

Tied Kronecker Product Graph Models to Capture Variance in Network Populations

Sebastian Moreno*, Sergey Kirshner⁺, Jennifer Neville^{**}, S.V.N. Vishwanathan⁺⁺

*Department of Computer Science, ⁺Department of Statistics

Purdue University

West Lafayette, IN, 47907 USA

Email: {smorenoa, skirshne, neville, vishy}@purdue.edu

Abstract—Much of the past work on mining and modeling networks has focused on understanding the observed properties of *single* example graphs. However, in many real-life applications it is important to characterize the structure of *populations* of graphs. In this work, we investigate the distributional properties of Kronecker product graph models (KPGMs) [1]. Specifically, we examine whether these models can represent the natural variability in graph properties observed *across* multiple networks and find surprisingly that they cannot. By considering KPGMs from a new viewpoint, we can show the reason for this lack of variance theoretically—which is primarily due to the generation of each edge independently from the others. Based on this understanding we propose a generalization of KPGMs that uses tied parameters to increase the variance of the model, while preserving the expectation. We then show experimentally, that our *mixed-KPGM* can adequately capture the natural variability across a population of networks.

I. INTRODUCTION

Graphs and networks are a natural data representation for analysis of a myriad of domains, ranging from systems analysis (e.g., the Internet) to bioinformatics (e.g., protein interactions) to psycho-social domains (e.g., online social networks). Due to the recent interest in small-world networks and scale-free graphs, there has been a great deal of research focused on developing generative models of graphs that can reproduce skewed degree distributions, short average path length and/or local clustering (see e.g., [2], [3], [4], [5]). The majority of this work has focused on procedural modeling techniques (see [6] for a good overview). As an example, the preferential attachment model of [4] is an incremental generative method, which repeatedly adds a node to the graph with k edges and each of the edges is linked up to existing nodes in the graph with probability proportional to its current degree.

There are relatively few statistical models of graph structure that represent probability distributions over graph structures, with parameters that can be *learned* from example networks. One method is the Exponential Random Graph Model (ERGM) (also known as p^* models) [7]. ERGMs represent probability distributions over graphs with an exponential linear model that uses feature counts of local graph properties considered relevant by social scientists (e.g., edges, triangles, paths). Another method is the Kronecker product graph model (KPGM) [8]. The KPGM is a fractal model, which starts from a small adjacency matrix with specified probabilities for all

pairwise edges, and uses repeated multiplication (of the matrix with itself) to grow the model to a larger size.

The aim, for much of the past work on generative graph models, has been to accurately capture the observed properties of a *single* graph—either global properties such as average path length or local graph properties such as transitive triangles. As such, evaluation of the proposed models has generally centered on empirical validation that observed graph properties match those of the generated graphs. Specifically, empirical analysis has consisted of visual comparison of properties of the input and a small number of generated graphs.

However, in many real-life applications one would like to model *populations* of graphs. That is, rather than capturing the properties of a single observed network, we would like to be able to capture the *range* of properties observed over multiple samples from a *distribution* of graphs. For example, in social network domains, the social processes that govern friendship formation are likely to be consistent across college students in various Facebook networks, so we expect that the networks will have similar structure, but with some random variation. Descriptive modeling of these networks should focus on acquiring an understanding of both their average characteristics and their expected variation. Similarly, when analyzing the performance of network protocols or network classification methods, we would like to be able measure performance across a set of network structures that capture the natural variability in these domains.

In recent work [9], we investigated the distributional properties of state-of-the art generative models for graphs. Specifically, we considered the case when more than one instance of a network is available, and examined whether these models capture the natural variability in graph properties observed *across* multiple networks. Our analysis showed that KPGMs [8] and ERGMs [7], [10] do not generate graphs with sufficient variation to capture the natural variability in two social network domains. What was particularly surprising is how little variance (compared to the real networks) was produced in the graphs generated from each model class. Each of the models appears to place most of the probability mass in the space of graphs on a relatively small subset of graphs with very similar characteristics.

Some theoretical insights to explain this phenomenon is available in the context of ERGMs. In recent work it was

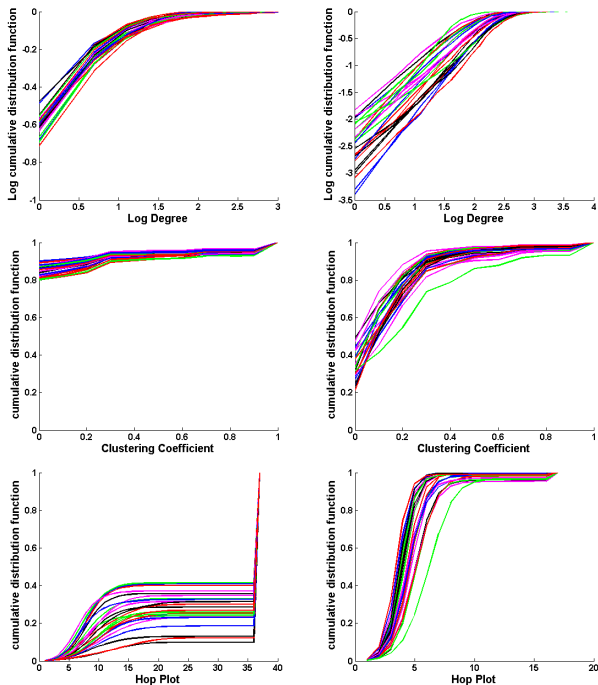


Fig. 1. Natural variability in the characteristics of a population of Facebook (left) and AddHealth (right) networks.

shown that learning ERGMs with only local features can lead to *degenerate* global models (i.e., the estimated distribution places most of its probability mass on either the empty or complete graphs) [11]. This indicates that the effect may be due to long-range dependencies in social networks that cannot be accurately captured by local features alone.

In this work, we investigate the issue in more detail for KPGM models. By considering KPGMs from a new viewpoint, we show the reason for this lack of variance theoretically—which is primarily due to the generation of each edge independently from the others. Based on this understanding we propose a generalization to KPGMs that uses tied parameters to increase the variance of the model, while preserving the expectation. We then show experimentally, that our *mixed-KPGM* can adequately capture the natural variability across a population of networks.

