RDBMS as an Efficient Tool to Mine Cliques on Complex Networks

Ana Paula Appel, Adriano Arantes Paterlini, Caetano Traina Junior

Computer Sciences Department, ICMC, University of São Paulo
Av. Trabalhador São-carlense, 400 - Centro
Postal code: 668 - CEP: 13560-970 - São Carlos - SP - Brazil
(anaappel, paterlini, caetano)@icmc.usp.br

Abstract. Complex networks are intrinsically present in a wide range of applications. Real world networks have several unique properties, such as, sparsity, node degree distribution, which follow a power law and a large amount of triangles that further form larger cliques. Triangles and cluster coefficient, which are usually used to find groups, are not always enough to distinguish a different node neighborhood topology. By using cliques of sizes 4 and 5, it is possible to study how triangles become involved to form large cliques. To retrieve these cliques called κ_4 and κ_5 a novel technique called "`FCR - Fast Clique Retrieval`" has been developed, taking advantage of the data management and optimization techniques of a relational database management system and SQL to query cliques of sizes 4 and 5. This paper demonstrates that cliques (3, 4 and 5) follow interesting power laws that allow identifying nodes with suspicious behaviors. It also presents an extension of the cluster coefficient formula, which may become a valuable equation to identify nodes that most influence the network first eigenvalue.

Categories and Subject Descriptors: H.2 Database Management [**H.2.8 Database Applications**]: Data Mining

Keywords: cliques, cluster coefficient, graph mining, power law, RDBMS

1. INTRODUCTION

Complex networks, such as biological (protein, DNA), academic (DBLP, Arxiv) and social (Facebook, LinkedIn), have been increasing in size very quickly. Furthermore, complex networks have attracted the interest of research communities with very interesting findings over the past years. Finding patterns in complex networks is extremely important, given that they help detecting abnormalities (outliers) and interesting regions in these networks. If most of the nodes in the network closely follow a power-law, then the few deviations that do exist are probably outliers. To find such patterns, there exists a large number of interesting tasks in complex network mining, such as node degree distribution, betweenness, cluster coefficient, among others [Newman 2003].

The fact that the majority of networks have a high number of triangles [Watts and Strogatz 1998] is common knowledge. For instance, in complex networks, especially in social ones, friends of friends are friends themselves. Plenty of research has investigated the behavior of triangles on a network and how they can indicate the existence of larger cliques [Tsourakakis 2008], [Du et al. 2009].

Cluster coefficient measures the percentage of a node's neighbors that are neighbors to one another [Watts and Strogatz 1998]. It measures the degree of "cliquishness" of a graph. Figure 1 shows two nodes with the same cluster coefficient value ($= 0.2$), but different topologies. The distinction among different topologies is important, given that they can help, for instance, personalizing product recommendation.

Also, when the cliques, specially κ_4 and κ_5, are analyzed on a network, what patterns do they follow? If someone has many "contacts" that are cliques κ_4 and κ_5, does that indicate popularity? The study of κ_4 and

Authors would like to thanks the support of FAPESP, CNPQ and CAPES to the project.

κ_5 allows evaluating the overlapping among the several social circles that one frequents. An example is shown in Figure 1. The two nodes in black represent the differences between the node topology even when they have the same cluster coefficient.

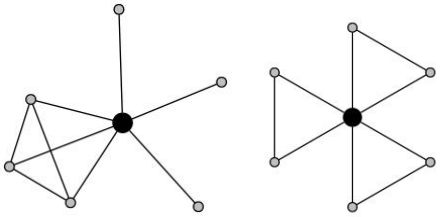


Fig. 1. Two graphs (a) and (b) showing a black node with the same cluster coefficient 0.2, but distinct neighborhood topology. The node in (a) might be more popular that the node in (b).

It is straightforward that cliques of larger sizes, such as sizes 4 and 5 (here called κ_4 and κ_5) can be very useful to spot nodes with different topologies given that their topology shows how the relationship of a node with its neighbors is. Besides, one can have a better picture of the social circles that a network has and also quantify how important a node is to the first eigenvalue. This task is a very important, specially for immunization in epidemics [Chakrabarti et al. 2008] and network resilience [Albert et al. 2000]. In this context, this paper proposes the extension of the cluster coefficient, `Generic Cluster Coefficient`, allowing the distinction of nodes in situations like the one depicted in Figure 1 and identification of nodes that most influence the complex network first eigenvalue.

Another interesting contribution of this work is the *Degree Clique Law*, which shows that not only triangles (κ_3), but also cliques of other sizes, such as κ_4 and κ_5, follow a power law correlated with the node degree. Moreover, the relations between the cliques and clique distribution follow the power laws named *Power Clique Law* and *Clique Distribution Law*. These power laws help to investigate nodes with suspicious behaviors like spammers. For example, a fake user could mimic a small social circle (triangles) by adding a person and some of his/her friends, however it would be more difficult to mimic a large social circle (cliques of size 4 and 5).

Considering that complex networks can be already stored in a relational table, the aim is to verify how feasible it is to use an RDBMS to retrieve κ_4 and κ_5 cliques. The last 40 years have proven how database query languages are valuable to access a large amount of data. SQL queries are easier to modify and understand [Rustin 1974]. Also, all of the modern Relational Database Management System (RDBMS) use hash or B-tree indexes to accelerate data access. Most database systems also support multiple indexes per table. Thus, the query optimizer can decide which index to use for each query or whether to simply perform a brute-force sequential search [Pavlo et al. 2009].

To find κ_4 and κ_5 a novel technique called `FCR - Fast Clique Retrieval` has been proposed. It is based on a RDBMS and allows users to use SQL to find these cliques in a faster way. This technique breaks the network into small ones to find κ_4 and κ_5 more efficiently and can be up to **5x** faster for κ_4 and **12x** faster for κ_5 than the direct processing, named here Standard Approach.

This paper is organized as follows: Section 2 introduces the graph terminology and the symbols used in the paper; Section 3 presents the existing related work; Section 4 proposes the `Generic Cluster Coefficient`; Section 5 explains the proposed method; Section 6 presents the `FCR` scalability; Section 7 discusses the laws found over real-world complex networks and, finally, Section 8 concludes the work.

2. TERMINOLOGY

Table I shows the symbols employed in this paper. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph without self-edges, where $\mathcal{V}$ is the set of nodes, also called vertices, and $\mathcal{E}$ is the set of edges. For a given node $v_i \in \mathcal{V}$,

Table I. Symbols used in this work

Symbol	Description
G	a graph
$\mathcal{E}$	edges of a graph G
$\mathcal{V}$	nodes of a graph G
G_s	a subgraph of a graph G
$\mathcal{E}_s$	edges of a subgraph G_s
$\mathcal{V}_s$	nodes of a subgraph G_s
$d(v_i)$	degree of node v
$\mathcal{N}(v_i)$	neighborhood of node v
κ_t	a clique of size t
$\kappa_t(v_i)$	number of cliques of size t of node v_i
$C(v_i)$	Cluster Coefficient of node v_i
$C_k(v_i)$	Generic Cluster Coefficient of node v_i until clique of size k
u, v, w	nodes of graph G
λ_1	First eigenvalue of graph G
λ'_1	First eigenvalue of graph G without node v_i
λ_D	*eigenvalue drop*

$\mathcal{N}(v_i) = \{(v_i, w) | v_i, w \in \mathcal{V} \wedge (v_i, w) \in \mathcal{E}\}$ is defined as the neighborhood of node v_i. The number of edges in the neighborhood of node v_i is called its *node degree* $d(v_i)$.

