Announcements:
1) HW3: Due 9:30am Friday 11/7
   Snap site was down (Gates network problems)
   We give you 1 extra day if you are using a late period you still have to submit by Tue 9:30am)
2) HW4 has been posted
3) Project milestones due in 1 week
   We expect ~50% project completed
   Also no late periods

Kronecker Graphs
Macroscopic Evolution of Networks
How do networks evolve at the macro level?

- What are global phenomena of network growth?

Questions:

- What is the relation between the number of nodes $n(t)$ and number of edges $e(t)$ over time $t$?
- How does diameter change as the network grows?
- How does degree distribution evolve as the network grows?
Network Evolution

- $N(t)$ ... nodes at time $t$
- $E(t)$ ... edges at time $t$
- Suppose that
  \[ N(t + 1) = 2 \cdot N(t) \]
- Q: what is:
  \[ E(t + 1) = ? \quad \text{Is it } 2 \cdot E(t)? \]
- A: More than doubled!
  - But obeying the Densification Power Law
Q1) Network Evolution

- What is the relation between the number of nodes and the edges over time?
- First guess: constant average degree over time
- Networks are denser over time
- Densification Power Law:

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

\( a \) ... densification exponent (1 ≤ a ≤ 2)
Densification Power Law

- the number of edges grows faster than the number of nodes – **average degree is increasing**

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

or equivalently

\[ \frac{\log(E(t))}{\log(N(t))} = \text{const} \]

- densification exponent: \(1 \leq a \leq 2\):
  - \(a=1\): linear growth – constant out-degree (traditionally assumed)
  - \(a=2\): quadratic growth – fully connected graph
Prior models and intuition say that the network diameter slowly grows (like \( \log N \)).

Diameter shrinks over time:
- As the network grows, the distances between the nodes slowly decrease.

How do we compute diameter in practice?
- **Long paths**: Take 90\(^{th}\)-percentile or average path length (not the maximum)
- **Disconnected components**: Take only largest component or average only over connected pairs of nodes.
Diameter of a Densifying $G_{np}$

Is shrinking diameter just a consequence of densification?

Densifying random graph has increasing diameter

$\Rightarrow$ There is more to shrinking diameter than just densification!
Is it the degree sequence?

Compare diameter of a:

- Real network (red)
- Random network with the same degree distribution (blue)

Densification + degree sequence gives shrinking diameter
How does degree distribution evolve to allow for densification?

- **Option 1)** Degree exponent $\gamma_t$ is constant:
  - **Fact 1:** If $\gamma_t = \gamma \in [1, 2]$, then: $\alpha = 2/\gamma$

A consequence of what we learned in the Power law lecture:
- Power-laws with exponents <2 have infinite expectations.
- So, by maintaining constant degree exponent $\alpha$ the average degree grows.
- How does degree distribution evolve to allow for densification?
- Option 2) $\gamma_t$ evolves with graph size $n$:
  - **Fact 2:** If $\gamma_t = \frac{4n^{x-1}-1}{2n^{x-1}1}$, then: $\alpha = x$

Notice: $\gamma_t \rightarrow 2$ as $n_t \rightarrow \infty$

Remember, the expected degree in a power law is:

$$E[X] = \frac{\gamma_t - 1}{\gamma_t - 2} x_m$$

So $\gamma_t$ has to decay as a function of graph size $n_t$ for the avg. degree to go up.
Kronecker Graphs Model
Models of Networks

- What is the goal of modeling networks?
  - Discover structural properties of networks
    - Small-world, Edge clustering, Heavy-tailed degrees
  - Find a model that gives graphs with such properties
    - Erdos-Renyi, Watts-Strogatz, Barabasi-Albert model

- Today’s lecture:
  - Can we have a model that attempts to reproduce all of these properties?
  - Can we fit the model to a network and accurately reproduce the network?
How can we think of network structure recursively? **Intuition:** Self-similarity

- **Object is similar to a part of itself:** the whole has the same shape as one or more of the parts

**Mimic recursive graph/community growth:**

- **Kronecker graph** is a way of generating self-similar matrices
Kronecker: Graph Growth