The rest of the paper is organized as follows. First, we describe the network data sets considered in the paper and examine their variability across several graph metrics (Section II). Next, we provide background information on KPGM models, examine whether KPGM are capable of capturing such variability (Section III). We consider the generative process for KPGMs from a slightly different angle which leads us to a variant of KPGM that allows higher variance and more clustering in sampled graphs (Section IV). We then apply our new approach to multiple instances of networks (Section V) and discuss our findings and contributions (Section VI).

II. NATURAL VARIABILITY OF REAL NETWORKS

We conducted a set of experiments to explore three distributional properties of graphs found in natural social network

populations: (1) degree, (2) clustering coefficient, and (3) path lengths. The degree of a node d_i is simply the number of nodes in the graph that are connected to node i . Degree is a local property of nodes, but since many networks have heavy-tailed degree distributions, the overall degree distribution is often considered a global property to match. Clustering coefficient is calculated for a node i as: $c_i = \frac{2|\delta_i|}{(d_i-1)d_i}$, where δ_i is the number of triangles in which the node i participates and d_i is the number of neighbors of node i . Clustering coefficient measures the local clustering in the graph. For path length, we consider the hop plot distribution in the graph, which refers to the number of nodes that can be reached with h “hops” in the graph: $N_h = \sum_v N_h(v)$, where $N_h(v)$ is the number of neighbors that are $\leq h$ edges away from node v in G . The hop plot measures the global connectivity of the graph.

Specifically, we investigated two different real-world social network datasets. The first set is drawn from the public Purdue Facebook network. Facebook is a popular online social network site with over 150 million members worldwide. We considered a set of over 50000 Facebook users belonging to the Purdue University network with its over 400000 wall links consisting of a year-long period. To estimate the variance of real-world networks, we sampled 25 networks, each of size 1024 nodes, from the wall graph. To construct each network, we sampled an initial timepoint uniformly at random, then collected edges temporally from that point (along with their incident nodes) until the node set consisted of 1024 nodes. In addition to this node and initial edge set, we collected all edges among the set of sampled nodes that occurred within a period of 60 days from the initially selected timepoint. (This increased the connectivity of the sampled networks). The characteristics of the set of sampled networks is graphed in Figure 1 (left), with each line corresponding to the cumulative distribution of a single network. The figures show the similarity among the sampled networks as well as the variability that can be found in real domains for networks of the same size.

The second dataset consists of a set of social networks from the National Longitudinal Study of Adolescent Health (AddHealth) [12]. The AddHealth dataset consists of survey information from 144 middle and high schools, collected (initially) in 1994-1995. The survey questions queried for the students’ social networks along with myriad behavioral and academic attributes, to study how social environment and behavior in adolescence are linked to health and achievement outcomes in young adulthood. In this work, we considered the social networks from 25 schools with sizes varying from 800 to 2000 nodes. The characteristic of these networks are showed in Figure 1 (right), where, despite the fact that the networks are of different size, since we compare cumulative distributions, the networks have very similar characteristics. From this data, we also use a specific subset of 6 networks for some initial experiments, with sizes between 1100-1600 nodes and a density in the range [0.004-0.005].

We consider these sets of networks to be illustrative examples of *populations* of graphs (i.e., drawn from the same distribution). Both sets are likely to be affected/generated by

similar social processes (high school friendships and undergrad communication patterns respectively). In addition, both sets exhibit a remarkable similarity in their graph structures, yet with some variation due random effects.

III. VARIABILITY OF KPGM MODELS

In this section we evaluate KPGMs both empirically and analytically to investigate whether graphs generated from learned KPGM models can capture the distributional properties we observe in real-world social networks.

A. Background: KPGMs

The Kronecker product graph model (KPGM) [8] is a fractal model, which uses a small adjacency matrix with Bernoulli probabilities to represent pairwise edges probabilities, and uses repeated multiplication (of the matrix with itself) to grow the model to a larger size. To generate a sample graph from the model, the algorithm independently samples each edge according to its associated probability of existence in the model. It has been shown empirically that this approach successfully preserves a wide range of global properties of interest, including degree distributions, eigenvalue distributions, and path-length distributions [1].

More specifically, the model generates self-similar graphs, in a recursive way using Kronecker multiplication. The algorithm starts with a initial matrix $\mathcal{P}_1 = \Theta$ with b rows and columns, where each cell value is a probability. Typically $b = 2$ or 3 , e.g.:

$$\mathcal{P}_1 = \Theta = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}$$

To generate graphs of a larger size, the Kronecker product of \mathcal{P}_1 is taken k times with itself to generate a matrix:

$$\mathcal{P}_k = \mathcal{P}_1^{[k]} = \mathcal{P}_1^{[k-1]} \otimes \mathcal{P}_1 = \underbrace{\mathcal{P}_1 \otimes \dots \otimes \mathcal{P}_1}_{k \text{ times}}$$

with b^k rows and columns. We will denote the (u, v) -th entry of \mathcal{P}_k by $\pi_{uv} = \mathcal{P}_k[u, v]$, $u, v = 1, \dots, b^k$. Under KPGM, a graph $G = (V, E)$ with $V = \{1, \dots, N\}$ where $N = b^k$, $k \in \mathbb{N}$, is sampled by performing mutually independent Bernoulli trials for each pair (u, v) with probability $\pi_{uv} = \mathcal{P}_k[u, v]$ and placing an edge (u, v) into E if the trial for (u, v) results in a success.

To estimate a KPGM from an observed graph G^* , the learning algorithm uses maximum likelihood estimation to determine the values of Θ that have the highest likelihood of generating G^* : $l(\Theta) = \log P(G^*|\Theta) = \log \sum_{\sigma} P(G^*|\Theta, \sigma)P(\sigma)$ where σ defines a permutation of rows and columns of the graph G^* . The model assumes that each edge is a Bernoulli random variable, given \mathcal{P}_1 . Therefore the likelihood of the observed graph $P(G^*|\Theta, \sigma)$ is calculated as:

$$P(G^*|\Theta, \sigma) = \prod_{(u,v) \in E} \mathcal{P}_k[\sigma_u, \sigma_v] \prod_{(u,v) \notin E} (1 - \mathcal{P}_k[\sigma_u, \sigma_v])$$

where σ_u refers to the permuted position of node u in σ .

