ShatterPlots: Fast Tool for Mining Large Graphs

Ana Paula Appel Deepayan Chakrabarti
Christos Faloutsos Ravi Kumar Jure Leskovec
Andrew Tomkins

December 2008
CMU-ML-08-116

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon

ShatterPlots: Fast Tool for Mining Large Graphs

Ana Paula Appel[1] **Deepayan Chakrabarti**[2]
Christos Faloutsos **Ravi Kumar**[2] **Jure Leskovec**
Andrew Tomkins[2]

December 2008
CMU-ML-08-116

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Graphs appear in several settings, like social networks, recommendation systems, and numerous more. A deep, recurring question is "*How do real graphs look like?*" That is, how can we separate real graphs from synthetic or real graphs with masked portions? The main contribution of this paper is **ShatterPlots**, a simple and powerful algorithm to tease out patterns of real graphs that help us spot fake/masked graphs. The idea is to shatter a graph, by deleting edges, force it to reach a critical ("Shattering") point, and study the properties at that point. One of our most discriminative patterns is the "*NodeShatteringRatio* ": that can almost perfectly separate the real from the synthetic graphs of our extensive collection. Additional contributions of this paper are (a) the careful, scalable design of the algorithm that needs only $O(E)$ time, (b) extensive experiments on a large collection of graphs (19 in total), with up to hundred of thousand of nodes and million edges; and (c) a wealth of observations and patterns, which show how to distinguish synthetic or masked graphs from real ones.

[1]ICMC - USP São Carlos - Brazil
[2]Yahoo! Research

Keywords: algorithms, graph mining, percolation, edges deletion

1 Introduction

Graphs appear in numerous settings social networks, scientific publication network, conferences vs. authors, an so forth. Our goal is to find patterns to help us spot fake and "masked" graphs. (By "*masked*" we mean a graph that is a non-random sample of a real graph - for example, a real graph after one deletes all the nodes with degree ≤ 100). We propose to tease-out the characteristics of a large graphs with the novel tool of ShatterPlots. Moreover, we want our method to be scalable, so that to handle graphs that span MegaBytes, GigaBytes or more.

The main idea behind ShatterPlots reminds of high-energy physics, where particles are smashed, and experts study the results of the collisions to reach conclusions. Here, we propose to shatter the given graph, that is, to drive it to the "**Shattering point**", by deleting edges at random, and observing its behavior. The first research challenge is how to interpret the results of the Shattering. The second challenge is scalability and speed.

The answers to the above challenges are exactly the contributions of this work. For the first, we show that random edge deletion always leads to a high spike of the diameter, exactly at the critical point that we call "Shattering point".

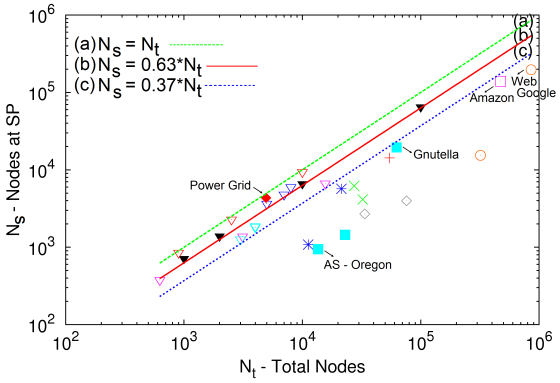


Figure 1: Our *NodeShatteringRatio* pattern allows an impressive distinction between fake/masked graphs (triangles, Amazon, Web Google) from real graphs (rest).

At the Shattering point, we give a list of surprising observations for several real graphs. The most surprising is the "30 per cent" pattern, which states that under random edge deletion, real graphs have 30% more nodes than edges when they reach their Shattering point, *regardless* of which the original graph was. Another interesting observation is that at the Shattering point the count of remaining edges is $1/\lambda_1$ of the original edge count, where λ_1 is the first eigenvalue of original graph; it is fascinating that $1/\lambda_1$ is the epidemic threshold of the graph [12].

The most striking pattern is the *NodeShatteringRatio* one, illustrated in Figure 1. This pattern allows us to perfectly separate the real graphs from fake/masked ones, at least for the graphs of our collection. Specifically, the fraction N_s/N_t of remaining nodes at the Shattering point, is much-much lower for most real graphs, while it is about 0.7 for the masked ones (and for the Erdős-Rényi

graphs). (N_s is the number of nodes at the Shattering point and N_t is the total number of nodes of original graph.)

Finally, for scalability, we propose a fast, adaptive algorithm that can quickly discover the Shattering point. Its performance is linear on the number of edges E, as shown empirically.

The rest of the paper is organized as follows. Section 2 surveys the related techniques. Section 3 proposes the data model and the formal problem specification. Section 3 further presents the algorithms. In Section 4, we evaluate the algorithms with real data. Section 5 and Section 6 we present patterns found, proofs and outliers spotted. The scalability is presented in Section 7. We conclude in section 8.

2 Related Work

There is a significant body on research related to our problem, which we categorize into the following groups: graph algorithms; graph patterns; epidemiology; phase transitions; and outliers detection.

Graph Algorithms: Intuitively we expect the graph to shatter at the point where natural communities or clusters break apart. Popular methods for partitioning graphs include the METIS algorithm [24], spectral partitioning techniques [23], flow-based methods [20] information-theoretic methods [14], and methods based on the "betweeness" of edges [22], among others. Note that our work is orthogonal to this, as we are using fast and scalable techniques to examine the structure of the graph. Probably the most related is the k-cores [8] decomposition that recursively "peels" the graph; a recent extension for bipartite graphs uses the KNC plots [29]. This approach would be complementary to ours, since they examine different aspects of the graph.

Graph patterns: Several old and recent patterns have been discovered for large, real graphs:

The first is the *skewed degree distribution* phenomenon, with power law tails, for the Internet [19], for the Web [26, 9], for citation graphs [40], for online social networksand many others. Deviations from the power-law pattern have also been noticed [38], but the distribution is still very skewed.

The second is the *Small diameter:* This is the the "small- world" phenomenon, or 'six degrees of separation' [47] The diameter of a graph is d if every pair of nodes can be connected by a path of length at most d. Following the computer network literature, we use the *effective diameter* [45]: The minimum number of hops in which some fraction (or quantile q, typically $q = 90\%$) of all connected pairs of nodes can reach each other. The effective diameter has been found to be small and decreasing over time for large real-world graphs, like Internet, Web, and social networks [4, 35, 32].

