PLOS ONE

RESEARCH ARTICLE

Fast Generation of Sparse Random Kernel Graphs

Aric Hagberg[1☯], **Nathan Lemons**[2☯*]

1 Center for Nonlinear Studies, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico, United States of America, **2** Applied Mathematics and Plasma Physics, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico, United States of America

☯ These authors contributed equally to this work.
* nlemons@lanl.gov

Citation: Hagberg A, Lemons N (2015) Fast Generation of Sparse Random Kernel Graphs. PLoS ONE 10(9): e0135177. doi:10.1371/journal.pone.0135177

Editor: Wen-Bo Du, Beihang University, CHINA

Received: May 11, 2015

Accepted: July 17, 2015

Published: September 10, 2015

Data Availability Statement: The Python software for the algorithm in the manuscript is available at https://github.com/networkx/networkx/pull/1509.

Funding: This work was funded by the Department of Energy at Los Alamos National Laboratory under contract DE-AC52-06NA25396 through the Laboratory Directed Research and Development Program.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

The development of kernel-based inhomogeneous random graphs has provided models that are flexible enough to capture many observed characteristics of real networks, and that are also mathematically tractable. We specify a class of inhomogeneous random graph models, called random kernel graphs, that produces sparse graphs with tunable graph properties, and we develop an efficient generation algorithm to sample random instances from this model. As real-world networks are usually large, it is essential that the run-time of generation algorithms scales better than quadratically in the number of vertices n. We show that for many practical kernels our algorithm runs in time at most $\mathcal{O}(n(\log n)^2)$. As a practical example we show how to generate samples of power-law degree distribution graphs with tunable assortativity.

1 Introduction

The broad adoption of graphs as a modeling language, together with the widespread importance of applications in social, computer, and biological systems [1], has resulted in many efforts to develop random graph models [2, 3]. Random graph models give insight into network structures and are often used for null models, anonymization, and studying dynamical processes [4–7]. Large-scale graphs are also used to construct benchmarks for testing algorithm performance on high-performance computing systems [8, 9].

A common goal in constructing random graph models is to match properties of real-world graphs. One approach is to explicitly specify a distribution of graphs with expected graph properties that can be analyzed. Important examples of this approach include Erdős-Rényi random graphs [10, 11], Chung-Lu (also called expected degree) random graphs [12], and graphs with a specified degree distribution [13]. To capture even more general graph features Söderberg introduced a model of sparse inhomogeneous random graphs and showed that it could produce a wide variety of sparse graphs [14]. Bollobás, Janson, and Riordan (BJR) formalized and extended the model of Söderberg by emphasizing that the random graphs could be defined in terms of a kernel [15]. They also focused the model on sparse graphs with $O(n)$ edges and n

vertices. The flexibility inherent in the kernel approach generalizes well-known models of sparse graphs while remaining mathematically tractable; the BJR model can produce graphs with power-law degree distributions [15], and graphs with tunable assortativity [16].

Models from which random uniform samples can be efficiently generated are even more useful. In particular, the efficient generation of random graph instances allows researchers to simulate complex graph phenomena and dynamics for which mathematical analysis is difficult or impossible [17]. There are models, such as the space of all graphs with a given degree sequence and clustering coefficients, from which we do not know how to sample uniformly. Simulation of such models is then confined to the regions of the distribution currently available to us.

Though the BJR model appears to be so general as to preclude an efficient, general generation algorithm, we provide a fast generation algorithm for an important class of kernels, which we call Random Kernel Graphs. Random Kernel Graphs are very general and exhibit the flexibility of the inhomogeneous random graph model including tunable expected degree sequences and tunable assortativity. For Random Kernel Graphs, we exploit the idea of sampling from a waiting-time distribution to design an algorithm for generating uniform n-node samples with complexity of $\mathcal{O}(n(\log n)^2)$. We demonstrate the utility of the model by showing how to generate large sparse random graphs with a power-law degree distribution and adjustable assortativity.

2 Random Kernel Graphs

The BJR random graph model is extremely general, and we do not know of an algorithm which can quickly and efficiently generate such graphs. Instead, we specify an important special case of the model, the Random Kernel Graph $G(n, \kappa)$ defined below, which is still very general and includes many models such as the Erdős-Rényi $G(n, p)$ [11], Chung-Lu $G(w)$ [12], and Söderberg models [14].

Definition 1 (The Kernel κ). A non-negative, bounded, symmetric, measurable function $\kappa: [0, 1]^2 \to \mathbb{R}$ is a *kernel* if there exists a finite set $D \subset [0, 1]$ such that κ is continuous at all points (x, y) for which neither x nor y belong to D.

Definition 2 (Random Kernel Graph $G(n, \kappa)$). For each positive integer n we define a distribution of graphs on $\mathbf{v}_n = \{i/n : i = 1, 2, \ldots, n\}$. Given a kernel κ, we define the *Random Kernel Graph* $G(n, \kappa)$ to be the graph obtained on the vertices $\mathbf{v}_n$ when edges (v_i, v_j) are chosen independently with probability p_{ij} given by

$$p_{ij} := \frac{\kappa(v_i, v_j)}{n}.$$

As κ is bounded and we are interested in the asymptotics of generating large graphs, we always assume that $\kappa \leq n$.