DEFINITION 1. ***Subgraph:*** *$G_s = (\mathcal{V}_s, \mathcal{E}_s)$ is a subgraph of $G = (\mathcal{V}, \mathcal{E})$ if $\mathcal{V}_s \subseteq \mathcal{V}$ and $\mathcal{E}_s \subseteq \mathcal{E}$.*

DEFINITION 2. ***Induced Subgraph****: G_s is an induced subgraph of G if $\mathcal{V}_s \subseteq \mathcal{V}$ and $\mathcal{E}_s$ contains all edges of $\mathcal{E}$ that connect nodes in $\mathcal{V}_s$.*

DEFINITION 3. ***complete graph or clique:*** *κ_t is a graph with t nodes such that for every node pair $u, v \in \mathcal{V}$ there is an edge $(u, v) \in \mathcal{E}$.*

In a graph G, a subset of nodes $V_s \subseteq \mathcal{V}$ is a clique of size t (κ_t) if the induced subgraph G_s on G is a complete graph of size t. A clique of size $t = 4$ is called κ_4 and $t = 5$ is κ_5. The number of cliques of size t that a node v_i participates in is represented as $\kappa_t(v_i)$.

The clustering coefficient $C(v_i)$ of a node v_i, given by Equation 1, is the proportion of edges between the node within its neighborhood divided by the number of edges that could possibly exist between them.

$$C(v_i) = \frac{2 * \kappa_3(v_i)}{d(v_i) * (d(v_i) - 1)} \tag{1}$$

where $\kappa_3(v_i)$ is the number of triangles containing v_i.

In spectral graph theory, an eigenvalue (λ) of a graph is defined as an eigenvalue of the graph's adjacency matrix A represented by $Ax = \lambda x$, where A is the adjacency matrix of graph $\mathcal{G}$, x is a vector and λ is an eigenvalue [Mihail and Papadimitriou 2002]. The set of eigenvalues of graph $\mathcal{G}$ is called *graph spectrum.* Recent developments in spectral graph theory have concerned the effectiveness of eigenvalues in the study of general graphs. An example is Google's PageRank algorithm based on graph's eigenvector [Page et al. 1998]. Usually, the eigenvalue of a graph means the graph connectivity. For example, in [Chakrabarti et al. 2008] the authors prove that the first (highest) eigenvalue (λ_1) is the epidemic threshold, which is very important to the prevention of a contagious disease spread over a population. They also state that the node that most affects the first eigenvalue should be the one to be vaccinated.

3. RELATED WORK

There exist a significant amount of research related to the problem in focus, which we categorize as cliques and other subgraphs, triangles, cluster coefficient, communities structure, and power law distributions.

Cliques and other subgraphs: The retrieval of either quasi-cliques or the largest clique in a graph have been studied by a large number of researchers [Liu and Wong 2008], [Zeng et al. 2007], [Modani and Dey 2008], [Stix 2004], [Du et al. 2009]. However, these works aimed to find maximal or quasi-cliques of any size, while the present work focuses only on cliques of sizes 4 and 5. One of the most pursued recent tasks in graph mining is how to discover subgraphs that frequently occur over a database with several graphs [Wang et al. 2005], [Han et al. 2007]. Many works, such as [Chakravarthy et al. 2004] and [Chakravarthy and Pradhan 2008] use a Relational Database System (RDBMS) to find the FSG (Frequent SubGraph), however, an FSG does not need to be a clique.

Triangles, cluster coefficient and Communities Structure: The network transitivity can be measured through the cluster coefficient [Watts and Strogatz 1998]. It is found that, in many networks, if node v is connected to node u and node u to node w, then there is a high probability that node v will also be connected to node w. In social networks, this means that a friend of your friends is also likely to be your friend. In terms of network topology, transitivity means the presence of many triangles in the network, which is a triad of three nodes connected among themselves.

Several recent works, such as [Becchetti et al. 2008] and [Tsourakakis 2008], have aimed to count triangles without identifying them. For instance, the eigenvalue multiplication is used to find out the total number of triangles. In [Latapy 2008], the author proposes a fast algorithm to count triangles in graphs with a degree distribution that follows a power law. A triangle is also a cycle of size 3, as in the works of [Fronczak et al. 2002], [Caldarelli et al. 2004], which proposed a different cluster coefficient to count cycles of different sizes. However, those techniques count cycles, not cliques.

The cluster coefficient tends to be considerably greater for real networks than for a random graph with similar numbers of nodes and edges. The cluster coefficient is also known to be dependent on the node degree [Dorogovtsev et al. 2002], [Ravasz and Barabsi 2003]. The cluster coefficient $C(v_i)$ of a node v_i decreases as its degree $d(v_i)$ increases, by following a power law for models like scale-free networks[Ravasz and Barabsi 2003]. This means that low-degree nodes tend to form highly connected groups, which are connected to each other and form larger groups. The presence of these larger groups, that is, cliques of size larger than 3, explains the "Small World" phenomenon [Watts and Strogatz 1998] and how the "Diameter evolves over time" [Leskovec et al. 2007]. In a social network these groups are seen as communities, where the edges between nodes represent friendship and nodes represent people. This property corresponds to the fact that people are more related to people from their own communities and less connected to people outside them.

In [Leskovec et al. 2008], the authors show that communities tend to be quite small, with no more than approximately 100 nodes, and barely connected to the rest of the network. Value 100 is known as Dunbar's number, which is the number of connections that a person can handle [Dunbar 1998]. Also, most graphs exhibit a *jellyfish* pattern [Tauro et al. 2001], which is a graph with a core that is a clique of high-degree nodes, and also a first layer whose nodes are adjacent to the core. Nodes in the first layer have more one-degree nodes connected to them than to the core.

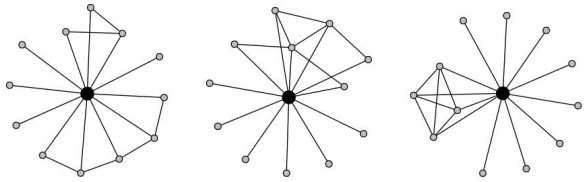
Power law distributions: Power-law distributions occur in many types of graphs of scientific interest and have significant consequences to the understanding of both natural and man-made phenomena. The growth of city populations, earthquake intensities and power outage ranges are very well known examples that follow power-law distributions. In graphs, we can highlight the node degree distribution [Chakrabarti and Faloutsos 2006], triangles distribution over node degree [Tsourakakis 2008], eigenvalue distribution [Faloutsos et al. 1999] and others all following power-law distributions. A distribution is a Power law if its probability density function (PDF) is as follows: $p(x) \propto x^{-\alpha}$, where $p(x)$ is the probability of x and α is the power law exponent [Newman 2005], [Clauset et al. 2009].

4. GENERIC CLUSTER COEFFICIENT

The analysis of a node neighborhood topology is an important task in social network mining. The Cluster Coefficient of a node indicates how strong the connectivity among its neighborhoods is. However, nodes that have the same degree and/or the same number of triangles are not always equivalent in terms of topology and connectivity, as the traditional Cluster Coefficient forces a social homogenization among the nodes of the network.

Figure 2 presents three subgraphs centered at nodes u, v and w. They have the same degree ($d(u) = d(v) = d(w) = 12$) and the same number of triangles ($\kappa_3(u) = \kappa_3(v) = \kappa_3(w) = 6$), hence the same cluster coefficient $\frac{2*6}{12*(12-1)} = 0.09$. However, they have different numbers of κ_4 ($\kappa_4(u) = 0$, $\kappa_4(v) = 1$, $\kappa_4(w) = 4$).