Phase transitions: The point where the graph shatters is ultimately a point of phase transition, i.e., a point where the connectivity structure abruptly changes. The Erdős-Rényi graphs exhibit phase transitions [17] in the size of the largest connected component. Several researchers argue that real systems are "at critical points" [6, 44], like avalanches, forests (with forest fires), mechanical tension causing earthquakes, and so on. If this also holds for real networks, then they should be

ready to "shatter", after a lot of edges insertion. Phase transition is also known as bond and site percolation threshold. An example of application is presented in [28].

Epidemiology: Most of the previous research on the flow of information and influence through the networks has been done in the context of epidemiology and the spread of diseases over the network [5, 12].

The work on spread of diseases in networks and immunization mostly focuses on determining the value of the *epidemic threshold* [5], a critical value of the virus transmission probability above which the virus creates an epidemic.

The epidemiology community has developed the so-called *SIR* and *SIS* models [5] of infection. The *SIS* model (*Susceptible – Infective – Susceptible*) is suitable for the common flu, where nodes may be infected, healed (and susceptible), and infected again.

Recent work showed that the epidemic threshold of a graph is $1/\lambda_1$, that is, the inverse of its strongest eigenvalue [12]. We give more details later, as well as its connection to the bond percolation threshold.

Outliers detection on graphs: Last, we focus on outlier detection, as the connectivity structure revealed by the ShatterPlots . Autopart [11] finds outlier edges in a general graph; however, we need to detect outliers *nodes*. Noble and Cook [36] study anomaly detection on general graph with labeled nodes; however, their goal is to identify abnormal substructure in the graph, not the abnormal *nodes*. Aggarwal and Yu [2] propose algorithms to find outliers in high-dimensional spaces, but its applicability to graphs is unclear: the nodes in a graph lie in a vector space formed by the graph nodes themselves, so the vector space and the points in it are related. As we will see later we observe very different patterns of shattered graphs when compared to simple models, which allows us to detect masked/fake non-realistic graphs.

3 Proposed Method

We start with the problem definition and the motivating questions. Then we describe our design decisions, and finally we give our algorithm.

3.1 Problem Definition.

Our goal is to find patterns at the Shattering point, that is a clear spike in the diameter after some edges deletion in real graphs like social networks, citation and web graphs, recommendation systems (users-to-products bipartite networks). We also want to analyze if fake/masked graphs have a different behavior than real graphs at the Shattering point. What can we say about real graphs at the Shattering Point? Can we find interesting patterns in real graph at this point? Can we use these patterns to spot fake/masked graphs?

The problem is defined as follows:

Problem 1 Given *a large, sparse graph check whether it is an masked or synthetic graph.*

In fact, we have two types of questions that we would like to check for all graphs. The first are "philosophical" questions, whose answers will settle some conjectures. The second set consists of "exploratory" questions. These questions refer to what properties we should expect to see, at the Shattering point of a graph (assuming that it does have a Shattering point).

3.1.1 "Philosophical" Questions

PhQ 1 *Do real graphs have a Shattering point?*

Real networks are very resilient [3] at random node deletions while some others, like Erdős-Rényi are not. One would expect so, if we have random edge deletion (*RED*). But are there exceptions in real graphs? Is it possible to have a real graph, which, under *RED*, the diameter increasing continuously, without an abrupt shattering?

PhQ 2 *Are real-life graphs just a bit above the Shattering point?*

One would expect so: For example, Bak [6] proposes the theory of SOC (Self-Organized Critically), arguing that several phenomena, are just at their critical point, like avalanches, finances of interrelated companies, tectonic plaques. Several graph generators also focus on 'optimized tolerance' [10, 18]. Thus one might expect that real graphs are connected, but barely so, and thus would be just above Shattering. A communication network that is way above Shattering point, would be wasting resources, one might argue.

3.1.2 Exploratory Questions.

Jumping ahead, it turns out that all the real and synthetic graphs we tried, do have a sharp Shattering point. This brings a whole wave of questions:

ExQ 1 *What is the Edge shattering ratio E_s/E_t (*i.e.*, the fraction of edges at the Shattering point)? How does it depend on the graph size, if at all?*

Where E_s is the number of edges and N_s is the number of nodes, both at Shattering point. E_t is the total number of edges of original graph and N_t is the total number of nodes in the original graph. The symbols are defined in table 1.

ExQ 2 *What about the Node shattering ratio N_s/N_t (*i.e.*, the fraction of nodes at the Shattering point)?*

ExQ 3 *Do synthetic graphs have the same behavior at the Shattering point? or do they follow different laws?*

ExQ 4 *What can we say about the node-to-edge ratio of a graph at the Shattering point? And about the giant connected component at the Shattering point?*

3.2 Design decisions.

*T*hinning methods: We tried several thinning methods, like Random Edge Deletion(*RED*), and several versions of "Hostile" edge deletion. The most striking patterns were with the former, and thus we shall exclusively focus on *RED* here.

*C*hoice of shattering criterion: The shattering criterion should ideally have a sharp transition. We considered several shattering criteria:

- Size (number of nodes) of the largest weakly connected component
- The effective diameter (number of hops at which 90% of all reachable pairs do reach each other)
- Total number of reachable pairs of nodes

We expected that the graph will shatter at all of the above criteria, *i.e.*, there will be a Shattering point in the edge deletion process, where the connectivity of the graph will be seriously disrupted: e.g., the graph gets disconnected, the size of the largest component drops, the diameter spikes, and the number of reachable pairs of nodes drops. We examine the results of shattering of our 19 network datasets in more detail in the following section.

3.3 Algorithm description

Next, we present the algorithm for creating ShatterPlots. However, instead of the algorithm starts with the full graph and delete edges at random, it starts with an empty graph and insert edges at random. Algorithm 3.3 shows the details.

The idea is to shuffle the edges file of a graph G and builds the temporary graph H adding some number of edges ($Step(t)$) at random. Both of them have the same nodes (N), and will be exactly the same in the end of the algorithm. After each insertion we measure the structural properties of the graph, like, e.g., the diameter, the number of reachable pairs of nodes, number of triangles, the first eigenvalue of the graph adjacency matrix or the size of the largest connected component. We keep repeating the process until the graph is full, *i.e.*, contains all edges from the edges file.

Ideally we would re-compute graph properties after insert of each and every edge. However, that approach would be slow, and thus we insert a batches of edges at a time. The question is what is the appropriate size of such batch, so that we will not overshoot and miss the Shattering point? Our answer is an adaptive method: we start with a small batch size, and if there is no major difference in the graph structure (say, the diameter), then, we increase the batch size. Conversely, we decrease it, if we seem to be reaching a spike. The same process could be applied in the other way around, that is, instead of inserting edges, we could start with a full graph and delete edges at random on it. Empirically, we notice that the algorithm is very fast, and it usually needs about 250 steps to locate the Shattering point.