For example if $\kappa \equiv c$ is constant, then $G(n, \kappa)$ is nothing more than the Erdős-Rényi random graph model in which each edge appears independently with probability c/n.

Note that $G(n, \kappa)$ defines a sequence of graph distributions, one distribution for each integer n. This is convenient from several perspectives. Mathematically we can analyze the $n \to \infty$ limit of the distributions; from a practical standpoint we can generate graphs at different scales which all come from the same model.

3 An efficient generation algorithm

When random graphs are defined through independent random variables, as is the case in for $G(n, \kappa)$, one need only test $\binom{n}{2}$ random Bernoulli variables to choose a graph on n vertices uniformly from the distribution. But when modeling real-world networks, which are usually large and sparse, algorithms which take $\mathcal{O}(n^2)$ steps are prohibitively slow. To produce a sparse

graph with m edges the ideal is to find algorithms that run in time $\mathcal{O}(m)$. Batagelj and Brandes found such an algorithm for producing Erdős-Rényi random graphs [18]. Instead of sampling consecutive Bernoulli random variables, the algorithm samples from a waiting time distribution (the Geometric distribution) to determine the next edge to be added. This method was extended to generate Chung-Lu random graphs in time $\mathcal{O}(m+n)$ [19].

As in the methods discussed above, our design of an efficient algorithm for $G(n, \kappa)$ begins by drawing from waiting-time distributions instead of drawing n^2 Bernoulli variables. The random variable from our waiting distribution tells us exactly how many "non-edges" are skipped before the next edge is added to the graph. Suppose that in generating a graph $G = G(n, \kappa)$ we have already determined that vertex v_i is adjacent to vertex v_j. We would like to determine the next neighbor of v_i in the ordering of the indices. (By symmetry it is sufficient to determine only those neighbors of v_i whose index is greater than i so we can assume that $j > i$.) We first pick a random number r from the uniform distribution on (0,1], and then set d to the smallest positive integer such that,

$$\prod_{k=j+1}^{j+d} 1 - p_{ik} < r\,. \tag{1}$$

The next neighbor of v_i is then v_{j+d} so the edge (v_i, v_{j+d}) is added to the graph. If there is no such d with $j+d \leq n$ then v_i has no more neighbors and we continue by searching for neighbors of the next vertex v_{i+1}.

The key to a fast algorithm for $G(n, \kappa)$ is efficiently calculating the index d for each generated edge. To see how to approach this problem consider the following approximations,

$$\prod_{k=j+1}^{j+d} 1 - p_{ik} = \prod_{k=j+1}^{j+d} 1 - \frac{\kappa(v_i, v_k)}{n} \tag{2}$$

$$\sim \exp\left(-\sum_{k=j+1}^{j+d} \frac{\kappa(v_i, v_j)}{n}\right) \tag{3}$$

$$\sim \exp\left(-\int_{j/n}^{(j+d)/n} \kappa(v_i, y)\, dy\right). \tag{4}$$

The product in Eq (2) has been reduced to calculating the exponential of a definite integral. This formula can be computed efficiently, especially if an analytical form for the integral of κ can be found.

Instead of using the approximation Eq (4) to compute the waiting times we take a different approach and define a new random kernel model G' based on this approximation. This new model can be generated exactly. We prove in Appendix A that the models G and G' are asymptotically equivalent.

Definition 3 (Random kernel graph $G'(n, \kappa)$). Let $\mathbf{v}_n$ and κ be given as in Definition 2. The random kernel graph $G'(n, \kappa)$ is the graph obtained on the vertices $\mathbf{v}_n$ when the edges (v_i, v_j) are chosen independently with probability p'_{ij} given by

$$p'_{ij} := 1 - \exp\left(-\int_{v_{j-1}}^{v_j} \kappa(v_i, y)\, dy\right).$$

We set the value of $v_0 = 0$ to allow computation of the integral but no vertex v_0 is added to the graph.

For the model $G'(n, \kappa)$ we now have for relation Eq (1) the following equations,

$$\begin{aligned}\prod_{k=j+1}^{j+d}(1-p'_{ik}) &= \prod_{k=j+1}^{j+d}\exp\left(-\int_{v_{k-1}}^{v_k}\kappa(v_i,y)\,dy\right)\\ &= \exp\left(-\sum_{k=j+1}^{j+d}\int_{v_{k-1}}^{v_k}\kappa(v_i,y)\,dy\right)\\ &= \exp\left(-\int_{j/n}^{(j+d)/n}\kappa(v_i,y)\,dy\right).\end{aligned}$$

