Uncovering the Nucleus of a Massive Reciprocal Network

Braulio Dumba
e-mail: braulio@cs.umn.edu
address: Department of Computer Science & Engineering
University of Minnesota 4-192 EECS Building 200 Union Street SE, Minneapolis, MN 55455-0159 USA

Zhi-Li Zhang
e-mail: zhzhang@cs.umn.edu
address: Department of Computer Science & Engineering
University of Minnesota 4-192 EECS Building 200 Union Street SE, Minneapolis, MN 55455-0159 USA

Uncovering the Nucleus of a Massive Reciprocal Network

Braulio Dumba · Zhi-Li Zhang

Received: date / Accepted: date

Abstract Google+ is a *directed* online social network where nodes have either *reciprocal* (bidirectional) edges or *parasocial* (one-way) edges. As reciprocal edges play an important role in the structural properties, formation and evolution of online social networks, we study the core structure of the subgraph formed by them, referred to as the *reciprocal network* of Google+ — in a sense, a reciprocal network can be viewed as the stable "skeleton" network of a directed online social network that holds it together. We develop an effective three-step procedure to *hierarchically* extract and unfold the *core* structure of a network by building up and generalizing ideas from the existing k-shell decomposition and clique percolation approaches. Our scheme produces higher-level representations of the core structure of the Google+ reciprocal network and it reveals that there are ten subgraphs ("communities") comprising of dense clusters of cliques lying at the center of the core structure of the Google+ reciprocal network. Together they form the core to which "peripheral" sparse subgraphs are attached. Furthermore, our analysis shows that the core structure of the Google+ reciprocal network is very stable as the network evolves. Our results have implications in the design of algorithms for information flow, and in development of techniques for analyzing the vulnerability or robustness of online social networks.

Keywords Google+ · Reciprocal Network · K-Shell Decomposition · Network Core · Dependence · Hypergraph

B. Dumba · Z. Zhang
Department of Computer Science and Engineering
University of Minnesota, Twin Cities, MN, 55455, USA
E-mail: {braulio,zhzhang}@cs.umn.edu

1 Introduction

Many online social networks (OSNs) are fundamentally directed: they consist of both *reciprocal* edges, i.e., edges that have already been linked back, and *parasocial* edges, i.e., edges that have not been or is not linked back [1]. Reciprocal edges represent the most stable type of connections or relations in directed network – they reflect strong ties between nodes or users [2–4], such as (mutual) friendships in an online social network or following each other in a social media network like Twitter and Google+.

Reciprocity is defined as the ratio of the number of reciprocal edges to the total number of edges in the network, and it is believed that it plays an important role in the structural properties, formation and evolution of online social networks. Hence, this metric has been widely studied in the literature in various contexts, see, e.g., [1,5–9]. Many studies have used reciprocity (a single-valued aggregate metric) to characterize massive *directed* OSNs, which we believe is inadequate. Instead, we consider the *reciprocal graph* (or *reciprocal network*) of a directed OSN – namely, the *bidirectional* subgraph formed by the reciprocal edges among users in a directed OSN (see Fig. 1 for an illustration). In a sense, this reciprocal network can be viewed as the stable "skeleton" network of the directed OSN that holds it together. We are interested in analyzing and uncovering the *core* structural properties of the reciprocal network of a directed OSN, as they could reveal the possible organizing principles shaping the observed network topology of an OSN [5]. For example, using the *core*, we can build network models that can help us to understand the topological features of the nodes and structural properties of the network, as well as, to predict the topological growth of the network and provide upper bounds of the distance between the nodes – see the jellyfish model of the Internet in [24]. Furthermore, unveiling the core structure (referred to as the "nucleus") of a reciprocal network may have implications in the design of algorithms for information flow, and in development of techniques for analyzing the vulnerability or robustness of OSNs (more in Sect. 9).

In this paper, we perform a comprehensive empirical analysis of the "core structure" of the reciprocal network of Google+. Based on a massive Google+ dataset (see Sect. 2 for a brief overview of Google+ and a description of the dataset), we find that out of more than 74 million nodes and ≈ 1.4 billion edges in (a snapshot of) the directed Google+ OSN, more than two-third of the nodes are part of Google+'s *reciprocal* network and more than a third of the edges are reciprocal edges (with a reciprocity value of roughly 0.31). This reciprocal network contains a *giant connected subgraph* with more than 40 million nodes and close to 200 million edges (see Sect. 3 for more details). Existence of this massive (giant connected) reciprocal (sub)graph in Google+ raises many interesting and challenging questions. How is this reciprocal network formed? Does it contain a "core" network structure? If yes, what does this structure look like?

In an attempt to address these questions, we develop an effective three-step procedure to *hierarchically* extract and unfold the *core* structure of Google+'s

reciprocal network[1], building up and generalizing ideas from the existing k-shell decomposition [11] and clique percolation approaches [12], extending our work in [10]: i) We first apply (a modified version of) the k-shell decomposition method to prune nodes and edges of sparse subgraphs that are likely to lie at the periphery of the Google+ reciprocal network (see Sect. 4). The standard k-shell decomposition method has been proposed to extract the "core" of a network, e.g., that of the Internet AS graph [11]. However, directly applying this method to the Google+ reciprocal yields a final graph – a clique of 290 nodes (the maximum clique of the Google+ reciprocal network) that consists of a close-knit community of users in Taiwan – which is unlikely to lie at the "core" of the Google+ reciprocal network (see discussion in Sect. 7, where we show this clique in fact lies more at the outer ring of Google+'s dense core structure). Instead, we introduce a new metric, the dependence value for a node that measures the location importance of a node in a network (see Sect. 5). Then, using this metric we propose a modified version of the k-shell decomposition method by identifying the k_C-index where we should stop pruning the network in order to preserve its core structure. This process yields a dense "core" subgraph of the Google+ reciprocal network with approximately 48K nodes and 6M edges. ii) Given this dense "core" subgraph, we first compute the maximal clique that each node is part of (using a simplified Bron-Kerbosh algorithm), and then form a new *directed* (hyper)graph – a form of clique percolation [12], where the vertices are (unique) cliques of various sizes, and there exists a directed edge from clique C_i to clique C_j if half of the nodes in C_i are contained in C_j (see Sect. 6). This new (hyper)graph provides a higher-level representation of the dense core graph of the Google+ reciprocal network: the intuition is that the maximal clique containing each node v represents the most stable structure that node v is part of, and the directed edge in a sense reflects the "attraction" (or "gravitational pull") that one clique (constellation) has over the other. We find that this (hyper)graph of cliques comprises of 1700+ connected components (CCs). iii) Finally, considering these CCs as the core "community" structures (a dense cluster of cliques) of the Google+ reciprocal network, we define three metrics to study the relations among these CCs in the underlying Google+ reciprocal network: the number of nodes shared by two CCs, the number of nodes that are neighbors in the two CCs, and the number of edges connecting these neighboring nodes (see Sect. 7). These metrics produce a set of new (hyper)graphs that succinctly summarize the (high-level) structural relations among the core "community" structures and provide a "big picture" view of the core structure of the Google+ reciprocal network and how it is formed. In particular, we find that there are ten CCs that lie at the center of this core structure through which the other CCs are most richly connected. We also find that the core structure of the Google+ reciprocal network is very stable as the network evolves (see Sect. 8). We discuss implications and related work in Sect. 9 and Sect 10. In Sect. 11, we conclude the paper with a brief discussion of the future work.

[1] Our methodology can also be applied to others online social networks.

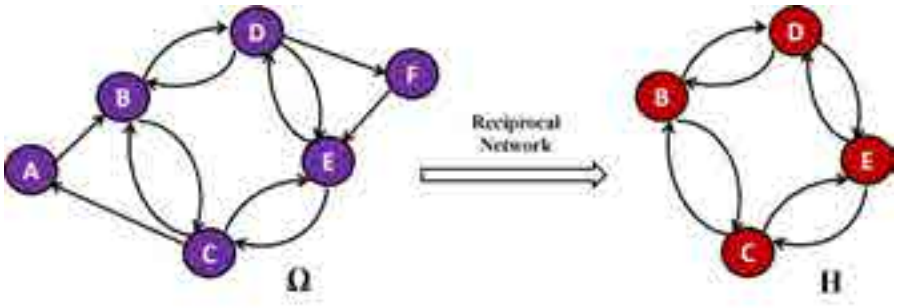


Fig. 1 Illustration of the reciprocal network (H) of a directed graph (Ω). Specifically, (B, C), (C, B), (B, D), (D, B), (D, E), (E, D), (C, E), (E, C) are reciprocal edges; (A, B), (C, A), (D, F), (F, E) are parasocial edges. The reciprocity of Ω is $8/12 = 0.67$.

We summarize the major contributions of our paper as follows. To the best our knowledge, our paper is the first study on the core structure of a "reciprocal network" extracted from a massive *directed* social graph. While this paper focuses on Google+, our approach is also applicable to other directed OSNs.

- We propose a new metric, the dependence value, that measures the location importance of a node in the network. Using this metric, we therefore modify the standard k-shell decomposition method to stop the process earlier, in order to extract a meaningful "core" for social networks
- We develop an effective three-step procedure to *hierarchically* extract and unfold the *core* structure of a reciprocal network arising from a directed OSN.
- We apply our method to the reciprocal network of the massive Google+ social network, and unfold its core structure. In particular, we find that there are ten subgraphs ("communities") comprising of dense clusters of cliques that lie at the center of the core structure of the Google+ reciprocal network, through which other communities of cliques are richly connected; together they form the core to which other nodes and edges that are part of sparse subgraphs on the peripherals of the network are attached.
- We observe that the core structure of the Google+ reciprocal network is very stable as the network evolves: the size of the core communities (hyper)graph increases as the network evolves, as well as, its density. Additionally, the set of nodes that participates in the core is very stable over time, with few percentage of nodes (e.g: 5% and 9%) that move away from the core to the periphery as the network evolves.
- We observe that the number of communities lying at the center of the core structure of the Google+ reciprocal network is also very stable: it increases from 10 to 11 core communities across snapshots $H_1 \rightarrow H_2$ and from 11 to 13 core communities across snapshots $H_2 \rightarrow H_3$ in the core communities (hyper)graphs.

