MULTIFRACTAL NETWORK GENERATORS

AUSTIN R. BENSON, CARLOS RIQUELME, SVEN P. SCHMIT (20)

Abstract. Generating random graphs to model networks has a rich history. In this paper, we explore a recent generative model, the multifractal network generator, through both a theoretical and empirical lens. We derive concise analytic formulas for the moments of graph properties, including number of edges, number of wedges, and number of triangles. Our empirical experiments show that the multifractal network generator can simultaneously model a few graph properties of real-world networks to high accuracy using a very fast algorithm. Furthermore, we give evidence that fitting these local properties carries over well to global properties, such as degree distribution.

1. Generative graph models

1.1. **Prior work.** Methods for generating random graphs that can realistically model the structure in networks we see in the real-world have received much attention. The oldest and best understood random graph model is due to Erdős and Rényi [1], but it fails to resemble real-world networks. Subsequently, more complicated models have arisen, such as the Kleinberg's small-world model [4] and the Watts-Strogatz model [13].

1.1.1. *Stochastic Kronecker graphs.* More recently, Leskovec *et al.* introduced Stochastic Kronecker graphs (SKG) [5], and the compact generative model has effectively reproduced real-world graph properties such as heavy-tailed degree distributions and small diameter. SKGs [5, 6] constitute the most widely used model that is similar in structure to the MFNG. Given a graph G, the KronFit algorithm [5] is a maximum likelihood estimation technique for creating an SKG that generates graphs similar to G. Similar work has applied SKG to graph completion [3]. More theoretical rigor has been developed for SKG, including the estimation of graph properties [2, 11].

1.1.2. *Multifractal network generators.* The multifractal network generator (MFNG) was introduced by Palla, Lovász, and Vicsek [9], and follows a recursive structure that resembles SKG. An advantage of MFNG is that several graph parameters can be easily estimated from the small set of generative parameters (see Section 3). In this project, we explore how easy it is to control these parameters to generate realistic networks. In particular, given a network, we want to reproduce similar networks via MFNG. The theoretical analysis for SKG is quite technical, and the simple characterization of graph properties from MFNG is attractive to our problem. Recent work by Palla [10] has extended the MFNG framework, but finding MFNG that model many real-world network properties has not been explored.

1.2. **Contributions.** This paper derives several analytic expressions for moments of graph properties in MFNG (see Section 3). In particular, we provide expressions for:

- The expectation and variance of the number of edges.
- The expected number of d-stars. This includes the expected number of wedges, which are 2-stars.
- The expected number of t-cliques. This includes the expected number of triangles, which 3-cliques.
- The degree distribution.

Date: December 10, 2013.

The central theoretical result is Theorem 2, which allows us to compute probabilities of properties by only looking at the original MFNG parameters, which is both appealing from both a theoretical and computational standpoint, especially for large networks. This constitutes a significant improvement over expressions for first moments of some graph properties in [9]. Futhermore, our framework makes it easier to develop more theoretical results.

We then use these new methods to fit MFNG parameters to real-world networks in Section 4. Our experiments show that MFNG is a promising approach to modelling networks.

2. Preliminaries

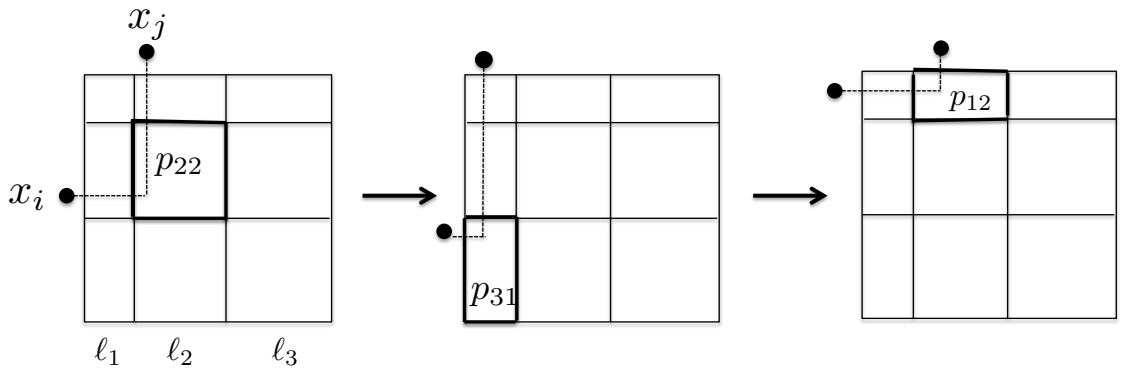
2.1. **Multifractal network generators.** The MFNG is a recursive generative model based on a generating measure, $\mathcal{W}_k$. The measure $\mathcal{W}_k$ consists of a set of probabilities p_{ij}, $p_{ij} = p_{ji}$, $1 \leq i, j \leq m$ along with a set of m lengths $l_1, \ldots, l_m$ with $\sum_{i=1}^{m} l_i = 1$. In this paper, we will refer to the intervals as *categories*. An undirected graph G is distributed according to $\mathcal{W}_k$ if it is generated by the following procedure:

(1) Partition $[0, 1]$ by the lengths l_i. For each interval, recursively partition the interval k times into m intervals, using the relative lengths l_i. This creates m^k intervals $l_{i_1,\ldots,i_k}$ of length $\prod_{x=1}^{k} l_{i_x}$ such that $\sum_{i_1,\ldots,i_k} l_{i_1,\ldots,i_k} = 1$.

(2) Sample N points uniformly from $[0, 1]$ and create nodes $x_1, \ldots, x_N$. Each node x_i lands in some interval of length $l_{i_1,\ldots,i_k}$. We can identify x_i by the sequence $(i_1, \ldots, i_k)$. In other words, x_i is identified by a k-tuple of categories.

(3) For every pair of nodes x_i and x_j identified by $(i_1, \ldots, i_k)$ and $(j_1, \ldots, j_k)$, we add edge (x_i, x_j) to G with probability $\prod_{x=1}^{k} p_{i_x j_x}$.

While the generation is intricate, MFNG admits a geometric interpretation. Consider first $k = 1$. We partition the unit square into rectangles according to the lengths l_i. The point $(x_i, x_j) \in [0, 1] \times [0, 1]$ lands in the unit square, inside a rectangle of area $l_{i_1} \cdot l_{j_1}$. The edge 'survives' to the next round with probability p_{i_1,j_1}. In the next round, we recursively create the unit square partition, scaled to the rectangle. The relative positions of x_i and x_j land the point in the rectangle of size $l_{i_2} \cdot l_{j_2}$. This process repeats k times. The idea is illustrated in Figure 1.



$$P((x_i, x_j) \in G) = p_{22}p_{31}p_{12}$$

Figure 1. Geometric interpretation of MFNG with $k = m = 3$. At each recursive level, the position of the nodes are relative to the position in the previous rectangle.