Therefore, if r is sampled uniformly from (0,1], then we need to find the minimal d such that

$$\int_{j/n}^{(j+d)/n}\kappa(v_i,y)\,dy > -\ln r\,. \tag{5}$$

Using inequality Eq (5) we present an efficient method for generating the model $G'(n, \kappa)$ in Algorithm 1. To simplify the exposition we use the following notation,

$$F(v_i, v_j, v_k) := \int_{v_j}^{v_k}\kappa(v_i,y)\,dy\,. \tag{6}$$

Algorithm 1 Fast Generation of $G'(n, \kappa)$
Input: : n, F
Output: : $G'(V, E)$
1: $V \leftarrow \{v_i = i/n : i = 1, 2, \ldots, n\}$
2: $E \leftarrow \emptyset$
3: $(v_i, v_j) \leftarrow (1/n, 1/n)$
4: **while** $v_i < 1$ **do**
5: Sample r uniformly from (0,1]
6: $r \leftarrow -\ln r$
7: **If** $F(v_i, v_j, v_n) \leq r$ **then**
8: $(v_i, v_j) \leftarrow (v_i + \frac{1}{n}, v_i + \frac{1}{n})$
9: **else**
10: Set d to smallest positive integer with $F(v_i, v_j, v_j+d/n) > r$
11: $E \leftarrow E \cup (v_i, v_k)$
12: $(v_i, v_j) \leftarrow (v_i, v_k)$
13: **end if**
14: **end while**

4 Algorithm scaling performance

If lines 6, 7, and 10 in Algorithm 1 can be calculated in $\mathcal{O}(1)$ time then the entire algorithm runs in time $\mathcal{O}(n)$. To see this, note that each time the *while* loop of the algorithm executes, either an edge is added to the graph or the index of v_i is increased by 1. Since the index of v_i never decreases, the *if* statement can only execute at most n times in total. Thus the *while* loop executes at most $m+n$ times. For graphs with bounded κ, which are the focus of this manuscript, $m = \mathcal{O}(n)$ and thus the while loop executes $\mathcal{O}(n)$ times. However, line 6 requires the evaluation of a logarithm, line 7 requires the evaluation of a definite integral, and line 10 requires a root-finding algorithm. In general these operations are not $\mathcal{O}(1)$ running time; the

speed of the integral and root-finding algorithms depend on κ. If however, F and its roots can be calculated in time $\mathcal{O}(1)$, then the entire algorithm runs in time $\mathcal{O}(n)$.

If numerical integration or root-finding is required, the complexity will be slightly worse. There are numerical integration algorithms which can calculate large classes of definite integrals in order $\mathcal{O}(\log n)$ (the implied constant will depend on κ) with an error bound of n^{-k} for fixed k. Root finding is also often fairly inexpensive. For large classes of functions, a root can be found in time $\mathcal{O}(\log n)$ (again the implied constant will depend on κ) with an error of at most $\mathcal{O}(n^{-k})$. For example if κ is a polynomial, it can be integrated analytically and Newton's method can be used to find its roots in $\mathcal{O}(\sqrt{\log n})$ time to a precision of n^{-k} for any fixed k.

We tested the scaling of Algorithm 1 by generating Erdős-Rényi random graphs with the parameter $p = 10/n$ ($\kappa \equiv 10$) at various scales for $n \leq 10^8$. While it is trivial to analytically solve for the integrals and roots with constant κ, we used a numerical root solver to demonstrate that even with root solving the algorithm works efficiently. Fig 1 shows the results of our timing experiments for a Python implementation of Algorithm 1 using the NetworkX software package [20]. The data show that with numerical root finding the algorithm scales at a better (faster) rate than the worst case $\mathcal{O}(n \log n)$.

Finally we note that Algorithm 1 can be trivially parallelized since the computation of the neighbors for each of the n vertices v_i can be done independently.

5 Generating graphs with tunable assortativity

We now provide examples for random kernel graphs with tunable assortativity, or mixing coefficients. This provides a way to generate graphs uniformly from a distribution with specified, and analytically computable, asymptotic assortativity.

5.1 Assortativity

The assortativity coefficient of a graph G, denoted $\rho(G)$, is defined in the following way. Pick an edge (u, v) uniformly over all the edges in G. Define random variables D_u, D_v to be the degrees of u and v respectively. Note that D_u and D_v by symmetry have the same distribution. Then $\rho(G)$ is given by

$$\rho(G) := \frac{\mathrm{Cov}(D_u, D_v)}{\mathrm{Var}(D_v)}. \tag{7}$$

The asymptotic assortativity of $G(n, \kappa)$ can be computed directly using the kernel κ [15] (see Appendix B). The formula, found in Eq (15), contains terms related to the number of copies of small subgraphs in the graph. We can use this formula to design a kernel κ with a specific asymptotic assortativity. In the following we give an example and compare the numerically computed assortativity from Eq (7) with the asymptotic value from Eq (15).

5.2 Power-law graphs with assortativity

To demonstrate the flexibility of the Random Kernel Graph model generator we now show how to produce graphs with a cutoff power law degree sequence and with tunable assortativity. We choose a cutoff power-law degree sequence as representative of the power-law like degree sequences which are ubiquitous in real networks [21]. For a similar reason we chose the Pearson correlation coefficient, or assortativity, as a second tunable parameter; the assortativity is widely used and its value is known for most real networks of interest [22]. The cutoff is designed to bound the maximum (expected) degree.