Table 1 Main characteristics of Google+ snapshots: (start-date, duration) – Γ_1: (24-08-12, 17 days), Γ_2: (10-09-12, 11 days) and Γ_3: (20-06-13, N/A)

ID	# nodes	# edges	max(in)	max(out)	reciprocity	density
Γ_1	74,419,981	1,396,943,404	2,289,874	9,981	0.31	2.52×10^-7
Γ_2	97,150,410	1,849,319,588	3,463,060	9,872	0.27	1.95×10^-7
Γ_3	170,830,352	2,937,087,979	5,089,789	10,840	0.23	1.01×10^-7

Table 2 Main characteristics of the LWCC of Google+: (start-date, duration) – Ω_1: (24-08-12, 17 days), Ω_2: (10-09-12, 11 days) and Ω_3: (20-06-13, N/A)

ID	# nodes	# edges	max(in)	max(out)	reciprocity	density
Ω_1	66,237,724	1,291,890,737	1,822,999	9,981	0.34	2.94×10^-7
Ω_2	84,789,166	1,633,199,823	2,579,551	9,872	0.30	2.27×10^-7
Ω_3	145,478,563	2,548,275,802	3,793,031	10,840	0.26	1.20×10^-7

2 Google+ Overview and Dataset

In this section, we briefly describe key features of the Google+ service and a summary of our dataset.

Platform Description: On June 2011 Google launched its own social networking service called Google+. The platform was announced as a new generation of social network. Previous works in the literature [8,9] claim that Google+ cannot be classified as particularly asymmetric (Twitter-like), but it is also not as symmetric (Facebook-like) because Google+ features have some similarity to both Facebook and Twitter. Therefore, they labelled Google+ as a hybrid online social network[8]. Similar to Twitter (and different from Facebook) the relationships in Google+ are unidirectional. In graph-theoretical terms, if user[2] x follows user y this relationship can be represented as a directed social edge (x, y); if user y also has a directed social edge (y,x), the relationship x, y is called symmetric[13]. Similar to Facebook, each user has a stream, where any activity performed by the user appears (like the Facebook wall). For more information about the features of Google+ the reader is referred to [14,15].

Dataset: We obtained our dataset from an earlier study on Google+ [9]. The dataset is a collection of three massive directed graph (denoted as Γ_i, for $i = 1, 2, 3$) of the social links of the users[3] in *Google+*, collected from August, 2012 to June, 2013. We use Breadth-First-Search (BFS) to extract the *largest weakly connected component* (LWCC) of Γ_i. We label the extracted LWCC as subgraph Ω_i. Since the users Ω_i form the most important component of the Google+ network [9], we extract the *reciprocal network* of Google+ from the Ω_i subgraph (see Sect. 3). The main characteristics of Γ_i and Ω_i are summarized in Table 1 and Table 2, where each snapshot represents a complete graph

[2] In this paper we use the terms "user" and "node" interchangeable

[3] Google+ assigns each user a 21-digit integer ID, where the highest order digit is always 1 (e.g., 100000000006155622736)

Table 3 Main characteristics of the reciprocal network of Google+: H

ID	# nodes	# edges	max(degree)	density
H_1	40,403,216	197,838,519	4,294	2.42×10^-7
H_2	49,161,409	226,373,003	4,425	1.87×10^-7
H_3	74,539,728	327,204,637	4,743	1.78×10^-7

of the social relations among all users in Google+ and density is defined as $|E|/[|V|(|V|-1)$ for a *directed* graph, and $2|E|/[|V|(|V|-1)$ for an *undirected* graph – here $|V|$ is the number of nodes and $|E|$ is the number of edge. We observe that reciprocity and density decrease for both Γ_i and Ω_i. This is due to the fact that new users joining Google+ tend to be less "social" and they make fewer connections as the network evolves – findings reported by the authors in [16].

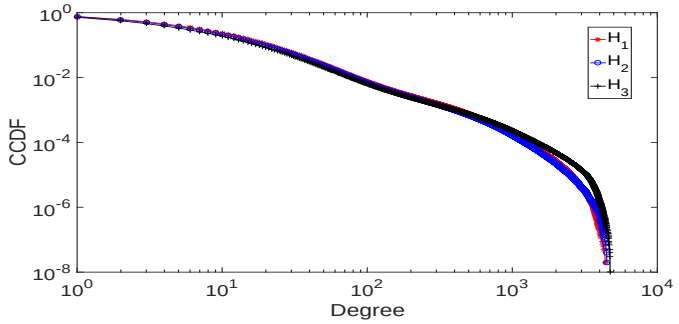
3 Overview of the Reciprocal Network

In this section, we first describe our methodology to extract the reciprocal network of Google+[4]. We then provide a brief overview of some global structural properties of the reciprocal network. Firstly, to derive the reciprocal network of Google+, we proceed as follows: from Ω, we extract the subgraph composed of nodes with at least one reciprocal edge. We label this new subgraph as G. However, G is not a connected subgraph. Hence, we use BFS (breadth-first-search) to extract its *largest connected component* (LCC); we label this new subgraph as H. In this paper, we consider this subgraph H as the "reciprocal network" of Google+[5]. The main statistics of subgraphs H_i are listed in Table 3.

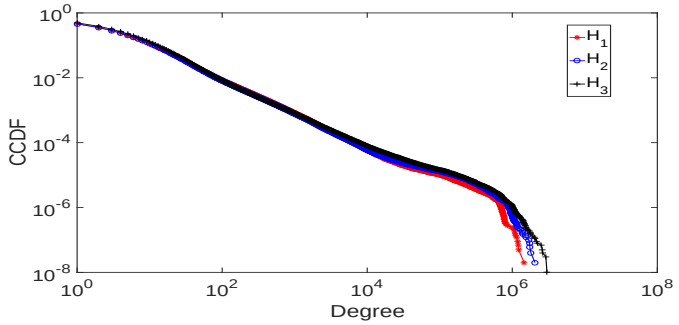
Figure 2 shows the complementary cumulative distribution function (CCDF) of the degrees of nodes in the subgraphs H_i – we note that they represent the *mutual* degrees or reciprocal degrees of the same nodes in Ω_i. For comparison, we also plot the CCDFs of the in-degrees and out-degrees for these nodes in Ω_i. We can see that these curves have approximately the shape of a power law distribution. The CCDF of a power law distribution is given by $Cx^{-\alpha}$ and $x, \alpha, C > 0$. By using the tool in [17,18], we estimate the exponent α that best models each of our distributions. We obtain $\alpha = 2.72$ for mutual degree, $\alpha = 2.41$ for out-degree and $\alpha = 2.03$ for in-degree distributions. We observe that the mutual degree and out-degree distributions have similar x-axis range and the out-degree curve drops sharply around 5000. We conjecture that this is because Google+ maintains a policy that allows only some special users to add more than 5000 friends to their circles [19]. The observed power-law trend

[4] For clarity of notation, we sometimes drop the subscript index i from the subgraphs notations, unless we are referring to a specific snapshot i

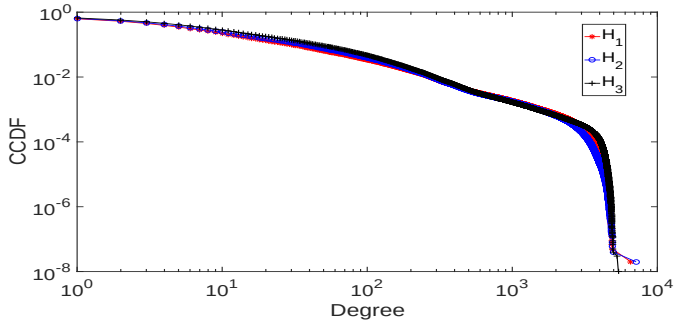
[5] It contains more than 90% of the nodes with at least one reciprocal edge in Google+. Hence, our analysis of the dataset is eventually approximate.



(a) Mutual degree distribuition



(b) In-degree distribution



(c) Out-degree distribution

Fig. 2 Log-log plot of a) mutual degree, b) in-degree and c) out-degree complementary cumulative distribution functions (CCDF) for several snapshots of the reciprocal network of Google+ (subgraphs H_i, i=1,2 and 3). All distributions show properties consistent with power-law networks.

in the distributions implies that a small fraction of users have a disproportionately large number of connections, while most users have a small number of connections – *this is characteristics of many social networks.*

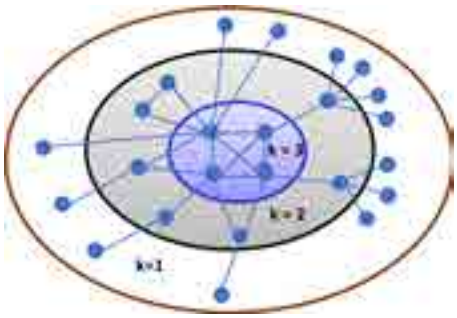


Fig. 3 A schematic representation of a network under k-shell decomposition: the network can be viewed as the union of shell 1 up to $k_{max} = 3$. The innermost core of the network is highlighted by the blue circle (the largest shell index: 3).

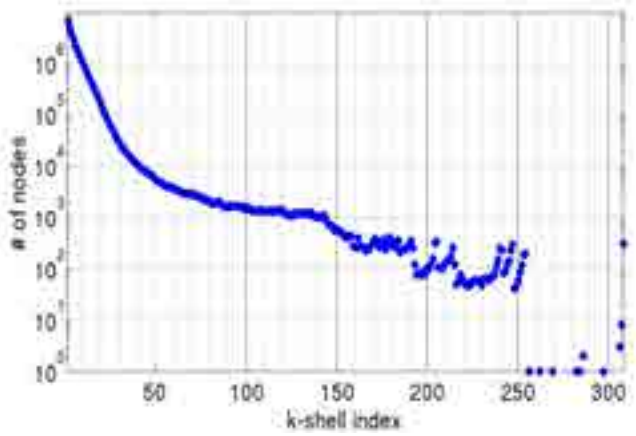


Fig. 4 The k-shell decomposition method on the reciprocal network of Google+ (subgraph H_1). For each k-shell, we plot the number of nodes belonging to the k-shell as k varies from 1 to $k_{max} = 308$.

