Kronecker Graphs:
An Approach to Modeling Networks

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

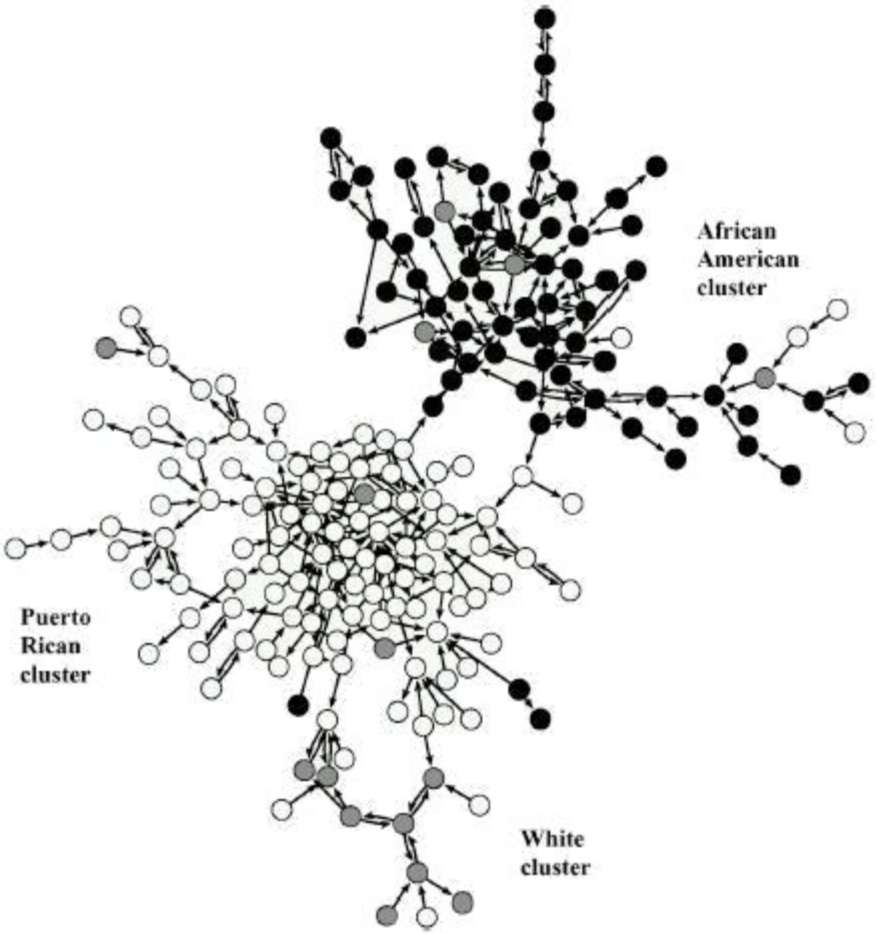
$$\otimes$$

Adapted from Leskovec et al. 2010, Leskovec's PhD dissertation (2008),
Eric Wang's slide (2011, Duke), Prof. Jeremy Kepner's slider on OpenCourseware (2012)

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**?



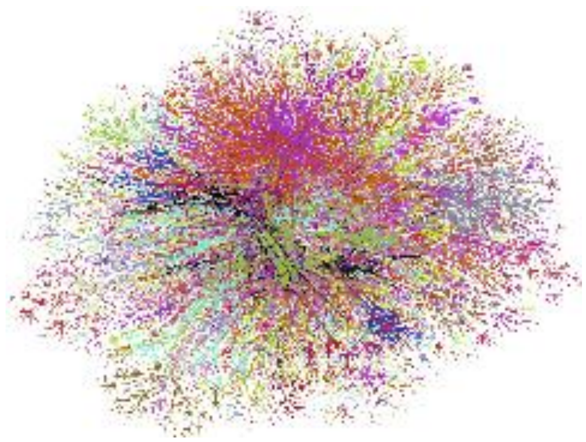
“Needle exchange” networks of drug users

Outlines

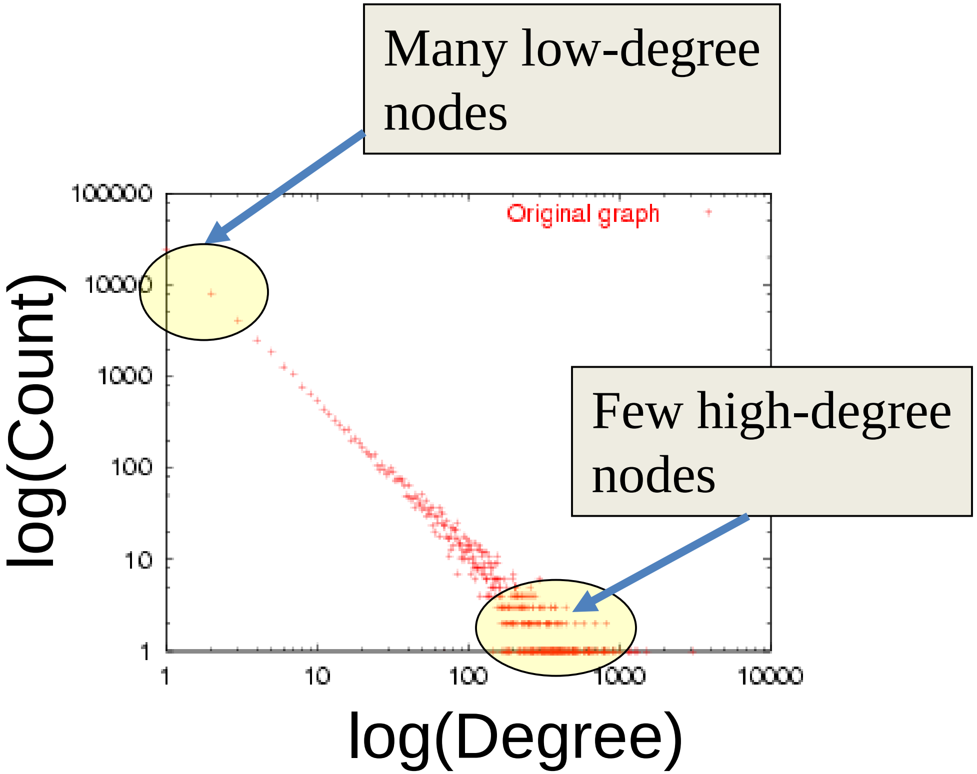
- Introduction
- Network properties – static & temporal
- Proposed graph generation model - Kronecker graph
- Stochastic Kronecker graph
- Properties of Kronecker graph
- Model estimation
- Experimental results
- Discussion