Scalability: next, we show that Adaptive ShatterPlot Algorithm scales well on the number of total edges E_t to. This shows that the Adaptive ShatterPlot is capable of handle large graphs. The

Algorithm 1 ShatterPlot algorithm

Adaptive ShatterPlot

Input: The input graph $G(N, E)$

Output: Point of shattering (and stats about it)

```
Shuffle the |E|
Temporary H(N,Ø), on N nodes
ε = 0.005 or ε = 1/λ1
t = 0
Step(t) = ε * |E|
while H ! = G do
    Insert Step(t) edges in H at random
    t = t + 1
    Measure structural properties of H (diameter, connected components, first eigenvalue, etc.)
    Dt = effective diameter of H
    if t > 1 then
        if Dt − Dt−1 ≥ 1 then
            Step(t) = Step(t − 1)/2
        else if Dt − Dt−1 ≤ −1 then
            Step(t) = 2 * Step(t − 1)
        end if
    else if ε = 1/λ1 then
        Step(t) = 0.005 * |E|
    end if
end while
```

algorithm scales even better, up to 8 times faster, using the *Eigenvalue* pattern presented in Section 6.

First, we assume edge insertion is a constant time operation. This is true for most of graph implementations. In some implementations it can be logarithmic/linear in the average degree of the graph but as real graphs are sparse this is practically constant. Second, thanks to the Approximate Neighborhood Function algorithm [37] (ANF), we can calculate the effective diameter of the graph in time linear $O(E)$ on the number of edges E in the graph.

Definition 1 *The effective diameter is the minimum number of hops in which 90% of all connected pairs of nodes can reach each other.*

Also, the effective diameter is a more robust measure of the pairwise distances between nodes of a graph.

However, this does not solve the problem immediately: if we use a naive implementation of the ShatterPlot algorithm and at every step add a constant number of edges, then the full algorithm would scale quadratically with the number of edges $O(E^2)$ ($O(E)$ for the number of ShatterPlot iterations, and a factor of $O(E)$ for running ANF at each step). Due to the adaptive nature of our algorithm that exponentially adjusts the number of edges it adds from the graph, we only need a roughly *constant* number of iterations, which makes our algorithm scales well to the number of edges.

We have two version of our algorithm. The first is called *Proportional ShatterPlots* in which in $Step(0)$ the initial value ϵ is 0.005. The other version is called *Eigenvalue ShatterPlots*, given that we use $1/\lambda_1$ as initial value for ϵ at $Step(0)$. For *Eigenvalue ShatterPlots* none of our extensive collection of graphs had the Shattering point missed. As we can see in the *Eigenvalue* pattern, presented in Section 6, all of ours graphs are above the line, that is, the E_s is higher than the $1/\lambda_1 * E_t$ at Shattering point. So we can overshooting the initial value of ϵ to $1/\lambda_1$. In case of *Eigenvalue ShatterPlots* miss the Shattering point, an easily solution is to backtrack the algorithm and apply the *Proportional ShatterPlots* between 0 and previous value of A_0. Later we present wallclock times, illustrating the scalability of our method and the improvements reached with *Eigenvalue ShatterPlots*.

4 Experiments

Here we give the answers to our posed questions, our observations and the results achieved.

4.1 Datasets.

Table 1 present the symbols used in this section. We define the *Shattering point* as the point where the shattering of the graph happens. Based on this definition we will present the results of other measures, such as nodes and edges of giant component, total number or reachable pairs of nodes, number of nodes, number of edges, diameter, highest degree, triangles and first eigenvalue in this point, named respectively N_{sgcc},E_{sgcc}, N_{Npairs}, N_s, E_s, D_s, d_s, Δ_S, and $\lambda_{1,s}$.

Symbols	Definitions
SP	Shattering point (= critical point)
REI	Random Edge insertion
ct	constant value
N_t	Total number of nodes in the graph
E_t	Total number of edges in the graph
Δ_t	Total number of Triangles in the graph
λ_1	Largest eigenvalue of original graph
N_s	Number of nodes at *SP*of degree ≥ 1
E_s	Number of edges at *SP*
d_s	Highest degree at *SP*
N_{sgcc}	Nodes in largest weakly conn. comp. at *SP*
E_{sgcc}	Edges in largest weakly conn. comp. at *SP*
$\lambda_{1,s}$	Largest eigenvalue at *SP*
Δ_s	Total number of Triangles at *SP*
D_s	Effective diameter at *SP*

Table 1: Symbols, acronyms and definitions

Table 2 presents all datasets used and the symbols that represent each of them in the plots shown in the following sections. The synthetic datasets were generating using the algorithm described in their respectively papers. For RB we used the model described [39] with 3, 4 and 7 levels for each of the three graphs. For Erdős-Rényi we use the model in [15], but instead of $G(n, p)$, where p is the probability to attach an edge, and n is the number of nodes, we used $G(n, m)$ as model, where m is the total number of edge in a graph. The number of nodes and edges used are $n = 1k, 2k, 10k, 100k$ and $m = 5k, 14k, 50k, 400k$ respectively.

The Preferential Attachment graphs (PA) were created using the model described in [7] using 3k and 4k nodes and 3 and 5 as parameter for degree. In Small Word graphs (SW), the generator follows the model presented in [47] using as parameters number of nodes (n), degree (d) and Rewire Probability (p). So, for our graphs we used: $n = 5k, 8k, 8k$, $d = 5, 6, 3$, $p = 0.4, 0.9, 0.5$ respectively. For 2D grids, we created 3 of them being 30x30, 50x50 and 1000x1000 without wrap up. All of the graphs were generated as undirected.

