand Oaks, CA, 2008, ch. Analyzing Social Networks via the Internet.
- [16] H. White, S. Boorman, and R. Breiger, "Social structure for multiple networks i. blockmodels of roles and positions," *American Journal of Sociology*, vol. 81, 1976.
- [17] S. Borgatti and M. Everett, "The class of all regular equivalences: Algebraic structure and computation," *Social Networks*, vol. 11, no. 1, pp. 65–88, March 1989.
- [18] C. Prieur, A. Stoica, and Z. Smoreda, "Extraction de réseaux sociaux dans un (trs grand) réseau social," *Bull. de Methodologie Sociol.*, no. 101, 2009.
- [19] A. Inokuchi, T. Washio, and H. Motoda, "An a priori-based algorithm for mining frequent substructures from graph data," in *PKDD '00*.
- [20] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *ICDM '01*.
- [21] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *ICDM '02*.
- [22] H. He and A. Singh, "Graphrank: Statistical modeling and mining of significant subgraphs in the feature space," in *ICDM '06*.
- [23] X. Yan, H. Cheng, J. Han, and P. Yu, "Mining significant graph patterns by leap search," in *SIGMOD '08*.
- [24] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, October 2002.
- [25] S. Wernicke, "Efficient detection of network motifs," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 3, no. 4, pp. 347–359, 2006.
- [26] C. Haythornthwaite, "Social networks and internet connectivity effects," *Information, Communication and Society*, vol. 8, no. 2, pp. 125–147, June 2005.
- [27] 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*, vol. 104, no. 18, pp. 7332–7336, May 2007.
- [28] J. Candia, M. C. Gonzalez, P. Wang, T. Schoenharl, G. Madey, and A. Barabási, "Uncovering individual and collective human dynamics from mobile phone records," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 6, pp. 224015+, June 2008.
- [29] L. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, 1977.
- [30] M. Latapy, "Main-memory triangle computations for very large (sparse (power-law)) graphs," *Theoretical Computer Science (TCS)*, no. 407, pp. 458–473, 2008.
- [31] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat, "Systematic topology analysis and generation using degree correlations," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 135–146, 2006.
- [32] C. Licoppe and Z. Smoreda, "Are social networks technologically embedded? how networks are changing today with changes in communication technology," *Social Networks*, vol. 27, no. 4, pp. 317–335, October 2005.
- [33] R. Lambiotte, V. Blondel, C. de Kerchove, E. Huens, C. Prieur, Z. Smoreda, and P. V. Dooren, "Geographical dispersal of mobile communication networks," *Physica A*, vol. 387, no. 21, pp. 5317–5325, September 2008.