Network properties – **static** & temporal

- **Power Law degree distributions**



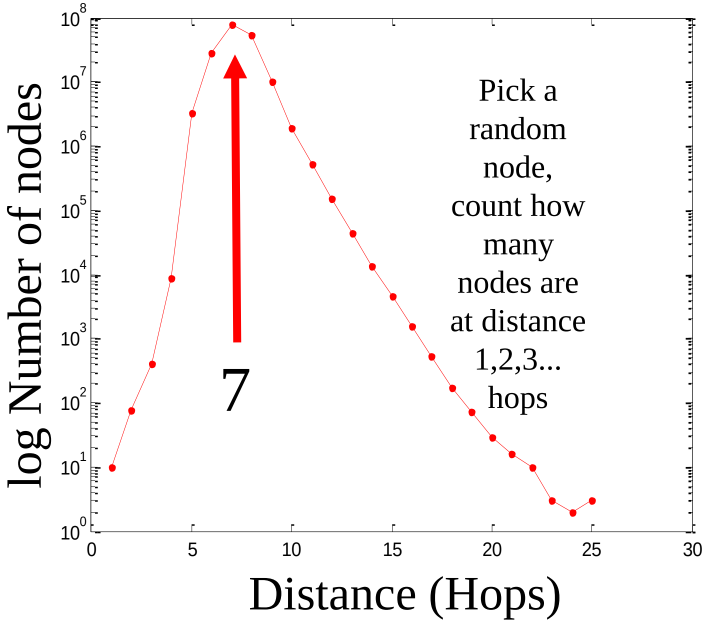
Internet in December 1998



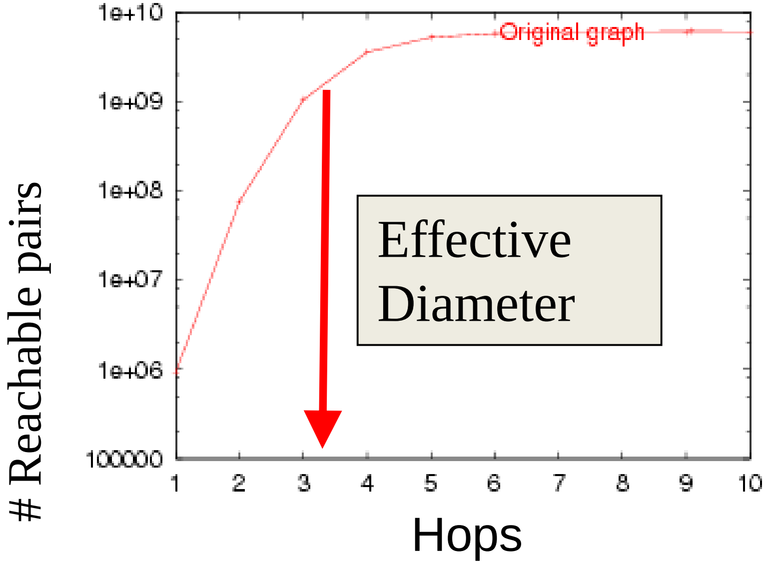
$$Y=a*X^b$$

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



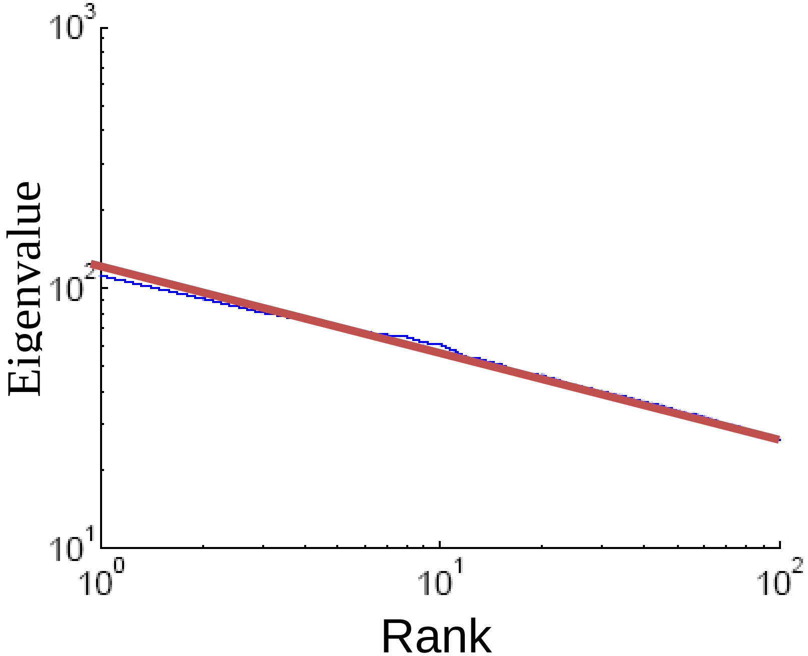
Network properties – **static** & temporal

- **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



Scree Plot

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

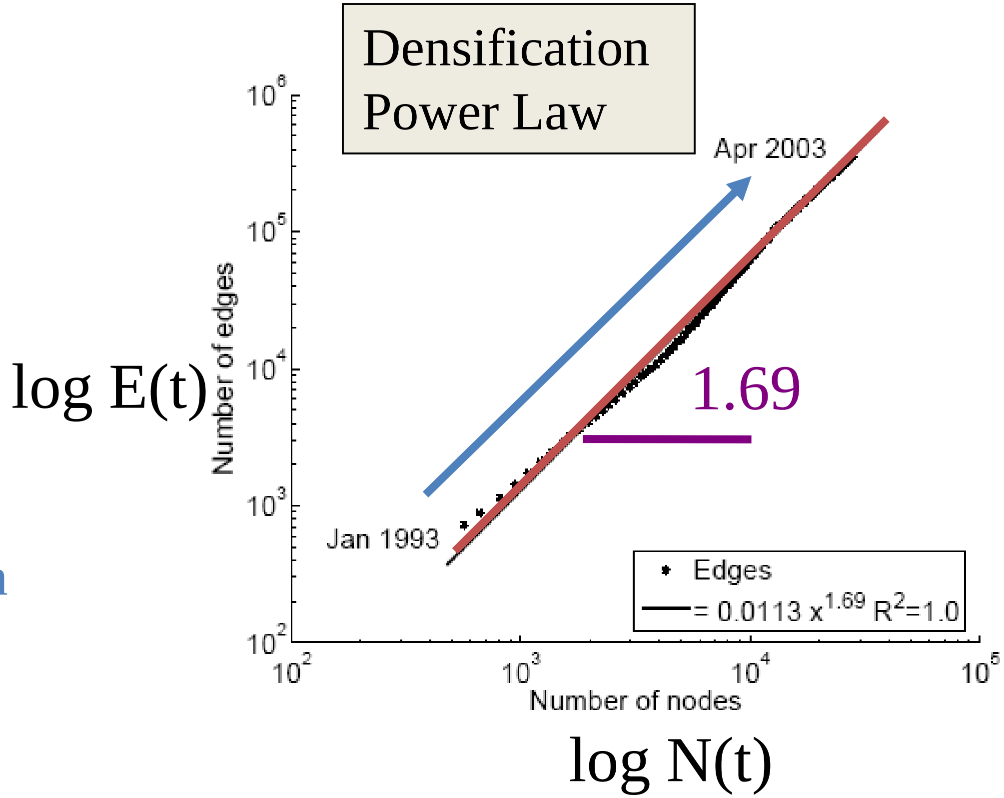
- “Recently” found [Leskovec, Kleinberg and Faloutsos, 2005]:
  - Densification Power Law: networks are becoming denser over time
  - Shrinking Diameter: diameter is decreasing as the network grows

