Block Kronecker Product Graph Model

S. Moreno
Purdue Unversity
101 N Grant St #110
West Lafayette, 47906
smorenoa@cs.purdue.edu

P. Robles
Purdue Unversity
101 N Grant St #110
West Lafayette, 47906
problesg@cs.purdue.edu

J. Neville
Purdue Unversity
101 N Grant St #110
West Lafayette, 47906
neville@cs.purdue.edu

ABSTRACT

Since Kronecker Product Graph Model (KPGM) was introduced, it has been widely used to model real networks. The characteristics of the model specially its single fractal structure have made KPGM one of the most important algorithm of the last years. However, the utilization of a single fractal structure decreases the potential of KPGM by limiting the graph space covered by the model. In this paper, we propose a new generalization of KPGM, called block-KPGM. This new model expands the graph space covered for KPGM utilizing multiple fractal structures, generating networks with characteristics not generated by previous Kronecker models. We evaluate the block-KPGM by comparing it against two types of Kronecker models. We compare the cumulative distribution functions over three characteristics to show that block-KPGMs are able to produce networks that more closely match real-world graphs, reducing the multidimensional Kolgomorov-Smirnov distance between 11% to 63%.

Keywords

Statistical graph models, Kronecker models, block-KPGM, fractal structure.

1. INTRODUCTION

Since Kronecker Product Graph Model (KPGM) was introduced in 2005-2007 [5, 7], it has been widely used to model different type of networks. KPGM is intuitively appealing for its small number parameters, elegant fractal structure, fast sampling algorithms (i.e., $O(|\mathbf{E}|)$), and its parallelism. These characteristics have positioned KPGMs as one of the most important generative models in last years, being even chosen to create graphs for the Graph500 supercomputer benchmark[1].

This popularity has led to several empirical and mathematical studies of their properties. For example, the degree distribution has often been claimed to be power-law [6], with some lognormal characteristics [4], or it can be expressed as a mixture of normal distributions [2]. Moreover, recently it has been demonstrated that it is most accurately characterized as fluctuating between a lognormal distribution and an exponential tail [11]. Other studies also include the relation of core number with the initial parameters [11] and the relation of the graph properties with the original parameters [8]. However, most important properties are given by its elegant stationary fractal structure, generated by the Kronecker product of a set of parameters with itself, that generates networks with heavy-tailed distributions for indegree, out-degree, eigenvalues, and eigenvectors [5]. These characteristics are among the most important network properties, allowing KPGM to match several real graphs.

Unfortunately, although probabilistic in nature, the utilization of a single fractal structure reduces the potential of KPGM, limiting the graph space covered by KPGM. Even though the single fractal structure allows to generate matrices of probabilities that are used to sample different adjacency matrices, networks that do not have this single fractal structure cannot be modeled by KPGM. For this reason, we analyzed a way to take advantage of the KPGM strengths and relax its single fractal structure. The idea is to model different parts of the graph using multiple KPGMs, then we replicate the entire network by attaching the blocks generated by each of the previous learned KPMGs. This does not necessarily imply that graphs are connected through other graphs since a block represents edges that are in different parts of the network. Moreover, the use of multiple fractal structure will allow us to model real world networks that show complex patterns and cannot be model by a single KPGM structure. For example, in [1], it is described that complex networks, such as the world wide web, consist in multiple fractal structure connected through a skeleton.

In this work, we propose a new generalization of KPGM to capture the non stationarity fractal structure observed in some real world networks, the block Kronecker Product Graph Model (block-KPGM). block-KPGM expands the graph space covered for KPGM by introducing multiple fractal structures to model a network. Specifically, block-KPGM utilizes multiple independent KPGMs to model different parts of a network, which are joined to model the entire graph. Thanks to the multiple fractal structures, block-KPGM generates networks with characteristics that previous Kronecker models cannot generate. For example, our initial experiments show that block-KPGM increases the clustering coefficient of the generated networks and reduces the number of isolated nodes.

The remainder of the paper is organized as follows. Sec-

[1] http://www.graph500.org/Specifications. html

Eleventh Workshop on Mining and Learning with Graphs. Chicago, Illinois, USA

tion 2 describes KPGM and mixed KPGM algorithm (another KPGM generalization [10]). Section 3 explains the new block-KPGM with its respective algorithm. Section 4 shows an empirical comparison and modeling of real networks. Section 5 has the conclusions and future works of this new model.

2. BACKGROUND

This section describes the Kronecker Product Graph Model (KPGM) and its training algorithm based on maximum likelihood estimation (MLE) [7]. We also describe the mixed Kronecker Product Graph Model (mKPGM) and its training algorithm based on simulated method of moments (SMM) [10].

2.1 Kronecker Product Graph Model

Let Θ be a $b \times b$ *initiator matrix* of parameters, where $\forall i,j \;\; \theta_{ij} \in [0,1]$. Then the KPGM algorithm generates a graph $G_K = (\mathbf{V}_K, \mathbf{E}_K)$, where $\mathbf{V}_K$ and $\mathbf{E}_K$ are the set of nodes and edges respectively, as follows. First, the model computes the K^{th} Kronecker power of the initiator matrix Θ via $K-1$ Kronecker products of Θ with itself. This produces a $b^K \times b^K$ matrix P_K, where $P_K(i,j)$ represents the probability of an edge existing between nodes i and j. P_K is used to generate a graph G_K with $|\mathbf{V}_K| = b^K$, by sampling each edge independently from a Bernoulli($P_K(i,j)$) distribution (i.e., if the trial is successful, the edge e_{ij} is added to $\mathbf{E}_K$).

