Kronecker Graphs: An Approach to Modeling Networks

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, Zoubin Ghahramani


Presented by Yijiang Huang
2-14-2018
Introduction

• Graphs are everywhere

• What can we do with graphs?
  – What patterns or “laws” hold for most real-world graphs?
  – Can we build models of graph generation and evolution?
Outlines

- Introduction

- Network properties – static & temporal

- Proposed graph generation model - Kronecker graph

- Stochastic Kronecker graph

- Properties of Kronecker graph

- Model estimation

- Experimental results

- Discussion
Network properties – static & temporal

- **Power Law degree distributions**

![Internet in December 1998 graph](image)

\[ Y = a \cdot X^b \]

- Many low-degree nodes
- Few high-degree nodes
Network properties – static & temporal

• **Small-world**
  
  [Watts, Strogatz]++
  
  – 6 degrees of separation
  
  – Small diameter

• **Effective diameter:**
  
  – Distance at which 90% of pairs of nodes are reachable
Network properties – static & temporal

• Scree plot
  [Chakrabarti et al]
  – Eigenvalues of graph adjacency matrix follow a power law
  – Network values (components of principal eigenvector) also follow a power-law
Network properties – static & temporal

• Conventional Wisdom:
  – **Constant average degree**: the number of edges grows linearly with the number of nodes
  – **Slowly growing diameter**: as the network grows the distances between nodes grow

• “Recently” found [Leskovec, Kleinberg and Faloutsos, 2005]:
  – **Densification Power Law**: networks are becoming denser over time
  – **Shrinking Diameter**: diameter is decreasing as the network grows
Network properties – static & temporal - Densification

- **Densification Power Law**
  - $N(t)$ … nodes at time $t$
  - $E(t)$ … edges at time $t$

- **Suppose that**
  $$N(t+1) = 2 * N(t)$$

- **Q: what is your guess for**
  $$E(t+1) = 2 * E(t)$$

- **A: over-doubled!**
  - But obeying the Densification Power Law

![Densification Power Law Graph](image)
Network properties – static & temporal - Densification

• **Densification Power Law**
  – networks are becoming denser over time
  – the number of edges grows faster than the number of nodes – average degree is increasing

\[
E(t) \propto N(t)^a
\]

• **Densification exponent a**: \(1 \leq a \leq 2\):
  – \(a=1\): linear growth – constant out-degree
    (assumed in the literature so far)
  – \(a=2\): quadratic growth – clique
Network properties – static & temporal – shrinking diameter

• Prior work on Power Law graphs hints at Slowly growing diameter:
  – diameter ~ O(log N)
  – diameter ~ O(log log N)

• Diameter Shrinks/Stabilizes over time
  – As the network grows the distances between nodes slowly decrease
Network properties – These Patterns hold in many graphs

- All these patterns can be observed in many real life graphs:
  - World wide web [Barabasi]
  - On-line communities [Holme, Edling, Liljeros]
  - Who call whom telephone networks [Cortes]
  - Autonomous systems [Faloutsos, Faloutsos, Faloutsos]
  - Internet backbone – routers [Faloutsos, Faloutsos, Faloutsos]
  - Movie – actors [Barabasi]
  - Science citations [Leskovec, Kleinberg, Faloutsos]
  - Co-authorship [Leskovec, Kleinberg, Faloutsos]
  - Sexual relationships [Liljeros]
  - Click-streams [Chakrabarti]
Problem Definition

• Given a growing graph with nodes $N_1, N_2, \ldots$
• Generate a realistic sequence of graphs that will obey all the patterns
  – Static Patterns
    • Power Law Degree Distribution
    • Small Diameter
    • Power Law eigenvalue and eigenvector distribution (scree plot)
  – Dynamic Patterns
    • Growth Power Law
    • Shrinking/Constant Diameters
  – And ideally we would like to prove them
Previous work

• Lots of work
  – Random graph [Erdos and Renyi, 60s]
  – Preferential Attachment [Albert and Barabasi, 1999]
  – Copying model [Kleinberg, Kumar, Raghavan, Rajagopalan and Tomkins, 1999]
  – Community Guided Attachment and Forest Fire Model [Leskovec, Kleinberg and Faloutsos, 2005]
  – Also work on Web graph and virus propagation [Ganesh et al, Satorras and Vespignani]++

• But all of these
  – Do not obey all the patterns
  – Or we are not able prove them
Why is all this important?

