Rigorous Analysis of Kronecker Graph Algorithms

Bharath Ramsundar, Group 49

December 10, 2012

1 Introduction

Real world graphs have been observed to display a number of surprising properties. These properties include heavy-tails for in- and out-degree distributions, small diameters, and a densification law [3]. These features do not arise from the classical Erdos-Renyi random graph model [2]. To address these difficulties, Kronecker Graphs were first introduced in [3] as a new method of generating graphs which match network data.

This work mathematically analyzes the convergence and runtime guarantees of generating and fitting algorithms for Kronecker Graphs. Existing references for Kronecker graphs [4] give heuristic descriptions of core algorithms, but suppress crucial mathematical details. There is source code available for these algorithms, but we don't wish to ensnare ourselves in language dependent syntax. Thus, we start by giving clean, detailed pseudocode for the algorithms we wish to analyze. From these descriptions, we derive a number of convergence, approximation, and lower bound results.

2 Kronecker Generation

We first analyze Algorithm (1) [4], a method of approximately generating a Stochastic Kronecker graph G given its initiator matrix Θ and product size k. The algorithm uses a normal approximation to sample the number of edges in G. Since each edge in G is equivalent to a choice of k elements of initiator matrix Θ, the algorithm proceeds to sample (u_j, v_j) according to a normalized version of Θ. The description in [4] claims that Algorithm 1 works well in practice for large graphs but does not prove any bounds. In this section, we justify the normal approximation used and prove associated error bounds.

Lemma 2.1. *Suppose given initiator matrix Θ of size $N_1 \times N_1$ and k. Let G denote a random Kronecker graph sampled from the k-th Kronecker power $\Theta^{[k]}$ of Θ. Let E denote the edge set of G. Then*

$$\mathbb{E}[|E|] = \left(\sum_{i,j=1}^{N_1} \Theta_{ij}\right)^k \qquad \mathbb{V}\text{ar}[|E|] = \left(\sum_{i,j=1}^{N_1} \Theta_{ij}\right)^k - \left(\sum_{i,j=1}^{N_1} \Theta_{ij}^2\right)^k$$

Proof. Recall the construction process of Stochastic Kronecker graph G. First start with empty graph G. Then for each pair $1 \leq u, v \leq N$, where $N = N_1^k$, add edge (u, v) with probability $\Theta^{[k]}[u, v]$. It follows that the number of edges present in the graph, $|E|$ is a sum of independent Bernoulli variables. Thus $|E|$ has mean and variance equal to the sum of the means and variances of the individual Bernoulli variables. We can consequently calculate that

$$\begin{aligned}
\mathbb{E}[|E|] &= \sum_{u,v=1}^{N} \mathbb{E}\left[\text{Bernoulli}\left(\Theta^{[k]}[u,v]\right)\right] = \sum_{u,v=1}^{N} \Theta^{[k]}[u,v] = \left(\sum_{i,j=1}^{N_1} \Theta_{ij}\right)^k \\
\mathbb{V}\text{ar}[|E|] &= \sum_{u,v=1}^{N} \mathbb{V}\text{ar}\left[\text{Bernoulli}\left(\Theta^{[k]}[u,v]\right)\right] = \sum_{u,v=1}^{N} \Theta^{[k]}[u,v](1-\Theta^{[k]}[u,v]) \\
&= \sum_{u,v=1}^{N} \Theta^{[k]}[u,v] - \sum_{u,v=1}^{N} \left(\Theta^{[k]}[u,v]\right)^2 \\
&= \left(\sum_{i,j=1}^{N_1} \Theta_{ij}\right)^k - \left(\sum_{i,j=1}^{N_1} \Theta_{ij}^2\right)^k
\end{aligned}$$

□

Note that $|E|$ is not necessarily Binomial since the Bernoulli variables for the edges have different means. Another note is that a direct application of the Central Limit Theorem does not inform us that $|E|$ is close to a normal distribution since CLT type theorems offer only asymptotic information. To get finite convergence information, we instead refer to a Berry-Esseen Theorem