Given an observed training network $G^\star = (\mathbf{V}^*, \mathbf{E}^*)$, the MLE learning algorithm finds the parameters Θ that maximizes the likelihood of the observed graph given a permutation (σ) of the rows and columns of the adjacency matrix [7]. The KPGM likelihood of the graph is:

$$P(G^\star|\Theta,\sigma) = \prod_{(i,j)\in\mathbf{E}^*} P_K(\sigma_i,\sigma_j) \prod_{(i,j)\notin\mathbf{E}^*} (1 - P_K(\sigma_i,\sigma_j)) \quad (1)$$

Here σ_i denotes the new position of node i according to the permutation σ. In practice, the true permutation is unknown and the learning algorithm uses a Metropolis-Hastings sampling approach to search over the factorial number of possible permutations of the network. The algorithm then uses a gradient descent approach to update the parameters Θ, where the derivative of the likelihood is approximated given the current σ and Θ.

2.2 Mixed Kronecker Product Graph Model

The mKPGM is a generalization of the KPGM, which uses *parameter tying* to capture the clustering and natural variation observed in real-world networks more accurately [10]. The marginal probabilities of edges in P_K are preserved but the edge probabilities are no longer independent.

Specifically, given Θ, K, and a parameter $\ell \in [1, \cdots, K]$ that specifies the level of parameter tying, the mKPGM generation process samples a network of size b^K as follows. First, the model uses the standard KPGM algorithm with initiator matrix Θ to calculate a probability matrix P_ℓ and sample a graph G_ℓ and its respective adjacency matrix A_ℓ. Then, a subsequent Kronecker product is computed to produce a new probability matrix $P_{\ell+1} = G_\ell \otimes \Theta$. The process of sampling a graph before computing subsequent Kronecker products produces dependencies among the sampled edges. Thus a graph $G_{\ell+1}$ is sampled from $P_{\ell+1}$ for further Kronecker products. This process is then repeated K–ℓ–1 times to generate the final network G_K. For more details see [10].

The parameter ℓ controls the level of tying and thus impacts the variance and clustering of the model. Lower values of ℓ produce larger dependencies among the edges and greater clustering among the nodes. When $\ell = K$ the model is equivalent to the KPGM model and this produces lower clustering and lower variance.

The mKPGM likelihood has two parts: the *untied* part is calculated as in the original KPGM, while the *tied* part is based on the $K-\ell$ Kronecker products where edges share parameters and adjacency matrix A_ℓ generated from G_ℓ. The mKPGM likelihood of the graph, given a permutation σ, is:

$$P(G^*|\Theta,\sigma) =$$

$$P(G^*_\ell|\theta,\sigma_\ell) \left(\prod_{e_{ij}\in\mathbf{E}^*} A_\ell\left(\left\lfloor \frac{i-1}{b^{K-\ell}} \right\rfloor, \left\lfloor \frac{j-1}{b^{K-\ell}} \right\rfloor \right) \prod_{k=1}^{K-\ell} \theta_{i_k j_k} \right.$$

$$\left. \prod_{e_{ij}\notin\mathbf{E}^*} \left(1 - A_\ell\left(\left\lfloor \frac{i-1}{b^{K-\ell}} \right\rfloor, \left\lfloor \frac{j-1}{b^{K-\ell}} \right\rfloor \right) \prod_{k=1}^{K-\ell} \theta_{i_k j_k} \right) \right) \quad (2)$$

Unfortunately, even though the mKPGM likelihood (Eq. 2) is similar to that of KPGMs (Eq. 1), it can not easily be used as an objective function to estimate the parameters of the model. However, a new learning algorithm for mKPGM was developed in [9], which is based on *the simulated method of moments* (SMM). The strength of this approach is that it is permutation invariant—thus it avoids the difficulty of search over permutation space. The SMM learning algorithm searches for parameters Θ that minimize the following objective function:

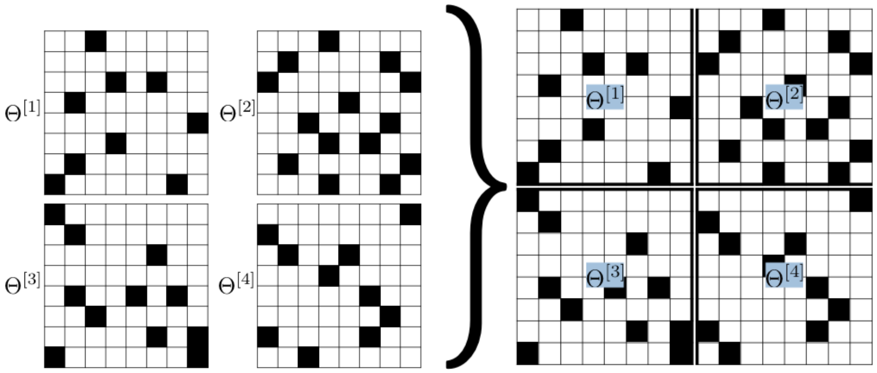
$$f(\Theta, \mathbf{F}^*) = \sum_{i=1}^{|\mathbf{F}|} \left(\frac{F_i^* - E[F_i|\Theta]}{F_i^*} \right) \quad (3)$$