4.2 Choice of shattering criterion.

Here we show that, among the several measures we can use to detect critical point/Shattering point, the best is the effective diameter D. The reason is that giant component and number of reachable pairs do show critical point (that is, a sudden increase, as we insert more and more edges), but it is not clear how to define the exact Shattering point. In contrast, the diameter always has a sharp peak, reminding us of the percolation threshold [43]. Indeed the diameter is widely use to evaluate the network breakdown during the random node deletion or highest degree node deletion [3]. Figure

	Nodes	Edges	Description
Online social networks			
◇	75,877	405,739	Epinions network [41]
◇	33,696	180,811	Enron email net [27]
Academic collaboration (co-authorship) networks			
*	21,363	91,286	Arxiv `cond-mat` [33]
*	11,204	117,619	Arxiv `hep-ph` [33]
Information (citation) networks			
x	34,401	420,784	Arxiv `hep-th` citations [21]
x	32,384	315,713	Blog citation (1 year) [30]
Web graphs			
⊙	319,717	1,542,940	Stanford – UC Berkeley
⊙	855,802	4,291,352	Google web graph [1]
Amazon Product co-purchasing networks			
⊡	473,315	3,505,519	Snapshot 2 [13]
Bipartite (authors-to-papers) networks			
+	54,498	131,123	Arxiv `astro-ph` [30]
Internet networks			
■	13,579	37,448	AS Oregon [31]
■	22,963	48,436	AS graph from M. Newman
■	62,561	147,878	Gnutella, 31 Mar 2000 [42]
Grid networks			
♦	4,941	6,594	Power Grid western US [47]
Synthetic networks			
▽	2D - Synthetic Grid		
▼	Erdős-Rényi random graphs [17]		
▽	BR - Barabasi Hierarchical Model [39]		
▽	SW - SmallWorld [47]		
▽	PA - Preferential Attachment [7]		

Table 2: Datasets that we consider in our study. They symbol at the beginning of the row is later used in figures to denote the datasets.

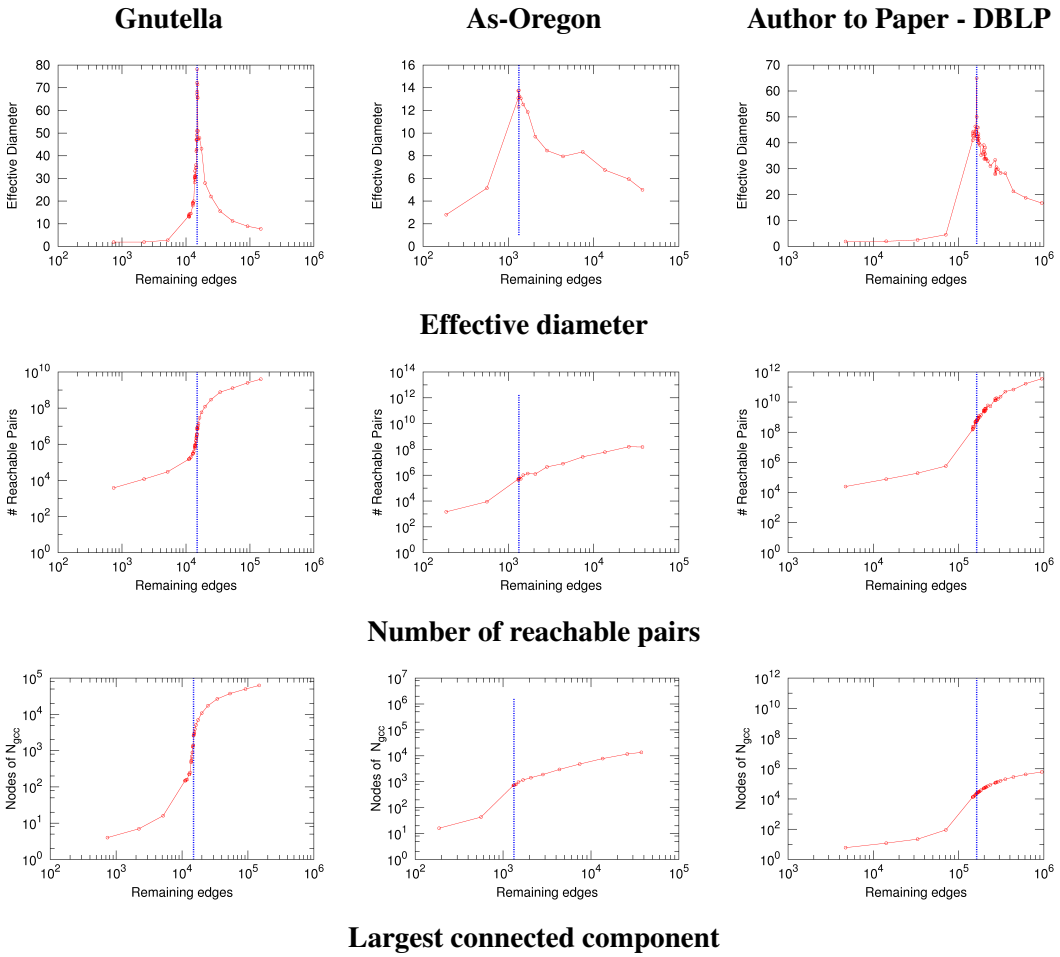


Figure 2: We randomly delete edges and measure graph structural properties. Graphs shatter in all measure but only diameter has a nice and clear spike.

2 shows Gnutella, AS-Oregon and Author to Paper datasets - we omit the rest for brevity, because they all have similar behavior.

Rows correspond to measures (diameter D_s, N_{sgcc} and N_{Npairs}). Each plot shows the measure of interest (diameter, etc), versus the number of retained edges, under random edge insertion (*REI*).

The vertical lines correspond to the spike of the diameter (Top row plot) As we can see in Figure 2, we can use ShatterPlots to find the critical point because it is the only one with a sharp, clear spike. The main point is that the Shattering point E_s at which the diameter spikes always falls in the region where the other measures have a sudden drop. Now, we define:

Definition 2 *The* Shattering point E_s *of a graph is the number of retained edges for which the (effective) diameter spikes.*

The ShatterPlot is exactly the plot of diameter D_s versus retained edges E_s, and we shall not use the rest of the measures any more.

Another important definition is:

Definition 3 *For Erdős-Rényi graphs, the Shattering point as defined above, coincides with the phase transition point.*

This is important, because we have several results from the theory of random graphs. These results are used as sanity checks to ours findings.

5 Results - Philosophical Questions

To answer philosophical and exploratory questions posed in section 3.1 we "shatter" many graphs (Table 2 presents the datasets used) and build some plots with the measures collected at Shattering point - N_s, E_s, d_s and $\lambda_{1,s}$ of all our real graphs. We also made these for synthetic, Erdős-Rényi graphs, 2D-grid graphs, Hierarchical graphs, Small Word, and Preferential Attachment for sanity check and comparisons.