- **Simulations** of new algorithms where real graphs are impossible to collect

- **Predictions** – predicting future from the past

- **Graph sampling** – many real world graphs are too large to deal with

- **What-if scenarios**
Main contribution

1. The authors propose a generative network model called the *Kronecker graph* that obeys all the static and some temporal network patterns exhibited in real work graphs.

2. The authors present a fast and scalable algorithm for fitting Kronecker graph generation model to large real networks.
Outlines

- Introduction

- Network properties – static & temporal

- Proposed graph generation model - Kronecker graph

- Stochastic Kronecker graph

- Properties of Kronecker graph

- Model estimation

- Experimental results

- Discussion
Problem definition

Given a growing graph with count of nodes $N_1, N_2, ...$
Generate a realistic sequence of graphs that will obey all the patterns

Idea: **Self-similarity**

Leads to power laws (degree distributions)
Communities within communities

...
Recursive graph generation

- There are many obvious (but wrong) ways

There are many obvious (but wrong) ways

\[ X_1 \]
\[ X_2 \]
\[ X_3 \]

Initial graph

- Does not obey Densification Power Law
- Has increasing diameter

Kronecker Product is a way of generating self-similar matrices
Kronecker product: Graph

Intermediate stage

\[
\begin{bmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{bmatrix} \quad \text{Adjacency matrix (3x3)}
\]

\[
\begin{bmatrix}
G_1 & G_1 & 0 \\
G_1 & G_1 & G_1 \\
0 & G_1 & G_1 \\
\end{bmatrix} \quad \text{Adjacency matrix (9x9)}
\]

\[G_2 = G_1 \otimes G_1\]
Kronecker product: Definition

• The Kronecker product of matrices $A$ and $B$ is given by

$$C = A \otimes B$$

$$N \times M \quad K \times L$$

$$\begin{pmatrix}
    a_{1,1}B & a_{1,2}B & \ldots & a_{1,m}B \\
    a_{2,1}B & a_{2,2}B & \ldots & a_{2,m}B \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{n,1}B & a_{n,2}B & \ldots & a_{n,m}B
\end{pmatrix}$$

$$N*K \times M*L$$

• We define a Kronecker product of two graphs as a Kronecker product of their adjacency matrices
Kronecker graphs

• We create the self-similar graphs recursively
  – Start with a initiator graph $G_1$ on $N_1$ nodes and $E_1$ edges
  – The recursion will then produce larger graphs $G_2$, $G_3$, ...
    $G_k$ on $N_1^k$ nodes

• We obtain a growing sequence of graphs by iterating the Kronecker product

$$G_k = G_1 \otimes G_1 \otimes \ldots G_1$$

$k$ times
Kronecker graphs

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on ...

\[
\begin{array}{ccc}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{array}
\]

$G_1$

$G_3$ adjacency matrix
Kronecker graphs – examples on bipartite graph

\[ P \]

Equal with the right permutation

\[
\begin{align*}
B(5,1) \otimes B(3,1) & \overset{P}{=} B(15,1) \cup B(3,5)
\end{align*}
\]

- Fundamental result [Weischel 1962] is that the Kronecker product of two complete bipartite graphs is two complete bipartite graphs
- More generally

\[
B(n_1, m_1) \otimes B(n_2, m_2) \overset{P}{=} B(n_1n_2, m_1m_2) \cup B(n_2m_1, n_1m_2)
\]
Kronecker graphs – examples on bipartite graph

Kronecker exponent of bipartite graph naturally produces exponential distribution
Outlines

- Introduction
- Network properties – static & temporal
- Proposed graph generation model - Kronecker graph
- Stochastic Kronecker graph
- Properties of Kronecker graph
- Model estimation
- Experimental results
- Discussion
Stochastic Kronecker graphs

\[
\begin{array}{ccc}
\rho_{ij} & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{array}
\]

\(G_1\) (3x3)

\[
\begin{array}{ccc}
G_1 & G_1 & 0 \\
G_1 & G_1 & G_1 \\
0 & G_1 & G_1 \\
\end{array}
\]

\(G_2 = G_1 \otimes G_1\) (9x9)

For each \(p_{uv}\) flip Bernoulli coin

\[
\Theta_1 = \begin{pmatrix}
0.5 & 0.2 \\
0.1 & 0.3 \\
\end{pmatrix}
\]

