NetMine: New Mining Tools for Large Graphs

Deepayan Chakrabarti, Yiping Zhan, Daniel Blandford,
Christos Faloutsos and Guy Blelloch
School of Computer Science
Carnegie Mellon University
{deepay, yzhan, dkb1, christos, guyb}@cs.cmu.edu

April 29, 2004

Abstract

Interesting patterns show up in large graphs in a variety of settings: power-laws and "bow-tie" structure in the World Wide Web, small diameter for Peer-to-Peer overlay graphs, and many others. Discovering such patterns and regularities has many wide-ranging applications, from understanding viral propagation to criminology and law-enforcement. The *NetMine* system includes a toolbox of patterns that show up in real graphs. Apart from previously studied patterns such as power-laws degree-distributions, it adds the new "min-cut plot", which our recently proposed *R-MAT* [13] generator (also a part of *NetMine*) can match well. We also propose adding a novel tool called *A-plots* to the graph miner's arsenal, and show how this can be used to find interesting patterns and outliers in large real-world graphs.

1 Introduction

There is increasing interest in finding common patterns in large graphs drawn from a surprisingly diverse number of settings. The World Wide Web exhibits power laws as well as a "bow-tie" structure [11]; most real graphs have surprisingly small diameters (about 19 for the web [6], about 5-6 for the Internet autonomous-system topologies [18]). Similar power laws and "small world" phenomena appear in peer-to-peer overlay graphs [29] and in the epinions.com who-trusts-whom network [28]. Exactly because of the importance of large graphs, there are several graph generators that try to create synthetic, but realistic-looking graphs [23, 5, 13]. Such generators are useful for simulations (eg., of internet routing protocols, or virus propagation analysis), extrapolations and what-if scenarios ("What will the internet look like, when it is double its current size?", "How will a virus propagate, if we immunize only the highest degree nodes?")

Discovering patterns, laws and regularities in large real networks has numerous applications: Analysis of virus propagation patterns, on both social/e-mail as well as physical-contact networks [33]; link analysis, for criminology and law enforcement [15]; food webs, to help us understand the importance of an endangered species;

networks of regulatory genes; networks of interacting proteins [7] and so on.

How could we analyze such large graphs automatically? Which patterns should we be looking for? How can we spot suspicious/erroneous subgraphs quickly? These are the questions that our *NetMine* system focuses on. The main contributions of this paper are that

- it proposes the "min-cut plots", an interesting pattern to check for while analyzing a graph
- it proposes the "*A-plots*" as a tool for quickly finding suspicious subgraphs/nodes
- it scales very well with size of the graph for all its tasks, and thus is able to quickly handle graphs of hundreds of thousands of nodes
- it shows how to interpret these plots, and how we found surprising patterns and outliers on real graphs

The layout of the paper is as follows. Section 2 details previous work in the field of graph mining. We then describe our proposed ideas for mining graphs in Section 3. This is followed by experiments in Section 4. We finally conclude in Section 5.

2 Background and Related Work

Let's establish some terminology first: A *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set $\mathcal{V}$ of N nodes and a set $\mathcal{E}$ of E edges between them. For example, the network of Internet routers and their physical links is an (undirected) graph. First we discuss important patterns that have been discovered in real graphs, and then we discuss some existing graph generators.

Patterns and "Laws" : Several interesting patterns often show up in real-world graphs. They can mainly be grouped into power-laws and small-diameter phenomena.

Power laws: Skewed distributions, and laws of the form $y = x^a$, appear very often. Such a law comes out as a line of slope a, when we plot y versus x in log-log scales. Plotting the number of nodes (c_k) with a certain degree k, versus their degree k on a log-log scale gives the degree-plot, which often exhibits a power-law. Such power-law degree distributions are found in the Internet topology [18], the World Wide Web [21], the citation graph [27] and many others. The eigenvalues of the adjacency matrix, when plotted versus their rank in log-log scales (called the "scree-plot"), also show a power-law. Very recently, deviations from the power laws have been observed [26], which may be explained using lognormal distributions [8].