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



Network properties – static & **temporal - Densification**

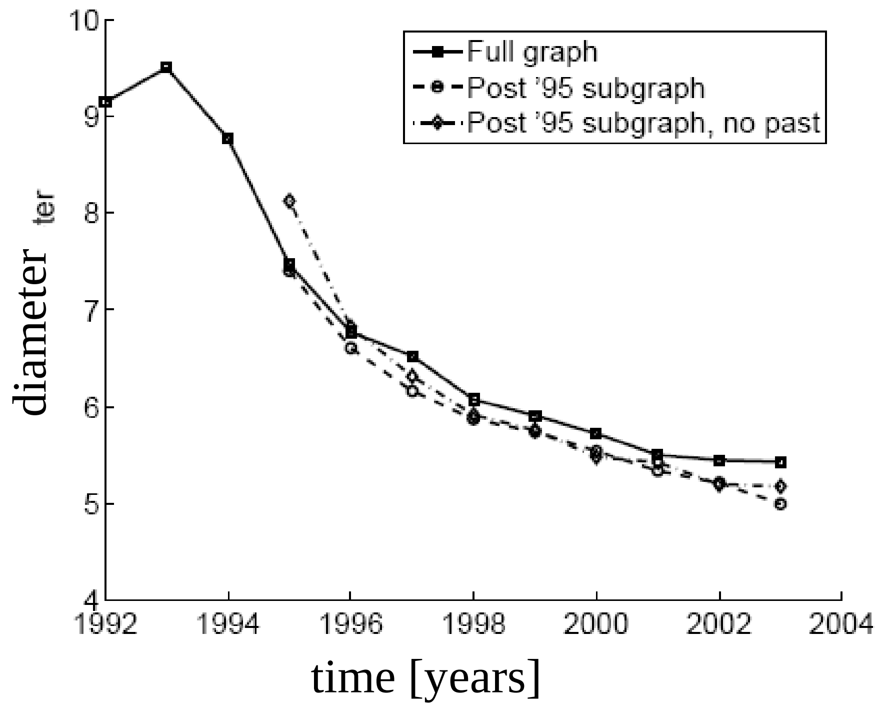
- Densification Power Law
  - networks are becoming denser over time
  - the number of edges grows faster than the number of nodes – average degree is increasing

$$\boxed{E(t) \propto N(t)^a}$$

- 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



Diameter over time

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, ...
- 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

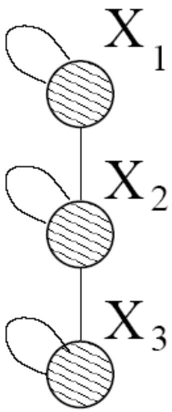
Idea: Self-similarity

- Leads to power laws (degree distributions)
- Communities within communities
- …

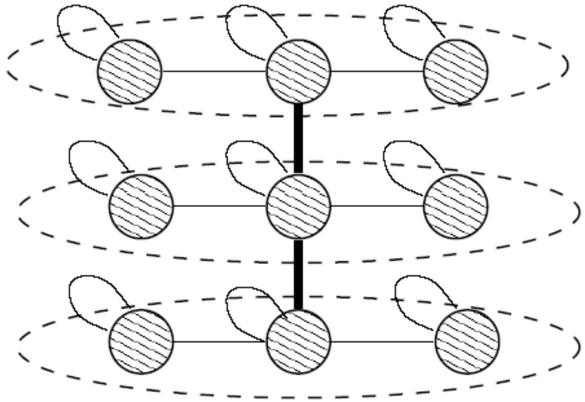
Recursive graph generation

- There are many obvious (but wrong) ways

There are many obvious (but wrong) ways



Initial graph



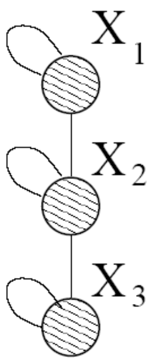
Recursive expansion

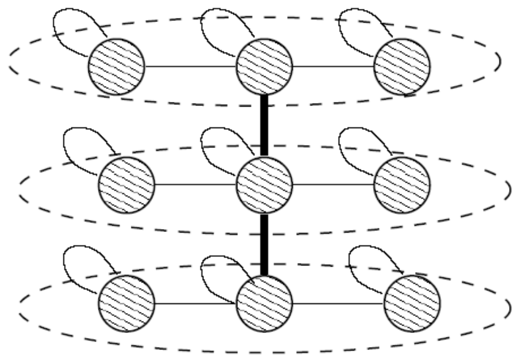
- Does not obey Densification Power Law

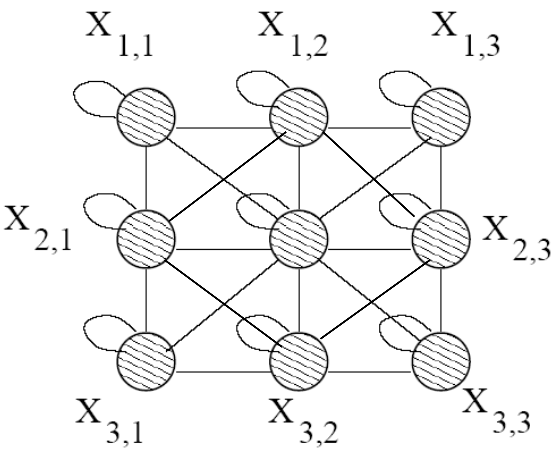
- Has increasing diameter

Kronecker Product is a way of generating self-similar matrices

Kronecker product: Graph







Intermediate stage

1	1	0
1	1	1
0	1	1

(3x3)

$$G_1$$

Adjacency matrix

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

(9x9)

$$G_2 = G_1 \otimes G_1$$

Adjacency matrix

Kronecker product: Definition

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

$$\underset{N \times M}{\mathbf{C} = \mathbf{A}} \otimes \underset{K \times L}{\mathbf{B}} \doteq \underbrace{\begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \dots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \dots & a_{n,m}\mathbf{B} \end{pmatrix}}_{N{*}K \times M{*}L}$$

- 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 product 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

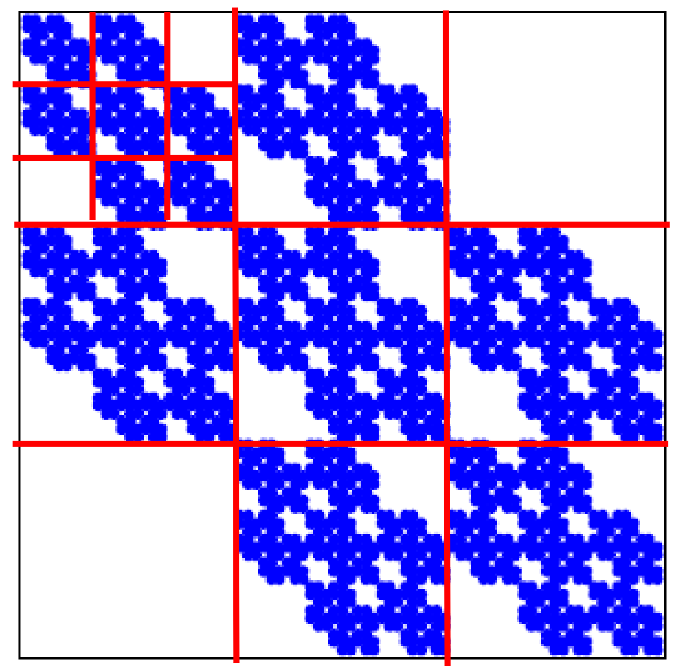
$$G_k = \underbrace{G_1 \otimes G_1 \otimes \ldots G_1}_{k\ times}$$

