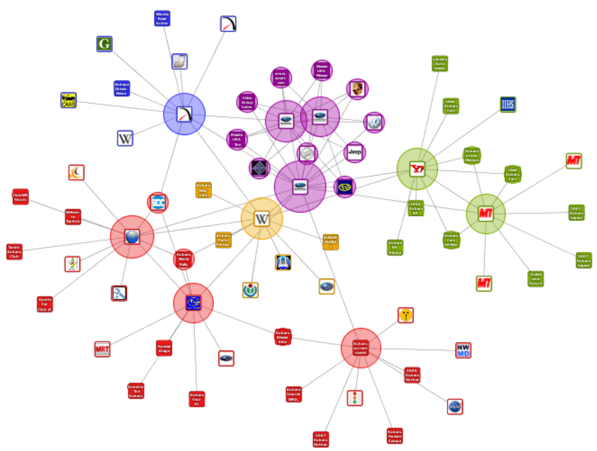
Jure Leskovec
Carnegie Mellon University

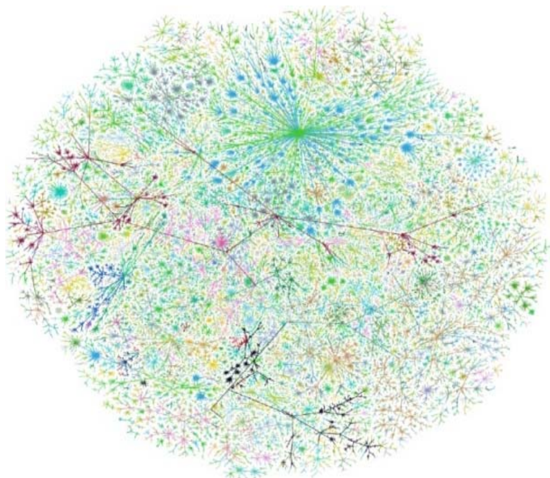
Kronecker Graphs

Joint work with Christos Faloutsos, Jon Kleinberg and Deepay Chakrabarti

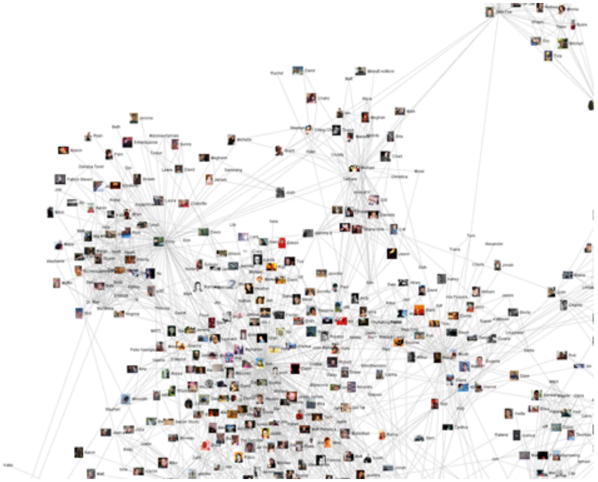
The networks



a) World wide web



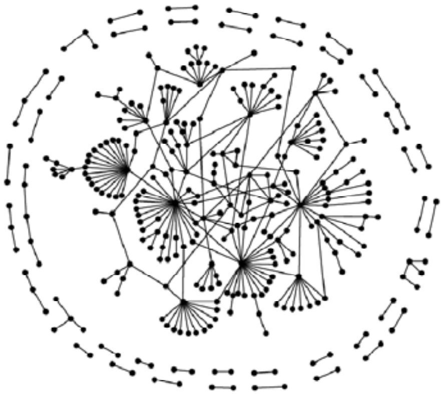
b) Internet (AS)