Diameter - small world phenomena: Most real graphs have surprisingly small diameters: the famous "six degrees of separation" in social networks [24], 19 for the Web [6], and low diameters for the Internet topology [32].

Network values of nodes: The elements $v_{1,i}$ in the first eigenvector $\vec{v_1}$ of the adjacency matrix roughly correspond to the "network values" of nodes in an undirected graph [28]. For a directed graph, the corresponding values are given by the first left singular vector $\vec{v_1}$ and the first right singular vector $\vec{u_1}$. We use these network values in our *A-plots*, and point out several striking patterns which show up.

Measures: There is a huge list of measures in the literature of computer networks, social networks and graph theory, including the following. For a node, we have the clustering coefficient, "prestige", and "importance" [14]; for a whole network, we can compute the "expansion", "resilience" and "distortion" [31], and the characteristic path length [12]; for each edge, the "stress" [19].

Graph Generators : There are several methods for generating graphs. The earliest model was the random graph model by Erdős and Rényi [16], but it does not match power-laws. Given a degree distribution (typically following a power-law), several models try to find a graph that matches this degree distribution [2, 3, 22, 25]. Other models try to provide a simple set of rules of placing edges between nodes; the typical representative here is the Barabási-Albert (BA) method [4] with the "*preferential attachment*" idea: Keep adding nodes; new nodes prefer to connect to nodes with high degrees. Several modifications and alternatives have been suggested [5, 21, 12, 26, 23]. Another class of generators consider geometry [10, 17].

A recently proposed graph generator is *R-MAT* [13], which has successfully matched many of the patterns mentioned before. The *R-MAT* (for **R**ecursive **MAT**rix) model recursively subdivides the adjacency matrix into four equal-sized partitions, and distributes edges within these partitions with unequal probabilities: Starting off with an empty adjacency matrix, we "drop" edges into the matrix one at a time. Each edge chooses one of the four partitions with probabilities a, b, c, d respectively. Of course, $a + b + c + d = 1$. The chosen partition is again subdivided into four smaller partitions, and the procedure is repeated until we reach a simple cell (=1×1 partition).

NetMine provides patterns and checkpoints which any graph generator must match, if it is to realistically model real-world graphs. Thus, the *NetMine* toolkit would be essential to evaluating the performance of graph generators.

3 Proposed Ideas

The contributions outlined here are twofold: (1) we present the "min-cut plot", a new checkpoint for comparing a synthetically-generated graph to a real one, and (2) we present a novel tool called *A-plots* for interactive inspection of graphs and for finding erroneous/outlier nodes and subgraphs. Both of these are described below.

"Min-cut plots": Several criteria have been previously proposed to compare a synthetic graph to a real-world graph. These include degree distributions, hop-plots, scree-plots and others. *NetMine* includes all these, and adds "min-cut plots".

A min-cut of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a partition of the set of vertices $\mathcal{V}$ into two sets $\mathcal{V}_1$ and $\mathcal{V} - \mathcal{V}_1$ such that both partitions are of approximately the same size, and the number of edges crossing partition boundaries is minimized. The number of such edges in the min-cut is called the min-cut size. Min-cut sizes of various classes of graphs has been studied extensively, and are known to have important effects on other properties of the graphs [30]. For example, Figure 1(a) shows a regular 2D grid graph, and one possible min-cut of the graph. We see that if the number of nodes is N, then the size of the min-cut (in this case) is $O(\sqrt{N})$.

The min-cut plot is built as follows: given a graph, its min-cut is found, and the set of edges crossing partition boundaries deleted. This divides the graph into two disjoint graphs; the min-cut algorithm is then applied recursively to each of these sub-graphs. This continues till the size of the graph reaches a small value (set to 20 in our case). Each application of the min-cut algorithm becomes a point in the min-cut plot. The graphs are drawn on a log-log scale. The x-axis is the number of edges in a given graph. The y-axis is the fraction of that graph's edges that were included in the edge-cut for that graph's separator.

