

Chapter 2

STATISTICAL PROPERTIES OF SOCIAL NETWORKS

Mary McGlohon

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

Leman Akoglu

*School of Computer Science
Carnegie Mellon University
lakoglu@cs.cmu.edu*

Christos Faloutsos

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

Abstract In this chapter we describe patterns that occur in the structure of social networks, represented as graphs. We describe two main classes of properties, static properties, or properties describing the structure of snapshots of graphs; and dynamic properties, properties describing how the structure evolves over time. These properties may be for unweighted or weighted graphs, where weights may represent multi-edges (e.g. multiple phone calls from one person to another), or edge weights (e.g. monetary amounts between a donor and a recipient in a political donation network).

Keywords: Power laws, network structure, weighted graphs

What do social networks look like on a global scale? How do they evolve over time? How do the different components of an entire network form? What

happens when we take into account multiple edges and weighted edges? Can we identify certain patterns regarding these weights?

There has been extensive work focusing on static snapshots of graphs, where fascinating properties have been discovered, the most striking ones being the ‘small-world’ phenomenon [38] (also known as ‘six degrees of separation’ [24]) and the power-law degree distributions [3, 12]. Time-evolving graphs have attracted attention only recently, where even more fascinating properties have been discovered, like *shrinking* diameters, and the so-called *densification power law* [18]. Moreover, we find interesting properties in terms of *multiple* edges between nodes, or edge weights.

In this chapter we will describe some of the most important properties apparent in social networks, with a particular emphasis on dynamic properties, and some of the newer findings with respect to edge weights.

The questions of interest are:

- *What do social networks look like, on a large scale?* Do most nodes have few connections, with several “hubs” or is the distribution more stable? What sort of clustering behavior occurs?
- *How do networks behave over time?* Does the structure vary as the network grows? In what fashion do new entities enter a network? Does the network retain certain graph properties as it grows and evolves? Does the graph undergo a “phase transition”, in which its behavior suddenly changes?
- *How do the non-giant weakly connected components behave over time?* One might argue that they grow, as new nodes are being added; and their size would probably remain a fixed fraction of the size of the GCC. Someone else might counter-argue that they shrink, and they eventually get absorbed into the GCC. What is happening, in real graphs?
- *What distributions and patterns do weighted graphs maintain?* How does the distribution of weights change over time— do we also observe a densification of weights as well as single-edges? How does the distribution of weights relate to the degree distribution? Is the addition of weight bursty over time, or is it uniform?

Answering these questions is important to understand how natural graphs evolve, and to (a) spot anomalous graphs and sub-graphs; (b) answer questions about entities in a network and what-if scenarios; and (c) discard unrealistic graph generators.

Let’s elaborate on each of the above applications: Spotting anomalies is vital for determining abuse of social and computer networks, such as link-spamming in a web graph, fraudulent reputation building in e-auction systems [29], detection of dwindling/abnormal social sub-groups in a social-networking site like Yahoo-360 (360.yahoo.com), Facebook (www.facebook.com) and LinkedIn

Symbol	Description
\mathcal{G}	Graph representation of datasets
\mathcal{V}	Set of nodes for graph \mathcal{G}
\mathcal{E}	Set of edges for graph \mathcal{G}
N	Number of nodes, or $ \mathcal{V} $
E	Number of edges, or $ \mathcal{E} $
$e_{i,j}$	Edge between node i and node j
$w_{i,j}$	Weight on edge $e_{i,j}$
w_i	Weight of node i (sum of weights of incident edges)
\mathbf{A}	0-1 Adjacency matrix of the unweighted graph
\mathbf{A}_w	Real-value adjacency matrix of the weighted graph
$a_{i,j}$	Entry in matrix \mathbf{A}
λ_1	Principal eigenvalue of unweighted graph
$\lambda_{1,w}$	Principal eigenvalue of weighted graph

Table 2.1. Table of Notations.

(www.linkedin.com), and network intrusion detection [17]. Analyzing network properties is also useful for identifying authorities and search algorithms [7, 9, 16], for discovering the “network value” of customers for using viral marketing [30], or to improve recommendation systems [5]. What-if scenarios are vital for extrapolation, provisioning and algorithm design: For example if we expect that the number of links will double within the next year, we should provision for the appropriate hardware to store and process the upcoming queries.

The rest of this chapter will examine both the static and dynamic properties, for weighted and unweighted graphs. However, before delving into these static and dynamic properties, we will next establish some terms and definitions we will use in the rest of the chapter.

1. Preliminaries

We will first provide some basic definitions and terms we will use, and then present some particular data sets we will reference. A full list of symbols can be shown in Table 2.1.

1.1 Definitions

1.1.1 Graphs. We can represent a social network as a *graph*. For the rest of the chapter we will use *network* and *graph* interchangeably.

A static, unweighted graph G consists of a set of nodes \mathcal{V} and a set of edges \mathcal{E} : $G = (\mathcal{V}, \mathcal{E})$. We represent the sizes of \mathcal{V} and \mathcal{E} as N and E . A graph may be *directed* or *undirected*— for instance, a phone call may be from one party to another, and will have a directed edge, or a mutual friendship may be represented as an undirected edge. Most properties we examine will be on undirected graphs.

Graphs may also be *weighted*, where there may be multiple edges occurring between two nodes (e.g. repeated phone calls) or specific edge weights (e.g. monetary amounts for transactions). In a weighted graph \mathcal{G} , let $e_{i,j}$ be the edge between node i and node j . We shall refer to these two nodes as the ‘*neighboring nodes*’ or ‘*incident nodes*’ of edge $e_{i,j}$. Let $w_{i,j}$ be the weight on edge $e_{i,j}$. The *total weight* w_i of node i is defined as the sum of weights of all its incident edges, that is $w_i = \sum_{k=1}^{d_i} w_{i,k}$, where d_i denotes its degree. As we show later, there is a relation between a given edge weight $w_{i,j}$ and the weights of its neighboring nodes w_i and w_j .

Finally, graphs may be *unipartite* or *multipartite*. Most social networks one thinks of are unipartite— people in a group, papers in a citation network, etc. However, there may also be multipartite— that is, there are multiple classes of nodes and edges are only drawn between nodes of different classes. Bipartite graphs, like the movie-actor graph of IMDB, consist of disjoint sets of nodes \mathcal{V}_1 and \mathcal{V}_2 , say, for authors and movies, with no edges among nodes of the same type.

We can represent a graph either visually, or with an *adjacency matrix* \mathbf{A} , where nodes are in rows and columns, and numbers in the matrix indicate the existence of edges. For unweighted graphs, all entries are 0 or 1; for weighted graphs the adjacency matrix contains the values of the weights. Figure 2.1 shows examples of graphs and their adjacency matrices.

We next introduce other important concepts we use in analyzing these graphs.

1.1.2 Components. Another interesting property of a graph is its *component distribution*. We refer to a *connected component* in a graph as a set of nodes and edges where there exists a path between any two nodes in the set. (For directed graphs, this would be a *weakly connected component*, where a *strongly connected component* requires a directed path between any given two nodes in a set.) We find that in real graphs over time, a giant connected component (GCC) forms. However, it is also of interest to study the smaller components— when do they choose to join the GCC, and what size do they reach before doing so?