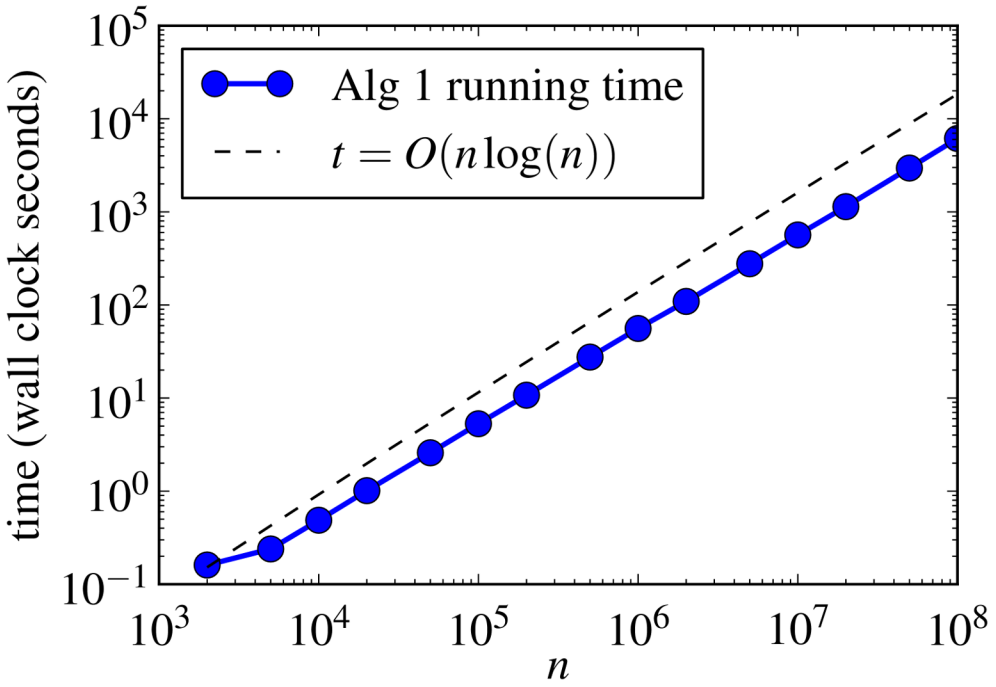

Fig 1. The running times for generating Erdős-Rényi $G(n, p)$ random graphs $\kappa \equiv 10$ using the method in Algorithm 1. The blue circles show the average wall-clock run-time for graphs at a given n. The dashed reference line $t = 10^{-5}\, n\log n$ is provided to show that the average run-time performance is slightly better than the worst case estimate $O(n\log n)$.

doi:10.1371/journal.pone.0135177.g001

To generate a graph with an expected degree sequence given by a function ψ we can use the kernel $\kappa(x, y) = c\psi(x)\psi(y)$ with $c = 1/\int_{[0,\,1]}\psi(x)dx$ (see Appendix B). To produce graphs with a cutoff power-law degree sequence we used the kernel

$$\kappa(x, y) = (x + .0001)^{-1/2}(y + .0001)^{-1/2}. \tag{8}$$

This produces graphs such that the number of vertices of degree k is proportional to k^{-3} up to the cutoff which occurs at $k = 100$ ($k = 100$ is the maximum expected degree). Note that in general, the kernel

$$\nu(x, y) = x^{-1/p}y^{-1/p},$$

for $p > 0$ will produce graphs with power-law like degree sequences. To see this, recall that the degree sequence will (in the asymptotic limit) follow the mixed Poisson distribution $\int_0^1 \lambda(x)dx$, where $\lambda(x) = \int_0^1 \nu(x, y)dy$. For large fixed k, we can approximate the number of vertices of degree at least k as follows. The measure of the set

$$\mu(\{x : \lambda(x) > k\}) = k^{-p}.$$

Thus by concentration of measure, the probability a vertex will have degree at least k will also scale as k^{-p}. Indeed it is not hard to check that if d_k is the fraction of vertices with degree k, then

$$d_k \sim ck^{-p-1},$$

for an appropriate constant $c > 0$ [16], Section 8.1].

The graphs produced by the kernel in Eq (8) will have expected assortativity $\rho = 0$; there are no degree-degree correlations. To add degree-degree correlations, we modify the kernel to

$$\kappa'(x,y) = (x + .0001)^{-1/2}(y + .0001)^{-1/2} + am_c(x,y), \tag{9}$$

where $m_c(x, y)$ is defined as

$$m_c(x,y) = \begin{cases} 1 & \text{if } x \le c \text{ and } y \le c \\ \dfrac{c^2}{(1-c)^2} & \text{if } x > c \text{ and } y > c \\ -\dfrac{c}{1-c} & \text{otherwise.} \end{cases} \tag{10}$$

Here, c is chosen in the interval (0,1] while a can be any nonnegative number such that κ' is nonnegative. Note that for any such choice of c and a the expected degree sequence of κ' is the same as that of κ since

$$\forall x,\ \forall y,\quad \int_0^1 am_c(x,y)dx = 0.$$

Thus the term $am_c(x, y)$ changes the structure of the graphs $G(n, \kappa')$ without modifying their (expected) degree sequences. In particular, as κ is monotone decreasing, by modifying the parameters a and c we can produce graphs with varying degree correlations.