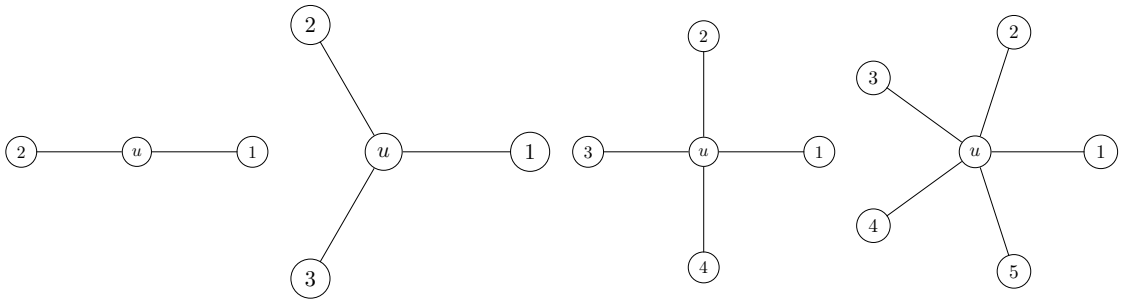


FIGURE 2. d-stars with center node u

The benefit of the multifractal network generator is that certain graph properties can be analytically computed given the generating measure and interval lengths. In [9], the authors provide simple, ineficient computations for a few graph properties. These formulas are given with respect to the extended $m^k \times m^k$ generating measure —that is, considering m^k interval. However, as we show in Section 3, we can efficiently compute many graph properties. In our approach the formulas are based on the independence of certain events happening at the k different levels of the process. In many cases, the expressions for MFNG are much simpler than the analogous ones for SKG. However, MFNG has not yet been shown to model real-world networks as well as SKG.

2.2. **Overview of graph properties.** In this section, we review the definitions and notation of graph properties considered in this paper. Given a graph $G = (V, E)$, we use the following definitions:

- A *wedge* is a pair of edges that share a single node, dentoed $\{(u, v), (u, w)\}$, where u, v, and w are all distinct.
- A *triangle* is a complete subgraph of three nodes, denoted (u, v, w). Note that the clustering coefficient of a graph is three times the number of triangles divided by the number of edges.
- A *d-star* is a node u along with edges $(u, v_1), \ldots, (u, v_d)$ for $u \neq v_i \in V$ distinct. Note that a 2-star is the same as a wedge. Figure 2 illustrates d-stars for $d \in \{2, 3, 4, 5\}$.

3. Theoretical results

In this section we state the main theoretical result of our work (Theorem 2). First, we give an interpretation and discuss why this is significant. Then we state the main theorem, followed by some corollaries. Proofs of all results are in Appendix A.

3.1. **Interpretation and significance.** The main theorem gives simple formulas for several graph related properties, where there is no need to expand the measure and use recursion. Expansion is used in [9], where they consider calculating properties after expanding the measure. This leads to a probability matrix of size $m^k \times m^k$, denoted by $\mathbf{P}$, compared to our method, which only needs to access the probabilities on the square, denoted by $\mathbf{p}$. Note that $\mathbf{p}$ is $m \times m$, and that k is implicitly a function of the number of nodes (in our experiments in Section 4, k is between $\lfloor \log_2 N/2 \rfloor$ and $\lceil \log_2 N \rceil$). Hence, using the theorem, we are able to scale these computations to graphs with arbitrary number of nodes.

3.2. **Notation.** Let $\mathcal{W}_k$ denote the distribution of a random graph generated by measure $\mathcal{W}$ with k levels. We call each of the m initial intervals a *category*, and we denote by c_i the category corresponding to the interval of length l_i, $1 \leq i \leq m$. $\mathcal{C}$ denotes the indices of all categories, i.e., $\mathcal{C} = \{1, \ldots, m\}$. In this section, all graphs have the same set of nodes V and $|V| = n$. A given node u belongs to k categories (one at each level), and the categories of node u are denoted by $c^u = (c_1, \ldots, c_k)$. In particular, we denote the i-th category, by c_i^u. Furthermore, the length of a category c is denoted by l_c.

3.3. **Main results.**

Lemma 1. *Let $\mathcal{W}$ be a given generating measure. Let graphs $H_1, \ldots, H_k \sim \mathcal{W}_1$, indepedently drawn, and also denote $H_i = (V, E_{H_i})$. Consider graph $G = (V, E_G)$ where, for each $u, v \in V$, $(u, v) \in E_G \iff (u, v) \in E_{H_i}$ for every $i = 1, \ldots, k$, i.e., G is the graph intersection of the H_i. Then, $G \sim \mathcal{W}_k$.*

The main result is the following theorem.

Theorem 2 (Main theorem)**.** *Consider a multifractal graph $G = (V, E)$, generated according to measure $\mathcal{W}_k$. For any event A on G that can be written as $A = \{S \subset E\}$, where $S \subset \{(i, j) : i, j \in \{1, \ldots, n\}, i < j\}$,*

$$\mathbb{P}_{\mathcal{W}_k}(A) = \mathbb{P}_{\mathcal{W}_1}(A)^k.$$

In other words, the probability that a subset of the edges exists if the graph is drawn from $\mathcal{W}_k$ is the k-th power of the probability that these edges exist if the graph is draw from $\mathcal{W}_1$. The condition that A can be written as $A = \{S \subset E\}$, where $S \subset \{(i, j) : i, j \in \{1, \ldots, n\}, i < j\}$ is very subtle. Let us therefore spend some time explaining its consequences. It states that the above theorem holds if we can specify a subset of the edges that must be present the graph. We can also be indifferent about certain edges, but the theorem does *not* allow us to specify that an edge is not present.

As an example, consider a graph with three nodes, u, v and w. We can use the above theorem to calculate the probability that it contains the wedge $\{(u, v), (v, w)\} \in E$ (and then use basic probability to calculate whether the graph contains a wedge). Furthermore, we can use the theorem to calculate the probability that a graph contains the triangle $(u, v, w) \in E$. However, we cannot use the theorem to calculate the probability that the wedge $\{(u, v), (v, w)\}$ is *not* a triangle, i.e. additionaly require $(u, w) \notin E$.

Let us look at one more anti-example. Define $\mu(G)$ to be the chromatic number of G. Then we cannot use the theorem to compute $\mathbb{P}(\mu(G) < 10)$, as that would give us $\mathbb{P}(\mu(G) < 10) = \mathbb{P}(\mu(H_1) < 10)^k$. But clearly, $\mathbb{P}(\mu(G) < 10) \geq \mathbb{P}(\mu(H_1) < 10)$ since taking the intersection of graphs can only reduce the chromatic number.

The following corrolary summarizes how we can efficiently compute several useful graph properties using Theorem 2.