In our observations we will focus on the size of the second- and third- largest components. We will also look at the large scale distribution of all component sizes, and how that distribution changes over time. Not surprisingly, components of *rank* ≥ 2 form a power law.

1.1.3 Diameter and Effective Diameter. We may want to answer the questions: How does the largest connected component of a real graph evolve over time? Do we start with one large CC, that keeps on growing? We propose to use the *diameter-plot* of the graph, that is, its diameter, over time, to answer these questions. For a given (static) graph, its *diameter* is defined as

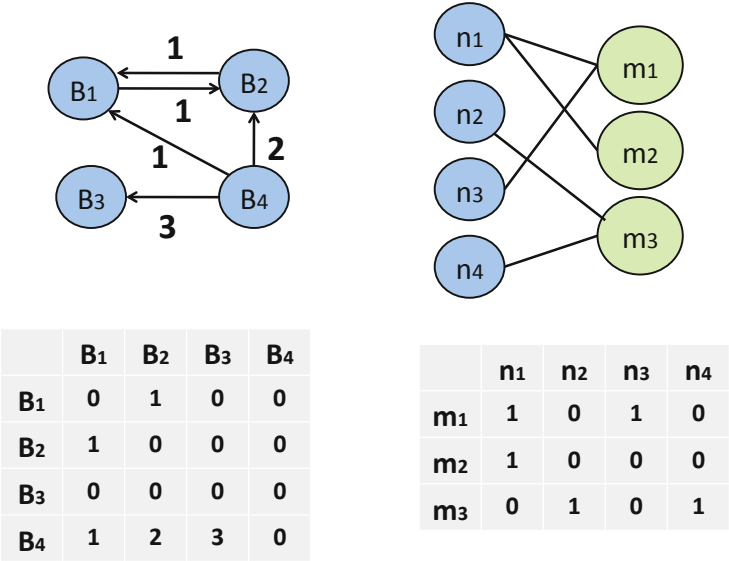


Figure 2.1. Illustrations of example graphs. On the left is a unipartite, directed, weighted graph and the corresponding adjacency matrix. On the right is an undirected, bipartite graph and the corresponding adjacency matrix.

the maximum *distance* between any two nodes, where distance is the minimum number of hops (i.e., edges that must be traversed) on the path from one node to another, ignoring directionality. Calculating graph diameter is $O(N^2)$. Therefore, we choose to estimate the graph diameter by sampling nodes from the giant component. For $s = \{1, 2, \dots, S\}$, we choose two nodes at random and calculate the distance (using breadth-first search). We then choose to record the 90 percentile value of distances, so we take the $.9S$ largest recorded value. The distance operation is $O(dk)$, where d is the graph diameter and k the maximum degree of any node—on average this is a much smaller cost. Intuitively, the diameter represents how much of a “small world” the graph is—how quickly one can get from one “end” of the graph to another. This is described in [35]. We use sampling to estimate the diameter; alternative methods would include ANF [28].

1.1.4 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) [12]. Heavy-tailed distributions attempt to model this. They are known as “heavy-tailed” because, while traditional exponential distributions have bounded variance (large deviations from the mean become nearly impossible), $p(x)$ decays polynomially quickly instead of exponentially as $x \rightarrow \infty$, creating a “fat tail” for extreme values on the PDF plot.

One of the more well-known heavy-tailed distributions is the power law distribution. Two variables x and y are related by a power law when:

$$y(x) = Ax^{-\gamma} \quad (2.1)$$

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

A random variable is distributed according to a power law when the probability density function (pdf) is given by:

$$p(x) = Ax^{-\gamma}, \quad \gamma > 1, x \geq x_{min} \quad (2.2)$$

The extra $\gamma > 1$ requirement ensures that $p(x)$ can be normalized. Power laws with $\gamma < 1$ rarely occur in nature, if ever [26].

Skewed distributions, such as power laws, occur very often in real-world graphs, as we will discuss. Figures 2.2(a) and 2.2(b) show two examples of power laws.

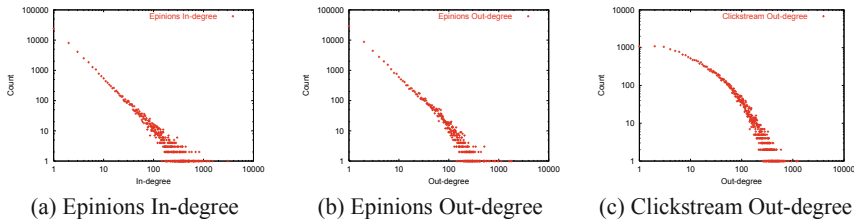


Figure 2.2. *Power laws and deviations:* Plots (a) and (b) show the in-degree and out-degree distributions on a log-log scale for the *Epinions* graph (an online social network of 75,888 people and 508,960 edges [11]). Both follow power-laws. In contrast, plot (c) shows the out-degree distribution of a *Clickstream* graph (a bipartite graph of users and the websites they surf [25]), which deviates from the power-law pattern.

While power laws appear in a large number of graphs, deviations from a pure power law are sometimes observed. Two of the more common deviations are exponential cutoffs and lognormals.

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:

$$y(x = k) \propto e^{-k/\kappa} k^{-\gamma} \quad (2.3)$$

where $e^{-k/\kappa}$ is the exponential cutoff term and $k^{-\gamma}$ is the power law term.

Similar distributions were studied by Bi et al. [6], 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 has been used to fit the degree distribution of a bipartite “clickstream” graph linking websites and users (Figure 2.2(c)), telecommunications and other data.

Methods for fitting heavy-tailed distributions are described in [26, 10].

1.1.5 Burstiness and Entropy Plots. Human activity, including weight additions in graphs, is often bursty. If that the traffic is *self-similar*, then we can measure the burstiness, using the intrinsic, or *fractal* dimension of the cloud of timestamps of edge-additions (or weight-additions). Let $\Delta W(t)$ be the total weight of edges that were added during the t -th interval, e.g., the total network flow on day t , among all the machines we are observing.

Among the many methods that measure self-similarity (Hurst exponent, etc. [31]), we choose the *entropy plot* [37], which plots the entropy $H(r)$ versus the resolution r . The resolution is the scale, that is, at resolution r , we divide our time interval into 2^r equal sub-intervals, sum the weight-additions

$\Delta W(t)$ in each sub-interval k ($k = 1 \dots 2^r$), normalize into fractions p_k ($= \Delta W(t)/W_{total}$), and compute the Shannon entropy of the sequence p_k : $H(r) = -\sum_k p_k \log_2 p_k$. If the plot $H(r)$ is linear in some range of resolutions, the corresponding time sequence is said to be *fractal* in that range, and the slope of the plot is defined as the *intrinsic* (or *fractal*) dimension D of the time sequence. Notice that a uniform weight-addition distribution yields $D=1$; a lower value of D corresponds to a more bursty time sequence like a Cantor dust [31], with a single burst having the lowest $D=0$: the intrinsic dimension of a point. Also notice that a variation of the 80-20 model, the so called ‘b-model’ [37], generates such self-similar traffic.

We studied several large real-world weighted graphs described in detail in Table 2.2. In particular, *BlogNet* contains blog-to-blog links, *NetworkTraffic* records IP-source/IP-destination pairs, along with the number of packets sent. Bipartite networks *Auth-Conf*, *Keyw-Conf*, and *Auth-Keyw* are from DBLP, representing submission records of authors to conferences with specified keywords. *CampaignOrg* is from the US FEC, a public record of donations between political candidates and organizations.