For our experiments, we chose $c = 0.001$ and then maximized or minimized a to produce maximal positive or negative assortativity. We used the value $a = -909$ to produce graphs with negative assortativity and $a = 30{,}119$ to produce graphs with positive assortativity. For these two kernels (defined by the positive and negative values of a), we directly calculated the expected asymptotic assortativity coefficient using the formula given in Section B. In the positive assortativity case, we calculated the asymptotic assortativity coefficient to be $\rho = 0.1876$, while in the negative assortativity case we calculated a value of $\rho = -0.0056$. The asymptotic values were then compared to averages obtained by generating multiple graphs $G'(n, \kappa')$ at varying scales and numerically calculating the associated assortativity coefficients. We have no theoretical results on the rate convergence to the asymptotic value as a function of number of vertices n. But the results in our experiment, shown in Fig 2, demonstrate that in this case the convergence is fast and by $n = 10^6$ vertices has reached nearly the asymptotic value.

Despite its widespread use, the Pearson correlation coefficient is an imperfect graph statistic. Litvak and van Hofstead have shown that graphs with power-law degree sequences can only achieve non-negative assortativity in the asymptotic limit [23]. We found that even with a power-law degree sequence with a cutoff, it was hard to adjust the assortativity coefficient much below zero. Our methods however, are not necessarily exhaustive; there may indeed be graphs with cutoff power-law degree sequences that also have strongly negative assortativity coefficients.

6 Discussion

There have been many approaches to modeling random graphs with given properties. For example graphs with a fixed degree sequence have many applications both in pure and applied mathematics. But the search for unbiased generators of these graphs has proved quite difficult, and the general problem of efficiently generating graphs with a specified nonuniform degree sequence is still open. Even when polynomial algorithms are known to exist they can be impractical for large n (see [24] for a short survey). There are two main approaches to generating random graphs with a given degree sequence. The first approach, called the configuration

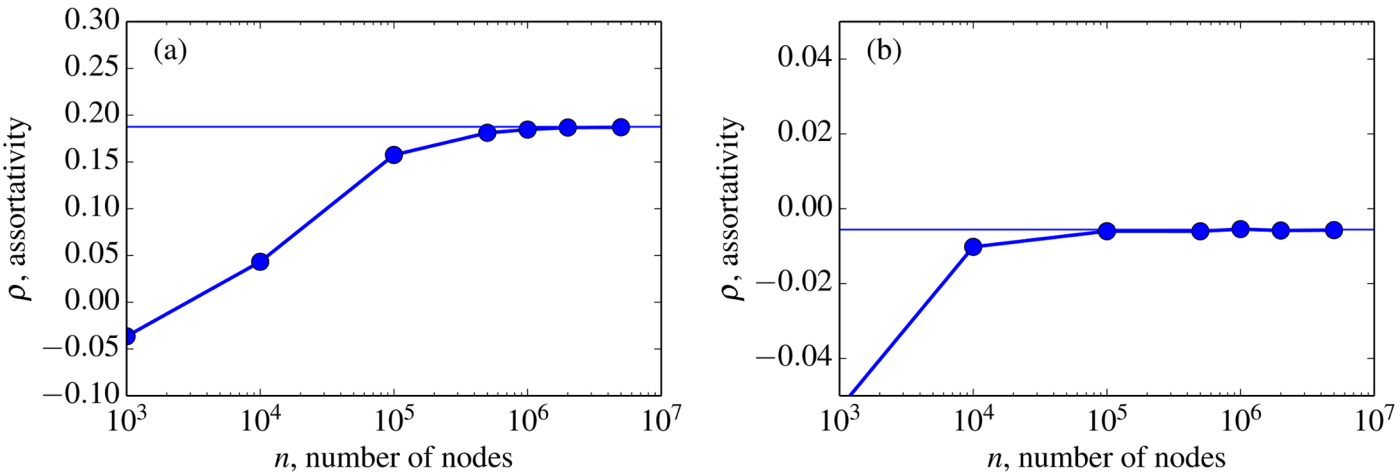

Fig 2. The average assortativity coefficient of graphs $G'(n, \kappa')$ generated from the kernel in Eq (8) for varying n. For each data point, shown by the solid circles, an ensemble of 10 graphs were generated and the average assortativity coefficient was computed for the ensemble. (a) Positive assortativity, $c = 0.001$, $a = 30{,}119$. (b) Negative assortativity, $c = 0.001$, $a = -909$. The horizontal line is the asymptotic calculation of the assortativity coefficient. The values converge to approximately the asymptotic value by $n = 10^6$.

doi:10.1371/journal.pone.0135177.g002

method, was pioneered by Bender and Canfeld [25] and Bollobás [13]. Here, "stubs" on each of the vertices are matched in pairs to form the edges of the graph. This basic idea was used to define an algorithm to produce uniformly graphs with a given degree sequence in time $\mathcal{O}(dm)$ where d, the max degree in the graph, is restricted by $d = \mathcal{O}(m^{1/4-\epsilon})$ [24, 26]. A second approach is to use a double edge-swap operation to define a Markov chain on the space of graphs with a given degree distribution [27]. Unfortunately, it is notoriously difficult to show that these Markov chains have fast mixing. In practice various heuristics are applied to determine when to stop swapping edges [28].