Kronecker multiplication

\[
\begin{pmatrix}
0.25 & 0.10 & 0.10 & 0.04 \\
0.05 & 0.15 & 0.02 & 0.06 \\
0.05 & 0.02 & 0.15 & 0.06 \\
0.01 & 0.03 & 0.03 & 0.09 \\
\end{pmatrix}
\]

\(\Theta_2 = \Theta_1 \otimes \Theta_1\)

Instance matrix \(K_2\)

Probability of edge \(p_{uv}\)
Outlines

- Introduction

- Network properties – static & temporal

- Proposed graph generation model - Kronecker graph

- Stochastic Kronecker graph

- Properties of Kronecker graph

- Model estimation

- Experimental results

- Discussion
Kronecker graphs: Intuition

1) **Recursive growth of graph communities**
   - Nodes get expanded to micro communities
   - Nodes in sub-community link among themselves and to nodes from different communities

2) **Node attribute representation**
   - Nodes are described by features
     - [likes ice cream, likes chocolate]
     - $u=[1,0], \ v=[1, 1]$
   - Parameter matrix gives the linking probability
     - $p(u,v) = 0.5 \times 0.1 = 0.05$
Properties of Kronecker graphs

- **We prove** that Kronecker multiplication generates graphs that obey [PKDD’05]
  - Properties of static networks
    - ✔ Power Law Degree Distribution
    - ✔ Power Law eigenvalue and eigenvector distribution
    - ✔ Small Diameter
  - Properties of dynamic networks
    - ✔ Densification Power Law
    - ✔ Shrinking/Stabilizing Diameter
- **Good news:** Kronecker graphs have the necessary **expressive power**
- **But:** How do we choose the parameters to **match** all of these at once?
Outlines

- Introduction
- Network properties – static & temporal
- Proposed graph generation model - Kronecker graph
- Stochastic Kronecker graph
- Properties of Kronecker graph
- Model estimation
- Experimental results
- Discussion
Model estimation: approach

• Maximum likelihood estimation
  – Given real graph $G$
  – Estimate Kronecker initiator graph $\Theta$ (e.g., \[
  \begin{array}{ccc}
  1 & 1 & 0 \\
  1 & 1 & 1 \\
  0 & 1 & 1 \\
  \end{array}
  \])
  which

  \[
  \arg \max_{\Theta} P(G \mid \Theta)
  \]

• We need to (efficiently) calculate

  $P(G \mid \Theta)$

• And maximize over $\Theta$ (e.g., using gradient descent)
Fitting Kronecker graphs

- Given a graph $G$ and Kronecker matrix $\Theta$ we calculate probability that $\Theta$ generated $G$, $P(G|\Theta)$

$$P(G|\Theta) = \prod_{(u,v) \in G} \Theta_k[u,v] \prod_{(u,v) \notin G} (1 - \Theta_k[u,v])$$

| $\Theta$  | $\Theta_k$  | $P(G|\Theta)$ | $G$ |
|----------|-------------|----------------|-----|
| 0.5 0.2  | 0.25 0.10 0.10 0.04 |                |     |
| 0.1 0.3  | 0.05 0.15 0.02 0.06 |                |     |
| 1.0 1.1  |                |                |     |
| 0.05 0.02 0.15 0.06 |                |                |     |
| 0.01 0.03 0.03 0.09 |                |                |     |
| 1.0 1.0 1.0 1.0 |                |                |     |
| 1.1 1.1 1.1 1.1 |                |                |     |
Challenge 1: Node correspondence

- Nodes are unlabeled
- Graphs $G'$ and $G''$ should have the same probability $P(G'|\Theta) = P(G''|\Theta)$
- One needs to consider all node correspondences $\sigma$

$$P(G|\Theta) = \sum_{\sigma} P(G|\Theta, \sigma) P(\sigma)$$

- All correspondences are a priori equally likely
- There are $O(N!)$ correspondences
Challenge 2: calculating $P(G|\Theta, \sigma)$

- Assume we solved the correspondence problem
- Calculating

$$P(G | \Theta) = \prod_{(u,v) \in G} \Theta_k [\sigma_u, \sigma_v] \prod_{(u,v) \notin G} (1 - \Theta_k [\sigma_u, \sigma_v])$$

- Takes $O(N^2)$ time
- Infeasible for large graphs ($N \sim 10^5$)

\[ \begin{array}{cccc} 0.25 & 0.10 & 0.10 & 0.04 \\ 0.05 & 0.15 & 0.02 & 0.06 \\ 0.05 & 0.02 & 0.15 & 0.06 \\ 0.01 & 0.03 & 0.03 & 0.09 \end{array} \]

\[ \begin{array}{cccc} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{array} \]
Model estimation: solution