Corollary 3. *Let G be a multifractal graph with n nodes generated according to measure $\mathcal{W}_k = (m, k, l, p_{ij})$.*

(1) *The expected number of edges $|E|$ in G, is*

$$\mathbb{E}[|E|] = \binom{n}{2} s^k,$$

and its variance is given by

$$Var(|E|) = \binom{n}{2} s^k \left(1 - \binom{n}{2} s^k\right) + 2n \binom{n-1}{2} \omega + \binom{n}{2} \binom{n-2}{2} s^{2k},$$

where we define

$$s := \sum_{i,j \in \mathcal{C}} p_{ij} l_i l_j, \qquad \omega := \sum_{i,j,t \in \mathcal{C}} p_{ij} p_{it}\ l_i l_j l_t.$$

(2) *The expected number of wedges Λ in G is given by*

$$\mathbb{E}[\Lambda] = n \binom{n-1}{2} \omega.$$

(3) *The expected number of $t-$cliques C_t in G is given by*

$$\mathbb{E}[C_t] = \binom{n}{t}\ s_t^k, \qquad \text{where} \quad s_t := \sum_{i_1, \ldots, i_t \in \mathcal{C}} \left(\prod_{\substack{j,q \in [t] \\ j<q}} p_{i_j i_q} \right) l_{i_1} l_{i_2} \cdots l_{i_t}.$$

In particular, the expected number of triangles Δ is

$$\mathbb{E}[\Delta] = \binom{n}{3} s_3^k, \qquad \text{where} \quad s_3 := \sum_{i,j,t \in \mathcal{C}} p_{ij} p_{it} p_{jt}\ l_i l_j l_t.$$

(4) *We denote the expected number of nodes with degree d by E_d, which satisfies*

$$\mathbb{E}[E_d] = \mathbb{E}[X_d] - \sum_{i=d+1}^{n} \binom{i}{d} \mathbb{E}[E_i],$$

where X_d is the number of $d-$stars in the graph and its expectation is given by

$$\mathbb{E}[X_d] = n\binom{n-1}{d} \left[\sum_{i_1,\ldots,i_{d+1} \in \mathcal{C}} \left(\prod_{j=2}^{d+1} p_{i_1 i_j} \right) \prod_{j=1}^{d+1} l_{i_j} \right]^k.$$

In order to be able to solve the recurrences, we also have that $\mathbb{E}[E_{n-1}] = \mathbb{E}[X_{n-1}]$.

Note that, for a given measure $\mathcal{W}_k$, we could empirically compute the value of $\mathbb{E}[C_t]$ for each t until we find $\mathbb{E}[C_{t^*}] \geq 1 > \mathbb{E}[C_{t^*+1}]$, which is a good indicator of the expected clique number of G —size of the largest induced complete subgraph in G—, while a concentration result is still needed to claim that the clique number will be in a small neighborhood of t^* with high probability.

4. Experimental results

Given a graph G, we are interested in finding a generating measure $\mathcal{W}_k$ such that the MFNG has similar properties. Since MFNGs generate random graphs, we seek to match the *expected value* of several graph properties to the true graph properties. Since we can easily compute the moments of several graph properties from the generating measure, we use an optimization framework.

4.1. **Fitting graphs to MFNG.** Let $\mathcal{P} = \{P_j\}$ denote a set of graph properties that we want to match. For example, $P_j(G)$ could be the number of edges or the number of triangles in the graph. For a generating measure $\mathcal{W}$, we will interpret $P_j(\mathcal{W})$ to be a random variable. We will also define a distance function $\mathbf{D}$ that measures the distance between a graph G and a generating measure $\mathcal{W}$, i.e., $\mathbf{D}(\mathcal{W}, G) \in \mathbb{R}$. $\mathbf{D}$ should reflect the graph properties that we want to match. For this report, we will restrict $\mathbf{D}$ to be of the following form:

$$\mathbf{D}(\mathcal{W}, G) = \sum_{i=1}^{|\mathcal{P}|} \frac{\left(\mathbb{E}[P_j(\mathcal{W})] - P_j(G)\right)^2}{\left(P_j(G)\right)^2}$$

We could also include different weights w_j for different properties P_j. However, for simplicity, we equally weight each property. Suppose that k is fixed and m is upper bounded by m_u. Then we are interested solving the following optimization problem

$$\begin{aligned} \underset{p,l}{\text{minimize}} \quad & \mathbf{D}(\mathcal{W}, G) \\ \text{subject to} \quad & 0 \leq p_{ij} \leq 1, \quad 1 \leq i \leq j \leq m_u \\ & 0 \leq l_i \leq 1, \qquad\quad 1 \leq i \leq m_u \\ & \sum_{i=1}^{m_u} l_i = 1 \end{aligned} \tag{4.1}$$

The objective function is nonlinear, and the constraints are linear. The number of variables is $\frac{1}{2}m_u(m_u + 1) + m_u$. Each variable has a lower and upper bound, and there is a single equality constraints (the lengths must sum to one), so there are $m_u(m_u - 1) + 4m_u + 1$ constraints. Note that, in this framework, m can effectively take any value in $\{1, \ldots, m_u\}$. The effective value of m is the number of non-zero l_i.

Graph property	MFNG	SKG
$2\mathbb{E}[E]$	$n(n-1)\left(\sum_{i=1}^{m}\sum_{j=1}^{m} l_i l_j p_{ij}\right)^k$	$(\alpha+2\beta+\gamma)^r-(\alpha+\gamma)^r$
$2\mathbb{E}[\Lambda]$	$n(n-1)(n-1)\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{h=1}^{m} l_i l_j l_h p_{ij} p_{ih}\right)^k$	$\left((\alpha+\beta)^2+(\beta+\gamma)^2\right)^r$ $-2\left(\alpha\left(\alpha+\beta\right)+\gamma\left(\gamma+\beta\right)\right)^r$ $-\left(\alpha^2+2\beta^2+\gamma^2\right)^r+2\left(\alpha^2+\gamma^2\right)^r$
$6\mathbb{E}[\Delta]$	$n(n-1)(n-2)\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{h=1}^{m} l_i l_j l_h p_{ij} p_{ih} p_{jh}\right)^k$	$\left(\alpha^3+3\beta^2\left(\alpha+\gamma\right)+\gamma^3\right)^r$ $-3\left(\alpha\left(\alpha^2+\beta^2\right)+\gamma\left(\beta^2+\gamma^2\right)\right)^r$ $+2\left(\alpha^3+\gamma^3\right)^r$

TABLE 1. Analytic expressions for the expectation of graph properties for SKG and MFNG in terms of model parameters, where n is the number of nodes in the graph. The results for SKG are from [2].

As a consequence of Theorem 2, we can compute $\mathbb{E}[P_j(\mathcal{W})]$ for several P_j in $O(m_u^3)$ time. This allows us to quickly evaluate $\mathbf{D}(\mathcal{W}, G)$ and efficiently use optimization software.