Kronecker graphs

- Continuing multiplying with G_1 we obtain G_4 and so on …

1	1	0
1	1	1
0	1	1

G_1

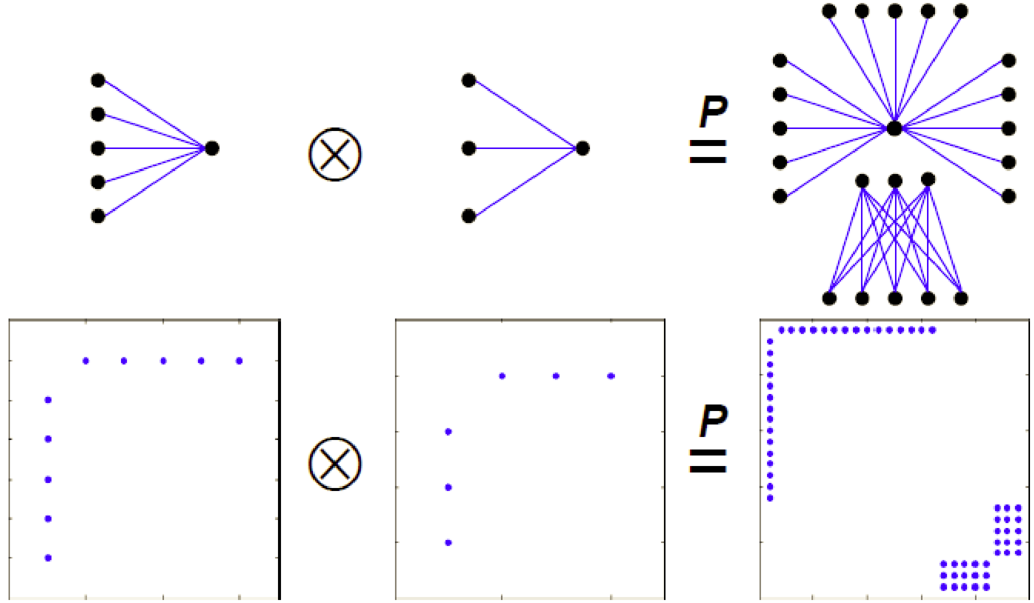


G_3 adjacency matrix

Kronecker graphs – examples on bipartite graph

$\overset{P}{=}$

Equal with the right permutation



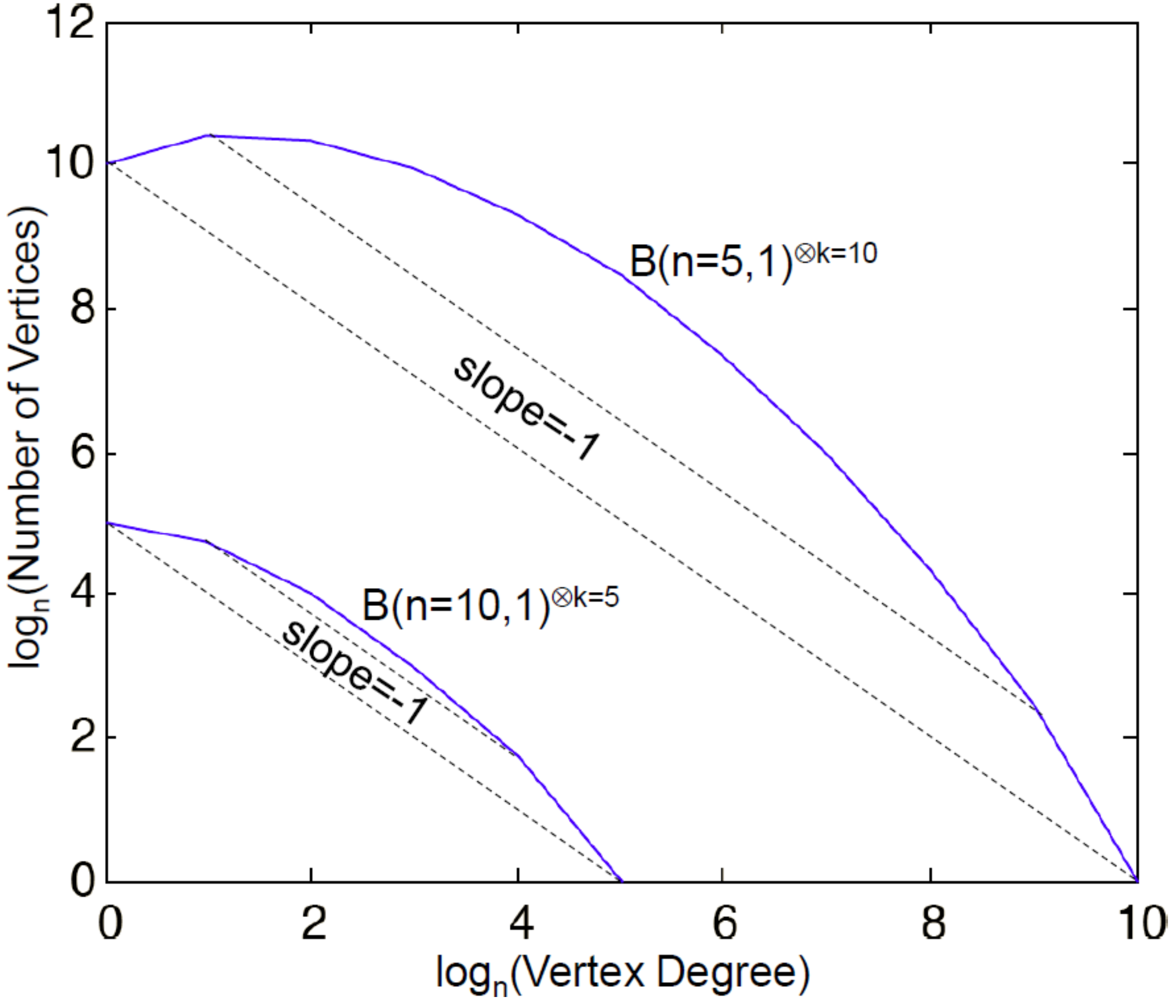
$$B(5,1) \otimes B(3,1) \overset{P}{=} B(15,1) \cup B(3,5)$$

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

$$B(n_1, m_1) \otimes B(n_2, m_2) \overset{P}{=} B(n_1 n_2, m_1 m_2) \cup B(n_2 m_1, n_1 m_2)$$

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

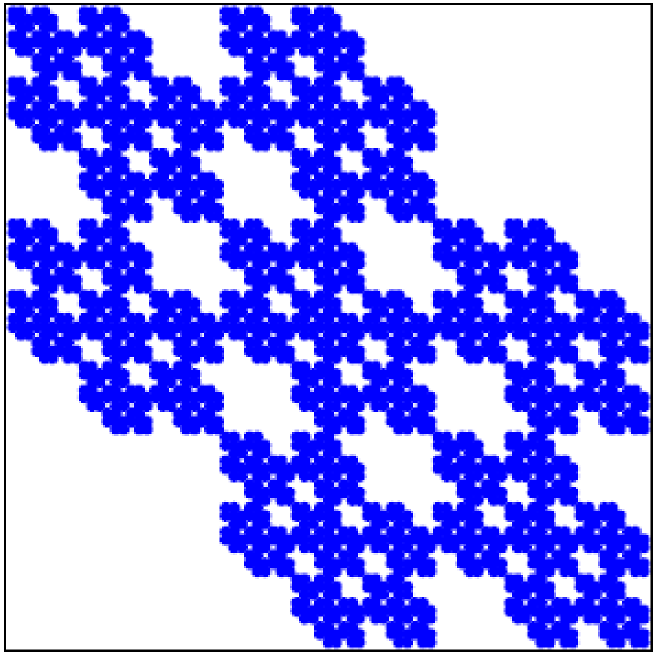
p_{ij}	1	0
1	1	1
0	1	1

G_1 (3x3)

