Graph Mining: Laws, Generators and Algorithms

DEEPAYAN CHAKRABARTI and CHRISTOS FALOUTSOS
Yahoo! Research and Carnegie Mellon University

How does the Web look? How could we tell an "abnormal" social network from a "normal" one? These and similar questions are important in many fields where the data can intuitively be cast as a graph; examples range from computer networks to sociology to biology and many more. Indeed, any $M : N$ relation in database terminology can be represented as a graph. A lot of these questions boil down to the following: "How can we generate synthetic but *realistic* graphs?" To answer this, we must first understand what *patterns* are common in real-world graphs, and can thus be considered a mark of normality/realism. This survey give an overview of the incredible variety of work that has been done on these problems. One of our main contributions is the integration of points of view from physics, mathematics, sociology and computer science. Further, we briefly describe recent advances on some related and interesting graph problems.

Categories and Subject Descriptors: E.1 [**Data Structures**]:

General Terms: Algorithms, Measurement

Additional Key Words and Phrases: Generators, graphs, patterns, social networks

1. INTRODUCTION

Informally, a graph is set of nodes, pairs of which might be connected by edges. In a wide array of disciplines, data can be intuitively cast into this format. For example, computer networks consist of routers/computers (nodes) and the links (edges) between them. Social networks consist of individuals and their interconnections (which could be business relationships, or kinship, or trust, etc.) Protein interaction networks link proteins which must work together to perform some particular biological function. Ecological food webs link species with predator-prey relationships. In these and many other fields, graphs are seemingly ubiquitous.

The problems of detecting abnormalities ("outliers") in a given graph, and of *generating* synthetic but realistic graphs, have received considerable attention recently. Both are tightly coupled to the problem of finding the distinguishing characteristics of real-world graphs, that is, the "patterns" that show up frequently in such

This material is based upon work supported by the National Science Foundation under Grants No. IIS-0083148, IIS-0113089, IIS-0209107 IIS-0205224 INT-0318547 SENSOR-0329549 EF-0331657IIS-0326322 CNS-0433540 by the Pennsylvania Infrastructure Technology Alliance (PITA) Grant No. 22-901-0001. Additional funding was provided by donations from Intel, and by a gift from Northrop-Grumman Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

graphs and can thus be considered as marks of "realism." A good generator will create graphs which match these patterns. Patterns and generators are important for many applications:

—*Detection of abnormal subgraphs/edges/nodes:* Abnormalities should deviate from the "normal" patterns, so understanding the patterns of naturally occurring graphs is a prerequisite for detection of such outliers.

—*Simulation studies:* Algorithms meant for large real-world graphs can be tested on synthetic graphs which "look like" the original graphs. For example, in order to test the next-generation Internet protocol, we would like to simulate it on a graph that is "similar" to what the Internet will look like a few years into the future.

—*Realism of samples:* We might want to build a small sample graph that is similar to a given large graph. This smaller graph needs to match the "patterns" of the large graph to be realistic.

—*Graph compression:* Graph patterns represent regularities in the data. Such regularities can be used to better compress the data.

Thus, we need to detect patterns in graphs, and then generate synthetic graphs matching such patterns automatically.

This is a hard problem. What patterns should we look for? What do such patterns mean? How can we generate them? A lot of research ink has been spent on this problem, not only by computer scientists but also physicists, mathematicians, sociologists and others. However, there is little interaction among these fields, with the result that they often use different terminology and do not benefit from each other's advances. In this survey, we attempt to give an overview of the main ideas. Our focus is on combining sources from all the different fields, to gain a coherent picture of the current state-of-the-art. The interested reader is also referred to some excellent and entertaining books on the topic [Barabási 2002; Watts 2003; Dorogovtsev and Mendes 2003].

The organization of this survey is as follows. In section 2, we discuss graph patterns that appear to be common in real-world graphs. Then, in section 3, we describe some graph generators which try to match one or more of these patterns. Typically, we only provide the main ideas and approaches; the interested reader can read the relevant references for details. In all of these, we attempt to collate information from several fields of research. In section 4, we consider some interesting questions from Social Network Analysis which are particularly relevant to social networks. Some of these appear to have no analogues in other fields. We briefly touch upon other recent work on related topics in section 5. We present a discussion on open topics of research in section 6, and finally conclude in section 7. Table I lists the symbols used in this survey.

2. GRAPH PATTERNS

What are the distinguishing characteristics of graphs? What "rules" and "patterns" hold for them? When can we say that two different graphs are *similar* to each other? In order to come up with models to generate graphs, we need some way of comparing a natural graph to a synthetically generated one; the better the match, the better

Symbol	Description
N	Number of nodes in the graph
E	Number of edges in the graph
k	Degree for some node
$< k >$	Average degree of nodes in the graph
CC	Clustering coefficient of the graph
$CC(k)$	Clustering coefficient of degree-k nodes
γ	Power law exponent: $y(x) \propto x^{-\gamma}$
t	Time/iterations since the start of an algorithm

Table I. *Table of symbols*

the model. However, to answer these questions, we need to have some basic set of graph attributes; these would be our vocabulary in which we can discuss different graph types. Finding such attributes will be the focus of this section.

What is a "good" pattern? One that can help distinguish between an actual real-world graph and any fake one. However, we immediately run into several problems. First, given the plethora of different natural and man-made phenomena which give rise to graphs, can we expect all such graphs to follow any particular patterns? Second, is there any *single* pattern which can help differentiate between all real and fake graphs? A third problem (more of a constraint than a problem) is that we want to find patterns which can be computed efficiently; the graphs we are looking at typically have at least around 10^5 nodes and 10^6 edges. A pattern which takes $O(N^3)$ or $O(N^2)$ time in the number of nodes N might easily become impractical for such graphs.

The best answer we can give today is that while there are many differences between graphs, some patterns show up regularly. Work has focused on finding several such patterns, which *together* characterize naturally occurring graphs. The main ones appear to be:

—Power laws,

—Small diameters, and

—"Community" effects.

Our discussion of graph patterns will follow the same structure. We look at power laws in Section 2.1, small diameters in Section 2.3, "community" effects in Section 2.4, and list some other patterns in Section 2.5. For each, we also give the computational requirements for finding/computing the pattern, and some real-world examples of that pattern. Definitions are provided for key ideas which are used repeatedly. In Section 2.6, we will discuss some patterns in the *evolution* of graphs over time. Finally, in Section 2.7, we discuss patterns specific to some well-known graphs, like the Internet and the WWW.

2.1 Power Laws

While the Gaussian distribution is common in nature, there are many cases where the probability of events far to the right of the mean is significantly higher than in Gaussians. In the Internet, for example, most routers have a very low degree (perhaps "home" routers), while a few routers have extremely high degree (perhaps

the "core" routers of the Internet backbone) [Faloutsos et al. 1999]. Power-law distributions attempt to model this.

We will divide the following discussion into two parts. First, we will discuss "traditional" power laws: their definition, how to compute them, and real-world examples of their presence. Then, we will discuss deviations from pure power laws, and some common methods to model these.

2.1.1 *"Traditional" Power Laws*

DEFINITION 2.1 POWER LAW. *Two variables x and y are related by a power law when:*

$$y(x) = Ax^{-\gamma} \tag{1}$$

where A and γ are positive constants. The constant γ is often called the power law exponent.

DEFINITION 2.2 POWER LAW DISTRIBUTION. *A random variable is distributed according to a power law when the probability density function (pdf) is given by:*

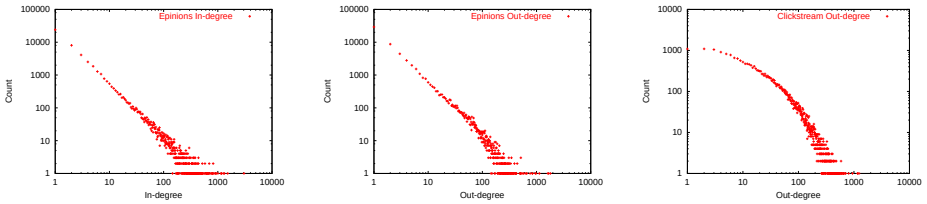
$$p(x) = Ax^{-\gamma}, \quad \gamma > 1, x \geq x_{min} \tag{2}$$

The extra $\gamma > 1$ requirement ensures that $p(x)$ can be normalized. Power laws with $\gamma < 1$ rarely occur in nature, if ever [Newman 2005].

Skewed distributions, such as power laws, occur very often. In the Internet graph, the degree distribution follows such a power law [Faloutsos et al. 1999]; that is, the count c_k of nodes with degree k, versus the degree k, is a line on a log-log scale. The eigenvalues of the adjacency matrix of the Internet graph also show a similar behavior: when eigenvalues are plotted versus their rank on a log-log scale (called the scree plot), the result is a straight line. A possible explanation of this is provided by Mihail and Papadimitriou [2002]. The World Wide Web graph also obeys power laws [Kleinberg et al. 1999]: the in-degree and out-degree distributions both follow power-laws, as well as the number of the so-called "bipartite cores" ($\approx$ communities, which we will see later) and the distribution of PageRank values [Brin and Page 1998; Pandurangan et al. 2002]. Redner [1998] shows that the citation graph of scientific literature follows a power law with exponent 3. Figures 1(a) and 1(b) show two examples of power laws.

The significance of a power law distribution $p(x)$ lies in the fact that it decay only polynomially quickly as $x \to \infty$, instead of exponential decay for the Gaussian distribution. Thus, a power law degree distribution would be much more likely to have nodes with a very high degree (much larger than the mean) than the Gaussian distribution. Graphs exhibiting such degree distributions are called *scale-free* graphs, because the form of $y(x)$ in Equation 1 remains unchanged to within a multiplicative factor when the variable x is multiplied by a scaling factor (in other words, $y(ax) = by(x)$). Thus, there is no special "characteristic scale" for the variables; the functional form of the relationship remains the same for all scales.

Computation issues: The process of finding a power law pattern can be divided into three parts: creating the scatter plot, computing the power law exponent, and checking for goodness of fit. We discuss these issues below, using the detection of power laws in degree distributions as an example.



(a) Epinions In-degree (b) Epinions Out-degree (c) Clickstream Out-degree

Fig. 1. *Power laws and deviations:* Plots (a) and (b) show the in-degree and out-degree distributions on a log-log scale for the *Epinions* graph (an online social network of 75,888 people and 508,960 edges [Domingos and Richardson 2001]). Both follow power-laws. In contrast, plot (c) shows the out-degree distribution of a *Clickstream* graph (a bipartite graph of users and the websites they surf [Montgomery and Faloutsos 2001]), which deviates from the power-law pattern.

Creating the scatter plot (for the degree distribution): The algorithm for calculating the degree distributions (irrespective of whether they are power laws or not) can be expressed concisely in SQL. Assuming that the graph is represented as a table with the schema `Graph(fromnode, tonode)`, the code for calculating in-degree and out-degree is given below. The case for weighted graphs, with the schema `Graph(fromnode, tonode, weight)`, is a simple extension of this.

```
SELECT outdegree, count(*)
FROM
   (SELECT count(*) AS outdegree
    FROM Graph
    GROUP BY fromnode)
GROUP BY outdegree
```

```
SELECT indegree, count(*)
FROM
   (SELECT count(*) AS indegree
    FROM Graph
    GROUP BY tonode)
GROUP BY indegree
```

Computing the power law exponent: This is no simple task: the power law could be only in the tail of the distribution and not over the entire distribution, estimators of the power law exponent could be biased, some required assumptions may not hold, and so on. Several methods are currently employed, though there is no clear "winner" at present.

(1) *Linear regression on the log-log scale:* We could plot the data on a log-log scale, then optionally "bin" them into equal-sized buckets, and finally find the slope of the linear fit. However, there are at least three problems: (i) this can lead to biased estimates [Goldstein et al. 2004], (ii) sometimes the power law is only in the *tail* of the distribution, and the point where the tail begins needs to be hand-picked, and (iii) the right end of the distribution is very noisy [Newman 2005]. However, this is the simplest technique, and seems to be the most popular one.

(2) *Linear regression after logarithmic binning:* This is the same as above, but the bin widths increase exponentially as we go towards the tail. In other words, the number of data points in each bin is counted, and then the height of each bin is then divided by its width to normalize. Plotting the histogram on a log-log scale would make the bin sizes equal, and the power-law can be fitted to the

heights of the bins. This reduces the noise in the tail buckets, fixing problem (iii). However, binning leads to loss of information; all that we retain in a bin is its average. In addition, issues (i) and (ii) still exist.

(3) *Regression on the cumulative distribution:* We convert the pdf $p(x)$ (that is, the scatter plot) into a *cumulative distribution* $F(x)$:

$$F(x) = P(X \geq x) = \sum_{z=x}^{\infty} p(z) = \sum_{z=x}^{\infty} Az^{-\gamma} \tag{3}$$

The approach avoids the loss of data due to averaging inside a histogram bin. To see how the plot of $F(x)$ versus x will look like, we can bound $F(x)$:

$$\begin{aligned} & \int_x^{\infty} Az^{-\gamma} dz < F(x) < Ax^{-\gamma} + \int_x^{\infty} Az^{-\gamma} dz \\ \Rightarrow\ & \frac{A}{\gamma - 1} x^{-(\gamma-1)} < F(x) < Ax^{-\gamma} + \frac{A}{\gamma - 1} x^{-(\gamma-1)} \\ \Rightarrow\ & F(x) \sim x^{-(\gamma-1)} \end{aligned} \tag{4}$$

Thus, the cumulative distribution follows a power law with exponent $(\gamma - 1)$. However, successive points on the cumulative distribution plot are not mutually independent, and this can cause problems in fitting the data.

(4) *Maximum-Likelihood Estimator (MLE):* This chooses a value of the power law exponent γ such that the likelihood that the data came from the corresponding power law distribution is maximized. Goldstein et al [2004] find that it gives good unbiased estimates of γ.

(5) *The Hill statistic:* Hill [1975] gives an easily computable estimator, that seems to give reliable results [Newman 2005]. However, it also needs to be told where the tail of the distribution begins.

(6) *Fitting only to extreme-value data:* Feuerverger and Hall [1999] propose another estimator which is claimed to reduce bias compared to the Hill statistic without significantly increasing variance. Again, the user must provide an estimate of where the tail begins, but the authors claim that their method is robust against different choices for this value.

(7) *Non-parametric estimators:* Crovella and Taqqu [1999] propose a non-parametric method for estimating the power law exponent without requiring an estimate of the beginning of the power law tail. While there are no theoretical results on the variance or bias of this estimator, the authors empirically find that accuracy increases with increasing dataset size, and that it is comparable to the Hill statistic.

Checking for goodness of fit: The correlation coefficient has typically been used as an informal measure of the goodness of fit of the degree distribution to a power law. Recently, there has been some work on developing statistical "hypothesis testing" methods to do this more formally. Beirlant et al. [2005] derive a bias-corrected Jackson statistic for measuring goodness of fit of the data to a generalized Pareto distribution. Goldstein et al. [2004] propose a Kolmogorov-Smirnov test to

determine the fit. Such measures need to be used more often in the empirical studies of graph datasets.

Examples of power laws in the real world: Examples of power law degree distributions include the Internet AS[1] graph with exponent $2.1 - 2.2$ [Faloutsos et al. 1999], the Internet router graph with exponent ~ 2.48 [Faloutsos et al. 1999; Govindan and Tangmunarunkit 2000], the in-degree and out-degree distributions of subsets of the WWW with exponents 2.1 and $2.38 - 2.72$ respectively [Barabási and Albert 1999; Kumar et al. 1999; Broder et al. 2000], the in-degree distribution of the African web graph with exponent 1.92 [Boldi et al. 2002], a citation graph with exponent 3 [Redner 1998], distributions of website sizes and traffic [Adamic and Huberman 2001], and many others. Newman [2005] provides a comprehensive list of such work.

2.2 Deviations from Power Laws

Informal description: While power laws appear in a large number of graphs, deviations from a pure power law are sometimes observed. We discuss these below.

Detailed description: Pennock et al. [2002] and others have observed deviations from a pure power law distribution in several datasets. Two of the more common deviations are exponential cutoffs and lognormals.

2.2.1 *Exponential cutoffs.* Sometimes, the distribution looks like a power law over the lower range of values along the x-axis, but decays very fast for higher values. Often, this decay is exponential, and this is usually called an exponential cutoff:

$$y(x = k) \propto e^{-k/\kappa} k^{-\gamma} \tag{5}$$

where $e^{-k/\kappa}$ is the exponential cutoff term and $k^{-\gamma}$ is the power law term. Amaral et al. [2000] find such behaviors in the electric power-grid graph of Southern California and the network of airports, the vertices being airports and the links being non-stop connections between them. They offer two possible explanations for the existence of such cutoffs. One, high-degree nodes might have taken a long time to acquire all their edges and now might be "aged", and this might lead them to attract fewer new edges (for example, older actors might act in fewer movies). Two, high-degree nodes might end up reaching their "capacity" to handle new edges; this might be the case for airports where airlines prefer a small number of high-degree hubs for economic reasons, but are constrained by limited airport capacity.

2.2.2 *Lognormals or the "DGX" distribution.* Pennock et al. [2002] recently found while the whole WWW does exhibit power law degree distributions, subsets of the WWW (such as university homepages and newspaper homepages) deviate significantly. They observed unimodal distributions on the log-log scale. Similar distributions were studied by Bi et al. [2001], who found that a discrete truncated lognormal (called the Discrete Gaussian Exponential or "DGX" by the authors)

[1]Autonomous System, typically consisting of many routers administered by the same entity.

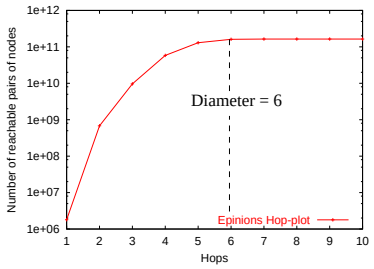


Fig. 2. *Hop-plot and effective diameter:* This is the hop-plot of the *Epinions* graph [Domingos and Richardson 2001; Chakrabarti et al. 2004]. We see that the number of reachable pairs of nodes flattens out at around 6 hops; thus the effective diameter of this graph is 6.

gives a very good fit. A lognormal is a distribution whose logarithm is a Gaussian; it looks like a truncated parabola in log-log scales. The DGX distribution extends the lognormal to discrete distributions (which is what we get in degree distributions), and can be expressed by the formula:

$$y(x = k) = \frac{A(\mu, \sigma)}{k} \exp\left[-\frac{(\ln k - \mu)^2}{2\sigma^2}\right] \quad k = 1, 2, \ldots \tag{6}$$

where μ and σ are parameters and $A(\mu, \sigma)$ is a constant (used for normalization if $y(x)$ is a probability distribution). The DGX distribution has been used to fit the degree distribution of a bipartite "clickstream" graph linking websites and users (Figure 1(c)), telecommunications and other data.

Examples of deviations from power laws in the real world: Several datasets have shown deviations from a pure power law [Amaral et al. 2000; Pennock et al. 2002; Bi et al. 2001; Mitzenmacher 2001]: examples include the electric power-grid of Southern California, the network of airports, several topic-based subsets of the WWW, Web "clickstream" data, sales data in retail chains, file size distributions, and phone usage data.

2.3 Small Diameters

Informal description: Travers and Milgram [1969] conducted a famous experiment where participants were asked to reach a randomly assigned target individual by sending a chain letter. They found that for all the chains that completed, the average length of such chains was six, which is a very small number considering the large population the participants and targets were chosen from. This leads us to believe in the concept of "six degrees of separation": the diameter of a graph is an attempt to capture exactly this.

Detailed description: Several (often related) terms have been used to describe the idea of the "diameter" of a graph:

—*Expansion and the "hop-plot"*: Tangmunarunkit et al. [2001] use a well-known metric from theoretical computer science called "expansion," which measures

the rate of increase of neighborhood with increasing h. This has been called the "hop-plot" elsewhere [Faloutsos et al. 1999].

DEFINITION 2.3 HOP-PLOT. *Starting from a node u in the graph, we find the number of nodes $N_h(u)$ in a neighborhood of h hops. We repeat this starting from each node in the graph, and sum the results to find the total neighborhood size N_h for h hops ($N_h = \sum_u N_h(u)$). The hop-plot is just the plot of N_h versus h.*

—*Effective diameter or Eccentricity*: The hop-plot can be used to calculate the *effective diameter* (also called the *eccentricity*) of the graph.

DEFINITION 2.4 EFFECTIVE DIAMETER. *This is the minimum number of hops in which some fraction (say, 90%) of all connected pairs of nodes can reach each other [Tauro et al. 2001].*

Figure 2 shows the hop-plot and effective diameter of an example graph.

—*Characteristic path length*: For each node in the graph, consider the shortest paths from it to every other node in the graph. Take the average length of all these paths. Now, consider the average path lengths for *all* possible starting nodes, and take their median. This is the characteristic path length [Bu and Towsley 2002].

—*Average diameter*: This is calculated in the same way as the characteristic path length, except that we take the mean of the average shortest path lengths over all nodes, instead of the median.

While the use of "expansion" as a metric is somewhat vague (Tangmunarunkit et al. [2001] use it only to differentiate between exponential and sub-exponential growth), most of the other metrics are quite similar. The advantage of eccentricity is that its definition works, as is, even for disconnected graphs, whereas we must consider only the largest component for the characteristic and average diameters. Characteristic path length and eccentricity are less vulnerable to outliers than average diameter, but average diameter might be the better if we want worst case analysis.

A concept related to the hop-plot is that of the *hop-exponent*: Faloutsos et al. [1999] conjecture that for many graphs, the neighborhood size N_h grows exponentially with the number of hops h. In other words, $N_h = ch^{\mathcal{H}}$ for h much less than the diameter of the graph. They call the constant $\mathcal{H}$ the hop-exponent. However, the diameter is so small for many graphs that there are too few points in the hop-plot for this premise to be verified and to calculate the hop-exponent with any accuracy.

Computation issues: One major problem with finding the diameter is the computational cost: all the definitions essentially require computing the "neighborhood size" of each node in the graph. One approach is to use repeated matrix multiplications on the adjacency matrix of the graph; however, this takes asymptotically $O(N^{2.88})$ time and $O(N^2)$ memory space. Another technique is to do breadth-first searches from each node of the graph. This takes $O(N + E)$ space but requires $O(NE)$ time. Another issue with breadth-first search is that edges are not accessed sequentially, which can lead to terrible performance on disk-resident graphs. Palmer et al. [2002] find that randomized breadth-first search algorithms are also ill-suited for large graphs, and they provide a randomized algorithm for finding the

hop-plot which takes $O((N + E)d)$ time and $O(N)$ space (apart from the storage for the graph itself), where N is the number of nodes, E the number of edges and d the diameter of the graph (typically very small). Their algorithm offers provable bounds on the quality of the approximated result, and requires only sequential scans over the data. They find the technique to be far faster than exact computation, and providing much better estimates than other schemes like sampling.

Examples in the real world: The diameters of several naturally occurring graphs have been calculated, and in almost all cases they are very small compared to the graph size. Faloutsos et al. [1999] find an effective diameter of around 4 for the Internet AS level graph and around 12 for the Router level graph. Govindan and Tangmunarunkit [2000] find a 97%-effective diameter of around 15 for the Internet Router graph. Broder et al. [2000] find that the average path length in the WWW (when a path exists at all) is about 16 if we consider the directions of links, and around 7 if all edges are considered to be undirected. Albert et al. [1999] find the average diameter of the webpages in the `nd.edu` domain to be 11.2. Watts and Strogatz [1998] find the average diameters of the power grid and the network of actors to be 18.7 and 3.65 respectively. Many other such examples can be found in the literature; Tables 1 and 2 of [Albert and Barabási 2002] and table 3.1 of [Newman 2003] list some such work.

2.4 "Community" Structure

A community is generally considered to be a set of nodes where each node is "closer" to the other nodes within the community than to nodes outside it. This effect has been found (or is believed to exist) in many real-world graphs, especially social networks: Moody [2001] finds groupings based on race and age in a network of friendships in one American school, Schwartz and Wood [1993] group people with shared interests from email logs, Borgs et al. [2004] find communities from "cross-posts" on Usenet, and Flake et al. [2000] discover communities of webpages in the WWW.

We will divide the following discussion into two parts. First, we will describe the *clustering coefficient*, which is one particular measure of community structure that has been widely used in the literature. Next, we will look at methods for *extracting* community structure from large graphs.

2.4.1 *Clustering Coefficient.*

Informal description: The clustering coefficient measures the "clumpiness" of a graph, and has relatively high values in many graphs.