We will use SKG as a comparison against MFNG. Again, we will use $\mathbf{D}(\Theta, G)$ to denote a distance function between a SKG with an initiator matrix Θ and again interpret $P_j(\Theta)$ as a random variable. Furthermore, we restrict Θ to be a 2×2 matrix:

$$\Theta = \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}.$$

The optimization problem for SKG is:

$$\begin{aligned} \underset{\alpha,\beta,\gamma}{\text{minimize}} \quad & \mathbf{D}(\Theta, G) \\ \text{subject to} \quad & 0 \le \alpha \le 1 \\ & 0 \le \beta \le 1 \\ & 0 \le \gamma \le 1 \end{aligned} \tag{4.2}$$

There are three variables and six constraints. More sophisticated techniques to solving (4.2) exist [2]; however, we since our focus is on SKG, we use this simple formulation.

4.2. **Fitting edges, wedges, and triangles.** For our experiments, we attempt to match the number of edges (E), wedges (Λ), and triangles (Δ) of real-world graphs. These properties are easy to compute for SKG [2], which we will use for comparison. Table 1 summarizes the expected value of these properties for MFNG and SKG. We solve (4.1) and (4.2) using Matlab's active set nonlinear optimization solver. For each optimization problem, we use 2000 trials with a random starting vector and take the best result over all trials. Note that the graph properties for MFNG depend on k, the number of recursive layers. We sample k uniformly from $\{\lfloor \log_2(N) \rfloor / 2, \ldots, \lceil \log_2(N) \rceil\}$ and solve the optimization problem with k fixed. A disadvantage of SKG is that the exponent r in Table 1 is fixed at $\log_2(N)$. For large r, this makes the function evaluations quite sensitive to changes in parameters. We also use KronFit to generate a maximum-likelihood SKG initiator matrix. The approach of KronFit is fundamentally different from the non-linear optimization approach, and we provide the results for comparison. The following notation is used in the results:

- Let P_1 be the number of edges in the graph and define $E_{err} = \left(\mathbb{E}[P_1(\mathcal{W})] - P_1(G)\right)^2 / \left(P_1(G)\right)^2$ to be the error in the number of edges.
- Let P_2 be the number of wedges in the graph and define $\Lambda_{err} = \left(\mathbb{E}[P_2(\mathcal{W})] - P_2(G)\right)^2 / \left(P_2(G)\right)^2$.
- Let P_3 be the number of triangles in the graph and define $\Delta_{err} = \left(\mathbb{E}[P_3(\mathcal{W})] - P_3(G)\right)^2 / \left(P_3(G)\right)^2$.

Network	n	E	Λ	Δ
Pokec	1.63e06	3.06e07	2.09e09	3.26e07
cit-HepPh	3.45e04	4.20e05	2.63e07	1.28e06
roadNet-CA	1.96e06	2.77e06	6.00e06	1.20e05

TABLE 2. Summary of graph properties of the three networks used in our experiments. n is the number of nodes, E the number of edges, Λ the number of wedges, and Δ the number of triangles.

Network	Method	E_{err}	Λ_{err}	Δ_{err}	(p_{11}, p_{12}, p_{22})	(l_1, l_2)	k	(α, β, γ)
Pokec								
	MFNG	2.87e-12	8.96e-12	1.26e-12	(0.91, 0.09, 0.70)	(0.57, 0.43)	14	–
	SKG	2.87e-5	3.09e-04	0.96	–	–	–	(0.52, 0.41, 1.00)
	SKG (KF)	0.17	0.46	1.00	–	–	–	(0.82, 0.55, 0.37)
cit-HepPh								
	MFNG	6.08e-11	3.37e-13	3.14e-11	(0.68, 0.18, 0.98)	(0.36, 0.64)	13	–
	SKG	1.38e-6	0.0019	0.92	–	–	–	(0.23, 0.56, 1.00)
	SKG (KF)	1.94e-05	0.12	0.98	–	–	–	(1.00, 0.50, 0.35)
roadNet-CA								
	MFNG	0.011	0.002	7.73e-11	(1.00, 0.04, 0.76)	(0.43, 0.57)	17	–
	SKG	0.0034	0.0062	0.13	–	–	–	(0.96, 0.08, 1.00)
	SKG (KF)	0.032	0.018	0.995	–	–	–	(0.67, 0.41, 0.57)

TABLE 3. Results of fitting MFNG and SKG to real-world graphs using (4.1) and (4.2), as well as the results for KronFit for comparison. MFNG reproduces the expected number of edges, wedges, and triangles to high accuracy, while SKG has difficulty matching the number of triangles.

Our test graphs consist of three networks representing vastly different data. The first is the Pokec social network [12], which contains data about all links between users in Slovakia. This network is especially interesting because it contains the full graph of a social network, which therefore suffers less from selection bias as 'ego networks'. Although this network is directed, we transform it and treat it as an undirected network. The second network is a citation network in the high energy physics community (cit-HepPh) [7]. Apart from the two social networks above, we also consider a "non-social" network, the California road network (roadNet-CA) [8]. For a background and analysis of the networks, we refer to the cited papers. Table 2 summarizes the relevant graph propreties of these networks.

The results of the optimization procedure and KronFit are in Table 3. We include the values of the probability matrix of MFNG (p_{11}, p_{12}, p_{22}), the lengths of the intervals of MFNG (ℓ_1, ℓ_2), and the SKG parameters (α, β, γ). Note that KronFit is *not* trying to match the graph properties, and the results are given for comparison. The results show that MFNG accurately matches the expected number of edges, wedges, and triangles of the networks. On the other hand, SKG has difficulty matching the expected number of triangles. KronFit does a reasonable job at matching the number of edges and wedges, even though the algorithm is not directly optimizing for these graph properties. The MFNG fit shows that properties along the diagonal (p_{11} and p_{22}) tend to be very large. This is expected. In clustered networks, nodes falling into the same category in MFNG should be more likely to connect, and this is reflected by the large diagonal probabilities. In all cases, the lengths (l_1 and l_2) for MFNG are not skewed. Heuristically, this tells us that the fit is realistic and not too 'extreme'.

Network	$S^2 = \Lambda$	S^3	S^4	S^5
Pokec	2.08e09	1.04e12	2.83e15	7.40e18
cit-HepPh	2.63e07	1.34e09	1.04e11	9.88e12
roadNet-CA	6.00e06	2.95e06	5.50e05	2.95e04

TABLE 4. Number of d-stars in the test netwroks for $d \in \{2, 3, 4, 5\}$. The number of 2-stars is the same as the number of wedges.

Network	E_{err}	Λ_{err}	Δ_{err}	S^3_{err}	S^4_{err}	S^5_{err}	(p_{11}, p_{12}, p_{22})	(l_1, l_2)	k
Pokec	0.21	0.040	0.96	0.0036	0.089	0.0073	(1.00, 0.71, 0.24)	(0.19, 0.81)	13
cit-HepPh	0.029	4.84e-05	0.0011	0.036	0.0063	0.047	(0.23, 0.18, 1.00)	(0.57, 0.43)	7
roadNet-CA	0.34	0.59	5.7e-11	0.67	0.50	0.08	(0.91, 0.03, 0.99)	(0.52, 0.48)	20

TABLE 5. Fit of MFNG to real-world networks while trying to match several graph properties. MFNG can match the citation network but has difficulty with the Pokec and roads networks.