After "Shatter" real and synthetic graphs, we could answered both philosophical questions PhQ 1 and PhQ 2 with the following patterns:

Pattern 1 *All measures have a Shattering point at about the same point for a given graph, but only the diameter has a clear spike.*

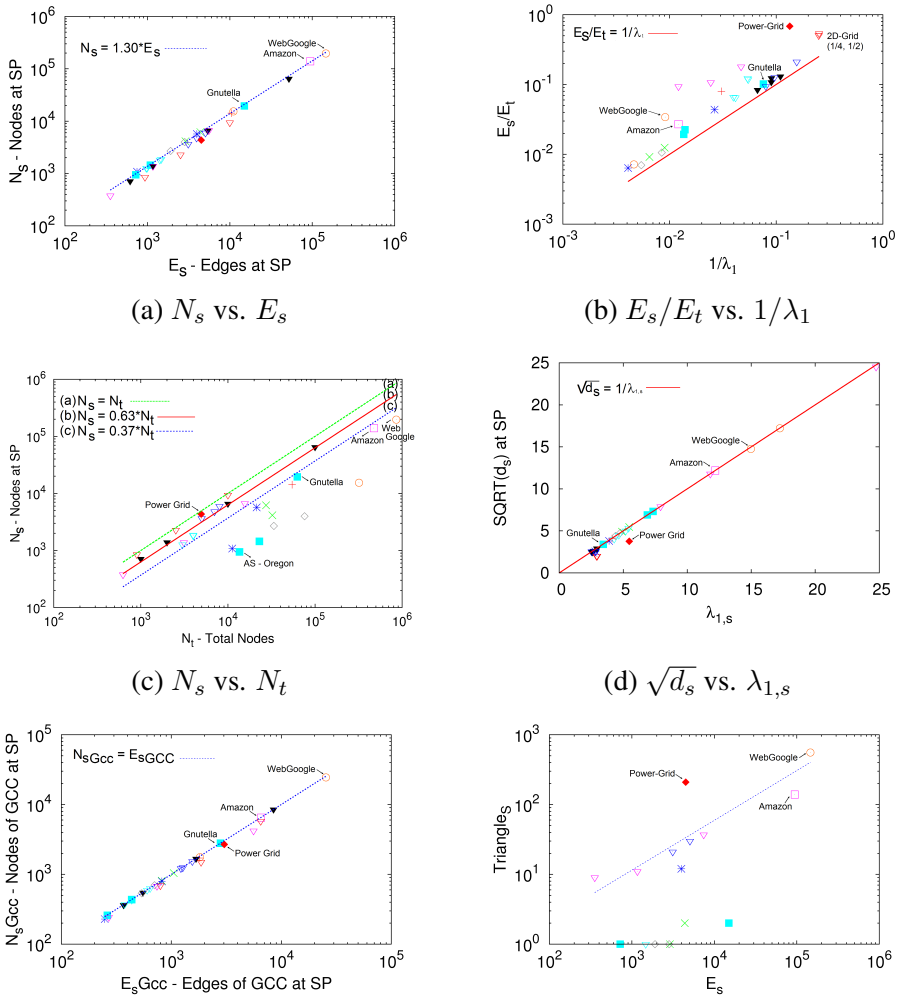
Pattern 2 *All graphs have a Shattering point, under* REI.

Figure 3 shows the plots of structural measures at Shattering point. The axis scaling is linear - linear to (d), and log - log to (a), (b), (c), (e) and (f) and we also show the theoretical/expected fitting curve (all of them with coefficient above 0.98), when there seem to be a strong correlation. Moreover, we show fitting lines, a blue one for results we gotten, and a red one for theoretical or expected ones. All experiments are average of 10 runs. The results for the Erdős-Rényi graphs are shown with dark down triangles, and synthetic with down triangles for better viewing in black-and-white. However, the paper is better viewing in color.

6 Results - Exploratory Questions

As we can see (Figure 3 (a)) all graphs have a Shattering point. The nodes-edges ratio at Shattering point N_s/E_s of all graphs follow a line which has the slope 1.30. This means that at the Shattering point the number of nodes N_s is about 30% higher than E_s. This observation also answered Question PhQ2 and part of ExQ3 and ExQ4.

It turns out that our *REI* procedure, when applied to Erdős-Rényi graphs, leads to a Shattering point which is exactly the one predicted by theory. In all our Erdős-Rényi graphs, the Shattering value E_s satisfied $E_s = N_t/2$ and $N_s = N_t * (1 - 1/e)$, which is exactly the condition for phase transition [15].



(a) N_s vs. E_s

(b) E_s/E_t vs. $1/\lambda_1$

(c) N_s vs. N_t

(d) $\sqrt{d_s}$ vs. $\lambda_{1,s}$

(e) N_{sgcc} vs. E_{sgcc}

(f) Δ_s vs. E_s

Figure 3: Structural observations at Shattering point (*SP*), where the graph shatters. Synthetic graphs in triangles; Erdős-Rényi ones in black triangles. (a) number of non-isolated nodes (N_s) versus number of edges E_s at Shattering point (*30-per-cent* pattern); (b) number of retained edges E_s over total number of edges E_t versus one over λ_1, first eigenvalue of original graph (*Eigenvalue* pattern); Amazon is deviate from the line, also synthetic graphs RB, PA and 2D-grid. (c) number of survivors nodes N_s versus total number of nodes in the original graph N_t (*NodeShatteringRatio* pattern) 2D-grid are above the Erdős-Rényi line; SW are together with Erdős-Rényi , RB and PA are above line 'c' and below line 'b' (d)square root of highest degree at Shattering point d_s versus $\lambda_{1,s}$ at Shattering point (*Root-degree* pattern); (e) number of nodes N_{sgcc} versus number of edges E_{sgcc} in giant component at Shattering point (*TreeGCC* pattern); (f) number of Triangles Δ_s versus number of edges E_s (*TriangleRatio* pattern). We can see that Power Grid has disproportionate number of triangles. Only the graphs with one or more triangles appear.

6.1 *30-per-cent* pattern.

Pattern 3 (*30-per-cent*) *All real graphs shatter when N_s is about 30% higher than E_s.*

Theoretical Justification For Erdős-Rényi graphs, the *30-per-cent* pattern can be proved: For Erdős-Rényi graphs at phase transition (= Shattering point), we have

$$E_s = 1/2 * N_s * e/(e-1) = 0.79 * N_s \tag{1}$$

where $e = 2.718$. Identically, $N_s = 1.26E_s$, very close to 30%.

Proof: At Shattering point, we have $N_s = N_t * (1 - 1/e)$ and $E_s = N_t/2$. Substituting N_t, completes the proof. **QED**

Discussion: It is surprising that the rest of the graphs also obey this pattern, reasonably close. It is even more surprising, because, as we see later, at the Shattering point, real graphs clearly differ from Erdős-Rényi graphs, when we consider other aspects than the ratio E_s/N_s (Question ExQ3).

Outliers: This is one of the few patterns we discovered that seems universal, and thus can not help us spot outliers and masked/synthetic graphs. Several of our upcoming patterns do, though.

6.2 *Eigenvalue* pattern.