Detailed description: We will first define the clustering coefficient for one node, following [Watts and Strogatz 1998] and [Newman 2003]:

Definition 2.5 Clustering Coefficient. *Suppose a node i has k_i neighbors, and there are n_i edges between the neighbors. Then the clustering coefficient of node i is defined as*

$$C_i = \begin{cases} \frac{n_i}{k_i} & k_i > 1 \\ 0 & k_i = 0 \text{ or } 1 \end{cases} \tag{7}$$

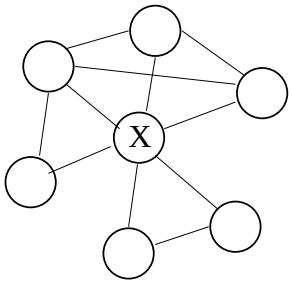


Fig. 3. *Clustering coefficient:* Node X has $k_X = 6$ neighbors. There are only $n_X = 5$ edges between the neighbors. So, the local clustering coefficient of node X is $n_X/k_X = 5/15 = 1/3$.

Thus, it measures the degree of "transitivity" of a graph; higher values imply that "friends of friends" are themselves likely to be friends, leading to a "clumpy" structure of the graph. See Figure 3 for an example.

For the clustering coefficient of a *graph* (the *global* clustering coefficient), there are two definitions:

(1) Transitivity occurs iff *triangles* exist in the graph. This can be used to measure the global clustering coefficient as

$$C = \frac{3 \times \text{number of triangles in the graph}}{\text{number of connected triples in the graph}} \tag{8}$$

where a "connected triple" is a triple of nodes consisting of a central node connected to the other two; the flanking nodes are unordered. The equation counts the fraction of connected triples which actually form triangles; the factor of three is due to the fact that each triangle corresponds to three triples.

(2) Alternatively, Watts and Strogatz [1998] use equation 7 to define to a *global* clustering coefficient for the graph as

$$C = \sum_{i=1}^{N} C_i/N \tag{9}$$

The second definition leads to very high variance in the clustering coefficients of low-degree nodes (for example, a degree 2 node can only have $C_i = 0$ or 1). The results given by the definitions can actually be quite different. The first definition is usually easier to handle analytically, while the second one has been used extensively in numerical studies.

Computation of the clustering coefficient: Alon et al. [1997] describe a deterministic algorithm for counting the number of triangles in a graph. Their method takes $O(N^\omega)$ time, where $\omega < 2.376$ is the exponent of matrix multiplication (that is, matrix multiplication takes $O(N^\omega)$ time). However, this is more than quadratic in the number of nodes, and might be too slow for large graphs. Bar-Yossef et al. [2002] describe algorithms to count triangles when the graph is in *streaming* format, that is, the data is a stream of edges which can be read only sequentially and only once. The advantage of streaming algorithms is that they

require only one pass over the data, and so are very fast; however, they typically require some temporary storage, and the aim of such algorithms is to minimize this space requirement. They find an $O(\log N)$-space randomized algorithm for the case when the edges are sorted on the source node. They also show that if there is no ordering on the edges, it is impossible to count the number of triangles using $o(N^2)$ space.

Clustering coefficients in the real world: The interesting fact about the clustering coefficient is that it is almost always larger in real-world graphs than in a random graph with the same number of nodes and edges (random graphs are discussed later; basically these are graphs where there are no biases towards any nodes). Watts and Strogatz [1998] find a clustering coefficient of 0.79 for the actor network (two actors are linked if they have acted in the same movie) whereas the corresponding random graph has a coefficient of 0.00027. Similarly, for the power grid network, the coefficient is 0.08, much greater than 0.005 for the random graph.

Extension of the clustering coefficient idea: While the global clustering coefficient gives an indication of the overall "clumpiness" of the graph, it is still just one number describing the entire graph. We can look at the clustering coefficients at a finer level of granularity by finding the average clustering coefficient $C(k)$ for all nodes with a particular degree k. Dorogovtsev et al. [2002] find that for scale-free graphs generated in a particular fashion, $C(k) \propto k^{-1}$. Ravasz and Barabási [2002] investigate the plot of $C(k)$ versus k for several real-world graphs. They find that $C(k) \propto k^{-1}$ gives decent fits to the actor network, the WWW, the Internet AS level graph and others. However, for certain graphs like the Internet Router level graph and the power grid graph, $C(k)$ is independent of k. The authors propose an explanation for this phenomenon: they say that the $C(k) \propto k^{-1}$ scaling property reflects the presence of hierarchies in the graph. Both the Router and power-grid graphs have geographical constraints (it is uneconomic to lay long wires), and this presumably prevents them from having a hierarchical topology.

2.4.2 *Methods for Extracting Graph Communities.* The problem of extracting communities from a graph, or of dividing the nodes of a graph into distinct communities, has been approached from several different directions. In fact algorithms for "community extraction" have appeared in practically all fields: social network analysis, physics and computer science among others. Here, we collate this information and present the basic ideas behind some of them. The computational requirements for each method are discussed alongside the description of each method. A survey specifically looking at clustering problems from bioinformatics is provided in [Madeira and Oliveira 2004], though it focuses only on bipartite graphs.

Dendrograms: Traditionally, the sociological literature has focused on communities formed through *hierarchical clustering* [Everitt 1974]: nodes are grouped into hierarchies, which themselves get grouped into high-level hierarchies and so. The general approach is to first assign a value V_{ij} for every pair (i, j) of nodes in the graph. Note that this value is different from the *weight* of an edge; the weight is a part of the data in a weighted graph, while the value is computed based on some

property of the nodes and the graph. This property could be the distance between the nodes, or the number of node-independent paths between the two nodes (two paths are node-independent if the only nodes they share are the endpoints). Then, starting off with only the nodes in the graph (with no edges included), we add edges one by one in decreasing order of value. At any stage of this algorithm, each of the connected components corresponds to a community. Thus, each iteration of this algorithm represents a set of communities; the *dendrogram* is a tree-like structure, with the individual nodes of the graph as the leaves of the tree and the communities in each successive iteration being the internal nodes of the tree. The root node of the tree is the entire graph (with all edges included).

While such algorithms have been successful in some cases, they tend to separate fringe nodes from their "proper" communities [Girvan and Newman 2002]. Such methods are also typically costly; however, a carefully constructed variation [Clauset et al. 2004] requires only $O(Ed \log N)$ time, where E is the number of edges, N the number of nodes, and d the depth of the dendrogram.

Edge betweenness or Stress: Dendrograms build up communities from the bottom up, starting from small communities of one node each and growing them in each iteration. As against this, Girvan and Newman [2002] take the entire graph and remove edges in each iteration; the connected components in each stage are the communities. The question is: how do we choose the edges to remove? The authors remove nodes in decreasing order of their "edge-betweenness," as defined below.

Definition 2.6 Edge betweenness or Stress. *Consider all shortest paths between all pairs of nodes in a graph. The edge-betweenness or stress of an edge is the number of these shortest paths that the edge belongs to.*

The idea is that edges connecting communities should have high edge-betweenness values because they should lie on the shortest paths connecting nodes from different communities. Tyler et al. [2003] have used this algorithm to find communities in graphs representing email communication between individuals.

The edge-betweenness of all edges can be calculated by using breadth-first search in $O(NE)$ time; we must do this once for each of the E iterations, giving a total of (NE^2) time. This makes it impractical for large graphs.

Goh et al. [2002] measure the distribution of edge-betweenness, that is, the count of edges with an edge-betweenness value of v, versus v. They find a power-law in this, with an exponent of 2.2 for protein interaction networks, and 2.0 for the Internet and the WWW.

Max-flow min-cut formulation: Flake et al. [2000] define a community to be a set of nodes with more intra-community edges than inter-community edges. They formulate the community-extraction problem as a minimum cut problem in the graph; starting from some *seed* nodes which are known to belong to a community, they find the minimal-cut set of edges that disconnects the graph so that all the seed nodes fall in one connected component. This component is then used to find new seed nodes; the process is repeated till a good component is found. This component is the community corresponding to the seed nodes.

One question is the choice of the original seed nodes. The authors use the HITS

algorithm [Kleinberg 1999a], and choose the hub and authority nodes as seeds to bootstrap their algorithm. Finding these seed nodes requires finding the first eigenvectors of the adjacency matrix of the graph, and there are well-known iterative methods to approximate these [Berry 1992]. The min-cut problem takes polynomial time using the Ford-Fulkerson algorithm [Cormen et al. 1992]. Thus, the algorithm is relatively fast, and is quite successful in finding communities for several datasets.

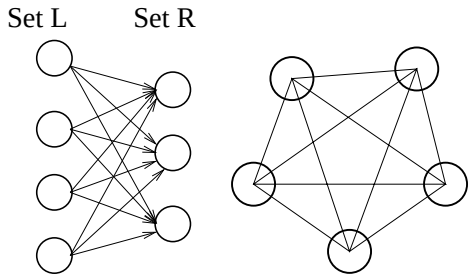
Graph partitioning: A very popular clustering technique involves graph partitioning: the graph is broken into two partitions or communities, which may then be partitioned themselves. Several different measures can be optimized for while partitioning a graph. The popular `METIS` software tries to find the best separator, minimizing the number of edges cut in order to form two disconnected components of relatively similar sizes [Karypis and Kumar 1998]. Other common measures include *coverage* (ratio of intra-cluster edges to the total number of edges) and *conductance* (ratio of inter-cluster edges to a weighted function of partition sizes) [Brandes et al. 2003]. Detailed discussions on these are beyond the scope of this work.

Several heuristics have been proposed to find good separators; *spectral clustering* is one such highly successful heuristic. This uses the first few singular vectors of the adjacency matrix or its *Laplacian* to partition the graph (the Laplacian matrix of an undirected graph is obtained by subtracting its adjacency matrix from a diagonal matrix of its vertex degrees) [Alon 1998; Spielman and Teng 1996]. Kannan et al. [2000] find that spectral heuristics give good separators in terms of both coverage and conductance. Another heuristic method called *Markov Clustering* [2000] uses random walks, the intuition being that a random walk on a dense cluster will probably not leave the cluster without visiting most of its vertices. Brandes et al. [2003] combine spectral techniques and minimum spanning trees in their `GMC` algorithm.

In general, graph partitioning algorithms are slow; for example, spectral methods taking polynomial time might still be too slow for problems on large graphs [Kannan et al. 2000]. However, Drineas et al. [1999] propose combining spectral heuristics with fast randomized techniques for singular value decomposition to combat this. Also, the *number* of communities (e.g., the number of eigenvectors to be considered in spectral clustering) often needs to be set by the user, though some recent methods try to find this automatically [Tibshirani et al. 2001; Ben-Hur and Guyon 2003].

Bipartite cores: Another definition of "community" uses the concept of *hubs* and *authorities*. According to Kleinberg [1999a], each hub node points to several authority nodes, and each authority node is pointed to by several hub nodes. Kleinberg proposes the HITS algorithm to find such hub and authority nodes. Gibson et al. [1998] use this to find communities in the WWW in following fashion. Given a user query, they use the top (say, 200) results on that query from some search engine as the seed nodes. Then they find all nodes linking to or linked from these seed nodes; this gives a subgraph of the WWW which is relevant to the user query. The HITS algorithm is applied to this subgraph, and the top 10 hub and authority nodes are together returned as the core community corresponding to the user query.

Kumar et al. [1999] remove the requirement for a user query; they use *bipartite*



(a) Bipartite core (b) Clique

Fig. 4. *Indicators of community structure:* Plot (a) shows a 4×3 bipartite core, with each node in Set L being connected to each node in Set R. Plot (b) shows a 5-node clique, where each node is connected to every other node in the clique.

cores as the seed nodes for finding communities. A bipartite core in a graph consists of two (not necessarily disjoint) sets of nodes L and R such that every node in L links to every node in R; links from R to L are optional (Figure 4). They describe an algorithm that uses successive elimination and generation phases to generate bipartite cores of larger sizes each iteration. As in [Gibson et al. 1998], these cores are extended to form communities using the HITS algorithm.

HITS requires finding the largest eigenvectors of the $A^t A$ matrix, where A is the adjacency matrix of the graph. This is a well-studied problem. The elimination and generation passes have bad worst case complexity bounds, but Kumar et al. [1999] find that it is fast in practice. They attribute this to the strongly skewed distributions in naturally occurring graphs. However, such algorithms which use hubs and authorities might have trouble finding *webrings*, where there are no clearly defined nodes of "high importance."

Local Methods: All the previous techniques used *global* information to determine clusters. This leads to scalability problems for many algorithms. Virtanen [2003] devised a clustering algorithm based solely on *local* information derived from members of a cluster. Defining a fitness metric for any cluster candidate, the author uses simulated annealing to locally find clusters which approximately maximize fitness. The advantage of this method is the online computation of locally optimal clusters (with high probability) leading to good scalability, and the absence of any "magic" parameters in the algorithm. However, the memory and disk access requirements of this method are unclear.

Cross-Associations: Recently, Chakrabarti et al. [2004] (also see [Chakrabarti 2004]) devised a scalable, parameter-free method for clustering the nodes in a graph into groups. Following the overall MDL (Minimum Description Length) principle, they define the goodness of a clustering in terms of the quality of lossless compression that can be attained using that clustering. Heuristic algorithms are used to find good clusters of nodes, and also to automatically determine the *number* of node clusters. The algorithm is linear in the number of edges E in the graph, and is thus scalable to large graphs.

Communities via Kirchoff's Laws: Wu and Huberman [2004] find the community around a given node by considering the graph as an electric circuit, with each edge having the same resistance. Now, one Volt is applied to the given node, and zero Volts to some other randomly chosen node (which will hopefully be outside the community). The voltages at all nodes are then calculated using Kirchoff's Laws, and the nodes are split into two groups by (for example) sorting all the voltages, picking the median voltage, and splitting the nodes on either side of this median into two communities. The important idea is that the voltages can be calculated approximately using iterative methods requiring only $O(N+E)$ time, but with the quality of approximation depending *only* on the number of iterations and not on the graph size.

This is a fast method, but picking the correct nodes to apply zero Volts to is a problem. The authors propose using randomized trials with repetitions, but further work is needed to prove formal results on the quality of the output.

2.5 Other Static Graph Patterns

Apart from power laws, small diameters and community effects, some other patterns have been observed in large real-world graphs. These include the resilience of such graphs to random failures, and correlations found in the *joint* degree distributions of the graphs. We will explore these below.

2.5.1 *Resilience.*

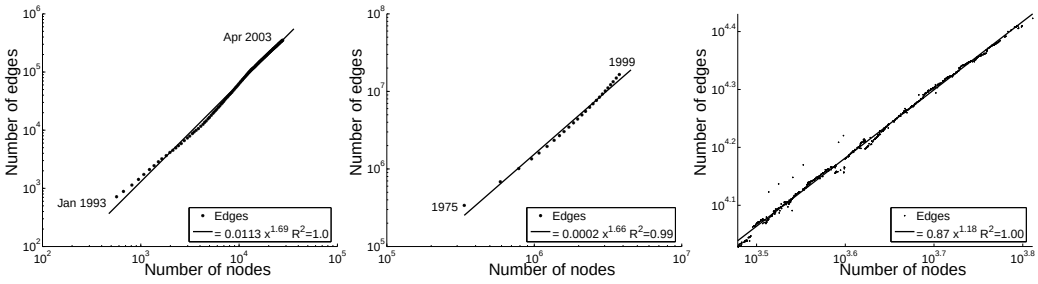
Informal description: The resilience of a graph is a measure of its robustness to node or edge failures. Many real-world graphs are resilient against random failures but vulnerable to *targeted* attacks.

Detailed description: There are at least two definitions of resilience:

—Tangmunarunkit et al. [2001] define resilience as a function of the number of nodes n: the resilience $R(n)$ is the "minimum cut-set" size within an n-node ball around any node in the graph (a ball around a node X refers to a group of nodes within some fixed number of hops from node X). The "minimum cut-set" is the minimum number of edges that need to be cut to get two disconnected components of roughly equal size; intuitively, if this value is large, then it is hard to disconnect the graph and disrupt communications between its nodes, implying higher resilience. For example, a 2D grid graph has $R(n) \propto \sqrt{n}$ while a tree has $R(n) = 1$; thus, a tree is less resilient than a grid.

—Resilience can be related to the graph diameter: a graph whose diameter does not increase much on node or edge removal has higher resilience [Palmer et al. 2002; Albert et al. 2000].

Computation issues: Calculating the "minimum cut-set" size is NP-hard, but approximate algorithms exist [Karypis and Kumar 1998]. Computing the graph diameter is also costly (see Section 2.3), but fast randomized algorithms exist [Palmer et al. 2002].



(a) arXiv (b) Patents (c) Autonomous Systems

Fig. 5. *The Densification Power Law:* The number of edges $E(t)$ is plotted against the number of nodes $N(t)$ on log-log scales for (a) the arXiv citation graph, (b) the patents citation graph, and (c) the Internet Autonomous Systems graph. All of these grow over time, and the growth follows a power law in all three cases [Leskovec et al. 2005].

Examples in the real world: In general, most real-world networks appear to be resilient against random node/edge removals, but are susceptible to targeted attacks: examples include the Internet Router-level and AS-level graphs, as well as the WWW [Palmer et al. 2002; Albert et al. 2000; Tangmunarunkit et al. 2001].

2.5.2 *Joint Distributions.* While most of the focus regarding node degrees has fallen on the in-degree and the out-degree distributions, there are "higher-order" statistics that could also be considered. We combine all these statistics under the term *joint distributions*, differentiating them from the degree-distributions which are the *marginal distributions*. Here, we note some of these statistics.

—*In and out degree correlation:* The in and out degrees might be independent, or they could be (anti)correlated. Newman et al. [2002] find a positive correlation in email networks, that is, the email addresses of individuals with large address books appear in the address books of many others. However, it is hard to measure this with good accuracy. Calculating this well would require a lot of data, and it might be still be inaccurate for high-degree nodes (which, due to power law degree distributions, are quite rare).

—*Average neighbor degree:* We can measure the average degree $d_{av}(i)$ of the neighbors of node i, and plot it against its degree $k(i)$. Pastor-Satorras et al. [2001] find that for the Internet AS level graph, this gives a power law with exponent 0.5 (that is, $d_{av}(i) \propto k(i)^{-0.5}$).

—*Neighbor degree correlation:* We could calculate the joint degree distributions of adjacent nodes; however this is again hard to measure accurately.

2.6 Patterns in Evolving Graphs

The search for graph patterns has focused primarily on static patterns, which can be extracted from one snapshot of the graph at some time instant. Many graphs, however, evolve over time (such as the Internet and the WWW) and only recently have researchers started looking for the patterns of graph evolution. Two key patterns have emerged:

—*Densification Power Law:* Leskovec et al. [2005] found that several real graphs grow over time according to a power law: the number of nodes $N(t)$ at time t is related to the number of edges $E(t)$ by the equation:

$$E(t) \propto N(t)^{\alpha} \quad 1 \leq \alpha \leq 2 \tag{10}$$

where the parameter α is called the Densification Power Law exponent, and remains stable over time. They also find that this "law" exists for several different graphs, such as paper citations, patent citations, and the Internet AS graph. This quantifies earlier empirical observations that the average degree of a graph increases over time [Barabási et al. 2002]. It also agrees with theoretical results showing that only a law like Equation 10 can maintain the power-law degree distribution of a graph as more nodes and edges get added over time [Dorogovtsev et al. 2001]. Figure 5 demonstrates the densification law for several real-world networks.

—*Shrinking Diameters:* Leskovec et al. [2005] also find that the effective diameters (definition 2.4) of graphs are actually *shrinking* over time, even though the graphs themselves are growing.

These surprising patterns are probably just the tip of the iceberg, and there may be many other patterns hidden in the dynamics of graph growth.

2.7 The Structure of Specific Graphs

While most graphs found naturally share many features (such as the small-world phenomenon), there are some specifics associated with each. These might reflect properties or constraints of the domain to which the graph belongs. We will discuss some well-known graphs and their specific features below.

2.7.1 *The Internet.* The networking community has studied the structure of the Internet for a long time. In general, it can be viewed as a collection of interconnected routing domains; each domain is a group of nodes (such routers, switches etc.) under a single technical administration [Calvert et al. 1997]. These domains can be considered as either a *stub* domain (which only carries traffic originating or terminating in one of its members) or a *transit* domain (which can carry any traffic). Example stubs include campus networks, or small interconnections of Local Area Networks (LANs). An example transit domain would be a set of backbone nodes over a large area, such as a wide-area network (WAN).

The basic idea is that stubs connect nodes locally, while transit domains interconnect the *stubs*, thus allowing the flow of traffic between nodes from different stubs (usually distant nodes). This imposes a *hierarchy* in the Internet structure, with transit domains at the top, each connecting several stub domains, each of which connects several LANs.

Apart from hierarchy, another feature of the Internet topology is its apparent *Jellyfish* structure at the AS level (Figure 6), found by Tauro et al. [2001]. This consists of:

—*A core*, consisting of the highest-degree node and the clique it belongs to; this usually has 8–13 nodes.

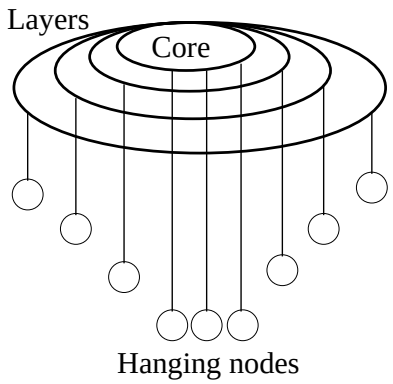


Fig. 6. *The Internet as a "Jellyfish":* The Internet AS-level graph can be thought of as a core, surrounded by concentric layers around the core. There are many one-degree nodes that hang off the core and each of the layers.

—*Layers around the core.* These are organized as concentric circles around the core; layers further from the core have lower importance.

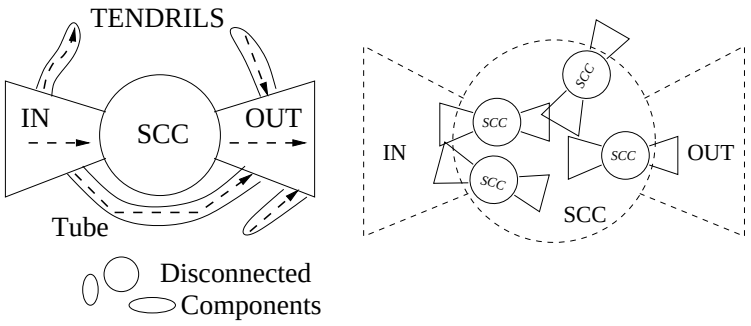
—*Hanging nodes*, representing one-degree nodes linked to nodes in the core or the outer layers. The authors find such nodes to be a large percentage (about 40–45%) of the graph.

2.7.2 *The World Wide Web (WWW).* Broder et al. [2000] find that the Web graph is described well by a "bowtie" structure (Figure 7(a)). They find that the Web can be broken in 4 approximately equal-sized pieces. The core of the bowtie is the *Strongly Connected Component* (`SCC`) of the graph: each node in the `SCC` has a directed path to any other node in the `SCC`. Then, there is the `IN` component: each node in the `IN` component has a directed path to all the nodes in the `SCC`. Similarly, there is an `OUT` component, where each node can be reached by directed paths from the `SCC`. Apart from these, there are webpages which can reach some pages in `OUT` and can be reached from pages in `IN` without going through the `SCC`; these are the `TENDRILS`. Occasionally, a tendril can connect nodes in `IN` and `OUT`; the tendril is called a `TUBE` in this case. The remainder of the webpages fall in *disconnected components*. A similar study focused on only the Chilean part of the Web graph found that the disconnected component is actually very large (nearly 50% of the graph size) [Baeza-Yates and Poblete 2003].

Dill et al. [2001] extend this view of the Web by considering subgraphs of the WWW at different scales (Figure 7(b)). These subgraphs are groups of webpages sharing some common trait, such as content or geographical location. They have several remarkable findings:

(1) *Recursive bowtie structure*: Each of these subgraphs forms a bowtie of its own. Thus, the Web graph can be thought of as a hierarchy of bowties, each representing a specific subgraph.

(2) *Ease of navigation*: The `SCC` components of all these bowties are tightly connected together via the `SCC` of the whole Web graph. This provides a naviga-



(a) The "Bowtie" structure (b) Recursive bowties

Fig. 7. *The "Bowtie" structure of the Web*: Plot (a) shows the 4 parts: `IN`, `OUT`, `SCC` and `TENDRILS` [Broder et al. 2000]. Plot (b) shows *Recursive Bowties*: subgraphs of the WWW can each be considered a bowtie. All these smaller bowties are connected by the navigational backbone of the main `SCC` of the Web [Dill et al. 2001].

tional backbone for the Web: starting from a webpage in one bowtie, we can click to its `SCC`, then go via the `SCC` of the entire Web to the destination bowtie.