Finally, we note that the model $G(n, \kappa)$ will produce very few triangles. Asymptotically, the number of subgraphs $K_3 = 0$. For some applications, such as in social networks, real-world networks have many triangles. Models that can match the triangle density or even triangles correlated with the graph degree are important [29]. It is possible to extend the $G(n, \kappa)$ model and algorithm to use kernels that will produce triangles and or other subgraphs of interest. Indeed the more general BJR model has already been extended in this way [16]. Again one could specify a suitable subclass which would produce inhomogeneous random graphs with tunable clustering. The generation algorithm would then involve evaluating certain two-dimensional kernels (as well as the one dimensional case treated here). The detailed description and analysis of such and algorithm is beyond the scope of this paper and we leave it for future studies.

A Model equivalence

Though the two models $G(n, \kappa)$ and $G'(n, \kappa)$ appear slightly different, we now show that they are asymptotically equivalent under realistic assumptions. Recall that the power of the model $G(n, \kappa)$ comes from the fact the model can be related to the kernel κ. Specifically, many asymptotic properties can be computed directly using the kernel. The same statement holds for the variant model $G'(n, \kappa)$; asymptotic properties of the graphs can be computed from κ. Given this, it is natural to expect that the two models, $G(n, \kappa)$ and $G'(n, \kappa)$, are in some sense equivalent in the asymptotic limit, and we now prove this.

Asymptotic equivalence is defined in the following way [30]. Let $G(n, p_{ij})$ and $H(n, q_{ij})$ be two random graph models defined on $\mathbf{v}_n$ with edges chosen independently: $v_i \sim v_j$ with

probability p_{ij} (q_{ij}, respectively). These two models are *asymptotically equivalent* if for every sequence (A_n) of sets of graphs on defined on $\mathbf{v}_n$, we have

$$\mathbb{P}[G_n \in A_n] - \mathbb{P}[H_n \in A_n] \to 0\,,$$

for G_n and H_n sampled form $G(n, p_{ij})$ and $H(n, q_{ij})$ respectively.

To show that the models $G(n, \kappa)$ and $G'(n, \kappa)$ are, under reasonable assumptions, asymptotically equivalent, we will make use of the following theorem.

Theorem 1 (Janson [30]) *The models $G(n, p_{ij})$ and $H(n, q_{ij})$ are asymptotically equivalent if*

- $q_{ij} = p_{ij} + \mathcal{O}(p_{ij}^2)$ *(the implied constant is uniform over* n *and choice of* i *and* j*)*
- $\sum_{i<j} p_{ij}^3 = o(1)$.

Janson further showed that if $\int_{[0,\,1]\times[0,\,1]} \kappa^2\, dxdy < \infty$, then $\sum_{i<j} p_{ij}^3 = o(1)$ holds for the model $G(n, \kappa)$. We will use this result together with Theorem 1 to show that $G(n, \kappa)$ and $G'(n, \kappa)$ are asymptotically equivalent when κ is bounded away from zero and has bounded derivative $\partial\kappa/\partial x$.

Proposition 1. *Let κ be be bounded away from 0. Suppose also that κ is differentiable at all points (x, y) for x, $y \notin D$ and that the derivative $\partial\kappa/\partial x$ is bounded. Then the models $G(n, \kappa)$ and $G'(n, \kappa)$ are asymptotically equivalent.*

Proof. By symmetry, the derivatives of κ are bounded, so κ is also bounded. Thus by Janson's work, described above, we know that $\sum_{i<j} p_{ij}^3 = \mathcal{O}(1)$. By Theorem 1 it is thus sufficient to show that $p'_{ij} = p_{ij} + \mathcal{O}(p_{ij}^2)$. Without loss of generality, let $i < j$. Define $f(x)$ as

$$f(x) := -\int_{v_i}^{x} \kappa(x, v_j)dx = \int_{x}^{v_i} \kappa(x, v_j)dx.$$

Then applying Taylor's Theorem to approximate $f(v_i - 1/n)$ by $f(v_i)$,

$$\int_{v_i - 1/n}^{v_i} \kappa(x, v_j)dx = \frac{1}{n}\kappa(v_i, v_j) - \frac{1}{2n^2}\frac{\partial\kappa}{\partial x}(x', v_j)\,,$$

for some $x' \in [v_i - 1/n, v_i]$. Since κ bounded away from zero and $\partial\kappa/\partial x$ is bounded, there exists a constant C, depending only on κ, such that $|\,\frac{\partial\kappa}{\partial x}(x', v_j)\,| \leq C\kappa(v_i, v_j)^2$ and therefore

$$\int_{v_i - 1/n}^{v_i} \kappa(x, v_j)dx \leq \frac{1}{n}\kappa(v_i, v_j) + \mathcal{O}\left(\left(\frac{\kappa(v_i, v_j)}{n}\right)^2\right).$$

We have shown that $I_{i,j} := \int_{v_i - 1/n}^{v_i} \kappa(x, v_j)dx$ satisfies

$$I_{i,j} = p_{ij} + \mathcal{O}(p_{ij}^2).$$

Now, by definition,

$$p'_{ij} = 1 - \exp(-I_{i,j}),$$

which implies that

$$I_{i,j} \geq p'_{ij} \geq I_{i,j} - \frac{I_{i,j}^2}{2}.$$

Thus indeed,

$$p'_{ij} = p_{ij} + \mathcal{O}(p_{ij}^2).$$

This asymptotic equivalence condition for the two models $G(n, \kappa)$ and $G'(n, \kappa)$ is quite strong and relies on the assumptions that there exists an ε with $\kappa > \varepsilon$ and that κ and $\partial\kappa/\partial x$ are bounded.