For *NetworkTraffic* and *CampaignOrg* datasets, the weights on the edges are actual weights representing number of packets and donation amounts. For the remaining datasets, the edge weights are simply the number of occurrences of the edges. For instance, if author i submits a paper to conference j for the first time, the weight $w_{i,j}$ of edge $e_{i,j}$ is set to 1. If author i later submits another paper to the same conference, the edge weight becomes 2.

A complete list of the symbols used throughout text is listed in Table 2.1.

1.2 Data description

We will illustrate some properties described in this chapter on different real-world social networks. These are described in detail in Table 2.2. This includes both bipartite and unipartite, and weighted and unweighted graphs.

Several of our graphs had no obvious weighting scheme: for example, a single paper or patent will cite another only a single time. The graphs that did have weights are also further divided into two schemes, *multi-edges* and *edge-weights*. In the edge-weights scheme, there is an obvious weight on edges, such as amounts in campaign donations, or packet-counts in network traffic. For multi-edges, weights are added if there is more than one interaction between two nodes. For instance, if a blog cites another blog at a given time, its weight is 1. If it cites the blog again later, the weight becomes 2.

The datasets are gathered from publicly available data. *NIPS*¹, *Arxiv* and *Patent* [19] are academic paper or patent citation graphs with no weighting

¹www.cs.toronto.edu/~roweis/data.html

Name	Weights	N , E ,time	Description
<i>PostNet</i>	Unweighted	250K, 218K, 80 d.	Blog post citation network
<i>NIPS</i>	Unweighted	2K, 3K, 13 yr.	Paper citation network
<i>Arxiv</i>	Unweighted	30K, 60K, 13 yr.	Paper citation network
<i>Patent</i>	Unweighted	4M, 8M, 17 yr.	Patent citation network
<i>IMDB</i>	Unweighted	757K, 2M, 114 yr.	Bipartite actor-movie network
<i>Netflix</i>	Unweighted	125K, 14M, 72 mo.	Bipartite user-movie ratings
<i>BlogNet</i>	Multi-edges	60K, 125K, 80 d.	Social network of blogs based on citations
<i>Auth-Conf</i>	Multi-edges	17K, 22K, 25 yr.	Bipartite DBLP Author-to-Conference associations
<i>Key-Conf</i>	Multi-edges	10K, 23K, 25 yr.	Bipartite DBLP Keyword-to-Conference associations
<i>Auth-Key</i>	Multi-edges	27K, 189K, 25 yr.	Bipartite DBLP Author-to-Keyword associations
<i>CampOrg</i>	Edge-weights (Amounts)	23K, 877K, 28 yr.	Bipartite U.S. electoral campaign donations from organizations to candidates (available from FEC)
<i>CampIndiv</i>	Edge-weights (Amounts)	6M, 10M, 22 yr.	Bipartite election donations from individuals to organizations

Table 2.2. The datasets referred to in this chapter.

scheme. *IMDB* indicates movie-actor information, where an edge occurs if an actor participates in a movie [3]. *Netflix* is the dataset from the Netflix Prize competition², with user-movie links (we ignored the ratings); we also noticed that it only contained users with 100 or more ratings. *BlogNet* and *PostNet* are two representations of the same data, hyperlinks between blog posts [21]. In *PostNet* nodes represent individual posts, while in *BlogNet* each node represents a blog. Essentially, *PostNet* is a paper citation network while *BlogNet* is an author citation network (which contains multi-edges).

Auth-Conf, *Key-Conf*, and *Auth-Key* are all from DBLP³, with the obvious meanings. *CampOrg* and *CampIndiv* are bipartite graphs from U.S. Federal Election Commission, recording donation amounts from organizations to political candidates and individuals to organizations⁴.

In all the above cases, we assume that edges are never deleted, because edge deletion never explicitly appeared in these datasets.

²www.netflixprize.com

³dblp.uni-trier.de/xml/

⁴www.cs.cmu.edu/~mmcgloho/fec/data/fec_data.html

2. Static Properties

We next review *static* properties of social graphs. While all networks we examine are evolving over time, there are properties that are measured at single points in time, that is, static snapshots of the graphs. For the purposes of organization we will further divide these properties into those applying to unweighted graphs and to weighted graphs.

2.1 Static Unweighted Graphs

Here, we present the ‘laws’ that apply to static snapshots of real graphs without considering the weights on the edges. Those include the patterns in degree distributions, the number of hops pairs of nodes can reach each other, local number of triangles, eigenvalues and communities. Next, we describe the related patterns in more detail.

2.1.1 S-1: Heavy-tailed Degree Distribution. The degree distribution of many real graphs obey a power law of the form $f(d) \propto d^{-\alpha}$, with the exponent $\alpha > 0$, and $f(d)$ being the fraction of nodes with degree d . Such power-law relations as well as many more have been reported in [8, 12, 15, 26]. Intuitively, power-law-like distributions for degrees state that there exist many low degree nodes, whereas only a few high degree nodes in real graphs.

2.1.2 S-2: Small Diameter. One of the most striking patterns that real-world graphs have is a small diameter, which is also known as the ‘small-world phenomenon’ or the ‘six degrees of separation’.

For a given static graph, its diameter is defined as the maximum *distance* between any two nodes, where distance is the minimum number of hops (i.e., edges that must be traversed) on the path from one node to another, usually ignoring directionality. Intuitively, the diameter represents how much of a “small world” the graph is— how quickly one can get from one “end” of the graph to another.

Many real graphs were found to exhibit surprisingly small diameters— for example, 19 for the Web [2], and the well-known “six-degrees of separation” in social networks [4]. It has also been observed that the diameter spikes at the ‘gelling point’ [22].

Since the diameter is defined as the *maximum*-length shortest path between all possible pairs, it can easily be hijacked by long chains. Therefore, often the *effective diameter* is used as a more robust metric, which is the 90-percentile of the pairwise distances among all reachable pairs of nodes. In other words, the *effective diameter* is the minimum number of hops in which some fraction (usually 90%) of all connected node pairs can be reached [34].

Computing all-pairs-shortest-path lengths is practically intractable for very large graphs. The exact algorithm is prohibitively expensive (at least $O(N^2)$); while one can use sampling to estimate it, alternative methods would include ANF [28].

2.1.3 S-3: Triangle Power Law (TPL). The number of triangles Δ and the number of nodes that participate in Δ number of triangles should follow a power-law in the form of $f(\Delta) \propto \Delta^\sigma$, with the exponent $\sigma < 0$ [36]. The TPL intuitively states that while many nodes have only a few triangles in their neighborhoods, a few nodes participate in many number of triangles with their neighbors. The local number of triangles is related to the clustering coefficient of graphs.

2.1.4 S-4: Eigenvalue Power Law (EPL). Siganos et.al. [33] examined the spectrum of the adjacency matrix of the AS Internet topology and reported that the 20 or so largest eigenvalues of the Internet graph are power-law distributed. Michail and Papadimitriou [23] later provided an explanation for the ‘Eigenvalue Power Law’, showing that it is a consequence of the ‘Degree Power Law’.