Proposition 2.1 (Berry-Esseen)**.** *[1] Let $X_1, X_2, \ldots$ be independent random variables with $\mathbb{E}[X_i] = 0$, $E[X_i^2] = \sigma_i^2 > 0$ and $\mathbb{E}[|X_i|^3] = \rho_i < \infty$. Let*

$$S_n = \frac{X_1 + \ldots + X_n}{\sqrt{\sigma_1^2 + \ldots + \sigma_n^2}}$$

Let F_n be the cdf of S_n and Φ the cdf of the normal distribution. Then there exists constant C such that

$$\sup_x |F_n(x) - \Phi(x)| \leq C\psi, \qquad \psi = \frac{\max\limits_{1\leq i\leq n} \frac{\rho_i}{\sigma_i^2}}{\sqrt{\sum_{i=1}^n \sigma_i^2}}$$

We can use this result to prove a bound on the error for a normal approximation to the distribution of $|E|$. Let X_{ij} be an indicator variable that is 1 if and only edge (i,j) exists in G. Then X_{ij} is Bernoulli with distribution $\mu_{ij} = \Theta^{[k]}[i,j]$. Let $Y_{ij} = X_{ij} - \mu_{ij}$ where $\mu_{ij} = \mathbb{E}[X_{ij}]$. Note that

$$\mathbb{E}[Y_{ij}] = 0, \qquad \mathbb{E}[Y_{ij}^2] = \mathbb{V}\text{ar}[X_{ij}] = \sigma_{ij}^2 = \mu_{ij}(1-\mu_{ij})$$

Note that since $0 \leq \mu_{ij} \leq 1$ and X_{ij} is a $0,1$ indicator variables, it follows that $|Y_i|^3 \leq 1$. Then $\mathbb{E}[|Y_i|^3] < 1$. The required conditions for the Berry-Esseen condition hold, so we can apply the theorem to achieve the following lemma

Lemma 2.2. *The approximation*

Algorithm 1: KronGen: Generate Kronecker Graph From Parameter Matrix

Input: Θ: Parameter Matrix; k: degree of desired Kronecker Graph;
Output: Generated Kronecker Graph G

$\mu = \left(\sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\Theta_{ij}\right)^k$;
$\sigma^2 = \left(\sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\Theta_{ij}\right)^k - \left(\sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\Theta_{ij}^2\right)^k$;
$|E| \sim \mathcal{N}(\mu, \sigma^2)$;
for $i = 1$ **to** $|E|$ **do**
  for $j = 1$ **to** k **do**
    $(u_j, v_j) \sim \frac{1}{\mu^{1/k}}\Theta$;
  $u = (u_1, \ldots, u_k)$; $v = (v_1, \ldots, v_k)$;
  Add edge (u, v) to G;
return $\widehat{\Theta}$;

$$|E| \sim \mathcal{N}\left(\sum_{i,j=1}^{N}\mu_{ij}, \sum_{i,j=1}^{N}\mu_{ij}(1-\mu_{ij})\right)$$
$$\sim \mathcal{N}\left(\left(\sum_{i,j=1}^{N_1}\Theta_{ij}\right)^k, \left(\sum_{i,j=1}^{N_1}\Theta_{ij}\right)^k - \left(\sum_{i,j=1}^{N_1}\Theta_{ij}^2\right)^k\right)$$

has error bound

$$\sup_x |F_n(x) - \Phi(x)| \le C\psi = \frac{\max_{1\le i,j\le n}\frac{\rho_{ij}}{\sigma_{ij}^2}}{\sqrt{\sum_{i,j=1}^{n}\sigma_{ij}^2}}$$
$$\le \frac{1}{\left(\min_{1\le i,j\le N_1}\Theta_{ij}^k(1-\Theta_{ij}^k)\right)\sqrt{\left(\sum_{i,j=1}^{N_1}\Theta_{ij}\right)^k - \left(\sum_{i,j=1}^{N_1}\Theta_{ij}^2\right)^k}}$$

3 MLE Parameter Fitting

The next algorithm (2) from [4] describes how the MLE estimate $\widehat{\Theta}$ is calculated. This algorithm hill-climbs along the gradient to find a local minima (in this section, we assume for simplicity of analysis that the number of local minima is not too large). However, due to the exponential size of Kronecker graphs (for $N = N_1^k$), computing the gradient is nontrivial. Consequently, the gradient is estimated probabilistically. This estimation uses a MCMC process. In the next two sections, we consider the theoretical guarantees on the probabilistic estimate and on the MCMC process.