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

$G_2 = G_1 \otimes G_1$

(9x9)



(27x27)

0.5	0.2
0.1	0.3

Θ_1

probability matrix

Kronecker multiplication

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

$\Theta_2 = \Theta_1 \otimes \Theta_1$

Probability of edge p_{uv}

Instance matrix K_2

For each p_{uv} flip Bernoulli coin

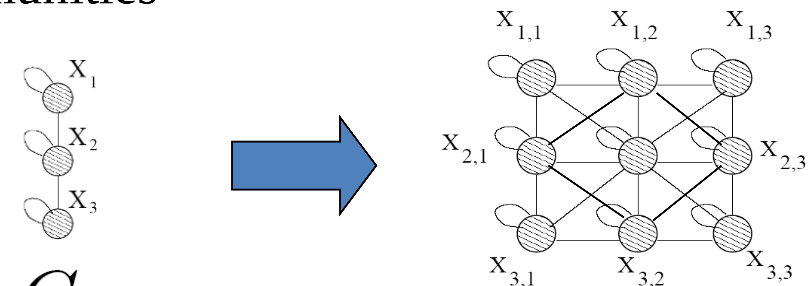
Outlines

- 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



$$G_1$$

$$G_2 = G_1 \otimes G_1$$

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 * 0.1 = 0.05

$$\begin{array}{cc} & \begin{matrix} 1 & 0 \end{matrix} \\ \begin{matrix} 1 \\ 0 \end{matrix} & \begin{bmatrix} 0.5 & 0.2 \\ 0.1 & 0.3 \end{bmatrix} \end{array}$$

$$\Theta_1$$

Properties of Kronecker graphs

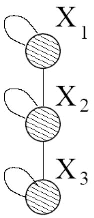
- We prove that Kronecker multiplication generates graphs that obey [PKDD'05]
  - Properties of static networks
    - ✓Power Law Degree Distribution
    - ✓Power Law eigenvalue and eigenvector distribution
    - ✓Small Diameter
  - Properties of dynamic networks
    - ✓Densification Power Law
    - ✓Shrinking/Stabilizing 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 Θ (e.g., $\begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$) which



$$\arg\max_{\Theta} P(G \mid \Theta)$$

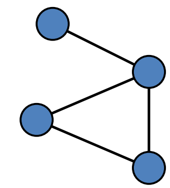
- We need to (efficiently) calculate

$$P(G \mid \Theta)$$

- And maximize over Θ (e.g., using gradient descent)

Fitting Kronecker graphs

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



0.5	0.2
0.1	0.3

Θ

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

Θ_k

1	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

G

$\boxed{P(G|\Theta)}$

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

Challenge 1: Node correspondence

Θ

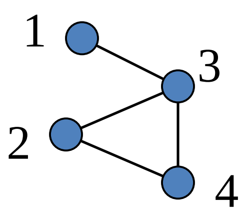
0.5	0.2
0.1	0.3

Θ_k

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

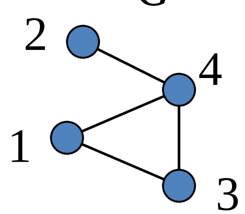
σ

G'



1	0	1	0
0	1	1	1
1	1	1	1
0	0	1	1

G''



1	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

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

- 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 σ

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

- All correspondences are a priori equally likely
- There are $O(N!)$ correspondences

Challenge 2: calculating *P(G|Θ,σ)*

- Assume we solved the correspondence problem
- Calculating

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

σ… node labeling

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

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

Θ_{kc}

σ

1	0	1	1
0	1	0	1
1	0	1	1
0	0	1	1

G

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

Model estimation: solution

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

- We can estimate the parameters of Kronecker graph in linear time $O(E)$

Solution 1: Node correspondence

- Log-likelihood

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

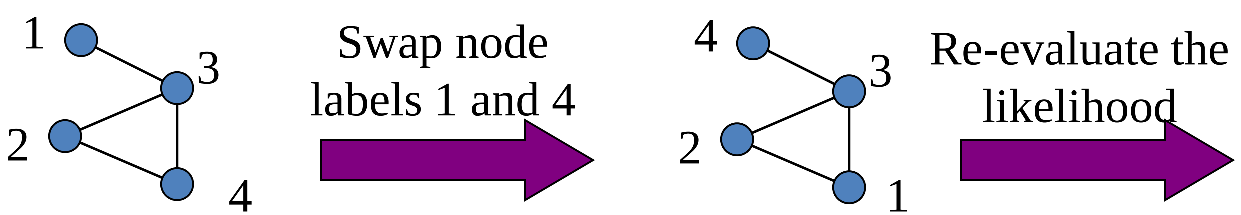
- Gradient of log-likelihood

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

- 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) $P(\sigma | G, \Theta)$
    - If new is worse accept with probability proportional to the ratio of likelihoods



1	1	0	1	0
2	0	1	1	1
3	1	1	1	1
4	0	1	1	1

4	1	1	1	0
2	1	1	1	0
3	1	1	1	1
1	0	0	1	1

Can compute efficiently:
Only need to account for changes in 2 rows / columns

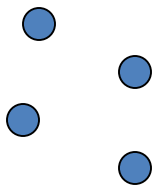
Solution 2: Calculating $P(G|\Theta,\sigma)$

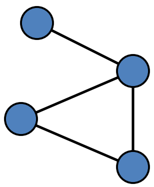
- Calculating naively $P(G|\Theta,\sigma)$ takes $O(N^2)$
- Idea:
  - First calculate likelihood of empty graph, a graph with 0 edges
  - Correct the likelihood for edges that we observe in the graph

- By exploiting the structure of Kronecker product we obtain closed form for likelihood of an empty graph

Solution 2: Calculating *P(G|Θ,σ)*

- We approximate the likelihood:





$$l(\Theta) \approx \underbrace{l_e(\Theta)}_{\text{Empty graph}} + \sum_{(u,v)\in G} \underbrace{-\log(1-\Theta_k[\sigma_u,\sigma_v])}_{\text{No-edge likelihood}} + \underbrace{\log(\Theta_k[\sigma_u,\sigma_v])}_{\text{Edge likelihood}}$$

- The sum goes only over the edges
- Evaluating *P(G|Θ,σ)* takes *O(E)* time
- Real graphs are sparse, $E << N^2$

Model estimation: overall solution

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

initialize $\hat{\Theta}_1$
while *not converged* **do**
  $\boxed{\text{evaluate gradient: } \frac{\partial}{\partial \hat{\Theta}_t} l(\Theta_t)}$
  update parameter estimates: $\hat{\Theta}_{t+1} = \hat{\Theta}_t + \lambda \frac{\partial}{\partial \hat{\Theta}_t} l(\hat{\Theta}_t)$
end
return $\hat{\Theta} = \hat{\Theta}_t$