(3) *Resilience*: The union of a random collection of subgraphs of the Web has a large `SCC` component, meaning that the `SCCs` of the individual subgraphs have strong connections to other `SCCs`. Thus, the Web graph is very resilient to node deletions and does not depend on the existence of large taxonomies such as `yahoo.com`; there are several alternate paths between nodes in the `SCC`.

3. GRAPH GENERATORS

Graph generators allow us to create synthetic graphs, which can then be used for, say, simulation studies. But when is such a generated graph "realistic?" This happens when the synthetic graph matches all (or at least several) of the patterns mentioned in the previous section. Graph generators can provide insight into graph creation, by telling us which processes can (or cannot) lead to the development of certain patterns.

Graph models and generators can be broadly classified into five categories (Figure 8):

(1) *Random graph models:* The graphs are generated by a random process. The basic random graph model has attracted a lot of research interest due to its phase transition properties.

(2) *Preferential attachment models:* In these models, the "rich" get "richer" as the network grows, leading to power law effects. Some of today's most popular models belong to this class.

(3) *Optimization-based models:* Here, power laws are shown to evolve when risks are minimized using limited resources. Together with the preferential attachment models, they try to provide mechanisms that automatically lead to power laws.

(4) *Geographical models:* These models consider the effects of geography (i.e., the *positions* of the nodes) on the growth and topology of the network. This is

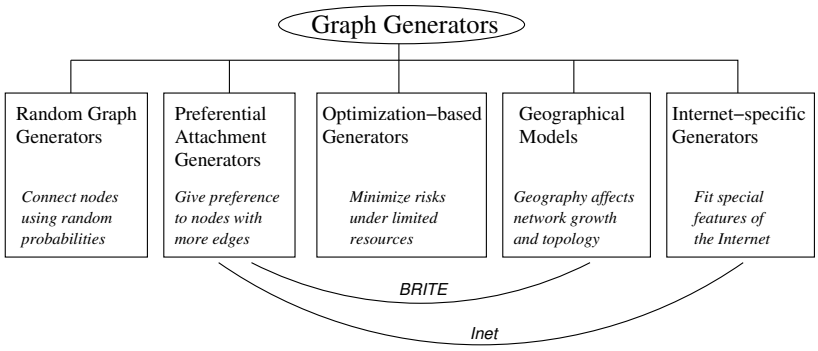


Fig. 8. *Overview of graph generators:* Current generators can be mostly placed under one of these categories, though there are some hybrids such as *BRITE* and *Inet*.

especially important for modeling router or power-grid networks, which involve laying wires between points on the globe.

(5) *Internet-specific models:* As the Internet is one of the most important graphs in computer science, special-purpose generators have been developed to model its special features. These are often hybrids, using ideas from the other categories and melding them with Internet-specific requirements.

We will discuss graph generators from each of these categories in Sections 3.1-3.5. This is not a complete list, but we believe it includes most of the key ideas from the current literature. Section 3.6 presents work on comparing these graph generators. In Section 3.7, we discuss the recently proposed R-MAT generator, which matches many of the patterns mentioned above. For each generator, we will try to provide the specific problem it aims to solve, followed by a brief description of the generator itself and its properties, and any open questions. Tables II and III provide a taxonomy of these.

3.1 Random Graph Models

Random graphs are generated by picking nodes under some random probability distribution and then connecting them by edges. We first look at the basic Erdös-Rényi model, which was the first to be studied thoroughly [Erdős and Rényi 1960], and then we discuss modern variants of the model.

3.1.1 *The Erdös-Rényi Random Graph Model.*

Problem being solved: Graph theory owes much of its origins to the pioneering work of Erdös and Rényi in the 1960s [Erdős and Rényi 1960; 1961]. Their random graph model was the first and the simplest model for generating a graph.

Description and Properties: We start with N nodes, and for every pair of nodes, an edge is added between them with probability p (as in Figure 9). This defines a *set* of graphs $G_{N,p}$, all of which have the same parameters (N, p).

Generator	Graph type						Degree distributions			
							Power law			*Exponential*
	Undir.	*Dir.*	*Bip.*	*Self loops*	*Mult. edges*	*Geog. info*	*Plain*	*Exp. cutoff*	*Deviation*	
Erdös–Rényi [1960]	√			√	√					√
PLRG [Aiello et al. 2000], PLOD [Palmer and Steffan 2000]	√			√	√		any γ (Eq. 15) (user-defined)			
Exponential cutoff [Newman et al. 2001]	√			√	√		any γ (Eq. 16) (user-defined)	√		
BA [Barabási and Albert 1999]	√						$\gamma = 3$			
Initial attractiveness [Dorogovtsev and Mendes 2003]		√		√	√		$\gamma \in [2, \infty)$ (Eq. 21)			
AB [Albert and Barabási 2000]	√			√	√		$\gamma \in [2, \infty)$ (Eq. 22)			√
Edge Copying [Kumar et al. 1999], [Kleinberg et al. 1999]		√		√	$\gamma \in (1, \infty)$		√ (Eqs. 23, 24)			
GLP [Bu and Towsley 2002]	√			√	√		$\gamma \in (2, \infty)$ (Eq. 26)			
Accelerated growth [Dorogovtsev and Mendes 2003], [Barabási et al. 2002]	√			√	√		Power-law mixture of $\gamma = 2$ and $\gamma = 3$			
Fitness model [Bianconi and Barabási 2001]	√						$\gamma = 2.255$[1]			
Aiello et al. [2001]		√					$\gamma \in [2, \infty)$ (Eq. 30)			
Pandurangan et al. [2002]		√		√	$\gamma =?$		√			
Inet-3.0 [Winick and Jamin 2002]	√						$\gamma =?$[2]	√		
Forest Fire [Leskovec et al. 2005]		√					$\gamma =?$			
Pennock et al. [2002]	√			√	√		$\gamma \in [2, \infty)$[3]		√	
Small-world [Watts and Strogatz 1998]	√					√				√
Waxman [1988]	√					√		√		
BRITE [Medina et al. 2000]	√					√	$\gamma =?$			
Yook et al. [2002]	√					√	$\gamma =?$		√	
Fabrikant et al. [2002]	√					√	$\gamma =?$			
R-MAT [Chakrabarti et al. 2004]	√	√	√	√	√		$\gamma =?$		√ (DGX)	

Table II. *Taxonomy of graph generators*: This table shows the graph types and degree distributions that different graph generators can create. The graph type can be undirected, directed, bipartite, allowing self-loops or multi-graph (multiple edges possible between nodes). The degree distributions can be power-law (with possible exponential cutoffs, or other deviations such as lognormal/DGX) or exponential decay. If it can generate a power law, the possible range of the exponent γ is provided. Empty cells indicate that the corresponding property does not occur in the corresponding model.

Generator	Diameter or Avg path len.	Community		Clustering coefficient	Remarks
		Bip. core vs size	*$C(k)$ vs k*		
Erdös–Rényi [1960]	$O(\log N)$		Indep.	Low, $CC \propto N^{-1}$	
PLRG [Aiello et al. 2000], PLOD [Palmer and Steffan 2000]	$O(\log N)$	Indep.		$CC \to 0$ for large N	
Exponential cutoff [Newman et al. 2001]	$O(\log N)$			$CC \to 0$ for large N	
BA [Barabási and Albert 1999]	$O(\log N)$ or $O(\frac{\log N}{\log \log N})$			$CC \propto N^{-0.75}$	
Initial attractiveness [Dorogovtsev and Mendes 2003]					
AB [Albert and Barabási 2000]					
Edge copying [Kleinberg et al. 1999], [Kumar et al. 1999]		Power-law			
GLP [Bu and Towsley 2002]				Higher than AB, BA, PLRG	Internet only
Accelerated growth [Dorogovtsev et al. 2001], [Barabási et al. 2002]				Non-monotonic with N	
Fitness model [Bianconi and Barabási 2001]					
Aiello et al. [2001]					
Pandurangan et al. [2002]					
Inet [Winick and Jamin 2002]					Specific to the AS graph
Forest Fire [Leskovec et al. 2005]	"shrinks" as N grows				
Pennock et al. [2002]					
Small-world [Watts and Strogatz 1998]	$O(N)$ for small N, $O(\ln N)$ for large N, depends on p			$CC(p) \propto (1-p)^3$, Indep of N	N=num nodes p=rewiring prob
Waxman [1988]					
BRITE [Medina et al. 2000]	Low (like in BA)			like in BA	BA + Waxman with additions
Yook et al. [2002]					
Fabrikant et al. [2002]					Tree, density 1
R-MAT [Chakrabarti et al. 2004]	Low (empirically)				

Table III. *Taxonomy of graph generators (Contd.)*: The comparisons are made for graph diameter, existence of community structure (number of bipartite cores versus core size, or Clustering coefficient $CC(k)$ of all nodes with degree k versus k), and clustering coefficient. N is the number of nodes in the graph. The empty cells represent information unknown to the authors, and require further research.

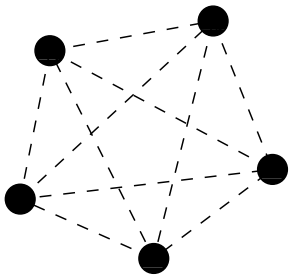


Fig. 9. *The Erdös-Rényi model:* The black circles represent the nodes of the graph. Every possible edge occurs with equal probability.

Degree Distribution: The probability of a vertex having degree k is

$$p_k = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{z^k e^{-z}}{k!} \quad \text{with } z = p(N-1) \tag{11}$$

For this reason, this model is often called the "Poisson" model.

Size of the largest component: Many properties of this model can be solved exactly in the limit of large N. A property is defined to hold for parameters (N, p) if the probability that the property holds on every graph in $G_{N,p}$ approaches 1 as $N \to \infty$. One of the most noted properties concerns the size of the largest component (subgraph) of the graph. For a low value of p, the graphs in $G_{N,p}$ have low density with few edges and all the components are small, having an exponential size distribution and finite mean size. However, with a high value of p, the graphs have a *giant component* with $O(N)$ of the nodes in the graph belonging to this component. The rest of the components again have an exponential size distribution with finite mean size. The changeover (called the *phase transition*) between these two regimes occurs at $p = \frac{1}{N}$. A heuristic argument for this is given below, and can be skipped by the reader.

Finding the phase transition point: Let the fraction of nodes not belonging to the giant component be u. Thus, the probability of random node not belonging to the giant component is also u. But the neighbors of this node also do not belong to the giant component. If there are k neighbors, then the probability of this happening is u^k. Considering all degrees k, we get

$$u \;=\; \sum_{k=0}^{\infty} p_k u^k$$

[1] $P(k) \propto k^{-2.255}/\ln k$; [Bianconi and Barabási 2001] study a special case, but other values of the exponent γ may be possible with similar models.
[2] Inet-3.0 matches the Internet AS graph very well, but formal results on the degree-distribution are not available.
[3] $\gamma = 1 + \frac{1}{\alpha}$ as $k \to \infty$ (Eq. 32)

$$= e^{-z} \sum_{k=0}^{\infty} \frac{(uz)^k}{k!} \quad \text{(using Eq 11)}$$
$$= e^{-z} e^{uz} = e^{z(u-1)} \tag{12}$$

Thus, the fraction of nodes in the giant component is

$$S = 1 - u = 1 - e^{-zS} \tag{13}$$

Equation 13 has no closed-form solutions, but we can see that when $z < 1$, the only solution is $S = 0$ (because $e^{-x} > 1 - x$ for $x \in (0, 1)$). When $z > 1$, we can have a solution for S, and this is the size of the giant component. The phase transition occurs at $z = p(N-1) = 1$. Thus, a giant component appears only when p scales faster than N^{-1} as N increases.

Tree-shaped subgraphs: Similar results hold for the appearance of trees of different sizes in the graph. The critical probability at which almost every graph contains a subgraph of k nodes and l edges is achieved when p scales as N^z where $z = -\frac{k}{l}$ [Bollobás 1985]. Thus, for $z < -\frac{3}{2}$, almost all graphs consist of isolated nodes and edges; when z passes through $-\frac{3}{2}$, trees of order 3 suddenly appear, and so on.

Diameter: Random graphs have a diameter concentrated around $\log N / \log z$, where z is the average degree of the nodes in the graph. Thus, the diameter grows slowly as the number of nodes increases.

Clustering coefficient: The probability that any two neighbors of a node are themselves connected is the connection probability $p = \frac{<k>}{N}$, where $< k >$ is the average node degree. Therefore, the clustering coefficient is:

$$CC_{random} = p = \frac{< k >}{N} \tag{14}$$

Open questions and discussion: It is hard to exaggerate the importance of the Erdös-Rényi model in the development of modern graph theory. Even a simple graph generation method has been shown to exhibit phase transitions and criticality. Many mathematical techniques for the analysis of graph properties were first developed for the random graph model.

However, even though random graphs exhibit such interesting phenomena, they do not match real-world graphs particularly well. Their degree distribution is Poisson (as shown by Equation 11), which has a very different shape from power-laws or lognormals. There are no correlations between the degrees of adjacent nodes, nor does it show any form of "community" structure (which often shows up in real graphs like the WWW). Also, according to Equation 14, $\frac{CC_{random}}{<k>} = \frac{1}{N}$; but for many real-world graphs, $\frac{CC}{<k>}$ is independent of N (See figure 9 from [Albert and Barabási 2002]).

Thus, even though the Erdös-Rényi random graph model has proven to be very useful in the early development of this field, it is not used in most of the recent work on modeling real graphs. To address some of these issues, researchers have extended the model to the so-called Generalized Random Graph Models, where the

degree distribution can be set by the user (typically, set to be a power law).

3.1.2 *Generalized Random Graph Models.*

Problem being solved: Erdös-Rényi graphs result in a Poisson degree distribution, which often conflicts with the degree distributions of many real-world graphs. Generalized random graph models extend the basic random graph model to allow arbitrary degree distributions.

Description and properties: Given a degree distribution, we can randomly assign a degree to each node of the graph so as to match the given distribution. Edges are formed by randomly linking two nodes till no node has extra degrees left. We describe two different models below: the PLRG model and the Exponential Cutoffs model. These differ only in the degree distributions used; the rest of the graph-generation process remains the same. The graphs thus created can, in general, include self-graphs and multigraphs (having multiple edges between two nodes).

The PLRG model: One of the obvious modifications to the Erdös-Rényi model is to change the degree distribution from Poisson to power-law. One such model is the Power-Law Random Graph (PLRG) model of Aiello et al. [2000] (a similar model is the *Power Law Out Degree* (PLOD) model of Palmer and Steffan [2000]). There are two parameters: α and β. The number of nodes of degree k is given by e^{α}/k^{β}.

PLRG degree distribution: By construction, the degree distribution is specifically a power law:

$$p_k \propto k^{-\beta} \tag{15}$$

where β is the power-law exponent.

PLRG connected component sizes: The authors show that graphs generated by this model can have several possible properties, based only on the value of β. When $\beta < 1$, the graph is almost surely connected. For $1 < \beta < 2$, a giant component exists, and smaller components are of size $O(1)$. For $2 < \beta < \beta_0 \sim 3.48$, the giant component exists and the smaller components are of size $O(\log N)$. At $\beta = \beta_0$, the smaller components are of size $O(\log N/\log\log N)$. For $\beta > \beta_0$, no giant component exists. Thus, for the giant component, we have a *phase transition* at $\beta = \beta_0 = 3.48$; there is also a change in the size of the smaller components at $\beta = 2$.

The Exponential cutoffs model: Another generalized random graph model is due to Newman et al. [2001]. Here, the probability that a node has k edges is given by

$$p_k = Ck^{-\gamma}e^{-k/\kappa} \tag{16}$$

where C, γ and κ are constants.

Exponential cutoffs degree distribution: This model has a power law (the $k^{-\gamma}$ term) augmented by an exponential cutoff (the $e^{-k/\kappa}$ term). The exponential cutoff, which is believed to be present in some social and biological networks, reduces the

heavy-tail behavior of a pure power-law degree distribution. The results of this model agree with those of [Aiello et al. 2000] when $\kappa \to \infty$.

Average path length for exponential cutoffs: Analytic expressions are known for the average path length of this model, but this typically tends to be somewhat less than that in real-world graphs [Albert and Barabási 2002].

Apart from PLRG and the exponential cutoffs model, some other related models have also been proposed. One important model is that of Aiello et al. [2001], who assign weights to nodes and then form edges probabilistically based on the product of the weights of their end-points. The exact mechanics are, however, close to preferential attachment, and we discuss this later in Section 3.2.8.

Similar models have also been proposed for generating directed and bipartite random graphs. Recent work has provided analytical results for the sizes of the strongly connected components and cycles in such graphs [Cooper and Frieze 2004; Dorogovtsev et al. 2001]. We do not discuss these any further; the interested reader is referred to [Newman et al. 2001].

Open questions and discussion: Generalized random graph models retain the simplicity and ease of analysis of the Erdös-Rényi model, while removing one of its weaknesses: the unrealistic Poisson degree distribution. However, most such models only attempt to match the degree distribution of real graphs, and no other patterns. For example, in most random graph models, the probability that two neighbors of a node are themselves connected goes as $O(N^{-1})$. This is exactly the clustering coefficient of the graph, and goes to zero for large N; but for many real-world graphs, $\frac{CC}{<k>}$ is independent of N (See figure 9 from [Albert and Barabási 2002]). Also, many real world graphs (such as the WWW) exhibit the existence of communities of nodes, with stronger ties within the community than outside (see Section 2.4.2); random graphs do not appear to show any such behavior. Further work is needed to accommodate these patterns into the random graph generation process.

3.2 Preferential Attachment and Variants

Problem being solved: Generalized random graph models try to model the power law or other degree distribution of real graphs. However, they do not make any statement about the *processes* generating the network. The search for a mechanism for network generation was a major factor in fueling the growth of the preferential attachment models, which we discuss below.

The rest of this section is organized as follows: in section 3.2.1, we describe the basic preferential attachment process. This has proven very successful in explaining many features of real-world graphs. Sections 3.2.3-3.2.11 describe progress on modifying the basic model to make it even more precise.

3.2.1 *Basic Preferential Attachment.* In the mid-1950s, Herbert Simon [1955] showed that power law tails arise when "the rich get richer." Derek Price applied this idea (which he called *cumulative advantage*) to the case of networks [de Solla Price 1976], as follows. We grow a network by adding vertices over time. Each vertex

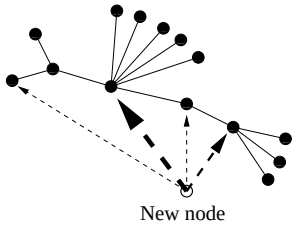


Fig. 10. *The Barabási-Albert model:* New nodes are added; each new node prefers to connect to existing nodes of high degree. The dashed lines show some possible edges for the new node, with thicker lines implying higher probability.

gets a certain out-degree, which may be different for different vertices but whose mean remains at a constant value m over time. Each outgoing edge from the new vertex connects to an old vertex with a probability proportional to the in-degree of the old vertex. This, however, leads to a problem since all nodes initially start off with in-degree zero. Price corrected this by adding a constant to the current in-degree of a node in the probability term, to get

$$P(\text{edge to existing vertex } v) = \frac{k(v) + k_0}{\sum_i (k(i) + k_0)} \tag{17}$$

where $k(i)$ represents the current in-degree of an existing node i, and k_0 is a constant.

A similar model was proposed by Barabási and Albert [1999]. It has been a very influential model, and formed the basis for a large body of further work. Hence, we will look at the Barabási-Albert model (henceforth called the BA model) in detail.

Description of the BA model: The BA model proposes that structure emerges in network topologies as the result of two processes:

(1) *Growth*: Contrary to several other existing models (such as random graph models) which keep a fixed number of nodes during the process of network formation, the BA model starts off with a small set of nodes and *grows* the network as nodes and edges are added over time.
(2) *Preferential Attachment*: This is the same as the "rich get richer" idea. The probability of connecting to a node is proportional to the current degree of that node.

Using these principles, the BA model generates an *undirected* network as follows. The network starts with m_0 nodes, and grows in stages. In each stage, one node is added along with m edges which link the new node to m existing nodes (Figure 10). The probability of choosing an existing node as an endpoint for these edges is given by

$$P(\text{edge to existing vertex } v) = \frac{k(v)}{\sum_i k(i)} \tag{18}$$

where $k(i)$ is the degree of node i. Note that since the generated network is undirected, we do not need to distinguish between out-degrees and in-degrees. The

effect of this equation is that nodes which already have more edges connecting to them, get even more edges. This represents the "rich get richer" scenario.

There are a few differences from Price's model. One is that the number of edges per new node is fixed at m (a positive integer); in Price's model only the mean number of added edges needed to be m. However, the major difference is that while Price's model generates a directed network, the BA model is undirected. This avoids the problem of the initial in-degree of nodes being zero; however, many real graphs are directed, and the BA model fails to model this important feature.

Properties of the BA model: We will now discuss some of the known properties of the BA model. These include the degree distribution, diameter, and correlations hidden in the model.

Degree distribution: The degree distribution of the BA model [Dorogovtsev et al. 2000] is given by:

$$p_k \approx k^{-3} \tag{19}$$

for large k. In other words, the degree distribution has a power law "tail" with exponent 3, independent of the value of m.

Diameter: Bollobás and Riordan [2002] show that for large N, the diameter grows as $O(\log N)$ for $m = 1$, and as $O(\log N / \log\log N)$ for $m \geq 2$. Thus, this model displays the *small-world* effect: the distance between two nodes is, on average, far less than the total number of nodes in the graph.

Correlations between variables: Krapivsky and Redner [2001] find two correlations in the BA model. First, they find that degree and age are positively correlated: older nodes have higher mean degree. The second correlation is in the degrees of neighboring nodes, so that nodes with similar degree are more likely to be connected. However, this asymptotically goes to 0 as $N \to \infty$.

Open questions and discussion: The twin ideas of *growth* and *preferential attachment* are definitely an immense contribution to the understanding of network generation processes. However, the BA model attempts to explain graph structure using *only* these two factors; most real-world graphs are probably generated by a slew of different factors. The price for this is some inflexibility in graph properties of the BA model.

—The power-law exponent of the degree distribution is fixed at $\gamma = 3$, and many real-world graphs deviate from this value.

—The BA model generates undirected graphs only; this prevents the model from being used for the many naturally occurring directed graphs.

—While Krapivsky and Redner show that the BA model should have correlations between node degree and node age (discussed above), Adamic and Huberman [2000] apparently find no such correlations in the WWW.

—The generated graphs have exactly one connected component. However, many real graphs have several isolated components. For example, websites for compa-

nies often have private set of webpages for employees/projects only. These are a part of the WWW, but there are no paths to those webpages from outside the set. Military routers in the Internet router topology are another example.

—The BA model has a constant average degree of m; however, the average degree of some graphs (such as citation networks) actually increases over time according to a Densification Power Law [Barabási et al. 2002; Leskovec et al. 2005; Dorogovtsev et al. 2001] (see Section 2.6).

—The diameter of the BA model increases as N increases; however, many graphs exhibit shrinking diameters (see Section 2.6).

Also, further work is needed to confirm the existence or absence of a community structure in the generated graphs.

While the basic BA model does have these limitations, its simplicity and power make it an excellent base on which to build extended models. In fact, the bulk of graph generators in use today can probably trace their lineage back to this model. In the next few sections, we will look at some of these extensions and variations; as we will see, most of these are aimed at removing one or the other of the aforementioned limitations.

3.2.2 *Initial attractiveness.*

Problem being solved: While the BA model generates graphs with a power law degree distribution, the power law exponent is stuck at $\gamma = 3$. Dorogovtsev et al. [2000; 2003] propose a simple one-parameter extension of the basic model which allows $\gamma \in [2, \infty)$. Other methods, such as the AB model described later, also do this, but they require more parameters.

Description and properties: The BA model is modified by adding an extra parameter $A \geq 0$ as follows:

$$P(\text{edge to existing vertex } v) = \frac{A + k(v)}{\sum_i (A + k(i))} \tag{20}$$

where $k(i)$ is the degree of node i. The parameter A models the "initial attractiveness" of each site, and governs the probability of "young" sites gaining new edges. Note that the BA model is a special case of this model (when $A = 0$).

Degree distribution: The degree distribution is found to be a power law with exponent

$$\gamma = 2 + \frac{A}{m}$$

where m is the number of new edges being added at each timestep. Thus, depending on the value of A and m, $\gamma \in [2, \infty)$.