Here F_i is a function over a network $G = (\mathbf{V}, \mathbf{E})$ that calculates a statistic of the graph, e.g., for number of edges: $F = |\mathbf{E}|$. Then, $\mathbf{F}^* = \{F_1^*, F_2^*, \cdots, F_m^*\}$ corresponds to a set of m *sample moments* of the training network $G^\star$ and $E[F_i|\Theta]$ is the expected value of those statistics (i.e., *distributional moments*) given particular values of Θ. The SMM learning algorithm [9] can utilize any moments that can be estimated from the generated networks. However, the authors suggests five moments: (i) the number of edges per nodes, (ii) average cluster coefficient, (iii) average geodesic distance, (iv) size of the largest connected component, and (v) nodes with degree greater than zero. These moments were selected because their can be estimated in linear time (approximating the geodesic distance) and are distinctive values for most networks.

3. BLOCK KRONECKER PRODUCT GRAPH MODEL

This section describes *block-KPGM* and its generation process. The description is complemented with a pseudo code, an example of the algorithm and a graphical comparison of generated networks.

The *block-KPGM* is a new generalization of KPGM that relax the stationary fractal structure of KPGMs by using multiple independent KPGM to model a network. Specifically, multiple KPGMs are utilized to model different part



$\Theta^{[\cdot]}$	POS_{XY}	
$\Theta^{[1]}$	1	1
$\Theta^{[2]}$	1	9
$\Theta^{[3]}$	9	1
$\Theta^{[4]}$	9	9

Figure 1: block-KPGM generation process. Four block are generated independently according to $\Theta^{[\cdot]}$ (left). The four independent blocks are put together generating the final network (middle), according to the matrix POS_{XY} (right).

of a network and then they are joined to recreate the entire network. This process extends the space graph covered by KPGM and keeps the elegant fractal structure of the model. Moreover, our initial experiments with this model, show that block-KPGMs generate networks with higher clustering coefficient, lower number of isolated nodes, and larger connected components in comparison to previous Kronecker models such as KPGM and mKPGM.

Formally, we can define the generative mechanism as follow. Consider the set of parameters $\Theta_b = \{\Theta^{[1]}, \cdots, \Theta^{[N_b]}\}$, where N_b corresponds to the number of blocks, and the vectors $\mathbf{b} = \{b_1, \cdots, b_{N_b}\}$ and $\mathbf{K} = \{K_1, \cdots, K_{N_b}\}$ which corresponds to the initial size of the parameters and the number of Kronecker multiplication for each $\Theta^{[i]}$ from Θ_b. With these parameters, the algorithm iterates over $\Theta^{[i]}$ generating a block of size $b_i^{K_i}$ according to KPGM. Specifically, given $\Theta^{[i]}$, the algorithm calculates the Kronecker product of $\Theta^{[i]}$ with itself $K_i - 1$ times to generate a probability matrix $\mathcal{P}_{ki}$ of size $b_i^{K_i}$. From $\mathcal{P}_{ki}$, every edge is sampled using a binomial distribution with probability $\mathcal{P}_{ki}[u, v]$ generating the block structures Bl_i. Note, that we call block rather than network, because Bl_i corresponds to a part of the final adjacency matrix with a specific fractal structure, rather than a set of nodes and edges.

Algorithm 1 block-KPGM generation algorithm

```
Require: $\Theta_b$, $\mathbf{b}$, $\mathbf{K}$, $\mathbf{XY}$
$N_b = |\Theta_b|$
for $i = 1; i++; i \leq N_b$ do
    Generate block $Bl_i$ using KPGM with $\Theta^{[i]}$, $b_i$, and $K_i$
for $i = 1; i++; i \leq N_b$ do
    Update adjacency matrix $A_{Bl}$, joining the block $Bl_i$
    according to $Pos_{XY}(i, 1)$ and $Pos_{XY}(i, 2)$
Return $A_{Bl}$
```

When each block Bl_i has been generated, the algorithm continues with the joining process of the N_b blocks. The algorithm starts placing each block Bl_i in the final adjacency matrix A_{Bl} according to the matrix Pos_{XY} which determine the exact position of every block. Pos_{XY} is a $N_b \times 2$ matrix with the coordinates of every block in the final adjacency matrix. For example, the block Bl_i should be put in the position $\{Pos_{XY}(i, 1), Pos_{XY}(i, 2)\}$. Once that all blocks are correctly positioned, the updated adjacency matrix A_{Bl} is returned by the algorithm. A pseudocode of this generation algorithm can be observed in algorithm 1.

A graphical example of this generation algorithm can be observed in figure 1. In this example, a final matrix with 16 nodes is generated using $N_b = 4$ different blocks of size 2^3 ($\Theta_b = \{\Theta^{[1]}, \cdots, \Theta^{[4]}\}$, $\mathbf{b} = \{2, \cdots, 2\}$, and $\mathbf{K} = \{3, \cdots, 3\}$). Once that all networks are generated (left part of the figure), the join process begins where Bl_1 is positioned at (1,1), Bl_2 at (1,9), Bl_3 at (9,1), and Bl_2 at (9,9). This generates the final adjacency matrix of size $N_v = 2^4$ that can be observed in the right part of the figure.