In a social network, nodes can be viewed as people and edges as friendship. Node u shows a stronger connectivity with a selected group of neighbors (friends), while node v shows a more uniform relationship of all its neighbors (friends). A node with a topology showing fewer $\kappa_4(v_i)$ and $\kappa_5(v_i)$ than others, but with same $d(v_i)$ and $\kappa_3(v_i)$, like node u, can represent a node that interacts clearly with more than one social cycle, given that most of its neighbors are disconnected among themselves. The distinction of node neighborhood topology is an interesting task for system recommendation and personalization, since it can help finding groups of nodes that interact more clearly and can buy the same product or service.



(a) center node u (b) center node v (c) center node w

Fig. 2. Three subgraphs where the center nodes u, v, w have the same degree $d(u) = d(v) = d(w) = 12$ and $\kappa_3(u) = \kappa_3(v) = \kappa_3(w) = 6$: (a) only triangles centered at node w, (b) one κ_4 centered at node v, (c) four κ_4 and one κ_5 centered at node u

Nodes with different topologies like, u, v and w in Figure 2 can play different roles in the network. If someone wants to know how closely a person interacts in his/her social groups, this interaction can be represented by node topology. For example, node w is probably a node that has a strong interaction with a group of friends, since they form a clique of larger size, while node u interacts almost the same way with all of its friends. Traditional Cluster Coefficient can not be used in situations like the one presented in Figure 2, since it is the same for all three nodes presented.

Based on this fact, we propose a modification in the cluster coefficient definition that considers a larger clique size. Besides, it is possible to emphasize the differences in node connectivity, as in the nodes shown in Figure 2, adding components to the original cluster coefficient up to a desired size t of counted cliques.

The maximum number of possible cliques of size t given the number of edges is obtained through the use of a combinatorial equation: $\binom{d(v_i)}{t-1} = \frac{d(v_i)!}{(t-1)!\cdot(d(v_i)-(t-1))!}$. Thus, a `Generic Cluster Coefficient` is defined in Equation 2 for a node v_i.

$$C_k(v_i) = \sum_{j=3}^{k} \frac{\kappa_j(v_i)}{\binom{d(v_i)}{j-1}} \tag{2}$$

The `Generic Cluster Coefficient` can be used with any value of t, that is, it will work for cliques of any size that can be found in the network. However as the size t increases, the computational cost to recover

all cliques of size up to t becomes unfeasible. For the proposal of this paper the use of $t = 5$ is enough and feasible. As a rule of thumb enough cliques occur when different topologies have different coefficient values. Equation 3 presents the `Generic Cluster Coefficient` considering the cliques until κ_5. The `Generic Cluster Coefficient` removes the social homogenization imposed by the traditional cluster coefficient.

$$C_5(v_i) = \frac{2 * \kappa_3(v_i)}{(d(v_i) * (d(v_i) - 1))} + \frac{6 * \kappa_4(v_i)}{(d(v_i) * (d(v_i) - 1) * (d(v_i) - 2))} + \frac{24 * \kappa_5(v_i)}{(d(v_i) * (d(v_i) - 1) * (d(v_i) - 2) * (d(v_i) - 3))} \quad (3)$$

in which $\kappa_3(v_i)$ is the number of triangles, $\kappa_4(v_i)$ is the number of 4-sized cliques, $\kappa_5(v_i)$ is the number of 5-sized cliques and $d(v_i)$ is the degree for node v_i.

4.1 Eigenvalue Influence Observation

One of the interesting observations is that the number of cliques which a node participates in has a high influence on the eigenvalue of the graph. First, we need to define the *eigenvalue drop* to explain node influence. The *eigenvalue drop* is defined as the original value of the first eigenvalue of a graph minus the first eigenvalue measured from the graph without the node whose influence we want to analyze. Then *eigenvalue drop* is given by the formula $\lambda_D = \lambda_1 - \lambda'_1$, where λ_D is the *eigenvalue drop*.

The eigenvalue experiment was carried out as follows: first, the first eigenvalue (λ_1) of graph G was measured, then a node v_i was removed from graph G, the first eigenvalue (λ'_1) was measured again and the *eigenvalue drop* $\lambda_{D_i} = \lambda_1 - \lambda_{1'_i}$ was computed. Node v_i was put back and another node v_{i+1} was removed. Then $\lambda_{1'_{i+1}}$ and $\lambda_{D_{i+1}}$ were computed again. The process was repeated until all the chosen nodes had been deleted.

The *eigenvalue drop* is usually related to node degree. Thus, by following this idea, two nodes with the same degree should have the same *eigenvalue drop*. However, as showed in Table II we can see that this is not true. Actually, nodes with different numbers of cliques (3, 4 and 5) influence the *eigenvalue drop* in different ways. Notice that for each row of Table II, the node analyzed and its edges were deleted and before the next node was deleted, the node and its edges had been reinserted.

Table II. Four nodes of AS network with the same degree ($d(v_i) = 7$) and $\kappa_3 = 7$, hence the same cluster coefficient (=0.16), but different influence on the *eigenvalue drop*.

κ_4	κ_5	*eigenvalue drop*	$C_5(v_i)$
0	0	0.003445	0.16
2	0	0.003946	0.28
3	0	0.004391	0.33
4	1	0.004746	0.43

5. PROPOSED METHOD

This section presents the proposed algorithm, which not only counts the numbers $\kappa_4(v_i)$ and $\kappa_5(v_i)$ of a given node v_i, but also lists them. Networks can be viewed as relationships on a database and many times are already stored on the database. Thus, the management and optimization techniques provided by the Relational Database Management System (RDBMS) to retrieve κ_4 and κ_5 from a graph can be explored. However, the conventional graph scheme, which stores only the edges file on the database, here called "Standard Approach", requires too

many join operations to retrieve these cliques and since most graphs are very large, it requires a long time to process a reasonably-sized graph. One of the techniques used to speed up queries is to store partial information used in these queries. Therefore, we propose `FCR - Fast Clique Retrieval`, which breaks the graph into small subgraphs that allow retrieving κ_4 and κ_5 for each node in a faster way.

In `FCR`, the graph is first divided into n small subgraphs, where n is the number of nodes, as presented in details in Algorithm 1 and in the following explanation: A subgraph $G_{s_i} = (\mathcal{V}_{s_i}, \mathcal{E}_{s_i}), i \in \{1, \ldots n\}$ is the induced subgraph centered at node v_i, such that $\mathcal{V}_{s_i} = \{u | u \in \mathcal{V} \wedge (v_i, u) \in \mathcal{E}\}$ and contains all edges of $\mathcal{E}$ that connect nodes in $\mathcal{V}_{s_i}$, which contains the neighborhood of a node v_i and each edge that connect these nodes. Each edge of $\mathcal{E}_{s_i} - \mathcal{N}(v_i) = \{(u, w) \in \mathcal{E}_{s_i} \wedge (u, w) \notin \mathcal{N}(v_i)\}$ is stored in an additional table, called "`Subgraph table`". Thus, the database needs only one extra table besides the edges file containing all edges of G. For each edge stored in the `Subgraph table`, there exist the edge source and destination and an identification attribute $graphi$, which is index i of subgraph G_{s_i}. There exist B-tree indexes in both tables for all the attributes in order to take advantage of the optimization of RDBMS.