Open questions and discussion: This model adds a lot of flexibility to the BA model while requiring just a single parameter. As an extension of this, we could consider assigning different "initial attractiveness" values to different nodes; for example, this might be more realistic for new websites coming online on the WWW. Some progress has been made by Barabási and Bianconi [2001], but their

"fitness" parameters are used differently, and it is an open question what would happen if the parameter A in equation 20 were to be replaced by A_v.

3.2.3 *Internal edges and Rewiring.*

Problem being solved: Graphs generated by the BA model have degree distributions with a power-law exponent of 3. However, the value of this exponent is often different for many naturally occurring graphs. The model described below attempts to remedy this.

Description and properties: In the BA model, one node and m edges are added to the graph every iteration. Albert and Barabási [2000] decouple this addition of nodes and edges, and also extend the model by introducing the concept of edge rewiring. Starting with a small set of m_0 nodes, the resulting model (henceforth called the AB model) combines 3 processes:

— *With probability p, add m ($m \leq m_0$) new edges*: For each edge, one endpoint is chosen at random, and the other endpoint is chosen with probability

$$p(v) = \frac{k(v)+1}{\sum_i (k(i)+1)} \tag{21}$$

where $p(v)$ represents the probability of node v being the endpoint, and $k(i)$ representing the degree of node i.

— *With probability q, rewire m links*: Choose a node i at random, and then choose one of its edges e_{ij}, remove it, and reconnect node i to some other node chosen using preferential attachment (Equation 21). This whole process is then repeated m times. This is effectively a way to locally reshuffle connections.

— *With probability $1-p-q$, add a new node with m edges*: One end of these m edges is the new node; the other ends are chosen using preferential attachment (Equation 21). This was the only step in the BA model.

Note that in general, graphs generated by the AB model might have self-loops and multiple edges between two nodes.

Degree distribution: This model exhibits either a power-law or exponential degree distribution, depending on the parameters used. When $q < q_{max} = min(1-p, (1-p+m)/(1+2m))$, the distribution is a power law with exponent γ given by

$$\gamma = \frac{2m(1-q)+1-p-q}{m} + 1 \tag{22}$$

However, for $q > q_{max}$, the distribution becomes exponential.

Validity of the model for the Internet graph: Chen et al. [2001] try to check the validity of these processes in the context of the Internet. Their findings are summarized below:

— *Incremental Growth*: The Internet AS graph does grow incrementally, with nodes and edges being added gradually over time.

—*Linear Preferential Attachment*: However, they find that new ASes have a much stronger preference for connecting to the high-degree ASes than predicted by linear preferential attachment.

—*Addition of Internal Edges*: They also consider the addition of new edges between pre-existing ASes; this corresponds to the creation of new internal edges in the AB model. For the addition of every new internal edge, they put the end vertex with the smaller degree in a "Small Vertex" class, and the other end vertex in the "Large Vertex" class. They compare the degree distributions of these classes to that from the AS graph and find that while the "Small Vertex" class matches the real graph pretty well, the distribution of the "Large Vertex" class is very different between the AB model and the Internet.

—*Edge Rewiring*: They find that rewiring is probably not a factor in the evolution of the Internet.

Open questions and discussion: The AB model provides flexibility in the power law exponent of the degree distribution. Further research is needed to show the presence or absence of a "community" structure in the generated graphs. Also, we are unaware of any work on analytically finding the diameter of graphs generated by this model.

3.2.4 *Edge Copying Models.*

Problem being solved: Several graphs show community behavior, such as topic-based communities of websites on the WWW. Kleinberg et al. [1999] and Kumar et al. [1999] try to model this by using the intuition that most webpage creators will be familiar with webpages on topics of interest to them, and so when they create new webpages, they will link to some of these existing topical webpages. Thus, most new webpages will enhance the "topical community" effect of the WWW.

Description and properties: The Kleinberg [1999] generator creates a directed graph. The model involves the following processes:

—*Node creation and deletion*: In each iteration, nodes may be independently created and deleted under some probability distribution. All edges incident on the deleted nodes are also removed.

—*Edge creation*: In each iteration, we choose some node v and some number of edges k to add to node v. With probability β, these k edges are linked to nodes chosen uniformly and independently at random. With probability $1 - \beta$, edges are *copied* from another node: we choose a node u at random, choose k of its edges (u, w), and create edges (v, w) (as shown in Figure 11). If the chosen node u does not have enough edges, all its edges are copied and the remaining edges are copied from another randomly chosen node.

—*Edge deletion*: Random edges can be picked and deleted according to some probability distribution.

This is similar to preferential attachment because the pages with high-degree will be linked to by many other pages, and so have a greater chance of getting copied.

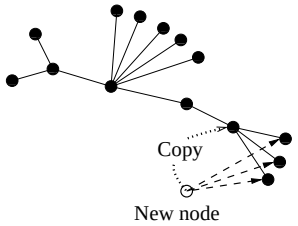


Fig. 11. *The edge copying model:* New nodes can choose to copy the edges of an existing node. This models the copying of links from other peoples' websites to create a new website.

Kumar et al. [1999] propose a very similar model. However, there are some important differences. Whenever a new node is added, only *one* new edge is added. The edge need not be incident on the new node. With a probability α, the tail of the new edge (recall that this is a directed graph; the edge points from head to tail) is the new node, otherwise it is the tail of some randomly chosen existing edge. Similarly, with a probability β, the head of the new edge is the new node, otherwise it is the head of some random edge. Thus, the copying process takes place when head or tail of some existing edge gets chosen as the endpoint of the new edge.

Since important nodes on each "topic" might be expected to start off with a large number of edges incident on them, the edge copying process would tend to enhance the number of edges linking to them. Thus, the graph would gain several "communities", with nodes in the community linking to the "important" nodes of that community.

This and similar models by analyzed by Kumar et al. [2000]. They found the following interesting properties.

Degree distributions: For the Kleinberg model [1999], the in-degree distribution is a power law with exponent given by

$$\gamma_{in} = \frac{1}{1-\beta} \tag{23}$$

For the model of Kumar et al. [1999], both the in-degree and out-degree distributions follow power laws

$$\begin{aligned} \gamma_{in} &= \frac{1}{1-\alpha} \\ \gamma_{out} &= \frac{1}{1-\beta} \end{aligned} \tag{24}$$

"Community" effect: They also show that such graphs can be expected to have a large number of bipartite cores (which leads to the community effect). However, more experiments might be needed to conclusively prove these results.

Open questions and discussion: The Kleinberg model [1999] generates a tree; no "back-edges" are formed from the old nodes to the new nodes. Also, in the model of Kumar et al. [1999], a fixed fraction of the nodes have zero in-degree or zero out-

degree; this might not be the case for all real-world graphs (see Aiello et al. [2001] for related issues).

However, the simple idea of copying edges can clearly lead to both power laws as well as community effects. "Edge copying" models are, thus, a very promising direction for future research.

3.2.5 *Modifying the preferential attachment equation.*

Problem being solved: Chen et al. [2001] had found the AB model somewhat lacking in modeling the Web (Section 3.2.3). Specifically, they found that the preference for connecting to high-degree nodes is stronger than that predicted by linear preferential attachment. Bu and Towsley [2002] attempt to address this issue.

Description and properties: The AB model [Albert and Barabási 2000] is changed by removing the edge rewiring process, and modifying the linear preferential attachment equation of the AB model to show higher preference for nodes with high degrees (as in [Chen et al. 2001]). Their new preferential attachment equation is:

$$p(v) = \frac{k(v) - \beta}{\sum_i (k(i) - \beta)} \tag{25}$$

where $p(v)$ represents the probability of node v being the endpoint, $k(i)$ representing the degree of node i, and $\beta \in (-\infty, 1)$ is a tunable parameter. The smaller the value of β, the less preference is given to nodes with higher degree. Since $\beta < 1$, any node of degree 1 has a non-zero probability of acquiring new edges. This is called the GLP (Generalized Linear Preference) model.

Degree distribution: The degree distribution follows a power law with exponent

$$\gamma = \frac{2m - \beta(1-p)}{(1+p)m} + 1 \tag{26}$$

Clustering coefficient: They also find empirically that the clustering coefficient for a GLP graph is much closer to that of the Internet than the BA, AB and Power-Law Random Graph (PLRG [Aiello et al. 2000]) models.

Bu and Towsley kept the preferential attachment equation linear (Equation 25); others such as Krapivsky and Redner [2001] have studied *non-linear* preferential attachment:

$$p(v) \propto k^{\alpha} \tag{27}$$

They achieve an important result, albeit a negative one. They find that power-law degree distributions occur only for linear attachment ($\alpha = 1$). When the preferential attachment is sublinear ($\alpha < 1$), the number of high-degree nodes decays faster than a power law. This goes against the findings of Chen et al. [2001]. In the superlinear case ($\alpha > 1$), a single "gel" node emerges, which connects to nearly all other nodes. Again, many graphs, like the Internet, do not have this property.

3.2.6 *Modeling increasing average degree.*

Problem being solved: The average degree of several real-world graphs (such as citation graphs) increases over time [Dorogovtsev et al. 2001; Barabási et al. 2002;

Leskovec et al. 2005], according to a Densification Power Law. Barabási et al. [Barabási et al. 2002] attempt to modify the basic BA model to accommodate this effect.

Description and properties: In their model, nodes join the graph at a constant rate, and form m edges to currently existing nodes with the linear preferential attachment equation (Equation 18), as in the BA model. *Also*, nodes already present in the graph form new internal edges, based on a different preferential attachment equation:

$$P(u, v) \propto k(u).k(v) \tag{28}$$

In other words, the edge chooses *both* its endpoints by preferential attachment. The number of internal nodes added per iteration is proportional to the the current number of nodes in the graph. Thus, it leads to the phenomenon of *accelerated growth*: the average degree of the graph increases linearly over time.

Degree distribution: However, the analysis of this model shows that it has two power-law regimes. The power law exponent is $\gamma = 2$ for low degrees, and $\gamma = 3$ for high degrees. In fact, over a long period of time, the exponent converges to $\gamma = 2$.

Open questions and discussion: While this model allows for increasing average degree over time, the degree distribution is constrained to a power law with fixed exponents. Also, it is unknown if this model matches the "shrinking diameter" effect observed for growing graphs (see Section 2.6).

3.2.7 *Node fitness measures.*

Problem being solved: The preferential attachment models noted above tend to have a correlation between the age of a node and its degree: higher the age, more the degree [Krapivsky and Redner 2001]. However, Adamic and Huberman find that this does not hold for the WWW [Adamic and Huberman 2000]. There are websites which were created late but still have far higher in-degree than many older websites. Bianconi and Barabási [2001] try to model this.

Description and properties: The model attaches a *fitness parameter* η_i to each node i, which does not change over time. The idea is that even a node which is added late could overtake older nodes in terms of degree, if the newer node has a much higher fitness value. The basic linear preferential attachment equation now becomes a *weighted* equation

$$P(\text{edge to existing vertex } v) = \frac{\eta_v k(v)}{\sum_i \eta_i k(i)} \tag{29}$$

Degree distribution: The authors analyze the case when the fitness parameters are drawn randomly from a uniform $[0, 1]$ distribution. The resulting degree distribution is a power law with an extra inverse logarithmic factor. For the case where all fitness values are the same, this model becomes the simple BA model.

Open questions and discussion: Having a node's popularity depend on its "fitness" intuitively makes a lot of sense. Further research is needed to determine

the distribution of node fitness values in real-world graphs. For this "fitness distribution," we also need to compute the corresponding degree distribution, and ensure that it matches reality.

3.2.8 *Generalizing preferential attachment.*

Problem being solved: The BA model is undirected. A simple adaptation to the directed case is: new edges are created to point from the new nodes to existing nodes chosen preferentially according to their *in-degree*. However, the out-degree distribution of this model would not be a power law. Aiello et al. [2001] propose a very general model for generating directed graphs which give power laws for both in-degree and out-degree distributions. A similar model was also proposed by Bollobás et al. [2003].

Description and properties: The basic idea is the following:

—Generate 4 random numbers $m(n,n)$, $m(n,e)$, $m(e,n)$ and $m(e,e)$ according to some bounded probability distributions; the numbers need not be independent.

—One node is added to the graph in each iteration.

—$m(n,n)$ edges are added from new node to new node (forming self-loops).

—$m(n,e)$ edges are added from the new node to random existing nodes, chosen preferentially according to their in-degree (higher in-degree nodes having higher chance of being chosen).

—$m(e,n)$ edges are added from existing nodes to the new node; the existing nodes are chosen randomly with probability proportional to their out-degrees.

—$m(e,e)$ edges are added between existing nodes. Again, the choices are proportional to the in-degrees and out-degrees of the nodes.

Finally, nodes with 0 in and out degrees are ignored.

Degree distributions: The authors show that even in this general case, both the in-degree and out-degree distributions follow power laws, with the following exponents:

$$\begin{aligned} \gamma_{in} &= 2 + \frac{m(n,n) + m(e,n)}{m(n,e) + m(e,e)} \\ \gamma_{out} &= 2 + \frac{m(n,n) + m(n,e)}{m(e,n) + m(e,e)} \end{aligned} \tag{30}$$

A similar result is obtained by Cooper and Frieze [2003] for a model which also allows some edge endpoints to be chosen uniformly at random, instead of always via preferential attachment.

Open questions and discussion: The work referenced above shows that even a very general version of preferential attachment can lead to power law degree distributions. Further research is needed to test for all the other graph patterns, such as diameter, community effects and so on.

3.2.9 *PageRank-based preferential attachment.*

Problem being solved: Pandurangan et al. [2002] found that the *PageRank* [Brin and Page 1998] values for a snapshot of the Web graph follow a power law. They

propose a model that tries to match this *PageRank* distribution of real-world graphs, *in addition to* the degree distributions.

Description and properties: They modify the basic preferential attachment mechanism by adding a *PageRank*-based preferential attachment component:

—With probability a, new edges preferentially connect to higher-degree nodes. This is typical preferential attachment.

—With probability b, new edges preferentially connect to nodes with high *PageRank*. According to the authors, this represents linking to nodes which are found by using a *search engine* which uses PageRank-based rankings.

—With probability $1 - a - b$, new edges connect to randomly chosen nodes.

Degree and PageRank distributions: They empirically show that this model can match both the degree distributions as well as the *PageRank* distribution of the Web graph. However, closed-form formulas for the degree distributions are not provided for this model.

Open questions and discussion: This model offers an intuitive method of incorporating the effects of Web search engines into the growth of the Web. However, the authors also found that the plain edge-copying model of Kumar et al. [1999] could *also* match the *PageRank* distribution (in addition to the degree distributions) without specifically attempting to do so. Thus, this work might be taken to be another alternative model of the Web.

3.2.10 *The Forest Fire model.*

Problem being solved: Leskovec et al. [2005] develop a preferential-attachment based model which matches the Densification Power Law and the shrinking diameter patterns of graph evolution, in addition to the power law degree distribution.

Description and properties: The model has two parameters: a *forward burning probability* p, and a *backward burning ratio* r. The graph grows one node at a time. The new node v adds links to the existing nodes according to a "forest fire" process:

(1) *Pick ambassador:* Node v chooses an *ambassador* node w uniformly at random, and links to w.

(2) *Select some of the ambassador's edges:* A random number x is chosen from a binomial distribution with mean $(1 - p)^{-1}$. Node v then selects x edges of w, both in-links and out-links, but selecting in-links with probability r times less than out-links. Let $w_1, w_2, \ldots, w_x$ be the other ends of these selected edges.

(3) *Follow these edges and repeat:* Node v forms edges pointing to each of these nodes $w_1, \ldots, w_x$, and then recursively applies step (2) to these nodes.

This process conceptually captures a "forest fire" in the existing graph; the fire starts at the ambassador node and then probabilistically spreads to the other nodes if they are connected to nodes which are currently "burning." Some nodes end up creating large "conflagrations," which forms many out-links before the fire dies out, thus resulting in power laws.

Degree distributions: Both the in-degree and out-degree distribution are empirically found to follow power laws.

Community structure: This method is similar to the edge copying model discussed earlier (section 3.2.4) because existing links are "copied" to the new node v as the fire spreads. This leads to a community of nodes, which share similar edges.

Densification Power Law and Shrinking Diameter: The Forest Fire model empirically seems to follow both of these patterns. The intuition behind densification is clear: as the graph grows, the chances of a larger fire also grow, and so new nodes have higher chances of getting more edges. However, the intuition behind the shrinking diameter effect is not clear.

Open questions and discussion: This is certainly a very interesting and intuitive model, but the authors note that rigorous analysis of this model appears to be quite difficult. The R-MAT generator (discussed later in section 3.7) and its recently proposed generalization into *Kronecker graphs* [Leskovec et al. 2005] is one possible approach that offers formal results for these graph patterns.

3.2.11 *Deviations from power laws.*

Problem being solved: Pennock et al. [2002] find that while the WWW as a whole might exhibit power-law degree distributions, subgraphs of webpages belonging to specific categories or topics often show significant deviations from a power law (see Section 2.2). They attempt to model this deviation from power-law behavior.

Description and properties: Their model is similar to the BA model (Section 3.2.1) , except for two differences:

—*Internal edges:* The m new edges added in each iteration need not be incident on the new node being added that iteration. Thus, the new edges could be *internal* edges.

—*Combining random and preferential attachment:* Instead of pure preferential attachment, the endpoints of new edges are chosen according to a linear combination of preferential attachment and uniform random attachment. The probability of a node v being chosen as one endpoint of an edge is given by:

$$p(v) = \alpha \frac{k(v)}{2mt} + (1-\alpha)\frac{1}{m_0 + t} \tag{31}$$

Here, $k(v)$ represents the current degree of node v, $2mt$ is the total number of edges at time t, $(m_0 + t)$ is the current number of nodes at time t, and $\alpha \in [0, 1]$ is a free parameter. To rephrase the equation, in order to choose a node as an endpoint for a new edge, we either do preferential attachment with probability α, or we pick a node at random with probability $(1 - \alpha)$.

One point of interest is that even if a node is added with degree 0, there is always a chance for it to gain new edges via the uniform random attachment process. The preferential attachment and uniform attachment parts of Equation 31 represent two

different behaviors of webpage creators (according to the authors):

—The preferential attachment term represents adding links which the creator became aware of because they were popular.

—The uniform attachment term represents the case when the author adds a link because it is relevant to him, and this is irrespective of the popularity of the linked page. This allows even the poorer sites to gain some edges.

Degree distribution: The authors derive a degree distribution function for this model:

$$P(k) \propto (k+c)^{-1-\frac{1}{\alpha}} \tag{32}$$

where c is a function of m and α. This gives a power-law of exponent $(1 + 1/\alpha)$ in the tail. However, for low degrees, it deviates from the power-law, as the authors wanted.

Power-law degree distributions have shown up in many real-world graphs. However, it is clear that deviations in this do show up in practice. This is one of the few models we are aware of that specifically attempt to model such deviations, and as such, is a step in the right direction.

Open questions and discussion: This model can match deviations from power laws in degree distributions. However, further work is needed to test for other graph patterns, like diameter, community structure and such.

3.2.12 *Implementation issues.* Here, we will briefly discuss certain implementation aspects. Consider the BA model. In each iteration, we must choose edge endpoints according to the linear preferential attachment equation. Naively, each time we need to add a new edge, we could go over all the existing nodes and find the probability of choosing each node as an endpoint, based on its current degree. However, this would take $O(N)$ time each iteration, and $O(N^2)$ time to generate the entire graph. A better approach [Newman 2003] is to keep an array: whenever a new edge is added, its endpoints are appended to the array. Thus, each node appears in the array as many times as its degree. Whenever we must choose a node according to preferential attachment, we can choose any cell of the array uniformly at random, and the node stored in that cell can be considered to have been chosen under preferential attachment. This requires $O(1)$ time for each iteration, and $O(N)$ time to generate the entire graph; however, it needs extra space to store the edgelist.

This technique can be easily extended to the case when the preferential attachment equation involves a constant β, such as $P(v) \propto (k(v) - \beta)$ for the GLP model (Equation 25). If the constant β is a negative integer (say, $\beta = -1$ as in the AB model, Equation 21), we can handle this easily by adding $|\beta|$ entries for every existing node into the array. However, if this is not the case, the method needs to be modified slightly: with some probability α, the node is chosen according to the simple preferential attachment equation (like in the BA model). With probability $(1 - \alpha)$, it is chosen uniformly at random from the set of existing nodes. For each iteration, the value of α can be chosen so that the final effect is that of choosing nodes according to the modified preferential attachment equation.

3.2.13 *Summary of Preferential Attachment Models.* All preferential attachment models use the idea that the "rich get richer": high-degree nodes attract more edges, or high-PageRank nodes attract more edges, and so on. This simple process, along with the idea of network growth over time, *automatically* leads to the power-law degree distributions seen in many real-world graphs. As such, these models made a very important contribution to the field of graph mining. Still, most of these models appear to suffer from some limitations: for example, they do not seem to generate any "community" structure in the graphs they generate. Also, apart from the work of Pennock et al. [2002], little effort has gone into finding reasons for deviations from power-law behaviors for some graphs. It appears that we need to consider additional processes to understand and model such characteristics.

3.3 Incorporating Geographical Information

Both the random graph and preferential attachment models have neglected one attribute of many real graphs: the constraints of geography. For example, it is easier (cheaper) to link two routers which are physically close to each other; most of our social contacts are people we meet often, and who consequently probably live close to us (say, in the same town or city), and so on. In the following paragraphs, we discuss some important models which try to incorporate this information.

3.3.1 *The Small-World Model.*

Problem being solved: The small-world model is motivated by the observation that most real-world graphs seem to have low average distance between nodes (a global property), but have high clustering coefficients (a local property). Two experiments from the field of sociology shed light on this phenomenon.

Travers and Milgram [1969] conducted an experiment where participants had to reach randomly chosen individuals in the U.S.A. using a chain letter between close acquaintances. Their surprising find was that, for the chains that completed, the average length of the chain was only six, in spite of the large population of individuals in the "social network." While only around 29% of the chains were completed, the idea of small paths in large graphs was still a landmark find.

The reason behind the short paths was discovered by Mark Granovetter [1973], who tried to find out how people found jobs. The expectation was that the job seeker and his eventual employer would be linked by long paths; however, the actual paths were empirically found to be very short, usually of length one or two. This corresponds to the low average path length mentioned above. Also, when asked whether a friend had told them about their current job, a frequent answer of the respondents was *"Not a friend, an acquaintance"*. Thus, this low average path length was being caused by acquaintances, with whom the subjects only shared *weak ties*. Each acquaintance belonged to a different social circle and had access to different information. Thus, while the social graph has high clustering coefficient (i.e., is "clique-ish"), the low diameter is caused by weak ties joining faraway cliques.

Description and properties: Watts and Strogatz [1998] independently came up with a model which had exactly these characteristics: it has *high clustering coefficient* but *low diameter* . Their model (Figure 12), which has only one parameter p, is described below.

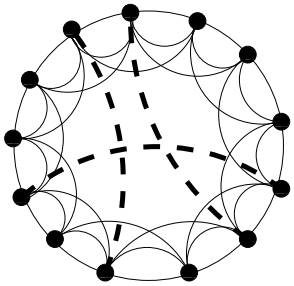


Fig. 12. *The small-world model:* Nodes are arranged in a ring lattice; each node has links to its immediate neighbors (solid lines) and some long-range connections (dashed lines).

—*Regular ring lattice:* Start with a ring lattice (N, k): this is a graph with N nodes set in a circle. Each node has k edges to its closest neighbors, with $k/2$ edges on each side. This is the set of *close friendships*, and has high clustering coefficient. Let $N \gg k \gg \ln N \gg 1$.

—*Rewiring:* For each node u, each of its edges (u, v) is rewired with probability p to form some different edge (u, w), where node w is chosen uniformly at random. Self-loops and duplicate edges are forbidden. This accounts for the *weak acquaintances*.

Distance between nodes, and Clustering coefficient: With $p = 0$, the graph remains a plain ring lattice. Both the clustering coefficient and the average distance between nodes are fairly high ($CC(p = 0) \sim 3/4$ and $L(p = 0) \sim N/2k \gg 1$). Thus, small-world structure is absent. When $p = 1$, both the clustering coefficient and the average distance are fairly low ($CC(p = 1) \sim k/N \ll 1$ and $L(p = 1) \sim \ln N / \ln k$). Thus, the graph is not "clique-ish" enough. However, there exists a range of p values for which $L(p) \sim L(1)$ but $CC(p) \gg CC(1)$; that is, the average distance remains low while the clustering coefficient is high. These are exactly the desired properties.

The reason for this is that the introduction of a few long-range edges (which are exactly the weak ties of Granovetter) leads to a highly nonlinear effect on the average distance L. Distance is contracted not only between the endpoints of the edge, but also their immediate neighborhoods (circles of friends). However, these few edges lead to a very small change in the clustering coefficient. Thus, we get a broad range of p for which the small-world phenomenon coexists with a high clustering coefficient.

Degree distribution: All nodes start off with degree k, and the only changes to their degrees are due to rewiring. The shape of the degree distribution is similar to that of a random graph, with a strong peak at k, and it decays exponentially for large k.

Open questions and discussion: The small-world model is very successful in combining two important graph patterns: small diameters and high clustering

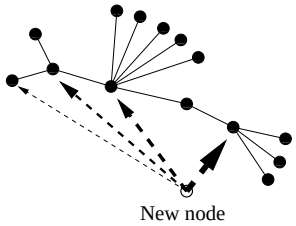


Fig. 13. *The Waxman model:* New nodes prefer to connect to existing nodes which are closer in distance.

coefficients. However, the degree distribution decays exponentially, and does not match the power-law distributions of many real-world graphs. Extension of the basic model to power law distributions is a promising research direction.

3.3.2 *The Waxman Model.*

Problem being solved: The Internet graph is constrained by geography: it is cheaper to link two routers which are close in distance. Waxman [1988] proposed a very simple model which focuses on this interaction of network generation with geography (Figure 13).

Description and properties: The Waxman generator places random points in Cartesian two-dimensional space (representing the placement of routers on land). An edge (u, v) is placed between two points u and v with probability

$$P(u, v) = \beta \exp \frac{-d(u, v)}{L\alpha} \tag{33}$$

Here, α and β are parameters in the range $(0, 1)$, $d(u, v)$ is the Euclidean distance between points u and v, and L is the maximum Euclidean distance between points.

The parameters α and β control the geographical constraints. The value of β affects the *edge density*: larger values of β result in graphs with higher edge densities. The value of α relates the short edges to longer ones: a small value of α increases the density of short edges relative to longer edges.

Open questions and discussion: The Waxman generator has been very popular in the networking community. However, it does not yield a power law degree distribution, and further work is needed to analyze the other graph patterns for this generator.

3.3.3 *The BRITE generator.*

Problem being solved: Medina et al. [2000] try to combine the geographical properties of the Waxman generator with the incremental growth and preferential attachment techniques of the BA model. Their graph generator, called BRITE, has been extensively used in the networking community for simulating the structure of the Internet.

Description and properties: The main features of BRITE are:

—*Node Placement*: The geography is assumed to be a square grid. Nodes can either

be placed randomly, or with a heavy-tailed distribution.

—*Links per node*: As in the BA model, this is set to m, a model parameter.

—*Incremental Growth*: Either we could start off by placing all the nodes and then adding links (as in the Waxman model), or we could add nodes and links as we go along (as in the BA model). The latter gives us incremental growth.

—*Wiring of edges*: The authors provide three cases. (1) The edges could link randomly chosen nodes. (2) We could have pure preferential connectivity, as in the BA model. (3) The interesting case is when we combine preferential connectivity with geographical constraints. Suppose that we want to add an edge to node u. The probability of the other endpoint of the edge being node v is a *weighted* preferential attachment equation, with the weights being the the probability of that edge existing in the pure Waxman model (Equation 33)

$$P(u,v) = \frac{w(u,v)k(v)}{\sum_i w(u,i)k(i)} \tag{34}$$
$$\text{where } w(u,v) = \beta \exp \frac{-d(u,v)}{L\alpha} \text{ as in Eq. 33}$$

Open questions and discussion: The emphasis of BRITE is on creating a system that can be used to generate different kinds of topologies. This allows the user a lot of flexibility, and is one reason behind the widespread use of BRITE in the networking community. However, there is little discussion of parameter fitting: how can the model parameters be set so as to generate synthetic graphs which successfully match the properties (such as the power law exponent) of some given real-world graph? Developing algorithms for parameter fitting, and understanding the scenarios which lead to power law graphs (such as the *Heuristically Optimized Tradeoffs* model described later in section 3.4.2), are interesting avenues for further research.

3.3.4 *Other Geographical Constraints.*

Problem being solved: Yook et al. [2002] find two interesting linkages between geography and networks (specifically the Internet):

(1) The geographical distribution of Internet routers and Autonomous Systems (AS) is a fractal, and is strongly correlated with population density. This is intuitive: more people require more bandwidth and more routers. This finding is at odds with most of the previous models, which usually expect the nodes to be spread uniformly at random in some geographical area (BRITE allows inhomogeneous distributions, but not fractals).

(2) They plot the probability of an edge versus the length of the edge and find that the probability is *inversely proportional* to the Euclidean distance between the endpoints of the edge. They explain this by saying that the cost of linking two routers is essentially the cost of administration (fixed) and the cost of the physical wire (proportional to distance). For long links, the distance-cost dominates, so the probability of the link should be inversely proportional to distance. However, in the Waxman and BRITE models, this probability decays exponentially with length (Equation 33).

Description and properties: To remedy the first problem, they suggest using a self-similar geographical distribution of nodes. For the second problem, they propose a modified version of the BA model. Each new node u is placed on the map using the self-similar distribution, and adds edges to m existing nodes. For each of these edges, the probability of choosing node v as the endpoint is given by a modified preferential attachment equation:

$$P(\text{node } u \text{ links to existing node } v) \propto \frac{k(v)^\alpha}{d(u,v)^\sigma} \tag{35}$$

where $k(v)$ is the current degree of node v and $d(u,v)$ is the Euclidean distance between the two nodes. The values α and σ are parameters, with $\alpha = \sigma = 1$ giving the best fits to the Internet. They show that varying the values of α and σ can lead to significant differences in the topology of the generated graph.

Similar geographical constraints may hold for social networks as well: individuals are more likely to have friends in the same city as compared to other cities, in the same state as compared to other states, and so on recursively. Watts et al. [2002] and (independently) Kleinberg [2001] propose a hierarchical model to explain this phenomenon; we will discuss both in more detail in section 5.3.

3.4 Topology from Resource Optimizations

Most of the methods described above have approached power-law degree distributions from the preferential-attachment viewpoint: if the "rich get richer", power-laws might result. However, another point of view is that power laws can result from *resource optimizations*. We will discuss some such models below.

3.4.1 *The Highly Optimized Tolerance model.*

Problem being solved: Carlson and Doyle [1999; 2000] have proposed an optimization-based reason for the existence of power laws in graphs. They say that power laws may arise in systems due to *tradeoffs* between yield (or profit), resources (to prevent a risk from causing damage) and tolerance to risks.

Description and properties: As an example, suppose we have a forest which is prone to forest fires. Each portion of the forest has a different chance of starting the fire (say, the dryer parts of the forest are more likely to catch fire). We wish to minimize the damage by assigning resources such as firebreaks at different positions in the forest. However, the total available resources are limited. The problem is to place the firebreaks so that the expected cost of forest fires is minimized.

In this model, called the *Highly Optimized Tolerance* (HOT) model, we have n possible events (starting position of a forest fire), each with an associated probability $p_i (1 \le i \le n)$ (dryer areas have higher probability). Each event can lead to some *loss* l_i, which is a function of the resources r_i allocated for that event: $l_i = f(r_i)$. Also, the total resources are limited: $\sum_i r_i \le R$ for some given R. The aim is to minimize the expected cost

$$J = \left\{ \sum_i p_i l_i \mid l_i = f(r_i), \sum_i r_i \le R \right\} \tag{36}$$

Degree distribution: The authors show that if we assume that cost and resource usage are related by a power law $l_i \propto r_i^{\beta}$, then, under certain assumptions on the probability distribution p_i, resources are spent on places having higher probability of costly events. In fact, resource placement is related to the probability distribution p_i by a power law. Also, the probability of events which cause a loss greater than some value k is related to k by a power law.

The salient points of this model are:

—high efficiency, performance and robustness to designed-for uncertainties
—hypersensitivity to design flaws and unanticipated perturbations
—nongeneric, specialized, structured configurations, and
—power laws.

Resilience under attack: This concurs with other research regarding the vulnerability of the Internet to attacks. Several researchers have found that while a large number of randomly chosen nodes and edges can be removed from the Internet graph without appreciable disruption in service, attacks *targeting* important nodes can disrupt the network very quickly and dramatically [Palmer et al. 2002; Albert et al. 2000]. The HOT model also predicts a similar behavior: since routers and links are *expected* to be down occasionally, it is a "designed-for" uncertainty and the Internet is impervious to it. However, a *targeted* attack is not designed for, and can be devastating.

Newman et al. [2002] modify HOT using a utility function which can be used to incorporate "risk aversion." Their model (called *Constrained Optimization with Limited Deviations* or COLD) truncates the tails of the power laws, lowering the probability of disastrous events.

HOT has been used to model the sizes of files found on the WWW. The idea is that dividing a single file into several smaller files leads to faster load times, but increases the cost of navigating through the links. They show good matches with this dataset.

Open questions and discussion: The HOT model offers a completely new recipe for generating power laws; power laws can result as a by-product of resource optimizations. However, this model requires that the resources be spread in an *globally-optimal* fashion, which does not appear to be true for several large graphs (such as the WWW). This led to an alternative model by Fabrikant et al. [2002], which we discuss below.

3.4.2 *The Heuristically Optimized Tradeoffs model.*

Problem being solved: The previous model requires globally-optimal resource allocations. However, graphs like the Internet appear to have evolved by *local* decisions taken by the engineers/administrators on the spot. Fabrikant et al. [2002] propose an alternative model in which the graph grows as a result of trade-offs made *heuristically* and locally (as opposed to optimally, for the HOT model).

Description and properties: The model assumes that nodes are spread out over

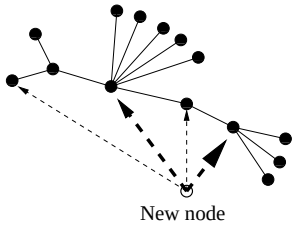


Fig. 14. *The Heuristically Optimized Tradeoffs model:* A new node prefers to link to existing nodes which are both close in distance and occupy a "central" position in the network.

a geographical area. One new node is added in every iteration, and is connected to the rest of the network with *one* link. The other endpoint of this link is chosen to optimize between two conflicting goals: (1) minimizing the "last-mile" distance, that is, the *geographical* length of wire needed to connect a new node to a pre-existing graph (like the Internet), and, (2) minimizing the transmission delays based on number of hops, or, the distance along the network to reach other nodes. The authors try to optimize a linear combination of the two (Figure 14). Thus, a new node i should be connected to an existing node j chosen to minimize

$$\alpha.d_{ij} + h_j \quad (j < i) \tag{37}$$

where d_{ij} is the distance between nodes i and j, h_j is some measure of the "centrality" of node j, and α is a constant that controls the relative importance of the two.

Degree distribution: The authors find that the characteristics of the network depend greatly on the value of α: when α is less than a particular constant based on the shape of the geography, the "centrality" constraints dominate and the generated network is a *star* (one central "hub" node which connects to all the other "spoke" nodes). On the other hand, when α grows as fast as $\log N$, the geographical constraints dominate and the degree distribution falls off exponentially fast. However, if α is anywhere in between, power-law degree distributions result.

Open questions and discussion: As in the *Highly Optimized Tolerance* model described before (Section 3.4.1), power laws are seen to fall off as a by-product of resource optimizations. However, only local optimizations are now needed, instead of global optimizations. This makes the *Heuristically Optimized Tradeoffs* model very appealing.

In its current version, however, the model only generates graphs of density 1 (that is, one edge per node). This also implies that the graph is actually a tree, whereas many real-world graphs have cycles (for example, a node might have multiple connections to the Internet to maintain connectivity in case one of its links fails). Also, in addition to the degree distribution, the generated graphs need to be analyzed for all other graph patterns too. Further research needs to modify the basic model to address these issues. One step in this direction is the recent work of Berger et al. [2005], who generalize the *Heuristically Optimized Tradeoffs* model, and show that it is equivalent to a form of preferential attachment; thus,

competition between opposing forces can give rise to preferential attachment, and we already know that preferential attachment can, in turn, lead to power laws and exponential cutoffs.

3.5 Generators for the Internet Topology

While the generators described above are applicable to any graphs, some special-purpose generators have been proposed to specifically model the Internet topology. Structural generators exploit the hierarchical structure of the Internet, while the Inet generator modifies the basic preferential attachment model to better fit the Internet topology. We look at both of these below.

3.5.1 *Structural Generators.*

Problem being solved: Work done in the networking community on the structure of the Internet has led to the discovery of *hierarchies* in the topology. At the lowest level are the Local Area Networks (LANs); a group of LANs are connected by *stub domains*, and a set of *transit domains* connect the stubs and allow the flow of traffic between nodes from different stubs. More details are provided in Section 2.7.1. However, the previous models do not explicitly enforce such hierarchies on the generated graphs.

Description and properties: Calvert et al. [1997] propose a graph generation algorithm which specifically models this hierarchical structure. The general topology of a graph is specified by six parameters, which are the numbers of transit domains, stub domains and LANs, and the number of nodes in each. More parameters are needed to model the connectivities within and across these hierarchies. To generate a graph, points in a plane are used to represent the locations of the centers of the transit domains. The nodes for each of these domains are spread out around these centers, and are connected by edges. Now, the stub domains are placed on the plane and are connected to the corresponding transit node. The process is repeated with nodes representing LANs.

The authors provide two implementations of this idea. The first, called *Transit-Stub*, does not model LANs. Also, the method of generating connected subgraphs is to keep generating graphs till we get one that is connected. The second, called *Tiers*, allows multiple stubs and LANs, but allows only one transit domain. The graph is made connected by connecting nodes using a minimum spanning tree algorithm.

Open questions and discussion: These models can specifically match the hierarchical nature of the Internet, but they make no attempt to match any other graph pattern. For example, the degree distributions of the generated graphs need not be power laws. Hence, while these models have been widely used in the networking community, the need modifications to be as useful in other settings.

Tangmunarunkit et al. [2001] compare such structural generators against generators which focus only on power-law distributions. They find that even though power-law generators do not explicitly model hierarchies, the graphs generated by them have a substantial level of hierarchy, though not as strict as with the generators described above. Thus, the hierarchical nature of the structural generators can also be mimicked by other generators.

3.5.2 *The Inet topology generator.*

Problem being solved: Winick and Jamin [2002] developed the Inet generator to model only the Internet Autonomous System (AS) topology, and to match features specific to it.

Description and properties: Inet-2.2 generates the graph by the following steps:

—Each node is assigned a degree from a power-law distribution with an exponential cutoff (as in Equation 16).

—A spanning tree is formed from all nodes with degree greater than 1.

—All nodes with degree one are attached to his spanning tree using linear preferential attachment.

—All nodes in the spanning tree get extra edges using linear preferential attachment till they reach their assigned degree.

The main advantage of this technique is in ensuring that the final graph remains connected.

However, they find that under this scheme, too many of the low degree nodes get attached to other low-degree nodes. For example, in the Inet-2.2 topology, 35% of degree 2 nodes have adjacent nodes with degree 3 or less; for the Internet, this happens only for 5% of the degree-2 nodes. Also, the highest degree nodes in Inet-2.2 do not connect to as many low-degree nodes as the Internet. To correct this, Winick and Jamin come up with the Inet-3 generator, with a modified preferential attachment system.

The preferential attachment equation now has a weighting factor which uses the degrees of the nodes on both ends of some edge. The probability of a degree i node connecting to a degree j node is

$$P(\text{degree } i \text{ node connects to degree } j \text{ node}) \propto w_i^j . j \tag{38}$$

$$\text{where } w_i^j = MAX\left(1, \sqrt{\left(\log\frac{i}{j}\right)^2 + \left(\log\frac{f(i)}{f(j)}\right)^2}\right) \tag{39}$$

Here, $f(i)$ and $f(j)$ are the number of nodes with degrees i and j respectively, and can be easily obtained from the degree distribution equation. Intuitively, what this weighting scheme is doing is the following: when the degrees i and j are close, the preferential attachment equation remains linear. However, when there is a large difference in degrees, the weight is the Euclidean distance between the points on the log-log plot of the degree distribution corresponding to degrees i and j, and this distance increases with increasing difference in degrees. Thus, edges connecting nodes with a big difference in degrees are preferred.

Open questions and discussion: Inet has been extensively used in the networking literature. However, the fact that it is so specific to the Internet AS topology makes it somewhat unsuitable for any other topologies.

3.6 Comparison Studies

While a large body of work has been done on developing new graph generators, effort has also gone into comparing different graph generators, especially on certain graphs like the Internet. However, different studies have used different metrics for comparing different graph generators. We combine the results of several studies in our discussion below [Tangmunarunkit et al. 2001; 2002; Bu and Towsley 2002; Albert et al. 2000].

We have already seen some of the patterns and metrics that were used in these studies: expansion (section 2.3), resilience (section 2.5.1), the hierarchical structure of the Internet (section 3.5.1), characteristic path lengths and average diameter (section 2.3), and the clustering coefficient (section 2.4.1). Apart from these, Tangmunarunkit et al. [2001; 2002] also looked at graph distortion, defined as follows.

DEFINITION 3.1 DISTORTION. *Consider any spanning tree T on the graph. Then, the* distortion for T *is the average distance on T between any two nodes that are connected by an edge in the graph. The* distortion for the graph *is the smallest such average over all possible spanning trees.*

The graph distortion measures the difference between the real graph and its spanning tree. Tangmunarunkit et al. [2001; 2002] use heuristics to evaluate this metric.

Now we will describe the results for each of these metrics. When possible, we will try to explain the reasons behind the results; we note, however, that most of these results are as yet formally unexplained.

Expansion: Both the Internet AS level and Router level graphs exhibit high expansion[2] [Tangmunarunkit et al. 2001; 2002]. The PLRG model (section 3.1.2) matches this pattern. The Tiers model (section 3.5.1) has low expansion.

Resilience under random failures: Both the Internet AS level and Router level graphs show high resilience under random failures [Tangmunarunkit et al. 2001; 2002; Palmer et al. 2002]. The PLRG and AB (section 3.2.3) models match this: in fact, power law graphs remain unaffected even when as many as 5% of the nodes are randomly chosen and removed [Albert et al. 2000; Bollobás et al. 2003]. However, for graphs with exponentially decaying degree distributions, such as the Erdós-Rényi random graph (section 3.1.1) and the Small-World model (section 3.3.1), the average diameter increases monotonically as random nodes are removed. The Transit-Stub model (section 3.5.1) also has low resilience.

Resilience under targeted attacks: When nodes are removed in decreasing order of degree, the situation is the complete reverse of the "random failures" scenario. Power-law graphs show drastic increases in average diameter (doubling the original value as the top 5% of the highest-degree nodes are removed), while exponential graphs exhibit more or less the same behavior as for the *failure* case. This

[2]For the expansion, resilience and distortion metrics, Tangmunarunkit et al. [2001; 2002] distinguish between only two states: "low" and "high" (for example, exponential expansion is high, and sub-exponential is low).

brings to the forefront the importance of the most-connected nodes in power-law graphs [Palmer et al. 2002; Albert et al. 2000; Bollobás et al. 2003].

Distortion: Both the Internet AS level and Router level graphs have low distortion, and the PLRG model matches this [Tangmunarunkit et al. 2001; 2002]. The Waxman model (section 3.3.2) has high distortion. Inet-3.0 is found to show similar distortion as the Internet AS level graph [Winick and Jamin 2002].

Hierarchical structure of the Internet: Even though power-law models do not lead to any explicit hierarchy of nodes, a hierarchy shows up nonetheless in graphs generated by such models. Tangmunarunkit et al. [2001; 2002] surmise that this hierarchy is a side effect of the power-law distribution: nodes with high degree function appear to be near the top of the hierarchy, while low-degree nodes form the leaves.

Characteristic path length: The AB, GLP, and Inet models (sections 3.2.3, 3.2.5, and 3.5.2) give similar path lengths as the Internet AS level graph with proper choice of parameters, while PLRG does not. PLRG also shows very high variance for this metric [Bu and Towsley 2002]. Inet-3.0 has similar characteristic path lengths over time as the Internet AS graph [Winick and Jamin 2002].

Clustering coefficient: The clustering coefficient of GLP is closer to the Internet AS level graph than those for AB, PLRG, and Inet [Bu and Towsley 2002]. Inet-3.0 exhibits lower clustering coefficient than the Internet AS graph, possibly because it does not have a large, dense clique connecting the high-degree "core" nodes, as is seen in the Internet graph [Winick and Jamin 2002].

In addition, Winick and Jamin [2002] compared the Inet-3.0 generator to the Internet AS graph for several other patterns, and observed good fits for many of these. Note, however, that Inet-3.0 was developed specifically for the Internet AS topology.

Reasons behind the resilience properties of the Internet: Much effort has gone into understanding the resilience properties of the Internet (resilient under random failures, but drastically affected by targeted attacks). Albert et al. [Albert et al. 2000] propose that in power law graphs like the Internet, the high-degree nodes are the ones "maintaining" most of the connectivity, and since there are so few of them, it is unlikely for them to be chosen for removal under random failures. Targeted attacks remove exactly these nodes, leading to severe connectivity failures.

Tauro et al. [2001] propose a solution based on the structure of the Internet AS graph. They say that the graph is organized as a *Jellyfish* (or concentric rings around a core), with the most important nodes in the core, and layers further from the core decreasing in importance (see Section 2.7.1). Random node removal mostly removes one-degree nodes which hang off the core or the layers, and do not affect connectivity. However, targeted node removal removes nodes from the (small) core and then successively from the important layers; since most nodes connect in or towards the central core, this leads to a devastating loss of connectivity. Perhaps

the Internet power law graph was generated in a fashion such that the "core" nodes achieved the highest connectivity; that would agree with both [Tauro et al. 2001] and [Albert et al. 2000].

Interestingly, similar behavior is exhibited by metabolic pathways in organisms; Jeong et al. [2000] show that the diameter does not change under random node removal, but increases fourfold when only 8% of the most connected nodes are removed. Solé and Montoya [2001] see the same thing with ecological networks, and Newman et al. [2002] for email networks.

3.7 The R-MAT (Recursive MATrix) graph generator

We have seen that most of the current graph generators focus on only one graph pattern – typically the degree distribution – and give low importance to all the others. There is also the question of how to fit model parameters to match a given graph. What we would like is a tradeoff between parsimony (few model parameters), realism (matching most graph patterns, if not all), and efficiency (in parameter fitting and graph generation speed). In this section, we present the R-MAT generator, which attempts to address all of these concerns.

Problem being solved: The R-MAT [Chakrabarti et al. 2004] generator tries to meet several desiderata:

—The generated graph should match several graph patterns, including *but not limited to* power-law degree distributions (such as hop-plots and eigenvalue plots).
—It should be able to generate graphs exhibiting deviations from power-laws, as observed in some real-world graphs [Pennock et al. 2002].
—It should exhibit a strong "community" effect.
—It should be able to generate directed, undirected, bipartite or weighted graphs with the same methodology.
—It should use as few parameters as possible.
—There should be a fast parameter-fitting algorithm.
—The generation algorithm should be efficient and scalable.

Description and properties:

The R-MAT generator creates directed graphs with 2^n nodes and E edges, where both values are provided by the user. We start with an empty adjacency matrix, and divide it into four equal-sized partitions. One of the four partitions is chosen with probabilities a, b, c, d respectively ($a + b + c + d = 1$), as in Figure 15. The chosen partition is again subdivided into four smaller partitions, and the procedure is repeated until we reach a simple cell (=1×1 partition). The nodes (that is, row and column) corresponding to this cell are linked by an edge in the graph. This process is repeated E times to generate the full graph. There is a subtle point here: we may have *duplicate* edges (ie., edges which fall into the same cell in the adjacency matrix), but we only keep one of them when generating an unweighted graph. To smooth out fluctuations in the degree distributions, some noise is added to the (a, b, c, d) values at each stage of the recursion, followed by renormalization (so that $a + b + c + d = 1$). Typically, $a \geq b$, $a \geq c$, $a \geq d$.

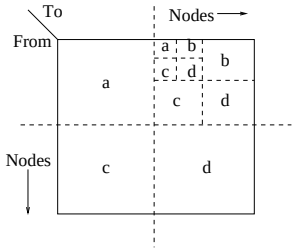


Fig. 15. *The R-MAT model:* The adjacency matrix is broken into four equal-sized partitions, and one of those four is chosen according to a (possibly non-uniform) probability distribution. This partition is then split recursively till we reach a single cell, where an edge is placed. Multiple such edge placements are used to generate the full synthetic graph.

Degree distribution: There are only 3 parameters (the partition probabilities a, b, and c; $d = 1 - a - b - c$). The skew in these parameters ($a \geq d$) leads to lognormals and the DGX [Bi et al. 2001] distribution, which can successfully model both power-law and "unimodal" distributions [Pennock et al. 2002] under different parameter settings.

Communities: Intuitively, this technique is generating "communities" in the graph:

—The partitions a and d represent separate groups of nodes which correspond to communities (say, "Linux" and "Windows" users).
—The partitions b and c are the *cross-links* between these two groups; edges there would denote friends with separate preferences.
—The recursive nature of the partitions means that we automatically get sub-communities within existing communities (say, "RedHat" and "Mandrake" enthusiasts within the "Linux" group).

Diameter, singular values and other properties: We show experimentally that graphs generated by R-MAT have small diameter and match several other criteria as well.

Extensions to undirected, bipartite and weighted graphs: The basic model generates directed graphs; all the other types of graphs can be easily generated by minor modifications of the model. For undirected graphs, a directed graph is generated and then made symmetric. For bipartite graphs, the same approach is used; the only difference is that the adjacency matrix is now rectangular instead of square. For weighted graphs, the number of *duplicate* edges in each cell of the adjacency matrix is taken to be the weight of that edge. More details may be found in [Chakrabarti et al. 2004].

Parameter fitting algorithm: We are given some input graph, and need to fit the R-MAT model parameters so that the generated graph matches the input graph in terms of graph patterns.

We can calculate the expected degree distribution: the probability p_k of a node

having outdegree k is given by

$$p_k = \frac{1}{2^n}\binom{E}{k}\sum_{i=0}^{n}\binom{n}{i}\left[\alpha^{n-i}(1-\alpha)^i\right]^k\left[1-\alpha^{n-i}(1-\alpha)^i\right]^{E-k}$$

where 2^n is the number of nodes in the R-MAT graph, E is the number of edges, and $\alpha = a+b$. Fitting this to the outdegree distribution of the input graph provides an estimate for $\alpha = a+b$. Similarly, the indegree distribution of the input graph gives us the value of $b+c$. Conjecturing that the $a:b$ and $a:c$ ratios are approximately 75 : 25 (as seen in many real world scenarios), we can calculate the parameters (a, b, c, d).

Next, we show experimentally that R-MAT can match both power-law distributions as well as deviations from power-laws.

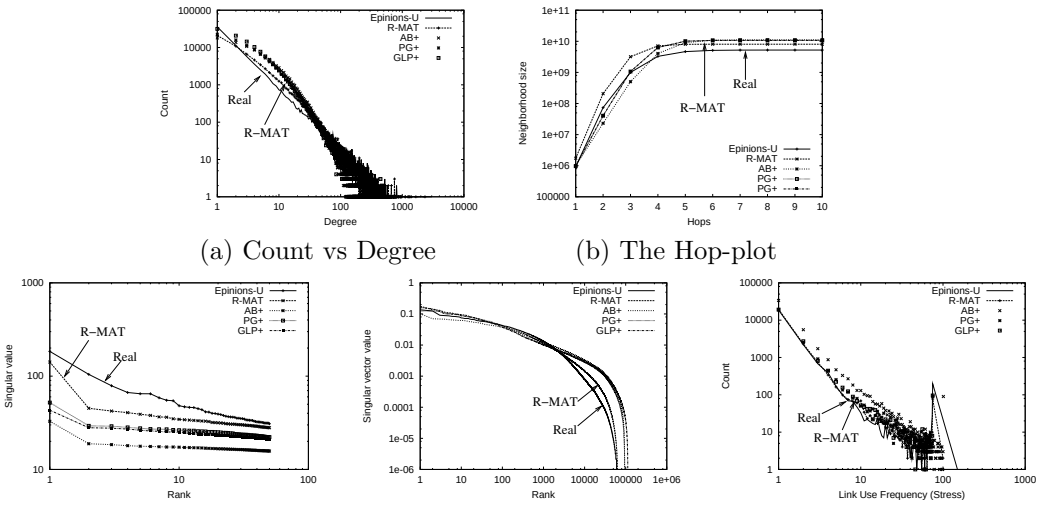
Experiments: We show experiments on the following graphs:
Epinions: A directed graph of who-trusts-whom from epinions.com [Richardson and Domingos 2002]: $N = 75,879; E = 508,960$.
Epinions-U: An undirected version of the *Epinions* graph: $N = 75,879; E = 811,602$.
Clickstream: A bipartite graph of Internet users' browsing behavior [Montgomery and Faloutsos 2001]. An edge (u, p) denotes that user u accessed page p. It has $23,396$ users, $199,308$ pages and $952,580$ edges.

The graph patterns we look at are:

(1) Both indegree and outdegree distributions.
(2) "Hop-plot" and "effective diameter": The "hop-plot" shows the number of reachable pairs of nodes, versus the number of hops (see Definitions 2.3 and 2.4).
(3) Singular value vs. rank plots: Singular values [Press et al. 1992] are similar to eigenvalues (they are the same for undirected graphs), but eigenvalues may not exist for bipartite graphs, while singular values do.
(4) "Singular vector value" versus rank plots: The "singular vector value" of a node is the absolute value of the corresponding component of the first singular vector of the graph. It can be considered to be a measure of the "importance" of the node, and as we will see later, is closely related to the widely-used concept of "Bonacich centrality" in social network analysis.
(5) "Stress" distribution: The "stress" of an edge is the number of shortest paths between node pairs that it is a part of (see Definition 2.6).

In addition to R-MAT, we show the fits achieved by some other models, chosen for their popularity or recency: these are the *AB*, *GLP*, and *PG* models (Sections 3.2.3, 3.2.5, and 3.2.11 respectively). All three can only generate undirected graphs; thus, we can compare them with R-MAT only on *Epinions-U*. The parameters of these three models are set by exhaustive search; we use the terms *AB+, PG+* and *GLP+* to stand for the original models augmented by our parameter fitting.

Epinions-U: Figure 16 shows the comparison plots on this undirected graph. R-MAT gives the closest fits. Also, note that all the y-scales are logarithmic, so small differences in the plots actually represent significant deviations.



(a) Count vs Degree (b) The Hop-plot

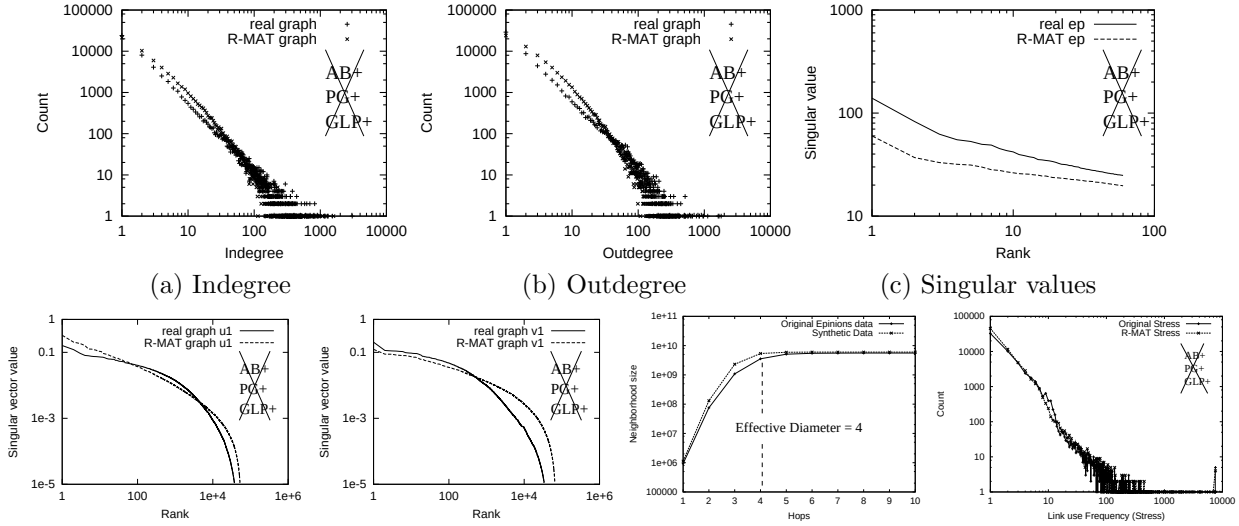
(c) Singular value vs Rank (d) Singular vector vs Rank (e) Stress

Fig. 16. *Epinions-U undirected graph:* We show (a) degree, (b) hop-plot, (c) singular value, (d) "singular vector value," and (e) stress distributions for the *Epinions-U* dataset. R-MAT gives the best matches to the *Epinions-U* graph, among all the generators. In fact, for the stress distribution, the R-MAT and *Epinions-U* plots are almost indistinguishable.

Epinions: Figure 17 shows results on this directed graph. The R-MAT fit is very good; the other models considered are not applicable.

Clickstream: Figure 18 shows results on this bipartite graph. As before, the R-MAT fit is very good. In particular, note that the indegree distribution is a power law while the outdegree distribution deviates significantly from a power law; R-MAT matches *both* of these very well. This is because R-MAT generates a "truncated discrete lognormal" (a DGX distribution [Bi et al. 2001]) which, under the correct parameter settings, can give good fits to power laws as well. Again, the other models are not applicable.

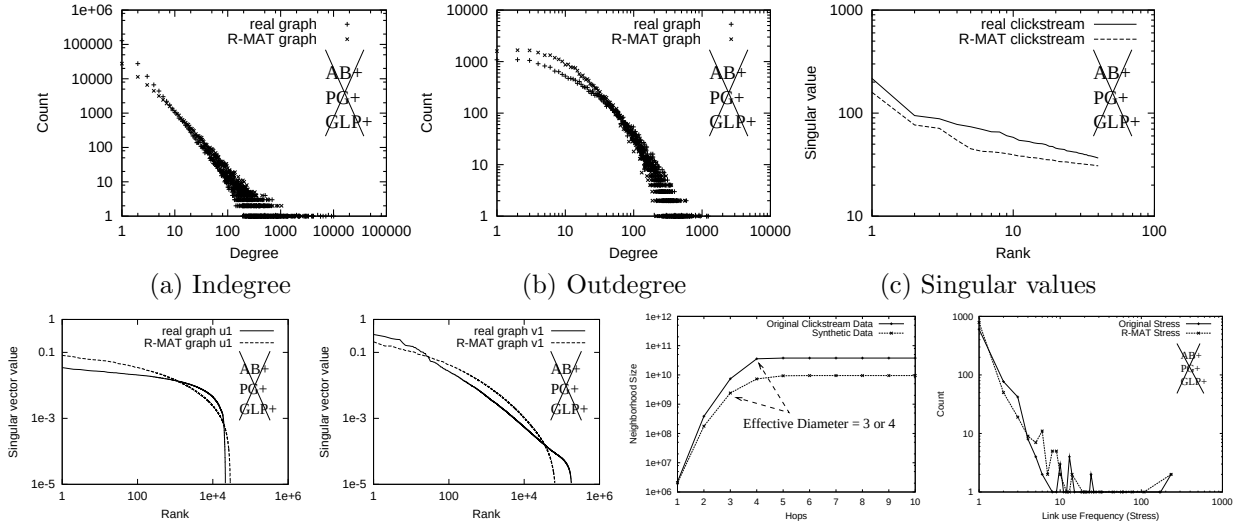
Open questions and discussion: While the R-MAT model shows promise, there has not been any thorough analytical study of this model. Also, it seems that only 3 parameters might not provide enough "degrees of freedom" to match all varieties of graphs; extensions of this model should be investigated. A step in this direction is the *Kronecker graph generator* [Leskovec et al. 2005], which generalizes the R-MAT model and can match several interesting patterns such as the Densification Power Law and the shrinking diameters effect (see Section 2.6) in addition to all the patterns that R-MAT matches.



(a) Indegree (b) Outdegree (c) Singular values

(d) First left singular vec. (e) First right singular vec. (f) Hop-plot (g) Stress

Fig. 17. *Epinions directed graph:* The *AB+, PG+* and *GLP+* methods **do not apply**. The crosses and dashed lines represent the R-MAT generated graphs, while the pluses and strong lines represent the real graph.



(a) Indegree (b) Outdegree (c) Singular values

(d) First left singular vec. (e) First right singular vec. (f) Hop-plot (g) Stress

Fig. 18. *Clickstream bipartite graph:* The *AB+, PG+* and *GLP+* methods **do not apply**. The crosses and dashed lines represent the R-MAT generated graphs, while the pluses and strong lines represent the real graph.

Term from Social Network Analysis	Meaning
Social network, or Sociogram	Graph
Actor	Node
Link	Edge
Ego	Current node under discussion
Alters	Other nodes, from the point of view of the Ego
Bonacich centrality of a node	Absolute value of the first eigenvector component corresponding to that node
Betweenness centrality	"Stress" (see Definition 2.6)

Table IV. *Social networks terminology:* We provide a list of typical terms used in Social Network Analysis, and their how they correspond to terms we are familiar with.

3.8 Graph Generators: A summary

We have seen many graph generators in the preceding pages. Is any generator the "best?" Which one should we use? The answer seems to depend on the application area: the *Inet* generator is specific to the Internet and can match its properties very well, the *BRITE* generator allows geographical considerations to be taken into account, "edge copying" models provide a good intuitive mechanism for modeling the growth of the Web along with matching degree distributions and community effects, and so on. However, the final word has not yet been spoken on this topic. Almost all graph generators focus on only one or two patterns, typically the degree distribution; there is a need for generators which can combine many of the ideas presented in this section, so that they can match most, if not all, of the graph patterns. R-MAT is a step in this direction.

4. SOCIAL NETWORKS

While the field of Graph Mining has been a relatively recent development in the Data Mining community, it has been studied under different guises by other groups of researchers, most notably by sociologists. Their work has culminated in the acceptance and usage of Social Network Analysis (SNA) as a tool for investigating the structure of social groups and organizations. For example, Cross et al. [2002] use it to analyze the "invisible" patterns of interaction in organizations and map the informal networks in use. Weeks et al. [2002] map the social network of drug users who exchange needles (and hence may spread AIDS).

Below, we give a very brief introduction to some of the important concepts. Interested readers are referred to the excellent book by Wasserman and Faust [1994]. A nice introduction can also be found in [Hanneman and Riddle 2005]. Many of the concepts discussed in the previous sections also show up in Social Network Analysis, but under different names; Table IV gives the meanings of some of them.

4.1 Mapping social networks

One important aspect of SNA is getting the data itself. Unlike networks like the Internet, WWW or metabolic pathways, social network mapping is not easily amenable to automated techniques. The primary method of explication involves interviewing the subjects. Formulation of interview questions is interesting in its own right, but not relevant to our work. Another question is: how do we choose the people to interview? There are two basic approaches:

—*Full network methods:* Here, we prespecify the set of actors, and then collect data on all the edges in the graph. This is similar to a census, and the final result is information about all the existing ties among the actors. Thus, the mapped social network is complete, but collecting the data is very hard.

—*Snowball methods:* We start with a focal actor or set of actors. We ask them to name their neighbors in the graph. These neighbors are added to the set, and we continue the process till some stopping criterion is reached. While this method is cheaper, it does not locate "isolated" actors, nor does it guarantee finding all connected individuals in the population. This method also tends to overstate the "solidarity" of the population.

4.2 Dataset characteristics

In general, the patterns we look for in SNA are similar to the ones discussed previously in Section 2. For each actor, we check his/her in-degree (related to his *prestige*), out-degree (related to his *influence*) and *distance* to other actors. For the entire network, we can calculate *size* (number of nodes), *density*, *reciprocity* (if I know you, what are the chances that you know me too?) and *transitivity* (equivalent to the clustering coefficient). Freeman [1977] defines *betweenness*, which we have already seen in previous sections. `UCINET` [Borgatti et al. 1999] is one well-known software implementing these.

An important aspect of SNA is the determination of an actor's *centrality* in the network. Many measures of centrality are in use, each having some distinct characteristics:

—*Degree centrality:* Nodes with high degree are considered to be more central. However, this weights a node only by its immediate neighbors and not by, say, its 2-hop and 3-hop neighbors.

—*Bonacich centrality:* This measure uses the degree of the indirect neighbors too [Bonacich 1987]. The Bonacich centralities of the nodes in a graph are precisely the components of the first eigenvector of the graph.

—*Closeness centrality:* This is the (normalized) inverse of the average distance metric. Nodes which have low distance to all other nodes in the graph have high closeness centrality.

—*Betweenness centrality:* Nodes with high betweenness values occur on more "shortest-paths", and are presumably more important than nodes with low betweenness.

—*Flow centrality:* This is similar to betweenness centrality, except that instead of considering only the shortest paths between pairs of nodes, we consider all paths.

4.3 Structure from data

The classic "community structure" is a clique. However, the strict clique definition may be too strong for various applications. Several relaxed definitions have been proposed, such as:

—*N-clique:* Each node in an N-clique must be able to reach every other node in it within N hops. However, these paths may pass through non-members of the N-clique.

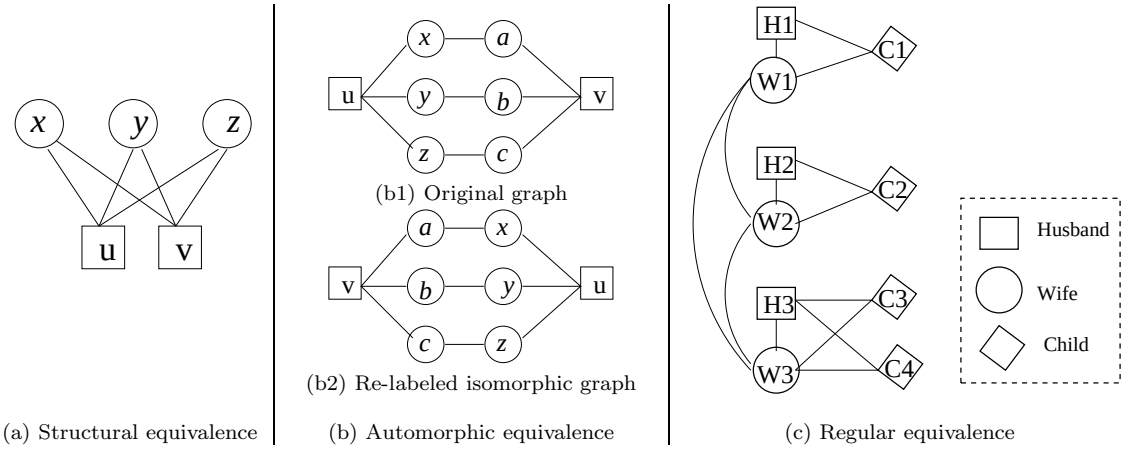


Fig. 19. *Social "roles" and equivalences:* (a) Nodes u and v are structurally equivalent because they are linked to exactly the same set of nodes: x, y, and z. (b) Nodes u and v of plot (b1) are automorphically equivalent because we can exchange their positions, and then re-label the rest of the nodes as in plot (b2), so that the new graph is isomorphic to the original. (c) The squares ("husbands"), circles ("wives") and rhombuses ("children") form three regularly-equivalent classes. A "husband" connects to a "wife" and a 'child"; a "wife" to a "husband," "child," and other "wives"; a "child" connects to a "husband" and a "wife." Thus, each class is defined by its set of relationships to the other classes. Note that child $C1$ and $C4$ are not structurally or automorphically equivalent, but they are regularly equivalent.

—*N-clan:* This is an N-clique with the restriction that all pairs of nodes in it should have a path of at most N hops passing only through other nodes of the N-clan.

—*K-plex:* Each member must have edges to all but K other members.

—*K-core:* Each member must have edges to *at least* K other members.

Another common notion in SNA is that of a core/periphery structure. Intuitively, a (sub)graph consists of a cohesive core with some sparse peripheral nodes. Borgatti and Everett [1999] model an "ideal" core/periphery structure as an adjacency matrix with a block of 1 values for the core-core edges and a block of 0 values for the periphery-periphery edges. To actually find the core and periphery nodes in a given network, they use a function optimization routine which tries to maximize the correlation between the given graph and such an "ideal" graph. This, however, might not be easy or efficient for huge graphs.

4.4 Social "roles"

This refers to an abstract concept regarding the "position" of a actor in society. This could be based, in part, on the relationships that the actor in question has with other actors. For example, the "husband" role is defined in part as being linked to another actor in a "wife" role. In other words, social roles could be thought of as representing regularities in the relationships between actors.

Actors playing a particular social role have to be equivalent/similar to each other by some metric. In general, the following three kinds of similarities are considered, in decreasing order of constraints [Hanneman and Riddle 2005; Borgatti and Everett 1989]:

—*Structural equivalence: Two actors u and v in a graph $G = (V, E)$ are structurally equivalent iff [Borgatti and Everett 1989]*

$$\begin{aligned} \forall \text{ actors } x \in V, \quad & (u, x) \in E \Leftrightarrow (v, x) \in E \\ \text{and,} \quad & (x, u) \in E \Leftrightarrow (x, v) \in E \end{aligned}$$

In other words, they are linked to exactly the same set of nodes, with (in the case of directed graphs) the arrows pointing in the same directions. Two structurally equivalent actors can exchange their positions without changing the network. Figure 19(a) shows an example of this.

—*Automorphic equivalence: Two actors u and v of a labeled graph G are automorphically equivalent iff all the actors of G can be re-labeled to form and isomorphic graph with the labels of u and v interchanged [Hanneman and Riddle 2005].* Two automorphically equivalent vertices share exactly the same label-independent properties. For example, in figure 19(b1), nodes u and v are not structurally equivalent (u and v have neighbors with different labels). However, if we exchange their positions, we can re-label the rest of the nodes (by exchanging x and a, y and b, and z and c) such that the new labeled graph of figure 19(b2) is isomorphic to the original. Thus, automorphic equivalence is a weaker condition than structural equivalence.

—*Regular equivalence: If $G = (V, E)$ is a connected graph and $\equiv$ is an equivalence relation on V, then $\equiv$ is a regular equivalence iff [Borgatti and Everett 1989]*

$$\forall a, b, c \in V, a \equiv b \Leftrightarrow \begin{cases} \textit{(i)}\ (a, c) \in E \Rightarrow \exists d \in V \textit{ such that } (b, d) \in E \textit{ and } d \equiv c \\ \textit{(ii)}\ (c, a) \in E \Rightarrow \exists d \in V \textit{ such that } (d, b) \in E \textit{ and } d \equiv c \end{cases}$$

Two actors u and v are regularly equivalent if they are equally related to equivalent others; thus, the definition is recursive. Figure 19(c) shows an example with three regularly-equivalent classes, each of which connects to a particular subset of classes (e.g., a "child" connects to a "husband" and a "wife.")

Structural equivalence has the strongest constraints, while regular equivalence has the weakest. However, regular equivalence is the hardest to compute, and is the equivalence of most interest to sociologists.

Practical computation of these equivalence classes can be computationally expensive [Hanneman and Riddle 2005], so the definitions are usually relaxed while analyzing real-world graphs. The following heuristics are often used:

—*Computing structural equivalence:* The correlation coefficient between actors is often used to measure the degree of structural equivalence between actors.

—*Computing automorphic equivalence:* Automorphic equivalence classes can be approximated using inter-actor distances: for each actor, we form a *sorted* vector of its distances to other actors. Two automorphic actors should have exactly the same distance profiles, and two "almost-automorphic" actors should have "similar" profiles. This idea is encapsulated in a heuristic which computes Euclidean distances between distance profiles, and clusters actors with low profile distances.

—*Computing regular equivalence:* Heuristics are used to compute some similarity measure between actors. However, irrespective of the similarity metric used, finding equivalent actors essentially reduces to a problem of clustering actors

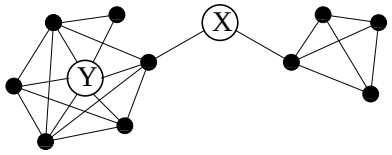


Fig. 20. *Two concepts of social capital:* Node X has importance because it bridges the structural hole between the two clusters. Node Y is in the middle of a dense web, which provides easy access to reliable information; thus Y also has a good position in the network.

based on the (perhaps thresholded) similarity matrix. One such technique uses *Tabu search* [Glover 1989] to group actors (forming blocks in the matrix) so that the variance within the blocks is minimized.

Finding such equivalence classes is a major goal of social network analysis, and any advances in algorithms or heuristics can have a major impact in this field.

4.5 Social capital

Social capital is essentially the idea that better connected people enjoy higher returns on their efforts. An individual occupying some special location in the social network might be in a position to broker information or facilitate the work of others or be important to others in some fashion. This importance could be leveraged to gain some profit. However, the problem is: what does "better connected" mean?

In general, there are two viewpoints on what generates social capital (Figure 20):

—*Structural holes:* Weak connections between groups are holes in the social structure, and create an advantage for individuals whose relationships span the holes [Burt 1992]. Such individuals get lots of brokerage opportunities, and can control the flow of information between groups to their benefit.

—*Network closure:* This is the view that networks with lots of connections are the source of social capital [Coleman 1988]. When the social network around an actor Y is dense, it means that information flow to Y is quick and usually reliable. Also, the high density means that no one around Y can escape the notice of others; hence, everyone is forced to be trustworthy (or face losing reputation). Thus, it is less risky for Y to trust others, and this can be beneficial to him/her.

Burt [2001] finds that these two viewpoints might not be totally at odds with each other. If a group has high closure but low contacts across holes, the group is cohesive but has only one perspective/skill. Low closure but high contacts across holes leads to disintegrated group of diverse perspectives. The best performance is achieved when both are high. Thus, structural holes and network closure supplement each other.

4.6 Recent research directions

Social Network Analysis has been used to analyze many networks, from organizational to networks of drug use to networks of friendship in schools, and many others. Now, SNA is moving in new directions, some of which are discussed below.

4.6.1 *Terrorist and covert networks.* Recent events have focused attention on the mapping and analysis of terrorist networks. There are several problems unique to this setting [Sparrow 1991], primarily due to lack of information:

—*Incompleteness* due to missing nodes and edges.

—*Fuzzy boundaries* due to not knowing whom to include or exclude form the mapped network.

—*Dynamism* of the network due to changing edges and the strengths of association of the nodes.

Baker and Faulkner [1993] find that while the need for efficiency drives the structure of legal social networks, secrecy is the paramount concern in illegal networks. When the information processing needs are low, it leads to decentralized structures which protect the "ringleaders." However, for high information processing situations, the leaders must necessarily be at the core of the illegal network, increasing their vulnerability.

Krebs [2001] tries to map the social network of the September-11 hijackers and some of their contacts, using only public data sources. He finds a very sparse network where many hijackers on the same team were far away from each other on the network. Coordination between far-off nodes is achieved via shortcuts in the network (as in the Small-World model of Watts and Strogatz [1998]). Trusted prior contacts kept the cells interconnected. Dombroski et al. [2003] use the high-clustering-coefficient and other properties of typical networks to estimate missing information in covert networks.

4.6.2 *The Key Player problem.* Who are the *most important* actors in a given social network? Borgatti [2002] defines two "key-player" problems:

—(KPP-1) Find a set of k nodes whose removal maximally disrupts/disconnects the network. These individuals might be targeted for immunization to prevent an infection from becoming an epidemic.

—(KPP-2) Find a set of k nodes which are maximally connected to the rest of the network. These individuals could be targeted to diffuse information in a social network in the shortest possible time.

The "importances" of nodes are related, and choosing one node to be part of the top-k set changes the importances of others. Thus, finding the best set requires combinatorial optimization, which is very costly. Borgatti suggests using a greedy heuristic to solve this. However, formal error bounds on such heuristics are still not available.

4.7 Differences from Graph Mining

We have seen in the previous paragraphs that Graph Mining and Social Network Analysis share many concepts and ideas. However, there are important differences too, the primary one being that of network size. Social networks are in general small, with the larger studies considering a few hundred nodes only. Graph Mining datasets, on the other hand, typically consist of hundreds of thousands of nodes and millions of edges. This difference in scale leads to many effects:

—*Power laws:* As we have seen in Section 2.1, power laws show up in a variety of situations in Graph Mining datasets, whereas they seem to be absent in almost all Social Network literature. This should be, in part, due to the fact that power law patterns can be observed reliably only in large datasets.

—*Focus on computation costs:* For small networks, the existence of efficient algorithms is not an issue; even brute force algorithms can finish in a reasonably short time. However, some of the algorithms in use in SNA (such as combinatorial optimizations) become impractical for large datasets.

There has also been a difference in problems attracting research interest. The Graph Mining community does not seem to have worked on anything equivalent to social "roles" or the "power" of nodes. This might be due to the different semantics of the datasets in use. However, as newer and larger social network datasets (such as the "who-trusts-whom" dataset from `epinions.com`, the "who-reads-whose-weblog" dataset from blogspace, or the "who-knows-whom" dataset from `friendster.com`) become available, we might see a confluence of research in these two communities.

5. OTHER RELATED WORK

Several topics are closely related to graph mining, but have a different focus. Relational learning looks at the graph formed by interlinked relations in a database, and attempts to find patterns in it. Studies of rumor or viral propagation in a network look for key properties of the network which determine the susceptibility to epidemics. New graph navigation algorithms try to devise graphs so that local routing decisions can lead to nearly-optimal global routes. These and other issues are discussed below.

5.1 Relational learning

Relational data mining has attracted a lot of research interest recently [Džeroski and Lavrač 2001]. While traditional data mining looks for patterns in a single database table, relational mining also uses the structure of linkage between multiple tables/relations. For example, given a database of movies, actors, awards and the labeled links between them (ie., a graph), McGovern and Jensen [2003] find the patterns associated with movies being nominated for awards. The patterns themselves can be described as subgraphs with some relations as the nodes and some links between these relations as the edges.

Finding such patterns involves searching through a space of possible hypotheses, and we can do this search exhaustively or heuristically. One essential ingredient is the *pruning* of search paths which are not expected to lead to the solution. A widely used technique is Inductive Logic Programming (ILP), where patterns are expressed as logic programs. An advantage of this approach is the ability to easily incorporate background knowledge specific to the problem in the form of logic clauses. An alternative technique involves converting the relational data into a flat propositional format and then using well-known data mining tools on this flat relation; however, this conversion is non-trivial. In both cases, we run into efficiency concerns: the space of possible hypotheses is much larger than the case when we have a single relation, and searching in this space can be very costly. For

example, even checking for the validity of a clause in ILP can be a costly affair, and solving a logic program typically involves checking the validity of many clauses.

Relational learning is a broad topic, and its details are beyond the scope of this work. Here, we only point out differences from Graph Mining :

—Relational learning typically focuses on finding small structures/patterns at the local level (such as a chemical motif that occurs frequently in carcinogenic compounds [Dehaspe et al. 1998]), while Graph mining looks far more at the global structure (for example, the distribution of eigenvalues of the entire graph).

—Graph mining has been (at least till the present) more about the topological structure and properties of a graph, while the graphs used in Relational learning usually have labeled nodes and edges and their semantics are important.

5.2 Finding frequent subgraphs

The mining of frequent patterns was first introduced in a databases context by Agrawal and Srikant [Agrawal and Srikant 1994], and is possibly one of the most popular data mining techniques. Recently, these ideas have been extended and applied to large graph datasets, to find the most common patterns or "motifs" hidden in the graph structure, to compress the graph dataset, and for many other problems. Below, we will discuss several of these methods.

5.2.1 *APRIORI-like algorithms.* Frequently occurring subgraphs in a large graph or a set of graphs could represent important motifs in the data. However, finding such motifs involves solving the graph and subgraph isomorphism problems, for which efficient solutions are not known (and subgraph isomorphism is known to be NP-complete). Most algorithms follow the general principle of the Apriori algorithm [Agrawal and Srikant 1994] for association rule mining. Inokuchi et al. [2000] develop an Apriori-inspired algorithm called `AGM`, where they find a "canonical code" for any adjacency matrix and use these canonical codes for subgraph matching. However, this suffers from computational intractability when the graph size becomes too large. Kuramochi and Karypis [2001] propose the `FSG` algorithm which also has the same flavor: starting with frequent subgraphs of 1 and 2 nodes, it successively generates larger subgraphs which still occur frequently in the graph. The algorithm expects a graph with colored edges and nodes; our graphs are a special case where all nodes and edges have only one color. However, it also needs to solve the graph and subgraph isomorphism problems repeatedly, and this is very slow and inefficient for graphs with only one color. Yan and Han [2002] propose using a different canonical code based on depth-first search on subgraphs, and report faster results using this coding scheme.

On graphs where vertex coordinates are available, a more constrained version of this problem requires finding frequent *geometric* subgraphs. In this case, subgraph matching involves both topological and layout matching. Kuramochi and Karypis [2002] find an algorithm that finds frequent geometric subgraphs that can be rotation, scaling and translation invariant. This extra constraint allows the algorithm to finish in polynomial time.

5.2.2 *"Difference from random" algorithm.* Milo et al. [2002] use another approach to find "interesting" motifs in a given graph. They define motifs to be

"patterns that recur more frequently [*in a statistically significant sense*] in the real network than in an ensemble of randomized networks" (italics added). Each randomized network is generated so that each node in it has the same in-degree and out-degree as the corresponding node in the real network. However, this method assumes that matching the in and out degrees of a graph gives a good model of the graph; the motifs found under this assumption might not be statistically frequent if we used a better graph model.

5.2.3 *Greedy algorithm.* Holder et al. [1994] try to solve a related but slightly different problem, that of compressing graphs using frequently occurring subgraphs. The subgraphs are chosen to minimize the minimum description length of the entire graph. They also allow *inexact matching* of subgraphs by assigning a cost to each "distortion" like deletion, insertion or substitution of nodes and edges. However, to avoid the excessive computational overhead, they use a (suboptimal) greedy beam search.

5.2.4 *Using Inductive Logic Programming (ILP).* Instead of defining a subgraph as just a labeled graph topology, Dehaspe and Toivonen [1999] use ILP to allow first order predicates in the description of frequent subgraphs. Their `WARMR` system lets many subgraphs have one succinct description, and can reveal a higher-order pattern than just the simple "propositional" subgraphs. However, this involves checking for equivalence of different first-order clauses, which is NP-complete. Nijssen and Kok [2001] use a weaker equivalence condition in their `FARMAR` system to speed up the search. Still, finding first-order patterns is harder than finding propositional patterns, and it is unclear how fast such techniques will work on very large graph datasets.

5.3 Navigation in graphs

The participants in Milgram's experiment [Travers and Milgram 1969] were able to build a chain to an unknown target individual, even though they knew only a few individuals in the full social network. We can navigate to websites containing the information we need, in spite of the immense size of the web. Such facts imply that large real-world graphs can be *navigated* with ease. In the following paragraphs, we will discuss methods of navigation that can be employed, and why real world graphs seem to be so easy to navigate.

5.3.1 *Methods of navigation.* Some common methods for navigating large graphs include crawling, focused crawling, guided search based on power laws, and gossip protocols. We discuss each of these below.

Crawling: One question of interest in a graph is: given a starting node s in the graph, how can we reach some randomly assigned target node t? One technique involves having a search engine "crawl" the graph and store its data in a searchable form in some centralized system. Queries regarding the target node t can then be directed to this central server; this is the technique used by Web search engines like *Google* [Brin and Page 1998] or *CiteSeer* [Giles et al. 1998].

Focused crawling: How should we crawl pages while specifically looking for a par-

ticular topic? Chakrabarti et al. [2002; 1999; 1999] propose a machine-learning approach to the problem using two "learners" working in tandem: (1) a "baseline learner" which can compute the degree of relevance of a given webpage to the required topic, and (2) an "apprentice learner" which computes the chances that a particular hyperlink points to a relevant webpage. Such a technique prevents the crawler from wasting effort on irrelevant portions of the Web; thus, the crawls can be conducted more frequently and the crawled data can be kept fresher.

Guided search using the power-law degree distribution: In case such a "directory" of nodes is not available, an alternative is to do a guided search. This involves making decisions based on local information only. Adamic et al. [2001] use a message-passing system to search efficiently for some target data in the Gnutella network. The start node polls all its neighbors to see if any of them contains the required data, and if not, the search is forwarded to the neighbor with the highest degree. This neighbor now searches for the data among its neighbors, and so on. Full backtracking is implemented to prevent getting stuck in loops, and nodes do not get to see the same query twice. This technique is based on two ideas: (1) In scale-free networks, the path to a node of very high degree is usually short, and (2) the highest-connected nodes can (presumably) quickly spread the query all other nodes in the network. However, this technique still requires $O(N)$ query messages to be sent between nodes, even though the path to the node containing the required data may be small.

Gossip protocols: Kempe et al. [2001] take a different approach to search for "resources" in a network. They use a *gossip protocol* to spread information throughout a network about the availability of a "resource" at a particular node. Nodes share information with each other; the probability of communication between nodes u and v is a non-uniform inverse-polynomial function. However, this work assumes that any pair of nodes can communicate between themselves, and does not take the underlying graph structure into account.

5.3.2 *Relationship between graph topology and ease of navigation.* We will first discuss how a "good-for-navigation" graph can be designed, and then briefly touch upon some work investigating the reasons behind the ease of navigation on the WWW.

5.3.2.1 *Designing "good-for-navigation" graphs.* Can we build a graph so that local routing decisions can be used to reach any target node via a short path? This is clearly what is happening in the social network of Milgram's experiment [Travers and Milgram 1969; Milgram 1967]: not only does a short path exist between two randomly chosen people in the network, but such paths can also be *found* by the people, who forward the letters based only on the (local) information they have about the network structure. The problem has been studied in several forms, as described below.

2D Grid: Kleinberg [1999b] considers a model similar to that of Watts and Strogatz [1998], but with a 2D lattice instead of a ring. Each node is connected to its neighbors in the lattice, but also has some *long-range* contacts; the probability of

such a contact decreases exponentially with distance: $P(u, v) \propto d(u, v)^{-r}$. Based on the value of r, there are several cases:

—When $r = 2$, local routing decisions can lead to a path of expected length $O(\log N)$.

—When $0 \leq r < 2$, a local routing algorithm cannot use any "clues" provided by the geometry of the lattice, due to which path lengths are polynomial in N. This is in spite of the fact that (say for $r = 0$) there *exist* paths of length bounded by $\log N$ with high probability.

—When $r > 2$, long-range contacts are too few. Thus, the speed of moving towards the target node is too slow, leading to path lengths polynomial in N.

Thus, local routing decisions lead to a good path *only* when $r = 2$. The important points are twofold:

(1) Minimizing the minimum expected number of steps from source to target is not necessarily the same as minimizing the diameter of the network.

(2) In addition to having short paths, a network should also contain some latent structural clues to help make good routing decisions based on local information.

Hierarchies of attributes: Watts et al. [2002] and (independently) Kleinberg [2001] also proposed a different model to explain the goodness of local routing choices. The basic idea is that each person has a set of attributes, and is more likely to know people with similar attributes.

—Each person (node) has an individual *identity*, consisting of H attributes such as location, job, etc.

—Each attribute leads to a hierarchy/tree of nodes. For example, everyone in Boston is one cluster/leaf, everyone in NY is another leaf, and the Boston and NY leaves are closer by tree distance than, say, the Boston and LA leaves. Similar hierarchies exist for each attribute. Note that these hierarchies are *extra information*, and are unrelated to the social network itself.

—Two nodes u and v have a edge between them with a probability depending on how close they are in the attribute hierarchies. Specifically, let $d^a(u, v)$ be the height of the lowest common ancestor of nodes u and v in the hierarchy for attribute a. This measures the "distance" between the two nodes according to attribute a. We take the minimum distance over all hierarchies to be the "distance between u and v" $d(u, v)$. The probability of an edge (u, v) is exponential in this distance: $P(u, v) \propto e^{-\alpha d(u,v)}$.

The parameter α defines the structure of the social network; when it is large, we get isolated cliques, and the network looks more like a random graph as α increases. To pass on a message towards a prespecified target, everyone in the chain makes a local decision: he/she passes the letter to the node perceived to be closest to the target in terms of the minimum distance mentioned above.

Watts et al. observe that there is a wide range of α and H (the number of attributes) which lead to good routing decisions. In fact, they find that the best routing choices are made when $H = 2$ or 3, which agrees very well with certain

sociological experiments [Killworth and Bernard 1978]. Kleinberg [2001] extends this to cases where we can form *groups* of individuals, which need not be hierarchies.

However, neither Watts et al. nor Kleinberg appear to consider the effect of power-law degree distributions in such graphs. Also, the probability of edge (u, v) is equal to that of the reverse edge (v, u); this might not hold always hold in practice.

5.3.2.2 *Navigation in real-world graphs.* In the context of social networks, Milgram's experiment [Travers and Milgram 1969; Milgram 1967] shows the ability of people to choose "good" people to forward a letter to, so that the target can receive the letter in only a few steps. Dill et al. [2001] find similar behavior in the WWW. In Section 2.7.2, we discussed the basic structure of the WWW, consisting of several subgraphs with one Strongly Connected Component (`SCC`) each. The authors find that the `SCCs` of the different subgraphs are actually very well-connected between themselves, via the `SCC` of the entire WWW. This allows easy navigation of webpages: starting from a webpage, we progress to its `SCC`, travel via the `SCC` of the Web to the `SCC` of the target webpage, and from there onwards to the target.

5.4 Spread of viruses in graphs

Given a network, how do diseases/information/computer-viruses spread across it? Answering this would help devise global strategies to combat viruses, instead of the local "curing" done by most current anti-virus software. We will divide our discussion into three parts: viral propagation models, "epidemic thresholds," and immunization policies.

Viral propagation models: Epidemiological models have been used to study this problem [Bailey 1974], and two models have been borrowed from the disease propagation literature:

— *The SIR model:* Each node can be in one of three states: Susceptible(S) meaning that it is not diseased but might get infected later, Infective(I) meaning that it is currently infected and can pass on the disease, and Removed(R) meaning that it has recovered from the disease and is immune to further infections.

— *The SIS model:* Once an Infective node is cured, it goes back into the Susceptible state (instead of the Removed state of the SIR model).

Associated with each edge (i, j) is its rate of spreading infection, called the birth rate β_{ij}. Each Infective node u also has a rate of getting cured, called the death rate δ_u. Typically, these are assumed to be homogeneous, that is, $\beta_{ij} = \beta$ and $\delta_u = \delta$ for all nodes i, j and u.

The "epidemic threshold": The spread of infections depends not on the exact values of β and δ, but on their ration β/δ. A primary focus of research is finding a critical value for this ratio, called the "epidemic threshold" τ, above which an "epidemic" outbreak is possible. However, the term "epidemic" has different meanings under the SIR and SIS models, and so we will look at each separately.

Epidemic threshold in the SIR model: In the SIR model, an infection becomes an epidemic when the initial outbreak spreads to a significant size. Callaway et al. [2000]

modeled the viral outbreak as a percolation problem (common in the physics literature), with the epidemic threshold occurring exactly at the percolation transition; the formation of a giant component beyond the transition corresponds to the infection becoming an epidemic. They find exact solutions only for a few cases, but the equations can be simulated for any set of conditions. However, given a network $\mathcal{G}$, their formulation considers *all* networks with the degree distribution of $\mathcal{G}$; the actual epidemic threshold for $\mathcal{G}$ might be different from the results they obtain.

Epidemic threshold in the SIS model: For the SIS model, an infection becomes an epidemic when it does *not* die out over time (called an *endemic* disease). Assuming homogeneous birth and death rates (ie., $\beta_{ij} = \beta$ and $\delta_u = \delta$ for all nodes i, j and u) and an Erdös-Rényi network, Kephart and White [1991; 1993] find that:

$$\frac{\beta}{\delta} > \tau_{KW} = \frac{1}{E[k]} \text{ implies that an epidemic is possible} \tag{40}$$

where $E[k]$ is the expected (average) degree of the nodes in the graph, and τ_{KW} is the epidemic threshold. Pastor-Satorras and Vespignani [2001b; 2001a; 2002a] find a formula for general graphs; their epidemic threshold is:

$$\tau_{SV} = \frac{E[k]}{E[k^2]} \tag{41}$$

where $E[k^2]$ represents the expected value of the squared-degree of nodes. An interesting conclusion is that for certain infinite scale-free networks, the epidemic threshold is zero. Thus, *any* initial infection can be expected to grow into an epidemic! Boguñá and Pastor-Satorras [2002] consider viral propagation on undirected *Markovian* networks, where the connectivity of a node is correlated with the degrees of its immediate neighbors. While correlations might exist in real networks, they may not necessarily be just Markovian.

However, none of the above models use the exact topology of the network under consideration; all graphs with the same degree distribution are treated equally. Wang et al. [2003] find that the epidemic threshold depends critically on the exact topology of the graph:

$$\tau_{WCWF} = \frac{1}{\lambda} \tag{42}$$

where λ is the largest eigenvalue of the adjacency matrix of the graph (considering a connected symmetric graph).

Immunization policies: Another related topic has been on finding the right immunization policy. Pastor-Satorras and Vespignani [2002b] find that randomly selecting nodes for immunization performs much worse than "targeted" immunization, which selects the nodes with the highest connectivity. This is as expected; removing the highest-degree nodes quickly disconnects the graph [Palmer et al. 2002; Albert et al. 2000; Bollobás et al. 2003], preventing the spread of infection.

5.5 Using social networks in other fields

5.5.1 *Viral Marketing.* Traditional mass marketing techniques promote a product indiscriminately to all potential customers. Better than that is direct marketing,

where we first attempt to select the most profitable customers to market to, and then market only to them. However, this only considers each person in isolation; the effects of one person's buying decisions on his/her neighbors in the social network are not considered. *Viral marketing* is based on the idea that considering social "word-of-mouth" effects might lower marketing costs.

Domingos and Richardson [2001] model the problem as finding a boolean vector of whether to market to a person or not, such that the expected profit over the entire network is maximized. The key assumption is that each person's decisions on whether to buy a product or not are independent of the entire network, *given* the decisions of his/her neighbors. However, this formalism has poor scalability. The same authors make linearity assumptions to make the problem tractable for large graphs, and also allow the exact amount of discount offered to each person to be optimized [Richardson and Domingos 2002]. The effect of a person on his neighbors might be mined from collaborative filtering systems or knowledge-sharing sites such as `www.epinions.com`. This is a very promising area of research, and is of prime interest to businesses.

5.5.2 *Recommendation systems.* Collaborative filtering systems have been widely used to provide recommendations to users based on previous information about their personal tastes. Examples include the *EachMovie* system to recommend movies (`www.research.compaq.com/src/eachmovie/`) and the product recommendation system of `www.amazon.com`. Such systems have been combined with social networks in the *ReferralWeb* system [Kautz et al. 1997]. Given a particular topic, it tries to find an expert on that topic *who is related to the user by a short path in the social network*. This *path of referrals* might give the user a measure of trust in the recommended expert; it might also encourage the expert to answer the user's query.

6. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

Several questions remain unanswered, or partially answered at best. These are rich topics for future investigation. Most of the work on graph mining has concentrated on undirected/directed self-graphs, where the endpoints of each edge are from the same set of nodes. Less attention has been paid to the modeling of real-world bipartite graphs, where the edge endpoints belong to two different sets of nodes. An example is that of web access patterns, where each edge connects a user to a website.

Another topic of interest is that of *weighted* graphs, where each edge has an associated weight. This weight could signify the strength of the link; for example, each link in the Internet router graph has an associated *bandwidth* specifying the maximum speed of communication over the link. Neglecting this weight might lead to incorrect estimates of the stress distribution and inaccurate identification of the bottlenecks or choke points in the graph.

The nodes and edges of a graph can also be labeled, and the relationships between the labels of connected nodes/edges provides extra information for data mining. For example, a social network could have edge labels specifying the relationship between the nodes at the endpoints (such as kinship, business relationship or social contact). The label of a node in the network would specify its "social role." This could be

extended by allowing each node (person) to have multiple labels/roles (such as wife and professor). Such graphs could be thought of as the superimposition of several graphs over the same set of nodes, each representing one type of linkage between nodes (such as a kinship graph overlaid on a financial transactions graph).

When a graph has multiple types of nodes, one interesting pattern to look for is "(dis)assortative mixing," or, the selective linking between nodes of (different) same types [Newman 2003]. For example, assortative mixing by race is often observed in social networks. On the other hand, disassortative mixing is often found in "informational" networks: for example, in the Internet topology, customers talk to ISPs (and rarely to each other) and ISPs talk to the backbone routers. This could be extended to patterns among nodes within a 2-hop radius, and so on.

A problem plaguing research on these topics is the lack of relevant real-world data. For example, the bandwidth of links between Internet routers is rarely made public. However, new studies might be able to collect/infer such data. All of the aforementioned topics are fertile areas for future work.

7. CONCLUSIONS

Naturally occurring graphs, perhaps collected from a variety of different sources, still tend to possess several common patterns. The most common of these are:

—Power laws, in degree distributions, in PageRank distributions, in eigenvalue-versus-rank plots and many others,

—Small diameters, such as the "six degrees of separation" for the US social network, 4 for the Internet AS level graph, and 12 for the Router level graph, and

—"Community" structure, as shown by high clustering coefficients, large numbers of bipartite cores, etc.

Graph generators attempt to create synthetic but "realistic" graphs, which can mimic these patterns found in real-world graphs. Recent research has shown that generators based on some very simple ideas can match some of the patterns:

—*Preferential attachment:* Existing nodes with high degree tend to attract more edges to themselves. This basic idea can lead to power-law degree distributions and small diameter.

—*"Copying" models:* Popular nodes get "copied" by new nodes, and this leads to power law degree distributions as well as a community structure.

—*Constrained optimization:* Power laws can also result from optimizations of resource allocation under constraints.

—*Small-world models:* Each node connects to all of its "close" neighbors and a few "far-off" acquaintances. This can yield low diameters and high clustering coefficients.

These are only some of the models; there are many other models which add new ideas, or combine existing models in novel ways. We have looked at many of these, and discussed their strengths and weaknesses. In addition, we discussed the recently proposed R-MAT model, which can match most of the graph patterns for several real-world graphs.

While a lot of progress has been made on answering these questions, a lot still needs to be done. More patterns need to be found; though there is probably a point of "diminishing returns" where extra patterns do not add much information, we do not think that point has yet been reached. Also, typical generators try to match only one or two patterns; more emphasis needs to be placed on matching the entire gamut of patterns. This cycle between finding more patterns and better generators which match these new patterns should eventually help us gain a deep insight into the formation and properties of real-world graphs.

REFERENCES

Adamic, L. A. and Huberman, B. A. 2000. Power-law distribution of the World Wide Web. *Science 287*, 2115.

Adamic, L. A. and Huberman, B. A. 2001. The Web's hidden order. *Communications of the ACM 44,* 9, 55–60.

Adamic, L. A., Lukose, R. M., Puniyani, A. R., and Huberman, B. A. 2001. Search in power-law networks. *Physical Review E 64,* 4, 046135 1–8.

Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In *International Conference on Very Large Data Bases*. Morgan Kaufmann, San Francisco, CA.

Aiello, W., Chung, F., and Lu, L. 2000. A random graph model for massive graphs. In *ACM Symposium on Theory of Computing*. ACM Press, New York, NY, 171–180.

Aiello, W., Chung, F., and Lu, L. 2001. Random evolution in massive graphs. In *IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

Albert, R. and Barabási, A.-L. 2000. Topology of evolving networks: local events and universality. *Physical Review Letters 85,* 24, 5234–5237.

Albert, R. and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics 74,* 1, 47–97.

Albert, R., Jeong, H., and Barabási, A.-L. 1999. Diameter of the World-Wide Web. *Nature 401*, 130–131.

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

Alon, N. 1998. Spectral techniques in graph algorithms. In *Lecture Notes in Computer Science 1380*, C. L. Lucchesi and A. V. Moura, Eds. Springer-Verlag, Berlin, 206–215.

Alon, N., Yuster, R., and Zwick, U. 1997. Finding and counting given length cycles. *Algorithmica 17,* 3, 209–223.

Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E. 2000. Classes of small-world networks. *Proceedings of the National Academy of Sciences 97,* 21, 11149–11152.

Baeza-Yates, R. and Poblete, B. 2003. Evolution of the Chilean Web structure composition. In *Latin American Web Congress*. IEEE Computer Society Press, Los Alamitos, CA.

Bailey, N. T. J. 1974. *The Mathematical Theory of Infectious Diseases and its Applications*, 2nd ed. Charles Griffin, London.

Baker, W. E. and Faulkner, R. R. 1993. The social organization of conspiracy: Illegal networks in the Heavy Electrical Equipment industry. *American Sociological Review 58,* 6, 837–860.

Bar-Yossef, Z., Kumar, R., and Sivakumar, D. 2002. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, PA.

Barabási, A.-L. 2002. *Linked: The New Science of Networks*, First ed. Perseus Books Group, New York, NY.

Barabási, A.-L. and Albert, R. 1999. Emergence of scaling in random networks. *Science 286*, 509–512.

Barabási, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., and Vicsek, T. 2002. Evolution of the social network of scientific collaborations. *Physica A 311*, 590–614.

Beirlant, J., de Wet, T., and Goegebeur, Y. 2005. A goodness-of-fit statistic for Pareto-type behaviour. *Journal of Computational and Applied Mathematics 186,* 1, 99–116.

Ben-Hur, A. and Guyon, I. 2003. Detecting stable clusters using principal component analysis. In *Methods in Molecular Biology*, M. J. Brownstein and A. Khudorsky, Eds. Humana Press, Totowa, NJ, 159–182.

Berger, N., Borgs, C., Chayes, J. T., D'Souza, R. M., and Kleinberg, B. D. 2005. Competition-induced preferential attachment. *Combinatorics, Probability and Computing 14*, 697–721.

Berry, M. W. 1992. Large scale singular value computations. *International Journal of Supercomputer Applications 6,* 1, 13–49.

Bi, Z., Faloutsos, C., and Korn, F. 2001. The DGX distribution for mining massive, skewed data. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY, 17–26.

Bianconi, G. and Barabási, A.-L. 2001. Competition and multiscaling in evolving networks. *Europhysics Letters 54,* 4, 436–442.

Boguñá, M. and Pastor-Satorras, R. 2002. Epidemic spreading in correlated complex networks. *Physical Review E 66,* 4, 047104.

Boldi, P., Codenotti, B., Santini, M., and Vigna, S. 2002. Structural properties of the African Web. In *International World Wide Web Conference*. ACM Press, New York, NY.

Bollobás, B. 1985. *Random Graphs*. Academic Press, London.

Bollobás, B., Borgs, C., Chayes, J. T., and Riordan, O. 2003. Directed scale-free graphs. In *ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, PA.

Bollobás, B. and Riordan, O. 2002. The diameter of a scale-free random graph. Combinatorica.

Bonacich, P. 1987. Power and centrality: a family of measures. *American Journal of Sociology 92,* 5 (March), 1170–1182.

Borgatti, S. 2002. The key player problem. In *Proceedings of the National Academy of Sciences Workshop on Terrorism*. National Academy of Sciences, Washington DC.

Borgatti, S. and Everett, M. G. 1989. The class of all regular equivalences: algebraic structure and computation. *Social Networks 11*, 65–88.

Borgatti, S. and Everett, M. G. 1999. Models of core/periphery structures. *Social Networks 21*, 275–295.

Borgatti, S., Everett, M. G., and Freeman, L. C. 1999. UCINET V User's Guide. Analytic Technologies.

Borgs, C., Chayes, J. T., Mahdian, M., and Saberi, A. 2004. Exploring the community structure of newsgroups (Extended Abstract). In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Brandes, U., Gaertler, M., and Wagner, D. 2003. Experiments on graph clustering algorithms. In *European Symposium on Algorithms*. Springer Verlag, Berlin, Germany.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30,* 1–7, 107–117.

Broder, A. Z., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. 2000. Graph structure in the web: experiments and models. In *International World Wide Web Conference*. ACM Press, New York, NY.

Bu, T. and Towsley, D. 2002. On distinguishing between Internet power law topology generators. In *IEEE INFOCOM*. IEEE Computer Society Press, Los Alamitos, CA.

Burt, R. S. 1992. *Structural Holes*. Harvard University Press, Cambridge, MA.

Burt, R. S. 2001. Structural holes versus network closure as social capital. In *Social Capital: Theory and Research*, N. Lin, K. S. Cook, and R. S. Burt, Eds. Aldine de Gruyter, Hawthorne, NY.

Callaway, D. S., Newman, M. E. J., Strogatz, S. H., and Watts, D. J. 2000. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters 85,* 25, 5468–5471.

Calvert, K. L., Doar, M. B., and Zegura, E. W. 1997. Modeling Internet topology. *IEEE Communications Magazine 35,* 6, 160–163.

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

Chakrabarti, D. 2004. AutoPart: Parameter-free graph partitioning and outlier detection. In *Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, Berlin, Germany.

Chakrabarti, D., Papadimitriou, S., Modha, D. S., and Faloutsos, C. 2004. Fully automatic Cross-associations. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Chakrabarti, D., Zhan, Y., and Faloutsos, C. 2004. R-MAT: A recursive model for graph mining. In *SIAM Data Mining Conference*. SIAM, Philadelphia, PA.

Chakrabarti, S. 1999. Recent results in automatic Web resource discovery. *ACM Computing Surveys 31,* 4, 17. Article Number 17.

Chakrabarti, S., Punera, K., and Subramanyam, M. 2002. Accelerated focused crawling through online relevance feedback. In *International World Wide Web Conference*. ACM Press, New York, NY, 148–159.

Chakrabarti, S., van den Berg, M., and Dom, B. E. 1999. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks 31,* 11–16, 1623–1640.

Chen, Q., Chang, H., Govindan, R., Jamin, S., Shenker, S., and Willinger, W. 2001. The origin of power laws in Internet topologies revisited. In *IEEE INFOCOM*. IEEE Computer Society Press, Los Alamitos, CA.

Clauset, A., Newman, M. E. J., and Moore, C. 2004. Finding community structure of very large networks. *Physical Review E 70*, 066111.

Coleman, J. S. 1988. Social capital in the creation of human capital. *American Journal of Sociology 94*, S95–S121.

Cooper, C. and Frieze, A. 2003. A general model of web graphs. *Random Structures and Algorithms 22,* 3, 311–335.

Cooper, C. and Frieze, A. 2004. The size of the largest strongly connected component of a random digraph with a given degree sequence. *Combinatorics, Probability and Computing 13,* 3, 319–337.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. 1992. *Introduction to algorithms*, 6th ed. MIT Press and McGraw-Hill Book Company, Cambridge, MA.

Cross, R., Borgatti, S., and Parker, A. 2002. Making invisible work visible: Using social network analysis to support strategic collaboration. *California Management Review 44,* 2, 25–46.

Crovella, M. and Taqqu, M. S. 1999. Estimating the heavy tail index from scaling properties. *Methodology and Computing in Applied Probability 1,* 1, 55–79.

de Solla Price, D. J. 1976. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science 27*, 292–306.

Dehaspe, L. and Toivonen, H. 1999. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery 3,* 1, 7–36.

Dehaspe, L., Toivonen, H., and King, R. D. 1998. Finding frequent substructures in chemical compounds. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Dill, S., Kumar, R., McCurley, K. S., Rajagopalan, S., Sivakumar, D., and Tomkins, A. 2001. Self-similarity in the Web. In *International Conference on Very Large Data Bases*. Morgan Kaufmann, San Francisco, CA.

Dombroski, M., Fischbeck, P., and Carley, K. M. 2003. Estimating the shape of covert networks. In *Proceedings of the 8th International Command and Control Research and Technology Symposium*.

Domingos, P. and Richardson, M. 2001. Mining the network value of customers. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. 2002. Pseudofractal scale-free web. *Physical Review E 65,* 6, 066122.

Dorogovtsev, S. N. and Mendes, J. F. 2003. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, Oxford, UK.

Dorogovtsev, S. N., Mendes, J. F., and Samukhin, A. N. 2000. Structure of growing networks with preferential linking. *Physical Review Letters 85,* 21, 4633–4636.

Dorogovtsev, S. N., Mendes, J. F., and Samukhin, A. N. 2001. Giant strongly connected component of directed networks. *Physical Review E 64*, 025101 1–4.

Doyle, J. and Carlson, J. M. 2000. Power laws, Highly Optimized Tolerance, and Generalized Source Coding. *Physical Review Letters 84,* 24 (June), 5656–5659.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., and Vinay, V. 1999. Clustering in large graphs and matrices. In *ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, PA.

Džeroski, S. and Lavrač, N. 2001. *Relational Data Mining*. Springer Verlag, Berlin, Germany.

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

Erdős, P. and Rényi, A. 1961. On the strength of connectedness of random graphs. *Acta Mathematica Scientia Hungary 12*, 261–267.

Everitt, B. S. 1974. *Cluster Analysis*. John Wiley, New York, NY.

Fabrikant, A., Koutsoupias, E., and Papadimitriou, C. H. 2002. Heuristically Optimized Trade-offs: A new paradigm for power laws in the Internet. In *International Colloquium on Automata, Languages and Programming*. Springer Verlag, Berlin, Germany, 110–122.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. 1999. On power-law relationships of the Internet topology. In *Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM Press, New York, NY, 251–262.

Feuerverger, A. and Hall, P. 1999. Estimating a tail exponent by modelling departure from a Pareto distribution. *The Annals of Statistics 27,* 2, 760–781.

Flake, G. W., Lawrence, S., and Giles, C. L. 2000. Efficient identification of Web communities. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Freeman, L. C. 1977. A set of measures of centrality based on betweenness. *Sociometry 40,* 1, 35–41.

Gibson, D., Kleinberg, J., and Raghavan, P. 1998. Inferring web communities from link topology. In *ACM Conference on Hypertext and Hypermedia*. ACM Press, New York, NY, 225–234.

Giles, C. L., Bollacker, K., and Lawrence, S. 1998. Citeseer: An automatic citation indexing system. In *ACM Conference on Digital Libraries*. ACM Press, New York, NY.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*. Vol. 99. National Academy of Sciences, Washington DC.

Glover, F. 1989. Tabu search – part 1. *ORSA Journal on Computing 1,* 3, 190–206.

Goh, K.-I., Oh, E., Jeong, H., Kahng, B., and Kim, D. 2002. Classificaton of scale-free networks. In *Proceedings of the National Academy of Sciences*. Vol. 99. National Academy of Sciences, Washington DC, 12583–12588.

Goldstein, M. L., Morris, S. A., and Yen, G. G. 2004. Problems with fitting to the power-law distribution. *The European Physics Journal B 41*, 255–258.

Govindan, R. and Tangmunarunkit, H. 2000. Heuristics for Internet map discovery. In *IEEE INFOCOM*. IEEE Computer Society Press, Los Alamitos, CA, 1371–1380.

Granovetter, M. S. 1973. The strength of weak ties. *The American Journal of Sociology 78,* 6 (May), 1360–1380.

Hanneman, R. A. and Riddle, M. 2005. Introduction to social network methods. http://faculty.ucr.edu/ hanneman/nettext/.

Hill, B. M. 1975. A simple approach to inference about the tail of a distribution. *The Annals of Statistics 3,* 5, 1163–1174.

Holder, L., Cook, D., and Djoko, S. 1994. Substructure discovery in the SUBDUE system. In *National Conference on Artificial Intelligence Workshop on Knowledge Discovery in Databases*. AAAI Press, Menlo Park, CA, 169–180.

Inokuchi, A., Washio, T., and Motoda, H. 2000. An apriori-based algorithm for mining frequent substructures from graph data. In *Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, Berlin, Germany.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, A.-L. 2000. The large-scale organization of metabolic networks. *Nature 407,* 6804, 651–654.

Kannan, R., Vempala, S., and Vetta, A. 2000. On clusterings – good, bad and spectral. In *IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

Karypis, G. and Kumar, V. 1998. Multilevel algorithms for multi-constraint graph partitioning. Tech. Rep. 98-019, University of Minnesota.

Kautz, H., Selman, B., and Shah, M. 1997. Referralweb: Combining social networks and collaborative filtering. *Communications of the ACM 40,* 3, 63–65.

Kempe, D., Kleinberg, J., and Demers, A. J. 2001. Spatial gossip and resource location protocols. In *ACM Symposium on Theory of Computing*. ACM Press, New York, NY.

Kephart, J. O. and White, S. R. 1991. Directed-graph epidemiological models of computer viruses. In *IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press, Los Alamitos, CA.

Kephart, J. O. and White, S. R. 1993. Measuring and modeling computer virus prevalence. In *IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press, Los Alamitos, CA.

Killworth, P. D. and Bernard, H. R. 1978. Reverse small world experiment. *Social Networks 1,* 2, 103–210.

Kleinberg, J. 1999a. Authoritative sources in a hyperlinked environment. *Journal of the ACM 46,* 5, 604–632.

Kleinberg, J. 1999b. The small-world phenomenon: an algorithmic perspective. Tech. Rep. 99-1776, Cornell Computer Science Department.

Kleinberg, J. 2001. Small world phenomena and the dynamics of information. In *Neural Information Processing Systems Conference*. MIT Press, Cambridge, MA.

Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 1999. The web as a graph: Measurements, models and methods. In *International Computing and Combinatorics Conference*. Springer, Berlin, Germany.

Krapivsky, P. L. and Redner, S. 2001. Organization of growing random networks. *Physical Review E 63,* 6, 066123 1–14.

Krebs, V. E. 2001. Mapping networks of terrorist cells. *Connections 24,* 3, 43–52.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., and Upfal, E. 2000. Stochastic models for the Web graph. In *IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. 1999. Extracting large-scale knowledge bases from the web. In *International Conference on Very Large Data Bases*. Morgan Kaufmann, San Francisco, CA.

Kuramochi, M. and Karypis, G. 2001. Frequent subgraph discovery. In *IEEE International Conference on Data Mining*. IEEE Computer Society Press, Los Alamitos, CA, 313–320.

Kuramochi, M. and Karypis, G. 2002. Discovering frequent geometric subgraphs. In *IEEE International Conference on Data Mining*. IEEE Computer Society Press, Los Alamitos, CA.

Leskovec, J., Chakrabarti, D., Kleinberg, J., and Faloutsos, C. 2005. Realistic, mathematically tractable graph generation and evolution, using Kronecker Multiplication. In *Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, Berlin, Germany.

Leskovec, J., Kleinberg, J., and Faloutsos, C. 2005. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Madeira, S. C. and Oliveira, A. L. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics 1,* 1, 24–45.

McGovern, A. and Jensen, D. 2003. Identifying predictive structures in relational data using multiple instance learning. In *International Conference on Machine Learning*. AAAI Press, Menlo Park, CA.

Medina, A., Matta, I., and Byers, J. 2000. On the origin of power laws in Internet topologies. In *Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM Press, New York, NY, 18–34.

Mihail, M. and Papadimitriou, C. H. 2002. On the eigenvalue power law. In *International Workshop on Randomization and Approximation Techniques in Computer Science*. Springer Verlag, Berlin, Germany.

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

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovshii, D., and Alon, U. 2002. Network motifs: Simple building blocks of complex networks. *Science 298*, 824–827.

Mitzenmacher, M. 2001. A brief history of generative models for power law and lognormal distributions. In *Proc. 39th Annual Allerton Conference on Communication, Control, and Computing*. UIUC Press, Urbana-Champaign, IL.

Montgomery, A. L. and Faloutsos, C. 2001. Identifying Web browsing trends and patterns. *IEEE Computer 34,* 7, 94–95.

Moody, J. 2001. Race, school integration, and friendship segregation in America. *American Journal of Sociology 107,* 3, 679–716.

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

Newman, M. E. J. 2005. Power laws, pareto distributions and Zipf's law. *Contemporary Physics 46*, 323–351.

Newman, M. E. J., Forrest, S., and Balthrop, J. 2002. Email networks and the spread of computer viruses. *Physical Review E 66,* 3, 035101 1–4.

Newman, M. E. J., Girvan, M., and Farmer, J. D. 2002. Optimal design, robustness and risk aversion. *Physical Review Letters 89,* 2, 028301 1–4.

Newman, M. E. J., Strogatz, S. H., and Watts, D. J. 2001. Random graphs with arbitrary degree distributions and their applications. *Physical Review E 64,* 2, 026118 1–17.

Nijssen, S. and Kok, J. 2001. Faster association rules for multiple relations. In *International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, San Francisco, CA.

Palmer, C., Gibbons, P. B., and Faloutsos, C. 2002. ANF: A fast and scalable tool for data mining in massive graphs. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Palmer, C. and Steffan, J. G. 2000. Generating network topologies that obey power laws. In *IEEE Global Telecommunications Conference*. IEEE Computer Society Press, Los Alamitos, CA.

Pandurangan, G., Raghavan, P., and Upfal, E. 2002. Using PageRank to characterize Web structure. In *International Computing and Combinatorics Conference*. Springer, Berlin, Germany.

Pastor-Satorras, R., Vásquez, A., and Vespignani, A. 2001. Dynamical and correlation properties of the Internet. *Physical Review Letters 87,* 25, 258701 1–4.

Pastor-Satorras, R. and Vespignani, A. 2001a. Epidemic dynamics and endemic states in complex networks. *Physical Review E 63,* 6, 066117 1–8.

Pastor-Satorras, R. and Vespignani, A. 2001b. Epidemic spreading in scale-free networks. *Physical Review Letters 86,* 14, 3200–3203.

Pastor-Satorras, R. and Vespignani, A. 2002a. Epidemic dynamics in finite size scale-free networks. *Physical Review E 65,* 3, 035108 1–4.

Pastor-Satorras, R. and Vespignani, A. 2002b. Immunization of complex networks. *Physical Review E 65,* 3, 036104 1–8.

Pennock, D. M., Flake, G. W., Lawrence, S., Glover, E. J., and Giles, C. L. 2002. Winners don't take all: Characterizing the competition for links on the Web. *Proceedings of the National Academy of Sciences 99,* 8, 5207–5211.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992. *Numerical Recipes in C*, 2nd ed. Cambridge University Press, Cambridge, UK.

Ravasz, E. and Barabási, A.-L. 2002. Hierarchical organization in complex networks. *Physical Review E 65*, 026112 1–7.

Redner, S. 1998. How popular is your paper? an empirical study of the citation distribution. *The European Physics Journal B 4*, 131–134.

Richardson, M. and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*. ACM Press, New York, NY, 61–70.

Schwartz, M. F. and Wood, D. C. M. 1993. Discovering shared interests using graph analysis. *Communications of the ACM 36,* 8, 78–89.

Simon, H. 1955. On a class of skew distribution functions. *Biometrika 42,* 3/4, 425–440.

Solé, R. V. and Montoya, J. M. 2001. Complexity and fragility in ecological networks. In *Proceedings of the Royal Society of London B*. Vol. 268. The Royal Society, London, UK, 2039–2045.

Sparrow, M. K. 1991. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks 13,* 3, 251–274.

Spielman, D. A. and Teng, S.-H. 1996. Spectral partitioning works: Planar graphs and finite element meshes. In *IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 96–105.

Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S., and Willinger, W. 2001. Network topologies, power laws, and hierarchy. Tech. Rep. 01-746, University of Southern California.

Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S., and Willinger, W. 2002. Network topology generators: Degree-based vs. structural. In *Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM Press, New York, NY.

Tauro, S. L., Palmer, C., Siganos, G., and Faloutsos, M. 2001. A simple conceptual model for the Internet topology. In *Global Internet*. IEEE Computer Society Press, Los Alamitos, CA.

Tibshirani, R., Walther, G., and Hastie, T. 2001. Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistical Society, B 63*, 411–423.

Travers, J. and Milgram, S. 1969. An experimental study of the Small World problem. *Sociometry 32,* 4, 425–443.

Tyler, J. R., Wilkinson, D. M., and Huberman, B. A. 2003. *Email as spectroscopy: Automated discovery of community structure within organizations*. Kluwer, B.V., Deventer, The Netherlands, 81–96.

van Dongen, S. M. 2000. Graph clustering by flow simulation. Ph.D. thesis, Univesity of Utrecht.

Virtanen, S. 2003. Clustering the Chilean Web. In *Latin American Web Congress*. IEEE Computer Society Press, Los Alamitos, CA.

Wang, Y., Chakrabarti, D., Wang, C., and Faloutsos, C. 2003. Epidemic spreading in real networks: An eigenvalue viewpoint. In *Symposium on Reliable Distributed Systems*. IEEE Computer Society Press, Los Alamitos, CA, 25–34.

Wasserman, S. and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK.

Watts, D. J. 2003. *Six Degrees: The Science of a Connected Age*, 1st ed. W. W. Norton and Company, New York, NY.

Watts, D. J., Dodds, P. S., and Newman, M. E. J. 2002. Identity and search in social networks. *Science 296*, 1302–1305.

Watts, D. J. and Strogatz, S. H. 1998. Collective dynamics of 'small-world' networks. *Nature 393*, 440–442.

Waxman, B. M. 1988. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications 6,* 9 (December), 1617–1622.

Weeks, M. R., Clair, S., Borgatti, S., Radda, K., and Schensul, J. J. 2002. Social networks of drug users in high-risk sites: Finding the connections. *AIDS and Behavior 6,* 2, 193–206.

Winick, J. and Jamin, S. 2002. Inet-3.0: Internet Topology Generator. Tech. Rep. CSE-TR-456-02, University of Michigan, Ann Arbor.

Wu, F. and Huberman, B. A. 2004. Finding communities in linear time: a physics approach. *The European Physics Journal B 38,* 2, 331–338.

Yan, X. and Han, J. 2002. gSpan: Graph-based substructure pattern mining. In *IEEE International Conference on Data Mining*. IEEE Computer Society Press, Los Alamitos, CA.

Yook, S.-H., Jeong, H., and Barabási, A.-L. 2002. Modeling the Internet's large-scale topology. *Proceedings of the National Academy of Sciences 99,* 21, 13382–13386.