The effect of the multiple fractal structure in block-KPGM can be seen in Figure 2. The figure shows network generated by KPGM, mKPGM and block-KPGM. While KPGM and mKPGM uses the same Θ, block-KPGM utilizes different Θ in each block to show different fractal structures. To avoid unfair comparison, the expected number of edges in all networks are the same for the three models, however the blocks have different densities among them. The networks sampled from KPGM and mKPGM exhibit the same fractal structure, being the main difference the group of edges obtained by mKPGM. On the contrary, block-KPGM shows three different fractal structures. The upper left block shows the same fractal structure than KPGM and mKPGM, the upper right block shows all edges disperse uniformly on the block, and the lower right show a high number of edges in the main diagonal.

4. EXPERIMENTS

This section compare block-KPGM against KPGM and mKPGM. We analyzed the graph space covered by these model in smaller networks and we show that block-KPGMs can model some real networks that are impossible to be model by previous kronecker models. We also demonstrate that block-KPGM can model some real networks better than previous Kronecker models.

4.1 Empirical analysis

We perform an empirical analysis to investigate the characteristics that can be generated from the three different models (KPGM, mKPGM and block-KPGM), showing an approximation of the graph space covered by each model on the selected characteristics. To realize the empirical anal-

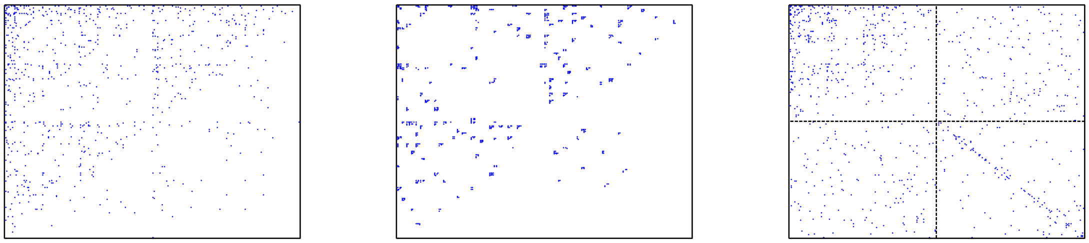


Figure 2: Generated networks of 2^8 nodes for different Kronecker product graph models: KPGM (left), mKPGM $\ell = 5$ (center), and block-KPGM (right).

ysis, we generated networks over a wide range of parameter values in Θ and measured the characteristics of the resulting graphs. For KPGM and mKPGM, we considered 22,060 different values of Θ for initial matrices of size $b = 2$. The parameters were generated considering every possible combination of Θ, such as $\theta_{11}, \theta_{12} \in \{0.01 : \delta : 1.00\}$, $\theta_{12} = \theta_{21}$, $\theta_{22} \in \{\theta_{11} : \delta : 1.00\}$ and $2.1 \leq S_\Theta \leq 2.4$ $\left(S_\Theta = \sum_{ij} \theta_{ij}\right)$. We utilized $\delta = 0.015$ and θ_{14} starting from θ_{11} to avoid repetition of the parameters with respect a permutation of Θ. For each Θ setting, we generated 75 undirected networks with $K = 9$ from KPGM ($\ell = K$) and mKPGM ($\ell = 6$). For block-KPGM, we considered $N_b = 4$, $\mathbf{b} = \{2, 2, 2, 2\}$, $\mathbf{K} = \{8, 8, 8, 8\}$, and $posXY = \{(1, 1), (1, 2^8+1), (2^8+1, 1), (2^8+1, 2^8+1)\}$ which generated four block of 2^8 nodes. Once that the block are joined, the final networks have $N_v = 2^9$ nodes. We considered 112,176 different values of Θ_b. The parameters of the first and forth block ($\Theta^{[1]}$ and $\Theta^{[4]}$), which correspond to undirected networks, were generated with every possible combination of Θ, such as $S_\Theta \geq 2$, $\theta_{11}, \theta_{12}, \theta_{22} \in \{0.01 : \delta : 1.00\}$ and $\theta_{12} = \theta_{21}$ with $\delta = 0.245$. Meanwhile, the parameters of the second and third blocks (directed networks) were generated by $S_\Theta \geq 1.5$, $\theta_{ij} \in \{0.01 : \delta : 1.00\}$ $\forall\ i, j \in \{1, 2\}$ with $\delta = 0.245$. Given that the final network is undirected, the third block is actually replaced by the transposed network generated by the second block. The final set Θ_b consists in all possible combinations of the parameters generated over each block, such as the expected number of edges in the final network will be between $[2.1^9, 2.4^9]$ (the expected number of edges for KPGM and mKPGM). From the final set Θ_b, we generated 75 networks for each set of parameters.