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

for $t := 1$ **to** T **do**
  $\boxed{\sigma_t := \texttt{SamplePermutation}\,(G, \Theta)}$
  $l_t = \boxed{\log P(G|\sigma^{(t)}, \Theta)}$
  $grad_t := \frac{\partial}{\partial \Theta} \log P(G|\sigma^{(t)}, \Theta)$
end
return $l(\Theta) = \frac{1}{T} \sum_t l_t$, and $\frac{\partial}{\partial \Theta} l(\Theta) = \frac{1}{T} \sum_t grad_t$

Solution 1: Metropolis sampling

Solution 2: Edge-wise prob computation

Outlines

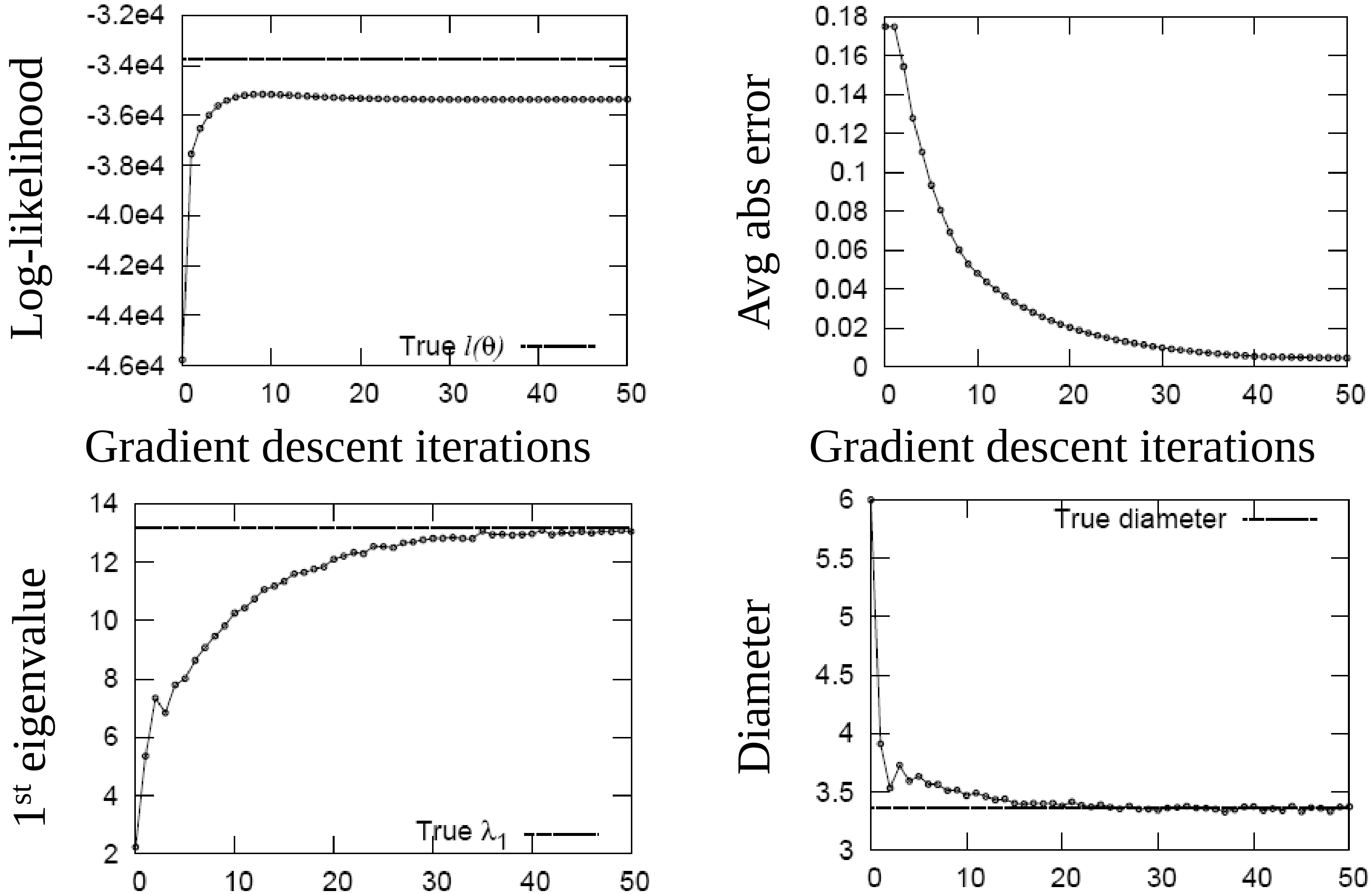
- Introduction
- Network properties – static & temporal
- Proposed graph generation model - Kronecker graph
- Stochastic Kronecker graph
- Properties of Kronecker graph
- Model estimation
- Experimental results
- Discussion

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

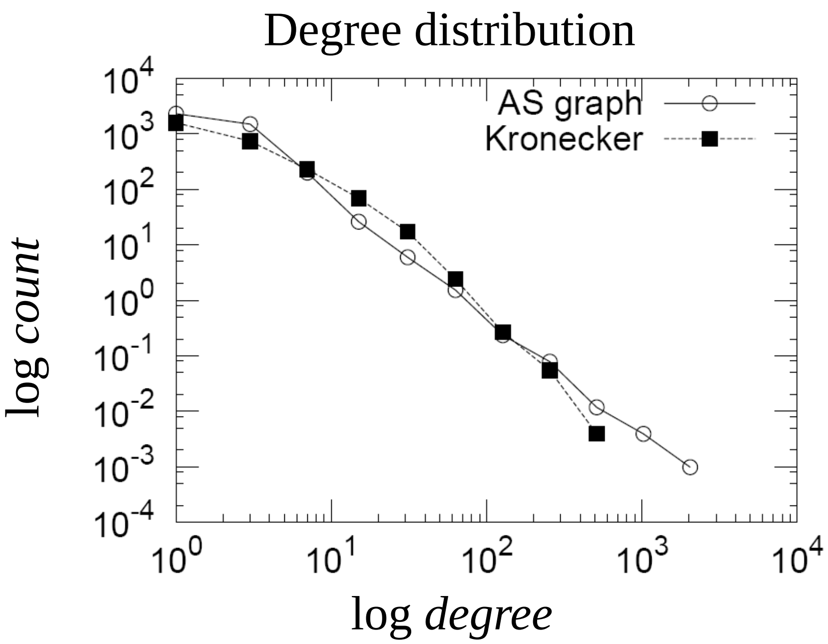
AS graph (N=6500, E=26500)