Let E_s/E_t be defined as the *Edge Shattering Ratio*, that is the fraction of edges that we need to retain, to be at Shattering point. Figure 3 (b) shows the percentage of edges remaining in the graph at Shattering point has a correlation with $1/\lambda$. This observation answered Question ExQ 1. Indeed, this pattern shows that the Edge Shattering Ratio does not depend on the size of the graph but the highest eigenvalue. Then we have:

Pattern 4 (*Eigenvalue*) *The edges ratio*

$$E_s/E_t = ct * 1/\lambda_1. \tag{2}$$

Theoretical Justification: The Edge Shattering Ratio is the percentage of edges that still create a giant connected component. λ_1 is the epidemic threshold for an SIS model (Susceptible-Infected-Susceptible), like the flu virus: The epidemic threshold in an SIS model is $\beta/\delta = 1/\lambda_1$, where β is the virus birth rate and δ the virus death rate and λ_1 is the highest eigenvalue of the original graph. See [12] **QED**

Discussion: β/δ is the number of attacks per edge that a virus-molecule can do, until the host recovers. Thus, during the lifetime of a virus-molecule, it sees $\delta \cong E_t$ edges available to it. At epidemic threshold, this edge count should be $\beta \cong E_s$. The ratio E_s/E_t is also known *Bond Percolation Threshold*. For 2D-grids the Bond Percolation Threshold is well defined as 0.5 [25].

Outliers: In this pattern we can see that some graphs like Preferential Attachment (PA), Hierarchical (RB) and 2D-grids do not follow it.

6.3 *NodeShatteringRatio* pattern.

Figure 3 (c) shows the Node Shattering Ration, that is the relation of nodes at the Shattering point N_s versus number of nodes N_t of entire graph. We fit three lines in Figure 3 (c). The line (a) - dotted line - is exactly $N_t = N_s$ that is the maximum bound, the line (b) - solid line - is the theoretical line of Erdős-Rényi and the line (c) - dashed line - is $N_s = 0.37 * N_t$ of which all real graphs are below it. As we can see, this pattern answered Questions ExQ 2 and ExQ 3.

Pattern 5 *Synthetic graphs are close to* $N_s = 0.63 * N_t$.

Theoretical Justification: As shown [15], for all Erdős-Rényi in the phase transition we have

$$N_s = N_t * (1 - 1/e) \tag{3}$$

and $(1 - 1/e) = 0.63$, where $e = 2.718$.

Discussion: The explanation is that most of real graphs have many nodes with degree $d = 1$, that is a heavy tail power law distribution and these nodes have high probability to be isolated at Shattering point. An example of this is the dataset AS Oregon in which the degree distribution is presented in 4 (c). On the other hand, graphs like 2D-grids have most of the nodes with degree four, and Erdős-Rényi graphs have a little variation, with most nodes having degree close to the average degree. All such graphs have very few isolated nodes when they shatter, with 2D-grids even fewer than Erdős-Rényi graphs. This is the reason that the orange triangles (2D-grids) are above the line of the black triangles (Erdős-Rényi graphs) have at Shattering point many more nodes than real ones. However some graphs, like Amazon and Gnutella are masked, this mean that they don't have a nice power law distribution, as shown in Figure 4 (a) and (b) respectively. For these graphs, we can seen in 3 (c) that they shatter faster than the other real graphs, like AS Oregon.

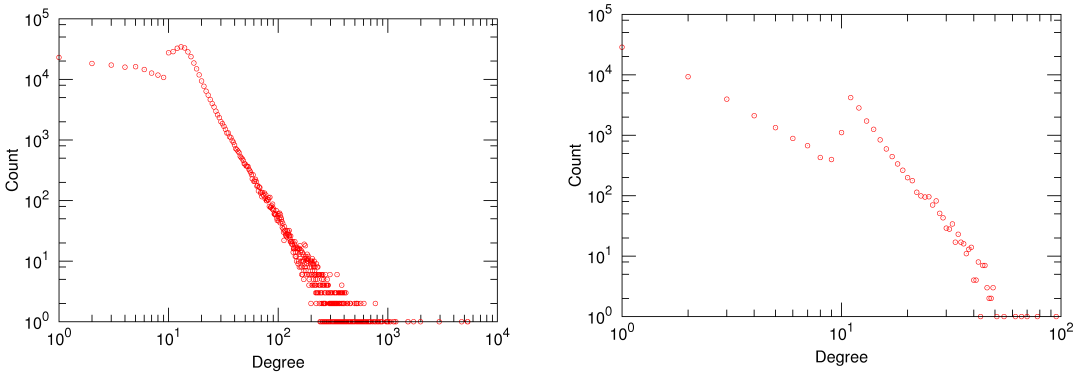
Outliers: The *NodeShatteringRatio* pattern is probably the best detector of synthetic and masked graphs, at least for the mix of graphs we studied. Notice that all synthetic graphs are close to the line 'b' and above the line 'c' - $N_s = 0.37 * N_t$ - in Figure 3(c).

6.4 *Root-degree* pattern.

Figure 3 (d) plots the highest eigenvalue at Shattering point $\lambda_{1,s}$, versus d_s, the square root of the highest degree in the graph at shattering point. The Figure also shows the line with equation $\lambda_{1,s} = \sqrt{d_s}$.

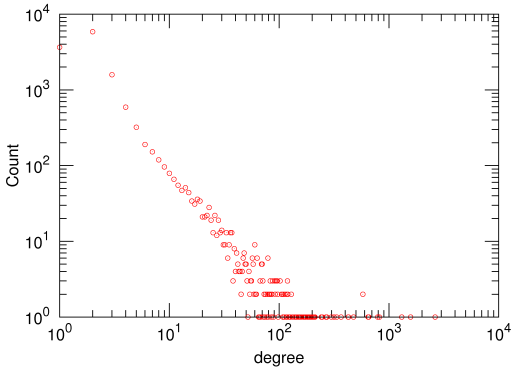
Pattern 6 *All graphs obey* $\lambda_{1,s} \geq \sqrt{d_s}$.