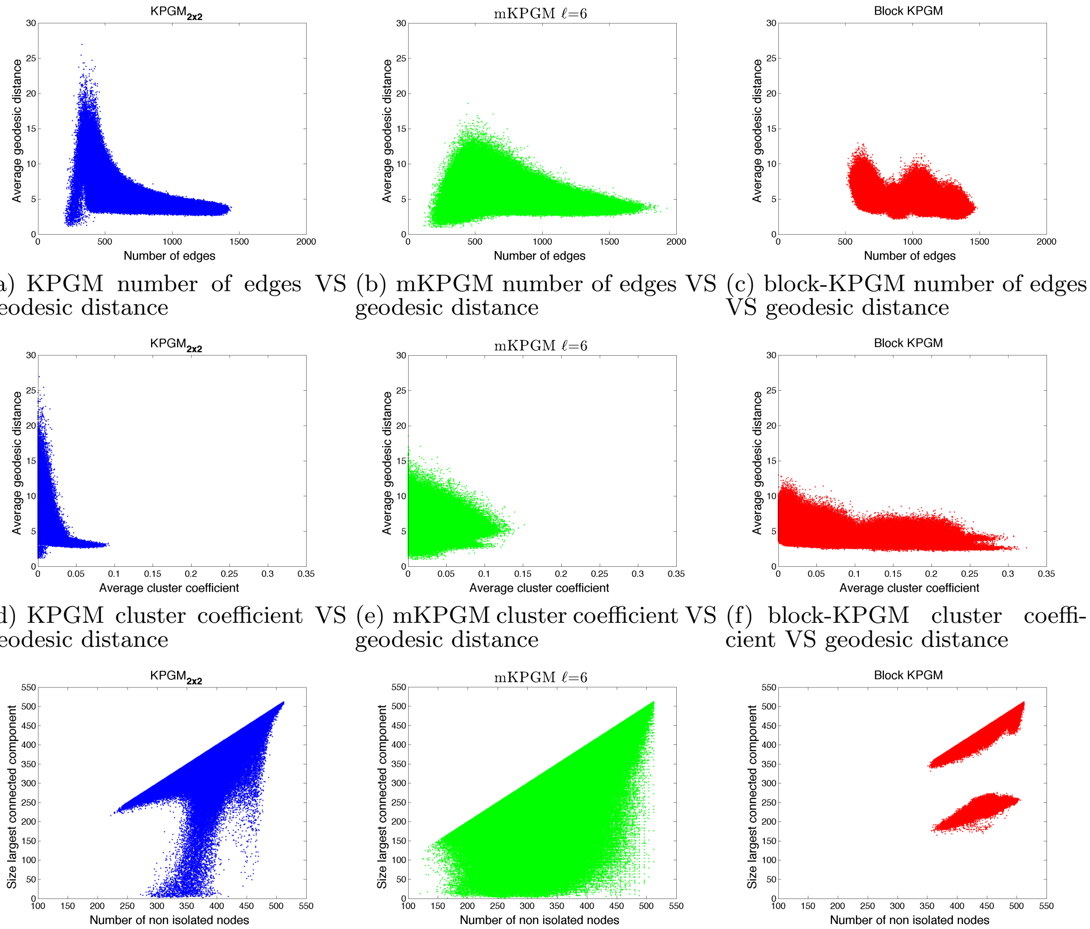
We compare the models plotting the five different network characteristics utilized in the mKPGM training algorithm (1) number of edges per nodes, (2) average cluster coefficient, (3) average geodesic distance, (4) number of non isolated nodes and (5) size of the largest connected component. The number of edges per nodes corresponds to the average degree of the nodes, where the degree d_i is simply the number of nodes in the graph that are connected to node i. The average clustering coefficient is the average of the clustering coefficient calculated over every 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. The average geodesic distance corresponds to the average over nodes geodesic distance, which is calculated as the average distance to reach the rest of the nodes in the network. The number of non isolated nodes corresponds to the number of nodes with at least one edge. Finally, the size of the largest connected component corresponds to the highest number of nodes linked in the network which analyze the connection in block-KPGM networks and determine if it generates multiple isolated components. We plot these characteristics to observe the behavior of block-KPGM in comparison to high number of isolated nodes generated by KPGM.

The results of this experiment can be observed in figure 3, where the first column corresponds to KPGM, the second column to mKPGM and the third column to block-KPGM. The first row of the figure shows the average geodesic distance against the number of nodes in the network. The number of edges, for all models, varies between $[2.1^9, 2.4^9]$; confirming a fair comparison among the models, where the results are not biased based on the number of edges. As can be appreciated in figure 3(c) block-KPGM has the lowest geodesic distance among the three models. Even though this could be something negative, most social networks have a small geodesic distance according to the small world phenomena [12, 13]. In contrast, KPGM and mKPGM generate networks with small number of edges and large geodesic distances. These types of networks correspond to multiple nodes connected as a chain, which do not resemble the structure of actual real world networks.

The second row shows the average geodesic distance against the clustering coefficient. Figure 3(f) shows that block-KPGM generates the highest cluster coefficient among the three models. It is even surprising the high cluster coefficient generated by block-KPGM networks, which duplicate the clustering coefficient generated by the other models.

The last row shows the number of isolated nodes against the size of the largest connected component. we can observe that block-KPGM reduces considerably the number of isolated nodes and at the same time, most of the nodes are connected 3(i). In average, for block-KPGM, only 7% of the nodes are isolated in comparison to the 13% or 29% of KPGM and mKPGM respectively. Similarly, the average sizes of the largest connected component are 468, 428 and 307 nodes for block-KPGM, KPGM and mKPGM respectively. These measures become more surprising when extreme values are considered. As is show in figures 3(g) and 3(h), KPGM and mKPGM may generate networks with more than 50% of isolated nodes or several disconnected networks, where the largest connected component has a size below five.