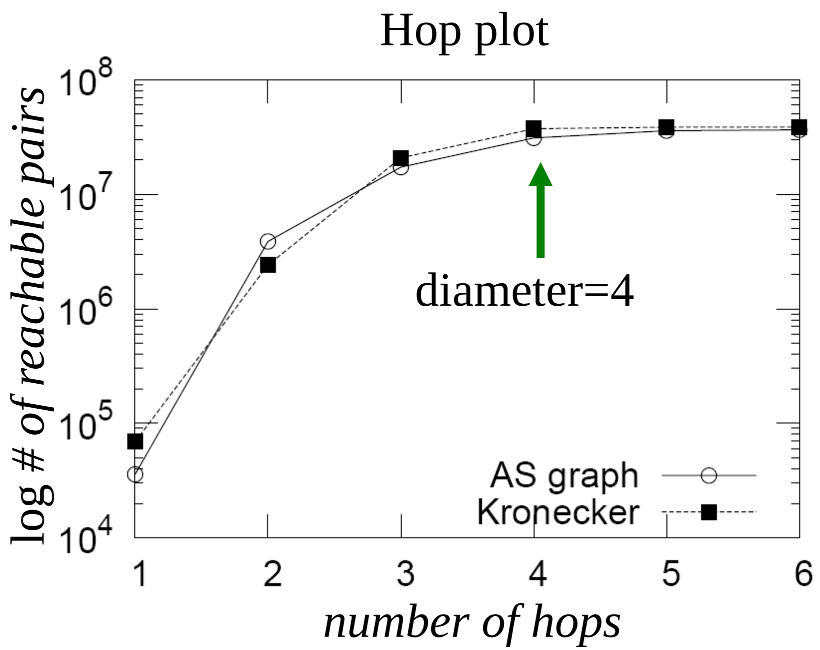
- Autonomous systems (internet)
- We search the space of $\sim 10^{50,000}$ permutations
- Fitting takes 20 minutes
- AS graph is undirected and estimated parameter matrix is symmetric:

0.98	0.58
0.58	0.06

AS: comparing graph properties

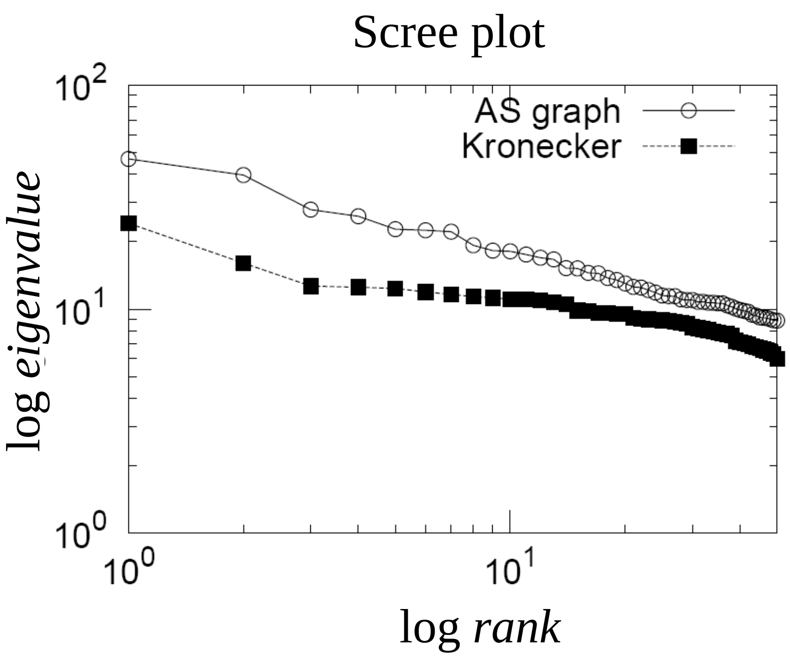
- Generate synthetic graph using estimated parameters
- Compare the properties of two graphs