There are some recent theorems that help us justify this behavior:

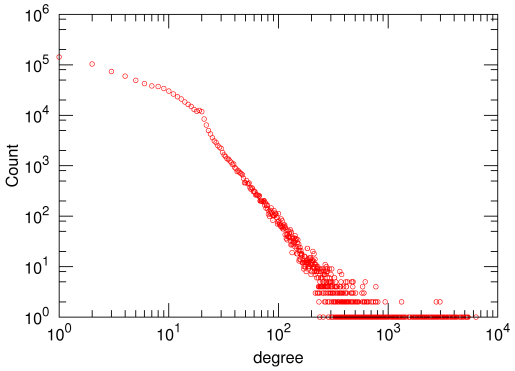


(a) Degree Amazon

(b) Degree Gnutella



(c) Degree AS Oregon



(d) Degree Web Google

Figure 4: Degree Distribution of initial Graphs: (a) Amazon, (b) Gnutella, (c) AS Oregon and (d) Web Google graphs

Theoretical Justification: As shown [34] for all graphs we have $\sqrt{d_i}(1-o(1)) \leq \lambda_i \leq \sqrt{d_i}(1+o(1)), i = 1, 2, ..., k$

where λ_i is the i-th eigenvalue and d_i is its respectively degree.

Theoretical Justification: As shown [16] for all Erdős-Rényi graphs we have $\lambda = [1 + o(1)] * \max(N * p, \sqrt{degree_{max}})$

where λ is the highest eigenvalue, N is the number of nodes of a graph, p is the probability of a node be connected and $degree_{max}$ is the maximum degree of the graph.

Discussion: The theory presented above hold for any graph, including the ones at shattering point. At shattering point, we have that $N * p$=1 for Erdős-Rényi graphs, given that the maximum degree will be > 1, we see why the pattern holds for Erdős-Rényi graphs.

Specifically for Erdős-Rényi graphs (black triangles), we see that their eigenvalue $\lambda_{1,s}$ is roughly constant, independent of the number of nodes N_t that the graph started with.

Pattern 7 *The $\lambda_{1,s}$ for Erdős-Rényi graphs seems to be constant:* ≈ 2.8

Power-Grid graph is below the line. This mean that it is well connected at the Shattering point. Figure 5 shows the highest degree node of Power-Grid in the original graph (Figure 5 (a)) and at Shattering point (Figure 5 (b)). We can see that the highest degree node still has some triangles and many connection even at Shattering point. We can also verify this looking at the *NodeShatteringRatio* pattern. As we can see, the Power Grid is very close to the line 'a'. That is the N_s is very close to N_t.

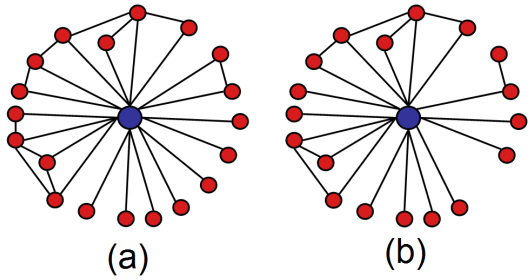


Figure 5: Highest degree node of Power-Grid: (a) Original Graph and (b) At Shattering Point

6.5 *TreeGCC* pattern.

Figure 3 (e) shows that all graph s at the Shattering point have the same amount of edges E_{sgcc} and nodes N_{sgcc} in the Giant Connected Component. As we can see, this pattern answered the second part of Question ExQ 4.

Pattern 8 *All giant connected component of all graphs at Shattering Point have $E_{sgcc} \cong N_{sgcc}$.*

Discussion: We know that above the Shattering point the graph is very connected and below it the graph is completely disconnected. So, at Critical/Shattering point we expected that the graph is barely connected. This means that a small amount of edges removed make the graph totally disconnected. Looking to this pattern we can see that Giant Connected Component at Shattering Point looks like a tree. Notice that some graphs are plotted slightly below the line (apparently, being 'fatter' than a tree), for example Power Grid. Also notice the subtle difference between this pattern and the *30-per-cent*: here we ignore the (several) nodes and edges that are outside the giant connected component, while in the *30-per-cent* pattern we include them.

6.6 *TriangleRatio* pattern.

Figure 3 (f) shows that, at the Shattering point, most of the graphs have very few triangles. In fact, we don't even plot the graphs with zero triangles, because of the logarithmic axis.

Pattern 9 *Graphs at Shattering point have few or no triangles ($\Delta_s \approx 0$).*

Outliers: The Power Grid graph stands out.

Discussion: We expected that graphs at Shattering point are barely connected. We can see this in *TreeGCC* pattern where the giant connected component seems to be a tree and in *Root-degree* pattern where we see that the $\lambda_{1,s}$ is strongly related to the highest degree at Shattering point. We also know that the number of triangles (Δ) a node participates in, increases with the degree of that node [46]. However some graphs, like Power Grid, have a lot of triangles at Shattering point. Why is the Power Grid exhibiting such a different behavior? We give some explanations next:

For Power Grid, we observe that it falls below the line in Figure 3 (d), which mean that it has more edges than nodes in the giant component, that is, the graph is "fatter" than a tree. Another fact is that it has the $\lambda_{1,s}$ is higher than the $\sqrt{d_s}$ as shown in Figure 3 (d). This mean that the eigenvalue is not correlated to the highest degree node, given that the highest degree node is better connected than a star as shown in Figure 5 (b).

Another fact is that the relation between the initial number of triangles (Δ_t) of Power Grid is much higher than the other graphs. For example, initially, Power Grid has $\Delta_t = 651$ while Web Google has $\Delta_t = 13,356,298$; at the Shattering point, Power Grid has $\Delta_s = 209$ while Web Google has $\Delta_t = 556$.

7 Scalability

The ShatterPlots is a fast tool that just needs to read the edge file once at every iteration The number of iterations depends on how quickly we can zoom to the shattering point E_s.

Figure 6 shows the scalability of *Proportional ShatterPlots* and *Eigenvalue ShatterPlots*, plotting the wall-clock time versus the dataset size. The input graphs are synthetic Erdős-Rényi graphs, where we controlled the number of initial edges E = 14k, 40k, 50k, 200k, 300k, 500k, 600k and the number of nodes was N=2k, 10K, 10k, 40K, 60k, 80k, 100k, respectively. The experiments ran on a Quad Xeon (2.66 GHz), with 8Gb of RAM, under Linux (Ubuntu).

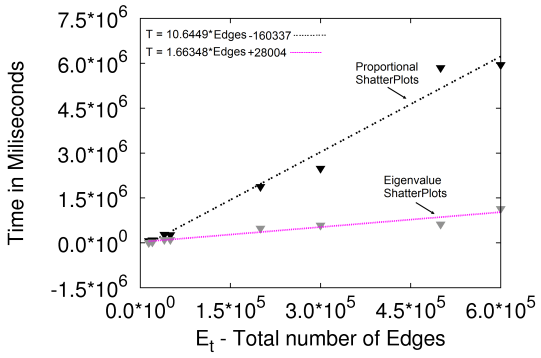


Figure 6: Scalability of *Proportional ShatterPlots* black double dotted line on dark triangles and *Eigenvalue ShatterPlots* pink dotted line on gray triangles.