4 Extracting the Nucleus of the Reciprocal Network using K-Shells

K-shell decomposition is a classical graph decomposition technique which has been used as an analysis and visualization tool to extract and study the "core" structure of complex networks, such as that of the Internet AS graph [11]. In this method, nodes are assigned a k-shell index according to their remaining degree, after pruning all nodes with degree smaller than the k value of the current shell. More specifically, this method works as follow: a) first, remove all nodes in the network with degree 1 (and their respective edges) – these nodes are assigned to the 1-shell; b) more generally, at step $k = 2, \ldots$, remove all nodes in the remaining network with degree k or less (and their respective edges) – these nodes are assigned to the k-shell; and c) the process stops when all nodes are removed at the last step – the highest shell index is labelled

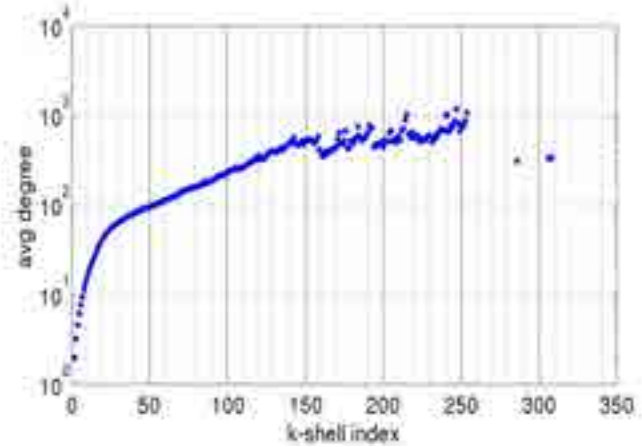
k_{max}. At the end of the k-shell decomposition process, each node v is assigned with a unique *k-shell index*, denoted by $shell(v)$. The network can be viewed as the union of all k_{max} shells – see Fig. 3 (note that this is distinct from *k-core decomposition*[6] defined in the literature [20,21], more in Sect. 10). The complexity of this procedure is $O(V + E)$ for a general graph.

For each k, we define the k-core (C_k) as the union of all shells with indices larger or equal to k or as the maximal induced subgraph $C_k \subseteq G$ such that if $v \in C_k$, then node v must have at least $k + 1$ neighbors that belong to C_{k-1} and $deg^k(v) > 0$ (we use $deg(v)$ to denote the degree of v in the network and $deg^k(v)$ to denote the degree of v in C_k). Similarly, k-shell (S_k) can be defined as the subgraph induced by the set of nodes with $d^{k-1}(v) \leq k$ and if $v \in S_k \rightarrow deg^k(v) = 0$.

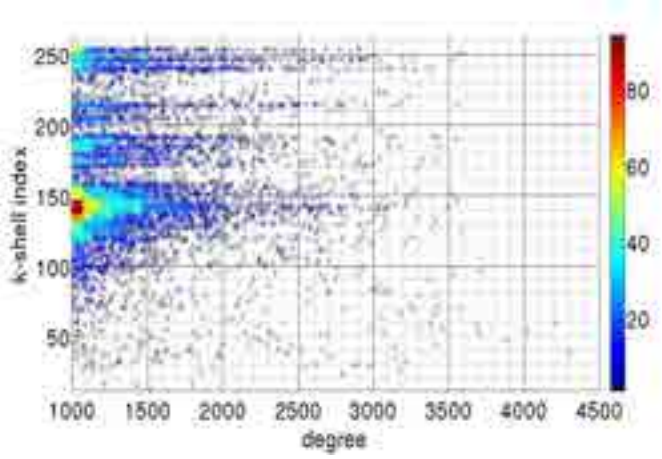
Clearly, for a node to belong to the k-core (thus $shell(v) \geq k$), it must have at least degree k, i.e., $deg(v) \geq k$. However, $deg(v) \geq k$ is not sufficient to guarantee it to belong to the k-core. For example, a node v with only neighbors of degree 1 (i.e., v is the root of a star structure) belongs to the 2-shell, i.e., $shell(v) = 2$, no matter how high its degree is. On the other hand, it is easy to see that if a node v is part of a clique of k nodes, then $shell(v) \geq k$. However, a node v does not need to be part of a k-clique to have $shell(v) \geq k$. Consider a tree T of n nodes (the sparsest graph with n nodes). We can in fact provide a complete characterization of nodes in T to have $shell(v) \geq k$ in a recursive manner: for v to have $shell(v) \geq k$, it must have at least k-neighbors u's with $shell(u) \geq k - 1$ – this characterization also applies to a general graph. We see that in the case of a tree, nodes with higher k-shell indices must lie more at the "core" (i.e., the increasingly "denser" part) of the tree. For a general graph, however, a node with a high k-shell index may not lie at the "core" of the graph: it can be part of a large clique that is "isolated" on a periphery of a massive graph. In such a case, the large clique will break off from the "core" of the network (e.g., as represented by the largest connected component remaining in the k-core) in the early stage of the k-shell decomposition process.

We apply the k-shell decomposition method to the Google+ reciprocal network for subgraph H_1 (we analyze the other subgraphs in Sect. 8). We find that the $k_{max} = 308$, and the k_{max}-core is a clique of size 290 nodes (the maximum clique in the Google+ reciprocal network). Figure 4 shows the number of nodes belonging to the k-shell as k varies from 1 to 308: we see that 99% of the nodes in our network fall in the lower k-shells (from $k = 1$ to 100). This is not surprising, as the majority of the nodes in our network have degree less than 100. Figure 5(a) shows the average degree of nodes in the k-shell, whereas in Fig. 5(b) we zoom in on nodes with $deg(v) \geq 1000$, and illustrate how they distribute across various k-shells. We see that while a large portion of high-degree nodes belong to higher k-shells, in fact the highest degree nodes belong to lower k-shells, suggesting that they do not lie at the "core" of the Google+ reciprocal network.

[6] Which simply removes all nodes with degree less than k in a graph



(a) Average degree of nodes in the k-shells



(b) K-shell distribuition of the nodes with $deg(v) \geq 1000$

Fig. 5 The k-shell decomposition method on the reciprocal network of Google+ (subgraph H_1). We plot the degree distributions for nodes in the k-shells, as k varies from 1 to $k_{max} = 308$: a) average degree of nodes in the k-shells, b) we zoom in on nodes with $deg(v) \geq 1000$, and illustrate how they distribute across various k-shells.

Figure 6 shows the size of the largest as well as those of the 2nd, 3rd and 4th largest connected components in the k-core, as k varies from 1 to 308. We note that at step $k = 121$, a small subgraph containing the maximum clique (of size 290) breaks off from the largest connected component which dissolves after $k = 253$, whereas this subgraph containing the maximum clique persists after $k = 252$ and becomes the largest component, and at $k_{max} = 308$, we are left with the maximum clique plus 10 additional nodes that are connected to the maximum clique. Closer inspection of nodes in the maximum clique reveals that its users belong to a single institution in Taiwan, forming

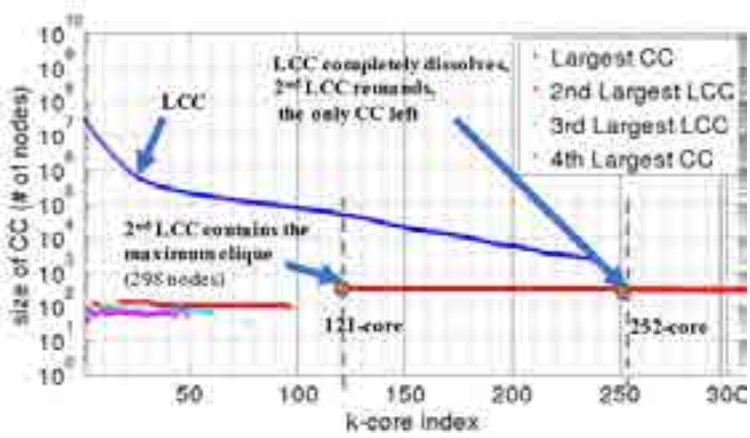


Fig. 6 The k-shell decomposition method on the reciprocal network of Google+ (subgraph H_1). For each k-core subgraph, we plot the size of the largest as well as those of the 2nd, 3rd and 4th largest connected components (LCC) in the k-core, as k varies from 1 to $k_{max} = 308$. At k-core=121, the 2nd LCC contains the maximum clique of the network and it becomes the 1st LCC in the network after k-core=252. This component persists up to k_{max}=308 (the network nucleus).

a close-knit community where each user follows everyone else. We see that directly applying the standard k-shell decomposition to the Google+ reciprocal network produces a clique of size 290, which we believe is unlikely to be the "core" of the Google+ reciprocal network.

From this result, we see that directly applying the standard k-shell decomposition to Google+'s reciprocal network produces an innermost structure that does not represent the core of this network. This is due to the fact that at a certain k-index, we reach the vicinity of the core; but going far beyond this index would destroy the core structure of the network.