**Intermediate stage**

**Initiator graph**

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{pmatrix}
\]  

(3x3)

**After the growth phase**

\[
K_2 = K_1 \otimes K_1
\]  

(9x9)
Kronecker graphs:

- A recursive model of network structure

\[ K_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \]

\[ K_2 = K_1 \otimes K_1 = \begin{pmatrix} K_1 & K_1 & 0 \\ K_1 & K_1 & K_1 \\ 0 & K_1 & K_1 \end{pmatrix} \]

3 x 3 \rightarrow 9 x 9 \rightarrow 81 x 81 adjacency matrix
Kronecker Product: Definition

- **Kronecker product** of matrices $A$ and $B$ is given by

$$C = A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,m}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}B & a_{n,2}B & \cdots & a_{n,m}B \end{pmatrix}$$

Size: $N \times M \times K \times L$ to $N \times K \times M \times L$

- Define a Kronecker product of two graphs as a Kronecker product of their adjacency matrices
Kronecker Graphs

- **Kronecker graph**: a growing sequence of graphs by iterating the Kronecker product:

\[ K_1^{[m]} = K_m = K_1 \otimes K_1 \otimes \ldots \otimes K_1 = K_{m-1} \otimes K_1 \]

- **Note**: One can easily use multiple initiator matrices \((K_1', K_1'', K_1''')\) (even of different sizes)
Kronecker Initiator Matrices

Initiator $K_1$

$K_1$ adjacency matrix

$K_3$ adjacency matrix
Kronecker Graphs: First Fun Fact

First fact about Kronecker Graphs!
- For $K_1$ on $N_1$ nodes and $E_1$ edges, $K_m$ ($m^{\text{th}}$ Kronecker power of $K_1$) has:
  - $N(m) = N_1^m$ nodes
  - $E(m) = E_1^m$ edges
- So, we get the densification power-law!
  - $E(t) \propto N(t)^a$, so: $E_1^t = (N_1^t)^a$ What is $a$?
  - $a = \frac{\log(E(t))}{\log(N(t))} = \frac{\log(E_1^t)}{\log(N_1^t)} = \frac{\log(E_1)}{\log(N_1)}$

Since $E(t) > N(t)$, then $a > 1$
Properties of deterministic Kronecker graphs (can be proved!)

- Properties of static networks:
  - Power-Law like Degree Distribution
  - Power-Law eigenvalue and eigenvector distribution
  - Constant Diameter

- Properties of evolving networks:
  - Densification Power Law *(just proved)*
  - Shrinking/Stabilizing Diameter (for Stochastic Kronecker graphs)
**Observation:** Edges in Kronecker graphs:

\[ \text{Edge} \ (X_{ij}, X_{kl}) \in G \otimes H \]

iff \( (X_i, X_k) \in G \) and \((X_j, X_l) \in H\)

where \( X \) are appropriate nodes in \( G \) and \( H \)

**Why?**

- An entry in matrix \( G \otimes H \) is a multiplication of entries in \( G \) and \( H \).
Theorem: **Constant diameter:** If graphs $G, H$ have diameter $d$ then $G \boxtimes H$ has diameter $d$

What is distance between nodes $u, v$ in $G \boxtimes H$?

- Consider some nodes $u = [a, b], v = [a', b']$ in $G \boxtimes H$
- Then, path $a$ to $a'$ in $G$ is less $d$ steps: $a_1, a_2, a_3, \ldots, a_d$
- And path $b$ to $b'$ in $H$ is less $d$ steps: $b_1, b_2, b_3, \ldots, b_d$
- **How many steps from $u$ to $v$?**
  - We know edge $([a_1, b_1], [a_2, b_2])$ is in $G \boxtimes H$
  - So it takes $< d$ steps to get from $u$ to $v$ in $G \boxtimes H$

**Consequence:**

- If $K_1$ has diameter $d$ then graph $K_k$ also has diameter $d$
Stochastic Kronecker Graphs
Stochastic Kronecker Graphs

- Create $N_1 \times N_1$ probability matrix $\Theta_1$
- Compute the $k^{th}$ Kronecker power $\Theta_k$
- For each entry $p_{uv}$ of $\Theta_k$ include an edge $(u, v)$ in $K_k$ with probability $p_{uv}$

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

Kronecker multiplication

Instance matrix $K_2$

Flip biased coins

Probability of edge $p_{uv}$
How do we generate an instance of a stochastic Kronecker graph?