c) Social networks

d) Communication

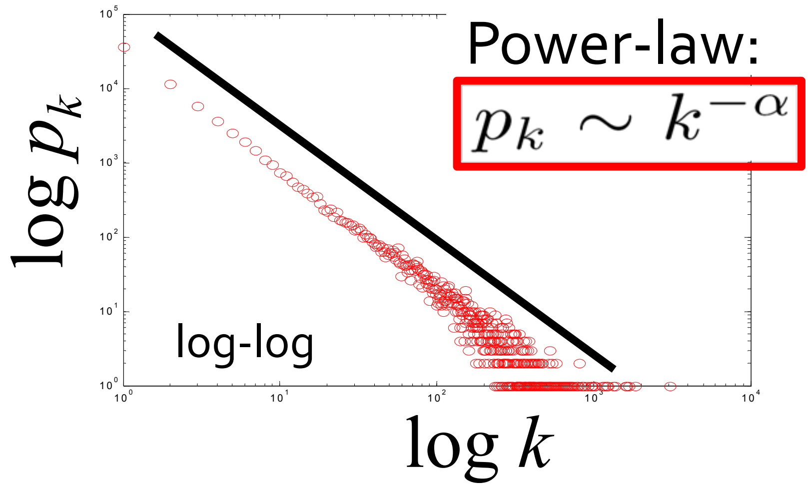
e) Citations

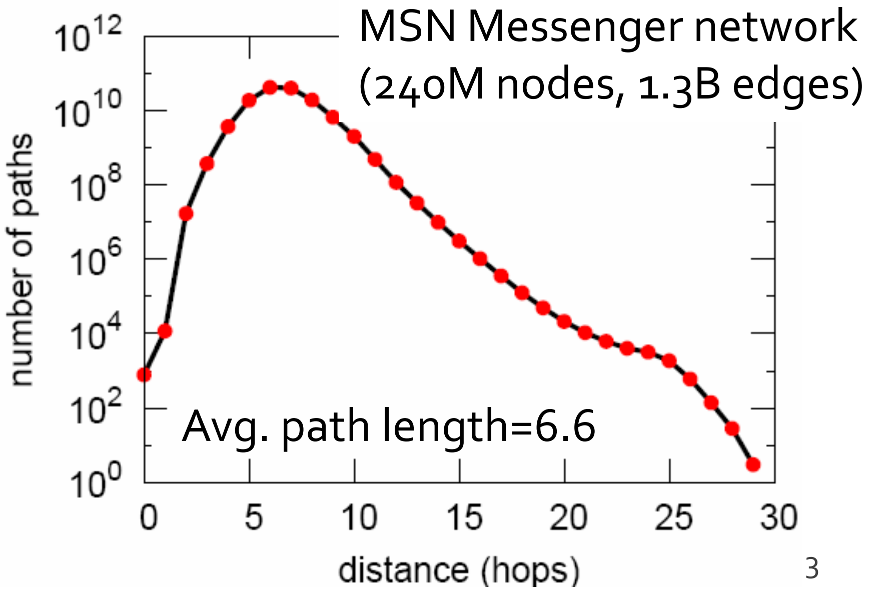


f) Protein interactions

Network properties

- Large networks share many structural properties
  - Scale-free (power-law degree distributions)
  - 6-degrees of separation
  - Transitivity
- And we have models to think about them:
  - Preferential attachment
  - Small world
  - Copying model





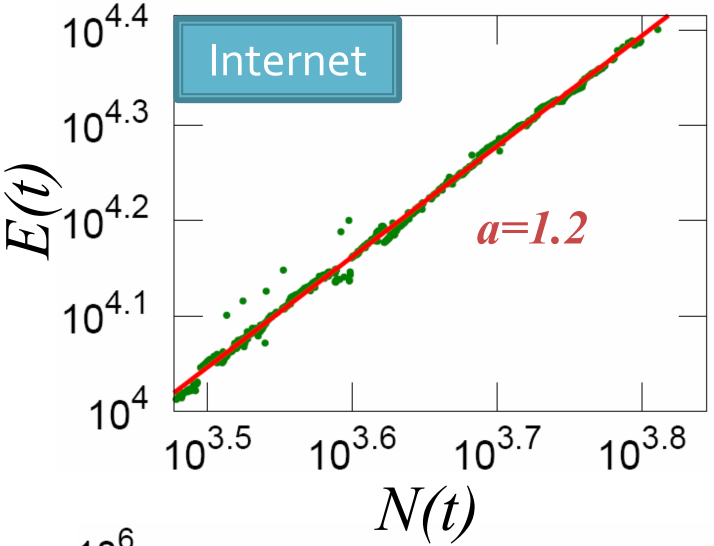
Network evolution

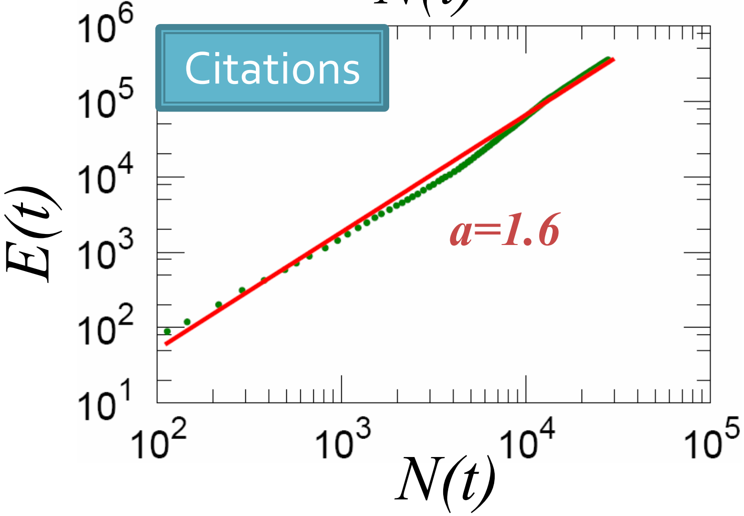
[KDD ’05]