5 The Depencence Value and Core Subgraph

In order to extract a meaningful core of the Google+ reciprocal network, we therefore modify the standard k-shell decomposition method to stop the process earlier. To achieve this, we propose a new metric that provides important information about the structural function of each node in the graph (we label it as "dependence" value) at each k-step:

The *dependence* value of node v at step k is defined as follows: for $v \in V$, $dep^0(v, \beta) = 0$ and for $k = 1, \ldots, c(v)$,

$$dep^k(v, \beta) := dep^{k-1}(v, \beta) + \delta^k(v) + \beta \times \Sigma_{u \in N^k(v)}[dep^{k-1}(u, \beta)] \qquad (1)$$

where β is a control parameter, $0 \leq \beta \leq 1$; $N^k(v)$ is the set of neighbors of node v that are removed at step k, and $\delta^k(v) = |N^k(v)|$. The dependency

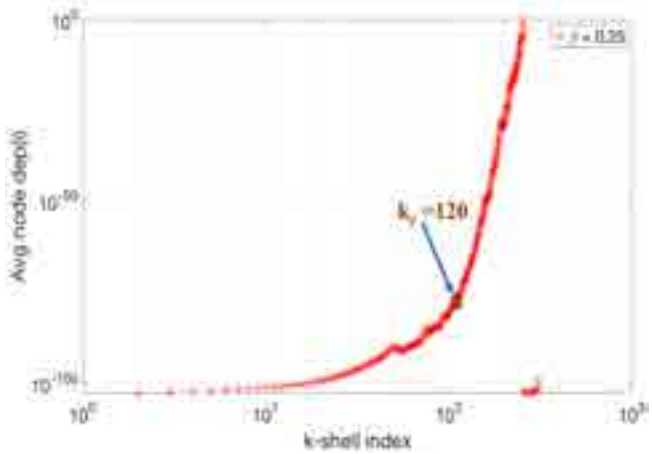


Fig. 7 Log-log plot of the average dependency values for the reciprocal network of Google+ (subgraph H_1). We plot the normalized average dependence value for the nodes in the k-shells, as k varies from 1 to $k_{max} = 308$.

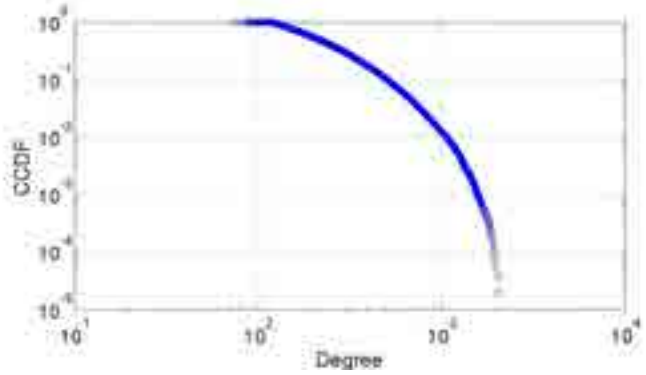


Fig. 8 Degree distribution for nodes in subgraph G_{120} extracted from H_1. Note that degree here refers to that of a node in G_{120}, the 120-core graph after the k_C-th shell decomposition process, it is not the (original) degree of the node in the Google+ reciprocal network.

of node v is recursively defined by measuring the number of nodes u (the h-hop neighbors of v, $h = 1, ..., k$) that are removed in earlier steps up to $k = c(v)$ – the *coreness* of node v (and for $k \geq c(v)$, by convention, we define $dep^k(v, \beta) = dep^{c(v)}(v, \beta)$).

Intuitively, $dep^k(v, \beta)$ captures the number of nodes recursively dependent on v that have been removed in earlier steps up to k. With $\beta = 0$, we note that $dep^k(v, \beta)$ captures the number of v's neighbors removed at each step up to k, and for $k \geq c(v)$, $dep^k(v, \beta) = deg(v) = \sum_k \delta(v)$, the degree of node v. With $\beta > 0$, $dep^k(v, \beta)$ captures not simply the dependence of its neighbors, but that of its neighbors' neighbors, and so forth. However, the number of nodes u removed at each step up to k does not influence the dependence value

of the node v uniformly. Their contribution is weighted by the parameter β in eq.(1). The parameter β quantifies the contribution of node u to the total dependence value of node v. More precisely, at the kth-step, we multiply the number of h-step removed neighbors of v by β^{h-1} (see the proof in the appendix). Thus, the further a node u is to node v, the less it will contribute to the total dependence value of node v. Hence, a node v having more nodes u with high dependence values in its vicinity will also have a high dependence value, creating the *dependency propagation* effect.

Given eq.(1), the dependence values of any $v \in V$ grows as k increases (more nodes are removed as we move from the periphery of the graph to its core). We posit that the network core should contain only nodes with very high dependence values. Hence, when we reach the vicinity of the network core, the nodes' dependence value will grow significantly as we increase k further, due to the dependency propagation effect. From this intuition, we develop the following empirical heuristic for terminating the k-shell decomposition process: *for any graph G with a dense core structure, we should stop the k-shell decomposition method at the k-index (k_C), where we observe a very sharp increase (largest "upward slope" or "gradient ascent") in the average dependence values of the nodes in the k-core graphs or k-shells of G, as k increases from 1 up to k_{max}.*

Our approach to calculate the dep(v, β) score for node v is dependent on the k-shell decomposition method and degree computation which have a complexity of $O(V+E)$. Then, given that the degree and coreness of each node are known, our procedure has a complexity of $O(E)$. Therefore, our methodology is highly scalable and can be applied to massive networks. Figure 7 shows the average dependency value per k-shell index for our massive Google+ reciprocal network (subgraph H_1). The parameter β is set to 0.25 (see the appendix for a discussion on the selection of this parameter). Applying the criteria described above, we therefore terminate the k-shell decomposition at $k_C = 120$, which yields the k_C-core graph with $k_C = 120$: this core graph G_{120} has 48,229 nodes and 6,378,596 edges, with an average degree of 132 and a density of approximately 0.00548, which is much greater than that of the reciprocal network H_1 as a whole. Figure 8 shows the degree distribution of the nodes in the 120-core graph (note that degree here refers to that of a node in G_{120}, the 120-core graph after the k_Cth shell decomposition process, it is *not* the (original) degree of the node in the Google+ reciprocal network). From Fig. 5(a) and Fig. 5(b), we see that G_{120} is comprised of many nodes with (original) high degrees in the Google+ reciprocal network, with an average (original) degree of roughly 500.

6 Constructing the Core Clique (Hyper)Graph

Given the dense core subgraph G_{120} (extracted in the previous section), how can we uncover its structure? To answer this question, we consider "maximal cliques" as the basic atomic (sub)structures of the network nucleus. Then, we

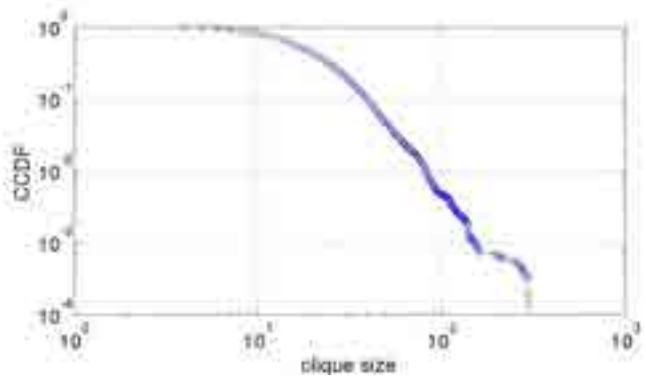


Fig. 9 Log-log plot of clique size complementary cumulative distribution function (CCDF) for the core subgraph G_{120} (extracted from H_1) – we extract these cliques using algorithms 1 and 2.

extract the minimal set of the largest maximal cliques that cover every node in G_{120}. Using these cliques substructures, we build a (hyper)graph as a higher-level representation of the nucleus[7] of a network. To achieve this, we proceed as following:

First, to find the largest maximal clique containing a given vertex in a network, we implement algorithm 1. It uses a variation of the popular Bron-Kerbosh algorithm [22] (we denote it as Simplified Bron-Kerbosh (SBK)) to extract maximal cliques. During the search for the largest maximal clique containing a given vertex v (thereafter referred to as C^v in short), our heuristic removes the vertices that cannot form cliques larger than the clique stored in the variable C_{max}. Furthermore, our algorithm considers only the set of neighbors of v that share at least one edge to another vertex adjacent to v at each step, instead of recursively considering all neighbors of v, and thus is much faster. This set (denoted by $N^i(v)$) is sorted in decreasing order based on the number of shared neighbors between v and $u \in N^i(v)$ for the following reason: in a relatively fairly connected subgraph, a vertex with the largest number of shared nodes with v is more likely to be a member of C^v compared to any other. Then, in the worst case, algorithm 1 loops over the complete set $N^i(v)$ at most $\triangle$ (max degree in the graph), calling the subroutine SBK at most $\triangle$. Thus, the time complexity of our heuristic is bounded by $O(\triangle^2)$. Using algorithm 1, we develop a procedure to extract the minimal set of the largest maximal cliques that cover every node in a given graph (algorithm 2). The resulting set of cliques returned from this method is always guaranteed to contain at least a unique node per clique. We apply this procedure to subgraph G_{120} and obtain 34,501 maximal cliques with an average clique size of 23.03 nodes. Figure 9 shows the clique size distribution.