Figure 1(b) shows the min-cut plot for the 2D grid graph. In plot (c), the value on the y-axis is averaged over all points having the same x-coordinate. The min-cut size is $O(\sqrt{N})$, so this plot should have a slope of -0.5, which is exactly what we observe.

A-plots: A simple way to find suspicious nodes/subgraphs in a large graph could be very useful in a variety of situations. However, the obvious approach of trying to visualize the graph does not work very well: visualization of large graphs is notoriously tough and time consuming, and is a research topic in its own right. Our proposed solution, called *A-plots*, consists of three types of plots for undirected graphs: (1) the plot of the adjacency matrix with nodes sorted in decreasing order by their network values (*RV-RV* plot, for Rank of network Value), (2) the plot of the degree of a node verses its rank of network value (*D-RV* plot, for Degree verses Rank of network Value), and (3) the plot of the adjacency matrix with nodes sorted in decreasing order by their degrees (*RD-RD* plot, for Rank of Degree). Together, these plots often reveal interesting patterns and properties of the graph. We propose these as valuable tools for getting an overall view of an undirected graph.

4 Experiments

We performed experiments to answer the following questions:

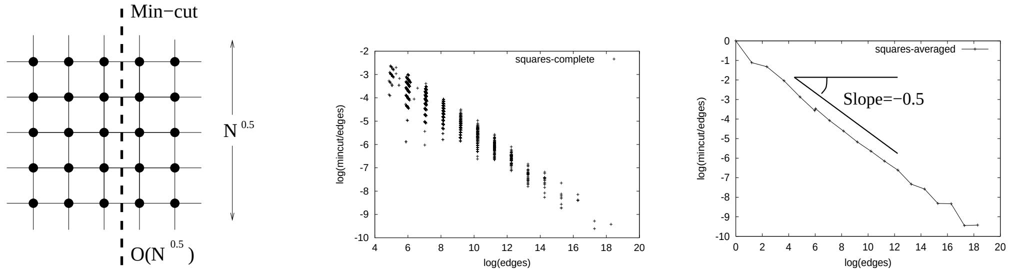
- **Q1:** How do the min-cut plots look for real-world graphs, and does *R-MAT* match them?

- **Q2:** How can *A-plots* be used for analyzing large graphs?

We used several natural and synthetic datasets in our experiments. *Epinions* is a graph of who-trusts-whom from `www.epinions.com`. *Lucent* is an undirected graph of network routers, obtained from `www.isi.edu/scan/mercator/maps.html`. *Router* is a larger graph (the SCAN+Lucent map) from the same URL, which subsumes the *Lucent* graph. *Clickstream* is a bipartite graph linking user-ids to web-domains. *Google* is a graph of webpage connectivity from the Google [1] programming contest. Characteristics of these datasets are shown in Table 1.

4.1 [$Q1$] Min-cut Plots

We plotted min-cut sizes for a variety of graphs. For each graph listed we used the Metis graph partitioning library [20] to generate a separator, as described by Blandford, Blelloch, and Kash [9].



(a) A grid and its min-cut (b) 2D grid min-cut plot (c) Averaged min-cut plot

Figure 1: Plot (a) shows a portion of a regular 400x400 2D grid, and a possible min-cut. Plot (b) shows the full min-cut plot, and plot (c) shows the averaged plot. If the number of nodes is N, the length of each side is $\sqrt{N}$. Then the size of the min-cut is $O(\sqrt{N})$, which leads to a slope of -0.5, which is exactly what we observe.

Graph	*Nodes*	*Edges*
Epinions	75888	508837
Lucent	112969	181639
Router	284805	898492
Clickstream	222704	952580
Google	916428	5105039