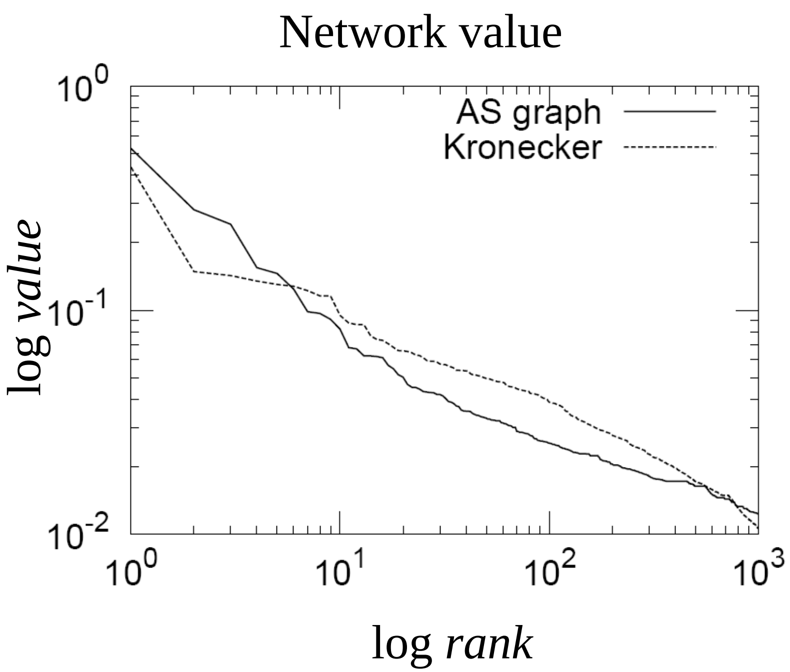




AS: comparing graph properties

- Spectral properties of graph adjacency matrices

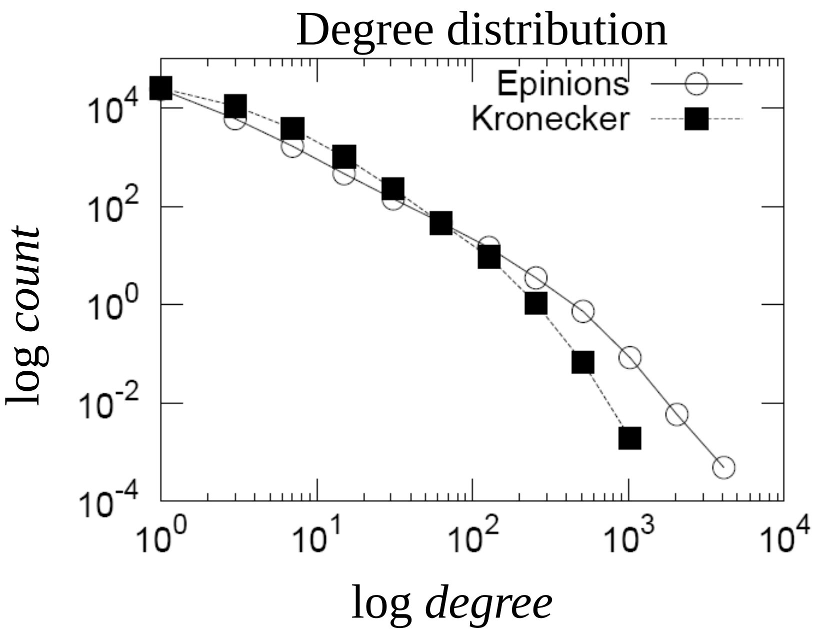


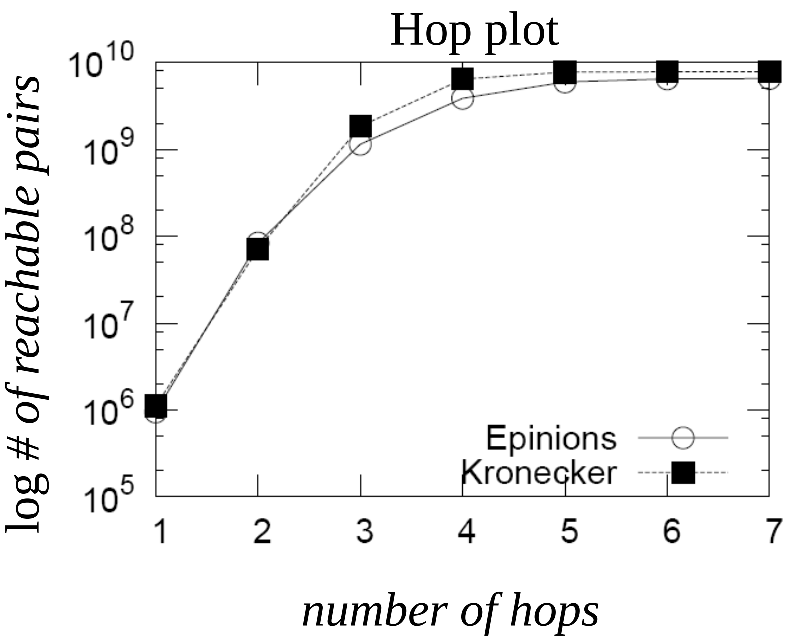


Epinions graph (N=76k, E=510k)

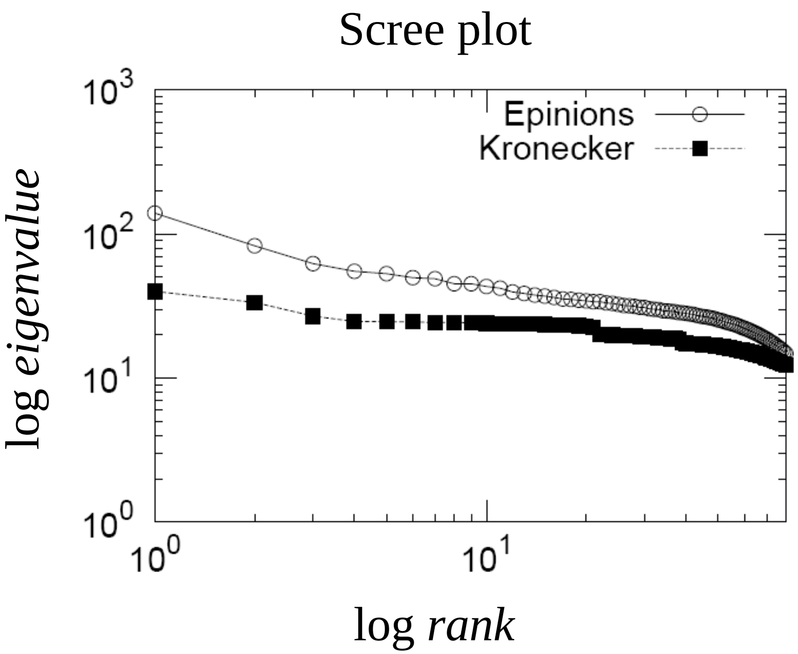
- We search the space of ~$10^{1,000,000}$ permutations
- Fitting takes 2 hours
- The structure of the estimated parameter gives insight into the structure of the graph

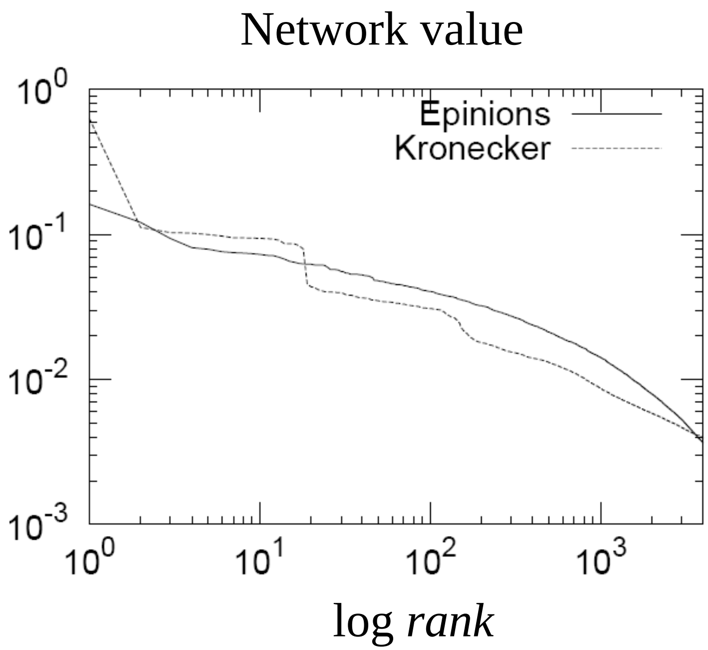
0.99	0.54
0.49	0.13





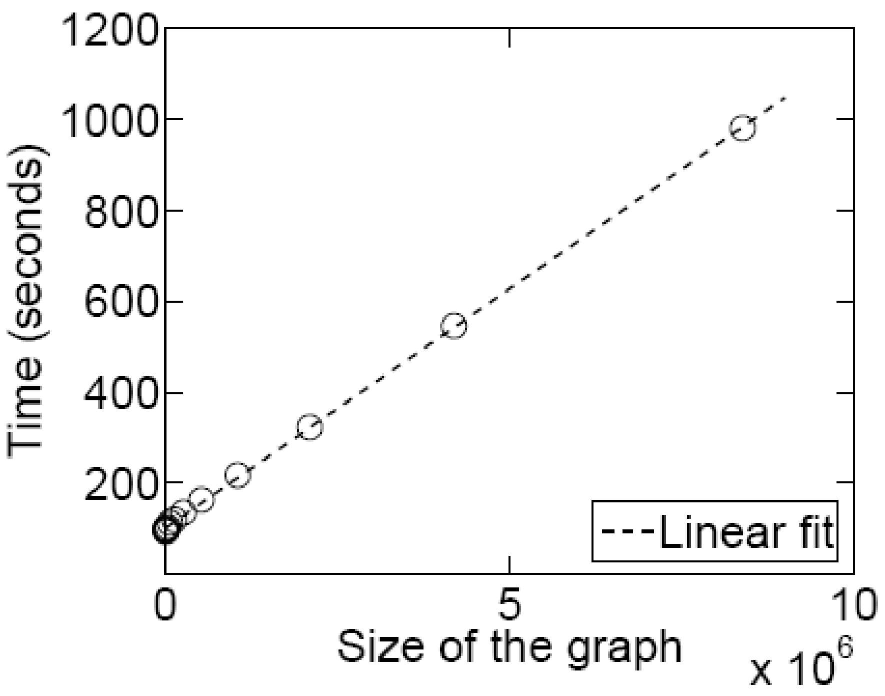
Epinions graph (N=76k, E=510k)





Scalability

- Fitting scales linearly with the number of edges



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

- Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., & Ghahramani, Z. (2010). Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, *11*(Feb), 985-1042.

- Leskovec, J. (2008). *Dynamics of large networks*. Carnegie Mellon University.

- Leskovec's slide: http://slideplayer.com/slide/4809425/

- Jeremy Kepner's slides for MIT OCW: https://ocw.mit.edu/resources/res-ll-005-d4m-signal-processing-on-databases-fall-2012/lecture-notes-and-class-videos/MITRES_LL-005F12_Lec8.pdf

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