An In-Depth Study of Stochastic Kronecker Graphs

C. Seshadhri, Ali Pinar, and Tamara G. Kolda
Sandia National Laboratories, Livermore, CA, USA
{*scomand,apinar,tgkolda*}*@sandia.gov*

***Abstract*—Graph analysis is playing an increasingly important role in science and industry. Due to numerous limitations in sharing real-world graphs, models for generating massive graphs are critical for developing better algorithms. In this paper, we analyze the stochastic Kronecker graph model (SKG), which is the foundation of the Graph500 supercomputer benchmark due to its many favorable properties and easy parallelization. Our goal is to provide a deeper understanding of the parameters and properties of this model so that its functionality as a benchmark is increased. We develop a rigorous mathematical analysis that shows this model *cannot* generate a power-law distribution or even a lognormal distribution. However, we formalize an enhanced version of the SKG model that uses random noise for smoothing. We prove both in theory and in practice that this enhancement leads to a lognormal distribution. Additionally, we provide a precise analysis of isolated vertices, showing that graphs that are produced by SKG might be quite different than intended. For example, between 50% and 75% of the vertices in the Graph500 benchmarks will be isolated. Finally, we show that this model tends to produce extremely small core numbers (compared to most social networks and other real graphs) for common parameter choices.**

***Keywords*-Stochastic Kronecker Graphs, Graph Mining, Social Networks, Graph500, Lognormal Degree Distributions, Random Graph Generation**

I. Introduction

The role of graph analysis is becoming increasingly important in science and industry because of the prevalence of graphs in diverse scenarios such as social networks, the Web, power grid networks, and even scientific collaboration studies. Massive graphs occur in a variety of situations, and we need to design better and faster algorithms in order to study them. However, it can be very difficult to get access to informative large graphs in order to test our algorithms. Companies like Netflix, AOL, and Facebook have vast arrays of data but cannot share it due to legal or copyright issues[1]. Moreover, graphs with billions of vertices cannot be communicated easily due to their sheer size.

As was noted in [1], good *graph models* are extremely important for the study and algorithmics of real networks. Such a model should be fairly easy to implement and have few parameters, while exhibiting the common properties of real networks. Furthermore, models are needed to test algorithms and architectures designed for large graphs. But the theoretical and research benefits are also obvious: gaining insight into the properties and processes that create real networks.

The *Stochastic Kronecker graph* (SKG) [2], [3], a generalization of *recursive matrix* (R-MAT) model [4], has been proposed for these purposes. It has very few parameters and can generate large graphs quickly. Indeed, it is one of the few models that can generate graphs fully in *parallel*. It has been empirically observed to have interesting real-network-like properties. We stress that this is not just of theoretical or academic interest—this model has been chosen to create graphs for the Graph500 supercomputer benchmark [5].

It is important to know how the parameters of this model affect various properties of the graphs. A mathematical analysis is important for understanding the inner working of a model. We quote Mitzenmacher [6]: "I would argue, however, that without validating a model it is not clear that one understands the underlying behavior and therefore how the behavior might change over time. It is not enough to plot data and demonstrate a power law, allowing one to say things about current behavior; one wants to ensure that one can accurately predict future behavior appropriately, and that requires understanding the correct underlying model."

A. Notation and Background

We explain the SKG model and notation. Our goal is to generate a graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. The general form of the SKG model allows for an arbitrary square generator matrix and assumes that n is a power of its size. Here, we focus on the 2×2 case (which is equivalent to R-MAT), defining the generating matrix as

$$T = \begin{bmatrix} t_1 & t_2 \\ t_3 & t_4 \end{bmatrix} \quad \text{with} \quad t_1 + t_2 + t_3 + t_4 = 1.$$

We assume that $n = 2^\ell$ for some integer $\ell > 0$. *For the sake of cleaner formulae, we assume that ℓ is even in our analyses.* Each edge is inserted according to the probabilities[2] defined by

$$P = \underbrace{T \otimes T \otimes \cdots \otimes T}_{\ell \text{ times}}.$$

In practice, the matrix P is never formed explicitly. Instead, each edge is inserted as follows. Divide the adjacency

[1] http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html

[2] We have taken a slight liberty in requiring the entries of T to sum to 1. In fact, the SKG model as defined in [3] works with the matrix mP, which is considered the matrix of probabilities for the existence of each individual edge (though it might be more accurate to think of it as an expected value).

matrix into four quadrants and choose one of them with the corresponding probability t_1, t_2, t_3, or t_4. Once a quadrant is chosen, repeat this recursively in that quadrant. Each time we iterate, we end up in a square submatrix whose dimensions are exactly halved. After ℓ iterations, we reach a single cell of the adjacency matrix, and an edge is inserted.

Note that all edges can be inserted in parallel. This is one of the major advantages of the SKG model and why it is appropriate for generating large supercomputer benchmarks.

B. Our Contributions

Our overall contribution is to provide a thorough study of the properties of SKGs and show how the parameters affects these properties. We focus on the degree distribution, the number of isolated nodes, the core sizes, and the trade-offs in these various goals. We give rigorous mathematical theorems and proofs explaining the degree distribution of SKG, a noisy version of SKG, and the number of isolated vertices. Due to space restrictions, full proofs are not included in this paper, but are available at [7].

1) **Degree distribution:** We provide a rigorous mathematical analysis of the degree distribution of SKGs. The degree distribution has often been claimed to be power-law, or sometimes as lognormal [4], [3], [8]. Kim and Leskovec [8] prove that the degree distribution has some lognormal characteristics. Groër et al. [9] give exact series expansions for the degree distribution, and express it as a mixture of normal distributions. This provides a qualitative explanation for the oscillatory behavior of the degree distribution (refer to Fig. 1). Since the distribution is quite far from being truly lognormal, there has been no simple closed form expression that closely approximates it. We fill this gap by providing a complete mathematical description. We prove that SKG *cannot* generate a power law distribution, or even a lognormal distribution. It is most accurately characterized as fluctuating between a lognormal distribution and an exponential tail. We provide a fairly simple formula that approximates the degree distribution.