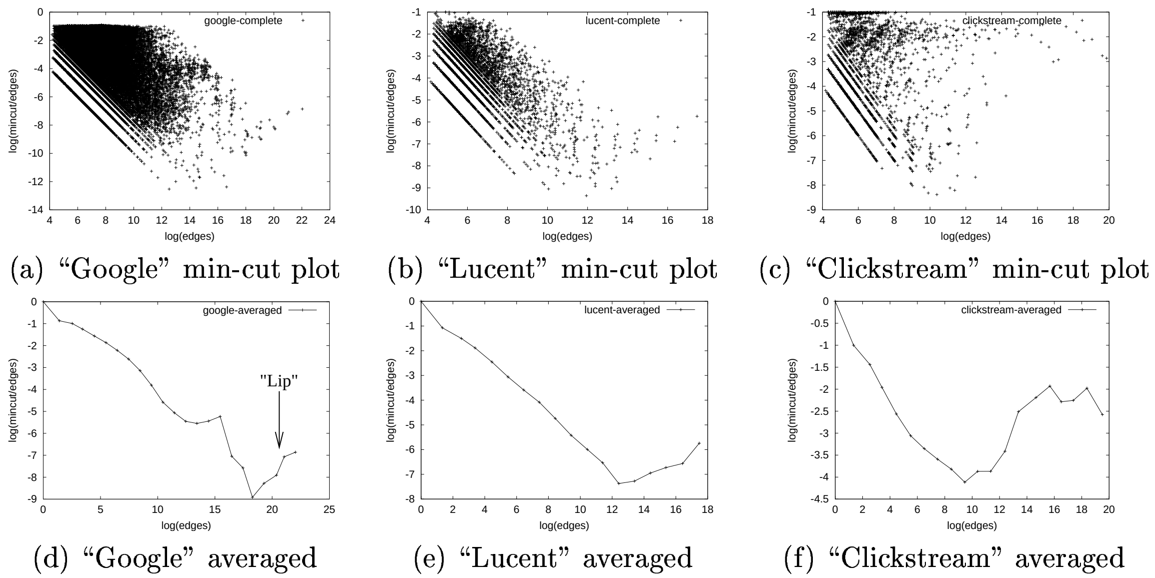
Table 1: List of datasets used in the experiments and their details.

Figure 2 shows min-cut sizes of some real-world graphs. For random graphs, we expect about half the edges to be included in the cut. Hence, the min-cut plot of a random graph would be a straight horizontal line with a y-coordinate of about $\log(0.5) = -1$. A very separable graph (for example, a line graph) might have only one edge in the cut; such a graph with N edges would have a y-coordinate of $\log(1/N) = -\log(N)$, and its min-cut plot would thus be on the line $y = -x$. As we can see from Figure 2, the plots for real-world graphs do not match either of these situations, meaning that real-world graphs are quite far from either random graphs or simple line graphs.

Observation 1 (Noise) *We see that real-world graphs seem to have a lot of "noise" in their min-cut plots, as shown by the first row of Figure 2.*

Observation 2 ("Lip") *The ratio of min-cut size to number of edges decreases with increasing edges, except for graphs with large number of edges, where we observe a "lip" in the min-cut plot.*

The min-cut plot contains important information about the graph [30]. Hence, any synthetically generated graph meant to simulate a real-world graph should match the min-cut plot of the real-world graph. In Figure 3, we compare the mincut-plots for the Epinions graph with a graph generated *R-MAT*. As can be seen, the basic shape of the plot is the same in both cases,



(a) "Google" min-cut plot (b) "Lucent" min-cut plot (c) "Clickstream" min-cut plot

(d) "Google" averaged (e) "Lucent" averaged (f) "Clickstream" averaged

Figure 2: These are the min-cut plots for several datasets. We plot the ratio of mincut-size to edges versus number of edges on a log-log scale. The first row shows the actual plots; in the second row, the cutsize-to-edge ratio is averaged over all points with the same number of edges.

though the *R-MAT* plot appears to be shifted slightly from the original.