Even though, we analyzed over 22,060 different Θ's for KPGM and mKPGM, and 112,176 different set of param-



(a) KPGM number of edges VS geodesic distance

(b) mKPGM number of edges VS geodesic distance

(c) block-KPGM number of edges VS geodesic distance

(d) KPGM cluster coefficient VS geodesic distance

(e) mKPGM cluster coefficient VS geodesic distance

(f) block-KPGM cluster coefficient VS geodesic distance

(g) KPGM non isolated nodes VS connected component

(h) mKPGM non isolated nodes VS connected component

(i) block-KPGM non isolated nodes VS connected component

Figure 3: Variation of graph properties for synthetic networks for KPGM, mKPGM and block-KPGM.

eters for block-KPGM, there are still other networks that can be generated reducing the value of δ and/or increasing the range of expected number of edges in the network. For example, reducing the value of $\delta = 0.245$ in block-KPGM, it is possible to obtain over millions of new parameters to analyze, that will extend the graph space covered by block-KPGM.

4.2 Modeling real networks

We compare the ability of block-KPGM to model real networks with respect to KPGM and mKPGM. To assess whether the generated networks capture the properties we observe in real network populations, we use evaluation measures and visual comparisons of the properties of networks generated from the models over two real networks. The results indicate that Block-KPGM can model some real network that are not covered by previous Kronecker models.

The first network is drawn from the public Purdue Facebook network. Facebook is a popular online social network site with over 845 million members worldwide. We considered a set of over 50000 Facebook users belonging to the Purdue University network with its over 400,000 wall links consisting of a year-long period. We selected a single network with 2187 nodes and 5760 edges from the wall graph. The second network consists of a set of social networks from the National Longitudinal Study of Adolescent Health (AddHealth) [3]. 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 a social network with 1155 nodes and 7884 edges.

To initialize the block-KPGM, we utilized $N_b = 4$ with $\mathbf{b} = \{2,2,2,2\}$, $\mathbf{K} = \{10,10,10,10\}$, $posXY = \{(1,1),(1,2^{10}+1),(2^{10}+1,1),(2^{10}+1,2^{10}+1)\}$ and searched for the set of parameters $\Theta^{[\cdot]}$ that reasonably matched to our example datasets. To achieve this, we considered an exhaustive search of the set of possible parameter values for $\Theta^{[\cdot]}$ based on the expected number of edges. From all possible parameters, we picked the closest set of parameters that match the average clustering coefficient and geodesic distance of the training network. On the contrary, we trained mKPGM and KPGM utilizing the methods described in section 2, with $b = 3$, $K = 7$ and $\ell = 5$, to increase the graph space covered by these models [9]. However, considering that block-KPGM utilizes 10 different parameters to model real networks, we

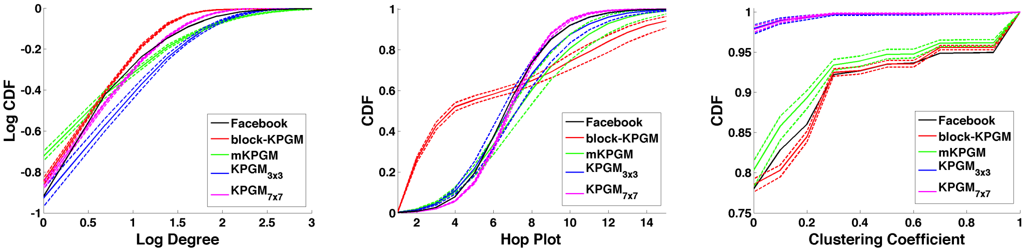


Figure 4: Variation of graph properties in generated Facebook networks.

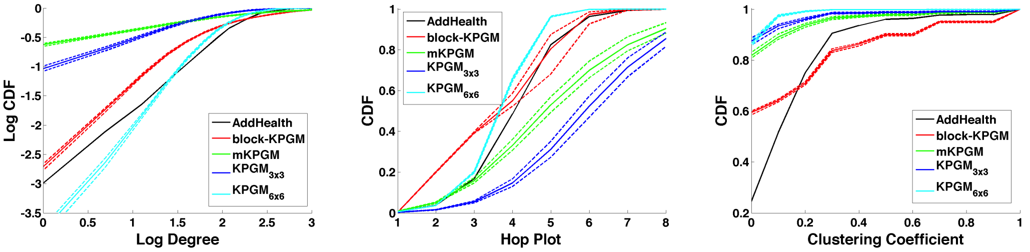


Figure 5: Variation of graph properties in generated AddHealth networks.

also increase the initiator matrix for KPGM to $b = 6, 7$ for AddHealth and Facebook respectively (modeling AddHealth and Facebook with 21 and 28 different parameters respectively).

Our evaluation investigates whether the models capture three important graph characteristics of real network datasets: degree, clustering coefficient, and hop plot (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 nodes that are $\leq h$ edges away from node v in G). To evaluate the ability of the models to capture these characteristics, we compare the cumulative distribution functions (CDFs) from 100 networks generated from the selected parameters to the CDFs of the original data. We plot each CDF independently and use those for visual comparison, but we would also like to evaluate the *relationships* among the distributions of characteristics to determine if the models are able to *jointly* capture the characteristics through the network. To measure this quantitatively, we utilize the 3D Kolmogorov-Smirnov distance (KS_{3D}), which measures the maximum distance between two multidimensional distributions [9]. We also compared the percentage of non isolated nodes, and largest connected component with relation to the number of non isolated nodes over each model. Finally, we generate a network from each of the model to determine the difference among them.