B Random Kernel Graph Characteristics

Many properties of the graph $G(n, \kappa)$ can be computed directly using the kernel κ. In particular, κ determines asymptotic properties of the graphs $G(n, \kappa)$ as $n \to \infty$ [15].

Degree sequences

Though the model $G(n, \kappa)$ cannot generate graphs with fixed degree sequences it can generate random graphs with a given expected degree as a generalization of the Chung-Lu model [12]. To see this, note that the degree $d_G(x)$ of a vertex x in $G = G(n, \kappa)$ satisfies

$$\lim_{n\to\infty} \mathbb{E}[d_G(x)] = \int_{[0,1]} \kappa(x, y)dy. \tag{11}$$

Thus if κ is a multiplicatively separable function, i.e. can be written as $\kappa(x, y) = \psi(x)\psi(y)$ for some $\psi:[0, 1] \to \mathbb{R}^+$, then the expected degree of a vertex x in G will be proportional to $\psi(x)$,

$$\lim_{n\to\infty} \mathbb{E}[d_G(x)] = \psi(x) \int_{[0,1]} \psi(y)dy.$$

The full degree distribution can be determined as follows. Let $\lambda(x) = \int_{[0,\,1]}\kappa(x, y)dy$. Then in the asymptotic limit, the degree distribution will converge in probability to the mixed Poisson distribution $\int_{[0,\,1]}\mathrm{Po}(\lambda(x))dx$ [15], Theorem 3.13].

Subgraph density

Since the expected degree of each vertex can be determined (asymptotically), it is not surprising that the edge density can also be computed. Let $e(G)$ denote the number of edges in $G = G(n, \kappa)$. Then

$$\lim_{n\to\infty} \frac{\mathbb{E}[e(G)]}{n} = \frac{1}{2} \int_{[0,1]^2} \kappa(x, y)dx\ dy. \tag{12}$$

In fact, it is just as easy to determine asymptotically the number of other expected subgraphs H. Here we give the expected number of paths and cycles in the graph; other subgraphs of interest can be similarly determined (see [15] and [16]). Let $P_k(G)$ and $C_k(G)$ denote the number of paths and cycles of length k in the random kernel graph $G = G(n, \kappa)$. Then

$$\lim_{n\to\infty} \frac{\mathbb{E}[P_k(G)]}{n} = \frac{1}{2} \int_{[0,1]^{k+1}} \kappa(x_0, x_1)\kappa(x_1, x_2) \cdots, \kappa(x_{k-1}, x_k)dx_0\ dx_1 \cdots dx_k\,, \tag{13}$$

and,

$$\lim_{n\to\infty} \frac{\mathbb{E}[C_k(G)]}{n} = \frac{1}{2k} \int_{[0,1]^k} \kappa(x_1, x_2) \cdots, \kappa(x_{k-1}, x_k), \kappa(x_k, x_1)dx_1\ dx_2 \cdots dx_k\,. \tag{14}$$

Assortativity

A common measure of the assortativity of a graph is Pearson's correlation coefficient, otherwise known as assortativity. The correlation coefficient can be written in terms of the number of copies of certain small subgraphs in the graph. For a fixed subgraph H, let $n(H, G)$ be the number of isomorphic copies of H in G. Define

$$\tau(\kappa, H) := \lim_{n\to\infty} \frac{1}{n}\mathbb{E}[n(H, G)].$$

In the asymptotic limit, the assortativity coefficient of G is given by

$$\lim_{n\to\infty} \rho(G) = \frac{\tau(P_3)\tau(P_1) - \tau(P_2)^2}{3\tau(K_{1,3})\tau(P_1) + \tau(P_1)\tau(P_2) - \tau(P_2)^2}. \quad (15)$$

Note that τ depends on the kernel κ but we dropped this dependence from our notation to improve readability. The graph $K_{1,3}$ is the complete bipartite graph with parts of size 1 and 3, that is a star with three leaves. This derivation of the asymptotic assortativity coefficient can be found in [16]. (Note that in that derivation there is a $\tau(K_3)$ term which we safely ignore since it is zero for all the graphs we study.)

Acknowledgments

We would like to thank Terry Haut, Joel Miller, and Pieter Swart for helpful comments and suggestions.

Author Contributions

Conceived and designed the experiments: AH NL. Performed the experiments: AH NL. Wrote the paper: AH NL.