With this formula, the algorithm uses a gradient descent approach to search for the MLE parameters $\hat{\Theta}$, however the gradient of $l(\theta)$ involves a summation over an exponential

number of permutations σ . To avoid this calculation the algorithm simulates draws from the permutation distribution $P(\sigma|G^*, \theta)$ until it converges. Then it calculates the expected values of $l(\theta)$ and the gradient. This sampling is performed with a Metropolis-Hastings algorithm. In every iteration of the algorithm, $l(\Theta)$ and its gradient are calculated T times, obtaining their corresponding mean. The parameters $\hat{\Theta}$ are updated with the following until convergence: $\hat{\Theta}_{t+1} = \hat{\Theta}_t + \lambda \frac{\partial l(\hat{\Theta})}{\partial \Theta_t}$.

B. Assessing Variability

To learn KPGM models, we selected a single network from each dataset to use as a training set. To control for variation in the samples, we selected the network that was closest to the median of the degree distributions in each dataset. In the Facebook data, we selected the first generated network with 2024 edges that was very close to the median of the degree, clustering coefficient and hop plot distributions. In the AddHealth data, we selected the network from school 117, with 1278 nodes and 7982 edges. This network is close to median of the degree and hop plot distribution but has a little higher clustering coefficient than the median of the 25 networks and the highest clustering of the subset of six network samples.

Using each selected network as a training set, we learned a KPGM model ($b = 2$) using ML estimation to estimate $\hat{\Theta}$. For each learned model, we generated 200 sample graphs. The KPGM graphs were generated using the estimated matrix $\mathcal{P}_k = \mathcal{P}_1^{[k]}$. Since the number of rows/columns of \mathcal{P}_k is generally larger than the target network size ($b^k \geq N$), we generate a graph G with b^k nodes and then simply drop the last $b^k - N$ nodes and the adjoining edges from G . From the 200 samples we estimated the empirical sampling distributions for degree, clustering coefficient, and hop plots.

The results are plotted in Figures 5 and 6 in Section V. For both the original datasets and the KPGM generated data, we plot the median and interquartile range for the set of observed network distributions. Solid lines correspond to the median of the distributions; dashed lines: the 25th and 75th percentiles.

The results in Figures 5 and 6 show two things. First, the KPGM performs as expected, capturing the graph properties that have been reported in the past. Specifically, the KPGM model is able to capture the degree and hop plot (i.e., long range connectivity) fairly well in both datasets, but it is not able to model the local clustering in either. Second, the KPGM model does not reproduce the amount of variance exhibited in the real networks. Moreover, it is surprising that the variance of the generated graphs is so slight that it is almost not apparent in some of the plots. (Recall that dashed lines indicate the 25th and 75th percentiles.) The lack of variance implies that while the KPGM model may be able to reasonably capture the patterns of the input graph (i.e., match on the means of the distributions), it cannot be used to generate multiple “similar” graphs—since it appears that the generated graphs are nearly isomorphic.

C. Increasing Variability

One obvious possibility that could explain the low variance in KPGMs is the small number of model parameters used in the initiator matrix. Recall, that the reported experiments use 2×2 initiator matrices. To investigate the change in variance for models with initiator matrices of varying sizes, we conducted the following simulation experiment. We first manually specified a 2×2 model: $\Theta_{2 \times 2} = \begin{bmatrix} 0.95 & 0.60 \\ 0.60 & 0.20 \end{bmatrix}$. Then to create a 4×4 matrix that would produce graphs similar to the 2×2 model, we computed the Kronecker product of the 2×2 matrix and then perturbed the parameter in each cell by adding a random number $\sim \mathcal{N}(0, 9E - 4)$.

From these specified matrices, we generated 100 networks of size 1024 and measured the variance in the graph distributions. Unfortunately, the variance does not show any noticeable increase. We also tried learning KPGM models of the real datasets (e.g., AddHealth) with initiator matrices up to size 6×6 with no noticeable increase in variance. This indicates that we are unlikely to achieve higher variance by increasing the parameterization of the model. Although larger initiator matrices (e.g., 30×30) would increase the number of model parameters, it would also decrease the number of Kronecker products needed to generate a graph of a particular size, which may impact the model's ability to represent fractal structure.

Based on these investigations, we conjecture that it is KPGM's use of independent edge probabilities and fractal expansion that leads to the small variation in generated graphs. We explore this issue analytically next.

D. Theoretical Analysis

It is possible to carry out a theoretical analysis for the variance in the number of edges in graphs sampled from KPGM, see Appendix A.

Denote by E_k the number of edges in a graph G with $N = b^k$ nodes randomly sampled according to KPGM with the initiator matrix \mathcal{P}_1 . Denote by θ_{ij} the entry in the i -th row and j -th column of \mathcal{P}_1 . Let $S = \sum_{i=1}^b \sum_{j=1}^b \theta_{ij}$ and $S_2 = \sum_{i=1}^b \sum_{j=1}^b \theta_{ij}^2$ be the sum of the entries and the squares of the entries, respectively, of the initiator matrix \mathcal{P}_1 . Then

$$E[E_k] = S^k \text{ and } Var(E_k) = S^k - S_2^k. \quad (1)$$

Note that $Var(E_k) \leq E(E_k)$ and $SD(E_k) \leq \sqrt{E(E_k)}$. However, in the real-world networks considered in this paper, the estimated variance significantly exceeds the mean—in Facebook the estimated mean number of edges is 1991, while the variance is 23451. Similarly, in AddHealth the mean number of edges is 8021, while the variance is 9045070. This indicates that KPGM models are incapable of reproducing the variance of these real-world network populations.