Observation 3 *The graphs generated by R-MAT appear to match the basic shape of the min-cut plot for several real-world graphs.*

4.2 [*Q*2] *A-plots*

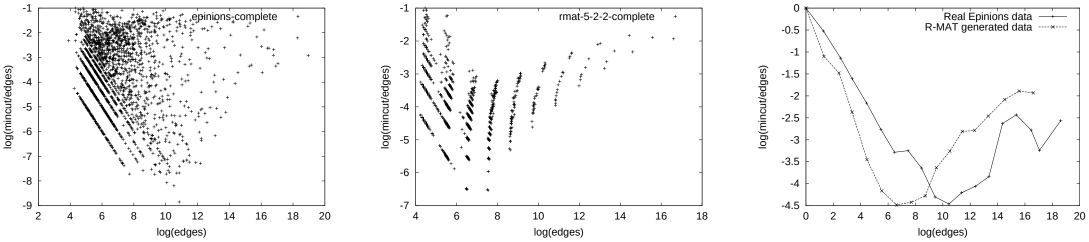
Figures 4 and 5 show *A-plots* for the Router dataset. Figure 4 shows the *RV-RV* and *RD-RD* plots, and Figure 5 shows the *D-RV* plot under different scalings. We can make the following observations:

Observation 4 ("Water-Drop") *The RV-RV plot has a clean and smooth oval-shaped boundary for the edges in the graph.*

Explanation: The boundary of the edges is defined by the one-degree nodes in the graph. There are many such nodes because of the power law distribution of the degrees. Let I_i denote the network value of node i; if node i has a degree of one and node j is the only node it is connected to, the properties of spectral decomposition of a matrix imply that

$$I_i = 1/\lambda_1 * I_j \tag{1}$$

where λ_1 is the largest eigenvalue of the adjacency matrix of the graph [34]. Therefore the boundary of edges in the *RV-RV* plot can be calculated from the first eigenvalue and eigenvector. Figure 4(b) shows just this; the solid curve represents degree-one nodes. These are obviously



(a) "Epinions" min-cuts (b) *R-MAT* min-cuts (c) Averaged min-cuts

Figure 3: Here, we compare min-cut plots for the Epinions dataset and a dataset generated by *R-MAT*, using properly chosen parameters (in this case, a=0.5, b=0.2, c=0.2, d=0.1) We see from plot (c) that the shapes of the min-cut plots are similar.

the boundary curves for plot (a).

We also see that there is no edge at all outside of the boundary. Let node i have network value I_i, and have node j with network value I_j as its most "important" neighbor (in the sense of high network value). Then, $I_i \geq 1/\lambda_1 * I_j$. Therefore all edges are confined within the boundary in the RV-RV plot.

Observation 5 (Nested Water-Drops) *There are a pair of "secondary" lines within the boundary of the edges in the RV-RV plot.*

Explanation: These lines are the results of some two-degree nodes. When a node i has two degrees and the two nodes it is connected to have about the same network values (say, I_j), we can calculate where the involved edges will show up in the *RV-RV* plot similar to the one-degree case:

$$I_i = 2/\lambda_1 * I_j \tag{2}$$

The dashed lines in Figure 4(b) show the results, which match with the *RV-RV* plot. The presence of these "secondary" lines in the plot means that a significant number of the two-degree nodes in the graph are connected to two "similar" (similar as is defined by similar network values) nodes. The presence of the faint "tertiary" lines can be explained accordingly.

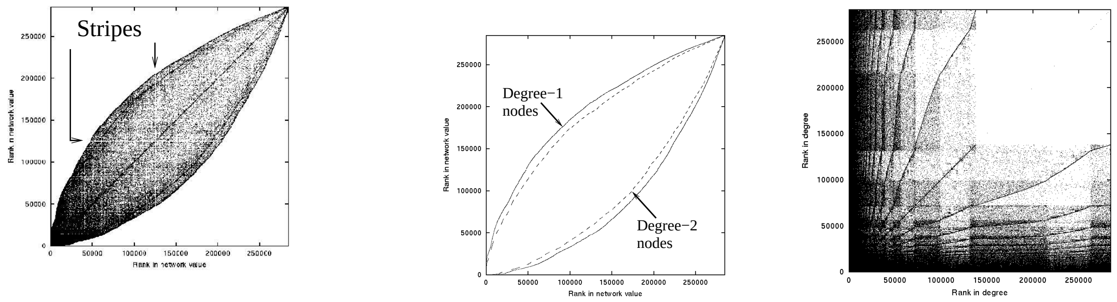
Observation 6 (Diagonal) *There is more or less a solid line that goes through the diagonal of the RV-RV plot even though the adjacency matrix does not include any self-edges.*

Explanation: This means a node is more likely to be connected with "similar" nodes.