2.1.5 S-5: Community Structure. Real-world graphs are found to exhibit a modular structure, with nodes forming groups, and possibly groups within groups [13, 14, 32]. In a modular graph, the nodes form communities where groups of nodes in the same community are tighter connected to each other than to those nodes outside the community. In [27], Newman and Girvan provide a quantitative measure for such a structure, called *modularity*.

2.2 Static Weighted Graphs

Here we try to find patterns that weighted graphs obey. In this section we consider graphs to be directed (and impose a single direction in bipartite graphs), as this will be an important consideration on the weights. The dataset consist of quadruples: (IP-source, IP-destination, timestamp, number-of-packets), where timestamp is in increments of, say, 30 minutes. Thus, we have multi-edges, as well as total weight for each (source, destination) pair. Let $W(t)$ be the total weight up to time t (ie., the grand total of all exchanged packets across all pairs), $E(t)$ the number of distinct edges up to time t , and $E_d(t)$ the number of multi-edges (the d subscript stands for *duplicate* edges), up to time t .

We present three “laws” that our datasets seem to follow: The first is the “weight power law” (WPL) correlating the total weight, the total number of edges and the total number of multi-edges, over time. The second is the “edge weights power law”, the same law as applied to individual nodes. The third is

the “snapshot power law” (SPL), correlating the in-degree with the in-weight, and the out-degree with the out-weight, for all the nodes of a graph, at a given time-stamp.

2.2.1 SW-1: Weight Power Law (WPL). As defined above, suppose we have $E(t)$ total unique edges up to time t (ie., count of pairs that know each other) and $W(t)$ being the total count of packets up to time t . Is there a relationship between $W(t)$ and $E(t)$? If every pair generated k packets, the relationships would be linear: if the count of pairs double, the packet count would double, too. This is reasonable, but it doesn’t happen! In reality, the packet count over-doubles, following the “WPL” below. We shall refer to this phenomenon as the “*fortification effect*”: more edges in the graph imply super-linearly higher total weight.

OBSERVATION 2.1 (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. Power-laws also link the number of nodes $N(t)$, and the number of multi-edges $E_d(t)$, to $E(t)$, with exponents n and $\text{dup}E$, respectively.

The weight exponent w ranges from 1.01 to 1.5 for the real graphs we have studied. The highest value corresponds to campaign donations: super-active organizations that support many campaigns also tend to spend even more money per campaign than the less active organizations. For bipartite graphs, we show the n_{src} , n_{dst} exponents for the source and destination nodes (which also follow power laws: $N_{src}(t) = E(t)^{n_{src}}$ and similarly for $N_{dst}(t)$).

Fig. 2.5 shows all these quantities, versus $E(t)$, for several datasets. The plots are all in log-log scales, and straight lines fit well. We report the slopes in Table 2.

2.2.2 SW-2: Edge Weights Power Law. We observe that the weight of a given edge and weights of its neighboring two nodes are correlated. Our observation is similar to Newton’s Gravitational Law stating that the gravitational force between two point masses is proportional to the product of the masses.

OBSERVATION 2.2 (EDGE WEIGHTS POWER LAW(EWPL)) *Given a real-world graph \mathcal{G} , ‘communication’ defined as the weight of the link between two given nodes has a power law relation with the weights of the nodes. In particular, given an edge $e_{i,j}$ with weight $w_{i,j}$ and its two neighbor nodes i*

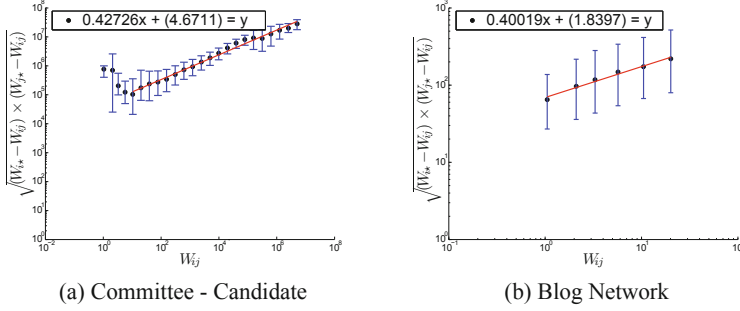


Figure 2.3. Illustration of the EWPL. Given the weight of a particular edge in the final snapshot of real graphs (x-axis), the multiplication of total weights(y-axis) of the edges incident to two neighboring nodes follow a power law. A line can be fit to the median values after logarithmic binning on the x-axis. Upper and lower bars indicate 75% and 25% of the data, respectively.

and j with weights w_i and w_j , respectively,

$$w_{i,j} \propto \left(\sqrt{(w_i - w_{i,j}) * (w_j - w_{i,j})} \right)^\gamma$$

We report corresponding experimental findings in Fig. 3.

2.2.3 SW-3: Snapshot Power Laws (SPL). What about a static snapshot of a graph? If node i has out-degree out_i , what can we say about its out-weight $outw_i$? It turns out that there is a “fortification effect” here, too, resulting in more power laws, both for out-degrees/out-weights as well as for in-degrees/in-weights.

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. An example of such a plot is in Fig. 2.4 (c) and (d). 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 [26], and, for each bin, we compute the median y .

We observed that the median values of weights versus mid-points of the intervals follow a power law for all datasets studied. Formally, the “Snapshot Power Law” is:

OBSERVATION 2.3 (SNAPSHOT POWER LAW (SPL)) Consider the i -th node of a weighted graph, at time t , and let out_i , $outw_i$ be its out-degree and out-weight. Then

$$outw_i \propto out_i^{ow}$$

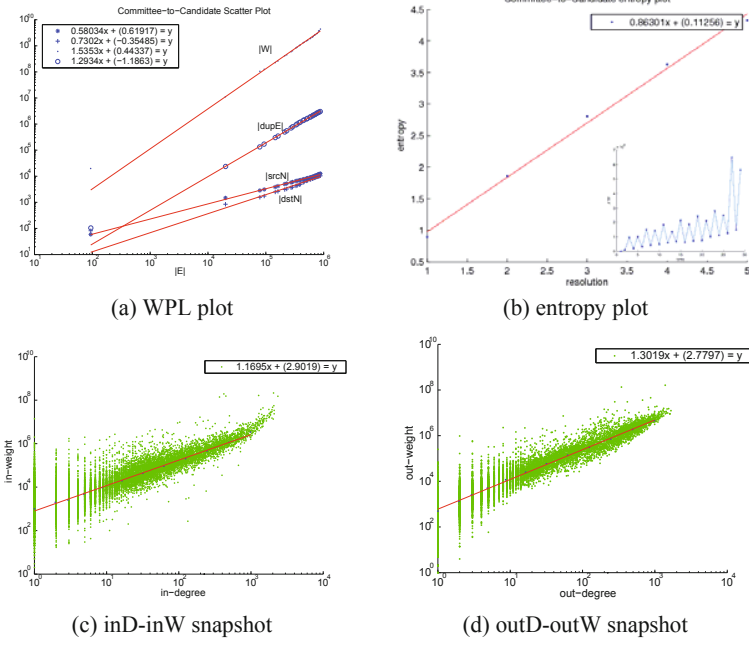


Figure 2.4. Weight properties of *CampOrg* donations: (a) shows all the power laws as well as the WPL; the slope in (b) is ~ 0.86 indicating bursty weight additions over time; (c) and (d) have slopes > 1 (“fortification effect”), that is, that the more campaigns an organization supports, the superlinearly-more 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.

where ow is the out-weight-exponent of the SPL. Similarly, for the in-degree, with in-weight-exponent iw .