IV. EXTENDING KPGM TO INCREASE VARIANCE

In this section, we propose a generalization of KPGM that permits larger variance in the properties of the generated graphs by introducing edge dependence in the generation process.

A. Another View of Graph Generation with KPGMs

Before introducing our model variant, we present a slightly different view of the graph generation under KPGM. This viewpoint provides an extra dimension to the model that if exploited allows a natural way to couple the edge generation and thus to increase the variance of the graph statistics.

Consider the graph generation, or *realization* process. Given a matrix of edge probabilities $\mathcal{P}_k = \mathcal{P}_1^{[k]}$, a graph G with adjacency matrix $E = R(\mathcal{P}_k)$ is *realized* (sampled or generated) by setting $E_{uv} = 1$ with probability $\pi_{uv} = \mathcal{P}_k[u, v]$ and setting $E_{uv} = 0$ with probability $1 - \pi_{uv}$. E_{uv} s are realized through a set of Bernoulli trials or binary random variables (e.g., $\pi_{uv} = \theta_{11}\theta_{12}\theta_{11}$). For example, in Figure 2a, we illustrate the process of a KPGM generation for $k = 3$ to highlight the multi-scale nature of the model. Each level correspond to a set of separate trials, with the colors representing the different parameterized Bernoullis (e.g., θ_{11}). For each cell in the matrix, we sample from three Bernoullis and then based on the set of outcomes the edge is either realized (black cell) or not (white cell).

To formalize this, we start with a description of probabilities in the stochastic Kronecker matrix \mathcal{P}_k . Assume $N = b^k$ and index the entries of the initiator matrix \mathcal{P}_1 with (i, j) , $i, j = 1, \dots, b$. For convenience, we will label the nodes $\{0, \dots, N - 1\}$ instead of $\{1, \dots, N\}$. Let $(v_1 \dots v_k)_b$ be a representation of a number v in base b . We will refer to it as a b -nary representation for v and will refer to v_l as the l -th b -it of v . Each $v \in \{0, N - 1\}$ has a unique representation in base b , $v = \sum_{l=1}^k (v_l - 1)b^{k-l}$ with each $v_l \in \{1, \dots, b\}$.

As was pointed out in [13] for $b = 2$, and mentioned in [1], for $u, v \in \{0, \dots, N - 1\}$ with b -nary representations $u = (u_1 \dots u_k)_b$ and $v = (v_1 \dots v_k)_b$,

$$\pi_{uv} = \mathcal{P}_k[u, v] = \prod_{l=1}^k \mathcal{P}_1[u_l, v_l] = \prod_{l=1}^k \theta_{u_l v_l}. \quad (2)$$

This description highlights the multiscale nature of KPGM. The probability of having an edge (u, v) in a graph realization from \mathcal{P}_k is equal to the product of contributions (probabilities $\theta_{u_l v_l} = \mathcal{P}_1[u_l, v_l]$) from different scales (l).

Alternatively, each E_{uv} can be thought of as drawn in k stages, one for each b -it of u and v . Let E_{uv}^l be a binary random variable with $P(E_{uv}^l = 1) = \theta_{u_l v_l}$ and $P(E_{uv}^l = 0) = 1 - \theta_{u_l v_l}$. Then $I_{uv} = \prod_{l=1}^k E_{uv}^l$ or in other words, an edge (u, v) is included if and only if the trials E_{uv}^l resulted in a success for *all* scales $l = 1, \dots, K$. Equivalently, an l -th scale adjacency matrix $E^l = (E_{uv}^l)$ is realized from $(\mathcal{P}_k)_l = \underbrace{\mathbf{1}_b \otimes \dots \otimes \mathbf{1}_b}_{l-1} \otimes \mathcal{P}_1 \otimes \underbrace{\mathbf{1}_b \otimes \dots \otimes \mathbf{1}_b}_{k-l}$ where $\mathbf{1}_b$ is a $b \times b$ matrix of ones. An adjacency matrix $E = E^1 \circ \dots \circ E^k$ is an entriwise (Hadamard) product of the adjacency matrices at k scales. See illustration in Fig 2(a).

Note that each matrix $(\mathcal{P}_k)_l$ consists only of the values of the initiator matrix \mathcal{P}_1 . Each of these values is repeated $b^{k-1} \times b^{k-1}$ and is contained in the intersection of b^{k-1} rows and columns, with the value θ_{ij} appearing in rows u with

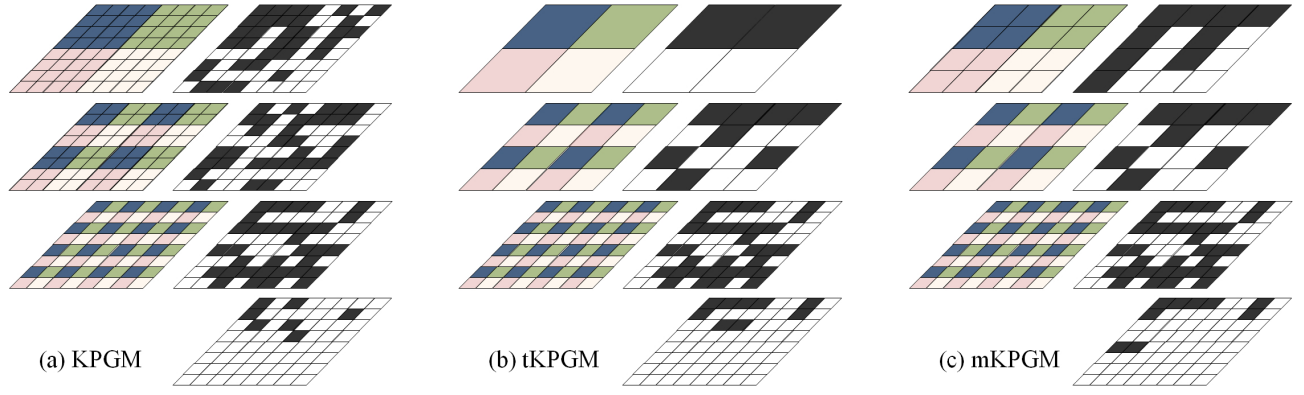


Fig. 2. Generative mechanisms for different Kronecker product graph models: (a) KPGM, (b) tKPGM, (c) mKPGM. Probability matrices are on the left, realizations are on the right. The bottom left matrices are the final sample from each model. For KPGM, $b^k \times b^k$ independent Bernoulli trials are performed at each level. For tKPGM, only $b^l \times b^l$ independent Bernoulli trials are performed at a level l ; the result of each trial is then replicated for all $b^{k-l} \times b^{k-l}$ entries in the corresponding submatrix. For mKPGM, the first l levels ($l = 2$ in the figure) are untied.