References

1. Newman M. Networks: An Introduction. New York, NY, USA: Oxford University Press, Inc.; 2010.
2. Bollobás B. Random Graphs. Cambridge University Press; 2001.
3. Newman M. The Structure and Function of Complex Networks. SIAM Review. 2003; 45(2):167–256. doi: 10.1137/S003614450342480
4. Aiello W, Chung F, Lu L. A Random Graph Model for Power Law Graphs. Experimental Mathematics. 2001; 10(1):53–66. doi: 10.1080/10586458.2001.10504428
5. Hay M, Miklau G, Jensen D, Towsley D, Li C. Resisting structural re-identification in anonymized social networks. The VLDB Journal. 2010; 19(6):797–823. doi: 10.1007/s00778-010-0210-x
6. Vespignani A. Modelling dynamical processes in complex socio-technical systems. Nat Phys. 2012 Jan; 8(1):32–39. doi: 10.1038/nphys2160
7. Durrett R. Some features of the spread of epidemics and information on a random graph. PNAS. 2010; 107(10):4491–4498. Available from: http://www.pnas.org/content/107/10/4491.abstract. doi: 10.1073/pnas.0914402107 PMID: 20167800
8. Chakrabarti D, Zhan Y, Faloutsos C. 43. In: R-MAT: A Recursive Model for Graph Mining. SIAM; 2004. p. 442–446. Available from: http://epubs.siam.org/doi/abs/10.1137/1.9781611972740.43.
9. Kolda TG, Pinar A, Plantenga T, Seshadhri C. A Scalable Generative Graph Model with Community Structure. SIAM Journal on Scientific Computing. 2014; 36(5):C424–C452. doi: 10.1137/130914218
10. Gilbert EN. Random graphs. The Annals of Mathematical Statistics. 1959;p. 1141–1144. doi: 10.1214/aoms/1177706098
11. Erdős P, Rényi A. On the evolution of random graphs. Magyar Tud Akad Mat Kutató Int Közl. 1960; 5:17–61.
12. Chung F, Lu L. Connected components in random graphs with given expected degree sequences. Ann Comb. 2002; 6(2):125–145. doi: 10.1007/PL00012580

13. Bollobás B. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. European J Combin. 1980; 1(4):311–316. doi: 10.1016/S0195-6698(80)80030-8
14. Söderberg B. General formalism for inhomogeneous random graphs. Phys Rev E. 2002 Dec; 66:066121. doi: 10.1103/PhysRevE.66.066121
15. Bollobás B, Janson S, Riordan O. The phase transition in inhomogeneous random graphs. Random Structures Algorithms. 2007; 31(1):3–122. doi: 10.1002/rsa.20168
16. Bollobás B, Janson S, Riordan O. Sparse random graphs with clustering. Random Structures Algorithms. 2011; 38(3):269–323. doi: 10.1002/rsa.20322
17. Barrett C, Eubank S, Marathe M. Modeling and Simulation of Large Biological, Information and Socio-Technical Systems: An Interaction Based Approach. In: Interactive Computation. Springer Berlin Heidelberg; 2006. p. 353–392. Available from: http://dx.doi.org/10.1007/3-540-34874-3_14.
18. Batagelj V, Brandes U. Efficient generation of large random networks. Phys Rev E. 2005 Mar; 71:036113. doi: 10.1103/PhysRevE.71.036113
19. Miller JC, Hagberg A. Efficient generation of networks with given expected degrees. In: Algorithms and models for the web graph. vol. 6732 of Lecture Notes in Comput. Sci. Heidelberg: Springer; 2011. p. 115–126.
20. Hagberg AA, Schult DA, Swart PJ. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy2008). Pasadena, CA USA; 2008. p. 11–15. Http://networkx.github.io/.
21. Albert R, Barabási AL. Statistical mechanics of complex networks. Reviews of Modern Physics. 2002 Jan; 74(1):47–97. doi: 10.1103/RevModPhys.74.47
22. Newman M. Assortative Mixing in Networks. Physical Review Letters. 2002; 89(20). doi: 10.1103/PhysRevLett.89.208701
23. Litvak N, van der Hofstad R. Uncovering disassortativity in large scale-free networks. Phys Rev E. 2013 Feb; 87:022801. doi: 10.1103/PhysRevE.87.022801
24. Blitzstein J, Diaconis P. A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees. Internet Mathematics. 2011; 6(4):489–522. doi: 10.1080/15427951.2010.557277
25. Bender EA, Canfield ER. The asymptotic number of labeled graphs with given degree sequences. Journal of Combinatorial Theory, Series A. 1978; 24(3):296–307. doi: 10.1016/0097-3165(78)90059-6
26. Bayati M, Kim JH, Saberi A. A sequential algorithm for generating random graphs. Algorithmica. 2010; 58(4):860–910. doi: 10.1007/s00453-009-9340-1
27. Kannan R, Tetali P, Vempala S. Simple Markov-chain algorithms for generating bipartite graphs and tournaments. Random Struct Algorithms. 1999 Jul; 14(4):293–308. doi: 10.1002/(SICI)1098-2418(199907)14:4%3C293::AID-RSA1%3E3.0.CO;2-G
28. Ray J, Pinar A, Seshadhri C. Are We There Yet? When to Stop a Markov chain while Generating Random Graphs. In: Bonato A, Janssen J, editors. Algorithms and Models for the Web Graph. vol. 7323 of Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2012. p. 153–164. Available from: http://dx.doi.org/10.1007/978-3-642-30541-2_12.
29. Seshadhri C, Kolda TG, Pinar A. Community structure and scale-free collections of Erdős-Rényi graphs. Phys Rev E. 2012 May; 85:056109. doi: 10.1103/PhysRevE.85.056109
30. Janson S. Asymptotic equivalence and contiguity of some random graphs. Random Structures Algorithms. 2010; 36(1):26–45. doi: 10.1002/rsa.20297