4.3. **Additional fitting of d-stars in MFNG.** One property that we have ignored so far is degree distribution. While there are formulas for degree distribution in [9], they are approximations based on asymptotics of the binomial distribution. Instead, we will count d-stars as a proxy for degree distributin. Let S^d be the number of d-stars in a graph. In addition to fitting edges, wedges, and triangles, we now also try to fit S^d, for $d \in \{3, 4, 5\}$. Table 4 lists the number of d-stars in our test networks. For the Pokec and citation network, $S^2 < S^3 < S^4 < S^5$ while for the road network, $S^2 > S^3 > S^4 > S^5$.

We solved (4.1) while fitting these six graph properties. Again, there were 2000 calls to the optimization solver with random starting points. Table 5 lists the results of the best solution. For the Pokec network, MFNG fit all properties except for the number of triangles, and for the citation network, MFNG fit all properties. However, MFNG had a difficult time fitting the roads network. The difficulty with roads is most likely a result of the fact that $S^k < S^{k+1}$ for $k = 2, 3, 4$.

4.4. **Generating MFNGs efficiently.** One of the downsides of the MFNG graphs is that it is computationally intensive to generate them. A naive method might calculate the $\mathbf{P}$ matrix, which is $m^k \times m^k$, and then loop over every pair of nodes (u, v) and throw a properly (un)balanced coin. For reasonably sized graphs, k often is too large to compute this matrix explicitly.

We have implemented an approximation algorithm inspired by the "ball-dropping" algorithm used to generate SGKs [5]. Our algorithm consists of two steps. First, we group nodes that belong to the same set of categories, i.e., we group all nodes u by $c^u = (c_1, \ldots, c_k)$. The second part is more intricate. We consider D possible edges. For each edge, find a box at the lowest recursive layer of the MFNG via a ball-dropping procedure, using both the probability matrix as the lenghts of intervals. In particular, the probability of the ball being dropped into a box at each level is proportional to $\mathbf{p} \circ \mathbf{ll}'$, where $\mathbf{l}$ is the vector of initial lengths. After repeating this procedure k times, we have reached a box at the lowest level of the MFNG. For now, assume that the box contains at least one candidate edge, i.e., there is a pair of nodes $x_i, x_j \in [0, 1]$ such that (x_i, x_j) is in the box. From the set of all edge candidates, choose one uniformly at random and connect them.

At face value, there is a central flaw in this approach. Since the nodes are generate uniformly on $[0, 1]$, some category groups may have have a disproportionally small or large number of edge candidates. To account for this, we instead consider λD possible edges, where $\lambda > 1$. When arriving at a box, we calculate the expected number of edge candidates in the box, n_p, and the actual number of edge candidates in the box, n_a. We select a random edge from the box, and add it to the graph with probability $\frac{n_a}{\lambda n_p}$. This corrects the discrepancy in the number of edge candidates.

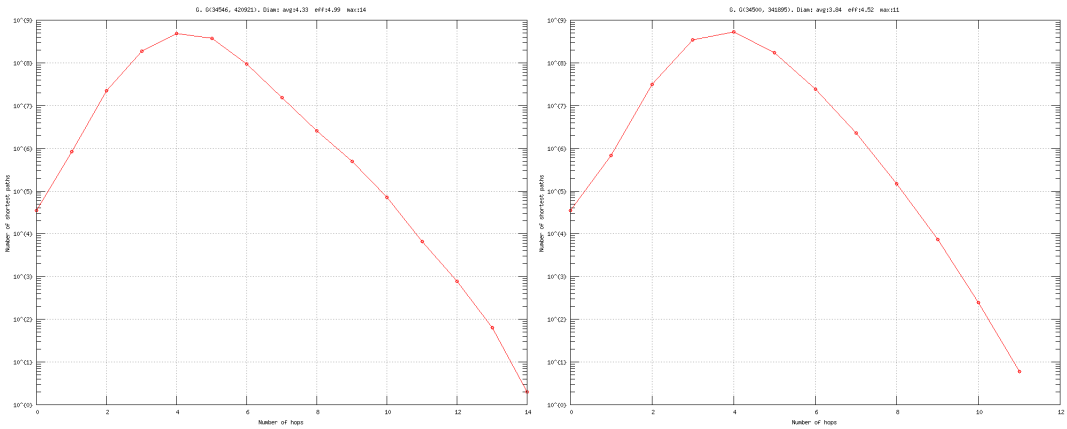


FIGURE 3. Hops plot for the Physics citation network (left) and the MFNG fitted graph (right).

We still have to pick λ. On one hand, we want λ to be as small as possible, as the larger λ, the more work we have to do. On the other hand, if λ too small, the quantity $\frac{n_a}{\lambda n_p}$ could be bigger than 1, resulting in artifacts in the graph. In practice, we find that λ between 2 and 3 works well.

Selecting D is another challenge. In the fast generation of SKGs, people often use $D = \mathbb{E}[|E|]$, but this is rather artificial. However, finding the distribution of $|E|$ is not straightforward. We opt for a compromise by computing $\mathbb{E}[|E|]$ and $\mathrm{Var}|E|$, and noting that since every edge is a Bernoulli trial, $|E|$ is tightly concentrated around $\mathbb{E}[|E|]$. Therefore, we sample $D \sim \mathcal{N}(\mathbb{E}[|E|], \mathrm{Var}[|E|])$. Emperically, this proves to be a very accurate approximation.

The last issue is empty boxes. We do not have a proper solution to this, except for avoiding them; to do this, we have to bound k, ensuring it is small enough to avoid empty boxes with high probability. While it would be desirable to get rid of this restriction, we can still generate realistic graphs with bounded k, as we see in the following section.

4.4.1. *Simulating large graphs.* The result is a method that can generate reasonably large graphs, that could also be parallelized for further speed-up. In this section, we compare the Physics citation network to a MFNG generated graph where the parameters are fitted to match edges, wedges, triangles, 3-stars, 4-stars and 5-stars. In particular, because we are only able to fit local parameters using the fast technique we described, it is both important and interesting to see how well this translates to global properties.

Figures 3, 4 and 5 show that we achieve mixed results. The hop plot (Figure 3) is very accurate, but the degree distribution (Figure 4) shows the same "staircase effect" as the (non-stochastic) Kronecker graphs [5]. The clustering coefficient (Figure 5) plot shows that, even though we can match the number of edges and wedges, that does not mean that the clustering coefficient per node is well matched.