$u_l = i$ and columns v with $v_l = j$. Because of the Kronecker product structure of $(\mathcal{P}_k)_l$, pairs (u, v) corresponding to the same probability values appear in blocks of $b^{k-l} \times b^{k-l}$. It is important to note that even though many of E_{uv}^l have the same probability distribution, under KPGM they are all sampled independently of one another. Relaxing this assumption will lead to extension of KPGMs capable of capture the variability of real-world graphs.

B. Tied KPGM

In this section, we consider a variant of KPGM that preserves the marginal probabilities of edges in a given location but does *not* treat them as independent. Our approach increases the variance in the number of edges by introducing positive covariance between the indicator variables E_{uv} for the presence of edges.

Even though the probability matrix \mathcal{P}_k exhibits hierarchical or multiscale structure, this hierarchy is not explicit in the graphs realized from KPGM because all of the trials at all scales are performed *independently*, or in other words, all of the edges are untied at all scales. We propose a model where the trials have a hierarchical structure as well, leading to a higher grouping of edges and a higher variance in the number of edges. In this model, the edges are tied at *all* common scales. For the KPGMs, a realization E is obtained from the edge probability matrix \mathcal{P}_k , $E = R(\mathcal{P}_k) = R(\mathcal{P}_1 \otimes \dots \otimes \mathcal{P}_1)$. Instead, we propose to realize an adjacency matrix *after each Kronecker multiplication*. We denote by $R_t(\mathcal{P}_1, k)$ a realization of this new model with the initiator \mathcal{P}_1 and k scales. We define R_t recursively, $R_t(\mathcal{P}_1, 1) = R(\mathcal{P}_1)$, and $R_t(\mathcal{P}_1, k) = R_t(R_t(\mathcal{P}_1, k-1) \otimes \mathcal{P}_1)$. If unrolled,

$$E = R_t(\mathcal{P}_1, k) = \underbrace{R(\dots R(R(\mathcal{P}_1) \otimes \mathcal{P}_1) \dots)}_{k \text{ realizations } R}.$$

Similar to section IV-A, we define the probability matrix for scale l , $(\mathcal{P}_k)_l = \mathcal{P}_1$ for $l = 1$, and $(\mathcal{P}_k)_l = R_t((\mathcal{P}_k)_{l-1}) \otimes \mathcal{P}_1$ for $l \geq 2$. Under this model, at scale l there are $b^l \times b^l$ independent Bernoulli trials rather than

$b^k \times b^k$ as $(\mathcal{P}_k)_l$ is a $b^l \times b^l$ matrix. These $b^l \times b^l$ trials correspond to different *prefixes* of length l for (u, v) , with a prefix of length l covering scales $1, \dots, l$. Denote these trials by $T_{u_1 \dots u_l, v_1 \dots v_l}^l$ for the entry (u', v') of $(\mathcal{P}_k)_l$, $u' = (u_1 \dots u_l)_b$, $v' = (v_1 \dots v_l)_b$. The set of all independent trials is then $T_{1,1}^1, T_{1,2}^1, \dots, T_{b,b}^1, T_{11,11}^2, \dots, T_{bb,bb}^2, \dots, T_{\underbrace{1 \dots 1}_k, \underbrace{1 \dots 1}_k}^k, \dots, T_{\underbrace{b \dots b}_k, \underbrace{b \dots b}_k}^k$. The probability of a success for a Bernoulli trial at a scale l is determined by the entry of the \mathcal{P}_1 corresponding to the l -th bits of u and v :

$$P(T_{u_1 \dots u_l, v_1 \dots v_l}^l) = \theta_{u_l v_l}.$$

One can construct E^l , a realization of a matrix of probabilities at scale l , from a $b^l \times b^l$ matrix T by setting $E_{uv}^l = T_{u_1 \dots u_l, v_1 \dots v_l}^l$ where $u = (u_1 \dots u_k)_b$, $v = (v_1 \dots v_k)_b$. The probability for an edge appearing in the graph is the same as under KPGM as

$$E_{uv} = \prod_{l=1}^k E_{uv}^l = \prod_{l=1}^k T_{u_1 \dots u_l, v_1 \dots v_l}^l = \prod_{l=1}^k \theta_{u_l v_l}.$$

Note that all of the pairs (u, v) that start with the same prefixes $(u_1 \dots u_l)$ in b -nary also share the same probabilities for E_{uv}^l , $l = 1, \dots, l$. Under the proposed models trials for a given scale t are shared or tied for the same value of a given prefix. We thus refer to our proposed model as *tied KPGM* or *tKPGM* for short. See Figure 2(b) for an illustration.

Just as with KPGM, we can find the expected value and the variance of the number of edges under tKPGM. Since the marginal probabilities for edges (u, v) are the same as under KPGM, the expected value for the number of edges is unchanged, $E[E_k] = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} E[E_{uv}] = S^k$. The variance $Var(E_k)$ can be derived recursively by conditioning on the trials with prefix of length $l = 1$:

$$Var(E_k) = S \times Var(E_{k-1}) + (S - S_2) S^{2(k-1)}, \quad (3)$$

with $Var(E_1) = S - S_2$. The solution to this recursion is

$$Var(E_k) = S^{k-1} (S^k - 1) \frac{S - S_2}{S - 1}. \quad (4)$$

Note that under this model, all edges sharing a prefix are no longer independent as they either do not appear together (if either of the trials corresponding to the prefix resulted in a failure), or have a higher probability of appearing together (if all of the trials in the prefix are successful). In the case of success at all l scales for a prefix of length l , the expected number of edges for the possible $b^{k-l} \times b^{k-l}$ pairs with this prefix is S^{k-l} . The resulting proportion of edges to pairs is then $(S/b^2)^{k-l}$ higher than $(S/b^2)^k$ for all possible $b^k \times b^k$ pairs ($S = \sum_{i=1}^b \sum_{j=1}^b \theta_{ij} \leq b^2$). This suggests that the resulting graph will consist of several groups of nodes connected by many edges rather than edges spread among the nodes in the graph.