Algorithm 2: KronFit: Fit Kronecker Graph to Data

Input: N_1: Size of Parameter Matrix; G: Graph of $N = N_1^k$ nodes; η: learning rate;
Output: MLE parameters $\widehat{\Theta}$

1 initialize $\widehat{\Theta} = U(N_1, N_1) \in \mathbb{R}^{N_1 \times N_1}$ as random matrix;
2 **while** $\widehat{\Theta}$ *not converged* **do**
3 $\quad \frac{\partial}{\partial \widehat{\Theta}} \ell(\widehat{\Theta}) \leftarrow \text{ApproxGradient}(\Theta)$;
4 $\quad \widehat{\Theta} \leftarrow \widehat{\Theta} + \eta \frac{\partial}{\partial \widehat{\Theta}} \ell(\widehat{\Theta})$;
5 **return** $\widehat{\Theta}$;

3.1 Approximate Log-Likelihood and Gradient

We start with the following useful identities

$$\begin{aligned} \ell(\Theta) &= \log \sum_{\sigma} \mathbb{P}(G \mid \Theta, \sigma)\, \mathbb{P}(\sigma) \\ \frac{\partial}{\partial \Theta} \ell(\Theta) &= \sum_{\sigma} \frac{\partial \log \mathbb{P}(G \mid \Theta, \sigma)}{\partial \Theta}\, \mathbb{P}(\sigma \mid G, \Theta) \\ &= \mathbb{E}_{\sigma \mid G, \Theta} \left[\frac{\partial \log \mathbb{P}(G \mid \Theta, \sigma)}{\partial \Theta} \right] \end{aligned}$$

The algorithm (3) approximately calculates the log-likelihood and gradient for parameter matrix Θ. We can view this algorithm as a method to estimate the expectation of random variable $\frac{\partial \log \mathbb{P}(G|\Theta,\sigma)}{\partial \Theta}$. Then we have the following result.

Lemma 3.1. *Assume that samples σ_t drawn in line 3 of Algorithm (3) are drawn from the true distribution and are uncorrelated and independent. Then Algorithm 3 unbiasedly estimates $\ell(\Theta)$ and $\frac{\partial}{\partial \Theta} \ell(\Theta)$. The variances of these estimates are finite and shrink to 0 as $T \to \infty$.*

Proof. We only give proofs for the gradient estimates. The proofs for the likelihood estimates are similarly derived. We start by calculating

$$\begin{aligned} \mathbb{E}_{\sigma \mid G, \Theta} \left[\frac{\partial}{\partial \theta} \widehat{\ell}(\Theta) \right] &= \mathbb{E}_{\sigma \mid G, \Theta} \left[\frac{1}{T} \sum_{t=1}^{T} \nabla_t \right] = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\sigma_t \mid G, \Theta} [\nabla_t] \\ &= \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\sigma \mid G, \Theta} \left[\frac{\partial}{\partial \Theta} \ell(\Theta) \right] \\ &= \mathbb{E}_{\sigma \mid G, \Theta} \left[\frac{\partial}{\partial \Theta} \ell(\Theta) \right] \end{aligned}$$

The third equality above uses the assumption that samples σ_t are drawn from the true distribution $\sigma \mid G, \Theta$. We now perform the corresponding variance calculation.

Algorithm 3: ApproxGradient: Calculate Gradient Approximately

Input: Θ: Parameter Matrix; G: Graph of $N = N_1^k$ nodes;
Output: Log-Likelihood $\widehat{\Theta}$

initialize $\widehat{\Theta} = U(N_1, N_1) \in \mathbb{R}^{N_1 \times N_1}$ as random matrix;
for $t = 1$ **to** T **do**
  $\sigma_t = \text{SamplePermutation}(G, \Theta)$;
  $\ell_t = \log \mathbb{P}(G \mid \sigma_t, \Theta)$;
  $\nabla_t = \frac{\partial}{\partial \Theta} \log \mathbb{P}(G \mid \sigma_t, \Theta)$;
return $\widehat{\ell}(\Theta) = \frac{1}{T} \sum_{t=1}^{T} \ell_t, \quad \frac{\partial}{\partial \Theta} \widehat{\ell}(\Theta) = \frac{1}{T} \sum_{t=1}^{T} \nabla_t$;

