

## (Introduction)

The question of how one can generate synthetic, realistic, and time-evolving graphs bares great relevance on such enterprises as graph mining, finding abnormalities in networks, conducting extrapolations, simulating possible what-if scenarios, and studying the laws that a graph generator should obey in order for its generations to be realistic.

The earliest stochastic generative model for complex networks was the random graph model introduced by P. Erdos and A. Renyi in “On The Evolution of Random Graphs”(1960), in which each node pair,  $(i, j) \in V(G)$  has an equivalent, independent linking probability,  $p$ . This generative model, while permitting for extensive mathematical theory, produces graphs which fail to match real-world networks, especially in their lack of heavy-tailed degree distributions. The majority of the models which attempted to mitigate the noted short-comings of the ER-Model advanced some variation of *Preferential Attachment* (See Albert and Barabasi in “Emergence of Scaling in Random Networks” and “Statistical Mechanics of Complex Networks”) in which new nodes are added to an existing graph sequentially, and are connected preferentially to nodes with higher degrees. These models, while producing power-law tails and low diameter networks, failed to produce graphs whose diameter grows slowly with  $n := |V|$  (also called the “Shrinking Diameter Property”).

Most generators focus on a small number of static patterns and neglect others, leading to models which either fail to generate graphs capable of matching the evolving complexity and characteristic richness of real networks or are too complex to be mathematically tractable. These issues are addressed in Leskovec et. al. “Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication” [1], wherein they introduce The Kronecker Graph Generator and demonstrate its superiority to competing models.

## (Kronecker)

There are a number of challenges posed by working with natural graphs. These challenges include: that *the difficulty of obtaining natural data-sets is high; existing data-sets for natural graphs, and of sufficient size, are few; synthetic graphs lack certain characteristics found in natural graphs*. These “characteristics of natural graphs” include the property of having a degree distribution following a power law probability distribution and the property of self-similarity, or of large scale connections between parts of the graph reflecting the constituent subgraphs. An accurate generative model (where accuracy is measured with respect to natural graphs) will preserve these properties. Thus enters the stochastic kronecker graph model.

At the essence of The Mathematical Formulation for the Kronecker generation model is the so called *Kronecker Product*: a 2-ary matrix operation in which one operand is “nested” or copied inside the other. The application of this operation to the adjacency matrices of graphs is the novel step in the Kronecker Model. Using the Kronecker Product on a graph and its copy can easily

produce a self-similar graph (as does taking the similarly defined *Kronecker Power*). This simple loop: of repeatedly performing the Kronecker operation on an initiator graph, readily — and deterministically — produces accurate synthetic networks. While graphs generated thusly have a variety of desired properties, their discrete nature produces unwanted staircase effects in the degree and eigenvalue distributions ( mostly because of the high multiplicity of individual values). To generate *random*, natural graphs, we start with an adjacency matrix,  $P$ , for which  $P_{ij} \in [0, 1]$ . This gives us the *stochastic* Kronecker Model.

As outlined in [1], the (naive) method of generation for a Kronecker graph is provided by the following recursive construction:

- I. Start with an *initiator matrix*:  $M \in \mathbb{R}^{n \times n}$  and an integer  $k \in \mathbb{N}^+$ . We compute the  $k^{th}$  Kronecker Product of  $M$  to produce a large probability matrix,  $R$ , where the  $R_{ij}$  —  $th$  entry corresponds to nodes  $(i, j) \in V$ . The explicit formula for  $R_{ij}$  is given by:

$$\prod_{u=0}^{k-1} \theta \left[ \left\lfloor \frac{i}{n^u} \right\rfloor \bmod n, \left\lfloor \frac{j}{n^u} \right\rfloor \bmod n \right]$$

- II. With  $R$  generated thusly, the random Kronecker Graph,  $G_k := \{V, E\} : n := |V|, m := |E|$ , is generated by appending  $(i, j)$  to  $E$  with probability  $R_{ij}$ .

With an initiator matrix,  $M \in \mathbb{R}^{n \times n}$ , and  $k$  iterations, the generated matrix,  $R$ , will be of  $n^k \times n^k$  entries, and since we will take a product of  $k$  values to obtain each cell of  $R$ , and since there will be  $n^k$  by  $n^k$  cells, we get a runtime of  $O(kn^{2k})$ .

This algorithm can be augmented into a faster, edge-oriented procedure for which the run-time in the generation of sparser, real-world graphs, is  $O(kn^k)$ . This approach, as well a parallel algorithm which permits us to generate every edge in the graph independently of every other edge, thereby allowing us to parallelize the graph's generation and produce very large networks, is discussed in great detail by Leskovec et. al. in "Kronecker graphs: an approach to modeling networks".

The properties of stochastic Kronecker Graphs, discussed extensively in [1] and Nazim, et al. in "Properties of stochastic Kronecker Graphs", include the following:

- I. **Multinomial Degree Distributions:** Kronecker Graphs have multinomial degree distributions for in- and out-degrees.
- II. **Multinomial eigenvalue Distributions:** The Kronecker Graph  $G_k$  has a multinomial degree distributions for its eigenvalues.
- III. **DPL:** Kronecker graphs follow the Densification Power Law (DPL) with a densification exponent:  $a = \log(m)/\log(n)$

**(Random-Dot Product)**

(.)

**(Sources)**

Leskovec, Jure, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: an approach to modeling networks.