C. Mixed KPGM

Even though tKPGM provides a natural mechanism for clustering the edges and for increasing the variance in the graph statistics, the resulting graphs exhibit *too much* variance (see Section V). One of the possible reasons is that the edge clustering and the corresponding Bernoulli trial tying should not begin at the highest shared scale (to model real-world networks). To account for this, we introduce a modification to tKPGM that ties the trials starting with prefix of length $l + 1$, and leaves the first l scales untied. Since this model will combine or *mix* the KPGM with tKPGM, we refer to it as *mKPGM*. Note that mKPGM is a generalization of both KPGM ($l \geq k$) and tKPGM ($l = 1$). The effect of tying can be seen in Figure 3—the graph sampled from KPGM exhibits little grouping of the edges, the graph sampled from tKPGM exhibits strong grouping, and the graph sampled from mKPGM falls in between the other two. How close would the properties of a graph from mKPGM resemble one of the other two depends on the proportion of untied scales.

Formally, we can define the generative mechanism in terms of realizations. Denote by $R_m(\mathcal{P}_1, k, l)$ a realization of *mKPGM* with the initiator \mathcal{P}_1 , k scales in total, and l untied scales. Then $R_m(\mathcal{P}_1, k, l)$ can be defined recursively as $R_m(\mathcal{P}_1, k, l) = R(\mathcal{P}_k)$ if $k \leq l$, and $R_m(\mathcal{P}_1, k, l) = R_t(R_m(\mathcal{P}_1, k - 1, l) \otimes \mathcal{P}_1)$ if $k > l$. Scales $1, \dots, l$ will require $b^l \times b^l$ Bernoulli trials each, while a scale $s \in \{l + 1, \dots, k\}$ will require $b^s \times b^s$ trials. See Figure 2(c) for an illustration.

Intuitively, the graph sampling mechanism under mKPGM can be viewed as generating a binary $b^l \times b^l$ matrix according to KPGM with $\mathcal{P}_l = \mathcal{P}_1^{[l]}$, and then for each success (1 in the matrix) generating a $b^{k-l} \times b^{k-l}$ matrix according to tKPGM with initiator \mathcal{P}_1 and $k-l$ scales. Failure (0) in the intermediate matrix results in a $b^{k-l} \times b^{k-l}$ matrix of zeros. These $b^{k-l} \times b^{k-l}$ then serve as submatrices of the realized $b^k \times b^k$ adjacency matrix.

Since the marginal probabilities for edges are unchanged, $P(E_{uv}) = \pi_{uv} = \prod_{l=1}^k \theta_{u_l v_l}$, the expected value for the number of edges is unchanged as well, $E[E_k] = S^k$. However, the variance expression is different from that in (4), and it can be obtained conditioning on the Bernoulli trials of the l highest

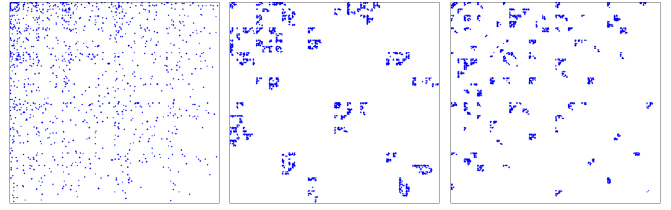


Fig. 3. Generated networks of 2^8 nodes for different Kronecker product graph models: KPGM (left), tKPGM (center), mKPGM (right). For mKPGM the number of untied scale was 5.

order scales:

$$\text{Var}(E_k) = S^{k-1} (S^{k-l} - 1) \frac{S - S_2}{S - 1} + (S^l - S_2^l) S^{2(k-l)}. \quad (5)$$

Note that for $l = 1$, the variance is equal to $S^{k-1} (S^k - 1) \frac{S - S_2}{S - 1}$, the same as for tKPGM, and the variance of mKPGM is smaller than that of tKPGM for $l > 1$. When $l = k$, the variance is equal to $S^k - S_2^k$, the same as for KPGM, and the variance of mKPGM is greater than that of KPGM for $l < k$.

D. Experimental Evaluation

We perform a short empirical analysis of mKPGMs using simulations. Figure 4 shows the variability over four different graph characteristics, calculated over 300 sampled networks of size 2^{10} with $\Theta = \begin{bmatrix} 0.99 & 0.20 \\ 0.20 & 0.77 \end{bmatrix}$, for $l = \{1, \dots, 10\}$. In each plot, the solid line represents the mean of the analysis (median for plot (b)) while the dashed line correspond to the mean plus/minus one standard deviation (first and third quartile for plot(b)).

In Figure 4(a), we can see that the total number of edges does not change significantly with the value of l , however the variance of this characteristic decreases for higher values of l , confirming that the KPGM ($l = 10$) has the lowest variance of all. Figure 4(b) shows how the median degree of a node increases proportionally to the value of l . Also, considering that the number of nodes in the network remains constant for all values of l , it is clear that as l increases the edges are assigned more uniformly throughout the nodes compared to the tKPGM ($l = 1$)—where some nodes get the majority of the edges.

This uniform distribution of the edges in KPGM generates large chain of nodes with few connections among them, leading to a small clustering coefficient and a large diameter (Fig 4c-d). On the other hand, the large connected sets of nodes generated by tKPGM produces a higher clustering coefficient among these groups with a small diameter (plots (c) and (d)). Finally as l increases, the mKPGM ($1 \leq l \leq 10$) will generate a larger number of (smaller) connected sets of nodes, leading lower clustering coefficients and larger diameters. These effects can be observed in the generated network presented in Figure 3 as well as Figure 4c-d.