Algorithm 1 `Subgraph table` creation

Require: The input graph $G(N, E)$
Ensure: Loaded `Subgraph table`
1: Create edges table
2: Insert graph G in the edges table
3: Create the `Subgraph table`
4: Divide graph G in to $|N| = n$ subgraph
5: **for** i=1 to n **do**
6: **for all** edges connecting neighbors of node v_i **do**
7: Label these edges as v_i.
8: Insert these edges in the `Subgraph table`
9: **end for**
10: **end for**
11: create index in both tables

The `Subgraph table` keeps the edges of all the subgraphs, except the ones that neither have edges connecting their neighbors and nor one-degree nodes. Since most of the real graphs are sparse, i.e., they have many one-degree nodes, this approach does not need an unfeasible extra disk space.

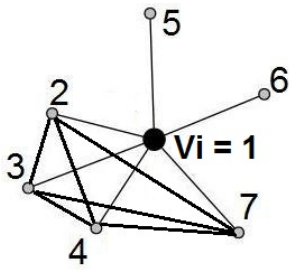
Figure 3 (a) shows an example of a graph $G = (\mathcal{V}, \mathcal{E})$ and the corresponding node $v_i = 1$ as a black dot in the graph. This graph will be used to explain how the `FCR` works. The edge table of G, in this paper exemplified by Figure 3 (a), is stored on the database as a relational table, with some tuples shown by Figure 3(b). Part of one `Subgraph table` is represented in Figure 3 (c), which shows some of the edges of subgraphs G_{s_i} in Figure 3 (a). An example of the amount of tuples in the `Subgraph table` compared with the number of total edges of a graph and the amount of cliques (3, 4 and 5) will be presented in Section 6.

The idea to find all κ_4 of a node v_i is the following:

(1) Retrieve the node neighborhood $\mathcal{N}_{v_i}$;
(2) For each edge (v_i, v_j), retrieve all edges (v_i, v_j, z) in the `Subgraph table`
(3) Check if the two-by-two combination of index z of all retrieved edges is in G.

Now, following the steps above to find the $\kappa_4(v_i)$ of node $v_i = 1$ from Figure 3 (a) we have:

(1) The neighborhood of node v_i, $(1, 2); (1, 3); (1, 4); (1, 5); (1, 6); (1, 7)$, is retrieved.
(2) For each retrieved edge, check in the `Subgraph table` if it belongs to any subgraph. For example, for edge $(1, 2)$, the following tuples are retrieved: $(1, 2, 3); (1, 2, 4); (1, 2, 7)$. Edge $(1, 2)$ belongs to $\mathcal{E}_{s_i}$ of node 3, 4 and 7.



SRC	DEST
1	2
1	3
1	4
1	5
1	6
1	7
2	1
2	3
3	4
...	...

SRC	DEST	i
2	3	1
2	4	1
3	2	1
3	4	1
4	2	1
4	3	1
1	2	3
1	2	4
1	2	7
...	...	...

(a) - A graph G (b) - Edge File (c) - Subgraph Table

Fig. 3. (a) a graph G and node $v_i = 1$, whose edges, represented by thick lines, are stored in the `Subgraph table`. Table (b) is the traditional edges table (the only one used in the Standard Approach). Table (c) is the `Subgraph table` that stored the edges from subgraphs G_{s_i}.

(3) Thus, there exists a $\kappa_4(v_i)$ if there is an edge between the nodes retrieved, i.e., it is necessary to check if edges $(3,4);(3,7);(4,7)$ exists Each of them represents the existence of one $\kappa_4(v_i)$.

The SQL query to retrieve all κ_4 from Epinions network is presented in Figures 4 and 5. First, the SQL for the standard approach, which requires 11 joins to retrieve all κ_4, is presented.

Clique κ_4 is composed of 6 edges (a,b,c,d,e,f) that correspond to each table in the query presented in Figure 4. The join operation in SQL query corresponds to the graph nodes that compose κ_4. The intuition of the SQL query is represented by the graph in Figure 6.

```
SELECT a.src, a.dest, b.dest, c.dest
FROM epinions a, epinions b, epinions c, epinions d, epinions e,
     epinions f
WHERE and a.src  =  b.src  and a.src  =  c.src  and d.src  = a.dest
      and d.dest =  c.dest and e.src  =  b.dest and e.dest = c.dest
      and f.src  =  a.dest and f.dest =  b.dest and a.dest <> b.dest
      and a.dest <> c.dest and b.dest <> c.dest;
```

Fig. 4. SQL query to retrieve all κ_4 from Epinions network using the standard approach

By using the `Subgraph table`, only 4 join operations are needed, as represented by the query in Figure 5. This query is simpler and more efficient than the one in Figure 4. The improvement is obtained by avoiding the verification of an edge existence more than once.

```
SELECT a.src, a.dest, m.src, m.dest
FROM epinions a, epinionssubgraph m, epinionssubgraph n
WHERE a.src = m.grafhi and a.dest = n.grafhi and m.src = n.src
      and m.dest = n.dest;
```

Fig. 5. SQL query to retrieve all κ_4 from Epinions network by using the `Subgraph table`

To find all κ_5 of a node v_i, one can follow the steps below:

(1) Retrieve the node neighborhood $\mathcal{N}_{v_i}$;
(2) For each edge (v_i, v_j) retrieve all edges (v_i, v_j, z) in the `Subgraph table`;
(3) Check if the three-by-three combination of index z, (z_1, z_2, z_3) corresponds to a tuple in the `Subgraph table`;

Fig. 6. Graph representing the following SQL queries: (a) Figure 4 - (b) Figure 5

Now, instancing the steps above to find the $\kappa_5(v_i)$ of node $v_i = 1$ from Figure 3 (a) we have:

(1) The neighborhood of node v_i, $(1,2;1,3;1,4;1,5;1,6;1,7)$ is retrieved.
(2) For each retrieved edge, check in the `Subgraph table` if it belongs to any subgraph. For instance, for edge $(1,2)$ the following tuples are retrieved: $(1,2,3;1,2,4;1,2,7)$. Edge $(1,2)$ belongs to $\mathcal{E}_{s_i}$ of node 3, 4 and 7.
(3) Thus, there exists a $\kappa_5(v_i)$ if there are edges among the nodes retrieved, i.e., it is necessary to check if tuple $(3,4,7)$ is in the `Subgraph table`.

Actually, using the `Subgraph table` it is necessary to verify if tuple (z_1, z_2, z_3) in in `Subgraph table`, instead of checking if there are three tuples ((z_1, z_2), (z_1, z_3) and (z_2, z_3)) in the edges table.

The SQL query to retrieve all κ_5 from Epinions network is presented in Figure 7 and Figure 8. First, the standard approach, which requires 21 join operations. Second, the `FCR` approach, which requires only 7 join operation, reducing the computational cost.

```
SELECT a.src, a.dest, b.dest, c.dest, d.dest
FROM epinions a, epinions b, epinions c, epinions d, epinions e,
     epinions f, epinions g, epinions h, epinions i, epinions j
WHERE a.src = b.src and a.src = c.src and a.src = d.src
      and a.dest =  e.src  and a.dest =  g.src  and a.dest =  i.src
      and b.dest =  e.dest and b.dest =  h.src  and b.dest =  j.src
      and c.dest =  g.dest and c.dest =  j.dest and c.dest =  f.src
      and d.dest =  f.dest and d.dest =  h.dest and d.dest =  i.dest
      and c.dest <> d.dest and a.dest <> b.dest and a.dest <> c.dest
      and a.dest <> d.dest and b.dest <> c.dest and b.dest <> d.dest;
```

Fig. 7. SQL query to retrieve all κ_5 from Epinions network by using the standard approach

```
SELECT z.src, z.dest, z.grafoi, a.src, a.dest
FROM epinionssubgraph z, epinionssubgraph a,
     epinionssubgraph b, epinionssubgraph c
WHERE z.src = a.grafoi and z.dest = b.grafoi and z.grafoi = c.grafoi
      and a.src  = b.src  and a.src  = c.src
      and a.dest = b.dest and a.dest = c.dest;
```

Fig. 8. SQL query to retrieve all κ_5 from Epinions network by using the `Subgraph table`

In practice, to find all κ_5 of a node, almost the same proceedings to find all κ_4 of a node are followed. The difference is that the existence of three edges connected is checked, instead of one edge as in κ_4. It is easy for the `FCR` to find these tree edges, since they are connected and stored as a tuple in the `Subgraph table`. The total number of edges stored in the `Subgraph table` for each graph is presented in Table III.