5. Conclusion and future work

This paper investigated both the theoretical underlying of MFNGs as well as the practical implications by using it to fit real-world graphs. In particular, we have shown that it is possible to compute several important graph properties using very simple computations that only depend on the initial measure $\mathcal{W}$, and for which the running time is independent of k. It is remarkable that we are able to fit the MFNG model to large-scale graphs in negligible time, indeed it is much faster than calculating the properties of the graph we intend to fit. Furthermore, empirical results show that fitting local structures such as d-stars and triangles also leads to the fitting of global structures such as degree distribution.

There are several areas for future work. First, we would like to provide more rigorous results on fast generation of MFNG. This is three-fold. First, we have to upperbound k to deal with empty boxes, which

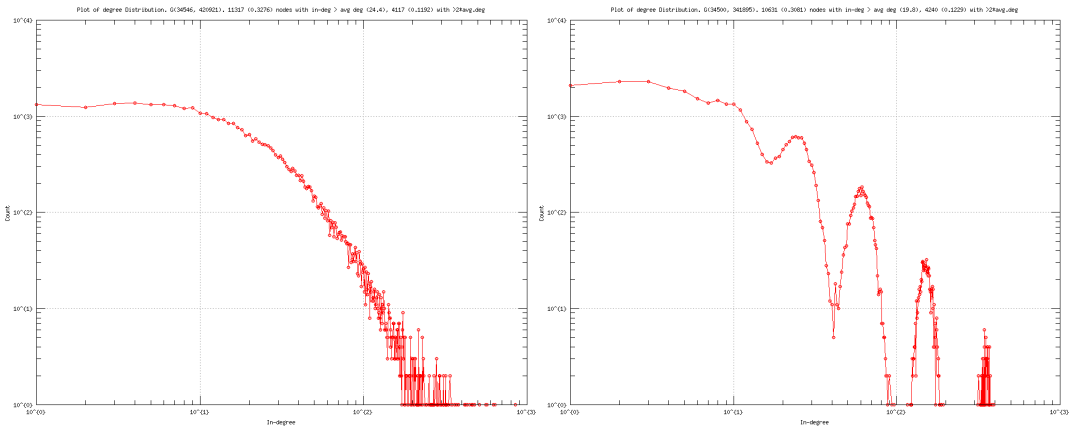


FIGURE 4. Degree distribution plot for the Physics citation network (left) and the MFNG fitted graph (right).

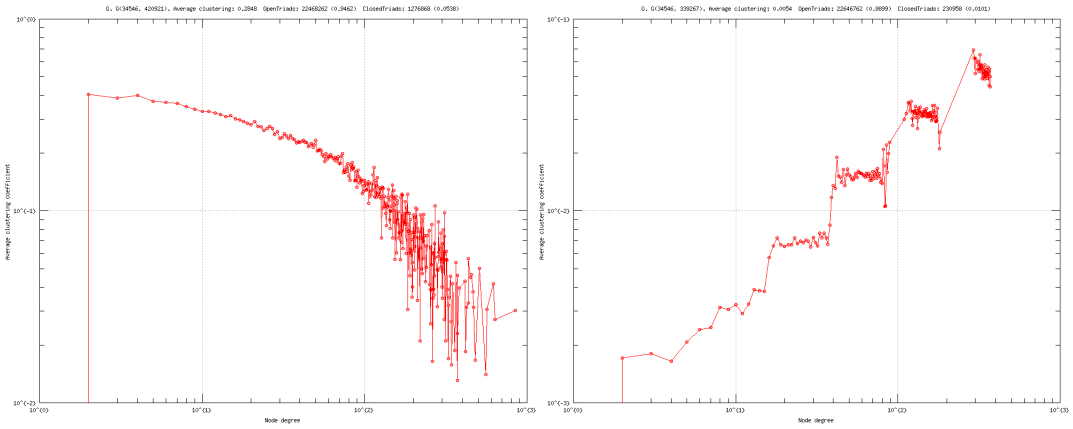


FIGURE 5. Clustering coefficient plot for the Physics citation network (left) and the MFNG fitted graph (right).

is restrictive. Second, the fast graph generator is still too slow to generate graphs with more than a few hundred thousand nodes. It would be very interesting to scale the method up such that it can be used to generate graphs with more than, say, a million nodes. We note that apart from improving the current algorithm, it is also possible to parallelize the current algorithm and generate larger graphs that way. Third, we notice the awkward artifacts in the degree distribution that are similar to the non-stochastic Kronecker graphs as noted in [5] when generating MFNG graphs. How to get rid of these artifacts is not exactly clear and would be desirable.

The optimization algorithm we use is fast but slightly ad hoc. More application-specific optimization routines can be explored. We also note that many different measures lead to an equally good fit of the model. This could point to room for additional properties to be taken into account, thereby differentiating between these models.

Furthermore, there is room for more theoretical results of graph properties. Can Theorem 2 be extended such that we can deal with more general events or used to calculate other properties of such graphs?

Lastly, it is also not clear that fitting the expected value of several graph properties is the best approach. We have shown some encouraging evidence that it might be, but further theoretical and empirical evidence is needed.

References

[1] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[2] D. F. Gleich and A. B. Owen. Moment-based estimation of stochastic kronecker graph parameters. *Internet Mathematics*, 8(3):232–256, 2012.

[3] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SDM*, pages 47–58, 2011.

[4] J. Kleinberg. The small-world phenomenon: an algorithm perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000.

[5] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

[6] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *Proceedings of the 24th international conference on Machine learning*, pages 497–504. ACM, 2007.

[7] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[8] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[9] G. Palla, L. Lovász, and T. Vicsek. Multifractal network generator. *Proceedings of the National Academy of Sciences*, 107(17):7640–7645, 2010.

[10] G. Palla, P. Pollner, and T. Vicsek. Rotated multifractal network generator. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(02):P02003, 2011.

[11] C. Seshadhri, A. Pinar, and T. G. Kolda. An in-depth analysis of stochastic kronecker graphs. *Journal of the ACM (JACM)*, 60(2):13, 2013.

[12] L. Takac and M. Zabovsky. Data analysis in public social networks.

[13] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

Appendix A. Proofs of main results

This appendix contains the proofs Lemma 1 and Theorem 2 along with calculations and proofs of the formulae in Corollary 3.

A.1. **Proof of Lemma 1.**

Proof. We prove the lemma by showing that a graph generated using the graphs $H_1, \ldots, H_k$ satisfies the construction of a graph whose distribution follows $\mathcal{W}_k$.

The proof consists of two parts:

(1) First, we show that

$$\mathbb{P}(c^u = (c_1, \ldots, c_k)) = \prod_{i=1}^{k} l_{c_i}.$$

(2) Then, we show that

$$\mathbb{P}((u,v) \in E_G | c^u = (c_1^u, \ldots, c_k^u), c^v = (c_1^v, \ldots c_k^v)) = \prod_{i=1}^{k} p_{c_i^u c_i^v},$$

independently of other edges.

