Kronecker graphs and the Structure of Large Networks

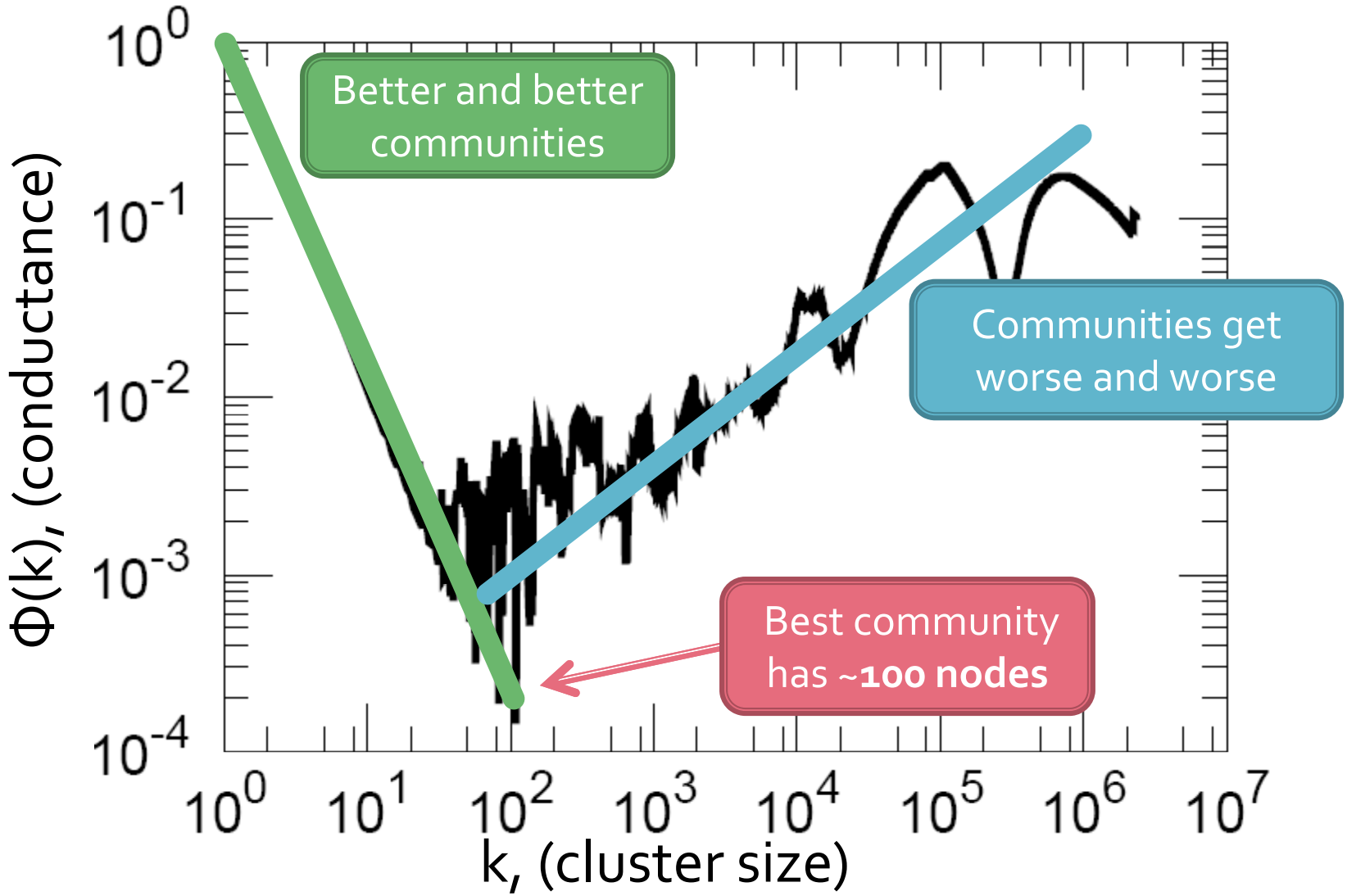
CS224W: Social and Information Network Analysis

Jure Leskovec, Stanford University

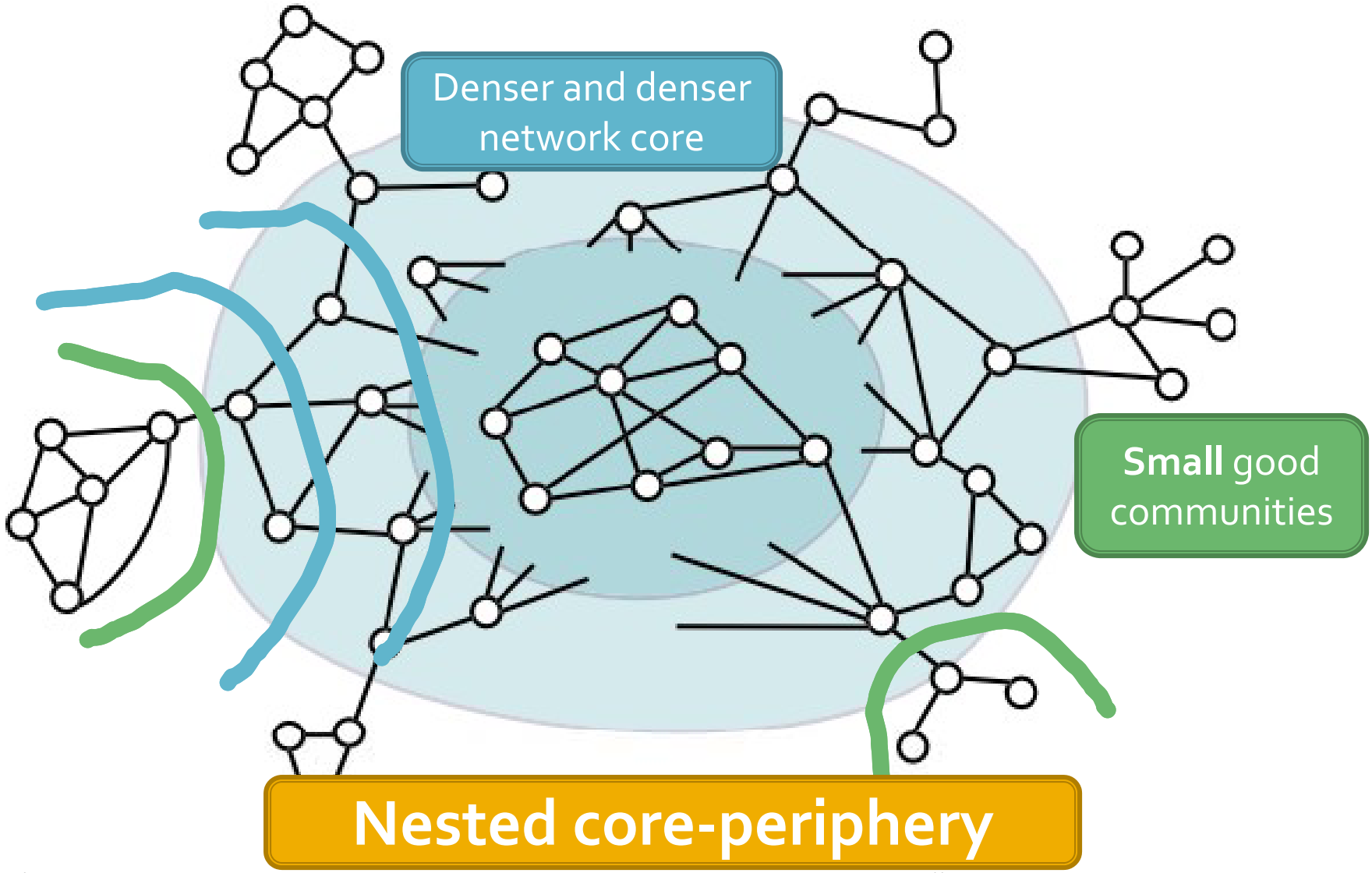
http://cs224w.stanford.edu

[Internet Mathematics '09]