$$\begin{aligned}
\mathbb{V}\text{ar}_{\sigma|G,\Theta}[\widehat{\ell}(\Theta)] &= \mathbb{V}\text{ar}_{\sigma|G,\Theta}\left[\frac{1}{T}\sum_{t=1}^{T} \nabla_i\right] = \frac{1}{T^2}\sum_{t=1}^{T} \mathbb{V}\text{ar}_{\sigma_t|G,\Theta}\left[\nabla_t\right] \\
&= \frac{1}{T^2}\sum_{t=1}^{T} \mathbb{V}\text{ar}_{\sigma|G,\Theta}\left[\frac{\partial}{\partial\Theta}\ell(\Theta)\right] \\
&= \frac{1}{T}\,\mathbb{V}\text{ar}_{\sigma|G,\Theta}\left[\frac{\partial}{\partial\Theta}\ell(\Theta)\right]
\end{aligned}$$

The second equality uses the assumption that the σ_t are uncorrelated and independent. The third equality uses the assumption that samples σ_t are drawn from the true distribution. Thus it now suffices to prove that $\mathbb{V}\text{ar}_{\sigma|G,\Theta}[\frac{\partial}{\partial\Theta}\ell(\Theta)] < \infty$ to achieve our result. However, since σ can achieve only finitely many values, the finiteness of the variance follows, and we have achieved the desired result. □

3.2 Sampling a Permutation

The problem of deriving MLE estimates for initiator matrix Θ suffers from a problem of permutations. Two MLE estimates which differ only in how elements of the Stochastic Kronecker matrix are associated to graph elements should have the same likelihood. To resolve this problem, assume graph G, initiator matrix Θ, and power k as given. Let σ be a permutation of the labelling of nodes of graph G. We can compute the likelihood of permutation σ as follows

$$\begin{aligned}
P(\sigma \mid G, \theta) &= \frac{P(\sigma, G, \Theta)}{\sum_\tau P(\tau, G, \Theta)} = \frac{P(\sigma, G, \Theta)}{Z} \\
&= \frac{1}{Z} \prod_{(u,v)\in G} \Theta^{[k]}(\sigma(u), \sigma(v)) \prod_{(u,v)\notin G} \left(1 - \Theta^{[k]}(\sigma(u), \sigma(v))\right)
\end{aligned}$$

The above distribution is difficult to sample from, since normalization factor Z is unknown and since the products in the above expression contain exponentially many terms ($N^2 = N_1^{2k}$). However, the form above suggests that MCMC steps might make sampling easier since most terms

will cancel. Algorithm (4) [4] states the Metropolis-Hastings sampling procedure used to sample σ. In this section, we study the mixing time of this procedure.

[4] shows that each iteration of the inner loop in Algorithm (4) takes time $O(kN)$. This linear efficiency is a strong achievement. However, the analysis in [4] does not consider the number of iterations required for the drawn sample to mix correctly. That is, the required number of iterations of the inner loop is not considered. Let S_N represent the set of permutations of N elements. We start with some definitions.

Definition 3.1. Let $\pi = P(\sigma \mid G, \Theta)$. Let $P^t(\sigma, \cdot)$ be the distribution over permutations after t iterations of the inner loop of Algorithm (4) starting with $\sigma^{(0)} = \sigma$. Then define

$$
\begin{aligned}
d(t) &= \frac{1}{2} \max_{\sigma \in S_N} \left(\sum_{\psi \in S_N} |P^t(\sigma, \psi) - \pi(\psi)| \right) \\
\bar{d}(t) &= \frac{1}{2} \max_{\sigma, \psi \in S_N} \left(\sum_{\eta \in S_N} |P^t(\sigma, \eta) - P^t(\psi, \eta)| \right) \\
t_{\text{mix}} &= \min\{t : \ d(t) \leq \epsilon\}
\end{aligned}
$$

The mixing time is the most common measure of the convergence of a Markov Chain algorithm. To obtain good samples σ_t, the inner loop of Algorithm (4) should be run at least $t_{\text{mix}}(\varepsilon)$ for $\varepsilon < \frac{1}{2}$. The following proposition is easily proven

Proposition 3.1. *[5]*

$$d(t) \leq \bar{d}(t) \leq 2d(t)$$

With this background in place, we start with the following lemma.