2) **Noisy SKG:** It has been mentioned in passing [4] that adding noise to SKG at each level smoothens the degree distribution, but this has never been formalized or studied. We define a specific noisy version of SKG (NSKG). We prove theoretically and empirically that NSKG leads to a lognormal distribution. The lognormal distribution is important since it has been observed in real data [10], [11], [12], [13]. One of the major benefits of our enhancement is that only ℓ additional random numbers are needed in total. Using Graph500 parameters, Fig. 1 plots the degree distribution of standard SKG and NSKG for two levels of noise. We clearly see that noise dampens the oscillations, leading to a lognormal distribution.

These results have been communicated to the Graph500 committee, which has decided to update the Graph500 benchmark (next year) to our proposed NSKG model [14].

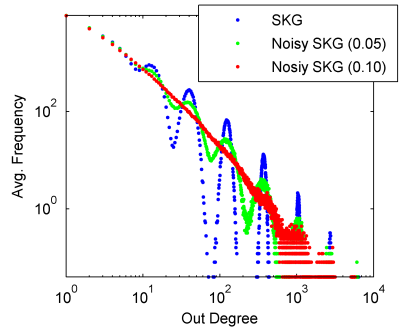


Figure 1: Comparison of degree distributions (averaged over 25 instances) for SKG and two noisy variations, using the T from the Graph500 Benchmark parameters with $\ell = 16$.

Table I: Expected percentage of isolated vertices and repeat edges (according to [9]), along with average degree of *non-isolated* nodes for the Graph 500 benchmark. Excluding the isolated vertices results in a much higher average degree than the value of 16 that is specified by the benchmark.

ℓ	% Isolated Nodes	% Repeat Edges	Avg. Degree
26	51	1.2	32
29	57	0.7	37
32	62	0.4	41
36	67	0.2	49
39	71	0.1	55
42	74	0.1	62

3) **Isolated vertices:** An isolated vertex is one that has no edges incident to it (and hence is not really part of the output graph). We provide an easy to compute formula that (very accurately) estimates the fraction of isolated vertices. We discover the rather surprising result that in the Graph500 benchmark graphs, 50-75% vertices are isolated; see Tab. I. This is a major concern for the benchmark, since the massive graph generated has a much reduced size. Furthermore, the average degree is now much higher than expected.

Our analysis solves the mystery of isolated vertices, and is being used by the Graph500 committee to design next year's benchmark [14].

4) **Core numbers:** The study of k-cores is an important tool used to study the structure of social networks because it is a mark of the connectivity and special processes that generate these graphs [1], [15], [16], [17], [18], [19], [20]. We empirically show how the core numbers have surprising correlations with SKG parameters. We observed that for most of the current SKG parameters used for modeling real graphs, max core numbers are extremely small (much smaller than most corresponding real graphs). We show how modifying the matrix T affects core numbers. Most strikingly, we observe that changing T to increase the max core number actually leads to an increase in the fraction of

isolated vertices.

C. Parameters for empirical study

Throughout the paper, we discuss a few sets of SKG parameters. The first is the Graph500 benchmark [5]. The other two are parameters used in [3] to model a co-authorship network (CAHepPh) and a web graph (WEBNotreDame). We list these parameters here for later reference.

- **Graph500**: $T = [0.57, 0.19; 0.19, 0.05]$, $\ell \in \{26, 29, 32, 36, 39, 42\}$, and $m = 16 \cdot 2^{\ell}$.
- **CAHepPh**: $T = [0.42, 0.19; 0.19, 0.20]$, $\ell = 14$, and $m = 237,010$.
- **WEBNotreDame**[3]: $T = [0.48, 0.20; 0.21, 0.11]$, $\ell = 18$, and $m = 1,497,134$.

II. Previous Work

The R-MAT model was defined by Chakrabarti et al. [4]. The general and more powerful SKG model was introduced by Leskovec et al. [21] and fitting algorithms were proposed by Leskovec and Faloutsos [2] (combined in [3]). This model has generated significant interest and notably was chosen for the Graph500 benchmark [5]. Kim and Leskovec [8] defined the Multiplicative Attribute Graph (MAG) model, a generalization of SKG where each level may have a different matrix T. They suggest that certain configurations of these matrices could lead to power-law distributions.

Since the appearance of the SKG model, there have been analyses of its properties. The original paper [3] provides some basic theorems and empirically show a variety of properties. Mahdian and Xu [22] specifically study how the model *parameters* affect the graph properties. They show phase transition behavior (asymptotically) for occurrence of a large connected component and shrinking diameter. They also initiate a study of isolated vertices. When the SKG parameters satisfy a certain condition, then the number of isolated vertices *asymptotically* approaches n. Their theorems are quite strong, but do not give information about the number of isolated vertices for a fixed SKG instance. In the analysis of the MAG model [8], it is shown that the SKG degree distribution has some lognormal characteristics. (Lognormal distributions have been observed in real data [10], [11], [13]. Mitzenmacher [12] gives a survey of lognormal distributions.)

Sala et al. [23] perform an extensive empirical study of properties of graph models, including SKGs. Miller et al. [24] show that they can detect anomalies embedded in an SKG. Moreno et al. [25] study the distributional properties of families of SKGs.

As noted in [4], the SKG generation procedure may give repeated edges. Hence, the number of edges in the graph differs slightly from the number of insertions (though, in practice, this is barely 1% for Graph500). Groër et al. [9] prove that the number of vertices of a given degree is normally distributed, and provide algorithms to compute the expected number of edges in the graph (as a function of the number of insertions) and the expected degree distribution.

[3] In [3], ℓ was 19. We make it even because, for the sake of presentation, we perform experiments and derive formulae for even ℓ.

III. Degree Distribution

In this section, we analyze the degree distribution of SKGs, which are known to follow a multinomial distribution. While an exact expression for this distribution can be written, this is unfortunately a complicated sum of binomial coefficients. Eyeballing the log-log plots of the degree distribution, one sees a general heavy-tail like behavior, but there are large oscillations. The degree distribution is far from being monotonically decreasing. Refer to Fig. 2 to see some examples of SKG degree distributions (plotted in log-log scale). Groër et al. [9] show that the degree distribution behaves like the sum of Gaussians, giving some intuition for the oscillations. Recent work of Kim and Leskovec [8] provide some mathematical analysis explaining connections to a lognormal distribution. But this is only the beginning of the story. What does the distribution oscillate between? Is the distribution bounded below by a power law? Can we approximate the distribution with a simple closed form function? None of these questions have satisfactory answers.

