Modeling Network Structure using Kronecker Multiplication*

Jure Leskovec, Carnegie Mellon University, jure@cs.cmu.edu

ABSTRACT

Given a large, real graph, how can we generate a synthetic graph that matches its properties, *i.e.*, it has similar degree distribution, similar (small) diameter, similar spectrum, etc? We propose to use "Kronecker graphs", which naturally obey all of the above properties. We present a fast *linear* time algorithm for fitting the Kronecker graph generation model to real networks. Experiments on large real and synthetic graphs show that Kronecker graphs indeed mimics very well the patterns found in the target graphs. Once fitted, the model parameters and the resulting synthetic graphs can be used for anonymization, extrapolations, and graph summarization.

1. INTRODUCTION

Large, real graphs have a lot of structure: they typically obey power laws in their in- and out-degree distributions; they have small diameter; and they often have a self-similar structure, with communities within communities.

Although several graph generators have been proposed in the past (like the preferential attachment, the small-world model, the forest fire model, etc.), very little work exists on how to fit the parameters of such models.

This is exactly the problem we examine. Given a large real network, we want to choose the most realistic generator and estimate its parameters, so that our resulting synthetic graph matches the statistical properties of the real graph.

We examine the Kronecker graphs [2] which are based on Kronecker matrix multiplication. Kronecker model can generate graphs that obey many of the patterns found in real graphs. Moreover, we develop a fast and scalable algorithm for fitting Kronecker graphs by using maximum likelihood. When calculating the likelihood one needs to consider all mappings of nodes to the graph adjacency matrix, which becomes intractable for graphs with more than a few nodes. Even when given "true" mapping evaluating the likelihood is prohibitively expensive. We present solutions to both problems: We develop Metropolis sampling algorithm for node mapping and approximate the likelihood to obtain a *linear* time algorithm that scales to large graphs.

Once the model is fitted to the real graph, there are several benefits and applications: (a) The parameters give us information about the structure of the graph itself; (b) *Extrapolations:* we can use the model to generate a larger graph, to predict how the network will look like in the future. (c) *Sampling:* conversely, we can also generate a smaller graph, which may be useful for running simulation experiments.

*This a short version of the paper that appeared in the Proceedings of the International Conference on Machine Learning (ICML) 2007.

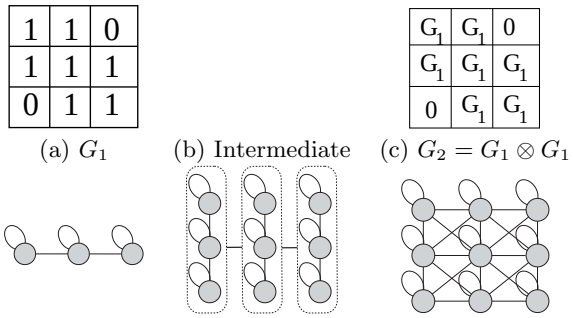


Figure 1: *Kronecker multiplication:* Top row: structure of adjacency matrices. Bottom: corresponding graphs – "3-chain" and its Kronecker product with itself; each of the nodes gets expanded into 3 nodes, which are then linked.

2. KRONECKER GRAPHS

Kronecker matrix multiplication was recently proposed for realistic graph generation, and shown to be able to produce graphs that match many of the patterns found in real graphs [2]. Kronecker graphs are based on a recursive construction. A procedure that is best described in terms of the *Kronecker product* of graph adjacency matrices.

Deterministic Kronecker Graphs The main idea is to create self-similar graphs, recursively. We begin with an *initiator* graph G_1, with N_1 nodes, and by recursion we produce successively larger graphs $G_2 \ldots G_n$ such that the k^{th} graph G_k is on $N_k = N_1^k$ nodes. *Kronecker product* is a perfect tool for this goal:

DEFINION 1 (KRONECKER PRODUCT OF MATRICES). *Given two matrices* $\mathbf{U} = [u_{i,j}]$ *and* $\mathbf{V}$ *of sizes* $n \times m$ *and* $n' \times m'$ *respectively, the Kronecker product matrix* $\mathbf{S}$ *of dimensions* $(n * n') \times (m * m')$ *is given by*

$$\mathbf{S} = \mathbf{U} \otimes \mathbf{V} \doteq \begin{pmatrix} u_{1,1}\mathbf{V} & u_{1,2}\mathbf{V} & \cdots & u_{1,m}\mathbf{V} \\ u_{2,1}\mathbf{V} & u_{2,2}\mathbf{V} & \cdots & u_{2,m}\mathbf{V} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n,1}\mathbf{V} & u_{n,2}\mathbf{V} & \cdots & u_{n,m}\mathbf{V} \end{pmatrix} \quad (1)$$

Kronecker product of two graphs is defined as Kronecker product of their adjacency matrices. We denote k^{th} Kronecker power of G_1 as $G_1^{[k]}$, where $G_k = G_1^{[k]} = G_{k-1} \otimes G_1$.

Figure 1 shows the recursive construction of Kronecker graphs. We start with G_1, a 3-node chain, and Kronecker power it to obtain G_2. To produce G_k from G_{k-1}, we "expand" (replace) nodes of G_{k-1} by copies of G_1, and join the copies according to the adjacencies in G_{k-1} (see fig. 1). Intuitively communities in the graph grow recursively, with community recursively getting expanded into miniature copies of the community. Nodes in the sub-community then link among themselves and to nodes from other communities.

Stochastic Kronecker Graphs Here we will be working with a stochastic version of Kronecker Graphs. The difference is that now initiator matrix is stochastic: we start with a $N_1 \times N_1$ *probability matrix* $\Theta = [\theta_{ij}]$, where the element $\theta_{ij} \in [0, 1]$ is the probability that edge (i, j) is present. We compute the k^{th} Kronecker power $\mathcal{P} = \Theta^{[k]}$; And then for each $p_{uv} \in \mathcal{P}$, include edge (u, v) with probability p_{uv}.

3. FITTING KRONECKER GRAPHS

Suppose we are given a graph G on $N = N_1^k$ nodes (for some positive integer k), and a N_1 by N_1 Stochastic Kronecker Graph initiator matrix Θ. Θ is a parameter matrix, a set of parameters that we aim to estimate. Assume N_1 is given (see [3] for the solution of how to relax this assumption). Next, we create a Stochastic Kronecker Graph probability matrix $\mathcal{P} = \Theta^{[k]}$, where every cell p_{ij} of $\mathcal{P}$ contains a probability that node i links to node j. We evaluate the probability that G is a realization of $\mathcal{P}$. The task is to find such Θ that has the highest probability of generating G. Formally, we are solving:

$$\arg\max_{\Theta} P(G|\Theta) \tag{2}$$

A permutation σ of the set $\{1, \ldots, N\}$ defines the mapping of nodes from G to stochastic adjacency matrix $\mathcal{P}$. The node labeling is arbitrary and carries no significant information. A priori all labelings are equally likely. To evaluate the likelihood of G one needs to consider all possible mappings of N nodes of G to rows of $\mathcal{P}$. We work with *log-likelihood* $l(\Theta)$, and solve $\arg\max_\Theta l(\Theta)$, where $l(\Theta)$ is defined as:

$$\begin{aligned} l(\Theta) &= \log P(G|\Theta) = \log \sum_{\sigma} P(G|\Theta, \sigma) P(\sigma|\Theta) \\ &= \log \sum_{\sigma} P(G|\Theta, \sigma) P(\sigma) \end{aligned} \tag{3}$$