<table>
<thead>
<tr>
<th>Probability of edge $p_{uv}$</th>
<th>Need to flip $n^2$ coins!! Way too slow!!</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.10</td>
</tr>
<tr>
<td>0.05</td>
<td>0.15</td>
</tr>
<tr>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>0.01</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Flip biased coins

Is there a faster way? YES!

Idea: Exploit the recursive structure of Kronecker graphs

“Drop” edges one by one
# Generation of Kronecker Graphs

- **A faster way to generate Kronecker graphs**

\[ \Theta = \begin{pmatrix}
  a & b \\
  c & d \\
\end{pmatrix} \]

- **How to “drop” an edge into a graph** \( G \) on \( n = 2^m \) nodes

\[
\begin{array}{c|c|c|c}
  & v_1 & v_2 & v_3 & v_4 \\
\hline
v_1 & a \cdot a & a \cdot b & b \cdot a & b \cdot b \\
\hline
v_2 & a \cdot c & a \cdot d & b \cdot c & b \cdot d \\
\hline
v_3 & c \cdot a & c \cdot b & d \cdot a & d \cdot b \\
\hline
v_4 & c \cdot c & c \cdot d & d \cdot c & d \cdot d \\
\end{array}
\]

\[
\begin{pmatrix}
  a & b \\
  c & d \\
\end{pmatrix}
\]

Adjacency matrix \( G \)
A faster way to generate Kronecker graphs

\[ \Theta = \begin{array}{cc} a & b \\ c & d \end{array} \]

How to “drop” an edge into a graph \( G \) on \( n = 2^m \) nodes

Adjacency matrix \( G \)
A faster way to generate Kronecker graphs

\[ \Theta = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \]

How to “drop” an edge into a graph \( G \) on \( n = 2^m \) nodes

Adjacency matrix \( G \)
A faster way to generate Kronecker graphs

How to “drop” an edge into a graph $G$ on $n = 2^m$ nodes:

- We may get a few edges colliding. We simply reinsert them.
Fast Kronecker generator algorithm:

- **Insert 1 edge on graph $G$ on $n = 2^m$ nodes:**
  - Create normalized matrix $L_{uv} = \Theta_{uv}/(\sum_{op} \Theta_{op})$
  - For $i = 1 \ldots m$
    - start with $x = 0, y = 0$
    - Pick an row/column $(u, v)$ with prob. $L_{uv}$
    - Descend into quadrant $(u, v)$ at level $i$ of $G$
      - This means: $x += u \cdot 2^{m-i}, y += v \cdot 2^{m-i}$
  - Add an edge $G[x, y] = 1$
Problem: Spikes in Node Degrees!

- SKG
- Noisy SKG (0.05)
- Nosiy SKG (0.10)
Solution: Noisy SKG

- **Solution: Noisy Stochastic Kronecker Graphs**
  - **Idea:** Add noise to the matrix $\Theta$
    - There are many ways how one could do this, but here is the correct way!
  - Assume $\Theta = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $G$ has $2^m$ nodes
  - **Then create “noisy” matrices** $\Theta_1 \ldots \Theta_m$ where:
    - $\Theta_i = \begin{bmatrix} a - \frac{2x_i a}{a+d} & b + x_i \\ c + x_i & d - \frac{2x_i d}{a+d} \end{bmatrix}$
    - Where $x_i$ is a random number on interval $[-X, +X]$
    - And $X$ is the noise level.
  - **Apply Kronecker generator to this set of matrices**
SKG Degree Distribution

- SKG, no noise
- Noisy SKG, X=0.1
Stochastic Kronecker Graphs

What is known about Stochastic Kronecker?