- Naïvely estimating the Kronecker initiator takes $O(N!N^2)$ time:
  - $N!$ for graph isomorphism
    - Metropolis sampling: $N! \rightarrow (\text{big}) \text{ const}$
  - $N^2$ for traversing the graph adjacency matrix
    - Properties of Kronecker product and sparsity ($E \ll N^2$): $N^2 \rightarrow E$

- We can estimate the parameters of Kronecker graph in linear time $O(E)$
Solution 1: Node correspondence

- Log-likelihood

\[ l(\Theta) = \log \sum_{\sigma} P(G|\Theta, \sigma) P(\sigma) \]

- Gradient of log-likelihood

\[ \frac{\partial}{\partial \Theta} l(\Theta) = \sum_{\sigma} \frac{\partial \log P(G|\sigma, \Theta)}{\partial \Theta} P(\sigma|G, \Theta) \]

- **Sample** the permutations from \( P(\sigma|G, \Theta) \) and average the gradients
Solution 1: Node correspondence

- **Metropolis sampling:**
  - Start with a random permutation
  - Do local moves on the permutation
  - Accept the new permutation
    - If new permutation is better (gives higher likelihood) \( P(\sigma \mid G, \Theta) \)
    - If new is worse accept with probability proportional to the ratio of likelihoods

\[
\begin{array}{cccc}
1 & 1 & 0 & 1 \\
2 & 0 & 1 & 1 \\
3 & 1 & 1 & 1 \\
4 & 0 & 1 & 1 \\
\end{array}
\]

Swap node labels 1 and 4

\[
\begin{array}{cccc}
4 & 1 & 1 & 1 \\
2 & 1 & 1 & 0 \\
3 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
\end{array}
\]

Re-evaluate the likelihood

Can compute efficiently:
Only need to account for changes in 2 rows / columns
Solution 2: Calculating $P(G|\Theta,\sigma)$

- Calculating naively $P(G|\Theta,\sigma)$ takes $O(N^2)$
- Idea:
  - First calculate likelihood of empty graph, a graph with 0 edges
  - Correct the likelihood for edges that we observe in the graph

- By exploiting the structure of Kronecker product we obtain closed form for likelihood of an empty graph
**Solution 2: Calculating $P(G|\Theta,\sigma)$**

- We approximate the likelihood:

$$l(\Theta) \approx l_e(\Theta) + \sum_{(u,v) \in G} -\log(1 - \Theta_k[\sigma_u, \sigma_v]) + \log(\Theta_k[\sigma_u, \sigma_v])$$

- The sum goes only over the edges
- Evaluating $P(G|\Theta,\sigma)$ takes $O(E)$ time
- Real graphs are sparse, $E \ll N^2$
Model estimation: overall solution

input : size of parameter matrix \( N_1 \), graph \( G \) on \( N = N_1^k \) nodes, and learning rate \( \lambda \)
output: MLE parameters \( \hat{\Theta} \) (\( N_1 \times N_1 \) probability matrix)

1. initialize \( \hat{\Theta}_1 \)
2. while not converged do
   3. evaluate gradient: \( \frac{\partial}{\partial \hat{\Theta}} l(\hat{\Theta}_t) \)
   4. update parameter estimates: \( \hat{\Theta}_{t+1} = \hat{\Theta}_t + \lambda \frac{\partial}{\partial \hat{\Theta}} l(\hat{\Theta}_t) \)
3. end
4. return \( \hat{\Theta} = \hat{\Theta}_t \)

input : Parameter matrix \( \Theta \), and graph \( G \)
output: Log-likelihood \( l(\Theta) \), and gradient \( \frac{\partial}{\partial \Theta} l(\Theta) \)

1. for \( t := 1 \) to \( T \) do
   2. \( \sigma_t := \text{SamplePermutation}(G, \Theta) \)
   3. \( l_t = \log P(G|\sigma(t), \Theta) \)
   4. \( \text{grad}_t := \frac{\partial}{\partial \Theta} \log P(G|\sigma(t), \Theta) \)
5. end
6. return \( l(\Theta) = \frac{1}{T} \sum_t l_t \), and \( \frac{\partial}{\partial \Theta} l(\Theta) = \frac{1}{T} \sum_t \text{grad}_t \)

\textbf{Solution 1}: Metropolis sampling

\textbf{Solution 2}: Edge-wise prob computation
Outlines

- Introduction

- Network properties – static & temporal

- Proposed graph generation model - Kronecker graph

- Stochastic Kronecker graph

- Properties of Kronecker graph

- Model estimation

- Experimental results

- Discussion
Experiments: artificial data

• Can gradient descent recover true parameters?