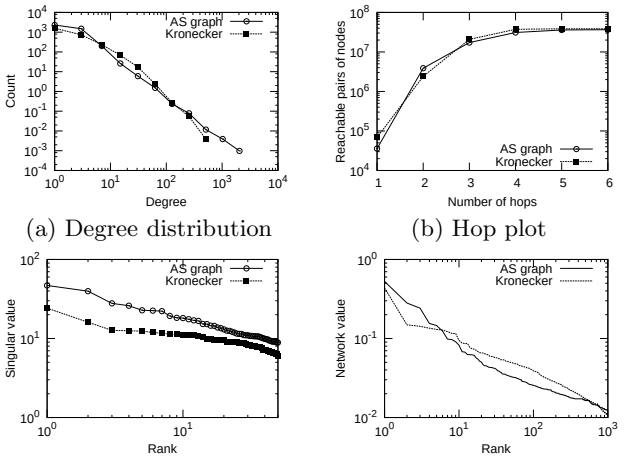
To obtain $P(G|\Theta, \sigma)$ we use Θ to create the Stochastic Kronecker graph adjacency matrix $\mathcal{P} = \Theta^{[k]}$. Permutation σ defines the mapping of nodes of G to the rows and columns of stochastic adjacency matrix $\mathcal{P}$. Modeling edges as binomial random variables we evaluate the likelihood:

$$P(G|\mathcal{P}, \sigma) = \prod_{(u,v)\in G} \mathcal{P}[\sigma_u, \sigma_v] \prod_{(u,v)\notin G} (1 - \mathcal{P}[\sigma_u, \sigma_v]), \tag{4}$$

where we denote σ_i as the i^{th} element of the permutation σ, and $\mathcal{P}[i, j]$ is the element at row i, and column j of matrix $\mathcal{P} = \Theta^{[k]}$. The products go over all edges present in graph G, and all edges missing from G.

As the problem is introduced we are summing over exponentially many permutations in equation 3. Third, the evaluation of equation 4 as it is written takes $O(N^2)$ and needs to be evaluated $N!$ times. So, naively calculating the likelihood takes $O(N!N^2)$.

In short, we [3] develop a linear $O(E)$ time algorithm for fitting Kronecker graphs. To get around the super-exponential sum over the permutations we use Metropolis sampling. When given a permutation we need to evaluate the likelihood. Here we exploit the structure of Kronecker multiplication. We calculate the likelihood of an empty graph (graph with no edges) in constant time and then just need to iterate over all the edges to correct the likelihood. This way we can evaluate the full likelihood (gradient) in time $O(E)$. We then use stochastic gradient descent to obtain parameters. For details see [3].



(a) Degree distribution (b) Hop plot

(c) Scree plot (d) "Network" value

Figure 2: *Autonomous Systems:* Overlayed patterns of real graph and the fitted Kronecker graph. Notice that the fitted Kronecker graph matches patterns of the real graph.

4. EXPERIMENTS

As the optimization problem is not convex [3] we check whether we can recover the parameters of a synthetic Kronecker graph. We generated 100 synthetic Kronecker graphs on 16,384 (2^{14}) nodes and 1.4 million edges, with a randomly chosen 2×2 parameter matrix Θ^*. For each of the 100 graphs we start gradient descent from a different random location Θ', and try to recover Θ^*. In 98% of the cases the descent converged to the true parameters.

Next, we take real graph G, find parameters $\hat{\Theta}$ using the tools we just developed, then generate synthetic graph K using $\hat{\Theta}$, and compare their statistical properties.

Figure 2 shows properties of Autonomous Systems graph, and compares them with the properties of a synthetic Kronecker graph generated using the fitted parameters $\hat{\Theta}$ of size 2×2. Notice that properties of both graphs match swell.

Autonomous Systems is undirected graph and the fitted parameter matrix $\hat{\Theta} = [.98, .58; .58, .06]$ is also symmetric. This means that without a priori biasing the fitting towards undirected graphs, the recovered parameters obey this. Fitting AS graph from a random set of parameters, performing gradient descent for 50 iterations and at each iteration sampling half a million permutations, took less than 20 minutes on a standard desktop PC. This is a significant speedup over [1], where by using a similar permutation sampling approach for just calculating the likelihood of a preferential attachment model on similar AS graph took two days on a cluster of 50 machines.

5. REFERENCES

[1] I. Bezáková, A. Kalai, and R. Santhanam. Graph model selection using maximum likelihood. In *ICML*, 2006.

[2] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *PKDD*, pages 133–145, 2005.

[3] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *ICML*, 2007.