- What is the relation between the number of nodes and the edges over time?
- ~~Prior work assumes: constant average degree over time~~
- Networks are denser over time
- **Densification Power Law:**

$$E(t) \propto N(t)^a$$

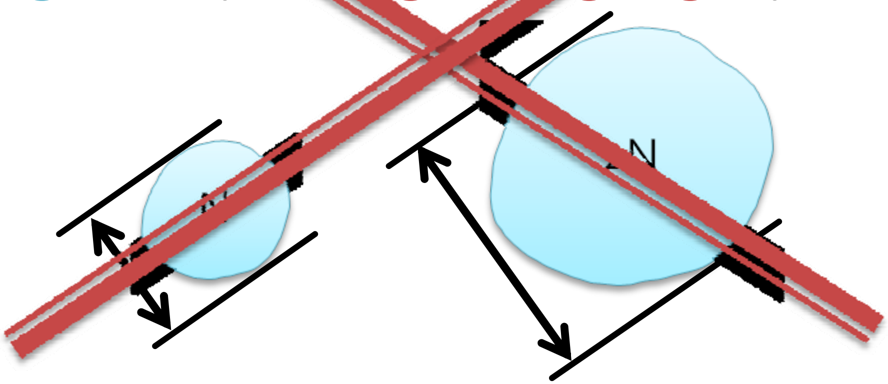
a … densification exponent (1 ≤ a ≤ 2)

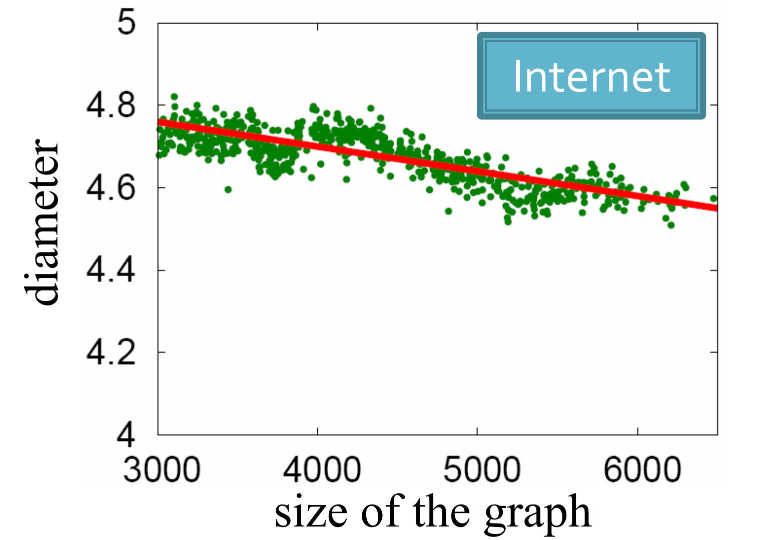


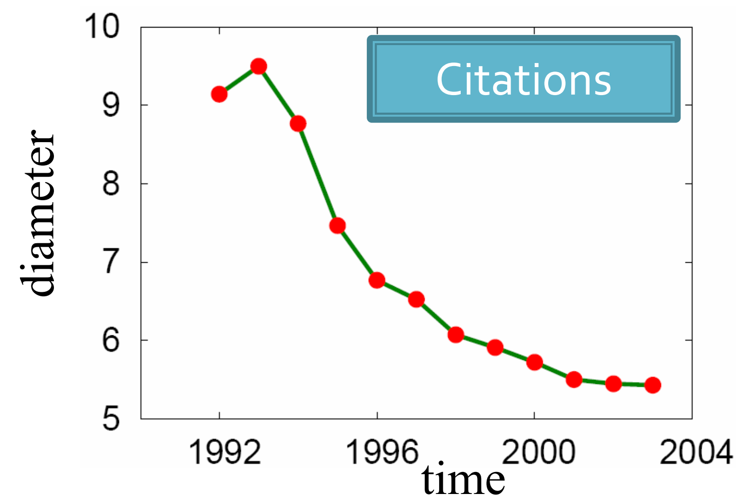


Network evolution

- ~~Prior models and intuition say that the network diameter slowly grows (like log N, log log N)~~



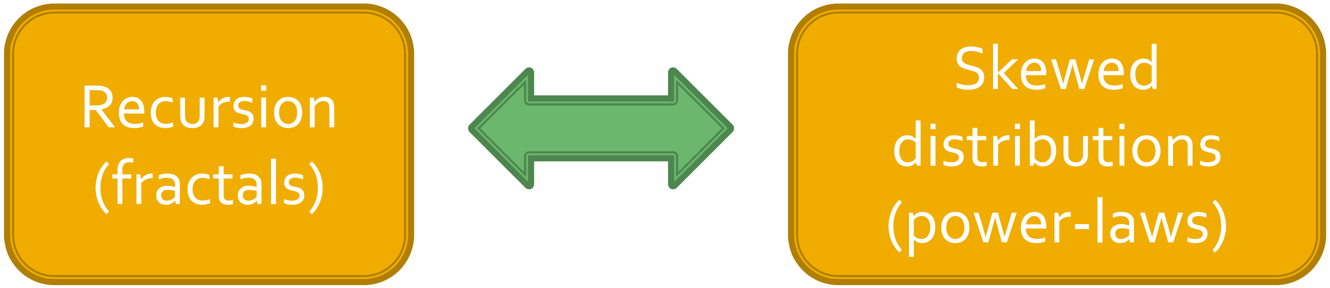




- **Diameter shrinks over time**
  - as the network grows the distances between the nodes slowly **decrease**

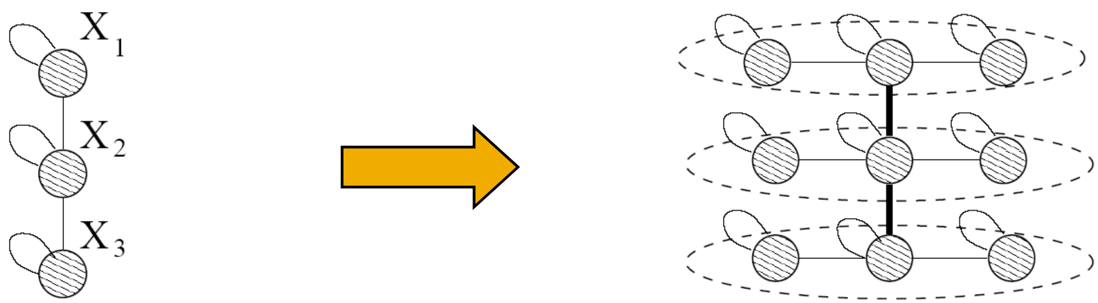
Need a new network model

- None of the existing models generates graphs with these properties
- We need a new model
- **Idea:** Generate graphs recursively



Idea: Recursive graph generation

- There are many obvious (but wrong) ways:

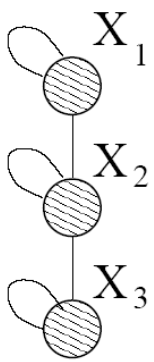


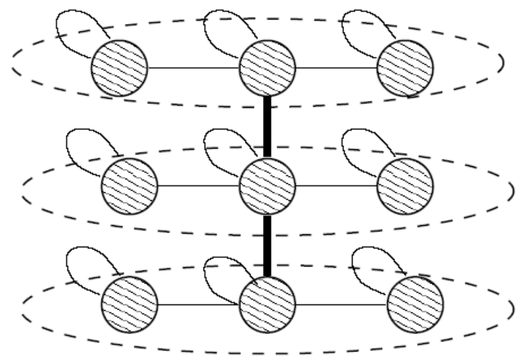
Initial graph Recursive expansion

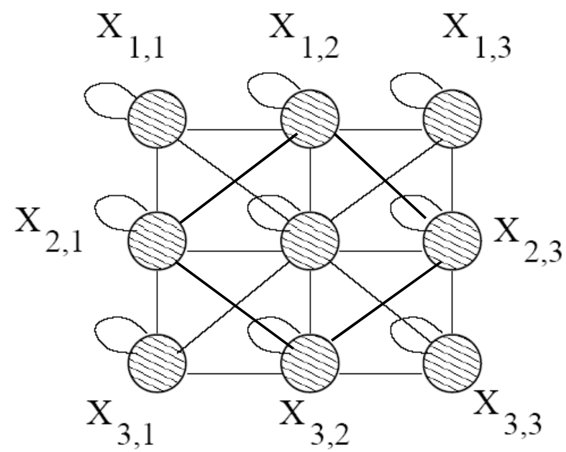
  - Does not densify, has increasing diameter
- **Kronecker Product** is a way of generating self-similar matrices

[PKDD ’05]

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

- Kronecker product of matrices A and B is given by

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

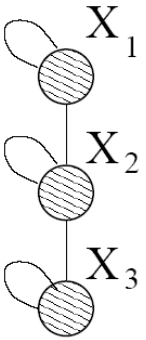
- We define [PKDD '05] a Kronecker product of two graphs as a Kronecker product of their adjacency matrices

[PKDD ’05]

Kronecker graphs

- We propose a growing sequence of graphs by iterating the Kronecker product

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



G_1

- Each Kronecker multiplication exponentially increases the size of the graph
- G_k has N_1^k nodes and E_1^k edges, so we get densification

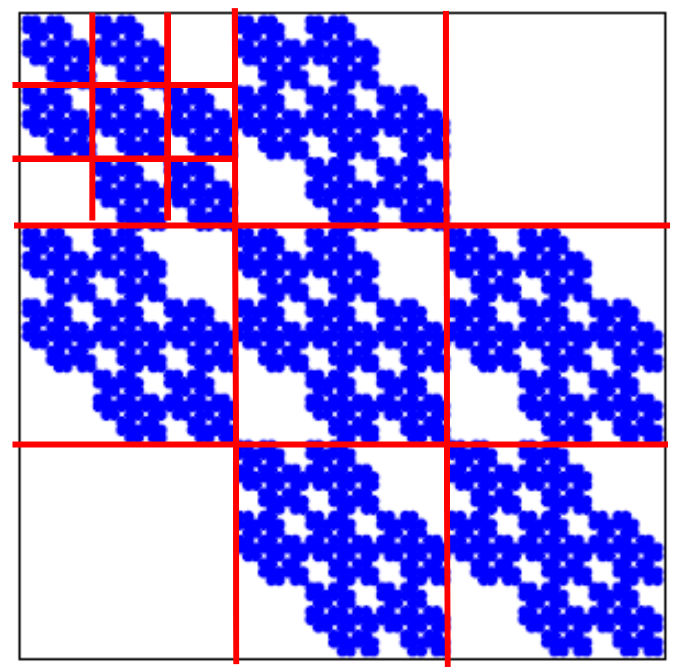
Kronecker product: Graph

[PKDD ’05]

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

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$$

G_1



G_4 adjacency matrix

[PKDD '05]

Stochastic Kronecker graphs

- Create $N_1 \times N_1$ probability matrix P_1
- Compute the k^{th} Kronecker power P_k
- For each entry p_{uv} of P_k include an edge (u,v) with probability p_{uv}

0.5	0.2
0.1	0.3

P_1

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

$P_2 = P_1 \otimes P_1$

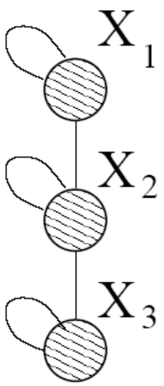
Probability of edge p_{ij}

Instance matrix K_2

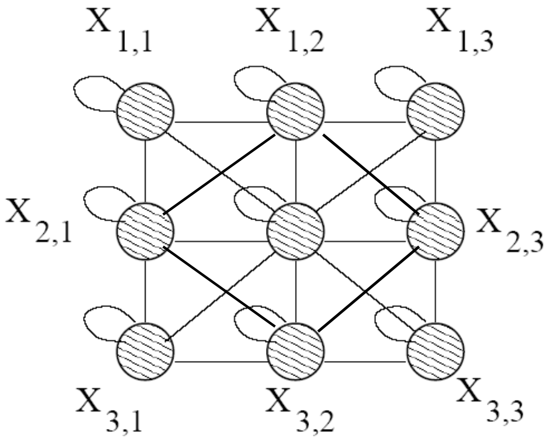
flip biased coins

Kronecker graphs: Intuition (1)

- **Intuition**
  - 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



$G_2 = G_1 \otimes G_1$

Kronecker graphs: Intuition (2)

- Node attribute representation
  - Nodes are described by attributes
    - u=[1,0], v=[1, 1]
  - Parameter matrix gives linking probability:

    p(u,v) = 0.1 * 0.5 = 0.15

	1	0
1	0.5	0.2
0	0.1	0.3

Kronecker multiplication

v

u	11	10	01	00
11	0.25	0.10	0.10	0.04
10	0.05	0.15	0.02	0.06
01	0.05	0.02	0.15	0.06
00	0.01	0.03	0.03	0.09

Kronecker graphs: Intuition (2)

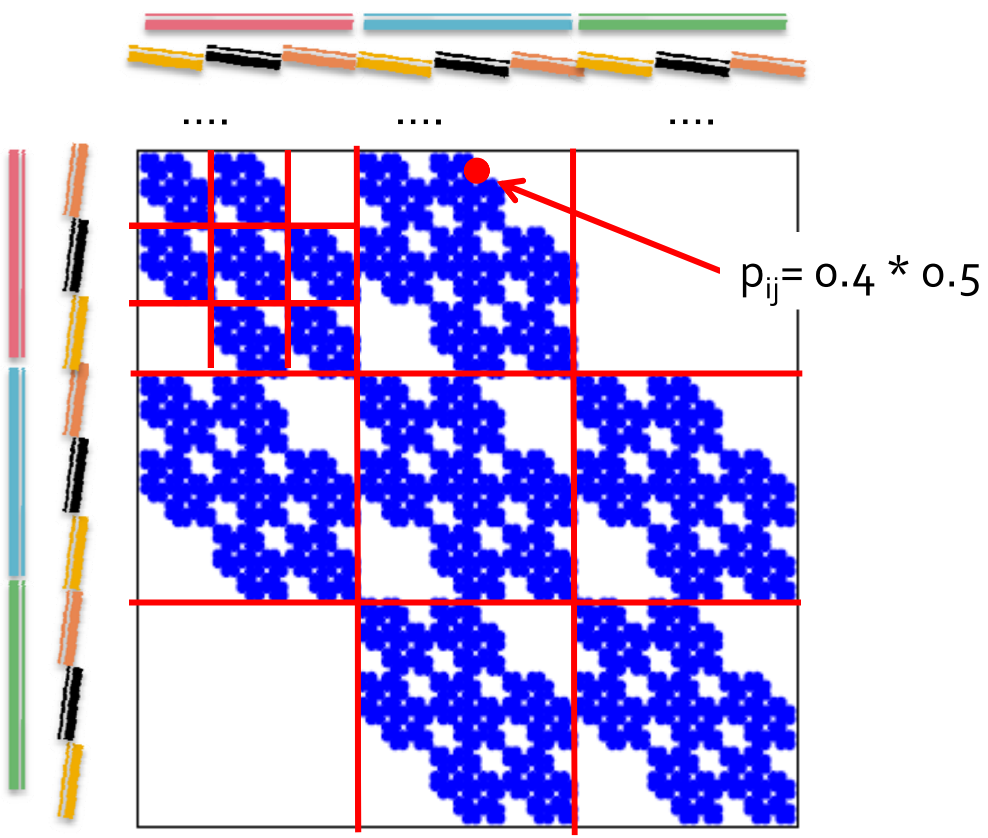
- Using multiple initiators:

Attribute 1:

0.9	0.4	0.2
0.3	0.8	0.2
0.3	0.1	0.3

Attribute 2:

0.8	0.5	0.1
0.2	0.9	0.4
0.4	0.2	0.3



Properties of Kronecker graphs

- We prove [PKDD05] that Kronecker graphs have the following structural properties:
  - 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

[PKDD '05]

Degree distribution

- Theorem: Kronecker Graphs have multinomial in- and out-degree distribution

  (which can be made to behave like a Power Law)

- Proof:
  - Let G_1 have degrees $d_1, d_2, \ldots, d_N$
  - Kronecker multiplication with a node of degree d gives degrees $d \cdot d_1, d \cdot d_2, \ldots, d \cdot d_N$
  - After Kronecker powering G_k has multinomial degree distribution

[PKDD '05]

Eigen-value/-vector Distribution

- Theorem: The Kronecker Graph has multinomial distribution of its eigenvalues
- Theorem: The components of each eigenvector in Kronecker Graph follow a multinomial distribution
- Proof: Trivial by properties of Kronecker multiplication

Temporal Patterns: Densification

- Theorem: Kronecker graphs follow a Densification Power Law with densification exponent

$$a = \log(E_1)/\log(N_1)$$

- Proof:
  - If G_1 has N_1 nodes and E_1 edges then G_k has $N_k = N_1^k$ nodes and $E_k = E_1^k$ edges
  - And then $E_k = N_k^a$
  - Which is a Densification Power Law

Constant Diameter – Proof Sketch

- Theorem: Constant diameter: If G_1 has diameter d then graph G_k also has diameter d

- Observation: Edges in Kronecker graphs:

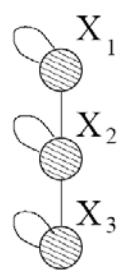
$$Edge\ (X_{ij}, X_{kl}) \in G \otimes H$$

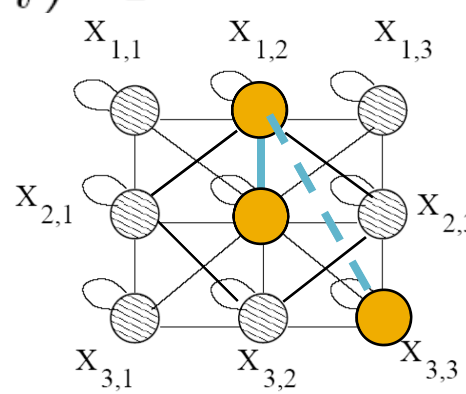
$$iff\ (X_i, X_k) \in G\ \ and\ (X_j, X_l) \in H$$

where X are appropriate nodes

- Example:

G_1





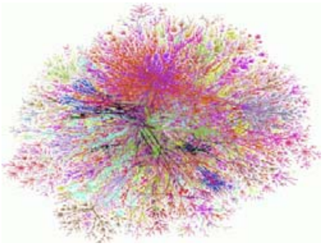
$G_2 = G_1 \otimes G_1$

Why is this important?

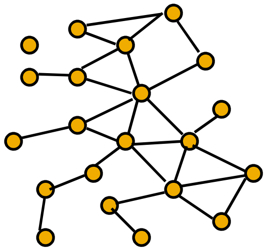
- Why models and realistic synthetic graphs:
  - Anomaly detection – abnormal behavior, evolution
  - Predictions – predicting future from the past
  - Simulations of new algorithms where real graphs are hard/impossible to collect
  - Graph sampling – many real world graphs are too large to deal with
  - "What if" scenarios

Generating realistic graphs

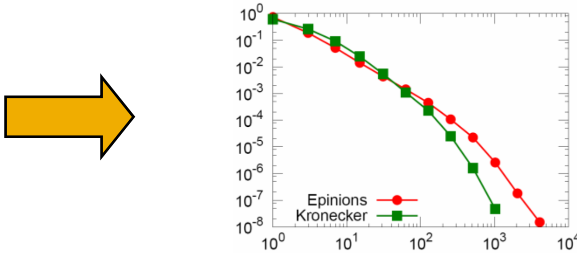
- Want to generate realistic networks:



Given a
real network



Generate a
synthetic network



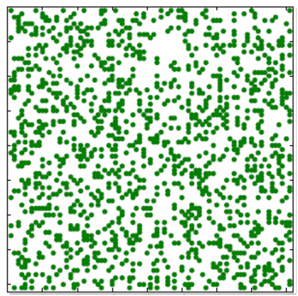
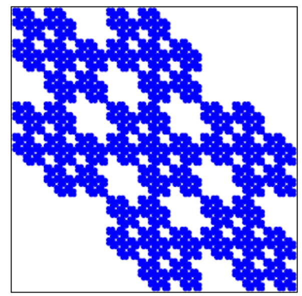
Compare graphs properties,
e.g., degree distribution

- Good news: Kronecker graphs have the expressive power
- But: How do we choose the parameters to match all of these at once?

- **Q:** Which network properties do we care about?
- **A:** Don’t commit, let’s match adjacency matrices

Kronecker graphs: Estimation

- Maximum likelihood estimation:

$$\arg\max_{\Theta} P\Big($$  $$\Big|$$  $$\xleftarrow{\text{Kronecker}} \Theta \Big)$$

- Naïve estimation takes *$O(N!N^2)$*:
  - *$N!$* for different node labelings:
  - *N^2* for traversing graph adjacency matrix
- Do stochastic gradient descent

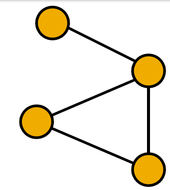
$$\Theta = \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array}$$

We (KronFit) estimate the model in **O(E)**

[ICML ’07]

KronFit: Likelihood

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



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

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

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

Challenge 1: Node correspondence

Θ

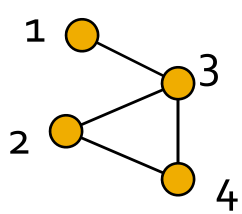
0.5	0.2
0.1	0.3

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

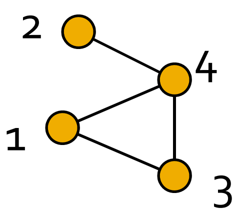
σ

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'|Θ) = P(G''|Θ)*
- 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 $P(G|\Theta,\sigma)$

- Assume we solved the correspondence problem
- Calculating

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

σ… node labeling

- Takes ***O(N²)*** 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

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

Solution 1: Node correspondence

- Log-likelihood

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

- Gradient of log-likelihood

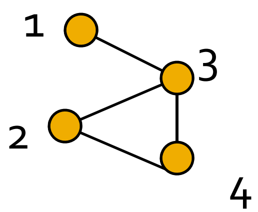
$$\frac{\partial}{\partial \Theta} l(\Theta) = \sum_{\sigma} \frac{\partial \log P(G|\sigma, \Theta)}{\partial \Theta} \boxed{P(\sigma|G, \Theta)}$$

- Sample the permutations from P(σ|G,Θ) and average the gradients

Solution 1: Node correspondence

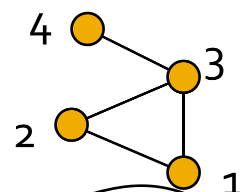
- Metropolis sampling
  - Start with a random permutation σ
  - Do local moves on the permutation
  - Accept the new permutation σ'
    - If new permutation is better (gives higher likelihood)
    - else accept with prob. proportional to the ratio of likelihoods (**no need to calculate the normalizing constant!**)

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



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

Swap node labels 1 and 4



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

Can compute efficiently

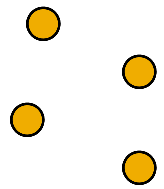
Only need to account for changes in 2 rows / columns

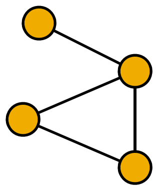
Solution 2: Calculating $P(G|\Theta,\sigma)$

- 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 $P(G|\Theta,\sigma)$

- 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 $P(G|\Theta,\sigma)$ takes $O(E)$ time
- Real graphs are sparse, $E << N^2$

We estimate the model in O(E)

Experiments: real networks

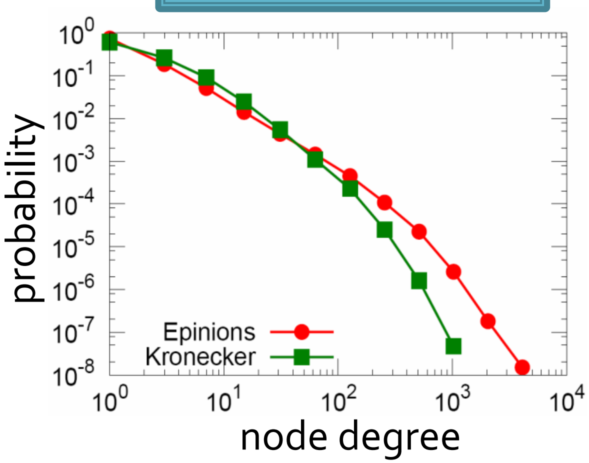
- Experimental setup
  - Given real graph
  - Stochastic gradient descent from random initial point
  - Obtain estimated parameters
  - Generate synthetic graphs
  - Compare properties of both graphs
- We do not fit the properties themselves
- We fit the likelihood and then compare the graph properties

Estimation: Epinions (N=76k, E=510k)

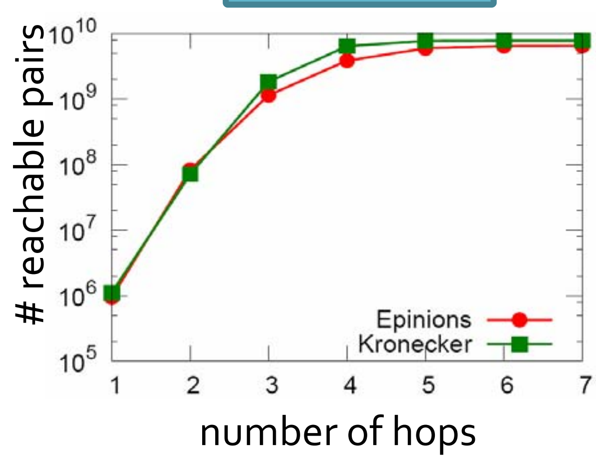
- We search the space of $\sim 10^{1,000,000}$ permutations
- Fitting takes 2 hours
- Real and Kronecker are very close

$$\hat{\Theta} = \begin{array}{|c|c|} \hline 0.99 & 0.54 \\ \hline 0.49 & 0.13 \\ \hline \end{array}$$

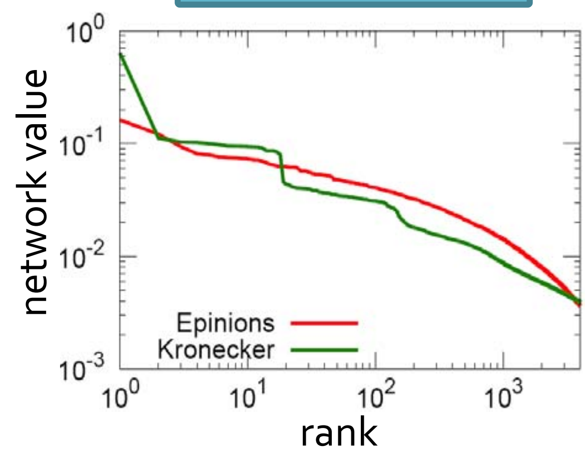
Degree distribution



Path lengths

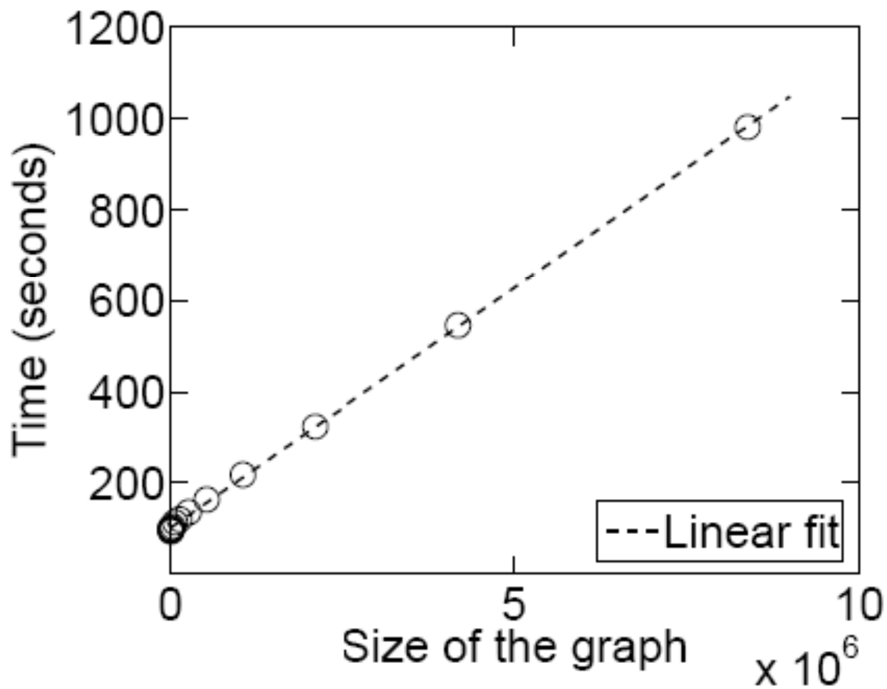


"Network" values



Scalability

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



Kronecker generalizes G_{np} and RMat

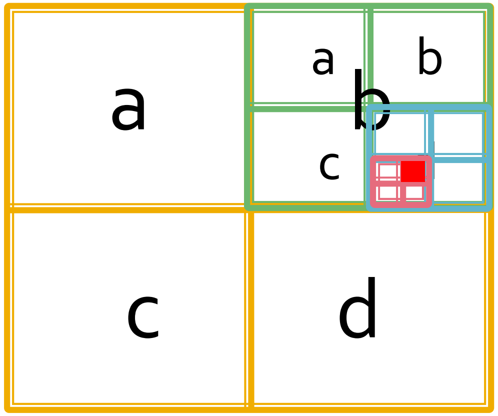
- Kronecker generalizes Erdos-Renyi (G_{np}):
  - Use 1x1 initiator (all cells have the same prob.)
- Kronecker generalizes RMat
  - Use 2x2 initiator
- Relation between Kronecker and RMat
  - like G_{np} (n nodes, p edge prob), G_{nm} (n nodes, m edges):
  - Kronecker also encodes the number of edges
  - (in RMat that is a separate parameter)

Quickly generating Kroneckers

- In practice we generate Kronecker graphs using the RMat like recursive deepening
  - Given prob. Kronecker initiator P
  - Expected number of edges: E = sum(P)^k
  - Normalize P’ = 1/sum(P) * P
  - Perform recursive deepening using P’

$$P' = \begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline \end{array}$$

a+b+c+d=1



Conclusion

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

References

- Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations, by Jure Leskovec, Jon Kleinberg, Christos Faloutsos, ACM KDD 2005
- Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication, by Jure Leskovec, Deepay Chakrabarti, Jon Kleinberg and Christos Faloutsos, PKDD 2005
- Scalable Modeling of Real Graphs using Kronecker Multiplication, by Jure Leskovec and Christos Faloutsos, ICML 2007
- Graph Evolution: Densification and Shrinking Diameters, by Jure Leskovec, Jon Kleinberg and Christos Faloutsos, ACM TKDD 2007