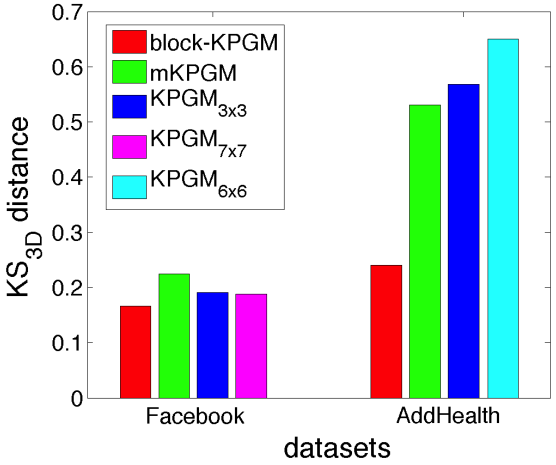
Figure 4 shows the CDFs for the Facebook data. We can observe that mKPGM, block-KPGM and $KPGM_{7x7}$ are the best models for Facebook data. Block-KPGM is one of the best model to match the degree and clustering coefficient of the real network, while stills model the hop plot. $KPGM_{7x7}$ is the best model to match the hop plot distribution and degree, however KPGMs do not match the clustering coefficient. Finally, mKPGM can partially model the three characteristics but is not able to model the three characteristics at the same time, as can be determined by the high KS_{3D} distance (Figure 6 left).

Figure 5 shows the CDFs for the AddHealth data, where block-KPGM is the best model to match these CDFs. Block-KPGM is the closest model in the degree distribution, it has an almost perfect match in the hop plot distribution, and it is the closest model for the clustering coefficient. By the contrary KPGMs and mKPGM do not model any of the characteristics. The low performance of KPGMs and mKPGM can be explained by the high clustering coefficient of the network. While KPGMs can not model the clustering coefficient, mKPGM needs to decrease the value of ℓ, increasing the variability of the networks; however, we are not modeling a population of network but a single network.

Figure 6 left shows the KS_{3D} distance over the three models. In both datasets, block-KPGM obtains the lowest error, confirming that can model the relation among degree, clustering coefficient and geodesic distance much better than other Kronecker models. In Facebook data, block-KPGM obtains a reduction of 11%, 13% and 26% in comparison to $KPGM_{7x7}$, $KPGM_{3x3}$ and mKPGM respectively. These percentages are increased to 63%, 58% and 55% with respect to $KPGM_{6x6}$, $KPGM_{3x3}$ and mKPGM, when we compare the KS_{3D} distance in AddHealth data. These results confirm the ability of block-KPGM to model new networks in comparisons to previous Kronecker models.

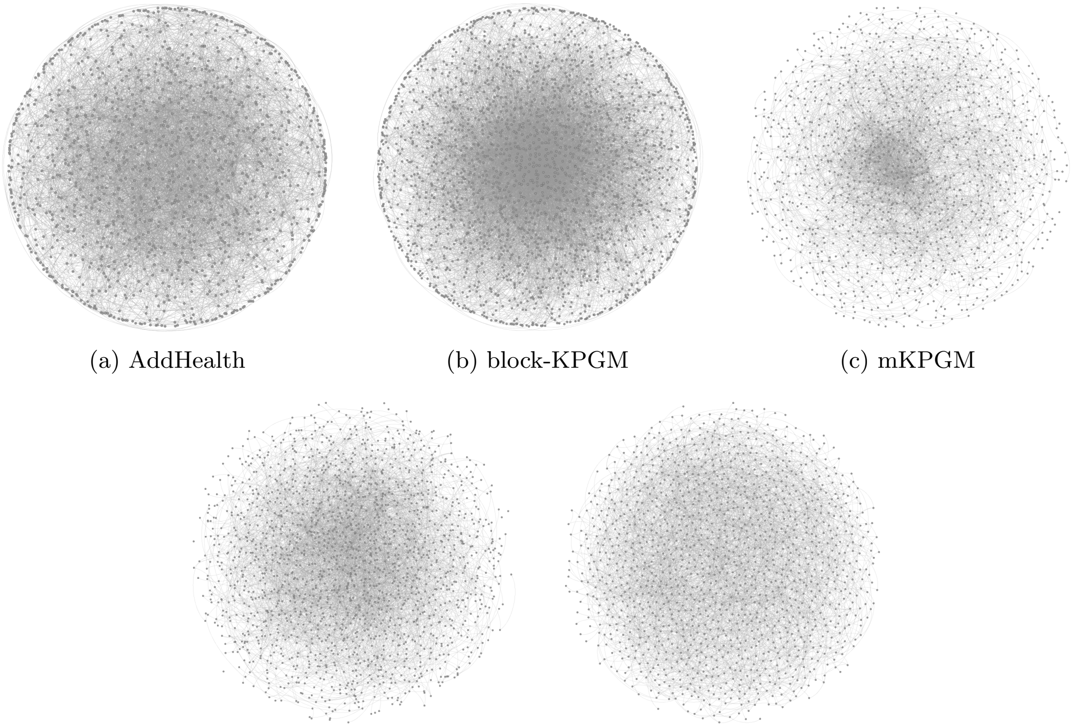
Figure 6 right shows the percentage of non isolated nodes and largest connected component over the three models. block-KPGM is the model with the highest number of non isolated nodes on Facebook (83%) and it has almost all nodes with edges on AddHealth (99%). On the contrary, KPGMs and mKPGM show a higher number of nodes with zero degree on Facebook and AddHealth (except by $KPGM_{6x6}$). This implies that an important number of nodes utilized by KPGM and mKPGM do not have any edge at all. Moreover, block-KPGM generates largest connected components similar to the expected in real network, in comparison to the other Kronecker models.