We studied the snapshot plots for several time-stamps (for brevity, we only report the slopes for the final timestamp in Table 2 for all the datasets we studied). We observed that SPL exponents of a graph over time remains almost constant. In Fig. 2.4 (c) ((d)), the inset plot shows how the $iw(ow)$ exponent changes over time (years) for the *CampOrg* dataset. We notice that iw and ow take values in the range $[0.9-1.2]$ and $[0.95-1.35]$, respectively. That is:

OBSERVATION 2.4 (PERSISTENCE OF SNAPSHOT POWER LAW) *The in- and out-exponents iw and ow of the SPL remain about constant, over time.*

Looking at Table 2, we observe that all SPL exponents are > 1 , which imply a “fortification effect” with super-linear growth.

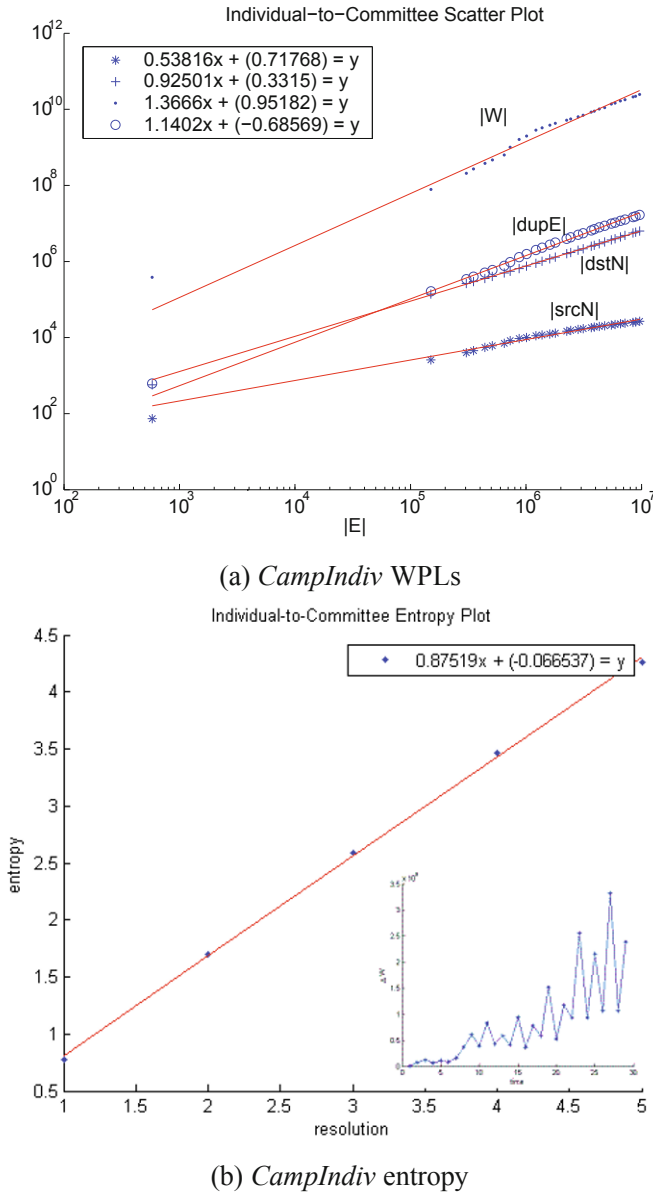


Figure 2.5. Properties of weighted networks. Top: weight power laws for *CampIndiv*(W , E_d , N ; vs E). The slopes for weight W and multi-edges E_d are above 1, indicating “fortification”. Bottom: entropy plots for weight addition. Slope away from 1 indicates burstiness (eg., 0.88 for *CampIndiv*) The inset plot shows the corresponding time sequence ΔW versus time.

	<i>w</i>	<i>nsrc</i>	<i>ndst</i>	<i>dupE</i>	<i>iw</i>	<i>ow</i>	<i>fd</i>
<i>CampOrg</i>	1.53	0.58	0.73	1.29	1.16	1.30	0.86
<i>CampIndiv</i>	1.36	0.53	0.92	1.14	1.05	1.48	0.87
<i>BlogNet</i>	1.03	0.79	NA	NA	1.01	1.10	0.96
<i>Auth-Key</i>	1.01	0.90	0.70	NA	1.01	1.04	0.95
<i>Auth-Conf</i>	1.08	0.96	0.48	NA	1.04	1.81	0.96
<i>Key-Conf</i>	1.22	0.85	0.54	NA	1.26	2.14	0.95

Table 2.3. Power law exponents for all the weighted datasets we studied: The x-axis being the number of non-duplicate edges E , w : WPL exponent, $nsrc$, $ndst$: WPL exponent for source and destination nodes respectively (if the graph is unipartite, then $nsrc$ is the number of all nodes), $dupE$: exponent for multi-edges, iw , ow : SPL exponents for indegree and outdegree of nodes, respectively. Exponents above 1 indicate fortification/superlinear growth. Last column, fd : slope of the entropy plots, or information fractal dimension. Lower fd means more burstiness.

3. Dynamic Properties

We next present several *dynamic* properties. These are typically studied by looking at a *series* of static snapshots and seeing how measurements of these snapshots compare. Like the static properties we presented previously, we also divide these into properties that take into account weights and those that don't.

3.1 Dynamic Unweighted Graphs

The patterns in dynamic time-evolving graphs that do not consider edge weights include the shrinking diameter property, the densification law, oscillating around a constant size secondary largest connected components, the largest eigenvalue law and the bursty and self-similar edge additions over time. We next describe these laws in detail.

3.1.1 D-1: Shrinking Diameter. Leskovec. et al. [18] showed that not only is the diameter of real graphs small, but it also *shrinks* and then *stabilizes* over time [18]. This pattern can be attributed to the ‘gelling point’ and the ‘densification’ in real graphs both of which are described in the following sections. Briefly, at the ‘gelling point’ many small disconnected components merge and form the largest connected component in the graph. This can be thought as the ‘coalescence’ of the graph at which point the diameter ‘spikes’. Afterwards, with the addition of new edges the diameter keeps shrinking until it reaches an equilibrium.

3.1.2 D-2: Densification Power Law (DPL). Time-evolving graphs follow the ‘Densification Power Law’ with the equation $E(t) \propto N(t)^\beta$, at all

time ticks t [18], where β is the densification exponent, and $E(t)$ and $N(t)$ are the number of edges and nodes at time t , respectively.

All our real graphs we studied obeyed the DPL, with exponents between 1.03 and 1.7. The power-law exponent being greater than 1 indicates a super-linearity between the number of nodes and the number of edges in real graphs. That is, it indicates that for example when the number of nodes N in a graph doubles, the number of edges E more than doubles—hence the densification. It also explains away the shrinking diameter phenomenon observed in real graphs described earlier. We will attempt to reproduce this property in a generative model later in this chapter.