Our analysis gives a precise explanation for the SKG degree distribution. We prove that the SKG degree distribution oscillates between a lognormal and exponential tail, and we precisely characterize how. We provide plots and experimental results to back up and provide more intuition for our theorems.

The oscillations are a somewhat disappointing feature of SKG. Real degree distributions do not have large oscillations (they are by and large monotonically decreasing), and more importantly, do not have any exponential tail behavior. This is a major issue both for modeling and benchmarking purposes, since degree distribution is one of the primary characteristics that distinguishes real networks.

But how do we rectify the oscillations of the SKG degree distribution? We apply a certain model of noise to SKG and provide both mathematical and empirical evidence that this "straightens out" the degree distribution. Indeed, small amounts of noise lead to a degree distribution that is predominantly lognormal. This also shows a very appealing aspect of our degree distribution analysis. We can very naturally explain how noise affects the degree distribution and give explicit bounds on these affects.

We set some parameters that grant simplified expressions.

- $\Delta = m/n$ (average degree)
- $\sigma = (t_1 + t_2) - 1/2$ (We refer to this as the *skew*.)
- $\tau = (1 + 2\sigma)/(1 - 2\sigma)$
- $\lambda = \Delta(1 - 4\sigma^2)^{\ell/2}$

Slices: The vertices of the graph are numbered from 0 to $n-1$. Each vertex has an ℓ-bit binary representation and therefore corresponds to an element of the boolean hypercube $\{0,1\}^\ell$. We can partition the vertices into *slices*, where each slice consists of vertices whose representations have the same number of 0's (same Hamming weight). Recall that we assume ℓ is even. For $r \in [-\ell/2, \ell/2]$, we say that *slice* r consists of all vertices whose binary representations have exactly $(\ell/2 + r)$ 0's.

These binary representations and slices are intimately connected with edge insertions in the SKG model. For each insertion, we are trying to randomly choose a source-sink pair. First, let us simply choose the first bit (of the representations) of the source and the sink. Note that there are 4 possibilities (first bit for source, second for sink): 00, 01, 10, and 11. We choose one of the combinations with probabilities t_1, t_2, t_3, and t_4 respectively. This fixes the first bit of the source and sink. We perform this procedure again to choose the second bit of the source and sink. Repeating ℓ times, we finally decide the source and sink of the edge. Since t_1 is the largest value, we tend to add more edges between vertices that have many zeroes. Note that as r becomes larger, a vertex in an r-slice tends to have higher degree.

A. Analysis

We begin by stating and explaining the main result of this section. The next subsection gives a verbal explanation of the results and the intuition behind how we proved them. Recall that, for the sake of presentation, we assume that ℓ is even. All theorems can be suitably modified for the general case.

For a real number x, $\lfloor x \rceil$ is the closest integer to x. We use $o(1)$ as a shorthand for a quantity that is negligible. Typically, this becomes asymptotically zero very rapidly as d or ℓ increases. We write $A = (1 \pm o(1))B$ to indicate that the quantities A and B only differ by a lower order term. To provide clean expressions, we make certain approximations which are slightly off for certain regions of d and ℓ (essentially, when d is either too small or too large). Furthermore, as our figures will make amply clear, our expressions very tightly approximate the degree distribution. The following theorem provides a fairly simple closed form upper bound. We focus on outdegrees, but analogous theorems hold for indegrees as well.

Theorem 1: For degree d, let $\theta_d = \ln(d/\lambda)/\ln\tau$. Define $\Gamma_d = \lfloor \theta_d \rceil$ and $\gamma_d = |\theta_d - \Gamma_d|$. The expected outdegree distribution of a SKG is bounded above by a function that oscillates between a lognormal and an exponential tail. Formally, assume $(e\ln 2)\ell \le d \le \sqrt{n}$. If $\Gamma_d \ge \ell/2$, then the expected number of vertices of degree d is negligible (expectation is $o(1)$). If $\Gamma_d < \ell/2$, the expected number of vertices of degree d is bounded above (up to a small constant factor) by

$$\frac{1}{\sqrt{d}} \exp\left(\frac{-d\gamma_d^2 \ln^2 \tau}{2}\right) \binom{\ell}{\ell/2 + \Gamma_d}.$$

Note that $\Gamma_d = \lfloor \ln(d/\lambda)/\ln\tau \rceil = \Theta(\ln d)$. Hence $\binom{\ell}{\ell/2+\Gamma_d}$ can be thought of as $\binom{\ell}{\ell/2+\Theta(\ln d)}$. The function $\binom{\ell}{\ell/2+x}$ represents a normal distribution of x, and therefore this is a *lognormal distribution of* d. This is multiplied by $\exp(-d\gamma_d^2 \ln^2 \tau/2)$. We can see that $\gamma_d \in [0, 1/2]$. When γ_d is very close to 0, then the exponential term is almost 1. Hence the product represents a lognormal tail. On the other hand, when γ_d is a constant (say > 0.2), then the product becomes an exponential tail. Observe that γ_d oscillates between 0 and $1/2$, leading to the characteristic behavior of SKG. As θ_d becomes closer to an integer, there are more vertices of degree d. As it starts to have a larger fraction part, the number of such vertices plummets exponentially. Note that there are many values of d (a constant fraction) where $\gamma_d > 0.2$. Hence, *for all these* d, the degrees are bounded above by an exponential tail. As a result, the degree distribution cannot be a power law or a lognormal.

The estimates provided by Thm. 1 for our three different SKG parameter sets are shown in Fig. 2. Note how this simple estimate matches the oscillations of the actual degree distribution very accurately. We provide a slightly more complex expression in Lem. 3 that completely captures the degree distribution.

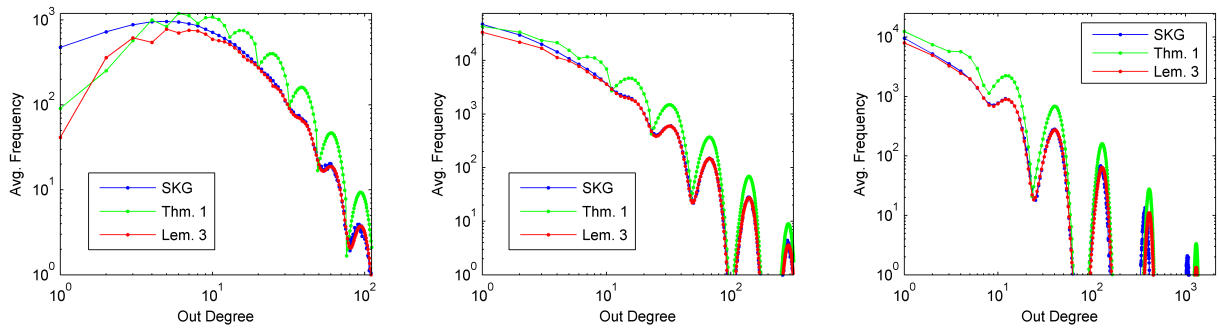
The details of proving Thm. 1 are provided in the full version [7]. We just state some of the main lemmas and give an intuitive explanation in the next subsection. The following lemma bounds the probability that a vertex v at slice r has degree d. This lemma needs the technical assumption that $d \le \sqrt{n}$. Hence, our formula becomes slightly inaccurate when d becomes large, but as our figures show, it is not a major issue. This technical condition can be removed, at the cost of making the expressions more messy.

Lemma 2: Let v be a vertex in slice r and suppose that $d \le \sqrt{n}$. Then the probability that v has (out-)degree d is

$$(1 \pm o(1)) \frac{\lambda^d}{d!} \frac{(\tau^r)^d}{\exp(\lambda\tau^r)}.$$

The following lemma is the main technical result. Thm. 1 is a direct corollary of this lemma. Let X_d be the random variable for the number of vertices of (out-)degree d. In the following, the expectation is over the random choice of the graph. Observe that the following bound is a tight estimate.

Lemma 3: Let $\theta_d = \ln(d/\lambda)/\ln\tau$, $r_d = \lfloor \theta_d \rfloor$, and $\delta_d = \theta_d - r_d$. Let $(e\ln 2)\ell \le d \le \sqrt{n}$. If $r_d \ge \ell/2$, $\mathbf{E}[X_d]$ is



(a) CAHepPh (b) WEBNotreDame (with $\ell = 18$) (c) Graph500 (with $\ell = 16$)

Figure 2: We plot the degree distribution of graphs generated using our three different SKG parameter sets. We then plot the respective bounds predicted by Thm. 1 and Lem. 3. Observe how Thm. 1 correctly guesses the peaks and troughs of the degree distribution. Lem. 3 is practically an exact match (except when the degree is below 2ℓ or, in Graph500, slight inaccuracies when the degree is too large).

negligible. Otherwise, we have

$$\mathbf{E}[X_d] = \frac{1 \pm o(1)}{\sqrt{2\pi d}} \exp\left(\frac{-d\delta_d^2 \ln^2 \tau}{2}\right) \binom{\ell}{\ell/2 + r_d} + \frac{1 \pm o(1)}{\sqrt{2\pi d}} \exp\left(\frac{-d(1-\delta_d)^2 \ln^2 \tau}{2}\right) \binom{\ell}{\ell/2 + r_d + 1}.$$

We plot the bound given by this lemma in Fig. 2. Note how it completely captures the behavior of the degree distribution (barring a slight inaccuracy for larger degrees of the Graph500 graph because we start exceeding the upper bound for d in Lem. 3).

B. Understanding the degree distribution

The following is a verbal explanation of our proof strategy and captures the essence of the math. It will be convenient to think of the parameters having some fixed simple values. Let $\lambda = 1$ and $\tau = e$. (This can be achieved with a reasonable choice of T, ℓ, Δ.) We begin by looking at the different slices of vertices. Vertices in a fixed r-slice have an identical behavior with respect to the degree distribution. Lem. 2 uses elementary probability arguments to argue that the probability that a vertex in slice r has (out-)degree d is roughly

$$\frac{\exp(dr - e^r)}{d!} \tag{1}$$

When $r \gg \ln d$, the numerator will be less than 1, and the overall probability is $\ll 1/d!$. Therefore, those slices will not have many (or any) vertices of degree d. If $r \ll \ln d$, the numerator is $o(d!)$ and the probability is still (approximately) at most $1/d!$. Observe that when r is negative, then this probability is extremely small, even for fairly small values of d. This shows that half of the vertices (in slices where the number of 1's is more than 0's) have extremely small degrees.

It appears that the "sweet spot" is around $\ln d$. Applying Taylor approximations to appropriate ranges of r, we can show that a suitable approximation of the probability of a slice r vertex having degree d is roughly $\exp(-d(r - \ln d)^2)$. We can now show that the SKG degree distribution is bounded *above* by a lognormal tail. Only the vertices in slice $r \approx \ln d$ have a good chance of having degree d. This means that the expected number of vertices of degree d is at most $\binom{\ell}{\ell/2 + \ln d}$. Since the latter is normally distributed as a function of $\ln d$, it (approximately) represents lognormal tail. A similar conclusion was drawn in [8], though their approach and presentation is very different from ours.

This is where we significantly diverge. The crucial observation is that r is a *discrete* variable, not a continuous one. When $|r - \ln d| \geq 1/3$ (say), the probability of having degree d is at most $\exp(-d/9)$. That is an exponential tail, so we can safely assume that vertices in those slices have no vertices of degree d. Refer to Fig. 3. Since $\ln d$ is not necessarily integral, it could be that for *all* values of r, $|r - \ln d| \geq 1/3$. In that case, there are (essentially) *no vertices of degree* d. For concreteness, suppose $\ln d = 100/3$. Then, regardless of the value of r, $|r - \ln d| \geq 1/3$. And we can immediately bound the fraction of vertices that have this degree by the exponential tail, $\exp(-d/9)$. When $\ln d$ is close to being integral, then for $r = \lfloor \ln d \rceil$, the r-slice (and only this slice) will contain many vertices of degree d. The quantity $|\ln d - \lfloor \ln d \rceil|$ fluctuates between 0 and $1/2$, leading to the oscillations in the degree distribution.

Let $\Gamma_d = \lfloor \ln d \rceil$ and $\gamma_d = |\Gamma_d - \ln d|$. Putting the arguments above together, we can get a very good estimate of the number of vertices of degree d. This quantity is essentially $\exp(-\gamma_d^2 d)\binom{\ell}{\ell/2 + \Gamma_d}$, as stated in Thm. 1. A more nuanced argument leads to the bound in Lem. 3.

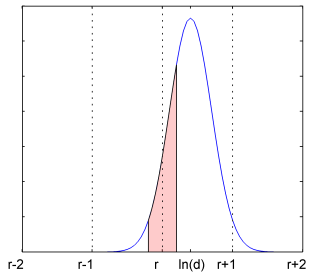
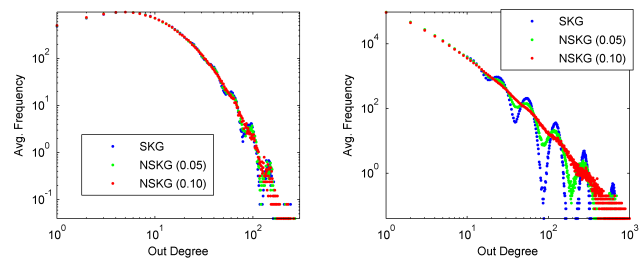


Figure 3: Probability of nodes of degree d for various slices. If $\ln d$ is far from integral, a vertex from slice r will have almost no vertices of degree d. Adding noise can be thought of as an average over the Gaussian. The probability of a slice r vertex having degree d is now the area of the shaded region.



(a) CAHepPh (b) WEBNotreDame

Figure 4: The figures show the degree distribution of standard SKG and NSKG as the averages of 25 instances. Notice how effectively a noise of 0.1 straightens the degree distribution.

C. Enhancing SKG with Noise

Let us now focus on a noisy version of SKG that removes the fluctuations in the degree distribution. The idea is quite simple. For each level $i \leq \ell$, define a new matrix T_i in such a way that the expectation of T_i is just T. We will assume that T is symmetric[4].

At level i in the edge insertion, we use the matrix T_i to choose the appropriate quadrant. Let b be a *noise parameter* chosen such that $b \leq \min((t_1+t_4)/2, t_2)$. For level i, choose μ_i to be a uniform random number in the range $[-b, +b]$. Formally, the matrix of probabilities is

$$P = T_1 \otimes T_2 \otimes \cdots \otimes T_\ell$$

where

$$T_i = \begin{bmatrix} t_1 - \frac{2\mu_i t_1}{t_1+t_4} & t_2 + \mu_i \\ t_3 + \mu_i & t_4 - \frac{2\mu_i t_4}{t_1+t_4} \end{bmatrix}.$$

Note that T_i is symmetric, its entries sum to 1, and all entries are positive. Each level involves only one random number μ_i, which changes all the entries of T in a linear fashion. Hence, we only need ℓ random numbers in total.

In Figures 1, 4a, and 4b, we show the effects of noise. Observe how even a noise parameter as small as 0.05 (which is extremely small compared to the matrix values) significantly reduces the magnitude of oscillations. A noise of 0.1 completely removes the oscillations, and we attain a true lognormal distribution (Thm. 4). (Even this is very small noise, since the standard deviation of this noise parameter is just 0.06.) This completely annihilates the undesirable exponential tail behavior of SKG, and leads to a truly monotone decrease.

D. Why does noise help?

We first state our formal theorem. Essentially, when the noise is large enough, then we can show that the degree distribution is at least a lognormal tail on average. This is a significant change from SKG, where many degrees are below an exponential tail. The full proof is given in [7] and we only give an intuitive sketch here. Nonetheless, the reader does not have to take our words on faith, since our figures provide clear evidence that the degree distribution of NSKG (with noise = 0.1) is lognormal.

Theorem 4: Choose noise b so that it satisfies

$$b > \sqrt{\frac{3}{\ell}} \left(\frac{(1+2\sigma)(t_1+t_4)\ln\tau}{16\sigma} \right)$$

Then the expected degree distribution for NSKG is bounded below by a lognormal. Formally, when $\Gamma_d \leq \ell/2$ and $(e \ln 2)\ell \leq d \leq \sqrt{n}$,

$$\mathbf{E}[X_d] \geq \frac{1}{e^2\sqrt{2\pi\ell d}} \binom{\ell}{\ell/2 + \Gamma_d}.$$

Observe that the condition for b is just $\Omega(1/\sqrt{\ell})$, since the large complicated part in parenthesis is just some constant that depends on T. For each of our case study SKG parameters, this quantity ranges from 0.08 to 0.11. In practice, noise less than this bound suffices. More importantly, this bound tells us that as ℓ increases, we need *less* noise to get a lognormal tail.

We now provide a verbal description of the main ideas. Let us assume that $\lambda = 1$ and $\tau = e$, as before. We focus attention on a vertex v of slice r, and wish to compute the probability that it has degree d. Note the two sources of randomness: one coming from the choice of the noisy SKG matrices, and the second from the actual graph generation. We associate a *bias parameter* ρ_v with every vertex v. This can be thought of as some measure of how far the degree behavior of v deviates from its noiseless version. Actually, it is the random variable $\ln \rho_v$ that we are interested in. It can be shown that $\ln \rho_v$ is distributed like a Gaussian. The *distribution* of ρ_v is identical for all vertices in slice r. (Though it does not matter for our purposes, for a given

[4]It can easily be generalized, but this is omitted due to space.

instantiation of the noisy SKG matrices, vertices in the same slice can have different biases.)

We approximate the probability that v has degree d by

$$\exp(dr + d\ln\rho_v - \rho_v e^r)/d!.$$

After some simplifying, this is roughly equal to $\exp(-d(r - \ln d - \ln\rho_v)^2)$. The additional $\ln\rho_v$ will act as a *smoothing* term. Observe that even if $\ln d$ has a large fractional part, we could still get vertices of degree d. Suppose $\ln d = 10.5$, but $\ln\rho_v$ happened to be close to 0.5. Then vertices in slice $\lfloor \ln d \rceil$ would have degree d with some reasonable probability. Contrast this with regular SKG, where there is almost no chance that degree d vertices exist.

Think of the probability as $\exp(d(r - \ln d - X)^2)$, where X is a random variable. The expected probability will be an average over the distribution of X. Intuitively, instead of the probability just being $\exp(d(r - \ln d)^2)$ (in the case of SKG), it is now the *average* value over some interval. If the standard deviation of X is sufficiently large, even though $\exp(d(r - \ln d)^2)$ is small, the average of $\exp(d(r - \ln d - X)^2)$ can be large. Refer to Fig. 3.

We know that X is a Gaussian random variable (with some standard deviation σ). So we can formally express the (expected) probability that v has degree d as an integral,

$$\int_{-\infty}^{+\infty} \exp(d(r - \ln d - X)^2) \cdot e^{-X^2/2\sigma^2} dX.$$

This definite integral can be evaluated exactly (since it is just a Gaussian). Intuitively, this is roughly the average value of $\exp(d(r - \ln d - X)^2)$, where X ranges from $-\sigma$ to $+\sigma$. Suppose $\sigma > 1$. Since r ranges over the integers, there is always *some* r such that $|r - \ln d| < 1$. For this value of r, the average of $\exp(d(r - \ln d - X)^2)$ over the range $X \in [-1, +1]$ will have a reasonably large value. This ensures that (in expectation) many vertices in this slice r have degree d. This can be shown for all degrees d, and we can prove that the degree distribution is at least lognormal.

E. Subtleties in adding noise

One might ask why we add noise in this particular fashion, and whether other ways of adding noise are equally effective. Since we only need ℓ random numbers, it seems intuitive that adding "more noise" could only help. For example, we might add noise on a per edge, basis, i.e., at each level i of *every* edge insertion, we choose a new random perturbation T_i of T. Interestingly, this version of noise does not smooth out the degree distribution, as shown in Fig. 5. In this figure, the red curve corresponds to the degree distribution of the graph generated by NSKG with Graph500 parameters, $\ell = 26$, and $b = 0.1$. The blue curve corresponds to generation by adding noise per edge. As seen in this figure, adding noise per edge has hardly any effect on the oscillations, while NSKG provides a smooth degree distribution curve. (These results are fairly consistent over different parameter choices.)

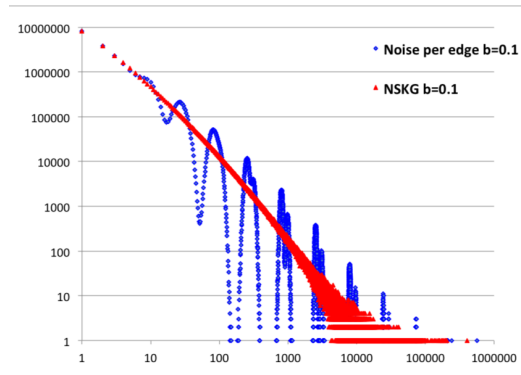


Figure 5: Comparison of degree distribution of graphs generated by NKSG and by adding noise per edge for Graph500 parameters and $\ell = 26$.

It is crucial that we use the *same* noisy $T_1, \ldots, T_\ell$ for every edge insertion.

IV. Isolated Vertices

In this section, we give a simple formula for the number of isolated vertices in SKG. This can be derived from elementary probability calculations described in [7]. We focus on the symmetric case[5], where $t_2 = t_3$ in the matrix T. We assume that ℓ is even in the following, but the formula can be extended for ℓ being odd. The real contribution here is not the methodology, but the final result, since it gives a clearer understanding of how many vertices SKG leaves isolated and how the SKG parameters affects this number. At the cost of a tiny error, the following gives a formula that is intuitive and easy enough to compute on a calculator.

Theorem 5: The number of isolated vertices can be approximated (within additive error $0.01n$) by

$$\sum_{r=-\ell/2}^{r=\ell/2} \binom{\ell}{\ell/2 + r} \exp(-2\lambda\tau^r) \tag{2}$$

The fraction of isolated vertices in a slice r is essentially $\exp(-\lambda\tau^r)$. Note that τ is larger than 1. Hence, this is a decreasing function of r. This is quite natural, since if a vertex v has many zeroes in its representation (higher slice), then it is likely to have a larger degree (and less likely to be isolated). This function is doubly exponential in r, and therefore decreases very quickly with r. The fraction of isolates rapidly goes to 0 (resp. 1) as r is positive (resp. negative).

Relation of SKG parameters to the number of isolated vertices: When λ decreases, the number of isolated vertices increases. Suppose we fix the SKG matrix and average degree Δ, and start increasing ℓ. Note that this is done in the Graph500 benchmark, to construct larger and larger graphs. The value of λ decreases exponentially in ℓ, so the number

[5] Our formula can be extended to the general case but is less elegant.

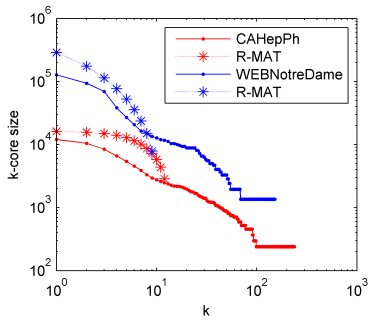


Figure 6: Core decompositions of real graphs and their SKG model. Observe that the max core of SKG is an order of magnitude smaller.

of isolated vertices will increase. Our formula suggests ways of counteracting this problem. The value of Δ could be increased, or the value σ could be decreased. But, in general, this will be a problem for generating large sparse graphs using a fixed SKG matrix.

When σ increases, then λ decreases and τ increases. Nonetheless, the effect of λ is much stronger than that of τ. Hence, the number of isolated vertices will increase as σ increases.

In Tab. I, we compute the estimated number of isolated vertices in graphs for the Graph500 parameters. Observe how the fraction of isolated vertices consistently increases as ℓ is increased. For the largest setting of $k = 42$, only one fourth of the vertices are not isolated.

V. k-CORES IN SKG

Structures of k-cores are a very important part of social network analysis [19], [16], [15], as they are a manifestation of the community structure and high connectivity of these graphs.

Definition 6: Given an undirected graph $G = (V, E)$, the *subgraph induced by set* $S \subseteq V$, is denoted by $G|_S := (S, E')$, where E' contains every edge of E that is completely contained in S. For an undirected graph, the *k-core* of G the largest induced subgraph of minimum degree k. The *max core number* of G is the largest k such that G contains a (non-empty) k-core. (These can be extended to directed versions: a k-out-core is a subgraph with min out-degree k.)

A bipartite core is an induced subgraph with every vertex has *either* a high in-degree or out-degree. The former are called *authorities* and the latter are *hubs*. Large bipartite cores are present in web graphs and are an important structural component [26], [27]. Note that if we make the a directed graph undirected (by simply removing the directions), then a bipartite core becomes a normal core. Hence, it is useful to compute cores in a directed graph by making it undirected.

We begin by comparing the sizes of k-cores in real graphs, and their models using SKG [3]. Refer to Fig. 6. We plot the size of the maximum k-core with k. The k at which the curve ends is the max core number. (For CAHepPh, we look at undirected cores, since this is an undirected graph. For WEBNotreDame, a directed graph, we look at out-cores. But the empirical observations we make holds for all other core versions.) For both our examples, we see how drastically different the curves are. By far the most important difference is that the curve for the SKG versions are extremely short. This means that the max core number is *much smaller* for SKG modeled graphs compared to their real counterparts. For the web graph WEBNotreDame, we see the presence of large cores, probably an indication of some community structure. The maximum core number of the SKG version is an *order of magnitude* smaller. Minor modifications (like increasing degree, or slight variation of parameters) to these graphs do not increase the core sizes or max cores numbers much. This is a problem, since this is strongly suggesting that SKG do not exhibit localized density like real web graphs or social networks.

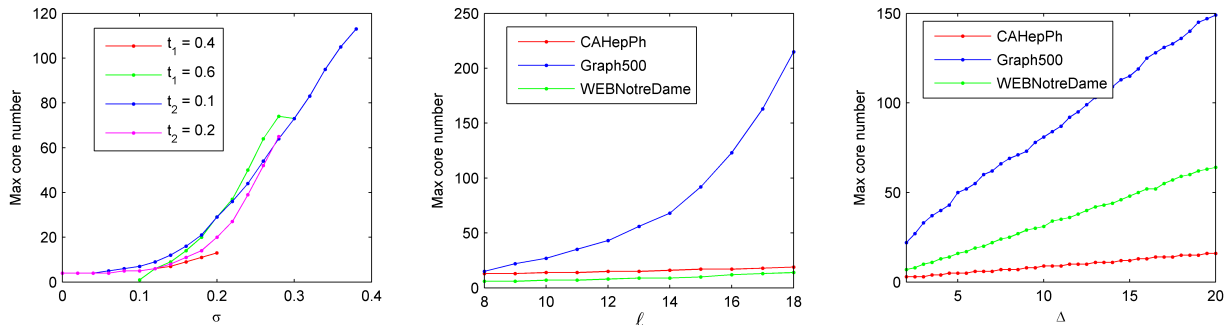
If we wish to use SKG to model real networks, then it is imperative to understand the behavior of max core numbers for SKG. Indeed, in Tab. II, we see that our observation is not just an artifact of our examples. SKGs consistently have very low max core number. Only for the peer-to-peer Gnutella graphs does SKG match the real data, and this is specifically for the case where the max core number is extremely small. For the undirected graph (the first three coauthorship networks), we have computed the undirected cores. The corresponding SKG is generated by copying the upper triangular part in the lower half to get a symmetric matrix (an undirected graph). The remaining graphs are directed, and we simply remove the direction on the edges and compute the total core. Our observations hold for in and out cores as well (given in full version), and a wide range of data. This is an indication that SKG is not generating dense enough subgraphs.

Table II: Core sizes in real graphs and SKG version

Graph	Real max core	SKG max core
CAGrQc	43	4
CAHepPh	238	16
CAHepTh	31	5
CITHepPh	30	19
CITHepTh	37	19
P2PGnutella25	5	5
P2PGnutella30	7	6
SOCEpinions	67	43
WEBNotreDame	155	31

We focus our attention on the max core number of SKG. How does this number change with the various parameters? The following summarizes our observations.

Empirical Observation 7: We focus on the case of sym-



(a) Varying σ (b) Varying ℓ (c) Varying Δ

Figure 7: We plot the max core number against various parameters. In the first picture, we plot the max core number of an (symmetric) SKG graph with increasing σ. Next, we show how the max core number increases with ℓ, the number of levels. Observe the major role that the matrix σ plays. For Graph500, σ is much larger than the other parameter sets. Finally, we show that regardless of the parameters, the max core number increases linearly with Δ.

metric T. As before, $\sigma = (t_1 + t_2) - 1/2$.

1) The max core number increases with σ. By and large, if $\sigma < 0.1$, max core numbers are extremely tiny.

2) Max core numbers grow with ℓ only when the values of σ are sufficiently large. Even then, the growth is much slower than the size of the graph. For smaller σ, max core numbers exhibit essentially negligible growth.

3) Max core numbers increase mostly linearly with Δ.

Large max core numbers require larger values of σ. As mentioned in §IV, increasing σ increases the number of isolated vertices. Hence, there is an inherent tension between increasing the max core number and decreasing the number of isolated vertices.

For the sake of consistency, we performed the following experiments on the max core after taking a symmetric version of the SKG graph. Our results look the same for in and out cores as well. In Fig. 7a, we show how increasing σ increases the max core number. We fix the values of $\ell = 16$ and $m = 6 \times 2^{16}$. (There is nothing special about these values. Indeed the results are basically identical, regardless of this choice.) Then, we fix t_1 (or t_2) to some value, and slowly increase σ by increasing t_2 (resp. t_1). We see that regardless of the fixed values of t_1 (or t_2), the max core consistently increases. But as long as $\sigma < 0.1$, max core numbers remain almost the same.

In Fig. 7b, we fix matrix T and average degree Δ, and only vary ℓ. For WEBNotreDame[6], we have $\sigma = 0.18$ and for CA-HEP-Ph, we have $\sigma = 0.11$. For both cases, increasing ℓ barely increases the max core number. Despite increasing the graph size by 8 orders of magnitude, the max core number only doubles. Contrast this with the Graph500 setting, where $\sigma = 0.26$, and we see a steady increase with larger ℓ. This is a predictable pattern we notice for many different parameter settings: larger σ leads to larger max core numbers as ℓ goes up. Finally, in Fig. 7c, we see that the max core number is basically linear in Δ.

[6]Even though the matrix T is not symmetric, we can still define σ. Also, the off diagonal values are 0.20 and 0.21, so they are almost equal.

VI. Conclusions

For a true understanding of a model, a careful theoretical and empirical study of its properties in relation to its parameters is imperative. This not only provides insights into why certain properties arise, but also suggests ways for enhancement. One strength of the SKG model is its amenability to rigorous analysis, which this paper exploited.

We prove strong theorems about the degree distribution, and more significantly show how adding noise can give a true lognormal distribution by eliminating the oscillations in degree distributions. Our proposed method of adding noise requires only ℓ random numbers all together, and is hence cost effective. We want to stress that our major contribution is in providing *both* the theory and matching empirical evidence. The formula for expected number of isolated vertices provides an efficient alternative to methods for computing the full degree distribution. Besides requiring fewer operations to compute and being less prone to numerical errors, the formula transparently relates the expected number of isolated vertices to the SKG parameters. Our studies on core numbers establish a connection between the model parameters and the cores of the resulting graphs. In particular, we show that commonly used SKG parameters generate tiny cores, and the model's ability to generate large cores is limited.

Acknowledgments

This work was funded by the applied mathematics program at the United States Department of Energy and performed at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United

States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

We are grateful to David Gleich for the MATLAB BGL library as well as many helpful discussions. We thank Todd Plantenga for running many SKG experiments, and especially for generating Fig. 5. We thank Jon Berry for checking our Graph500 predictions against real data, and also David Bader and Richard Murphy for discussions about the Graph500 benchmark. We also acknowledge Jennifer Neville and Blair Sullivan, who inspired us with their different work on SKGs during recent visits to Sandia.

References

[1] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Computing Surveys*, vol. 38, no. 1, 2006.

[2] J. Leskovec and C. Faloutsos, "Scalable modeling of real graphs using kronecker multiplication," in *ICML '07*. ACM, 2007, pp. 497–504.

[3] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Machine Learning Research*, vol. 11, pp. 985–1042, Feb. 2010. [Online]. Available: http://jmlr.csail.mit.edu/papers/v11/leskovec10a.html

[4] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A recursive model for graph mining," in *SDM '04*, 2004, pp. 442–446. [Online]. Available: http://siam.org/proceedings/datamining/2004/dm04_043chakrabartid.pdf

[5] Graph 500 Steering Committee, "Graph 500 benchmark," 2010, available at http://www.graph500.org/Specifications.html.

[6] M. Mitzenmacher, "The future of power law research," *Internet Mathematics*, vol. 2, no. 4, pp. 525–534, 2006. [Online]. Available: http://www.internetmathematics.org/volumes/2/4/mitzenmacher.pdf

[7] C. Seshadhri, A. Pinar, and T. G. Kolda, "An in-depth analysis of stochastic Kronecker graphs," arXiv:1102.5046, September 2011. [Online]. Available: http://arxiv.org/abs/1102.5046

[8] M. Kim and J. Leskovec, "Multiplicative attribute graph model of real-world networks," Sep. 2010, arXiv:1009.3499v2. [Online]. Available: http://arxiv.org/abs/1009.3499v2

[9] C. Groër, B. D. Sullivan, and S. Poole, "A mathematical analysis of the R-MAT random graph generator," Jul. 2010, available at http://www.ornl.gov/~b7r/Papers/rmat.pdf.

[10] Z. Bi, C. Faloutsos, and F. Korn, "The "DGX" distribution for mining massive, skewed data," in *KDD '01*. ACM, 2001, pp. 17–26.

[11] D. Pennock, G. Flake, S. Lawrence, E. Glover, and C. L. Giles, "Winners don't take all: Characterizing the competition for links on the web," *PNAS*, vol. 99, no. 8, pp. 5207–5211, 2002.

[12] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet Mathematics*, vol. 1, no. 2, pp. 226–251, 2003. [Online]. Available: http://projecteuclid.org/euclid.im/1089229510

[13] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[14] J. Berry and R. Murphy, Members of Graph500 steering committee, Personal communication.

[15] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Link Mining: Models, Algorithms, and Applications*. Springer, 2010, pp. 337–357.

[16] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani, "K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases," *Networks and Heterogenous Media*, vol. 3, no. 2, pp. 371–393, 2008.

[17] C. Gkantsidis, M. Mihail, and E. W. Zegura, "Spectral analysis of internet topologies," in *INFOCOM 2003*. IEEE, 2003, pp. 364–374.

[18] A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes, "k-core (bootstrap) percolation on complex networks: Critical phenomena and nonlocal effects," *Phys. Rev. E*, vol. 73, no. 5, p. 056101, May 2006.

[19] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, "A model of internet topology using k-shell decomposition," *PNAS*, vol. 104, no. 27, pp. 11 150–11 154, 2007.

[20] R. Andersen and K. Chellapilla, "Finding dense subgraphs with size bounds," in *Algorithms and Models for the Web-Graph*. Springer, 2009, pp. 25–37.

[21] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, "Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication," in *PKDD 2005*. Springer, 2005, pp. 133–145.

[22] M. Mahdian and Y. Xu, "Stochastic Kronecker graphs," *Random Structures & Algorithms*, in press, conference version appeared as [28].

[23] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao, "Measurement-calibrated graph models for social network experiments," in *WWW '10*. ACM, 2010, pp. 861–870.

[24] B. Miller, N. Bliss, and P. Wolfe, "Subgraph detection using eigenvector L1 norms," in *NIPS 2010*, 2010, pp. 1633–1641. [Online]. Available: http://books.nips.cc/papers/files/nips23/NIPS2010_0954.pdf

[25] S. Moreno, S. Kirshner, J. Neville, and S. V. N. Vishwanathan, "Tied Kronecker product graph models to capture variance in network populations," in *Proc. 48th Annual Allerton Conf. on Communication, Control, and Computing*, Oct. 2010, pp. 1137–1144.

[26] D. Gibson, J. Kleinberg, and P. Raghavan, "Inferring web communities from link topology," in *HYPERTEXT '98*. ACM, 1998, pp. 225–234.

[27] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[28] M. Mahdian and Y. Xu, "Stochastic kronecker graphs," in *Algorithms and Models for the Web-Graph*. Springer, 2007, pp. 179–186.