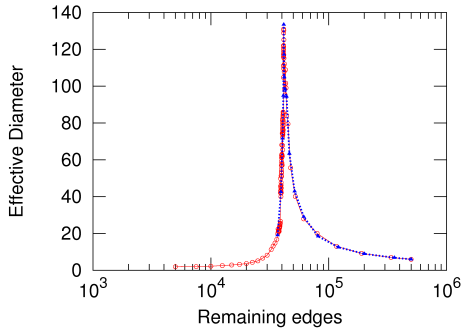


Figure 7: ShatterPlots of a Erdős-Rényi graph with 500k edges using *Eigenvalue ShatterPlots* (blue triangles) and *Proportional ShatterPlots* (red circles).

Black and gray triangles correspond to the *Proportional ShatterPlots* and *Eigenvalue ShatterPlots* methods, respectively. We used the same datasets for both algorithms. The fitting lines (dotted-black, and solid red) show that both methods seem to scale up linearly with the graph size, with *Eigenvalue ShatterPlots* being significantly faster (up to 8x).

8 Conclusions

Our goal is to find patterns in real graphs, to help us spot masked and synthetic graphs. The main idea is to use a "crash test" approach: we propose to shatter the graph, and observe its behavior. We proposed 'ShatterPlots', a new tool for studying graphs, and we showed how it can help us find surprising patterns. Our contributions are

- The careful, scalable design of the tool. ShatterPlots needs less than $O(E)$ effort on each iteration, and a small number of iterations, thanks to our adaptive method.
- The use of *Eigenvalue* pattern to optimize the ShatterPlots (up to 8 times).
- Our observations, and confirmation/demolition of conjectures:
  - all criteria shatter at the same point, with diameter being the only one with a clear, sharp edge.
  - real graphs are way above Shattering point
- Discovery of new patterns:
  - the Shattering point is at $1/\lambda_1 \cong E_s/E_t$, like one might expect from epidemic threshold theory;
  - Several additional patterns, like the *30-per-cent* and the *NodeShatteringRatio* patterns.
- Our patterns can spot synthetic/masked graphs

Future work could focus on the analysis of graphs over time, as well as on parallelization of the method, say, on a 'hadoop'/map-reduce architecture.

Acknowledgements

Ana Paula Appel work has been funded by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) PDEE project number 3960-07-2. Ana Paula Appel thanks CNPq and Fapesp.

References

[1] Google programming contest. `http://www.google.com/programming-contest/`, 2002.

[2] C. Aggarwal and P. Yu. Outlier detection for high-dimensional data. In *SIGMOD*, pages 37–46, 2001.

[3] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–381, 2000.

[4] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 2002.

[5] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Hafner Press, 2nd edition, 1975.

[6] Per Bak. How nature works : The science of self-organized criticality, September 1996.

[7] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, pages 509–512, 1999.

[8] Vladimir Batagelj and Matjaz Zaversnik. Generalized cores. *ArXiv*, (cs.DS/0202039), Feb 2002.

[9] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web: experiments and models. In *WWW Conf.*, 2000.

[10] J. M. Carlson and J. Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Physics Review E*, 60(2):1412–1427, 1999.

[11] D. Chakrabarti. AutoPart: Parameter-free graph partitioning and outlier detection. In *PKDD*, 2004.

[12] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jure Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.*, 10(4):1–26, 2008.

[13] A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks, August 2004.

[14] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 03)*, Washington, DC, August 24-27 2003.

[15] Rick Durrett. *Random Graph Dynamics*. Cambridge University Press, Cambridge, 2007.

[16] Andreas Engel. On large deviation properties of erdos-renyi random graphs. *Journal of Statistical Physics*, 117:387–426(40), November 2004.

[17] P. Erdös and A. Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Acadamy of Science*, 5:17–61, 1960.

[18] Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically Optimized Trade-offs: A new paradigm for power laws in the Internet (extended abstract), 2002.

[19] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *SIGCOMM*, pages 251–262, 1999.

[20] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71, 2002.

[21] J. Gehrke, P. Ginsparg, and J.M. Kleinberg. Overview of the 2003 KDD Cup. *SIGKDD Explorations*, 5(2):149–151, 2003.

[22] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proc. Natl. Acad. Sci. USA*, volume 99, 2002.

[23] R. Kannan, S. Vempala, and A. Vetta. On clusterings – good, bad and spectral. In *FOCS*, 2000.

[24] George Karypis and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48:96–129, 1998.

[25] H. Kesten. The critical probability of bond percolation on the square lattice equals 1/2. *Communications in Mathematical Physics*, 74:41–59, February 1980.

[26] J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Proceedings of the International Conference on Combinatorics and Computing*, 1999.

[27] Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *CEAS*, 2004.

[28] Joseph S. Kong, Behnam A. Rezaei, Nima Sarshar, Vwani P. Roychowdhury, and P. Oscar Boykin. Collaborative spam filtering using e-mail networks. *Computer*, 39(8):67–73, 2006.

[29] Ravi Kumar, Andrew Tomkins, and Erik Vee. Connectivity structure of bipartite graphs via the knc-plot. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 129–138, New York, NY, USA, 2008. ACM.

[30] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM '07: Proceedings of the XXXth SIAM Conference on Data Mining*, 2007.

[31] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):2, 2007.

[32] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceeding of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, 2005.

[33] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[34] Milena Mihail and Christos Papadimitriou. On the eigenvalue power law. In *RANDOM*, Harvard, MA, 2002.

[35] S. Milgram. The small-world problem. *Psychology Today*, 2:60–67, 1967.

[36] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[37] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. ANF: A fast and scalable tool for data mining in massive graphs. In *SIGKDD*, Edmonton, AB, Canada, 2002.

[38] David M. Pennock, Gary W. Flake, Steve Lawrence, Eric J. Glover, and C. Lee Giles. Winners don't take all: Characterizing the competition for links on the Web. *Proceedings of the National Academy of Sciences*, 99(8):5207–5211, 2002.

[39] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, September 2002.

[40] Sidney Redner. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B*, 4:131–134, 1998.

[41] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *ISWC*, 2003.

[42] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.

[43] Manfred Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. W.H. Freeman and Company, New York, 1991.

[44] Ricard Sole and Brian Goodwin. *Signs of Life: How Complexity Pervades Biology*. Perseus Books Group, New York, NY, 2000.

[45] S. L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the Internet topology. In *Global Internet, San Antonio, Texas*, 2001.

[46] Charalampos Tsourakakis. Fast counting of triangles in large real networks, without counting: Algorithms and laws. In *ICDM*, 2008.

[47] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon