Chapter 3

GRAPH MINING: LAWS AND GENERATORS

Deepayan Chakrabarti
Yahoo! Research
deepay@yahoo-inc.com

Christos Faloutsos
School of Computer Science
Carnegie Mellon University
christos@cs.cmu.edu

Mary McGlohon
School of Computer Science
Carnegie Mellon University
mmcgloho@cs.cmu.edu

Abstract *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. Many 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 gives 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.

Keywords: Power laws, structure, generators

C.C. Aggarwal and H. Wang (eds.), *Managing and Mining Graph Data*,
Advances in Database Systems 40, DOI 10.1007/978-1-4419-6045-0_3,

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 (business relationships, kinship, 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 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? Due to the ubiquity and wide applicability of graphs, 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

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 3.1. *Table of symbols*

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 [12, 81, 35].

The organization of this chapter 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. Table 3.1 lists the symbols we will use.

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 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. A large portion of the literature focuses on two major properties: power laws and small diameters. Our discussion will address both of these properties. For each pattern, 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. Next, we will discuss other patterns of interest, both in static snapshots of graphs and in evolving graphs. Finally, we discuss patterns specific to some well-known graphs, like the Internet and the WWW.

2.1 Power Laws and Heavy-Tailed Distributions

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) [43]. 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.

“Traditional” Power Laws.

Definition 3.1 (Power Law). *Two variables x and y are related by a power law when:*

$$y(x) = Ax^{-\gamma} \tag{3.1}$$

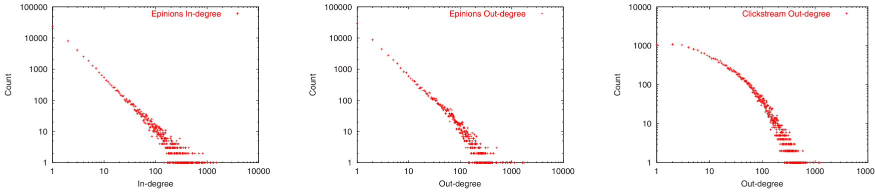
where A and γ are positive constants. The constant γ is often called the power law exponent.

Definition 3.2 (Power Law Distribution). *A random variable is distributed according to a power law when the probability density function (pdf) is given by:*

$$p(x) = Ax^{-\gamma}, \quad \gamma > 1, x \geq x_{min} \tag{3.2}$$

The extra $\gamma > 1$ requirement ensures that $p(x)$ can be normalized. Power laws with $\gamma < 1$ rarely occur in nature, if ever [66].

Skewed distributions, such as power laws, occur very often. In the Internet graph, the degree distribution follows such a power law [43]; that is, the count



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

Figure 3.1. *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 [34]). 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 [63]), which deviates from the power-law pattern.

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 [61]. The World Wide Web graph also obeys power laws [51]: 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 [23, 73]. Redner [76] shows that the citation graph of scientific literature follows a power law with exponent 3. Figures 3.1(a) and 3.1(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 3.1 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.

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 [45], (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 [66]. 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* $F(x)$:

$$F(x) = P(X \geq x) = \sum_{z=x}^{\infty} p(z) = \sum_{z=x}^{\infty} Az^{-\gamma} \tag{3.3}$$

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 $F(x)$:

$$\begin{aligned} & \int_x^\infty Az^{-\gamma}dz < F(x) < Ax^{-\gamma} + \int_x^\infty Az^{-\gamma}dz \\ \Rightarrow \quad & \frac{A}{\gamma-1}x^{-(\gamma-1)} < F(x) < Ax^{-\gamma} + \frac{A}{\gamma-1}x^{-(\gamma-1)} \\ \Rightarrow \quad & F(x) \, \mathrm{sim} \, x^{-(\gamma-1)} \end{aligned} \tag{3.4}$$

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 [45] find that it gives good unbiased estimates of γ.

5 *The Hill statistic:* Hill [48] gives an easily computable estimator, that seems to give reliable results [66]. However, it also needs to be told where the tail of the distribution begins.

6 *Fitting only to extreme-value data:* Feuerverger and Hall [44] 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 [31] 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. [15] derive a bias-corrected Jackson statistic for measuring goodness of fit of the data to a generalized Pareto distribution. Goldstein et al. [45] 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 $2.1 - 2.2$ [43], the Internet router graph with exponent $sim 2.48$ [43, 46], the in-degree and out-degree distributions of subsets of the WWW with exponents 2.1 and $2.38 - 2.72$ respectively [13, 54, 24], the in-degree distribution of the African web graph with exponent 1.92 [19], a citation graph with exponent 3 [76], distributions of website sizes and traffic [2], and many others. Newman [66] provides a comprehensive list of such work.

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. [75] 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.

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:

$$y(x = k) \propto e^{-k/\kappa} k^{-\gamma} \tag{3.5}$$

where $e^{-k/\kappa}$ is the exponential cutoff term and $k^{-\gamma}$ is the power law term. Amaral et al. [10] 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.

Lognormals or the "DGX" distribution Pennock et al. [75] 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. [17], who found that a discrete truncated lognormal (called the Discrete Gaussian Exponential or “DGX” by the authors) 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:

$$y(x = k) = \frac{A(\mu, \sigma)}{k} \exp\left[-\frac{(\ln k - \mu)^2}{2\sigma^2}\right] \quad k = 1, 2, \ldots \tag{3.6}$$

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 3.1(c)), telecommunications and other data.

Examples of deviations from power laws in the real world Several data sets have shown deviations from a pure power law [10, 75, 17, 62]: 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.

2.2 Small Diameters

Informal description:. Travers and Milgram [80] 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. [78] 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 [43].

  Definition 3.3 (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*

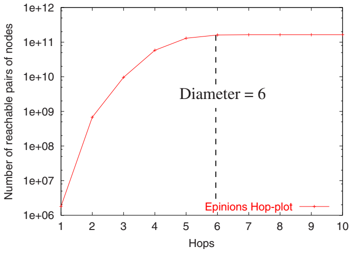


Figure 3.2. *Hop-plot and effective diameter* This is the hop-plot of the *Epinions* graph [34, 28]. We see that the number of reachable pairs of nodes flattens out at around 6 hops; thus the effective diameter of this graph is 6.

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 3.4 (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 [79].*

  Figure 3.2 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 [25].

- *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^2, 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. [43] 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.

Computational 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. [71] 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. [43] find an effective diameter of around 4 for the Internet AS level graph and around 12 for the Router level graph. Govindan and Tangmunarunkit [46] find a 97%-effective diameter of around 15 for the Internet Router graph. Broder et al. [24] 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. [8] find the average diameter of the webpages in the `nd.edu` domain to be 11.2. Watts and Strogatz [83] 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 [7] and table 3.1 of [65] list some such work.

2.3 Other Static Graph Patterns

Apart from power laws and small diameters, 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. Additionally, we observe structural patterns in the *edge weights* in static snapshots of graphs. We will explore these topics below.

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. [78] 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 [71, 9].

Computation issues. Calculating the "minimum cut-set" size is NP-hard, but approximate algorithms exist [49]. Computing the graph diameter is also costly, but fast randomized algorithms exist [71].

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 [71, 9, 78].

Patterns in weighted graphs.

Informal description. Edges in a graph often have *edge weights*. For instance, the size of packets transferred in a computer network, or length of phone calls (in seconds) in a phone-call network. These edge weights often follow patterns, as described in [59] and [5].

Detailed description. The first pattern we observe is the *Weight Power Law* (WPL). Let $E(t)$, $W(t)$ be the number of edges and total weight of a graph, at time t. They, they follow a power law

$$W(t) = E(t)^w$$

where w is the *weight* exponent.

The weight exponent w ranges from 1.01 to 1.5 for the real graphs studied in [59], which included blog graphs, computer network graphs, and political campaign donation graphs, suggesting that this pattern is universal to real social network-like graphs.

In other words, the more edges that are added to the graph, *superlinearly* more weight is added to the graph. This is counterintuitive, as one would expect the average weight-per-edge to remain constant or to increase linearly.

We find the same pattern for each node. If a node i has out-degree out_i, its out-weight $outw_i$ exhibits a "fortification effect"– there will be a power-law relationship between its degree and weight. We call this the *Snapshot Power Law* (SPL), and it applies to both in- and out- degrees.

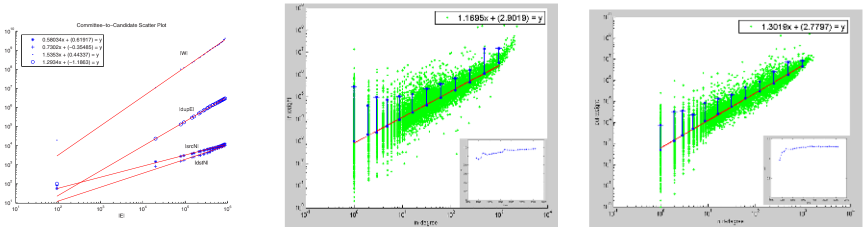
Specifically, at a given point in time, we plot the scatterplot of the in/out weight versus the in/out degree, for all the nodes in the graph, at a given time snapshot. Here, every point represents a node and the x and y coordinates are its degree and total weight, respectively. To achieve a good fit, we bucketize the x axis with logarithmic binning [64], and, for each bin, we compute the median y.

Examples in the real world. We find these patterns apply in several real graphs, including network traffic, blogs, and even political campaign donations. A plot of WPL and SPL may be found in Figure 3.3.

Several other weighted power laws, such as the relationship between the eigenvalues of the graph and the weights of the edges, may be found in [5].

Other metrics of measurement. We have discussed a number of patterns found in graphs, many more can be found in the literature. 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*. Some of these statistics include:

- *In and out degree correlation* The in and out degrees might be independent, or they could be (anti)correlated. Newman et al. [67] 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.



(a) WPL plot (b) inD-inW snapshot (c) outD-outW snapshot

Figure 3.3. Weight properties of the campaign donations graph: (a) shows all weight properties, including the densification power law and WPL. (b) and (c) show the Snapshot Power Law for in- and out-degrees. Both have slopes > 1 ("fortification effect"), that is, that the more campaigns an organization supports, the superlinearly-more money it donates, and similarly, the more donations a candidate gets, the more average amount-per-donation is received. Inset plots on (c) and (d) show iw and ow versus time. Note they are very stable over time.

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. [74] 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.

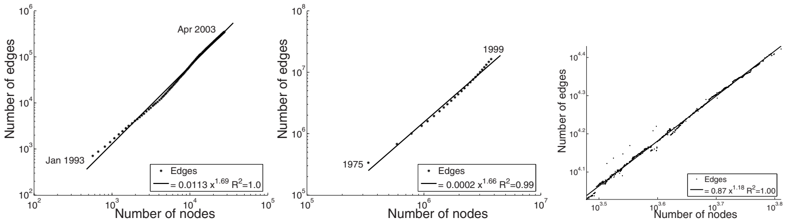
2.4 Patterns in Evolving Graphs

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. Some key patterns have emerged:

- *Densification Power Law:* Leskovec et al. [58] 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:

$$E(t) \propto N(t)^{\alpha} \quad 1 \leq \alpha \leq 2 \tag{3.7}$$

where the parameter α is called the Densification Power Law exponent, and remains stable over time. They also find that this "law" exists for



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

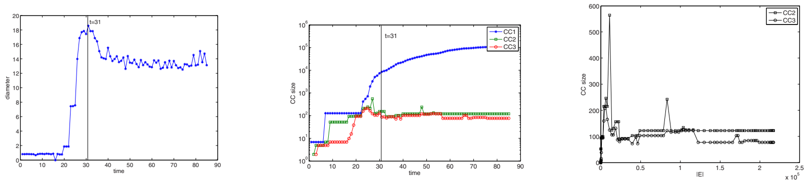
Figure 3.4. *The Densification Power Law* The number of edges $E(t)$ is plotted against the number of nodes $N(t)$ on log-log scales for (a) the arXiv citation graph, (b) the patents citation graph, and (c) the Internet Autonomous Systems graph. All of these grow over time, and the growth follows a power law in all three cases [58].

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 [14]. It also agrees with theoretical results showing that only a law like Equation 3.7 can maintain the power-law degree distribution of a graph as more nodes and edges get added over time [37]. Figure 3.4 demonstrates the densification law for several real-world networks.

- *Shrinking Diameters:* Leskovec et al. [58] also find that the effective diameters (definition 3.4) of graphs are actually *shrinking* over time, even though the graphs themselves are growing. This can be observed after the *gelling point*– before a certain point a graph is still building to normal properties. This is illustrated in Figure 3.5(a)– for the first few time steps the diameter grows, but it quickly peaks and begins shrinking.

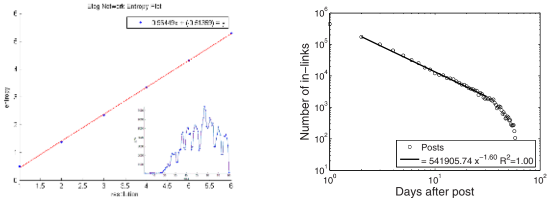
- *Component Size Laws* As a graph evolves, a giant connected component forms: that is, most nodes are reachable to each other through some path. This phenomenon is present both in random and real graphs. What is also found, however, is that once the largest component gels and edges continue to be added, the sizes of the *next-largest connected components* remain constant or oscillating. This phenomenon is shown in Figure 3.5, and discussed in [59].

- *Patterns in Timings:* There are also several interesting patterns regarding the timestamps of edge additions. We find that edge *weight* additions to a graph are bursty: over time, edges are not added to the overall graph uniformly over time, but are uneven yet self-similar [59]. We illustrate this in Figure 3.6. However, in the case of many graphs, timeliness of a particular *node* is important in its edge additions. As shown in [56], incoming edges to a blog post decay with a surprising power-law expo-



(a) Diameter(t) (b) Largest 3 components (c) CC2 and CC3 sizes

Figure 3.5. Connected component properties of Postnet network, a network of blog posts. Notice that we experience an early gelling point at (a), where the diameter peaks. Note in (b), a log-linear plot of component size vs. time, that at this same point in time the giant connected component takes off, while the sizes of the second and third-largest connected components (CC2 and CC3) stabilize. We focus on these next-largest connected components in (c).



(a) Entropy of edge additions (b) Decay of post popularity

Figure 3.6. Timing patterns for a network of blog posts. (a) shows the entropy plot of edge additions, showing burstiness. The inset shows the addition of edges over time. (b) describes the decay of post popularity. The horizontal axis indicates time since a post's appearance (aggregated over all posts), while the vertical axis shows the number of links acquired on that day.

nent of -1.5, rather than exponentially or linearly as one might expect. This is shown in Figure 3.6.

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.

2.5 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.

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 [26]. These domains can be considered as either a *stub* domain (which only carries traffic originating or

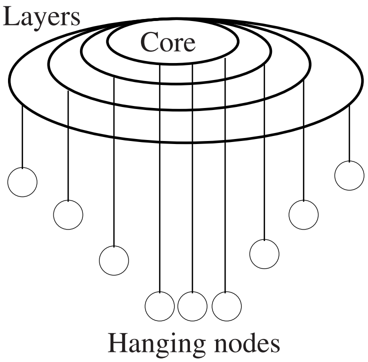


Figure 3.7. *The Internet as a "Jellyfish"* The Internet AS-level graph can be thought of as a core, surrounded by concentric layers around the core. There are many one-degree nodes that hang off the core and each of the layers.

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 3.7), found by Tauro et al. [79]. This consists of:

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

The World Wide Web (WWW). Broder et al. [24] find that the Web graph is described well by a "bowtie" structure (Figure 3.8(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) [11].

Dill et al. [33] extend this view of the Web by considering subgraphs of the WWW at different scales (Figure 3.8(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.

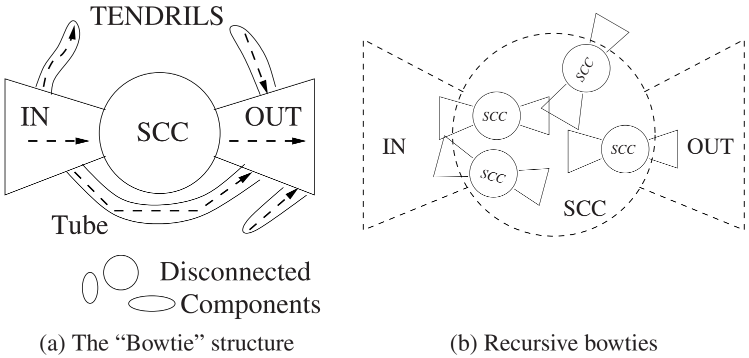
2 *Ease of navigation*: The SCC components of all these bowties are tightly connected together via the SCC of the whole Web graph. This provides a navigational backbone for the Web: starting from a webpage in one bowtie, we can click to its SCC, then go via the SCC of the entire Web to the destination 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.

We have discussed several patterns occurring in real graphs, and given some examples. Next, we would like to know, how can we re-create these patterns? What sort of mechanisms can help explain real-world behaviors? To answer these questions we turn to *graph generators*.

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.



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

Figure 3.8. *The "Bowtie" structure of the Web*: Plot (a) shows the 4 parts: `IN`, `OUT`, `SCC` and `TENDRILS` [24]. 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 [33].

Graph models and generators can be broadly classified into five categories:

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. This may be particularly relevant in the case of real-world networks that are constrained by geography. Together with the preferential attachment models, optimization-based models try to provide mechanisms that automatically lead to power laws.

4 *Tensor-based models:* Because many patterns in real graphs are self-similar, one can generate realistic graphs by using self-similar mechanisms through tensor multiplication.

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 this section. This is not a complete list, but we believe it includes most of the key ideas

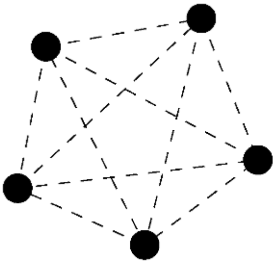


Figure 3.9. *The Erdős-Rényi model* The black circles represent the nodes of the graph. Every possible edge occurs with equal probability.

from the current literature. For each group of generators, we will try to provide the specific problem they aim to solve, followed by a brief description of the generator itself and its properties, and any open questions. We will also note variants on each major generator and briefly address their properties. While we will not discuss in detail all generators, we provide citations and a summary.

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 Erdős-Rényi model, which was the first to be studied thoroughly [40], and then we discuss modern variants of the model.

The Erdős-Rényi Random Graph Model.

Problem being solved. Graph theory owes much of its origins to the pioneering work of Erdős and Rényi in the 1960s [40, 41]. 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 3.9). This defines a *set* of graphs $G_{N,p}$, all of which have the same parameters (N, p).

Degree Distribution The probability of a vertex having degree k is

$$p_k = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{z^k e^{-z}}{k!} \quad \text{with } z = p(N-1) \qquad (3.8)$$

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

$$\begin{aligned} u &= \sum_{k=0}^{\infty} p_k u^k \\ &= e^{-z} \sum_{k=0}^{\infty} \frac{(uz)^k}{k!} \quad \text{(using Eq 3.8)} \\ &= e^{-z} e^{uz} = e^{z(u-1)} \end{aligned} \tag{3.9}$$

Thus, the fraction of nodes in the giant component is

$$S = 1 - u = 1 - e^{-zS} \tag{3.10}$$

Equation 3.10 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.

[1] $P(k) \propto k^{-2.255} / \ln k$; [18] 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. 3.16)

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 $z = -\frac{k}{l}$ [20]. 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:

$$CC_{random} = p = \frac{< k >}{N} \tag{3.11}$$

Open questions and discussion. It is hard to exaggerate the importance of the Erdős-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 3.8), 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 3.11, $\frac{CC_{random}}{<k>} = \frac{1}{N}$; but for many real-world graphs, $\frac{CC}{<k>}$ is independent of N (See figure 9 from [7]).

Thus, even though the Erdős-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).

Analytic techniques for studying random graphs involve generating functions. A good reference is by Wilf [85].

Generalized Random Graph Models. Erdős-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.

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 Erdős-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. [3] (a similar model is the *Power Law Out Degree* (PLOD) model of Palmer and Steffan [72]). There are two parameters: α and β. The number of nodes of degree k is given by e^{α}/k^{β}.

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

$$p_k \propto k^{-\beta} \tag{3.12}$$

where β is the power-law exponent.

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 $2 < \beta < \beta_0 \, \mathrm{sim} \, 3.48$, 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. [69]. Here, the probability that a node has k edges is given by

$$p_k = Ck^{-\gamma}e^{-k/\kappa} \tag{3.13}$$

where C, γ and κ are constants.

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 [3] when $\kappa \to \infty$.

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 [7].

Apart from PLRG and the exponential cutoffs model, some other related models have also been proposed, a notable model generalization being dot-product models [70]. Another important model is that of Aiello et al. [4], 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 will discuss later.

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 [30, 37]. We do not discuss these any further; the interested reader is referred to [69].

Open questions and discussion. Generalized random graph models retain the simplicity and ease of analysis of the Erdős-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 [7]). Also, many real world graphs (such as the WWW) exhibit the existence of communities of nodes, with stronger ties within the community than outside; 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.

Basic Preferential Attachment. In the mid-1950s, Herbert Simon [77] 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 [32], as follows. We grow a network by adding vertices over time. Each vertex 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

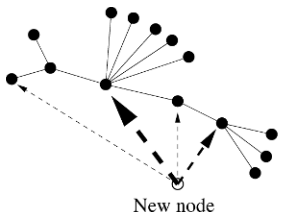


Figure 3.10. *The Barabási-Albert model* New nodes are added; each new node prefers to connect to existing nodes of high degree. The dashed lines show some possible edges for the new node, with thicker lines implying higher probability.

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

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

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 [13]. 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 3.10). The probability of choosing an existing node as an endpoint for these edges is given by

$$P(\text{edge to existing vertex } v) = \frac{k(v)}{\sum_i k(i)} \qquad (3.14)$$

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 [36] is given by:

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

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 [22] 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 [52] 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 [1] 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 companies 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 [14, 58, 37]
- The diameter of the BA model increases as N increases; however, many graphs exhibit shrinking diameters.

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.

Variants on Preferential Attachment.

Initial attractiveness. While the BA model generates graphs with a power law degree distribution, the power law exponent is stuck at $\gamma = 3$. Dorogovtsev et al. [36, 35] 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. In initial attractiveness, an extra "initial attractiveness" parameter is added which governs the probability of "young" sites gaining new edges. Adjusting this parameter will vary the degree distribution, adding significant flexibility to the BA model.

Internal edges and Rewiring. Albert and Barabási [6] proposed another method to add flexibility in the power law exponent. In the original BA model, one node and m edges are added to the graph every iteration. Albert and

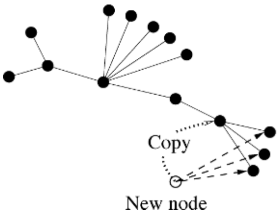


Figure 3.11. *The edge copying model* New nodes can choose to copy the edges of an existing node. This models the copying of links from other peoples' websites to create a new website.

Barabási 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: adding internal edges, removing/reconnecting ("rewiring") edges, and adding new nodes with some edges. This model exhibits either a power-law or exponential degree distribution, depending on the parameters used.

Edge Copying Models. Several graphs show community behavior, such as topic-based communities of websites on the WWW. Kleinberg et al. [51] and Kumar et al. [54] 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.

The Kleinberg [51] generator creates a directed graph. In this generator, nodes are independently created and deleted in each distribution, and edges incident on deleted nodes are also removed. Also, edges may be added to or deleted from existing nodes. Then, there is the key edge copying mechanism, where a node may copy edges from another node. An illustration is shown in Figure 3.11. 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.

Kumar et al. [54] propose a very similar model. However, there are some important differences. Whenever a new node is added, only *one* new edge is added. The copying process takes place when head or tail of some existing edge gets chosen as the endpoint of the new edge. This model may serve to create "communities" as there may be important nodes on each "topic".

This and similar models by analyzed by Kumar et al. [53]. In-degree distribution of Kleinberg's model follows a power law, and both in-and out-degree of Kumar et al.'s model follow power laws.

The Kleinberg model [51] generates a tree; no "back-edges" are formed from the old nodes to the new nodes. Also, in the model of Kumar et al. [54], 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. [4] 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.

Modifying the preferential attachment equation. Chen et al. [29] had found the AB model somewhat lacking in modeling the Web. Specifically, they found that the preference for connecting to high-degree nodes is stronger than that predicted by linear preferential attachment. Bu and Towsley [25] attempt to address this issue.

The AB model [6] 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 [29]). This is called the GLP (Generalized Linear Preference) model. The degree distribution follows a power law. Also, 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 [3]) models.

Others such as Krapivsky and Redner [52] have studied *non-linear* preferential attachment, finding this tended to produce degree decay faster than a power law.

Modeling increasing average degree. The average degree of several real-world graphs (such as citation graphs) increases over time [37, 14, 58], according to a Densification Power Law. Barabási et al. [14] attempt to modify the basic BA model to accommodate this effect. In the model, a new 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.

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 $\gamma = 2$.

Node fitness measures. 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 [52]. However, Adamic and Huberman find that this does not hold for the WWW [1]. There are websites which were created late but still have far higher in-degree than many older websites. Bianconi and Barabási [18] try to model this. Their model attaches a *fitness parameter* to each node, 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 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.

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.

Generalizing preferential attachment. 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. [4] 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. [21]. The work 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.

PageRank-based preferential attachment. Pandurangan et al. [73] found that the *PageRank* [23] 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. They modify the basic preferential attachment mechanism by adding a *PageRank*-based preferential attachment component– not only do edges preferentially connect to high degree nodes, but also high PageRank nodes. 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. The authors also found that the plain edge-copying model of Kumar et al. [54] 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.

The Forest Fire model. Leskovec et al. [58] 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. A node chooses an *ambassador* node uniformly at random, and then links recursively to the ambassador node's neighbors.

This creates preferential linking without explicitly assigning such probability. This method is similar to the edge copying model discussed earlier 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.

The Butterfly model. Most preferential-attachment based models will form a single connected component, when, in real graphs, there are many smaller components that evolve and occasionally join with each other. McGlohon et al. [59] develop a model that addresses this. Like in the Forest Fire model, there is an ambassador mechanism. However, there is no guarantee of linkage, so a node may become isolated and form its own new component for other nodes to join to. Additionally, instead of a single ambassador, a node may choose multiple ambassadors. This will allow components to join together.

The Butterfly model empirically produces power laws for both in- and out-degree, as well as reproducing the Densification Power Law and shrinking diameter. Furthermore, it reproduces oscillating patterns of the next-largest connected components mentioned earlier.

Deviations from power laws.

Problem being solved. Pennock et al. [75] 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. They attempt to model this deviation from power-law behavior.

Description and properties. Their model is similar to the BA model, 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:

$$p(v) = \alpha \frac{k(v)}{2mt} + (1-\alpha)\frac{1}{m_0 + t} \tag{3.15}$$

  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 3.15 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:

$$P(k) \propto (k+c)^{-1-\frac{1}{\alpha}} \tag{3.16}$$

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.

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 [65] 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 edge list.

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. If the constant β is a negative integer (say, $\beta = -1$ as in the AB model), 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.

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. [75], 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 Optimization-based generators

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*. There may be a number of constraints applied to the models– cost of connections, geographical distance, etc. We will discuss some models based on optimization of resources next.

The Highly Optimized Tolerance model.

Problem being solved:. Carlson and Doyle [27, 38] 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 $p_i (1 \leq i \leq n)$ (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: $\sum_i r_i \leq R$ for some given R. The aim is to minimize the expected cost

$$J = \left\{ \sum_i p_i l_i \mid l_i = f(r_i), \sum_i r_i \leq R \right\} \tag{3.17}$$

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 [71, 9]. 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.

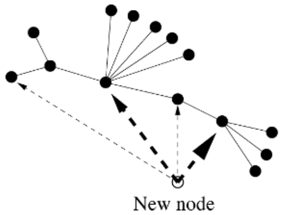


Figure 3.12. *The Heuristically Optimized Tradeoffs model* A new node prefers to link to existing nodes which are both close in distance and occupy a "central" position in the network.

Newman et al. [68] 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. [42], which we discuss next.

Modification: The Heuristically Optimized Tradeoffs model. Fabrikant et al. [42] 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).

The model assumes that nodes are spread out over 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 3.12). Thus, a new node i should be connected to an existing node j chosen to minimize

$$\alpha.d_{ij} + h_j \ \ (j < i) \tag{3.18}$$

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.

The authors find that the characteristics of the network depend greatly on the value of α, and may be a single hub or have an exponential degree distribution, but for a range of values power-law degree distribution results.

As in the *Highly Optimized Tolerance* model described before (Subsection 3.3.0), 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.

Other research in this direction is the recent work of Berger et al. [16], 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.

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.

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 [80] 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 [47], who tried to find out how people found jobs. The expectation was that the job

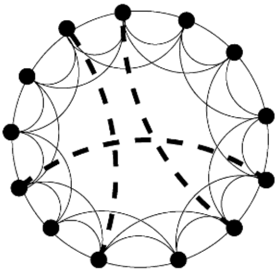


Figure 3.13. *The small-world model* Nodes are arranged in a ring lattice; each node has links to its immediate neighbors (solid lines) and some long-range connections (dashed lines).

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 [83] independently came up with a model with these characteristics: it has *high clustering coefficient* but *low diameter* . Their model (Figure 3.13), which has only one parameter p, consists of the following: begin with a ring lattice where each node has a set of "close friendships". Then rewire: for each node, each edge is rewired with probability p to a new random destination– these are the "weak ties".

Distance between nodes, and Clustering coefficient For $p = 0$ the graph remains a ring lattice, where both clustering coefficient and average distance between nodes are high. For $p = 1$, both values are very low. For a range of values in between, the average distance is low while clustering coefficient is high– as one would expect in real graphs. 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.

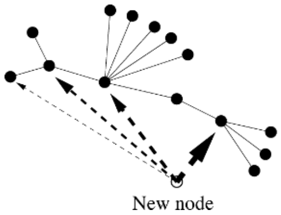


Figure 3.14. *The Waxman model* New nodes prefer to connect to existing nodes which are closer in distance.

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 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.

Other geographical models.

The Waxman Model. While the Small World model begins by constraining nodes to a local neighborhood, the Waxman model [84] explicitly builds the graph based on optimizing geographical constraints, to model the Internet graph.

The model is illustrated in Figure 3.14. Nodes (representing routers) are placed randomly in Cartesian 2-D space. An edge (u, v) is placed between two points u and v with probability

$$P(u, v) = \beta \exp \frac{-d(u, v)}{L\alpha} \tag{3.19}$$

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. While it does not yield a power-law degree distribution, it has been popular in the networking community.

The BRITE generator. Medina et al. [60] 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.

Nodes are placed on a square grid, with some m links per node. Growth occurs either all at once (as in Waxman) or incrementally (as in BA). Edges are wired randomly, preferentially, or combined preferential and geographical constraints as follows: 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 3.19)

$$\begin{aligned} P(u, v) &= \frac{w(u, v)k(v)}{\sum_i w(u, i)k(i)} \qquad (3.20) \\ \text{where } w(u, v) &= \beta \exp \frac{-d(u, v)}{L\alpha} \quad \text{as in Eq. 3.19} \end{aligned}$$

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, one limitation is that there has been little discussion of parameter fitting, an area for future research.

Yook et al. Model. Yook et al. [87] find two interesting linkages between geography and networks (specifically the Internet): First, the geographical distribution of Internet routers and Autonomous Systems (AS) is a fractal, and is strongly correlated with population density. Second, the probability of an edge occurring is *inversely proportional* to the Euclidean distance between the endpoints of the edge, likely due to cost of physical wire (which dominates over administrative cost for long links). However, in the Waxman and BRITE models, this probability decays exponentially with length (Equation 3.19).

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:

$$P(\text{node } u \text{ links to existing node } v) \propto \frac{k(v)^{\alpha}}{d(u, v)^{\sigma}} \qquad (3.21)$$

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. [82] and (independently) Kleinberg [50] propose a hierarchical model to explain this phenomenon.

PaC - utility based. Du et al. proposed an agent-based model "Pay and Call" or *PaC*, where agents make decisions about forming edges based on a perceived "profit" of an interaction. Each agent has a "friendliness" parameter. Calls are made with some "emotional dollars" cost, and agents may derive some benefit from each call. If two "friendly" agents interact, there is a higher benefit than if one or both agents are "unfriendly". The specific procedures are detailed in [39]. *PaC* generates degree, weight, and clique distributions as found in most real graphs.

3.4 Tensor-based

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 [28] 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 [75].
- 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.

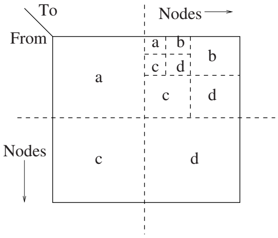


Figure 3.15. *The R-MAT model* The adjacency matrix is broken into four equal-sized partitions, and one of those four is chosen according to a (possibly non-uniform) probability distribution. This partition is then split recursively till we reach a single cell, where an edge is placed. Multiple such edge placements are used to generate the full synthetic graph.

- 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 3.15. 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 (i.e., 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, $a \geq b$, $a \geq c$, $a \geq d$.

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 [17] distribution, which can successfully model both power-law and "unimodal" distributions [75] 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 [28].

Parameter fitting algorithm Given some input graph, it is necessary 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).

Chakrabarti et al. showed experimentally that R-MAT can match both power-law distributions as well as deviations from power-laws [28], using a number of real graphs. The patterns matched by R-MAT include both in- and out-degree distributions, “hop-plot” and “effective diameter”, singular value vs. rank plots, “Network value” vs. rank plots, and “stress” distribution. Authors also compared R-MAT fits to those achieved by *AB*, *GLP*, and *PG* models.

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* [57], which generalizes the R-MAT model and can match several interesting patterns such as the Densification Power Law and the shrinking diameters effect in addition to all the patterns that R-MAT matches.

Graph Generation by Kronecker Multiplication. The R-MAT generator described in the previous paragraphs achieves its power mainly via a form of recursion: the adjacency matrix is recursively split into equal-sized quadrants over which edges are distributed unequally. One way to generalize this idea is via Kronecker matrix multiplication, wherein one small initial matrix is recursively "multiplied" with itself to yield large graph topologies. Unlike R-MAT, this generator has simple closed-form expressions for several measures of interest, such as degree distributions and diameters, thus enabling ease of analysis and parameter-fitting.

Description and properties. We first recall the definition of the Kronecker product.

Definition 3.5 (Kronecker product of matrices). *Given two matrices $\mathcal{A} = [a_{i,j}]$ and $\mathcal{B}$ of sizes $n \times m$ and $n' \times m'$ respectively, the Kronecker product matrix $\mathcal{C}$ of dimensions $(n * n') \times (m * m')$ is given by*

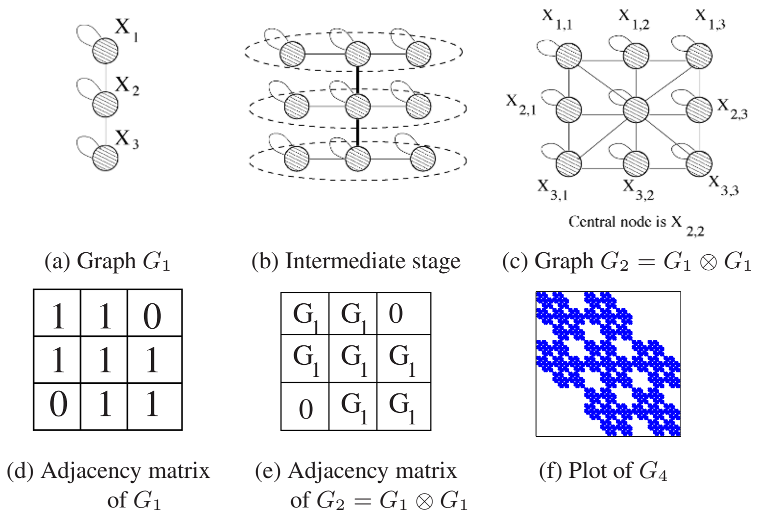
$$\mathcal{C} = \mathcal{A} \otimes \mathcal{B} \doteq \begin{pmatrix} a_{1,1}\mathcal{B} & a_{1,2}\mathcal{B} & \dots & a_{1,m}\mathcal{B} \\ a_{2,1}\mathcal{B} & a_{2,2}\mathcal{B} & \dots & a_{2,m}\mathcal{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathcal{B} & a_{n,2}\mathcal{B} & \dots & a_{n,m}\mathcal{B} \end{pmatrix} \tag{3.22}$$

In other words, for any nodes X_i and X_j in $\mathcal{A}$ and X_k and X_ℓ in $\mathcal{B}$, we have nodes $X_{i,k}$ and $X_{j,\ell}$ in the Kronecker product $\mathcal{C}$, and an edge connects them iff the edges (X_i, X_j) and (X_k, X_ℓ) exist in $\mathcal{A}$ and $\mathcal{B}$. The Kronecker product of two graphs is the Kronecker product of their adjacency matrices.

Let us consider an example. Figure 3.16(a–c) shows the recursive construction of $G \otimes H$, when $G = H$ is a 3-node path. Consider node $X_{1,2}$ in Figure 3.16(c): It belongs to the H graph that replaced node X_1 (see Figure 3.16(b)), and in fact is the X_2 node (i.e., the center) within this small H-graph. Thus, the graph H is recursively embedded "inside" graph G.

The Kronecker graph generator simply applies the Kronecker product multiple times over. Starting with a binary *initiator* graph, successively larger graphs are produced by repeated Kronecker multiplication. The properties of the generated graph thereby depend on those of the initiator graph.

There are several interesting properties of the Kronecker generator which are discussed in detail in [55]. Kronecker graphs have multinomial degree dis-



(a) Graph G_1 (b) Intermediate stage (c) Graph $G_2 = G_1 \otimes G_1$

1	1	0
1	1	1
0	1	1

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

(d) Adjacency matrix of G_1 (e) Adjacency matrix of $G_2 = G_1 \otimes G_1$ (f) Plot of G_4

Figure 3.16. *Example of Kronecker multiplication* Top: a "3-chain" and its Kronecker product with itself; each of the X_i nodes gets expanded into 3 nodes, which are then linked together. Bottom row: the corresponding adjacency matrices, along with matrix for the fourth Kronecker power G_4.

tributions, static diameter/effective diameter (if nodes have self-loops), multinomial distributions of eigenvalues, and community structure. Additionally, it provably follows the Densification Power Law.

Thanks to its simple mathematical structure, Kronecker graph generation allows the derivation of closed-form formulas for several important patterns. Of particular importance are the "temporal" patterns regarding changes in properties as the graph grows over time: both the constant diameter and the densification power law patterns are similar to those observed in real-world graphs [58], and are not matched by most graph generators.

While Kronecker multiplication allows several patterns to be computed analytically, its discrete nature leads to "staircase effects" in the degree and spectral distributions. A modification of the aforementioned generator avoids these effects: instead of a 0/1 matrix, the initiator graph adjacency matrix is chosen to have *probabilities* associated with edges. The edges are then chosen based on these probabilities.

RTM: Recursive generator for weighted, evolving graphs. Akoglu et al. [5] extend the Kronecker model to allow for multi-edges, or weighted edges. To the initial adjacency matrix, another dimension, or mode, is added to represent time. Then, in each iteration the *Kronecker tensor product* of the graph is taken. This will produce a growing graph that is self-similar in structure.

Since it shares many properties of the Kronecker generator, all static properties as well as densification are followed. Additionally, the weight additions over time will also be self-similar, as shown in real graphs in [59]. It was also shown to mimic other patterns for weighted graphs, such as the Weight Power Law and Snapshot Power Laws, as discussed in the previous section.

3.5 Generators for specific graphs

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.

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. However, the previous models do not explicitly enforce such hierarchies on the generated graphs.

Description and properties. Calvert et al. [26] 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. Also, the models use many parameters but provide only limited flexibility: what if we want a hierarchy with more than 3 levels? 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. [78] 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.

The Inet topology generator.

Problem being solved. Winick and Jamin [86] 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 3.13).
- 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

$$P(\text{degree } i \text{ node connects to degree } j \text{ node}) \propto w_i^j.j \tag{3.23}$$

$$\text{where } w_i^j = MAX\left(1, \sqrt{\left(\log\frac{i}{j}\right)^2 + \left(\log\frac{f(i)}{f(j)}\right)^2}\right) \tag{3.24}$$

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 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 subsection, so that they can match most, if not all, of the graph patterns. R-MAT is a step in this direction.

4. 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.

Notes

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

[2] Tangmunarunkit et al. [78] use it only to differentiate between exponential and sub-exponential growth

References

[1] Lada A. Adamic and Bernardo A. Huberman. Power-law distribution of the World Wide Web. *Science*, 287:2115, 2000.

[2] Lada A. Adamic and Bernardo A. Huberman. The Web's hidden order. *Communications of the ACM*, 44(9):55–60, 2001.

[3] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *ACM Symposium on Theory of Computing*, pages 171–180, New York, NY, 2000. ACM Press.

[4] William Aiello, Fan Chung, and Linyuan Lu. Random evolution in massive graphs. In *IEEE Symposium on Foundations of Computer Science*, Los Alamitos, CA, 2001. IEEE Computer Society Press.

[5] Leman Akoglu, Mary Mcglohon, and Christos Faloutsos. Rtm: Laws and a recursive generator for weighted time-evolving graphs. In *International Conference on Data Mining*, December 2008.

[6] R«eka Albert and Albert-L«aszl«o Barab«asi. Topology of evolving networks: local events and universality. *Physical Review Letters*, 85(24):5234–5237, 2000.

[7] R«eka Albert and Albert-L«aszl«o Barab«asi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[8] R«eka Albert, Hawoong Jeong, and Albert-L«aszl«o Barab«asi. Diameter of the World-Wide Web. *Nature*, 401:130–131, September 1999.

[9] R«eka Albert, Hawoong Jeong, and Albert-L«aszl«o Barab«asi. Error and attack tolerance of complex networks. *Nature*, 406:378–381, 2000.

[10] Lu«s A. Nunes Amaral, Antonio Scala, Marc Barth«el«emy, and H. Eugene Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences*, 97(21):11149–11152, 2000.

[11] Ricardo Baeza-Yates and Barbara Poblete. Evolution of the Chilean Web structure composition. In *Latin American Web Congress*, Los Alamitos, CA, 2003. IEEE Computer Society Press.

[12] Albert-L«aszl«o Barab«asi. *Linked: The New Science of Networks*. Perseus Books Group, New York, NY, first edition, May 2002.

[13] Albert-L«aszl«o Barab«asi and R«eka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[14] Albert-L«aszl«o Barab«asi, Hawoong Jeong, Z. N«eda, Erzs«ebet Ravasz, A. Schubert, and Tam«as Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.

[15] Jan Beirlant, Tertius de Wet, and Yuri Goegebeur. A goodness-of-fit statistic for Pareto-type behaviour. *Journal of Computational and Applied Mathematics*, 186(1):99–116, 2005.

[16] Noam Berger, Christian Borgs, Jennifer T. Chayes, Raissa M. D'Souza, and Bobby D. Kleinberg. Competition-induced preferential attachment. *Combinatorics, Probability and Computing*, 14:697–721, 2005.

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

[18] Ginestra Bianconi and Albert-László Barabási. Competition and multiscaling in evolving networks. *Europhysics Letters*, 54(4):436–442, 2001.

[19] Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Structural properties of the African Web. In *International World Wide Web Conference*, New York, NY, 2002. ACM Press.

[20] Béla Bollobás. *Random Graphs*. Academic Press, London, 1985.

[21] Béla Bollobás, Christian Borgs, Jennifer T. Chayes, and Oliver Riordan. Directed scale-free graphs. In *ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, 2003. SIAM.

[22] Béla Bollobás and Oliver Riordan. The diameter of a scale-free random graph. Combinatorica, 2002.

[23] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[24] Andrei Z. Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web: experiments and models. In *International World Wide Web Conference*, New York, NY, 2000. ACM Press.

[25] Tian Bu and Don Towsley. On distinguishing between Internet power law topology generators. In *IEEE INFOCOM*, Los Alamitos, CA, 2002. IEEE Computer Society Press.

[26] Kenneth L. Calvert, Matthew B. Doar, and Ellen W. Zegura. Modeling Internet topology. *IEEE Communications Magazine*, 35(6):160–163, 1997.

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

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

[29] Q. Chen, H. Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. The origin of power laws in Internet topologies revisited.

In *IEEE INFOCOM*, Los Alamitos, CA, 2001. IEEE Computer Society Press.

[30] Colin Cooper and Alan Frieze. 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, 2004.

[31] Mark Crovella and Murad S. Taqqu. Estimating the heavy tail index from scaling properties. *Methodology and Computing in Applied Probability*, 1(1):55–79, 1999.

[32] Derek John de Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27:292–306, 1976.

[33] Stephen Dill, Ravi Kumar, Kevin S. McCurley, Sridhar Rajagopalan, D. Sivakumar, and Andrew Tomkins. Self-similarity in the Web. In *International Conference on Very Large Data Bases*, San Francisco, CA, 2001. Morgan Kaufmann.

[34] Pedro Domingos and Matthew Richardson. Mining the network value of customers. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, New York, NY, 2001. ACM Press.

[35] Sergey N. Dorogovtsev and José Fernando Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, Oxford, UK, 2003.

[36] Sergey N. Dorogovtsev, José Fernando Mendes, and Alexander N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633–4636, 2000.

[37] Sergey N. Dorogovtsev, José Fernando Mendes, and Alexander N. Samukhin. Giant strongly connected component of directed networks. *Physical Review E*, 64:025101 1–4, 2001.

[38] John Doyle and Jean M. Carlson. Power laws, Highly Optimized Tolerance, and Generalized Source Coding. *Physical Review Letters*, 84(24):5656–5659, June 2000.

[39] Nan Du, Christos Faloutsos, Bai Wang, and Leman Akoglu. Large human communication networks: patterns and a utility-driven generator. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278, New York, NY, USA, 2009. ACM.

[40] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Acadamy of Science*, 5:17–61, 1960.

[41] Paul Erdős and Alfréd Rényi. On the strength of connectedness of random graphs. *Acta Mathematica Scientia Hungary*, 12:261–267, 1961.

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

[43] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*, pages 251–262, New York, NY, 1999. ACM Press.

[44] Andrey Feuerverger and Peter Hall. Estimating a tail exponent by modelling departure from a Pareto distribution. *The Annals of Statistics*, 27(2):760–781, 1999.

[45] Michael L. Goldstein, Steven A. Morris, and Gary G. Yen. Problems with fitting to the power-law distribution. *The European Physics Journal B*, 41:255–258, 2004.

[46] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet map discovery. In *IEEE INFOCOM*, pages 1371–1380, Los Alamitos, CA, March 2000. IEEE Computer Society Press.

[47] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, May 1973.

[48] Bruce M. Hill. A simple approach to inference about the tail of a distribution. *The Annals of Statistics*, 3(5):1163–1174, 1975.

[49] George Karypis and Vipin Kumar. Multilevel algorithms for multi-constraint graph partitioning. Technical Report 98-019, University of Minnesota, 1998.

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

[51] Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. The web as a graph: Measurements, models and methods. In *International Computing and Combinatorics Conference*, Berlin, Germany, 1999. Springer.

[52] Paul L. Krapivsky and Sidney Redner. Organization of growing random networks. *Physical Review E*, 63(6):066123 1–14, 2001.

[53] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. Stochastic models for the Web graph. In *IEEE Symposium on Foundations of Computer Science*, Los Alamitos, CA, 2000. IEEE Computer Society Press.

[54] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Extracting large-scale knowledge bases from the web. In *Inter-*

national Conference on Very Large Data Bases, San Francisco, CA, 1999. Morgan Kaufmann.

[55] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Gharamani. Kronecker graphs: an approach to modeling networks, 2008.

[56] Jure Leskovec, Mary Mcglohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. Cascading behavior in large blog graphs. *SIAM International Conference on Data Mining (SDM)*, 2007.

[57] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, and Christos Faloutsos. Realistic, mathematically tractable graph generation and evolution, using Kronecker Multiplication. In *Conference on Principles and Practice of Knowledge Discovery in Databases*, Berlin, Germany, 2005. Springer.

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

[59] Mary Mcglohon, Leman Akoglu, and Christos Faloutsos. Weighted graphs and disconnected components: Patterns and a generator. In *ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, August 2008.

[60] Alberto Medina, Ibrahim Matta, and John Byers. On the origin of power laws in Internet topologies. In *Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*, pages 18–34, New York, NY, 2000. ACM Press.

[61] Milena Mihail and Christos H. Papadimitriou. On the eigenvalue power law. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, Berlin, Germany, 2002. Springer Verlag.

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

[63] Alan L. Montgomery and Christos Faloutsos. Identifying Web browsing trends and patterns. *IEEE Computer*, 34(7):94–95, 2001.

[64] M. E. J. Newman. Power laws, pareto distributions and zipf's law, December 2004.

[65] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[66] Mark E. J. Newman. Power laws, pareto distributions and Zipf's law. *Contemporary Physics*, 46:323–351, 2005.

[67] Mark E. J. Newman, Stephanie Forrest, and Justin Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101 1–4, 2002.

[68] Mark E. J. Newman, Michelle Girvan, and J. Doyne Farmer. Optimal design, robustness and risk aversion. *Physical Review Letters*, 89(2):028301 1–4, 2002.

[69] Mark E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118 1–17, 2001.

[70] Christine Nickel. *Random Dot Product Graphs: A Model for Social Networks*. PhD thesis, The Johns Hopkins University, 2007.

[71] Christopher Palmer, Phil B. Gibbons, and Christos Faloutsos. 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*, New York, NY, 2002. ACM Press.

[72] Christopher Palmer and J. Gregory Steffan. Generating network topologies that obey power laws. In *IEEE Global Telecommunications Conference*, Los Alamitos, CA, November 2000. IEEE Computer Society Press.

[73] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal. Using PageRank to characterize Web structure. In *International Computing and Combinatorics Conference*, Berlin, Germany, 2002. Springer.

[74] Romualdo Pastor-Satorras, Alexei Vásquez, and Alessandro Vespignani. Dynamical and correlation properties of the Internet. *Physical Review Letters*, 87(25):258701 1–4, 2001.

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

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

[77] Herbert Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440, 1955.

[78] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topologies, power laws, and hierarchy. Technical Report 01-746, University of Southern California, 2001.

[79] Sudhir L. Tauro, Christopher Palmer, Georgos Siganos, and Michalis Faloutsos. A simple conceptual model for the Internet topology. In *Global Internet*, Los Alamitos, CA, 2001. IEEE Computer Society Press.

[80] Jeffrey Travers and Stanley Milgram. An experimental study of the Small World problem. *Sociometry*, 32(4):425–443, 1969.

[81] Duncan J. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton and Company, New York, NY, 1st edition, 2003.

[82] Duncan J. Watts, Peter Sheridan Dodds, and Mark E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

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

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

[85] H. S. Wilf. *Generating Functionology*. Academic Press, 1990.

[86] Jared Winick and Sugih Jamin. Inet-3.0: Internet Topology Generator. Technical Report CSE-TR-456-02, University of Michigan, Ann Arbor, 2002.

[87] Soon-Hyung Yook, Hawoong Jeong, and Albert-László Barabási. Modeling the Internet’s large-scale topology. *Proceedings of the National Academy of Sciences*, 99(21):13382–13386, 2002.