• Optimization problem is not convex

• How nice (without local minima) is optimization space?
  – Generate a graph from random parameters
  – Start at random point and use gradient descent
  – We recover true parameters 98% of the times
Convergence of properties

- How does algorithm converge to true parameters with gradient descent iterations?
Experiments: real networks

- Experimental setup:
  - Given real graph
  - Stochastic gradient descent from random initial point
  - Obtain estimated parameters
  - Generate synthetic graphs
  - Compare properties of both graphs

- We do not fit the properties themselves

- We fit the likelihood and then compare the graph properties
AS graph (N=6500, E=26500)

- Autonomous systems (internet)
- We search the space of $\sim 10^{50,000}$ permutations
- Fitting takes 20 minutes
- AS graph is undirected and estimated parameter matrix is symmetric:

\[
\begin{array}{cc}
0.98 & 0.58 \\
0.58 & 0.06 \\
\end{array}
\]
AS: comparing graph properties

- Generate synthetic graph using estimated parameters
- Compare the properties of two graphs

**Degree distribution**

- Log count vs. log degree
- AS graph and Kronecker graph

**Hop plot**

- Log # of reachable pairs vs. number of hops
- Diameter = 4
AS: comparing graph properties

- Spectral properties of graph adjacency matrices

Scree plot

\[ \text{log eigenvalue} \]

\[ \text{log rank} \]

Network value

\[ \text{log value} \]

\[ \text{log rank} \]
Epinions graph (N=76k, E=510k)

- We search the space of $\sim 10^{1,000,000}$ permutations
- Fitting takes 2 hours
- The structure of the estimated parameter gives insight into the structure of the graph

<table>
<thead>
<tr>
<th>Hop plot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log # of reachable pairs</td>
</tr>
<tr>
<td>Log degree</td>
</tr>
</tbody>
</table>

degree distribution

<table>
<thead>
<tr>
<th>degree</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^0</td>
<td>10^4</td>
</tr>
<tr>
<td>10^1</td>
<td>10^2</td>
</tr>
<tr>
<td>10^2</td>
<td>10^1</td>
</tr>
<tr>
<td>10^3</td>
<td>10^0</td>
</tr>
<tr>
<td>10^4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Epinions</th>
<th>Kronecker</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td>0.54</td>
</tr>
<tr>
<td>0.49</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Epinions graph (N=76k, E=510k)

- Scree plot
  - Log eigenvalue vs. log rank
  - Graph showing the distribution of eigenvalues for Epinions and Kronecker graphs.

- Network value
  - Log network value vs. log rank
  - Graph showing the network value decay for Epinions and Kronecker graphs.
### Scalability

- Fitting scales **linearly** with the number of edges

![Graph showing linear fit](image-url)
Outlines

- Introduction
- Network properties – static & temporal
- Proposed graph generation model - Kronecker graph
- Stochastic Kronecker graph
- Properties of Kronecker graph
- Model estimation
- Experimental results
- Discussion
Conclusion

• Kronecker Graph model has
  – provable properties
  – small number of parameters
• We developed scalable algorithms for fitting Kronecker Graphs
• We can efficiently search large space ($\sim 10^{1,000,000}$) of permutations
• Kronecker graphs fit well real networks using few parameters
• We match graph properties without a priori deciding on which ones to fit
Discussion

- Network evolution
  - Dynamic Bayesian network with first order Markov dependencies
  - A series of network snapshot evolving over time -> evolving initiator matrix
  - Deeper understanding of network evolution through the lens of generating parameters

- Different random process for stochastic Kronecker graph
  - Currently Bernoulli edge generation model
  - Modeling weighted or labelled networks

- Micro-scale network probe?
  - Random Dot Product Graphs – estimate the individual attribute values
  - Kronecker (product) graphs – attribute-attribute similarity matrix (initiator matrix)
  - Try to use given node attributes to infer “hidden” or missing node attribute values
References


- C++ implementation: Stanford Network Analysis Platform (SNAP):  
  - [https://github.com/snap-stanford/snap](https://github.com/snap-stanford/snap)  
  - general purpose network analysis and graph mining library, with Kronecker graph modeling functionality included.