Second, using the extracted 34,501 maximal cliques, we generate a new *directed* (hyper)graph, where the vertices are (unique) cliques of various sizes,

[7] In this paper we use the terms "core" and "nucleus" interchangeable

Algorithm 1 Largest Maximal Clique Extraction algorithm (LC)

Input: node u
Output: largest maximal clique containing u
R : currently growing maximal clique
$P := N[u]$: set of neighbors of vertex u
procedure LC(u)
    $N^i(u) = \{w_i, w_i, ... | w_{k=i,j..} \in N(u) \bigwedge d^u(w_i) > d^u(w_j)\}$
    $C_{max} = 0$
    $max = 0$
    for $w \in N^i(u)$ **do**
        $R = [u]$
        $P = N[w]$
        $C = SBK(R, P, max)$
        $k = size(C)$
        if $k > max$ **then**
            $C_{max} = C$
            $max = k$
    return C_{max}

Subroutine: Simplified Bron-Kerbosh (SBK)

procedure SBK(R, P, max)
    if $size(R) + size(P) \leq max$ **then**
        return ▷ it is not possible to find a clique larger than max
    else if $P := 0$ **then**
        report R as a maximal clique
    else
        Let u_{new} be the vertex with highest number of neighbors in P
        $R_{new} := R \cup \{u_{new}\}$
        $P_{new} := P \cap N[u_{new}]$
        SBK(R_{new}, P_{new}, max)

Algorithm 2 Extract Minimal Set of Maximal Cliques from a Graph

procedure EMC($G(V, E)$)
    construct a set W and $W := V$
    construct a ordered list S of the nodes in V based on their degree (decreasing order)
    select the first item in S, vertex i, as the pivot
    apply the LC algorithm using i as the pivot vertex
    add the reported maximal clique c_i containing i to the clique set $C_{total} = [c_n, c_m, ..]$
    remove the nodes in c_i from $W : W_j = W_i - c_i$
    select the next item in S, vertex j, as the next pivot vertex such that $j \notin C_{total}$ and repeat steps(5), (6) and (7) until $W = \emptyset$

and there exists a *directed* edge from clique C_i to clique C_j if more than half of the nodes in C_i are contained in C_j, i.e., $C_i \to C_j$ if $(|C_i| \cap |C_j|)/|C_i| \geq \theta = 0.5$. We vary the parameter θ from 0.5 to 0.7, and find that it does not fundamentally alter the connectivity structure of the (hyper)graph of cliques thus generated. We remark that the maximal clique containing each node v can be viewed as the most stable structure that node v is part of. The directed (hyper)graph of cliques captures the relations among these stable structures each node is part of: intuitively, each directed edge in a sense reflects the attraction (or gravitational pull) that one clique (a constellation of nodes) has over the other. Hence, this (hyper)graph of cliques provides us with a higher-

level representation of the dense core graph of the Google+ reciprocal network – how the most stable structures are related to each other. This procedure can be viewed as a form of clique percolation [12].

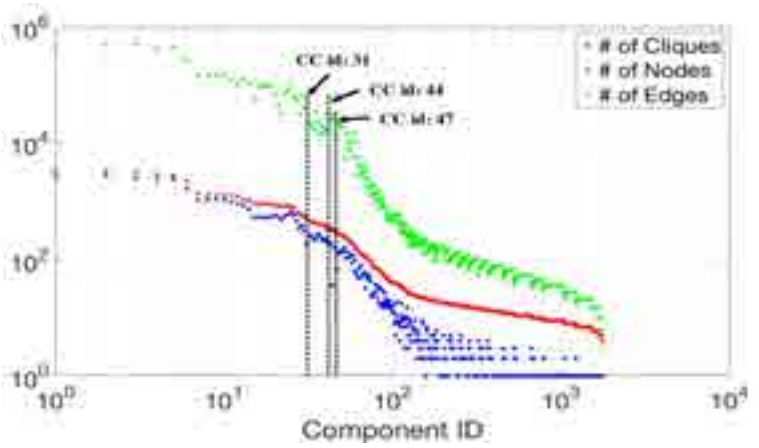
We find that this (hyper)graph of cliques comprises of 1,758 connected components (CCs). The largest component has 2,618 cliques, 3,295 nodes and 437,867 edges, while the smallest has 1 clique, 3 nodes and 3 edges respectively. We regard these connected components (CCs) as forming the *core communities* of the core graph of the Google+ reciprocal graph: each CC is composed of either one single clique (such a CC shares few than half of its members with other cliques or CCs), or two or more cliques (stable structures) (where one clique shares at least half of its member with another clique in the same CC, thus forming a closely knit community). Figure 10(a) shows the distributions of these components in terms of the number of cliques, the number of nodes and the number of edges. We observe that for CC id's from 1 to 100 (which contains 30 or more cliques), there is a strong correlation between the number of cliques, nodes and edges: in general the connected components with the highest number of cliques also have the highest number of nodes and edges.

Figure 10(b) shows the maximum, minimum, average and 75% percentile of clique size for each CC. We observe that there is not a relationship between the number of cliques and their respective sizes in the CCs. We observe that most cliques have sizes between 10 and 100 nodes. There are largest CCs composed with a huge number of cliques of small size (e.g., CC ids from 1 to 10), whereas there are also small CCs composed with few number of cliques but with very large sizes (e.g. CC ids: 31, 44, and 47). We note also that there are a number of CCs which contain only one clique, but some of these cliques are of large size also.

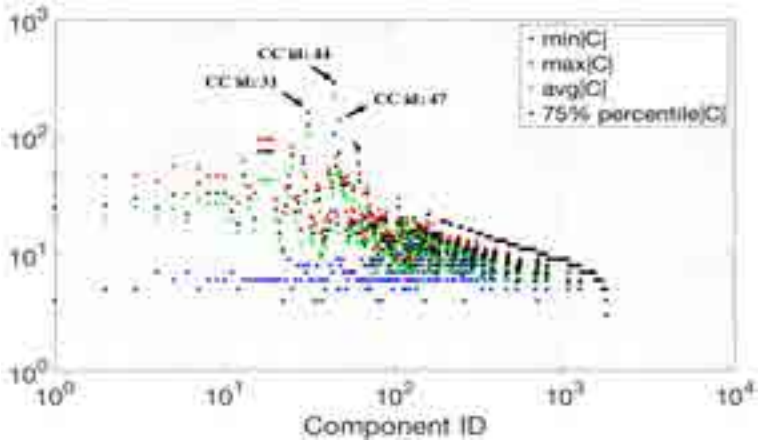
7 Analysis of the Core Community (Hyper)Graph & its Structure

We now investigate the relationship between the connected components (CCs) in our clique (hyper)graphs constructed in the previous section (Sect. 6), in particular the 70th largest CCs. Recall that we regard the CCs in the clique (hyper)graphs as forming the core communities within Google+ reciprocal network nucleus – each CC represents a dense cluster of cliques. In this section, we define three metrics to study the relations among these CCs in the underlying Google+ reciprocal network:

- **Shared Nodes**: the number of nodes that CC_i and CC_j have in common:
$$S(CC_i, CC_j) = |\{u \in V | u \in CC_i, u \in CC_j\}| \quad (2)$$
- **Shared Neighbors**: the number of nodes in CC_i that have an edge to another node in CC_j:
$$N(CC_i, CC_j) = |\{u \in CC_i, |\exists v \in CC_j : (u, v) \in E\}| \quad (3)$$
- **Cross-Edges**: the number of cross edges between two connected components (CC_i and CC_j):
$$B(CC_i, CC_j) = |\{(u, v) \in E | v \in CC_i, u \in CC_j\}| \quad (4)$$



(a) Number of cliques, nodes and edges



(b) Clique size: maximum, minimum, average and 75% percentile

Fig. 10 Statistics of the connected components in the (hyper)graph of cliques constructed from the core subgraph G_{120} (extracted from H_1): a) distribution of the number of cliques, nodes and edges and b) distribution of the clique size in terms of the maximum, minimum, average and 75% percentile of the clique size.

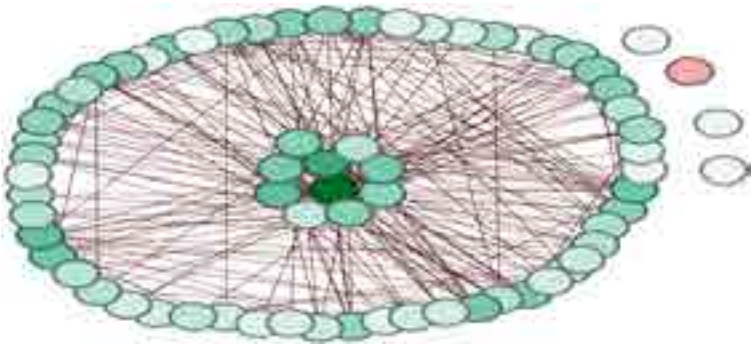
These metrics produce a set of three new (hyper)graphs that succinctly summarize the (high-level) structural relations among the core community structures: 1st) a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node; 2nd) a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ denotes that CC_i has the largest number of cross edges to nodes in CC_j; 3rd) a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes to nodes in CC_j. These (hyper)graphs provide a "big picture" view of the core graph of the Google+ reciprocal network and yield insights as to how it is formed.

Table 4 Summary of the statistics for the ten components that lie at the center in the core graph of the reciprocal network of Google+. Together they form the core to which peripheral sparse subgraphs are attached.

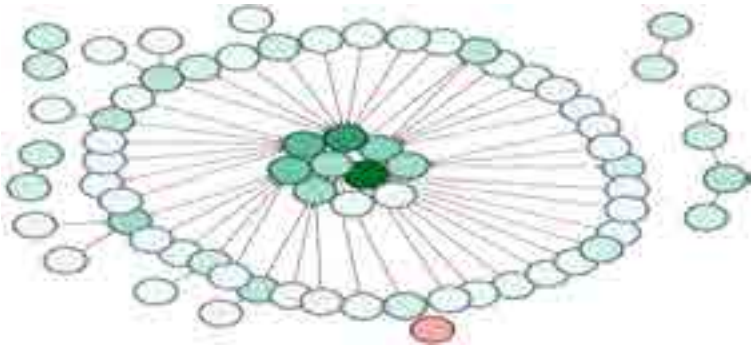
ID	# c	# nodes	# edges	avg $\|c\|$	max $\|c\|$	min $\|c\|$	75% percentile
1	2,618	3,295	437,867	30.0	47	4	25
2	2,745	3,256	494,867	20.2	46	5	26
3	2,437	3,059	499,356	25.5	47	5	30
4	2,324	2,877	416,098	20.2	42	7	25
5	2,340	2,737	449,225	24.3	56	6	32
7	1,040	1,362	146,151	29.2	55	5	40
15	513	923	60,191	16.0	33	6	20
22	473	808	32,031	10.0	23	4	11
37	262	396	14,324	9.2	15	4	10
47	69	297	22,629	50.3	139	5	73

Figures 11(a), 11(b), 11(c) show the (hyper)graphs of the relationship between the components based on the number of shared nodes, cross-edges and shared neighbors. These figures show that there are ten subgraphs (core communities") comprising of dense clusters of cliques that lie at the center of the nucleus of the Google+ reciprocal network, through which other communities of cliques are richly connected. Then, the 1,758 connected components (CCs) in the clique (hyper)graph form the core graph of the Google+ reciprocal network, to which other nodes and edges that are part of sparse subgraphs on the peripherals of the network are attached. Table 4 shows a summary of the statistics for the ten CCs, respectively. We observe that the largest CC has 2,618 cliques, 3,295 nodes and 437,867 edges, while the smallest has 69 cliques, 297 nodes and 22,629 edges. The set of components in table 4 contains some of the largest CC in our clique (hyper)graph.

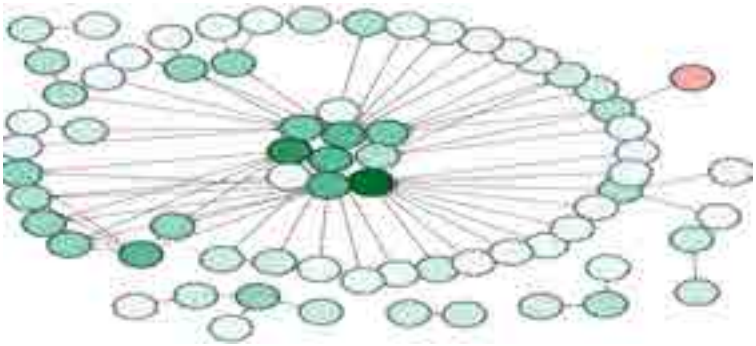
From figures 11(a), 11(b) and 11(c), we observe that in the periphery of our core communities (hyper)graphs, we find a small CC composed with 36 of the largest cliques in the Google+ reciprocal network. The average, minimum and maximum sizes of the cliques in this CC are 227, 105 and 290 – the latter is the maximum clique of the Google+ reciprocal network. This CC is highlighted by a "red circle" in the (hyper)graphs in Fig. 11. It shows this CC lies more at the outer ring of Google+'s dense core structure. As mentioned earlier in Sect. 4, the 290 users in this maximum clique of the Google+ reciprocal network belong to a single institution in Taiwan where every user follows every other. The users in this clique also form close relations with many other users, forming 35 other cliques. Together, these 35 cliques form a close-knit community. However, we see that this community in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network. Hence, we see that simply applying the conventional k-shell decomposition method to the Google+ reciprocal network would yield the maximum clique in the Google+ reciprocal network, but not its *core* structure. In contrast, the ten CCs mentioned above more likely lie at the "center" of the core graph of the Google+ reciprocal network.



(a) (hyper)graph of the structural relation among the core communities (CCs) based on the number of shared nodes: a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node.



(b) (hyper)graph of the structural relation among the core communities (CCs) based on the number of cross-edges: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of cross edges to nodes in CC_j.



(c) (hyper)graph of the structural relation among the core communities (CCs) based on the number of neighboring nodes: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes with CC_j.

Fig. 11 (Hyper)Graphs for the core communities (extracted from G_{120}) of the reciprocal network of Google+: snapshot - H_1. The color intensity of a CC is proportional to its degree. The CC highlighted in "red" is the core subgraph yielded by directly applying the standard k-shell decomposition to Google+'s reciprocal network. However, our core communities (hyper)graphs show that this structure in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network.

Table 5 Main characteristics of the core subgraph (G_C) for the reciprocal network of Google+ across several snapshots.

H_i	k_C	# nodes	# edges	avg(d)	density
1	120	48,229	6,378,596	132	0.00548
2	120	52,904	6,737,630	127	0.00482
3	130	94,112	14,260,691	152	0.00322

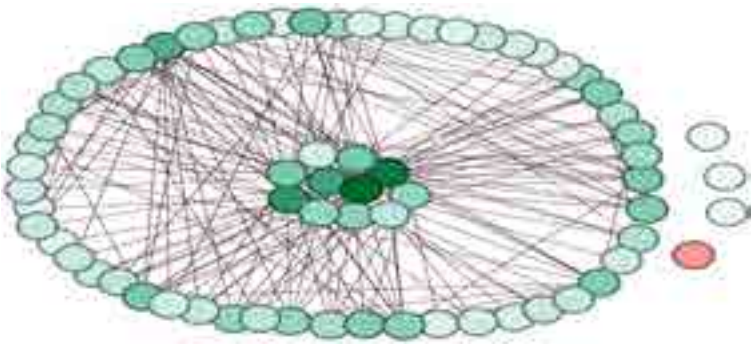
Table 6 Main statistics of the core communities (hyper)graphs for H_i: c - cliques; CC - connected components

H_i	# c	avg$\|c\|$	# CC	max$\|CC\|$	min$\|CC\|$
1	34,501	23.03	1,758	2,618	1
2	38,055	20.68	2,221	2,487	1
3	65,101	24.96	3,802	6,217	1

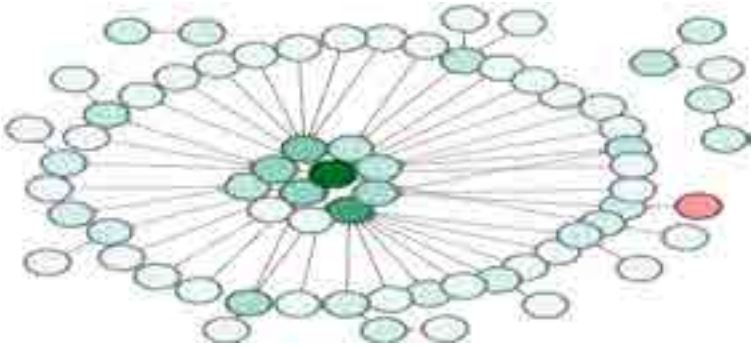
8 Evolution of the Core Community (Hyper)Graph

We now analyze how the core structure of the Google+ reciprocal network evolves over time using the remaining snapshots of subgraph H ($H_{i=2,3}$). To achieve this, we apply our methodology to uncover the core communities (hyper)graph for H_i. Table 5 shows the k_C-indices where we stop the k-shell decomposition method and provides statistics for the core subgraph (G_C) of the reciprocal network of Google+ across three different snapshots. We observe that the size of the nucleus increases as the network evolves, as well as, its density – although, we see a slight decrease at H_2 (this correlates with the release of a new Google+ feature reported by the authors in [16]). Table 6 provides statistics for the core communities (hyper)graphs. We observe that the number of cliques in the core subgraph (G_C) increases as the network evolves. Similarly, the number of core communities (CC) and the size of the largest CC in the clique (hyper)graph increase as the network evolves. In contrast, the size of the smallest CC remains the same across all the snapshots.

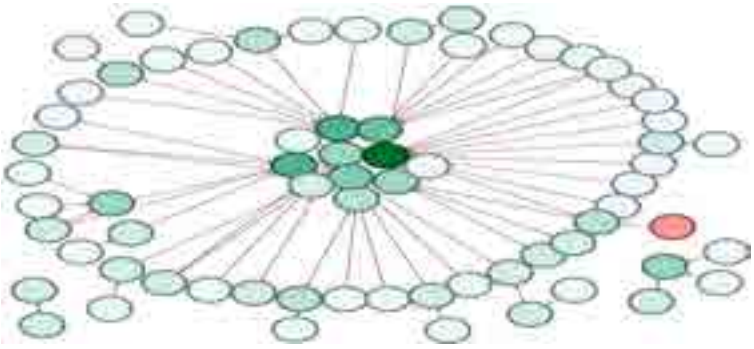
Analyzing the nodes that are found in the nucleus, we find that the set that participates is very stable over time. We find changes consisting of a few percentage of nodes that moved from the nucleus to a lower k-shell as the network evolves: 9% from $H_1 \rightarrow H_2$ and 5% from $H_2 \rightarrow H_3$. We also observe that the main structure of the core communities (hyper)graph is stable across all the snapshots: it consists of dense clusters of cliques that lie at the center of the core graph, through which other communities of cliques are richly connected. Additionally, we observe that the number of the most central communities in the core communities (hyper)graphs is also very stable: it increases from 10 to 11 across snapshots $H_1 \rightarrow H_2$ and from 11 to 13 across snapshots $H_2 \rightarrow H_3$. Lastly, we see that the community containing the "maximum clique" remains in the periphery of the core subgraph as the network evolves – see Fig. 12 and Fig. 13 for illustrations.



(a) (hyper)graph of the structural relation among the core communities (CCs) based on the number of shared nodes: a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node.

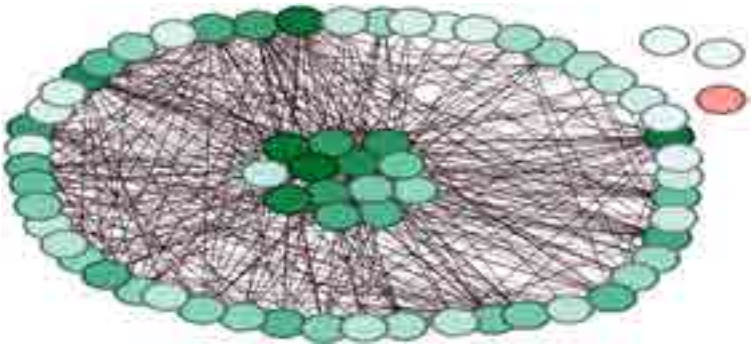


(b) (hyper)graph of the structural relation among the core communities (CCs) based on the number of cross-edges: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of cross edges to nodes in CC_j.

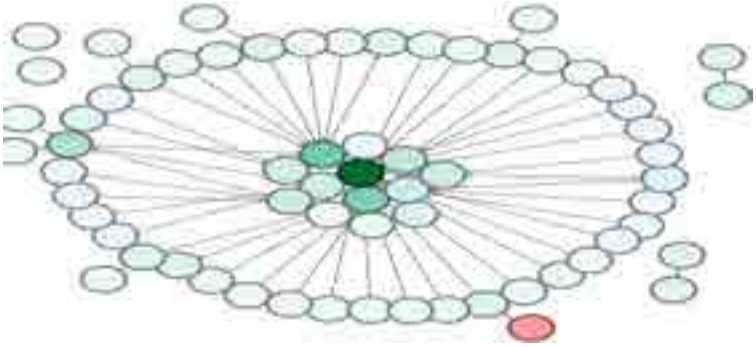


(c) (hyper)graph of the structural relation among the core communities (CCs) based on the number of neighboring nodes: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes with CC_j.

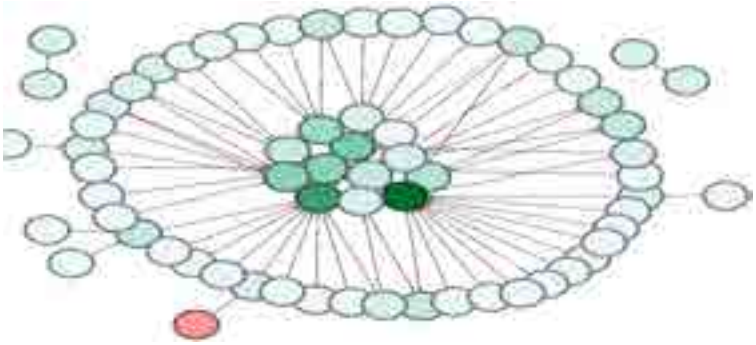
Fig. 12 (Hyper)Graphs for the core communities (extracted from G_{120}) of the reciprocal network of Google+: snapshot - H_2. The color intensity of a CC is proportional to its degree. The CC highlighted in "red" is the core subgraph yielded by directly applying the standard k-shell decomposition to Google+'s reciprocal network. However, our core communities (hyper)graphs show that this structure in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network.



(a) (hyper)graph of the structural relation among the core communities (CCs) based on the number of shared nodes: a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node.



(b) (hyper)graph of the structural relation among the core communities (CCs) based on the number of cross-edges: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of cross edges to nodes in CC_j.



(c) (hyper)graph of the structural relation among the core communities (CCs) based on the number of neighboring nodes: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes with CC_j.

Fig. 13 (Hyper)Graphs for the core communities (extracted from G_{120}) of the reciprocal network of Google+: snapshot - H_3. The color intensity of a CC is proportional to its degree. The CC highlighted in "red" is the core subgraph yielded by directly applying the standard k-shell decomposition to Google+'s reciprocal network. However, our core communities (hyper)graphs show that this structure in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network.

9 Implications

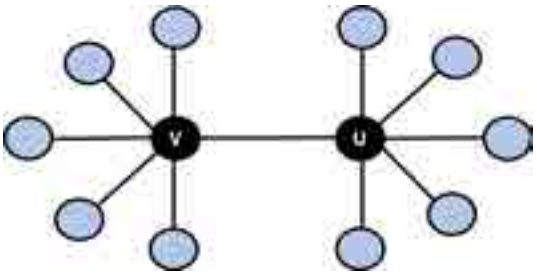
So far, we have demonstrated that our method can effectively uncover and extract the nucleus of the Google+ reciprocal network based on large-scale dataset. In this section, we discuss the implications of our method and results. While our findings are likely applicable to many different applications, we concentrate on their effect on the identification of influential spreaders, network formation, design and robustness:

Influential Spreaders: The "coreness" centrality or k-shell index has been argued to be a better measure than node degree for identifying influential spreaders in a network [28,29]. However, our results show that using k-shell indices as a predictor of spreading influence of a node can be misleading. This is due to the fact that for a node to have a high k-shell index, it just needs to be a part of a very strong structure (e.g., a clique). This structure, however, may be isolated and lie at the edge or periphery of the network, instead of its core (see Sect. 4). Our analysis shows that the dependency value of a node, $dep^k(i)$, provides important information about the structure function of each node in the graph. Thus, we believe that by using a node dependency value along with its k-shell index (dep^k, k), we can better predict the spreading influence of a node than simply using its k-shell index. We will investigate this in the future.

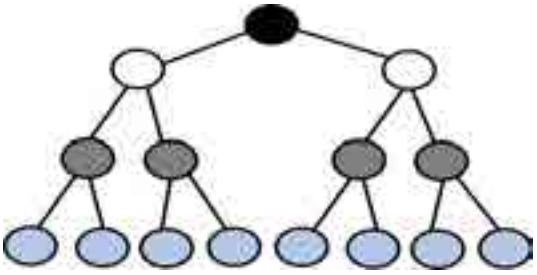
Network Formation: A network core gives a well-defined starting point and a way to explore the network topology systematically. For example, a network can be reconstructed layer by layer from the core to its periphery. Then, topological features of the nodes and structural properties of the network can be measured at each layer. Furthermore, using the core, we can build macroscopic models of the network that can help us predict the topological growth of the network and provide good upper bounds of the distance between the nodes – see the jellyfish model of the Internet in [24]. Therefore, unveiling the core structure of networks can help us uncover and understand possible organizing principles shaping the observed network topological structure and network formation.

Network Design: Observing the evolution patterns of the core structure of social networks can give insights for the design of future social networks by other social networking service providers who would like to enter the market. Furthermore, it can also help applications for social networks to be designed to take advantage of the network core properties.

Network Robustness: Robustness is often defined as the ability of a network to continue to function when it is subject to failures. Uncovering the core structure of networks is fundamental in the development of techniques for analyzing the vulnerability or robustness of networks. For example, in Google+ the tight core coupled with high link reciprocity implies that users in the core appear on large number of the shortest paths in the network. Thus, if malicious users are able to penetrate the core, they can destroy or remove the hubs of information flow (core nodes) in the network. Hence, disrupting the functionally of the network. Then, by strengthening the defenses in the core subgraph, we can increase the robustness of the social networks.



(a) double star graph



(b) binary tree

Fig. 14 Example networks of double star and binary tree graphs. These structures cannot be decomposed using k-core decomposition. However, they are decomposed into k-shells by k-shell decomposition: a) 1-shell: blue nodes and 2-shell: black nodes); b) 1-shell: blue nodes; 2-shell: grey and black nodes and 3-shell: white nodes.

10 Related Work

One of the most popular quantitative methods to investigate core-periphery structure was proposed by Borgatti and Everett in 1999 [30]. Based on this study, several methods for identifying the core-periphery of a network have been proposed [26,31,32]. These algorithms attempt to determine which nodes are part of a densely-connected core and which are part of a sparsely connected periphery by solving some complex optimization problem. In contrast, some studies simply define the network "core" as the maximal clique composed of the highest degree nodes in a network [24], while other studies focus instead on some notion of connectivity information (e.g. betweenness, closeness, etc.) to find the core and periphery of a network [26,27,31,33,34]. Consequently, most of these methods are computationally expensive and do not scalable to large networks.

The authors in [35] used the notion of α-β community to extract the "core" of a graph. An α-β community is a connected subgraph C with each vertex in C connected to at least β vertices of C and each vertex outside of C connected to at most α vertices of C ($\alpha < \beta$). They extract the network core structure by taking the intersection of α-β communities of different size k. A core thus corresponds to one or multiple dense regions of the graph. As a result, the proposed heuristics in [35] may return multiple dense regions ("cores") for a

given network. In addition, this algorithm does not guarantee to terminate within a reasonable amount of running time.

Closely related to our work, the authors in [20] propose the k-core decomposition to discover interesting structural properties of networks. A k-core of G is a subgraph G^* obtained by recursively removing all the vertices of degree less than k, until all the vertices in the remaining graph have degree at least k. This method is very scalable and it has a time complexity similar to the k-shell decomposition for general graphs: $(O(V+E))$. However, k-core decomposition is not equivalent to k-shell decomposition, where at each step k, we prune vertices of degree k or less. Different from k-shell decomposition, the k-core decomposition is unable to uncover the structural properties for certain type of graphs or substructures. For example, a double star-like graph S formed by two connected vertices v and u with high degrees that connect many vertices with degree one cannot be decomposed beyond 1-shell (or 1-core), containing all the vertices in graph S, no matter how high are the degree of the vertices v and u. Similarly, a binary tree graph T cannot be decomposed beyond the first shell, independently of the depth of the tree T – see Fig. 14 for an illustration.

11 Conclusion

In this paper, we have developed an effective three-step procedure to *hierarchically* extract and unfold the *core* structure of the reciprocal network of Google+. We first applied a modified version of the k-shell decomposition method to prune nodes and edges of sparse subgraphs that are likely to lie at the peripherals of the Google+ reciprocal network. We then performed a form of clique percolation to generate a new *directed* (hyper)graphs where vertices are maximal cliques containing the nodes in the dense "core" graph generated in the previous step, and there exists a directed edge from clique C_i to clique C_j if half of the nodes in C_i are contained in C_j. We found that this (hyper)graph of cliques comprises of 1700+ connected components (CCs), which represent the core "communities" of the Google+ reciprocal network. Finally, we introduced three metrics to study the relations among these CCs in the underlying Google+ reciprocal network: the number of nodes shared by two CCs, the number of nodes that are neighbors in the two CCs, and the number of edges connecting these neighboring nodes. These metrics produce a set of new (hyper)graphs that succinctly summarize the (high-level) structural relations among the core "community" structures and provide a "big picture" view of the core structure of the Google+ reciprocal network and how it is formed. In particular, we found that there are ten CCs that lie at the center of this core structure through which the other CCs are most richly connected.

Our proposed three-step hierarchical procedure assumes that the core subgraph of a network has a large number of cliques. Hence, it may fail to yield a meaningful structure for graphs with just a small number of cliques. To address this limitation, we can relax the notion of clique by constructing substructures which are *clique-like*. For example, a *k-relaxed clique* [23] is a set of nodes that

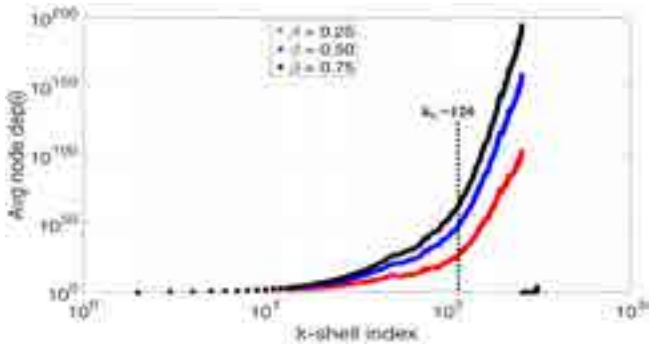


Fig. 15 Avg dependency values for Google+ (H_1) ($\beta = 0.25, 0.50, 0.75$)

connect to every node in the set except for at most k nodes (a 1-relaxed clique is a clique) [24]; and *k-clique* is a maximal subgraph such that the distance between each pair of its vertices is not larger than k. As part of ongoing and future work, we will develop a more rigorous characterization of the core graph of the Google+ reciprocal network based on the (modified) k-shell decomposition, and provide a more in-depth analysis of the (hyper)graph structures of the clique core graph and the (high-level) structural relations among the core "community" structures. We also plan to apply our method to a massive Twitter dataset (with more than 500 million nodes and $\approx$ 23 billion edges) and other OSNs.

Acknowledgments

This research was supported in part by DoD ARO MURI Award W911NF-12-1-0385, DTRA grant HDTRA1- 14-1-0040, NSF grant CNS-1411636, CNS-1618339 and CNS-1617729.

12 Appendix

Beta Parameter Selection: we now proof that the number of n-step removed neighbors of i is multiplied by β^{n-1}. We also present a discussion on how the selection of values for the β parameter in (1) impacts our criteria to stop the k-shell decomposition method presented in Sect. 5:

Given that $dep^0(i) = 0$ and $dep^1(i) = \delta^1(i)$, we can write an expression for $dep^2(i)$ as following:

$$\begin{aligned} dep^2(i) &= dep^1(i) + \delta^2(i) + \beta \times \Sigma_{j \in N^2(i)} dep^1(j) \\ &= \delta^1(i) + \delta^2(i) + \beta \times \Sigma_{j \in N^2(i)} \delta^1(j) \end{aligned} \tag{5}$$

Let's assume that node i has $c(i) = 4$, then $dep^4(i)$ is computed as following:

$$dep^4(i) = dep^3(i) + \delta^4(i) + \beta\Sigma_{j\in N^4(i)}[dep^3(j)] \tag{6}$$

Expanding (6) gives:

$$dep^4(i) = dep^3(i) + \delta^4(i) + \beta\Sigma_{j\in N^4(i)}[dep^2(j) + \delta^3(j) + \beta\Sigma_{j'\in N^3(j)}dep^2(j')]$$

Substituting (5) gives:

$$dep^4(i) := dep^3(i) + \delta^4(i) + \beta\Sigma_j[M^3(j) + \beta\delta^2(j)\rho^1(j'^*) + \beta\Sigma_{j'}[M^2(j') + \beta\delta^2(j')\rho^1(j'')]]$$

where $M^k(i) = \Sigma_k\delta^k(i)$ and $\delta^k(i) = \rho^k(i), \forall i \in V$.

Further simplify $dep^4(i)$ gives:

$$dep^4(i) := dep^3(i) + \delta^4(i) + \Sigma_j[\beta M^3(j) + \beta^2\delta^2(j)\rho^1(j'^*) + \Sigma_{j'}[\beta^2 M^2(j') + \beta^3\delta^2(j')\rho^1(j'')]]$$

We can rewrite the above expressions as:

$$dep^4(i) := dep^3(i) + \beta^0 A + \Sigma_j[\beta B + \beta^2 C + \Sigma_{j'}[\beta^2 D + \beta^3 E]] \tag{7}$$

Where:

- $A = \delta^4(i)$: 1-step neighbors of i removed at $k = 4$
- $B = M^3(j)$: 2-step neighbors of i removed at $k = 1, 2, 3$
- $C = \delta^2(j)\rho^1(j'^*)$: 3-step neighbors of i removed at $k = 1$
- $D = M^2(j')$: 3-step neighbors of i removed at $k = 1, 2$
- $E = \delta^2(j')\rho^1(j'')$: 4-step neighbors of i removed at $k = 1$

By generalizing equation (7) ($k = 5, ..., n$), we observe that at every k-index, the number of n-step removed neighbors of i is multiplied by β^{n-1}. This concludes our proof.

Essentially, the parameter β quantifies the contribution of node j to the total dependence value of node i. Thus, varying β in the range $]0, 1[$ will not have any impact on the value of the k-index where we should stop the k-shell decomposition method — by varying β, we are impacting the contribution of any node j to the total dependence value of node i by the same proportion. Thus varying the β^{n-1} does not have any impact in our criteria to stop the k-shell decomposition method introduced in Sect. 5 – see Fig. 15 for an illustration.

References

1. Gong, N.Z., Xu, W.: Reciprocal versus parasocial relationships in online social networks. Soc. Netw. Anal. Min. 4(1), 184–197 (2014)
2. Wolfe, A.: Social network analysis: methods and applications. Am. Ethnologist 24(1), 219–220 (1997)
3. Jamali, M., Haffari, G., Ester, M.: Modeling the temporal dynamics of social rating networks using bidirectional effects of social relations and rating patterns. In: WWW 2011, pp. 527–536. ACM (2011)
4. Li, Y., Zhang, Z.-L., Bao, J.: Mutual or unrequited love: identifying stable clusters in social networks with uni- and bi-directional links. In: Bonato, A., Janssen, J. (eds.) WAW 2012. LNCS, vol. 7323, pp. 113–125. Springer, Heidelberg (2012)
5. Garlaschelli, D., Loffredo, M.I.: Patterns of link reciprocity in directed networks. Phys. Rev. Lett. 93, 268–701 (2004)
6. Jiang, B., Zhang, Z.-L., Towsley, D.: Reciprocity in social networks with capacity constraints. In: KDD 2015, pp. 457–466. ACM (2015)
7. Hai, P. H., Shin, H.: Effective clustering of dense and concentrated online communities. In: Asia-Pacific Web Conference (APWEB) 2010, pp. 133–139. IEEE (2010)
8. Gong, N.Z., Xu, W., Huang, L., Mittal, P., Stefanov, E., Sekar, Song, D.: Evolution of the social-attribute networks: measurements, modeling, and implications using Google+. In: IMC 2015, pp. 131–144. ACM (2015)
9. Gonzalez, R., Cuevas, R., Motamedi, R., Rejaie, R., Cuevas, A.: Google+ or Google-? dissecting the evolution of the new OSN in its first year. In: WWW 2013, pp. 483–494. ACM (2013)
10. Dumba, B., Zhang Z.,: Unfolding the Core Structure of the Reciprocal Graph of a Massive Online Social Network. In: Proceedings of the 10th Annual International Conference on Combinatorial Optimization and Applications (COCOA'16), pp. 16–18 (2016)
11. Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y. and Shir, E.: A model of Internet topology using k-shell decomposition. PNAS 104, 11150–11154 (2007).
12. Palla, G., Dernyi, I., Farkas, I. and Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435(7043), 814–818 (2005).
13. Schiberg, D., Schneider, F., Schiberg, H., Schmid, S., Uhlig, S., Feldmann, A.: Tracing the birth of an OSN: social graph and profile analysis in Google+. In: WebSci 2012, pp. 265–274. ACM (2012)
14. Google+ Platform, http://www.google.com/intl/en/+/learnmore/
15. Google+, http://en.wikipedia.org/wiki/Google+
16. Dumba, B., Golnari, G., Zhang, Z.: Analysis of a Reciprocal Network Using Google+: Structural Properties and Evolution. In: Proceedings of the 5th International Conference on Computational Social Networks (CSoNet'16), pp. 14–26 (2016)
17. Clauset, A., Shalizi, C. R., and Newman, M. E. J.: Power-law distributions in empirical data. SIAM Rev. 51, 661–703 (2009)
18. Fitting Power Law Distribution, http://tuvalu.santafe.edu/ aaronc/powerlaws/
19. Magno, G., Comarela, G., Saez-Trumper, D., Cha, M., Almeida, V.: New kid on the block: exploring the Google+ social graph. In: IMC 2012, pp. 159–170. ACM (2012)
20. Alvarez-Hamelin, J. I., DallAsta, L., Barrat, A., Vespignani, A.: K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases. arXiv preprint cs/0511007 (2005).
21. Alvarez-Hamelin, J. I., DallAsta, L., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. Advances in neural information processing systems, 41–50 (2006).
22. Cazals, F. and Karande, C.: A note on the problem of reporting maximal cliques. Theoretical Computer Science, 407(1), 564–568 (2008).
23. Fortunato, S.: Community detection in graphs. Physics reports 486, 3 (2010), 75174.
24. Siganos, G., Tauro, S. L., Faloutsos, M.: Jellyfish: A conceptual model for the as internet topology. Communications and Networks, 8(3), 339–350 (2006).
25. Rossa, F. D, Dercole, F., Piccardi, C.: Profiling core-periphery network structure by random walkers. Scientific reports 3 (2013), 1467.

26. Holme, P. : Core-periphery organization of complex networks. Physical Review E 72, 4 (2005), 046111.
27. Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., Bhattacharjee, B: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 29–42.
28. Garas, A., Argyrakis, P., Rozenblat, C., Tomassini, M., Havlin, S: Worldwide spreading of economic crisis. New journal of Physics 12, 11 (2010), 113043.
29. Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H E., A Makse, H. A: Identification of influential spreaders in complex networks. Nature physics 6, 11 (2010), 888–893.
30. Borgatti, S. P, Everett, M. G: Models of core/periphery structures. Social networks 21, 4 (2000), 375–395.
31. Da Silva, M. R., Ma, H., Zeng, A.: Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. In: Proc. IEEE 96, 8 (2008), 1411–1420.
32. Rossa, F. D., Dercole, F., Piccardi, C.: Profiling coreperiphery network structure by random walkers. Scientific reports 3 (2013), 1467.
33. Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 29–42.
34. Shanahan, M., Wildie. M.: Knotty-centrality: finding the connective core of a complex network. PLoS One 7, 5 (2012), e36579.
35. Wang, L., Hopcroft, J., He, J., Liang, H., Suwajanakorn, S.: Extracting the core structure of social networks using (α, β) communities. Internet Mathematics 9, 1 (2013), 58–81.