6. SCALABILITY

This section shows that the FCR – Fast Clique Retrieval is up to 5 and 12 times faster than the Standard approach to query all κ_4 and κ_5, respectively. The evaluation was performed by using the open source RDBMS PostgreSQL 8.3.7 on a computer equipped with an Intel Core2 Quad 2.83GHZ processor and 4Gb of RAM. Table IV and Figure 9 present the results of the experiments to evaluate how fast FCR is to query all κ_4 and κ_5. The time was measured in seconds and the values shown are the average of three executions, with the cache being cleaned before each execution. To control the experiment, we used a graph based on US cities, that is, a graph composed of a set of latitudes and longitudes of 25,375 US cities. The graphs used to evaluate the scalability were created based on a US cities dataset by using a k-nearest neighbor query for each node, varying $k \in \{4, 5, 7, 10, 15, 20, 25\}$. Thus, every graph has the same number of nodes, but a varied number of edges, as described in Table III, which also reports the number of edges stored in the Subgraph table. All graphs are considered undirected. The highest speed is reached when $k = 25$, that is, querying all the κ_4 by using FCR is 4 times faster and querying all the κ_5 using FCR is 12 times faster than by using the standard approach.

Table III. US Cities dataset information (nodes, edges and edges in Subgraph table).

Dataset Information				
K-NN	**Edges**	Subgraph table	κ_4	κ_5
4	62,401	250,044	241,584	21,744
5	77,104	418,332	640,440	121,896
7	106,283	882,324	2,391,816	946,560
10	149,760	1,889,838	8,571,576	6,305,136
15	222,348	4,411,422	33,506,928	43,521,504
20	295,239	8,001,498	85,459,368	160,675,416
25	368,308	12,690,354	175,523,400	436,345,200

Table IV. Average execution time (in seconds) to count κ_4 and κ_5 for each node using both FCR and the Standard approach with the US Cities dataset.

K-NN	**Time κ_4**		**Time κ_5**	
	FCR	**Standard**	FCR	**Standard**
4	2.1	6.4	4.2	21.2
5	4.0	12.0	15.8	44.2
7	12.4	33.0	19.8	116.3
10	16.2	93.5	88.9	607.3
15	278.1	327.0	500.8	4,156.9
20	1,105.1	1,284.6	1,720.7	21,214.8
25	**6,731.1**	**29,579.5**	**4,894.3**	**60,593.4**

The second scalability experiment executed a query for a thousand nodes from RNT (our largest real graph in number of nodes and edges) which were randomly chosen. Figure 9 presents the average time per query of both the FCR and the standard approach. Before each query, the cache of the computer and the database were cleaned up. As we can see, the FCR approach is up to 8 times faster for κ_4 and 9 times faster for κ_5, when compared to the standard approach, which uses only the edges table stored in the RDBMS. As shown in our experiments, the FCR is a feasible approach to retrieve κ_4 and κ_5.

7. PATTERNS AND OBSERVATIONS

The findings about the complex networks tested are shown here. First, a description of the datasets is given and then the patterns for the real world networks are studied. Three newly discovered patterns the datasets seem to contain are also presented.

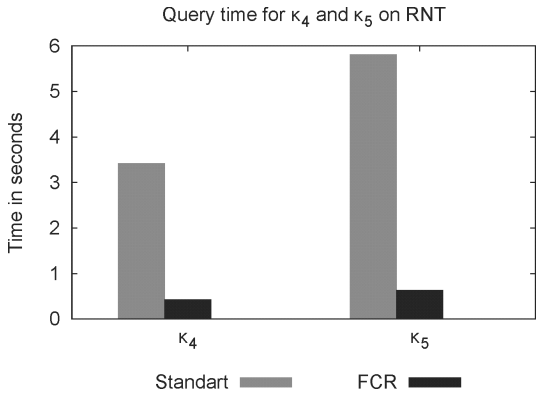


Fig. 9. Average time over thousand queries on RNT graph.

The first is the `Clique-Degree`, which shows that cliques κ_4 and κ_5 as κ_3 also exhibit a correlation with the node degree distribution. The second is the `Power Clique`, which presents the correlation among cliques, for example κ_4 vs. κ_3 and so on. The last one is the `Clique Distribution`, which shows that the amount of cliques κ_3, κ_4 and κ_5 follows a power law very close to the node degree distribution power law.

7.1 Dataset description

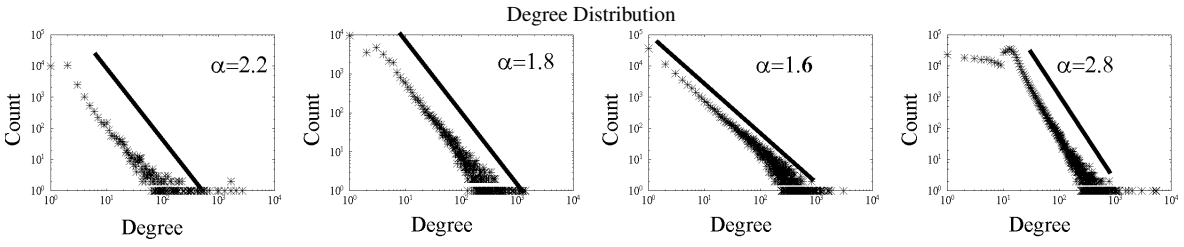
The `FCR` approach was tested with several network datasets, however, only the results of four datasets are presented, as the other have similar behavior. All datasets are undirected and do not contain self-edges. Figure 10 presents the node degree distribution of the evaluated datasets detailed below:

—**AS-Network**: The Internet can be organized into subgraphs called Autonomous Systems (AS). Each AS exchanges traffic flows with its neighbors (peers). A communication network of who-talks-to-whom from the BGP (Border Gateway Protocol) and its logs can be used to build a graph. We used the AS-Network dataset from Caida [asc 2007]. It has 26,389 nodes and 52,861 edges.

—**Email-Enron network**: It is a social network that contains data from users of Enron company [Klimt and Yang 2004].It has 33,696 nodes and 180,811 edges.

—**Epinions Network**: It is a real social network of who-trusts-whom from Epinions [Richardson et al. 2003], where nodes represent people and edges represent relationships. It has 75,877 nodes and 405,739 edges.

—**Recommendation (RNT)**: The RNT [Clauset et al. 2004] represents information about purchases in a store, where the nodes represent items and there is an edge from item v to another item u if u is frequently purchased by buyers of item v. It has 473,315 nodes and 3,505,519 edges.



(a) AS (b) Email-Enron (c) Epinions (d) RNT

Fig. 10. Node degree distribution of AS, Email-Enron, Epinions and Recommendation Network.