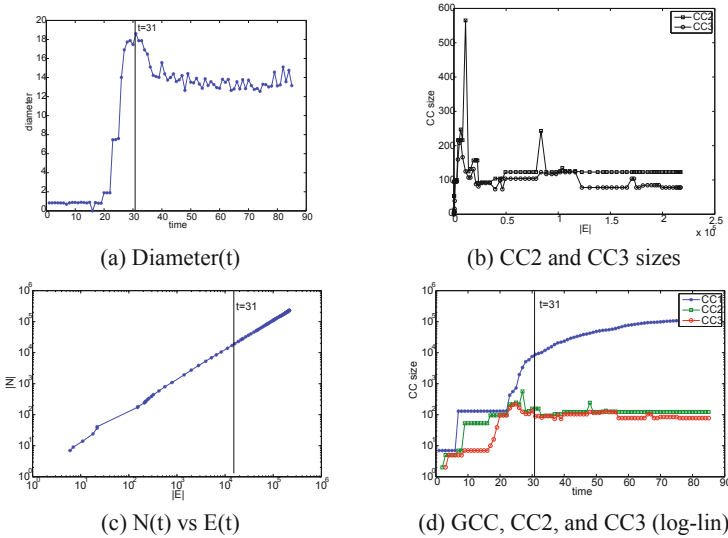


Figure 2.6. Properties of *PostNet* network. Notice that we experience an early gelling point at (a) (diameter versus time), stabilization/oscillation of the NLCC sizes in (b) (size of 2nd and 3rd CC, versus time). The vertical line marks the gelling point. Part (c) gives $N(t)$ vs $E(t)$ in log-log scales - the good linear fit agrees with the Densification Power Law. Part (d): component size (in log), vs time - the GCC is included, and it clearly dominates the rest, after the gelling point.

3.1.3 D-3: Diameter-plot and Gelling point. Studying the effective diameter of the graphs, we notice that there is often a point in time when the diameter spikes. Before that point, the graph is more or less in an establishment period, typically consisting of a collection of small, disconnected components. This “gelling point” seems to also be the time where the GCC “takes off”. After the gelling point, the graph obeys the expected rules, such as the den-

sification power law; its diameter decreases or stabilizes; the giant connected component keeps growing, absorbing the vast majority of the newcomer nodes.

OBSERVATION 2.5 (GELLING POINT) *Real graphs exhibit a gelling point, at which the diameter spikes and (several) disconnected components gel into a giant component.*

In most of these graphs, both unipartite and bipartite, there are clear gelling points. For example, in *NIPS* the diameter spikes at $t = 8$ years, which is a reasonable time for an academic community to gel. In some networks, we only see one side of the spike, due to massive network size (*Patent*).

We show full results for *PostNet* in Fig. 2.6, including the diameter plot (Fig. 2.6(a)), sizes of the NLCCs (Fig. 2.6(b)), densification plot (Fig. 2.6(c)), and the sizes of the three largest connected components in log-linear scale, to observe how the GCC dominates the others (Fig. 2.6(d)). Results from other networks are similar, and are shown in condensed form for space (Fig. 2.7 for unipartite graphs, and Fig. 2.8 for bipartite graphs). The left column shows the diameter plots, and the right column shows the NLCCs, which we describe next.

3.1.4 D-4: Constant/Oscillating NLCCs. We particularly studied the second and the third connected component over time. We notice that, after the gelling point, the sizes of these components *oscillate* over time. Further investigation shows that the oscillation may be explained as follows: newcomer nodes typically link to the GCC; very few of the newcomers link to the 2nd (or 3rd) CC, helping them to grow slowly; in very rare cases, a newcomer links both to an NLCC, as well as the GCC, thus leading to the absorption of the NLCC into the GCC. It is exactly at these times that we have a drop in the size of the 2nd CC: Note that edges are not removed, thus, what is reported as the size of the 2nd CC is actually the size of yesterday’s 3rd CC, causing the apparent “oscillation”.

An unexpected (to us, at least) observation is that the largest size these components can get seems to be a constant. This is counter-intuitive – based on random graph theory, we would expect the size of the NLCCs to grow with increasing N . Using scale-free arguments, we would expect the NLCCs to have size that would be a (small, but constant) fraction of the size of the GCC – to our surprise, this *never* happened, on any of the real graphs we tried. If some underlying growth does exist, it was small enough to be impossible to observe throughout the (often lengthy) time in the datasets.

The second columns of Fig. 2.7 and Fig. 2.8 show the NLCC sizes versus time. Notice that, after the “gelling” point (marked with a vertical line), they all oscillate about constant value (different for each network). The only extreme cases are datasets with unusually high connectivity. For example, *Netflix* has

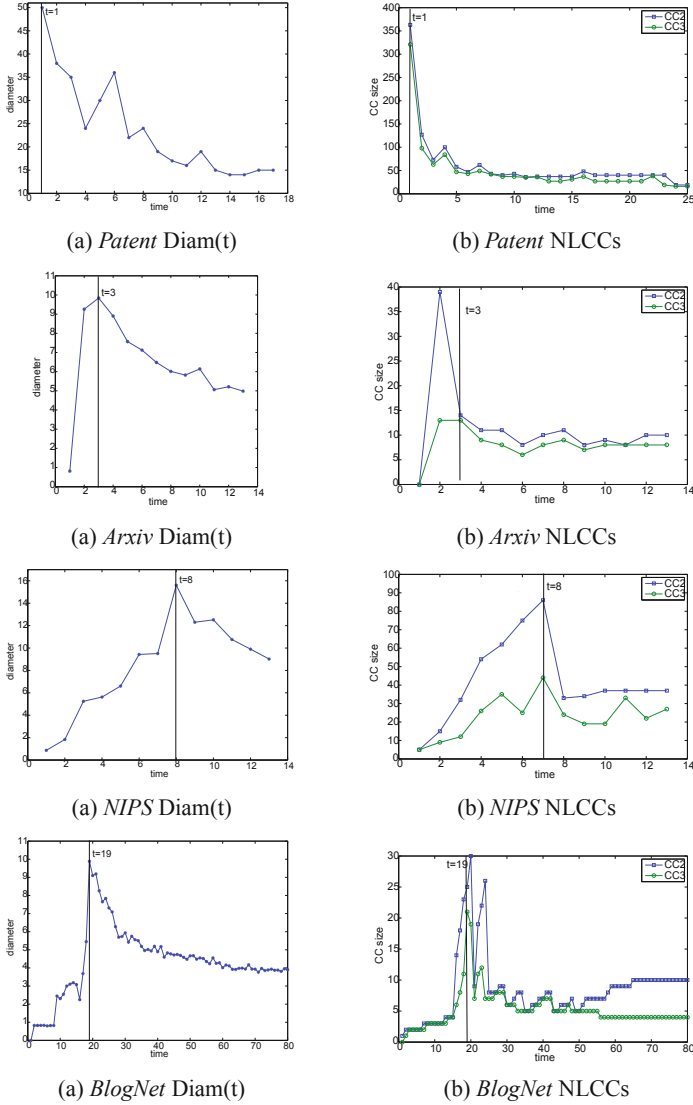


Figure 2.7. Properties of other unipartite networks. Diameter plot (left column), and NLCCs over time (right); vertical line marks the gelling point. All datasets exhibit an early gelling point, and stabilization of the NLCCs.

very small NLCCs. This may be explained by the fact the dataset is masked, omitting users with less than a hundred ratings (possibly to further protect the

privacy of the encrypted user-ids). Therefore, the graph has abnormally high connectivity.

OBSERVATION 2.6 (OSCILLATING NLCCs) *After the gelling point, the secondary and tertiary connected components remain of approximately constant size, with small oscillations.*

3.1.5 D-5: LPL: Principal eigenvalue over time. Plotting the largest (principal) eigenvalue of the 0-1 adjacency matrix \mathbf{A} of our datasets over time, we notice that the principal eigenvalue grows following a power law with increasing number of edges. This observation is true especially after the *gelling point*. The ‘gelling point’ is defined to be the point at which a giant connected component (GCC) appears in real-world graphs - after this point, properties such as densification and shrinking diameter become increasingly evident. See [18] for details.

OBSERVATION 2.7 (λ_1 POWER LAW (LPL)) *In real graphs, the principal eigenvalue $\lambda_1(t)$ and the number of edges $E(t)$ over time follow a power law with exponent less than 0.5, especially after the ‘gelling point’. That is,*

$$\lambda_1(t) \propto E(t)^\alpha, \alpha \leq 0.5$$

We report the power law exponents in Fig. 2.9. Note that we fit the given lines *after* the gelling point which is shown by a vertical line for each dataset. Notice that the given slopes are less than 0.5, with the exception of the *CampaignOrg* dataset, with slope ≈ 0.53 . This result is in agreement with graph theory. See [1] for details.

3.2 Dynamic Weighted Graphs

3.2.1 DW-1: Bursty/self-similar weight additions. We tracked how much weight a graph puts on at each time interval and looking at the entropy plots, we observed that the weight additions over time show self-similarity. For those weighted graphs where the edge weight is defined as the number of reoccurrences of that edge, the slope of the entropy plot was greater than 0.95, pointing out uniformity. On the other hand, for those graphs where weight is not in terms of multiple edges but some other feature of the dataset such as the amount of donations for the FEC dataset, we observed that weight additions are more bursty, the slope being as low as 0.6 for the Network Traffic dataset. Fig. 2.5 (b) column shows the entropy plots for the weighted datasets we studied. ΔW values over time are also shown in insets at the bottom right corner of each figure.

OBSERVATION 2.8 (BURSTY/SELF-SIMILAR WEIGHT ADDITIONS) *In all our graphs, the addition of weight ($\Delta W(t)$) was self-similar, with fractal dimension ranging from ≈ 1 (smooth/uniform), down to 0.6 (bursty).*

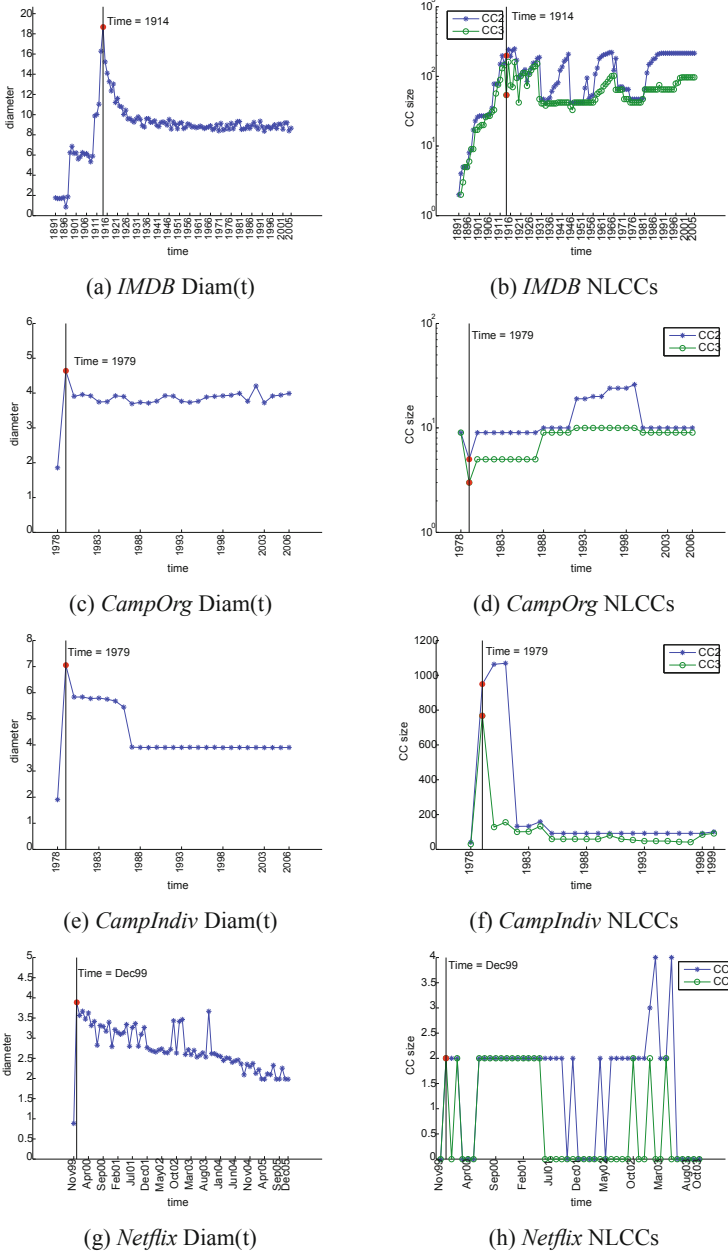


Figure 2.8. Properties of bipartite networks. Diameter plot (left column), and NLCCs over time (right), with vertical line marking the gelling point. Again, all datasets exhibit an early gelling point, and stabilization of the NLCCs. *Netflix* has strange behavior because it is masked (see text).

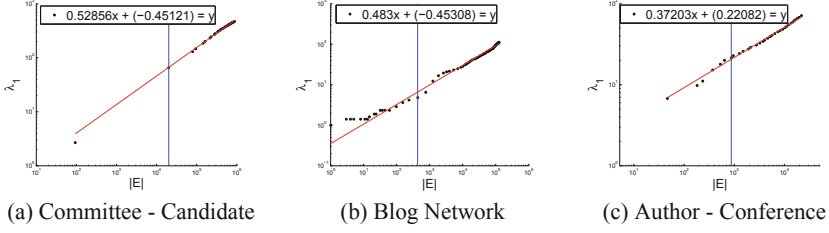


Figure 2.9. Illustration of the LPL. 1st eigenvalue $\lambda_1(t)$ of the 0-1 adjacency matrix \mathbf{A} versus number of edges $E(t)$ over time. The vertical lines indicate the gelling point.

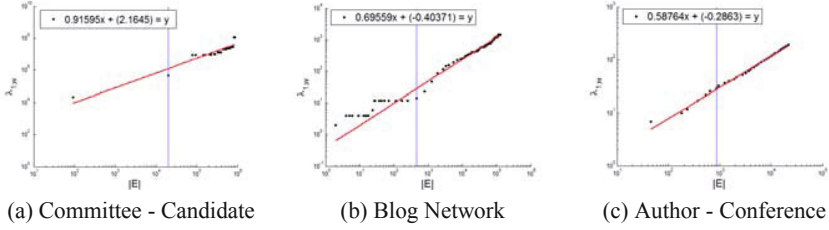


Figure 2.10. Illustration of the LWPL. 1st eigenvalue $\lambda_{1,w}(t)$ of the weighted adjacency matrix \mathbf{A}_w versus number of edges $E(t)$ over time. The vertical lines indicate the gelling point.