V. EXPERIMENTS

To assess the performance of the tied KPGM (tKPGM) and mixed KPGM (mKPGM), we repeated the experiments

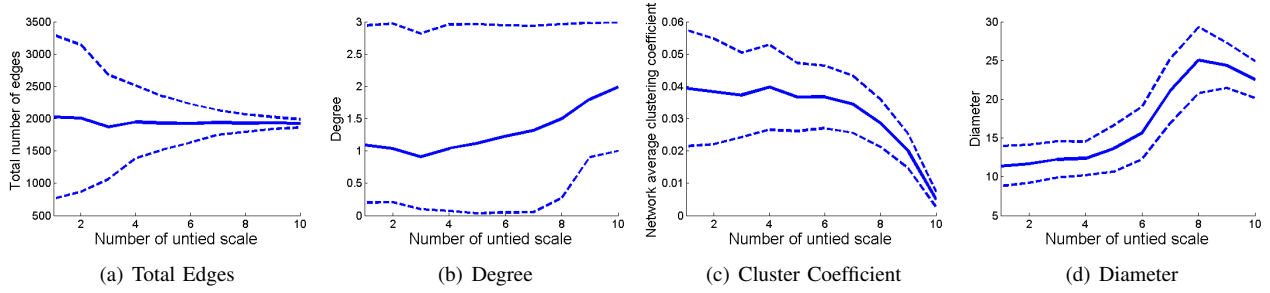


Fig. 4. Mean value (solid) \pm one sd (dashed) of characteristics of graphs sampled from mKPGM as a function of l , number of untied scales.

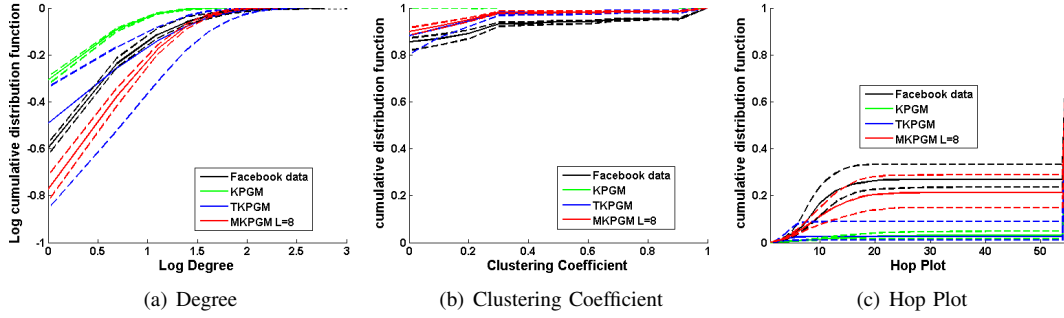


Fig. 5. Variation of graph properties in generated Facebook networks.

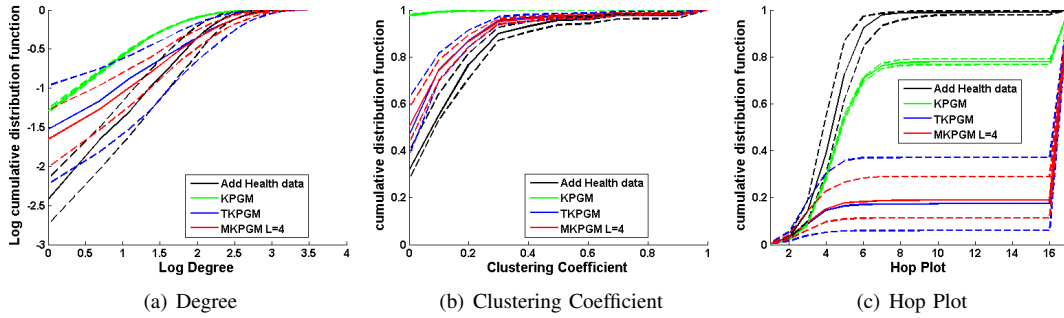


Fig. 6. Variation of graph properties in generated AddHealth networks.

described in Section III. We compared the two new methods to the original KPGM model, which uses MLE to learn model parameters $\hat{\Theta}_{MLE}$.

Although we outlined the representation and sampling mechanisms for tKPGM and mKPGM in Section IV, we do not yet have an automatic algorithm for estimating their parameters from data. Here we are more interested in the properties of the model(s) and whether they can accurately capture the variance of multiple network samples. We leave the development of efficient learning algorithms for future work.

To initialize the tKPGM and mKPGM models, we searched for a set of parameters Θ and L that reasonably matched to our example datasets. To achieve this, we first considered an exhaustive search of the set of possible parameter values for Θ and calculated their expected number of edges. We considered any parameters that matched the average number of edges in sample data ($\pm 1\%$). Within this set, we considered parameters in decreasing order of Θ_{11} (along with all possible values of L) and searched until we found a set that were a close match for the mean and variance of the degree distribution. Given the

best match, we used the same Θ values for both the tKPGM and the mKPGM models.

For the Facebook networks, the average number of edges is 1991. We selected $\Theta = \begin{bmatrix} 0.98 & 0.14 \\ 0.14 & 0.94 \end{bmatrix}$ with $E[E_k] = 1975.2$ and $L = 8$. For the AddHealth networks, the average number of edges is 8021. We selected $\Theta = \begin{bmatrix} 0.95 & 0.26 \\ 0.26 & 0.96 \end{bmatrix}$ with $E[E_k] = 8010.1$ and $L = 4$. The corresponding MLE parameters estimated by the KPGM model were $\hat{\Theta}_{MLE} = \begin{bmatrix} 0.66 & 0.25 \\ 0.25 & 0.84 \end{bmatrix}$ and $\hat{\Theta}_{MLE} = \begin{bmatrix} 0.93 & 0.43 \\ 0.43 & 0.47 \end{bmatrix}$ respectively.