Observation 7 (White Stripes) *There are white stripes (both vertical and horizontal) visible in both the RV-RV and the D-RV plots.*

Explanation: The stripes come from a large number of nodes that are connected to exactly the same nodes, usually just one or two. Since nodes that are connected the same way have exactly the same network values, they show up as a group and become visible in the *RV-RV* and *D-RV* plots (Figures 4(a) and 5(a, b) respectively).

Observation 8 (Isolated Components) *The largely empty white square in the corner of the RD-RD plot results from connections between one-degree nodes.*



(a) *RV-RV* plot (b) Estimated boundaries (c) *RD-RD* plot

Figure 4: *A-plots* for the "Router" graph: Plot (a) shows the *RV-RV* plot, and a very interesting "Water-Drop" pattern is immediately apparent. The outermost "boundary stripes" are due to nodes of degree one (solid curve) and two (broken curve), whose positions can be calculated using Equations 1,2 as shown by plot (b). Plot (c) shows the *RD-RD* plot.

Explanation: Any dots (edges) in this area correspond to two-node isolated components.

Observation 9 (Degree vs. Importance) *Figure 5(c) shows several points in the D-RV plot having high degree, but low network value (and thus low rank). Thus, high degree does not imply high "importance".*
Explanation: The *D-RV* plot in Figure 5(c) shows that the two highest-degree nodes actually have low 'network value'. This is counter-intuitive - how could it possibly be the case, in a power-law graph? Is it a data collection error?

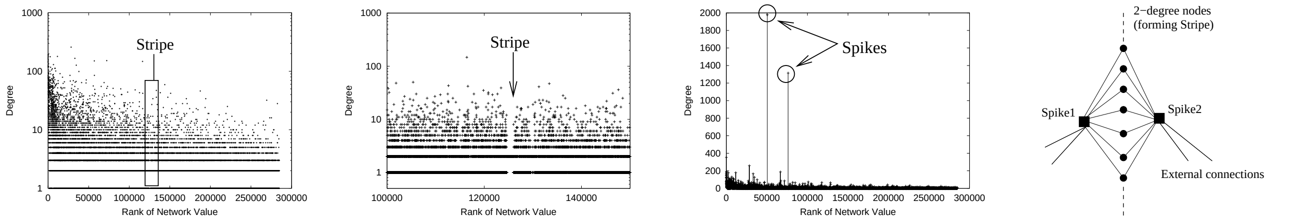
The answer is surprising, and actually also explains the white stripe in Figure 5(a,b): The two highest-degree nodes (labeled 'Spike1' and 'Spike2'), and a large number of two-degree nodes, form a subgraph like the one shown in Figure 5(d). 'Spike1' and 'Spike2', being away from the core of the network, have much lower network value than what their high degree would promise. Their satelites (= all the 2-degree nodes connected to them) have identical, relatively high network values, which cause the white strip in Figure 5(a,b). We are currently investigating with domain experts the reasons for such a weird sub-graph. However, our point is that the proposed *D-RV* and *RV-RV* plots exactly spotted this strange pattern, which would go undetected if we only used the traditional, or even recent tools, like degree-plots, scree-plots etc.

5 Conclusions

We propose several new tools for mining large graphs. Our emphasis is on scalable algorithms that can handle arbitrarily large graphs. When applied on real graphs, our new tools discovered patterns that were not visible with the known tools (like degree plots, hop-plots etc).

The contributions of this work are:

- *A-plots*: These plots provide new viewpoints for inspecting large graphs. We no-



(a)*D-RV* plot(log-lin) (b)*D-RV* plot(blowup) (c)*D-RV* plot(lin-lin) (d) Spikes-and-stripes

Figure 5: *A-plots* for the "Router" dataset: Plot (a) shows the *D-RV* plot, and plot (b) shows a blowup of a portion, clearly demonstrating the "white stripes" phenomenon. Plot (c) shows the *D-RV* plot in the linear-linear scale; nodes with the highest degree do not have the highest network value. Plot (d) shows the actual network configuration of routers involved in the stripe and spikes. An explanation is provided in Observation 9 in the text.

ticed some striking patterns ("water-drops", stripes, "lone" points), and we showed how to interpret them.