Since the graphs H_i are independent, it follows that the components of $c^u = (c_1^u, \dots, c_k^u)$ are independent. Note also that the categories are independent of other nodes due to construction of $\{H_i\}$. Moreover, we have that

$$\mathbb{P}(c^u = (c_1, \dots, c_k)) = \prod_{i=1}^{k} \mathbb{P}(c_i^u = c_i) = \prod_{i=1}^{k} l_{c_i},$$

which completes the proof of (1).

For (2), we note that the independence between edges given the categories follows directly from the definition of $\{H_i\}$. Furthermore,

$$\mathbb{P}\left((u,v) \in E_G | c^u, c^v\right) = \mathbb{P}\left((u,v) \in E_{H_i} \forall i | c^u, c^v\right) = \prod_{i=1}^{k} \mathbb{P}\left((u,v) \in E_{H_i} | c_i^u, c_i^v\right) = \prod_{i=1}^{k} p_{c_i^u, c_i^v}$$

which completes the proof. □

A.2. **Proof of Theorem 2.**

Proof. This is a straightforward consequence of Lemma 1. Let A be an event that can be written as $A = \{S \subset E\}$, where $S \subset \{(i,j) : i,j \in \{1,\dots,n\}, i < j\}$. Then we have

$$\begin{aligned}
\mathbb{P}_{\mathcal{W}_k}(A) &= \mathbb{P}_{\mathcal{W}_k}(s \in E_G, \forall s \in S) \\
&= \mathbb{P}_{(\mathcal{W}_1)^k}(s \in E_{H_i}, \forall s \in S, \forall i \in \{1,\dots,k\}) \\
&= \prod_{i=1}^{k} \mathbb{P}_{(\mathcal{W}_1)^k}(s \in E_{H_i}, \forall s \in S) \\
&= \mathbb{P}_{(\mathcal{W}_1)^k}(s \in E_{H_1}, \forall s \in S)^k \\
&= \mathbb{P}_{\mathcal{W}_1}(A)^k.
\end{aligned}$$

□

A.3. **Proof of Corollary 3.**

A.3.1. *Expected number of edges.* Let u and v be two random nodes of G. Let A denote the event $A = \{(u,v) \in E\}$, and we define $A^{(i)}$ to denote the analogous event restricted to level i in the multifractal generator. By Theorem 2, we have that

$$\mathbb{P}(A) = \prod_{i=1}^{k} \mathbb{P}(A^{(i)}) = \mathbb{P}(A^{(1)})^k.$$

Now, we see that if we are only concerned with the first —original— level, then

$$\begin{aligned}
\mathbb{P}(A^{(1)}) &= \sum_{i,j \in \mathcal{C}} \mathbb{P}(A^{(1)} | c_1^u = i, c_1^v = j)\ \mathbb{P}(c_1^u = i, c_1^v = j) \\
&= \sum_{i,j \in \mathcal{C}} p_{ij}\ \mathbb{P}(c_1^u = i) \mathbb{P}(c_1^v = j) \\
&= \sum_{i,j \in \mathcal{C}} p_{ij}\ l_i l_j := s.
\end{aligned}$$

We conclude that

$$\mathbb{P}(A) = \mathbb{P}((u,v) \in E) = s^k. \tag{A.1}$$

The expected number of edges is then given by

$$\mathbb{E}[|E|] = \binom{n}{2}\ s^k. \tag{A.2}$$

A.3.2. *Expected number of triangles.* Let u, v and w be three random nodes of the graph. We define $E_{u,v,w}$ to be the event that there is a triangle between u, v, w and, similarly, let $E^{(i)}_{u,v,w}$ be the event that there is a triangle between u, v, w in level $i \in [k]$. Then, by Theorem 2,

$$\mathbb{P}(E_{u,v,w}) = \prod_{i=1}^{k} \mathbb{P}(E^{(i)}_{u,v,w}) = \mathbb{P}(E^{(1)}_{u,v,w})^k. \tag{A.3}$$

Now, we compute the probability of a triangle happening between three random nodes at the first level, that is, according to $\mathcal{W}_1$. We see that

$$\begin{aligned}
\mathbb{P}(E^{(1)}_{u,v,w}) &= \mathbb{P}((u,v),(u,w),(v,w) \in E) \\
&= \sum_{i,j,t \in \mathcal{C}} \mathbb{P}((u,v),(u,w),(v,w) \in E | c_1^u = i, c_1^v = j, c_1^w = t)\ \mathbb{P}(c_1^u = i, c_1^v = j, c_1^w = t) \\
&= \sum_{i,j,t \in \mathcal{C}} \mathbb{P}((u,v),(u,w),(v,w) \in E | c_1^u = i, c_1^v = j, c_1^w = t)\ \mathbb{P}(c_1^u = i)\mathbb{P}(c_1^v = j)\mathbb{P}(c_1^w = t) \\
&= \sum_{i,j,t \in \mathcal{C}} \mathbb{P}((u,v) \in E | c_1^u = i, c_1^v = j)\mathbb{P}((u,w) \in E | c_1^u = i, c_1^w = t)\mathbb{P}((v,w) \in E | c_1^v = j, c_1^w = t)\ l_i l_j l_t \\
&= \sum_{i,j,t \in \mathcal{C}} p_{ij} p_{it} p_{jt}\ l_i l_j l_t =: s_3.
\end{aligned}$$

By (A.3), we conclude that

$$\mathbb{P}(E_{u,v,w}) = s_3^k. \tag{A.4}$$

We can now compute the expected number of triangles Δ in G as

$$\mathbb{E}[\Delta] = \sum_{\substack{S \subset V \\ |S|=3}} \mathbb{1}(E_S) = \binom{n}{3} \mathbb{P}(E_{u,v,w}) = \binom{n}{3} s_3^k. \tag{A.5}$$

A.3.3. *Expected number of $t-$cliques.* According to the notation we used in the previous section, given t random nodes $S = \{u_1, \ldots, u_t\}$ and letting E_S be the event that they form a t-clique, we have that

$$\mathbb{P}(E_S) = \prod_{i=1}^{k} \mathbb{P}(E_S^{(i)}) = \mathbb{P}(E_S^{(1)})^k.$$

Also, it follows that

$$\mathbb{P}(E_S^{(1)}) = \sum_{i_1, \ldots, i_t \in \mathcal{C}} \left(\prod_{\substack{j,q \in [t] \\ j \neq q}} p_{i_j i_q} \right) l_{i_1} l_{i_2} \cdots l_{i_t} := s_t$$