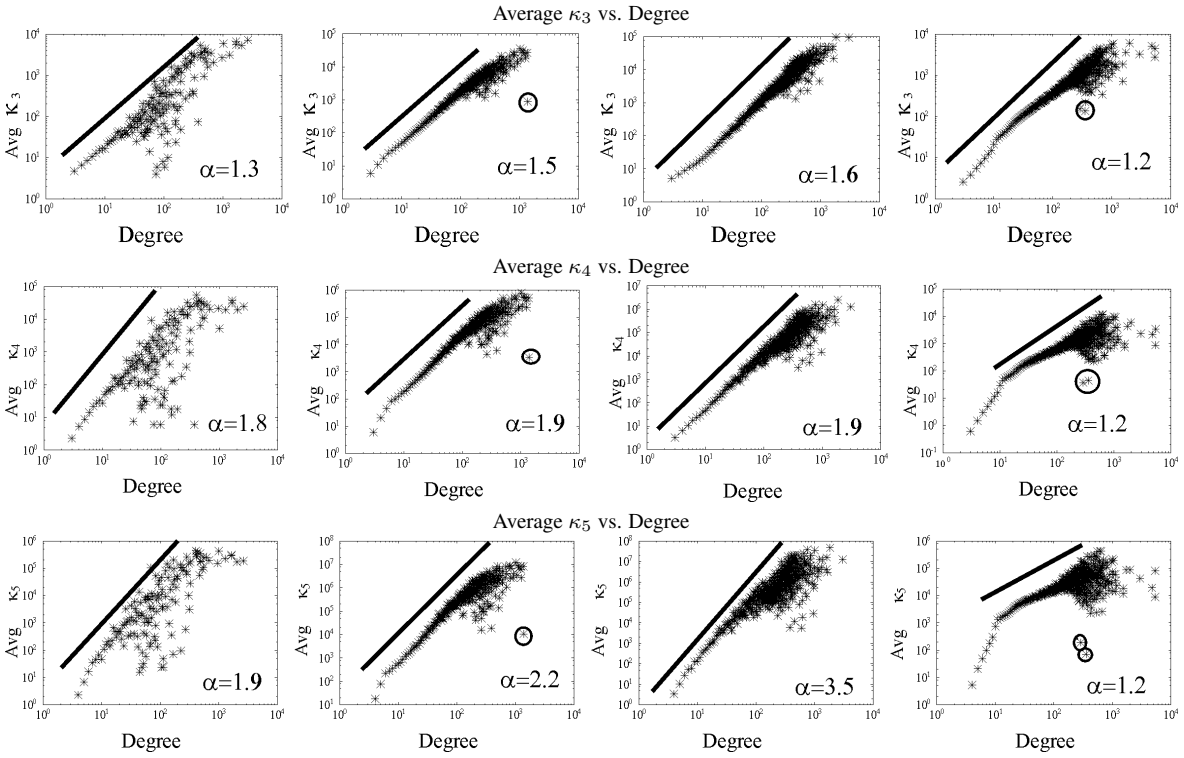
7.2 Cliques Power Laws

Figure 11 presents the average κ_t ($\overline{\kappa}_t$) for $t \in \{3, 4, 5\}$ versus node degree (d). All the plots of Figure 11 exhibit a correlation between degree and clique distribution, that is, all cliques, including κ_3 presented in [Tsourakakis 2008], follow a power law. Thus, it is possible to generalize the power law as being the *Degree Clique Law*.

Degree-Clique Law. The relation between the average number of t-sized cliques vs. degree d of a network that follows a power law with exponent $\alpha > 0$.

$$\overline{\kappa}_t = d^{\alpha} \tag{4}$$

where $\overline{\kappa_t}$ is the average number of t-sized cliques, d is the node degree and the power law exponent is $\alpha > 0$.



(a) AS (b) Email-Enron (c) Epinions (d) RNT

Fig. 11. Average of $\kappa_t(v_i)$ ($t \in 3, 4, 5$) vs Degree for Epinions and Recommendation Network.

As t increases, more nodes tend to deviate from the tendency shown in Figure 11. In most networks high-degree nodes have many one-degree nodes connected to them, which decreases the probability of a node participation in a clique. However, they still participating in many more cliques than in nodes with an inferior degree.

The nodes that deviate in this law are nodes that have fewer t-sized cliques than expected, as the two highlighted points in Figures 11 (b) and (d). These plots present the average of t-sized cliques with $t \in 3, 4, 5$ vs. degree. The highlighted points are high-degree nodes that are expected to have a higher number of t-sized cliques with $t = 3, 4, 5$, but as the plots show, they have fewer cliques. Thus, they are nodes that present a suspicious behavior and should be analyzed carefully. However, due to the lack of detailed information about the network, it is not possible to identify and analyze these nodes more deeply.

Figure 11 also shows that in these networks (columns (b), (c) and (d)) most of the nodes start deviating around degree 100, which is the best size for a community structure. Actually, communities tend to achieve the well-functioning size around 100 nodes. In the RNT dataset, given that it is a recommendation network, the deviation of high-degree nodes can be explained by the fact that it is rare to buy a large number of products together. Thus, it is improbable for high-degree node to have a large number of cliques.

A surprising observation occurs with the RNT dataset, Figure 11 (d), which seems to follow two different tendencies: one up to degree 10 and the other above degree 10, which is exactly the same point where the RNT degree distribution shown in Figure 10 has a "bend" in its tendency.

It is interesting to observe that the results of the Email-Enron network are almost identical to the Epinions graph, not only κ_t versus node degree (shown in Figure 11 (b) and (c)) but also the degree distribution (Figure 12 (b) and (c)). Thus, Email-Enron and Epinions have the same behavior, what is explained by the fact that both are social networks. Moreover, in Email-Enron there is one highlighted point, corresponding to the highest degree node in Figure 11 (b) that significantly deviates from all the other nodes in the plots. This node has a spammer behavior, since it has fewer cliques than the second highest degree nodes.

Table V. The $\kappa_3(G)$, $\kappa_4(G)$ and $\kappa_5(G)$ all networks.

Network	$\kappa_3(G)$	$\kappa_4(G)$	$\kappa_5(G)$
AS-network	205,590	1,184,544	9,169,320
Email Enron	4,345,594	56,177,760	697,083,240
Epinions	9,746,886	139,281,528	2,090,091,840
RNT	39,772,974	204,902,448	973,945,920

Table V presents the total numbers of κ_3, κ_4 and κ_5 for all four networks. As one can observe, the RNT has the largest number of κ_3 and κ_4. However, Epinions has many more κ_5 and fewer edges and nodes than the RNT, which means that it is more clustered than the RNT.

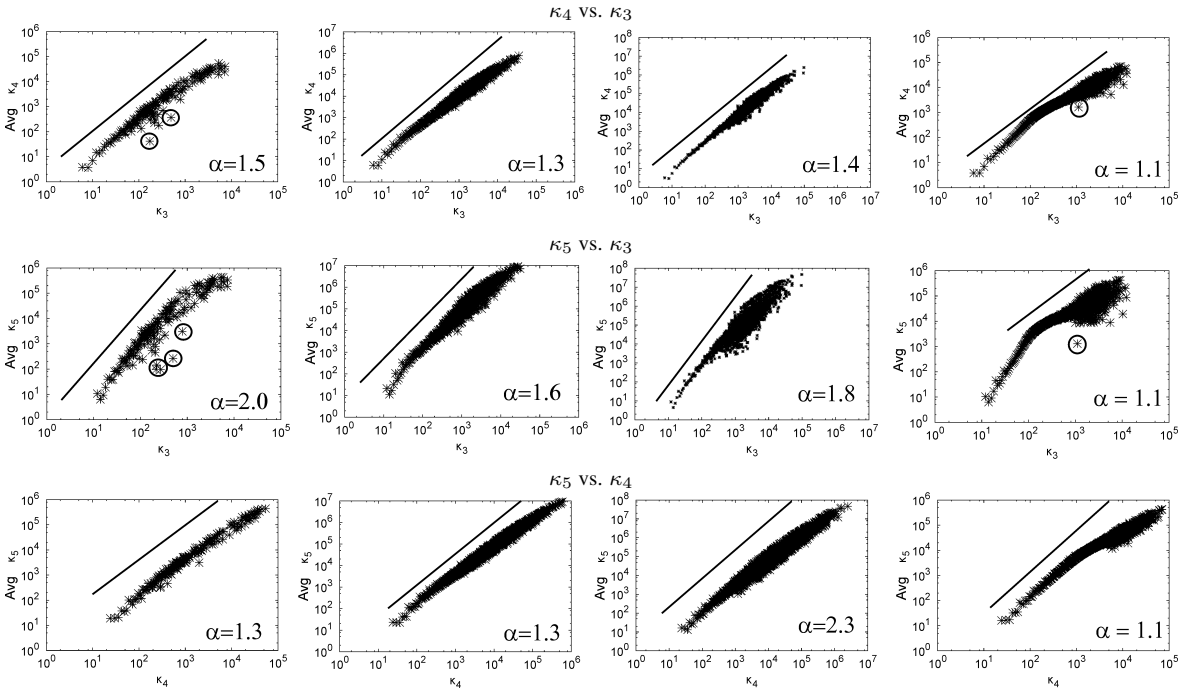
Figure 12 depicts the correlation of κ_{t+1} versus κ_t with $t = 3, 4, 5$. These relations show, for instance, how many κ_4 are on average κ_5. One can observe that all plots follow power laws, i.e., the triangles of a node tend to be connected to other triangles becoming larger cliques, such as κ_4 and κ_5. The higher number of $\kappa_4(v_i)$ and $\kappa_5(v_i)$ proves that the networks are very well connected exhibiting a network community structure. Epinions and Email-Enron networks, which are social networks are the ones that have fewer nodes deviating from the tendency. This fact was expected, since people in social networks usually take part in large social groups, like schools, sports and so forth. On the other hand, the RTN has fewer κ_4 and κ_5 than Epinions. This is also expected, since it is a recommendation network and most of the people buy fewer products at the same time. A surprising observation is related to the AS network, as most of the nodes that deviate from the tendency (highlighted nodes in Figure 12 column (a)) are not the high-degree nodes. This is explained by the AS topology, given that the high-degree nodes are connected and form a core and the other layers have average-degree nodes. Thus, the nodes on the AS network that deviate from the pattern are probably from the second and the third layers.

Power Clique Law. A given number of t-sized cliques usually becomes a number of j-sized cliques with $j > t$ following a power law with $\alpha > 0$.

$$\overline{\kappa}_j = {\kappa_t}^{\alpha} \tag{5}$$

where $\overline{\kappa}_j$ is the average of j-sized cliques and κ_t is the number of t-sized cliques with $t < j$.

The clique distribution also follows a power law, called **Clique Distribution Law**, as shown in Figure 13, and is similar to the degree distribution. They have almost the same slope (α) and curve, especially for cliques of the smallest size, like κ_3. An example of this observation is the RTN, which has a bend at degree 10 that appears in its clique distribution. However as t becomes higher, the bend tends to be dissolved in the clique distribution. For Epinions network, which is clearly a social network, one can observe that, as the clique size increases, the number of cliques of a given size a node participates in also increases.



(a) AS (b) Email-Enron (c) Epinions (d) RNT

Fig. 12. Average number of κ_4 vs. κ_3, Average number of κ_5 vs. κ_3 and Average number of κ_5 vs. κ_4 of Epinions and Recommendation Network

Clique Distribution Law. The distribution of cliques κ_t in a graph follows a power law with $\alpha > 0$.

$$P(\kappa_t) = {\kappa_t}^{-\alpha} \tag{6}$$

The number of possible $\kappa_3(v_i)$, $\kappa_4(v_i)$ and $\kappa_5(v_i)$ that a node might have is larger than the real number of κ_3, κ_4 and κ_5 that a node really has. Thus, a real network still has many triangles that do not become large cliques, i.e., although networks are very well connected, they are far from being only one large clique.

8. CONCLUSIONS

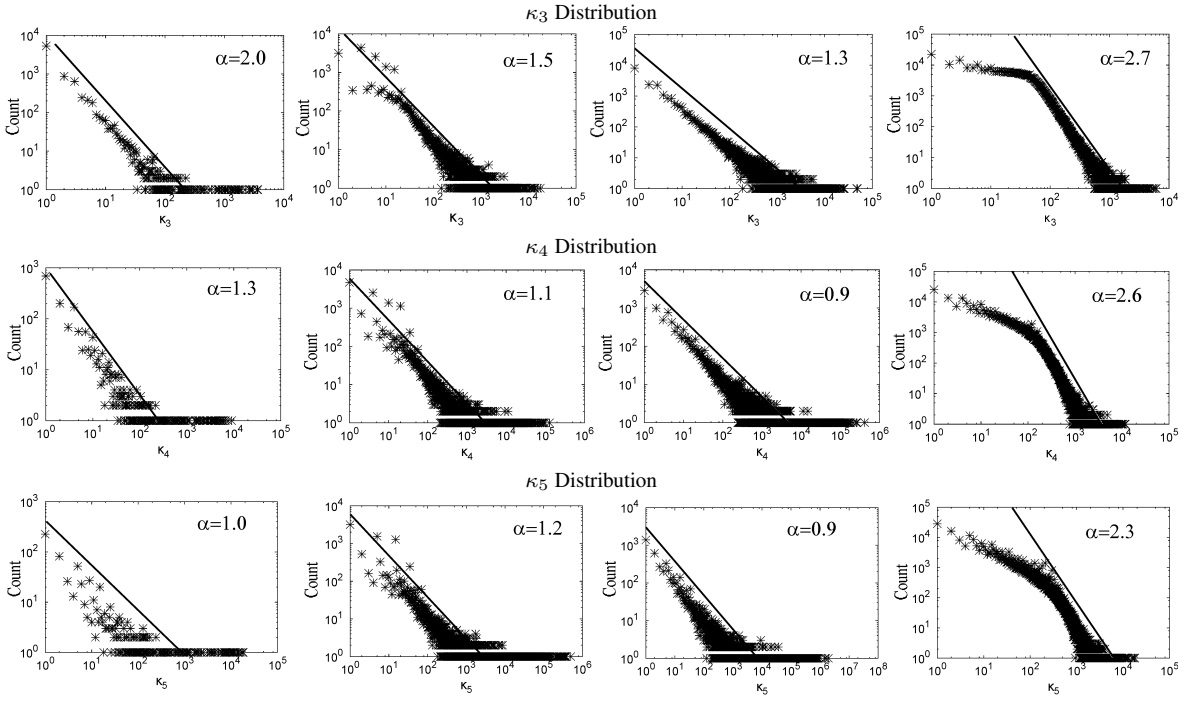
This paper highlighted the importance of studying cliques of a larger size than the triangles (cliques of size three). The main contributions of the paper are the following:

—The design of a cluster coefficient extension, called `Generic Cluster Coefficient`, which allows the identification of nodes that have distinct connectivity strength and can not have their topologies distinguished only by the number of triangles, which is important for community analyses and abnormalities.

—The *Degree Clique Law*, which shows the correlation of a node degree distribution with κ_3, κ_4 and κ_5 distributions. Different types of networks have a different node degree, which deviates from the κ_3, κ_4 and κ_5 distributions.

—The *Power Clique Law*, which is the power law distribution of a clique of a given size vs. cliques of smaller sizes.