For each of the methods we generated 200 sample graphs from the specified model. In Figures 5-6, we plot the median and interquartile range for the generated graphs, comparing to the empirical sampling distributions for degree, clustering coefficient, and hop plot to the observed variance of original data sets. As we discussed earlier, the KPGM graphs show almost no variance. The tKPGM graphs show the highest variance among all the models, with even more variance than the real data, so the tKPGM approach is equally inadequate for representing the natural variability in the domain. However,

with the mKPGM models, we are able to determine a value for L which produces graphs that reasonable the variance of real networks. With the exception of the hop plot distribution on the AddHealth data, the network generated from the specified mKPGM models match well on both the means and variances of the distributions. This demonstrates the potential for the mixed model to better represent network properties found in *populations* of networks, particularly those with skewed degree distributions (e.g., Facebook).

The key issue for using mKPGMs will be to select an appropriate set of parameters that both match the expected properties of a particular domain, and also reflect the natural variability. Our current approach of using exhaustive search, with a filter based on expected edge counts, is feasible to use in practice only if one has a sample of networks to evaluate the parameter settings. When only one network is available (e.g., a single Facebook network), the level of variance (i.e., L) will need to be selected based on domain understanding.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we investigated whether the state-of-the-art generative models for large-scale networks are able to reproduce the properties of multiple instances of real-world networks generated by the same source. Surprisingly KPGMs, one of the most commonly used models, produces very little variance in the simulated graphs, significantly less than that observed in real data. To explain this effect we showed analytically that KPGMs cannot capture the variance in the number of edges that we observe in real network populations.

Part of the problem with the lack of variance is that the edges under KPGM are drawn independently. We proposed a generalization to KPGM, *mixed-KPGM*, that introduces dependence in the edge generation process by performing the Bernoulli trials determining whether to add edges in a hierarchy. By choosing the level where the hierarchy begins, one can tune the amount that edges are grouped in a sampled graph. In our experiments with the multiple instance of networks from AddHealth and Facebook data sets, the model provides a good fit and importantly reproduces the observed variance in network characteristics.

In the future, we will investigate further the statistical properties of our proposed model. Among the issues, the most pressing is a systematic parameter estimation, a problem we are currently studying.

ACKNOWLEDGMENT

This research is supported by NSF and ARO under contract number(s) IIS-0916686 and W911NF-08-1-0238. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of ARO, NSF or the U.S. Government.

REFERENCES

[1] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 985–1042, 2010.

[2] O. Frank and D. Strauss, "Markov graphs," *Journal of the American Statistical Association*, vol. 81:395, pp. 832–842, 1986.

[3] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–42, 1998.

[4] A. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[5] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, "Stochastic models for the web graph," in *Proceedings of the 42st Annual IEEE Symposium on the Foundations of Computer Science*, 2000.

[6] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[7] S. Wasserman and P. E. Pattison, "Logit models and logistic regression for social networks: I. An introduction to Markov graphs and p^* ," *Psychometrika*, vol. 61, pp. 401–425, 1996.

[8] J. Leskovec and C. Faloutsos, "Scalable modeling of real graphs using Kronecker multiplication," in *Proceedings of the International Conference on Machine Learning*, 2007.

[9] S. Moreno and J. Neville, "An investigation of the distributional characteristics of generative graph models," in *Proceedings of the The 1st Workshop on Information in Networks*, 2009.

[10] G. Robins, T. Snijders, P. Wang, M. Handcock, and P. Pattison, "Recent developments in exponential random graph (p^*) models for social networks," *Social Networks*, vol. 29, pp. 192–215, 2006.

[11] M. S. Handcock, "Assessing degeneracy in statistical models of social networks," Center for Statistics and the Social Sciences, University of Washington, Working Paper 39, 2003.

[12] K. Harris, "The National Longitudinal Study of Adolescent health (Add Health), Waves I & II, 1994-1996; Wave III, 2001-2002 [machine-readable data file and documentation]," *Chapel Hill, NC: Carolina Population Center, University of North Carolina at Chapel Hill.*, 2008.

[13] M. Mahdian and Y. Xu, "Stochastic Kronecker graphs," in *5th International WAW Workshop*, 2007, pp. 179–186.

APPENDIX

A. Expectation and Variance for KPGM's Number of Edges

Let E_k denote the random variable for the number of edges in a graph generated from a KPGM with k scales and a $b \times b$ initiator matrix \mathcal{P}_1 . Then

$$\begin{aligned} E[E_k] &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} E[E_{uv}] = \sum_{i_1=1}^b \sum_{j_1=1}^b \cdots \sum_{i_k=1}^b \sum_{j_k=1}^b \prod_{l=1}^k \theta_{u_i, v_j} \\ &= \sum_{i_1=1}^b \sum_{j_1=1}^b \theta_{i_1 j_1} \cdots \sum_{i_k=1}^b \sum_{j_k=1}^b \theta_{i_k j_k} = \left[\sum_{i=1}^b \sum_{j=1}^b \theta_{ij} \right]^k \\ &= S^k \end{aligned}$$

with $S = \sum_{i=1}^b \sum_{j=1}^b \theta_{ij}$ is the sum of entries in the initiator matrix \mathcal{P}_1 , a result mentioned in [8], [13]. We can also find the variance $Var(E)$. Since E_{uv} s are independent,

$$\begin{aligned} Var(E_k) &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} Var(E_{uv}) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \pi_{uv} (1 - \pi_{uv}) \\ &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \pi_{uv} - \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \pi_{uv}^2 \\ &= \left[\sum_{i=1}^b \sum_{j=1}^b \theta_{uv} \right]^k - \left[\sum_{i=1}^b \sum_{j=1}^b \theta_{uv}^2 \right]^k = S^k - S_2^k \end{aligned}$$

where $S_2 = \sum_{i=1}^b \sum_{j=1}^b \theta_{ij}^2$ is the sum of the squares of the entries in the initiator matrix \mathcal{P}_1 .