Finally, let C_t be the expected number of $t-$cliques in G. We conclude that

$$\mathbb{E}[C_t] = \binom{n}{t}\ s_t^k. \tag{A.6}$$

A.3.4. *Expected number of wedges.* Let u, v, w be three distinct nodes of G. We define A to be the event that there is a wedge centered at u in G, that is, $A = \{(u,v),(u,w) \in E(G)\}$. Similarly, as in previous sections, we define $A^{(i)}$ to be the event restricted to level i. By Theorem2, we see that

$$\mathbb{P}(A) = \prod_{i=1}^{k} \mathbb{P}(A^{(i)}) = \mathbb{P}(A^{(1)})^k.$$

Now, by considering only the first level, we find that

$$\begin{aligned}\mathbb{P}(A^{(1)}) = \mathbb{P}((u,v),(u,w)\in E) &= \sum_{i,j,t\in\mathcal{C}} \mathbb{P}((u,v),(u,w)\in E | c_1^u = i, c_1^v = j, c_1^w = t)\mathbb{P}(c_1^u = i, c_1^v = j, c_1^w = t) \\ &= \sum_{i,j,t\in\mathcal{C}} \mathbb{P}((u,v)\in E | c_1^u = i, c_1^v = j)\mathbb{P}((u,w)\in E | c_1^u = i, c_1^w = t)\ l_i l_j l_t \\ &= \sum_{i,j,t\in\mathcal{C}} p_{ij}p_{it}\ l_i l_j l_t =: \omega.\end{aligned}$$

It follows that the expected number of wedges Λ in G is given by

$$\mathbb{E}[\Lambda] = n\binom{n-1}{2}\omega. \tag{A.7}$$

A.3.5. *Variance of the number of edges.* Let X_{ij} be the indicator random variable of the event $(v_i, v_j) \in E$ for $i \neq j$. We also define $X = \sum_{i\neq j} X_{ij}$, the total number of edges. Let us compute the second moment of X,

$$\begin{aligned}\mathbb{E}[X^2] &= \mathbb{E}\left[\left(\sum_{i<j} X_{ij}\right)\left(\sum_{i<j} X_{ij}\right)\right] \\ &= \mathbb{E}\left[\sum_{i<j} X_{ij}^2\right] + \mathbb{E}\left[\sum_{i,j\neq k} X_{ij}X_{ik}\right] + \mathbb{E}\left[\sum_{i\neq k, j\neq z, k\neq z} X_{ij}X_{kz}\right] \\ &= \mathbb{E}[X] + 2\mathbb{E}[\Lambda] + \sum_{i\neq k, j\neq z, k\neq z} \mathbb{E}\left[X_{ij}\right]\mathbb{E}\left[X_{kz}\right] \\ &= \mathbb{E}[X] + 2\mathbb{E}[\Lambda] + \binom{n}{2}\binom{n-2}{2}\mathbb{P}\left((v_i,v_j)\in E\right)^2 \\ &= \mathbb{E}[X] + 2\mathbb{E}[\Lambda] + \binom{n}{2}\binom{n-2}{2}s^{2k}.\end{aligned}$$

Hence, we see that

$$\begin{aligned}\mathrm{Var}(X) &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \\ &= \mathbb{E}[X] + 2\mathbb{E}[\Lambda] + \binom{n}{2}\binom{n-2}{2}s^{2k} - \mathbb{E}[X]^2 \\ &= \mathbb{E}[X](1-\mathbb{E}[X]) + 2n\binom{n-1}{2}\omega + \binom{n}{2}\binom{n-2}{2}s^{2k} \\ &= \binom{n}{2}s^k\left(1 - \binom{n}{2}s^k\right) + 2n\binom{n-1}{2}\omega + \binom{n}{2}\binom{n-2}{2}s^{2k}.\end{aligned}$$

A.3.6. *d-stars and Degree Distribution.* A $d-$star centered at node u is a graph containing $d+1$ vertices whose edges go from u to each of the other d vertices in the graph. We start by noting the following key and simple fact: the number of vertices with degree d in a graph G equals the number of copies of $d-$stars in G that are not part of any $(d+1)-$star in G. Let us define X_d to be the random variable that counts the number of $d-$stars in G, for any $d \in [n-1]$.

Let $d' > d$ and suppose that vertex u has degree d'. Then, u will contribute with $\binom{d'}{d}$ stars to X_d.

We define V_d to be the random variable that counts the number of nodes with degree $\geq d$. Similarly, we denote by E_d the number of nodes with degree d. We see that

$$E_d = V_d - V_{d+1},$$

which directly implies

$$\mathbb{E}[E_d] = \mathbb{E}[V_d] - \mathbb{E}[V_{d+1}].$$

Our goal is to write V_d as a function of X_d and X_{d+1}. We see that $E_{n-1} = X_{n-1}$ and

$$E_d = X_d - \sum_{i=d+1}^{n} \binom{i}{d} E_i.$$

Taking expectations, we conclude that

$$\mathbb{E}[E_d] = \mathbb{E}[X_d] - \sum_{i=d+1}^{n} \binom{i}{d} \mathbb{E}[E_i].$$

We can compute the expected number of $d-$stars in G. Let $u_1, \ldots, u_{d+1}$ be $d+1$ distinct nodes, and let S be the event that there is a $d-$star centered at u_1 in G. In other words, $S = \{(u_1, u_2), (u_1, u_3), \ldots, (u_1, u_{d+1}) \in E\}$. By Theorem 2,

$$\mathbb{P}(S) = \prod_{i=1}^{k} \mathbb{P}(S_i) = \mathbb{P}(S_1)^k.$$

Now, let us compute $\mathbb{P}(S_1)$.

$$\begin{aligned}
\mathbb{P}(S_1) &= \mathbb{P}((u_1, u_2), (u_1, u_3), \ldots, (u_1, u_{d+1}) \in E) \\
&= \sum_{i_1, \ldots, i_{d+1} \in \mathcal{C}} \mathbb{P}((u_1, u_2), (u_1, u_3), \ldots, (u_1, u_{d+1}) \in E | c_1^{u_j} = i_j \forall j)\ \mathbb{P}(c_1^{u_j} = i_j \forall j) \\
&= \sum_{i_1, \ldots, i_{d+1} \in \mathcal{C}} \mathbb{P}((u_1, u_2) \in E | c_1^{u_1} = i_1, c_1^{u_2} = i_2) \cdots \mathbb{P}((u_1, u_{d+1}) \in E | c_1^{u_1} = i_1, c_1^{u_{d+1}} = i_{d+1}) \prod_{j=1}^{d+1} l_{i_j} \\
&= \sum_{i_1, \ldots, i_{d+1} \in \mathcal{C}} \left(\prod_{j=2}^{d+1} p_{i_1 i_j} \right) \prod_{j=1}^{d+1} l_{i_j}.
\end{aligned}$$

We can now compute the expected number of $d-$stars in G:

$$\begin{aligned}
\mathbb{E}[X_d] &= n \binom{n-1}{d} \mathbb{P}(S) = n \binom{n-1}{d} \mathbb{P}(S_1)^k \\
&= n \binom{n-1}{d} \left[\sum_{i_1, \ldots, i_{d+1} \in \mathcal{C}} \left(\prod_{j=2}^{d+1} p_{i_1 i_j} \right) \prod_{j=1}^{d+1} l_{i_j} \right]^k.
\end{aligned}$$