—The *Clique Distribution Law*, which shows that clique distribution follows a power law similar to the degree distribution.



(a) AS (b) Email-Enron (c) Epinions (d) RNT

Fig. 13. κ_t distribution with $t = 3, 4, 5$ for (a) Epinions and (b) RTN networks. The clique distribution follows a power law very similar to the degree distribution with the same bend.

—The proposal of a new efficient approach called `FCR - Fast Clique Retrieval`, which allows listing and counting the number of κ_4 and κ_5 in a faster way by using an RDBMS. Queries to count κ_4 and κ_5 are executed, respectively, up to 5 and 12 times faster than the standard approach, by using only an extra table.

As future works, we suggest the analyses of κ_4 and κ_5 extended to other network models and how they evolve over time in real networks.

REFERENCES

The caida as relationships dataset (11/12/2007). http://www.caida.org/data/active/as-relationships/, 2007. dataset.

ALBERT, R., JEONG, H., AND BARABÁSI, A.-L. Error and attack tolerance of complex networks. *Nature* vol. 406, pp. 378–381, 2000.

BECCHETTI, L., BOLDI, P., CASTILLO, C., AND GIONIS, A. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*. Las Vegas, Nevada, USA, pp. 16–24, 2008.

CALDARELLI, G., SATORRAS, P. R., AND VESPIGNANI, A. Structure of cycles and local ordering in complex networks. *The European Physical Journal B - Condensed Matter* 38 (2): 183–186, March, 2004.

CHAKRABARTI, D. AND FALOUTSOS, C. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys* 38 (1): 2, 2006.

CHAKRABARTI, D., WANG, Y., WANG, C., LESKOVEC, J., AND FALOUTSOS, C. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security* 10 (4): 1–26, 2008.

CHAKRAVARTHY, S., BEERA, R., AND BALACHANDRAN, R. Db-subdue: Database approach to graph mining. In *Proceedings of the Pacific-Asia Conference Advances in Knowledge Discovery and Data Mining*, H. Dai, R. Srikant, and C. Zhang (Eds.). Lecture Notes in Computer Science, vol. 3056. Sydney, Australia, pp. 341–350, 2004.

CHAKRAVARTHY, S. AND PRADHAN, S. Db-fsg: An sql-based approach for frequent subgraph mining. In *Proceedings of the International Conference Database and Expert Systems Applications*, S. Bhowmick, J. Kng, and R. Wagner (Eds.). Lecture Notes in Computer Science, vol. 5181. Turin, Italy, pp. 684–692, 2008.

CLAUSET, A., NEWMAN, M. E. J., AND MOORE, C. Finding community structure in very large networks. *Physical Review E* vol. 70, pp. 066111, 2004.

CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. Power-law distributions in empirical data. *SIAM Review* 51 (4): 661–703, 2009.

DOROGOVTSEV, S. N., GOLTSEV, A. V., AND MENDES, J. F. F. Pseudofractal scale-free web. *Physical Review E* vol. 65, pp. 066122, 2002.

DU, N., FALOUTSOS, C., WANG, B., AND AKOGLU, L. Large human communication networks: patterns and a utility-driven generator. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*. Paris, France, pp. 269–278, 2009.

DUNBAR, R. *Grooming, Gossip, and the Evolution of Language*. Harvard University Press, 1998.

FALOUTSOS, M., FALOUTSOS, P., AND FALOUTSOS, C. On power-law relationships of the internet topology. In *Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communication*. Cambridge, Massachusetts, USA, pp. 251–262, 1999.

FRONCZAK, A., HOLYST, J. A., JEDYNAK, M., AND SIENKIEWICZ, J. Higher order clustering coefficients in barabasi-albert networks. *Physica A* 316 (1): 688–694, December, 2002.

HAN, J., CHENG, H., XIN, D., AND YAN, X. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery* 15 (1): 55–86, 2007.

KLIMT, B. AND YANG, Y. Introducing the enron corpus. In *Proceedings of the Conference on Email and Anti-Spam*. Vol. 1. Mountain View, USA, pp. 1–2, 2004.

LATAPY, M. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science* 407 (1-3): 458–473, 2008.

LESKOVEC, J., KLEINBERG, J., AND FALOUTSOS, C. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data* 1 (1): 2, 2007.

LESKOVEC, J., LANG, K. J., DASGUPTA, A., AND MAHONEY, M. W. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR* vol. abs/0810.1355, pp. 1–66, 2008.

LIU, G. AND WONG, L. Effective pruning techniques for mining quasi-cliques. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases*. Antwerp, Belgium, pp. 33–49, 2008.

MIHAIL, M. AND PAPADIMITRIOU, C. H. On the eigenvalue power law. In *Proceedings of the International Workshop on Randomization and Approximation Techniques*. London, UK, pp. 254–262, 2002.

MODANI, N. AND DEY, K. Large maximal cliques enumeration in sparse graphs. In *Proceedings of the ACM conference on Information and knowledge management*. Napa Valley, California, USA, pp. 1377–1378, 2008.

NEWMAN, M. E. J. The structure and function of complex networks. *SIAM Review* vol. 45, pp. 167–256, 2003.

NEWMAN, M. E. J. Power laws, pareto distributions and zipf's law. *Contemporary Physics* vol. 46, pp. 323, 2005.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the International World Wide Web Conference*. Brisbane, Australia, pp. 161–172, 1998.

PAVLO, A., PAULSON, E., RASIN, A., ABADI, D. J., DEWITT, D. J., MADDEN, S., AND STONEBRAKER, M. A comparison of approaches to large-scale data analysis. In *Proceedings of the SIGMOD international conference on Management of data*. Providence, Rhode Island, USA, pp. 165–178, 2009.

RAVASZ, E. AND BARABSI, A.-L. Hierarchical organization in complex networks. *Physical Review E* 67 (2): 026112, Feb, 2003.

RICHARDSON, M., AGRAWAL, R., AND DOMINGOS, P. Trust management for the semantic web. In Proceedings of the International Semantic Web Conference. *Lecture Notes in Computer Science* vol. 2870, pp. 351–368, January, 2003.

RUSTIN, R., editor. *Proceedings of the ACM-SIGMOD Workshop on Data Description, Access and Control, Ann Arbor, Michigan, May 1-3, 1974, 2 Volumes*. ACM, 1974.

STIX, V. Finding all maximal cliques in dynamic graphs. *Computational Optimization and Applications* 27 (2): 173–186, 2004.

TAURO, S. L., PALMER, C., SIGANOS, G., AND FALOUTSOS, M. A simple conceptual model for the internet topology. In *Proceedings of the Global Telecommunications Conference*. Vol. 3. San Antonio, TX, USA, pp. 1667 –1671, 2001.

TSOURAKAKIS, C. E. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *Proceedings of the IEEE International Conference on Data Mining*. Washington, DC, USA, pp. 608–617, 2008.

WANG, W., WANG, C., ZHU, Y., SHI, B., PEI, J., YAN, X., AND HAN, J. Graphminer: a structural pattern-mining system for large disk-based graph databases and its applications. In *Proceedings of the International Conference on Management of Data*. Baltimore, Maryland, USA, pp. 879–881, 2005.

WATTS, D. J. AND STROGATZ, S. H. Collective dynamics of 'small-world' networks. *Nature* 393 (6684): 440–442, June, 1998.

ZENG, Z., WANG, J., ZHOU, L., AND KARYPIS, G. Out-of-core coherent closed quasi-clique mining from large dense graph databases. *ACM Transactions on Database Systems* 32 (2): 13, 2007.