- **Undirected** Kronecker graph model with:
  - **Connected**, if:
    - \( b + c > 1 \)
  - **Connected component of size** \( \Theta(n) \), if:
    - \((a + b)(b + c) > 1\)
  - **Constant diameter**, if:
    - \( b + c > 1 \)
  - **Not searchable** by a decentralized algorithm

\[
\Theta_1 = \begin{bmatrix} a & b \\ b & c \end{bmatrix}
\]

\[a > b > c\]

[Mahdian-Xu, WAW ’07]
Kronecker Graphs: Estimation

How to estimate $\Theta$ given a $G$?

- **KronFit**: Maximum likelihood estimation
- Given real graph $G$
- Find Stochastic Kronecker initiator $\Theta$ which

$$\Theta = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\text{arg max}_{\Theta} P(G | \Theta)$$

To solve this we need to:

- Efficiently calculate $P(G | \Theta)$
- Then maximize over $\Theta$ (e.g., using gradient descent)
Given $G$ and $\Theta$ we calculate likelihood that $\Theta$ generated $G$: $P(G|\Theta)$

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

Likelihood of edges in the graph \quad Likelihood of edges not in the graph

<table>
<thead>
<tr>
<th>$\Theta_k$</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.10</td>
<td>0.10</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.15</td>
<td>0.02</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.02</td>
<td>0.15</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.09</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$G$</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Challenge 1: Node Correspondence

- Nodes are unlabeled
- Graphs $G'$ and $G''$ should have the same likelihood $P(G' | \Theta) = P(G'' | \Theta)$
- One needs to consider all node correspondences $\sigma$
- All correspondences are a priori equally likely
- There are $O(n!)$ correspondences

$P(G' | \Theta) = P(G'' | \Theta)$

$$
\begin{array}{cccc}
0.25 & 0.10 & 0.10 & 0.04 \\
0.05 & 0.15 & 0.02 & 0.06 \\
0.05 & 0.02 & 0.15 & 0.06 \\
0.01 & 0.03 & 0.03 & 0.09 \\
\end{array}
$$
Assume that we solved the node correspondence problem

Calculating:

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

Takes \(O(n^2)\) time!
Node correspondence:
- Node permutation $\sigma$ defines the mapping
- Randomly search over $\sigma$ to find good mappings

Calculating the likelihood $P(G|\Theta, \sigma)$
- Calculate likelihood of empty graph ($G$ with 0 edges)
- Correct it for edges that we observe in the graph

Details in Leskovec-Faloutsos, ICML '07

The algorithm (called Metropolis sampling):
1. Pick 2 nodes at random
2. Swap their IDs
3. Does it improve the fit $P(G|\Theta, \sigma)$? If, yes, keep the swap, else undo it
4. Go to (1)
Experiments: real networks

- **Experimental setup**
  - Given real graph $G$
  - Estimate parameters $\Theta$
  - Generate synthetic graph $K$ using $\Theta$
  - Compare properties of graphs $G$ and $K$

- **Note:**
  - We do not fit the graph properties themselves
  - We fit the likelihood and then compare the properties
Real and Kronecker are very close:

$$\Theta_1 = \begin{bmatrix} 0.99 & 0.54 \\ 0.49 & 0.13 \end{bmatrix}$$
What do estimated parameters tell us about the network structure?
What do estimated parameters tell us about the network structure?

\[ \Xi = \begin{pmatrix} 0.9 & 0.5 \\ 0.5 & 0.1 \end{pmatrix} \]
Small and large networks are very different:

\[ \Theta = \begin{bmatrix} 0.99 & 0.17 \\ 0.17 & 0.82 \end{bmatrix} \quad \Theta' = \begin{bmatrix} 0.99 & 0.54 \\ 0.49 & 0.13 \end{bmatrix} \]
Large scale network structure:

- **Nested Core-periphery**
  - Recursive onion-like structure of the network where each layer decomposes into a core and periphery
Implications (2)

- Remember the SKG theorems:
  - **Connected**, if $b+c > 1$:
    - $0.55 + 0.15 > 1$. No!
  - **Giant component**, if $(a+b) \cdot (b+c) > 1$:
    - $(0.99 + 0.55) \cdot (0.55 + 0.15) > 1$. Yes!
- Real graphs are in the parameter region analogous to the giant component of an extremely sparse $G_{np}$.