3.2.2 DW-2: LWPL: Weighted principal eigenvalue over time. Given that unweighted (0-1) graphs follow the λ_1 Power Law, one may ask if there is a corresponding law for weighted graphs. To this end, we also compute the largest eigenvalue $\lambda_{1,w}$ of the *weighted* adjacency matrix \mathbf{A}_w . The entries $w_{i,j}$ of \mathbf{A}_w now represent the actual edge weight between node i and j . We notice that $\lambda_{1,w}$ increases with increasing number of edges following a power law with a higher exponent than that of its λ_1 Power Law. We show the experimental results in Fig. 2.10.

OBSERVATION 2.9 ($\lambda_{1,w}$ POWER LAW (LWPL)) *Weighted real graphs exhibit a power law for the largest eigenvalue of the weighted adjacency matrix $\lambda_{1,w}(t)$ and the number of edges $E(t)$ over time. That is,*

$$\lambda_{1,w}(t) \propto E(t)^\beta$$

In our experiments, the exponent β ranged from 0.5 to 1.6.

4. Conclusion

We believe that the *Butterfly* model and the observation of constant NLCC's will shed light upon other research in the area, such as a recent, counter-intuitive discovery [20]: the GCC of several real graphs has *no* good cuts, so graph partitioning and clustering algorithms cannot help identify communities because no clear communities exist.

We have described the following static patterns:

- *Heavy-tailed degree distribution*, with a few “hubs” and most nodes having few neighbors.
- *Small diameter and community structure*— nodes form clusters, and it takes few “hops” to get between any two nodes in the network.
- Several power laws: *Triangle Power Law* and *Eigenvalue Power Law* for unweighted graphs, and the *Weight Power Law*, *Edge Weights Power Law*, and *Snapshot Power Laws* for weighted graphs.

We have also described the following dynamic patterns:

- *Shrinking diameter and densification*— the “world gets smaller” as more nodes are added— increasingly more edges are added which causes the diameter to shrink. There is also a *gelling point* at which this occurs.
- *Constant-size smaller components* The large component takes off in size, but the others will not grow beyond a certain point before joining it.
- Several other power laws: *LPL*, or principal eigenvalue over time (both weighted and unweighted), and *bursty weight additions*.

These patterns are helpful to spot anomalous graphs and sub-graphs, and answer questions about entities in a network and what-if scenarios. Let's elaborate on each of the above applications: Spotting anomalies is vital for determining abuse of social and computer networks, such as link-spamming in a web graph, fraudulent reputation building in e-auction systems [29], detection of dwindling/abnormal social sub-groups in a social-networking site like Yahoo-360 (360.yahoo.com), Facebook (www.facebook.com) and LinkedIn (www.linkedin.com), and network intrusion detection [17]. Analyzing network properties is also useful for identifying authorities and search algorithms [7, 9, 16], for discovering the “network value” of customers for using viral marketing [30], or to improve recommendation systems [5]. What-if scenarios are vital for extrapolation, provisioning and algorithm design: For example if we expect that the number of links will double within the next year, we should provision for the appropriate hardware to store and process the upcoming queries.

References

- [1] L. Akoglu, M. McGlohon, and C. Faloutsos. RTM: Laws and a recursive generator for weighted time-evolving graphs. *Carnegie Mellon University Technical Report*, Oct, 2008.
- [2] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Diameter of the world wide web. *Nature*, (401):130–131, 1999.
- [3] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.
- [4] Albert-Laszlo Barabasi. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003.
- [5] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM.
- [6] Zhiqiang Bi, Christos Faloutsos, and Filip Korn. The DGX distribution for mining massive, skewed data. In *KDD*, pages 17–26, ACMA, 2001. ACM.
- [7] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Trans. Inter. Tech.*, 5(1):231–297, 2005.
- [8] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-MAT: A recursive model for graph mining. *SIAM Int. Conf. on Data Mining*, April 2004.
- [9] Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the web’s link structure. *Computer*, 32(8):60–67, 1999.
- [10] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661+, Feb 2009.
- [11] Pedro Domingos and Matt Richardson. Mining the network value of customers. *KDD*, pages 57–66, 2001.
- [12] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *SIGCOMM*, pages 251–262, Aug-Sept. 1999.
- [13] Gary Flake, Steve Lawrence, C. Lee Giles, and Frans Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3), March 2002.

- [14] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99:7821, 2002.
- [15] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17, 1999.
- [16] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Core algorithms in the clever system. *ACM Trans. Inter. Tech.*, 6(2):131–152, 2006.
- [17] Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.
- [18] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of ACM SIGKDD*, pages 177–187, Chicago, Illinois, USA, 2005. ACM Press.
- [19] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, New York, NY, USA, 2005. ACM Press.
- [20] Jure Leskovec, Kevin Lang, Anirban Dasgupta, and Michael Mahoney. Community structure in real graphs: The “negative dimensionality” paradox. In *International World Wide Web Conference*, 2008.
- [21] Jure Leskovec, Mary Mcglohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. Cascading behavior in large blog graphs: Patterns and a model. In *Society of Applied and Industrial Mathematics: Data Mining (SDM07)*, 2007.
- [22] 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.
- [23] M. Mihail and C. Papadimitriou. The eigenvalue power law, 2002.
- [24] S. Milgram. The small-world problem. *Psychology Today*, 2:60–67, 1967.
- [25] Alan L. Montgomery and Christos Faloutsos. Identifying web browsing trends and patterns. *IEEE Computer*, 34(7):94–95, July 2001.
- [26] M. E. J. Newman. Power laws, pareto distributions and zipf’s law. *Contemporary Physics*, 46, 2005.

- [27] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.
- [28] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. Anf: A fast and scalable tool for data mining in massive graphs. In *SIGKDD*, Edmonton, AB, Canada, 2002.
- [29] Shashank Pandit, Duen H. Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 201–210, New York, NY, USA, 2007.
- [30] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing, 2002.
- [31] Manfred Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. W.H. Freeman and Company, New York, 1991.
- [32] Michael F. Schwartz and David C. M. Wood. Discovering shared interests among people using graph analysis of global electronic mail traffic. *Communications of the ACM*, 36:78–89, 1992.
- [33] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Power laws and the AS-level internet topology, 2003.
- [34] G. Siganos, S. L. Tauro, and M. Faloutsos. Jellyfish: a conceptual model for the as internet topology. *Journal of Communications and Networks*, 2006.
- [35] SL Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the Internet topology. 2001.
- [36] Charalampos E. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *ICDM*, 2008.
- [37] Mengzhi Wang, Tara Madhyastha, Ngai Hang Chang, Spiros Papadimitriou, and Christos Faloutsos. Data mining meets performance evaluation: Fast algorithms for modeling bursty traffic. *ICDE*, February 2002.
- [38] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, (393):440–442, 1998.



<http://www.springer.com/978-1-4419-8461-6>

Social Network Data Analytics

Aggarwal, C.C. (Ed.)

2011, XIV, 502 p., Hardcover

ISBN: 978-1-4419-8461-6