Recap: Network Community Profile

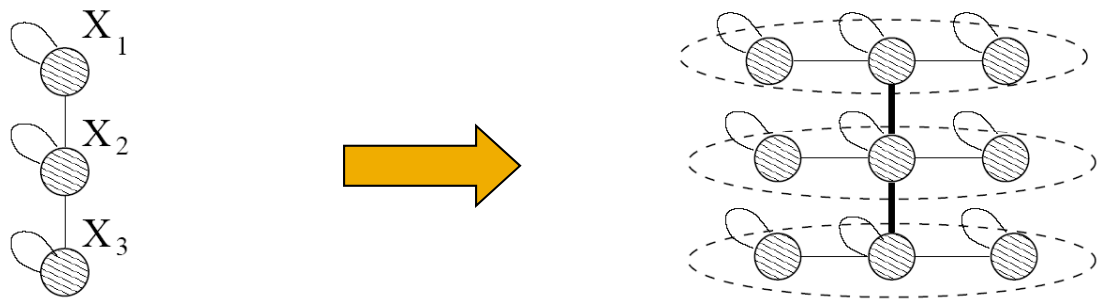


Explanation: Nested core-periphery



Idea: Recursive graph generation

- Intuition: self-similarity leads to power-laws
- Try to mimic recursive graph / community growth
- There are many obvious (but wrong) ways:

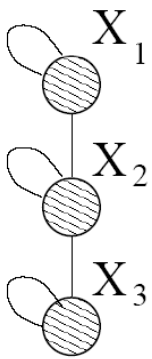


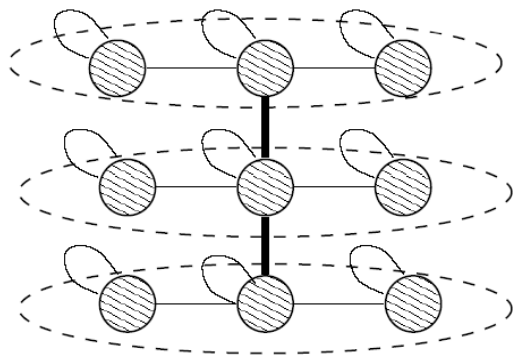
Initial graph Recursive expansion

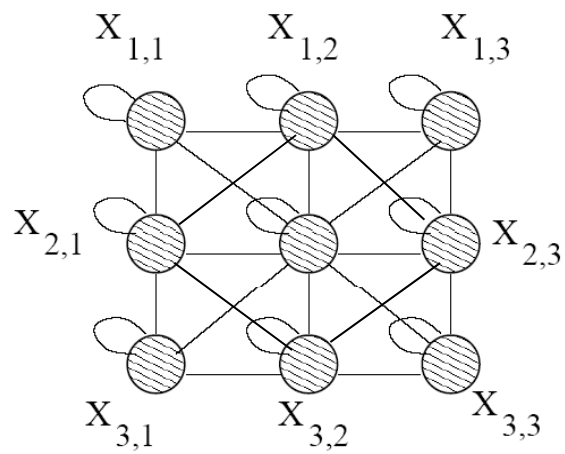
- Kronecker Product is a way of generating self-similar matrices

[PKDD '05]

Kronecker product: Graph







Intermediate stage

1	1	0
1	1	1
0	1	1

(3x3)

K_1

Adjacency matrix

K_1	K_1	0
K_1	K_1	K_1
0	K_1	K_1

(9x9)

$K_2 = K_1 \otimes K_1$

Adjacency matrix

Kronecker product: Definition

- Kronecker product of matrices A and B is given by

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

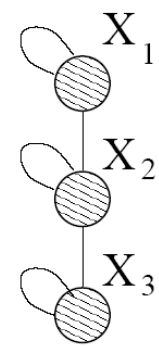
- Define a Kronecker product of two graphs as a Kronecker product of their adjacency matrices

[PKDD '05]

Kronecker graphs

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

$$K_1^{[k]} = K_k = \underbrace{K_1 \otimes K_1 \otimes \ldots K_1}_{k \text{ times}} = K_{k-1} \otimes K_1$$

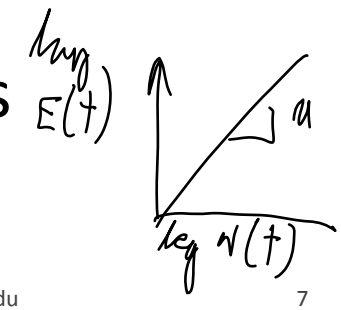


K_1

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

$a = \frac{\log E_0}{\log N_0}$ $\quad a = \frac{\log E(t)}{\log N(t)}$

- One can easily use multiple initiator matrices (K_1', K_1'', K_1''') that can be of different sizes



[PKDD '05]

Kronecker product: Graph

- Continuing multypling with K_1 we obtain K_4 and so on…

1	1	0
1	1	1
0	1	1

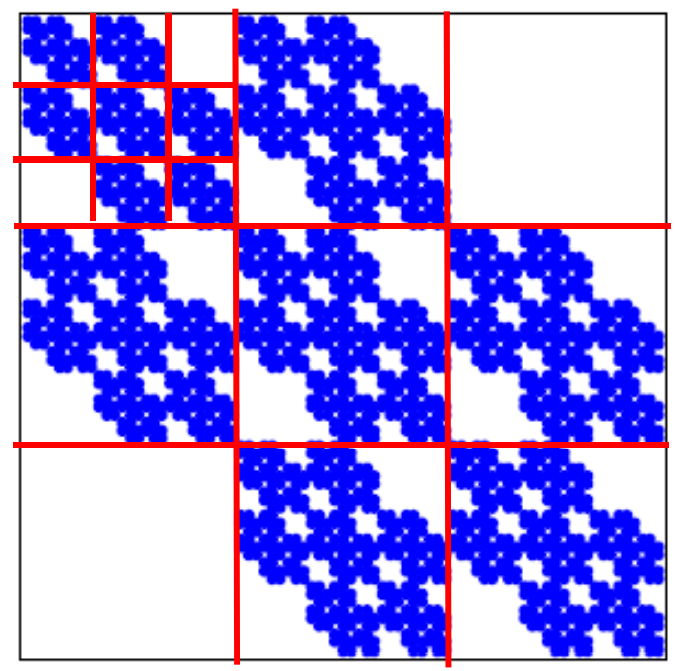
K_1

3x3

K_1	K_1	0
K_1	K_1	K_1
0	K_1	K_1

$K_2 = K_1 \otimes K_1$

9x9

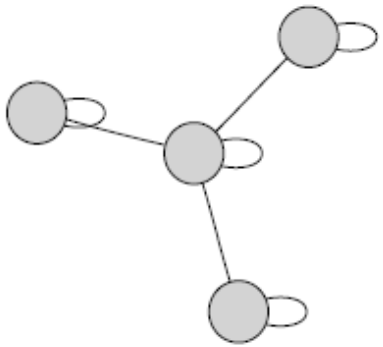


9x9

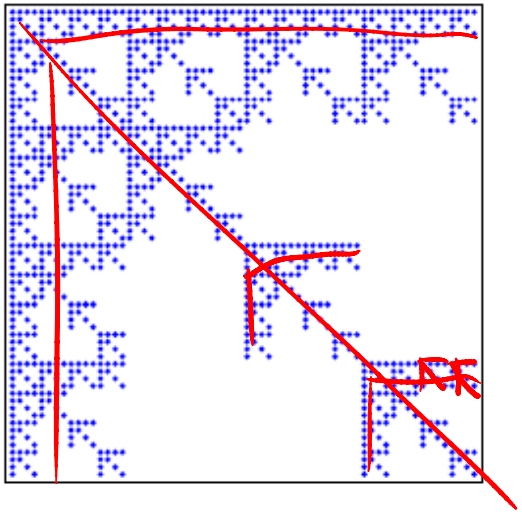
K_4 adjacency matrix

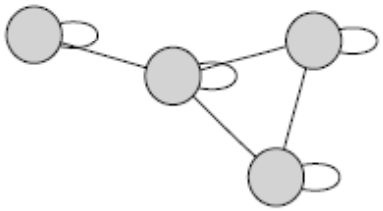
[PKDD '05]

Kronecker initiator matrices

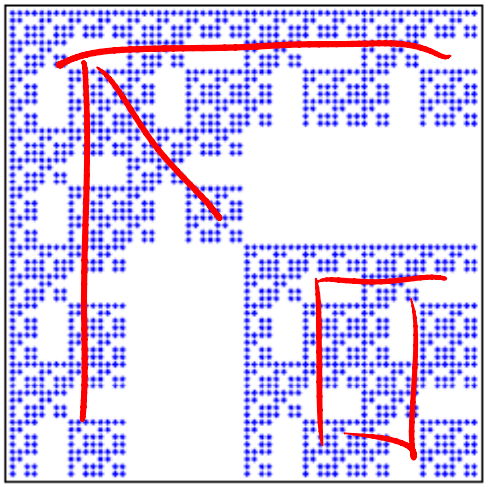


1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1





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



Initiator K_1 | K_1 adjacency matrix | K_3 adjacency matrix

[PKDD '05]

Properties of Kronecker graphs

- Kronecker graphs have many properties found in real networks:
  - Properties of static networks
    - Power-Law like Degree Distribution
    - Power-Law eigenvalue and eigenvector distribution
    - Small Diameter
  - Properties of dynamic networks
    - Densification Power Law
    - Shrinking/Stabilizing Diameter

[PKDD '05]

Constant Diameter – Proof Sketch

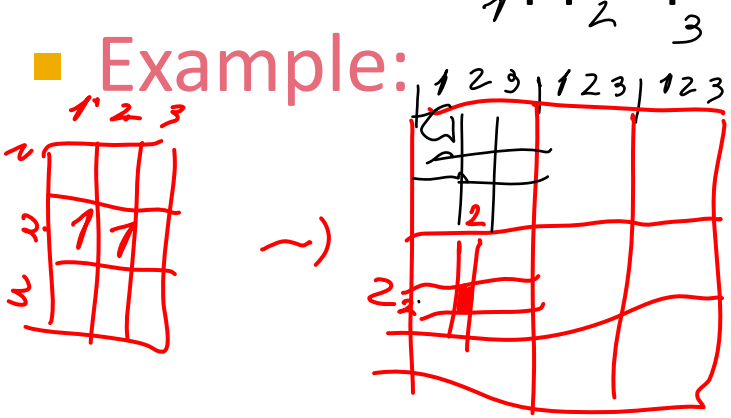
- Theorem: Constant diameter: If G_1 has diameter d then graph G_k also has diameter d
- Observation: Edges in Kronecker graphs:

$$Edge\ (X_{ij}, X_{kl}) \in G \otimes H$$
$$iff\ (X_i, X_k) \in G\ and\ (X_j, X_l) \in H$$

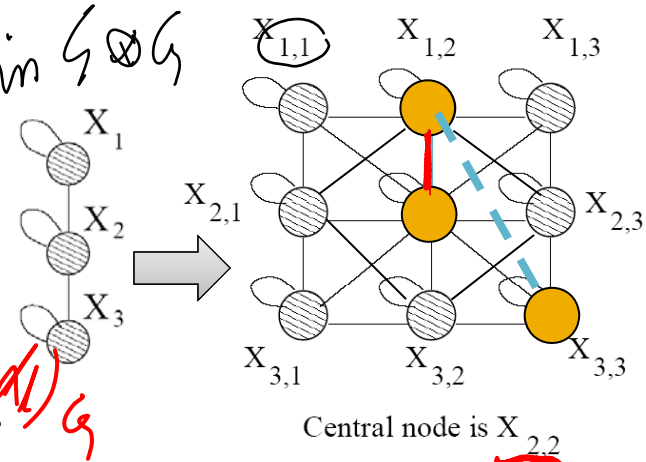
where X are appropriate nodes

$G \otimes G$
(X_{12}, X_{22})
(X_1, X_2)
(X_2, X_2)

- Example:



$d(X_{ij}, X_{kl})$ in $G \otimes G$
$= \max$
$d(X_i, X_k)$ in G, $d(X_j, X_l)$ in G



[PKDD '05]

Stochastic Kronecker graphs

$\Theta_k = (2^k, 1.1^k)$

- Create $N_1 \times N_1$ probability matrix Θ_1 $\quad 0 < \theta_{ij} < 1$
- Compute the kth Kronecker power Θ_k
- For each entry p_{uv} of Θ_k include an edge (u,v) in K_k with probability p_{uv}

Probability of edge p_{ij}

0.5	0.2
0.1	0.3

Θ_1

$N_1 = 2$

$E_1 = 1.1$

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

$\Theta_2 = \Theta_1 \otimes \Theta_1$

Instance matrix K_2

flip biased coins

[Mahdian-Xu, WAW '07]

Stochastic Kronecker

What is known about Stochastic Kronecker?

- Undirected Kronecker graph model with:

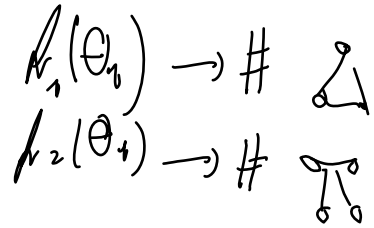
$$\Theta_1 = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

$a > b > c$

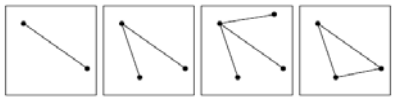
  - Connected, if:
    - $b+c > 1$
  - Connected component of size Θ(n), if:
    - $(a+b)(b+c) > 1$
  - Constant diameter, if:
    - $b+c > 1$
  - Not searchable by a decentralized algorithm

Estimating Kronecker graphs

- Given a real network G 
  Want to estimate initiator matrix:



$$\Theta_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

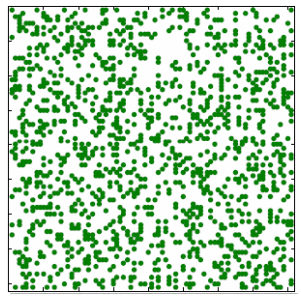
- Method of moments [Owen '09]
  - Compare counts of 
    and solve the system of equations
- Maximum likelihood [ICML '07]
  - arg max P( |  Θ_1)
- SVD [VanLoan-Pitsianis '93]
  - Can solve $\min \left\| G - \Theta_1 \otimes \Theta_2 \right\|_F^2$ using SVD

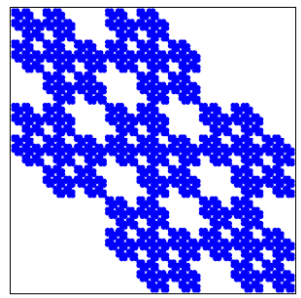
[ICML '07]

Kronecker graphs: Estimation

- Maximum likelihood estimation

$$\arg\max_{\Theta_1} P(\quad \mid \quad \xleftarrow{\text{Kronecker}} \Theta_1)$$





- Naïve estimation takes $O(N!N^2)$:

$$\Theta_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

  - $N!$ for different node labelings:
    - **Our solution:** Metropolis sampling: $N! \rightarrow$ ***(big) const***
  - N^2 for traversing graph adjacency matrix
    - **Our solution:** Kronecker product ($E << N^2$): $N^2 \rightarrow E$
- Do gradient descent

Parameter estimation: Approach

Maximum likelihood estimation

- Given real graph G
- Find Kronecker initiator graph Θ (i.e., $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$) which

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

- We need to (efficiently) calculate

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

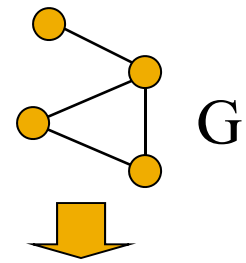
- And maximize over Θ

  (e.g., using gradient descent)

[ICML '07]

KronFit: Likelihood $P(G|\Theta)$

- 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

$P(G|\Theta)$

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

[ICML '07]

Challenge 1: Node correspondence

Θ

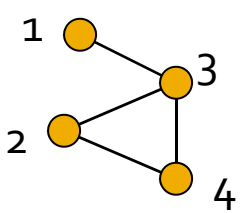
0.5	0.2
0.1	0.3

Θ_k

1	2	3	4
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

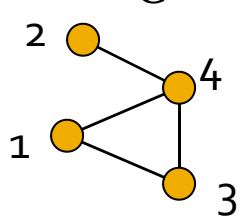
(1, 3, 4, 2) σ

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

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

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

[ICML '07]

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

- Assume we solved the correspondence problem
- Calculating

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

- Takes ***O(N²)*** time
- Infeasible for large graphs (N ~ 10^5)

σ... node labeling

$P(G|\Theta) = P(\bar{G}|\Theta)$

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

$\sum_{(u,v)\in G} -P(\text{no edge}) + P(\text{edge})$

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

[ICML '07]

Solution 1: Node correspondence

- Log-likelihood

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

- Gradient of log-likelihood

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

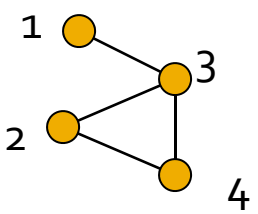
- Sample the permutations from $P(\sigma|G,\Theta)$ and average the gradients

[ICML '07]

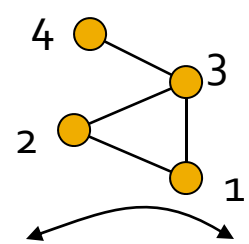
Solution 1: Node correspondence

- Metropolis sampling:
  - Start with a random permutation σ
  - σ' = swap two elements in permutation σ
  - 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!**)

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



Swap node labels 1 and 4



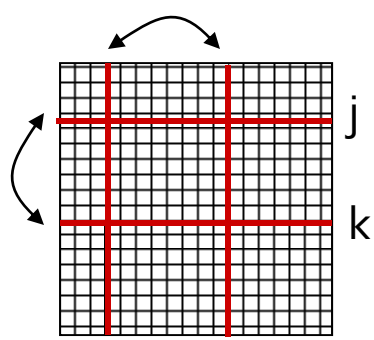
1	1	0	1	0
2	0	1	1	1
3	1	1	1	1
4	0	1	1	1

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

Sampling node labelings (2)

$\sigma^{(0)} := (1, \ldots, N)$
repeat
  Draw j and k uniformly from $(1, \ldots, N)$
  $\sigma^{(i)} := \texttt{SwapElements}(\sigma^{(i-1)}, j, k)$
  Draw u from $U(0,1)$
  if $u > \boxed{\frac{P(\sigma^{(i)}|G,\Theta)}{P(\sigma^{(i-1)}|G,\Theta)}}$ **then**
    $\sigma^{(i)} := \sigma^{(i-1)}$
  end if
  $\mathrm{i} = \mathrm{i} + 1$
until $\sigma^{(i)} \sim P(\sigma|G, \Theta)$
return $\sigma^{(i)}$

Metropolis permutation sampling algorithm



- Need to efficiently calculate the likelihood ratios
- But the permutations $\sigma^{(i)}$ and $\sigma^{(i+1)}$ only differ at 2 positions
- So we only traverse to update 2 rows (columns) of Θ_k
- We can evaluate the likelihood ratio efficiently

[ICML '07]

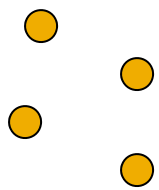
Solution 2: Calculating $P(G|\Theta,\sigma)$

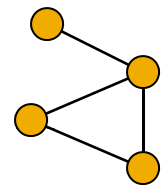
- Calculating naively $P(G|\Theta,\sigma)$ takes $O(N^2)$
- Idea:
  - First calculate likelihood of empty graph, a graph with 0 edges
  - Correct the likelihood for edges that we observe in the graph
- By exploiting the structure of Kronecker product we obtain closed form for likelihood of an empty graph

[ICML '07]

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

- We approximate the likelihood:



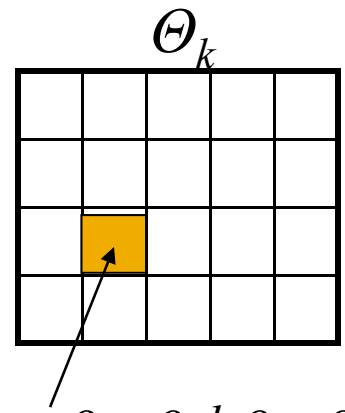


$$l(\Theta) \approx \underbrace{l_e(\Theta) +}_{\text{Empty graph}} \sum_{(u,v)\in G} \underbrace{-\log(1-\Theta_k[\sigma_u,\sigma_v])}_{\text{No-edge likelihood}} + \underbrace{\log(\Theta_k[\sigma_u,\sigma_v])}_{\text{Edge likelihood}}$$

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

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

- Real graphs are sparse so we first calculate likelihood of empty graph
- Probability of edge *(i,j)* is in general $p_{ij} = \theta_1^a \theta_2^b \theta_3^c \theta_4^d$
- By using Taylor approximation to p_{ij} and summing the multinomial series we obtain:



$$p_{ij} = \theta_1^a \theta_2^b \theta_3^c \theta_1^d$$

$$l_e(\Theta) = \sum_{i,j=1}^{N} \log(1 - p_{ij}) \approx -\left(\sum_{i,j=1}^{N_1} \theta_{i,j}\right)^k - \frac{1}{2}\left(\sum_{i,j=1}^{N_1} \theta_{i,j}^2\right)^k$$

Taylor approximation
$\log(1-x) \sim -x - 0.5\, x^2$

- We approximate the likelihood:

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

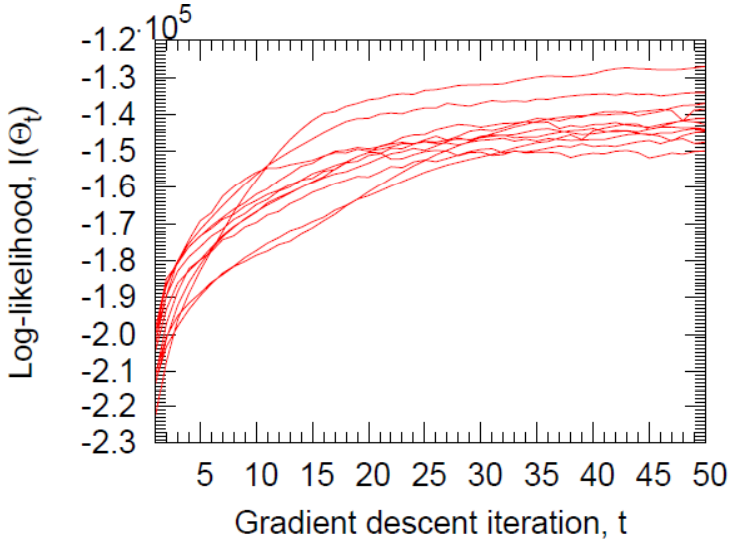
Experiments: real networks

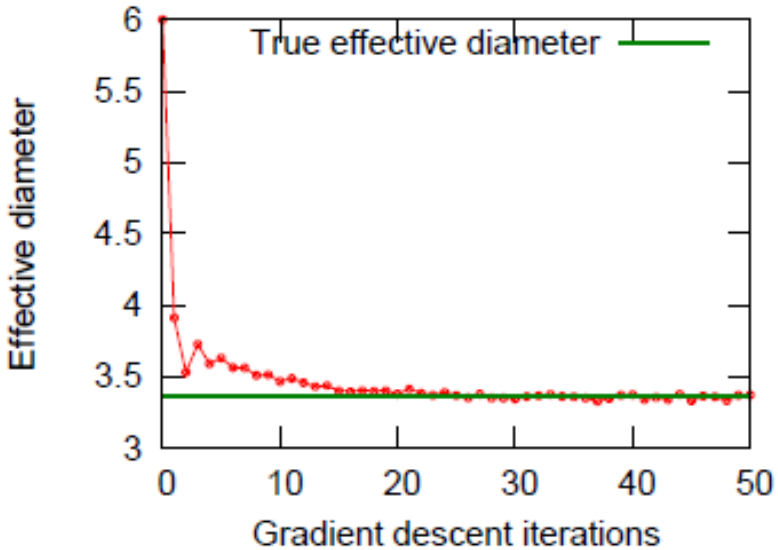
$\Theta = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

- Experimental setup
  - Given real graph G
  - Gradient descent from random initial point
  - Obtain estimated parameters Θ
  - Generate synthetic graph K using Θ
  - 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

Convergence of fitting

- Can gradient descent recover true parameters?
  - Generate a graph from random parameters
  - Start at random point and use gradient descent
  - We recover true parameters 98% of the times



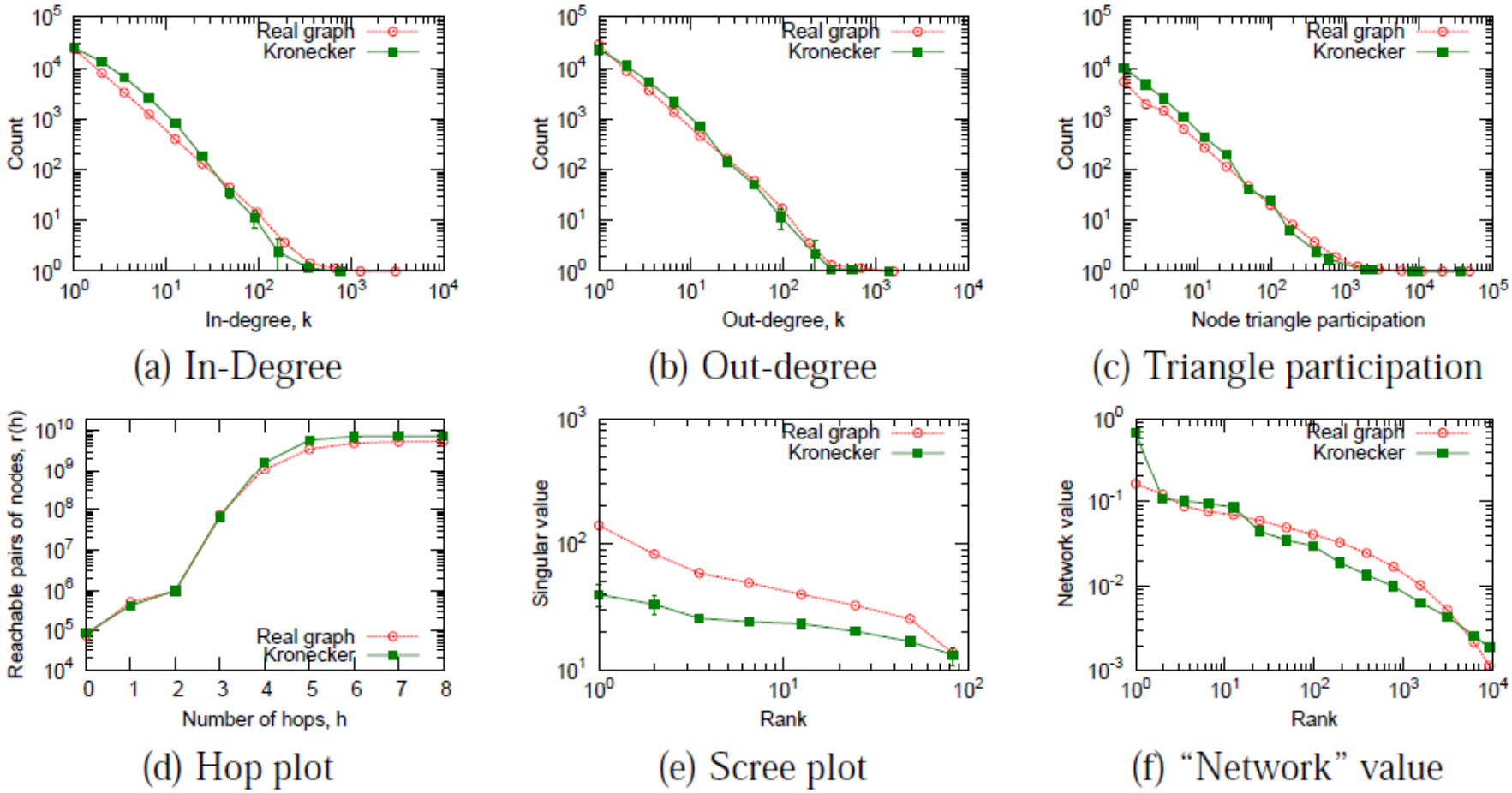


[ICML '07]

Estimation: Epinions (n=76k, m=510k)

- Real and Kronecker are very close:

$$\Theta_1 = \begin{bmatrix} 0.99 & 0.54 \\ 0.49 & 0.13 \end{bmatrix}$$



(a) In-Degree (b) Out-degree (c) Triangle participation

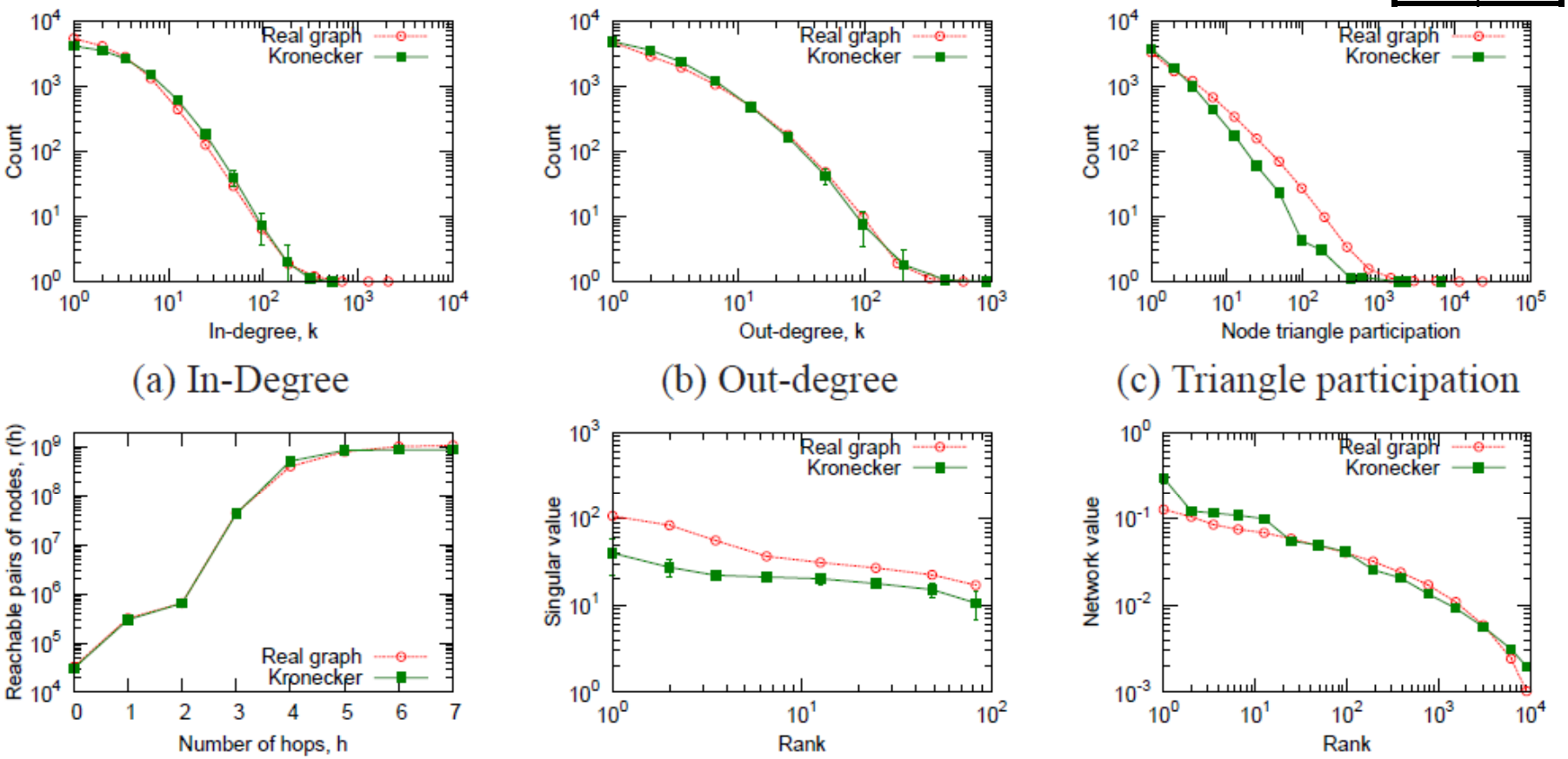
(d) Hop plot (e) Scree plot (f) "Network" value

[JMLR '10]

Estimation: Blogs (n=32k, m=318k)

- Real and Kronecker are very close:

$$\Theta_1 = \begin{bmatrix} 0.99 & 0.57 \\ 0.51 & 0.22 \end{bmatrix}$$



(a) In-Degree (b) Out-degree (c) Triangle participation

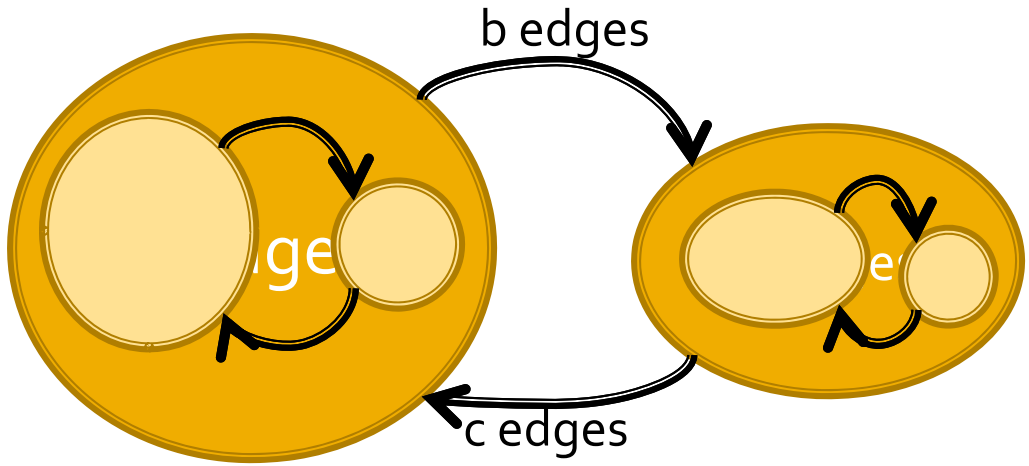
(d) Hop plot (e) Scree plot (f) "Network" value

[JMLR '10]

Kronecker & Network structure

- What do estimated parameters tell us about the network structure?

$$\Theta = K_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

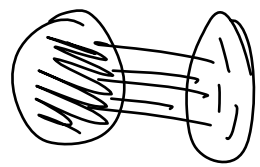


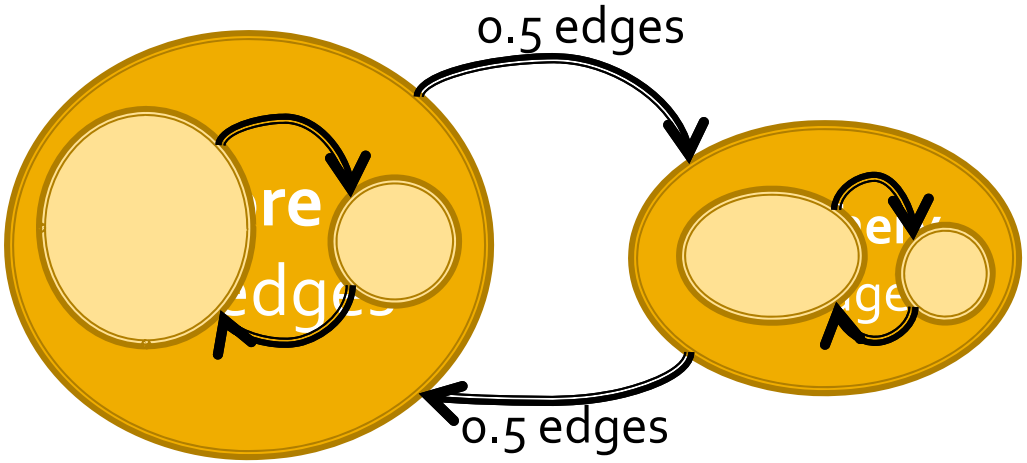
[JMLR '10]

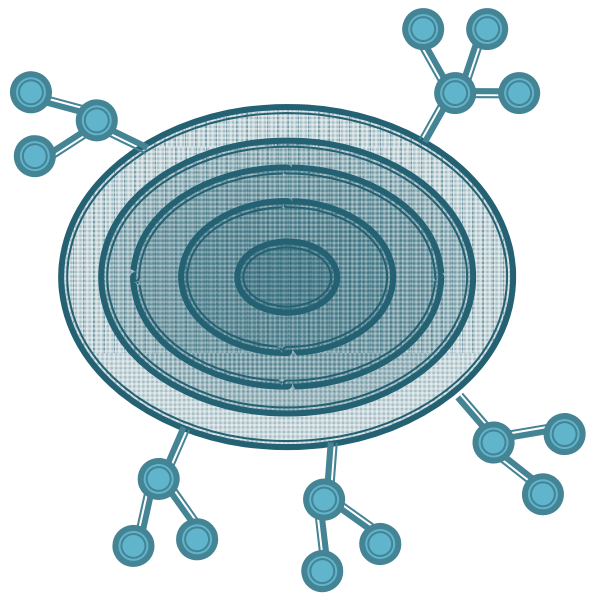
Kronecker & Network structure

- What do estimated parameters tell us about the network structure?

$$K_1 = \begin{bmatrix} 0.9 & 0.5 \\ 0.5 & 0.1 \end{bmatrix}$$





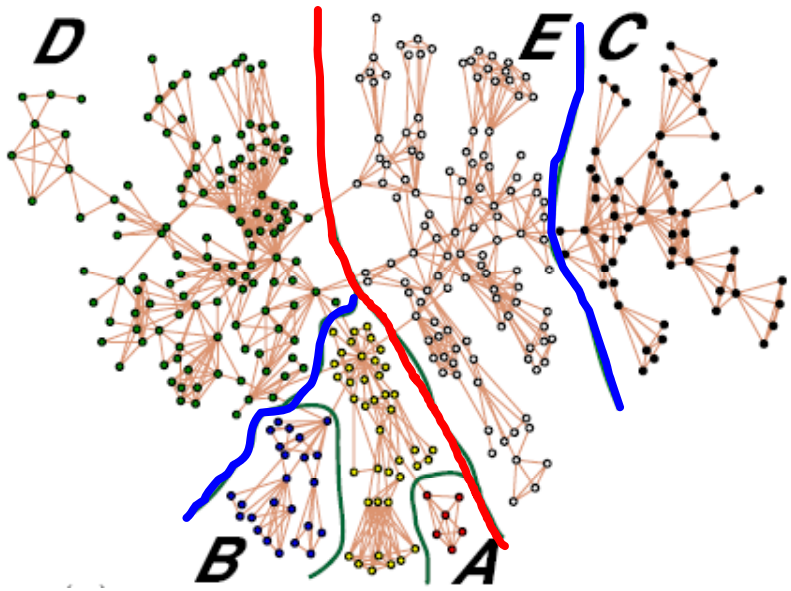


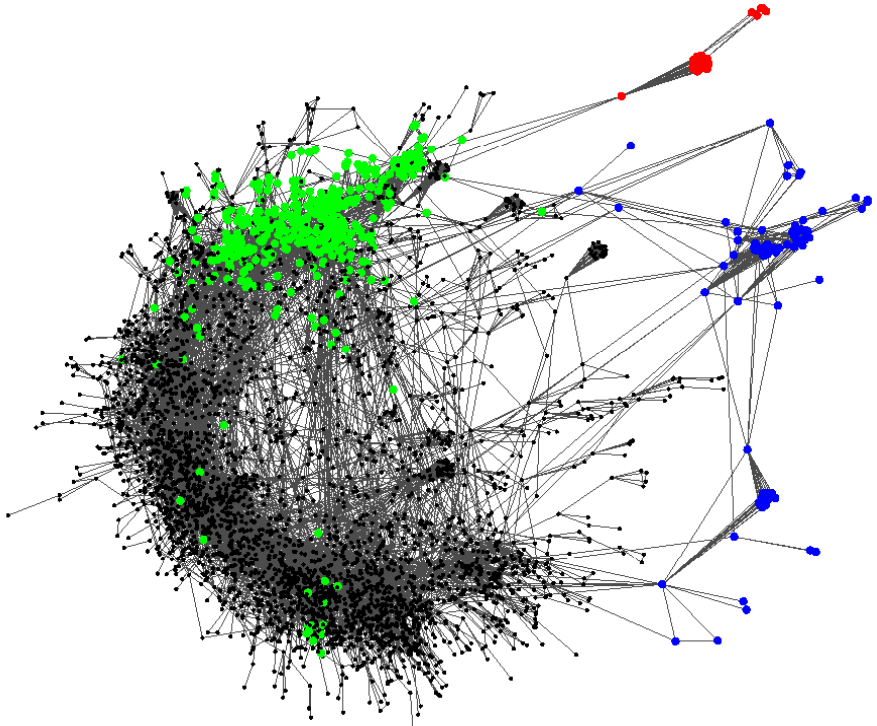
Nested Core-periphery

[JMLR '10]

Small vs. Large networks

- Small and large networks are very different:



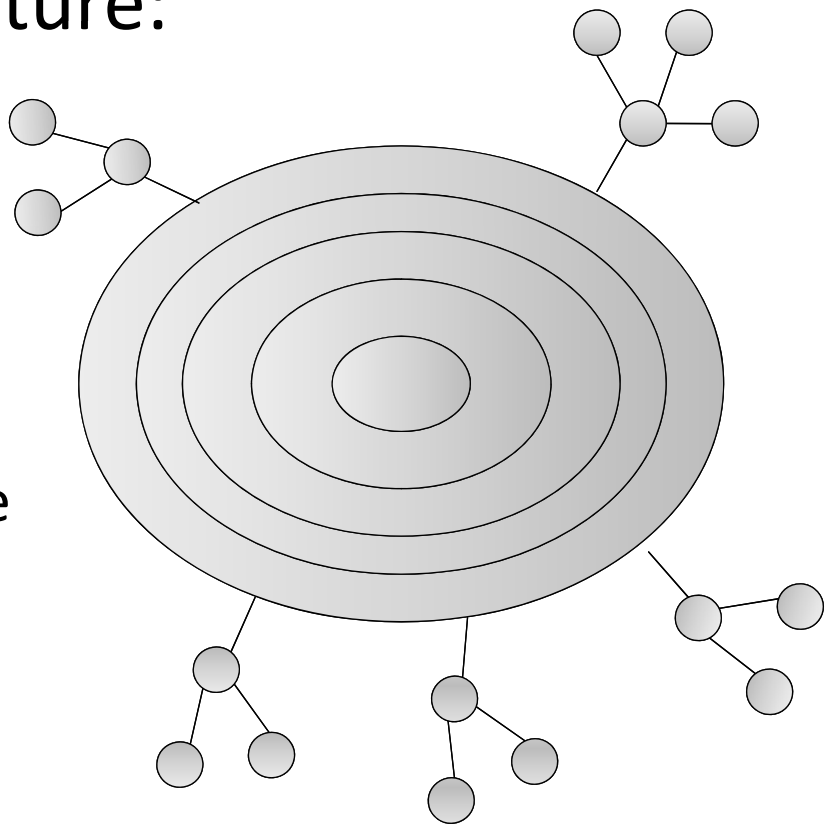


$$K_1 = \begin{bmatrix} 0.99 & 0.17 \\ 0.17 & 0.82 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} 0.99 & 0.54 \\ 0.49 & 0.13 \end{bmatrix}$$

Implications (1)

- Large scale network structure:
  - Large networks are different from small networks and manifolds
  - Nested Core-periphery
    - Recursive onion-like structure of the network where each layer decomposes into a core and periphery



Implications (2)

$$K_1 = \begin{bmatrix} 0.99 & 0.55 \\ 0.55 & 0.15 \end{bmatrix}$$

- 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)·(0.55+0.15) > 1. **Yes!**
- Real graphs are in the in the parameter region analogous to the giant component of an extremely sparse G_{np}

[WAW '10]

Kronecker model: Alternative view

- Each node u has associated binary vector A_u
  - Think of it as feature vector
- Initiator matrix K acts like a "similarity" matrix
- Probability of a link between nodes u, v:

$$P(u,v) = \prod_{i=1}^{k} K_1(A_u(i), A_v(i))$$

$$K_1 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & a & b \\ 1 & c & d \end{array}$$

	v_1	v_2	v_3	v_4
v_1	a·a	a·b	b·a	b·b
v_2	a·c	a·d	b·c	b·d
v_3	c·a	c·b	d·a	d·b
v_4	c·c	c·d	d·c	d·d

$K_2 = K_1 \otimes K_1$

=

	v_1	v_2	v_3	v_4
v_1	a	b	a	b
v_2	c	d	c	d
v_3	a	b	a	b
v_4	c	d	c	d

a **b** **c** **d**

$v_2 = (0,1)$

$v_3 = (1,0)$ $\quad P(v_2, v_3) = b \cdot c$

[WAW '10]

Multiplicative Attribute Graphs

- For each node *u* we have a binary vector A_u
- For each edge (u,v) determine prob.

$$P(u,v) = \prod_{i=1}^{k} K_i(A_u(i), A_v(i))$$

$$A_u = [\ 0 \quad 0 \quad 1 \quad 0\ ]$$
$$A_v = [\ 0 \quad 1 \quad 1 \quad 0\ ]$$

$$K_i = \begin{bmatrix} \alpha_1 & \beta_1 \\ \beta_1 & \gamma_1 \end{bmatrix} \begin{bmatrix} \alpha_2 & \beta_2 \\ \beta_2 & \gamma_2 \end{bmatrix} \begin{bmatrix} \alpha_3 & \beta_3 \\ \beta_3 & \gamma_3 \end{bmatrix} \begin{bmatrix} \alpha_4 & \beta_4 \\ \beta_4 & \gamma_4 \end{bmatrix}$$

$\alpha_i = \alpha_j$

$$P(u,v) = \alpha_1 \cdot \beta_2 \cdot \gamma_3 \cdot \alpha_4$$

[WAW '10]

MAG: Interpretation

- How to think of

$$K_i = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A_u = [\ 0 \quad 0 \quad 1 \quad 0\ ]$$
$$A_v = [\ 0 \quad 1 \quad 1 \quad 0\ ]$$

$$K_i = \begin{bmatrix} \alpha_1 & \beta_1 \\ \beta_1 & \gamma_1 \end{bmatrix} \begin{bmatrix} \alpha_2 & \beta_2 \\ \beta_2 & \gamma_2 \end{bmatrix} \begin{bmatrix} \alpha_3 & \beta_3 \\ \beta_3 & \gamma_3 \end{bmatrix} \begin{bmatrix} \alpha_4 & \beta_4 \\ \beta_4 & \gamma_4 \end{bmatrix}$$

$$P(u,v) = \alpha_1 \cdot \beta_2 \cdot \gamma_3 \cdot \alpha_4$$

- Attribute-attribute similarity matrix:
  - Can model homophily: $\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$
  - Heterophily: $\begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}$
  - Core-periphery: $\begin{bmatrix} 0.9 & 0.5 \\ 0.5 & 0.1 \end{bmatrix}$

[WAW '10]

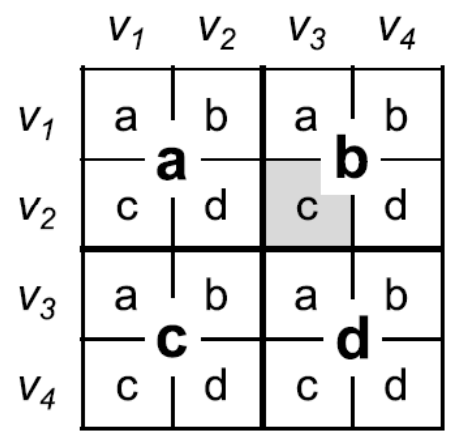
Simplified MAG model

- For each node *u* generate a binary vector A_u
  - draw k ($k \approx \log_2(|V|)$) independent samples from a Bernoulli(λ)
- For each pair of nodes (u,v) determine an edge prob.

$$P(u,v) = \prod_{i=1}^{k} K(A_u(i), A_v(i))$$

Feature vector view: Question

- 2 ingredients of Kronecker model:
  - (1) Each of 2^k nodes has a unique binary vector of length k
    - Node id expressed binary is the vector
  - (2) The initiator matrix K



- Question:
  - What if ingredient (1) is dropped?
    - i.e., do we need high variability of feature vectors?

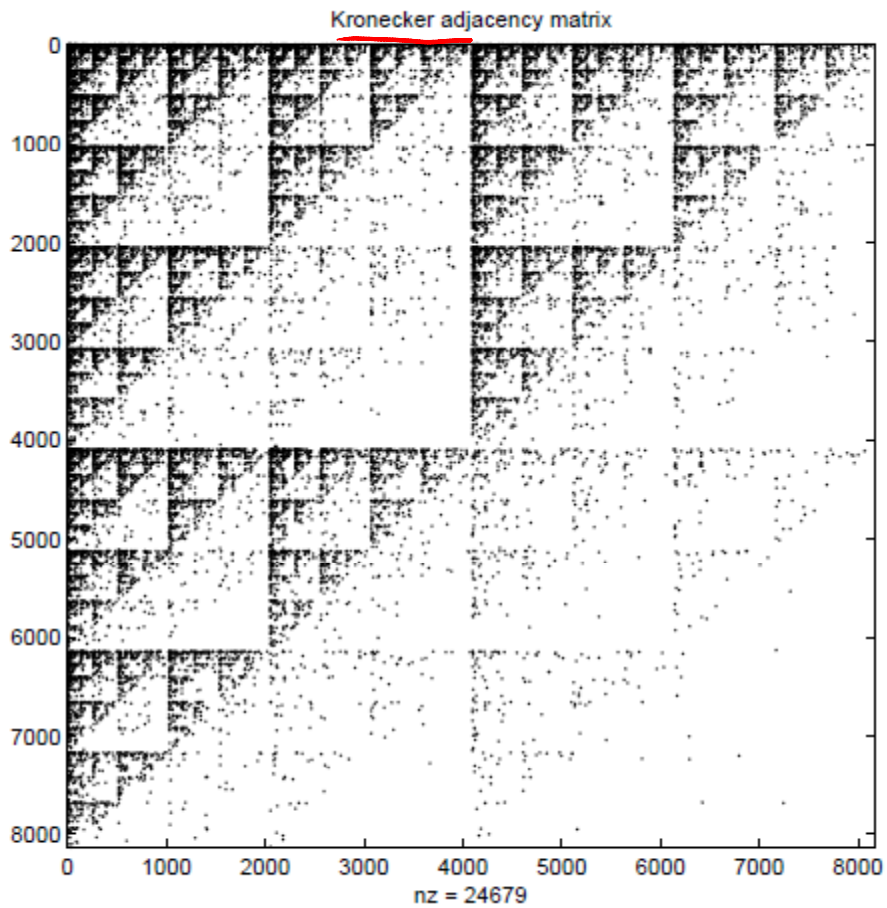
Comparison: Adjacency matrices

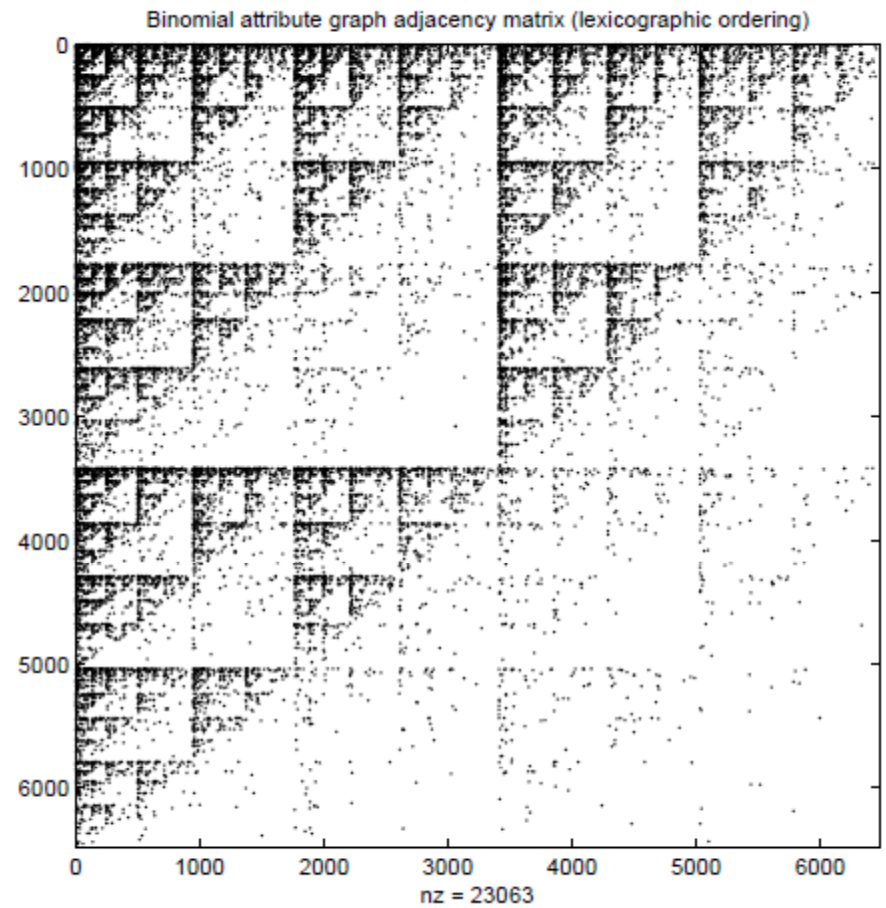
K

$A_u = [0\ 1\ 0\ 0\ 1]$

- Adjacency matrices:

K



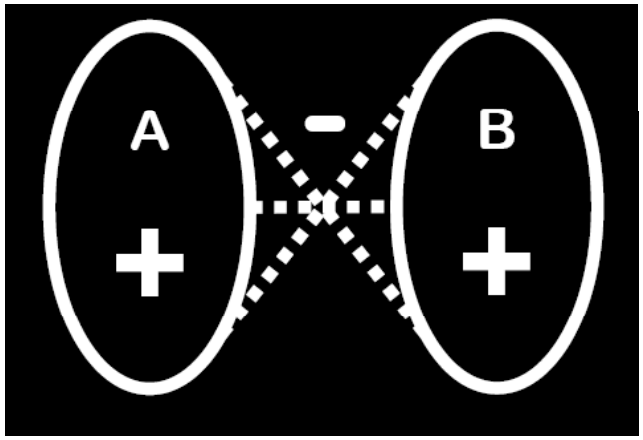


Coalitions in signed networks

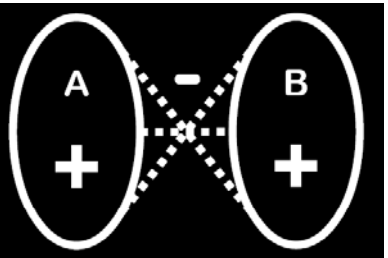
- Received 19 entries
- Top score: 14,690
- Top 10:

Rank	Name	Score	Reward
1	Wang,Fan	0	10%
2	Cui,Jingyu	+1	10%
3	Preston,Dan	+3	8%
4	Pham,Peter Thien Tan	+4	8%
5	Wang,Dakan	+9	6%
6	Moreinis,Stanislav	+13	4%
7	Wang,Chunyan	+17	2%
8	Kim,Hyung Jin	+19	2%
9	Wu,Yu	+28	
10	Jin,Ye	+31	

Random partitioning gives score of ~50,000



Winning solutions



- (1) Everyone used some form of greedy hill-climbing:
  - Repeat until no improvement:
    - (1) Pick a node (multiple nodes), move it to the other side if it improves a score
    - (2) Pick an edge, move the endpoints so that score is most improved
- (2) Randomization techniques and simulated annealing to escape local minima
  - Repeat (1) until no improvement
  - Randomize and restart
- (3) Signed Laplacian matrix