Lemma 3.2. *Assume matrix Θ has no elements equal to 0 or 1. Let $\sigma \in S_N$. Let*

$$t_{\text{diam}}(\sigma) = \min_t \left(\forall \psi \in S_N, \ P^t(\sigma, \psi) > 0 \right)$$

Then $t_{\text{diam}}(\sigma) \geq \lceil \frac{N}{2} \rceil$.

Proof. Note that if matrix Θ has no elements equal to 0 or to 1, then $P(\sigma \mid G, \Theta) > 0$ for all σ. Now let $\psi \in S_N$ be a derangement (a permutation that has no fixed points). Then $\psi \circ \sigma$ and σ do not agree on any values. Note that each MCMC move in the inner loop can affect at most 2 values of the current permutation. Since $\psi \circ \sigma$ and σ disagree on all N elements, we will need at least $\lceil N/2 \rceil$ steps to change σ to $\psi \circ \sigma$. For t less than this value, $P^t(\sigma, \psi \circ \sigma) = 0$. Thus $t_{\text{diam}}(\sigma) \geq \lceil N/2 \rceil$. □

We use the above lemma to lower bound the mixing time of Algorithm (4).

Theorem 3.1. *Let $\varepsilon < \frac{1}{2}$.*

$$t_{\text{mix}}(\varepsilon) \geq \left\lfloor \frac{N}{4} \right\rfloor$$

Algorithm 4: SamplePermutation: Metropolis Sampling of Node Permutation

Input: Θ: Parameter Matrix; G: Graph of $N = N_1^k$ nodes; $\sigma^{(0)}$: starting permutation;
Output: Permutation $\sigma^{(i)} \sim P(\sigma \mid G, \Theta)$

$i = 1$;
repeat
  Draw j, k from $\{1, \ldots, N\}$ uniformly;
  $\sigma^{(i)} = \text{SwapNodes}(\sigma^{(i-1)}, j, k)$;
  Draw u from $U(0, 1)$;
  if $u > \frac{P(\sigma^{(i)}|G,\Theta)}{P(\sigma^{(i-1)}|G,\Theta)}$ **then**
    $\sigma^{(i)} = \sigma^{(i-1)}$;
  $i = i + 1$;
until $\sigma^{(i)} \sim P(\sigma \mid G, \Theta)$;

Proof. Let $\sigma \in S_N$ and let $\psi \in S_N$ be a derangement. Let $D = t_{\text{diam}}(\sigma)$. Notice that $P^{\lfloor (D-1)/2 \rfloor}(\sigma, \cdot)$ and $P^{\lfloor (D-1)/2 \rfloor}(\psi \circ \sigma, \cdot)$ have disjoint supports. For if the supports of these two distributions met, by the reversibility of Metropolis-Hastings [5] we would have that $t_{\text{diam}} \leq D - 1$, which would be a contradiction. By definition of $\bar{d}$, it follows that $\bar{d}(\lfloor D/2 \rfloor) = 1$. Then $d(t) \geq \frac{1}{2}$ by Lemma (3.1). It follows that $t_{\min}(\varepsilon) > \lfloor D/2 \rfloor$. We now deal with the floors and ceilings and calculate

$$t_{\min}(\varepsilon) > \lfloor D/2 \rfloor \geq \left\lfloor \frac{1}{2} \lceil N/2 \rceil \right\rfloor \geq \left\lfloor \frac{1}{2} N/2 \right\rfloor = \lfloor N/4 \rfloor$$

□

4 Remaining Work

We started by extracting mathematically relevant details of [4] and produced clean pseudocode. From this base, we have proved a number of rigorous mathematical results about the convergence and error rates of these algorithms. However, a number of open questions still remain. Notably, we did not succeed in achieving upper bounds on the mixing time for Algorithm (4). The difficulty seems to be that the presence of free parameter G makes it difficult to calculate eigenvalues of the Markov Transition matrix (most upper bounds on mixing time result from knowledge of the eigenvalues of the transition matrix [5]). Future work might explore the MCMC literature more deeply to find inspiration for a proof.

References

[1] Berry esseen theorem. http://en.wikipedia.org/wiki/Berry-Esseen_theorem.

[2] P. Erdos and A. Renyi. On random graphs i. 1959.

[3] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplications. In *Knowledge Discovery in Database (PKDD)*, 2005.

[4] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research (JMLR)*, 2010.

[5] D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. Americal Mathematical Society, 2008.