- **Min-cut plots:** They show the relative size of the minimum cut in a graph partition. For regular 2-d and 3-d grid-style networks (like Delaunay triangulations for finite element analysis), these plots have a slope that depends on the intrinsic dimensionality of the grid. However, for real graphs, these plots show significantly more 'noise', as well as a 'lip'. We were pleasantly surprised when our recently proposed *R-MAT* model [13] showed a very similar behavior.

Thanks to the proposed tools, we were able to make several interesting observations. We showed how *A-plots* can be used to spot outliers, and make observations about the degree of nodes, by simply looking at the adjacency matrix in different ways.

Future work could focus on the introduction of additional tools. These have to be selected carefully so that they are orthogonal to the existing tools; moreover, their implementation has also to be done carefully, so that they are scalable for large graphs ($10^4 - 10^9$ nodes and edges).

References

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

[2] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *STOC*, pages 171–180, 2000.

[3] William Aiello, Fan Chung, and Linyuan Lu. Random evolution in massive graphs. In *FOCS*, 2001.

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

[5] R. Albert and A.-L. Barabási. Topology of complex networks: local events and universality. *Physical Review Letters*, 2000.

[6] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the world-wide web. *Nature*, 401:130–131, September 1999.

[7] A.-L. Barabási. *Linked: The New Science of Networks*. Perseus Publishing, first edition, May 2002.

[8] Zhiqiang Bi, Christos Faloutsos, and Flip Korn. The DGX distribution for mining massive, skewed data. In *KDD*, pages 17–26, 2001.

[9] Daniel Blandford, Guy E. Blelloch, and Ian Kash. Compact representations of separable graphs. In *SODA*, 2003.

[10] B.M.Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9), December 1988.

[11] 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.

[12] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *INFOCOM*, 2002.

[13] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-MAT: A recursive model for graph mining. In *SIAM Data Mining*, 2004.

[14] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2002.

[15] H. Chen, J. Schroeder, R. Hauck, L. Ridgeway, H. Atabaksh, H. Gupta, C. Boarman, K. Rasmussen, and A. Clements. Coplink Connect: Information and knowledge management for law enforcement. *CACM*, 46(1):28–34, January 2003.

[16] P. Erdős and A. Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Acadamy of Science*, 5:17–67, 1960.

[17] Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet (extended abstract), 2002.

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

[19] Christos Gkantsidis, Milena Mihail, and Ellen Zegura. Spectral analysis of internet topologies. In *INFOCOM*, 2003.

[20] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. Technical Report TR 95-035, 1995.

[21] 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.

[22] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale

knowledge bases from the web. In *VLDB*, Edinburgh, Scotland, 1999.

[23] A. Medina, I. Matta, and J. Byers. On the origin of power laws in internet topologies. In *SIGCOMM*, volume 30, pages 18–34, 2000.

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

[25] Christopher R. Palmer and J. Gregory Steffan. Generating network topologies that obey power laws. In *GLOBECOM*, November 2000.

[26] 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.

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

[28] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *SIGKDD*, pages 61–70, Edmonton, Canada, 2002.

[29] 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.

[30] Arnold L. Rosenberg and Lenwood S. Heath. *Graph Separators, with Applications*. Kluwer Academic/Plenum Pulishers, 2001.

[31] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topologies, power laws, and hierarchy. Technical Report 01-746, University of Southern California, 2001.

[32] 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.

[33] Chenxi Wang, J. C. Knight, and M. C. Elder. On computer viral infection and the effect of immunization. In *ACSAC*, pages 246–256, 2000.

[34] Y. Zhan. Tools for graph mining. Master's thesis, Carnegie Mellon University, 2003.