To further investigate the differences among the models, we plot the network structure itself, for all AddHealth net-



% non isolated nodes	Facebook	AddHealth
real network	100%	100%
block-KPGM	83%	99%
mKPGM	70%	56%
$KPGM_{3x3}$	76%	78%
$KPGM_{6x6}$	$--$	100%
$KPGM_{7x7}$	45%	$--$
% largest connected component	Facebook	AddHealth
real network	76%	100%
block-KPGM	85%	100%
mKPGM	84%	77%
$KPGM_{3x3}$	89%	92%
$KPGM_{6x6}$	$--$	100%
$KPGM_{7x7}$	93%	$--$

Figure 6: Left: Three dimensional Kolgomorov-Smirnov distance for Facebook and AddHealth. Right: percentage of non isolated nodes and size of largest connected component with respect to the non isolated nodes over the models



(a) AddHealth

(b) block-KPGM

(c) mKPGM

(d) $KPGM_{3x3}$

(e) $KPGM_{6x6}$

Figure 7: Example networks from AddHealth data.

works in Figure 7, where the original network is show in 7(a). The network generated with block-KPGM (7(b)) has the most similar structure to the original network. Both, the original and block-KPGM networks, have a great number of nodes in the middle of the network, and multiple nodes in the periphery which are connected among them and with the nodes in the middle of the network. On the contrary, none of the other Kronecker models show this behavior. In the case of mKPGM 7(c) and $KPGM_{3x3}$ 7(d), some nodes are concentrate in the middle of the network, however, most peripheral nodes are disconnected. Finally, $KPGM_{6x6}$ 7(d) do not show a high concentration of nodes in the middle and peripheral nodes are disconnected.

5. CONCLUSIONS

We presented a generalization of KPGM: the block-KPGM. The block-KPGM is a combination of multiple fractal structures, which extend the graph space covered by KPGM. Our empirical evaluation demonstrates that block-KPGM reduces the number of isolated nodes, increases the average cluster coefficient and reduces the isolated components of the generated networks. Moreover, block-KPGM is the best model to reproduce the characteristics over the two real networks analyzed in this paper, matching most CDFs, obtaining the lowest KS_{3D}, and reproducing similar networks.

There are a few directions to pursue for future work on block-KPGMs. First, we need to create a learning algorithm for block-KPGM to estimate the parameters from an observed network. Given that block-KPGM consists of multiple KPGMs, we need to understand how the separation of the network in different blocks could affect the structure of the network. This will allow us to determine if previous training methods can be utilized to train every block separately or if a new invariant permutation training algorithm should be implemented. Once that the training algorithm as been developed, we will compare the performance of block-KPGM against popular generative models.

Second, we need to explore the variability on the structure when different $b's$ values are utilized to generate a network. The number of nodes for KPGM and mKPGM is limited by the size of the initiator matrix (b) and the number of kronecker multiplications (K); joining multiple blocks with different values for b and K, will extend the use of block-KPGM to networks with any number of nodes, at the expense of the number of parameters.

6. REFERENCES

[1] K.-I. Goh, G. Salvi, B. Kahng, and D. Kim. Skeleton and fractal scaling in complex networks. *Phys. Rev. Lett.*, 96:018701, Jan 2006.

[2] C. Groer, B. D. Sullivan, and S. Poole. A mathematical analysis of the r-mat random graph generator. *Netw.*, 58(3):159–170, Oct. 2011.

[3] 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]. *University of North Carolina at Chapel Hill.*, 2008.

[4] M. Kim and J. Leskovec. Multiplicative attribute graph model of real-world networks. *Internet Mathematics*, 8(1-2):113–160, 2012.

[5] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *Proceedings of the 9th European conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD'05, pages 133–145. Springer-Verlag, 2005.

[6] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *JMLR*, 11(Feb):985–1042, 2010.

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

[8] M. Mahdian and Y. Xu. Stochastic kronecker graphs. *Random Structures & Algorithms*, 38(4):453–466, 2011.

[9] S. Moreno, S. Kirshner, and J. Neville. Learning mixed kronecker product graph models with simulated method of moments. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[10] S. Moreno, S. Kirshner, J. Neville, and S. Vishwanathan. Tied kronecker product graph models to capture variance in network populations. In *Allerton'10*, pages 17–61, 2010.

[11] C. Seshadhri, A. Pinar, and T. G. Kolda. An in-depth study of stochastic kronecker graphs. *Data Mining, IEEE International Conference on*, 0:587–596, 2011.

[12] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, pages 425–443, 1969.

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