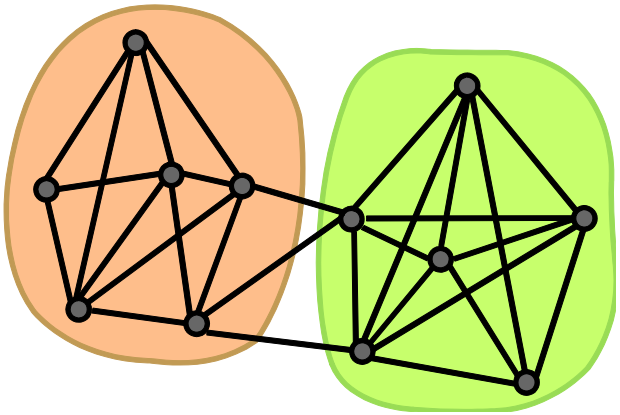
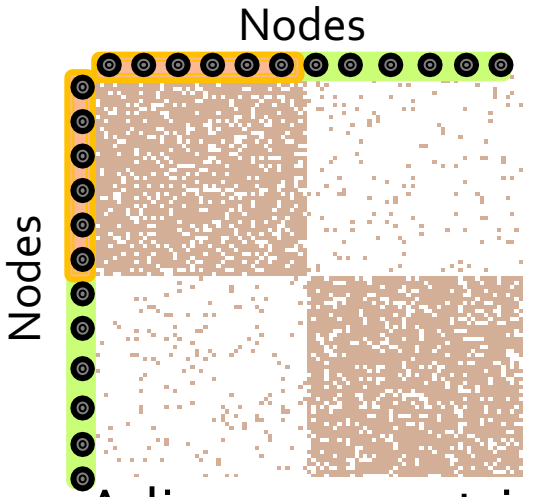
# Community Detection: Overlapping Communities

CS224W: Analysis of Networks
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

# Non-overlapping Communities



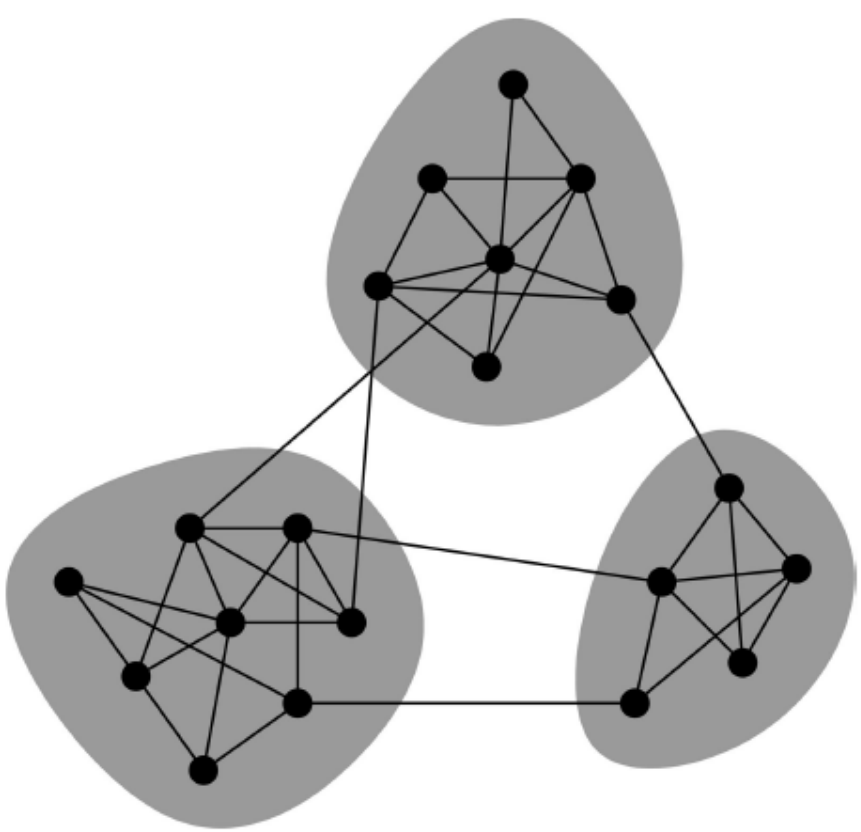
Network

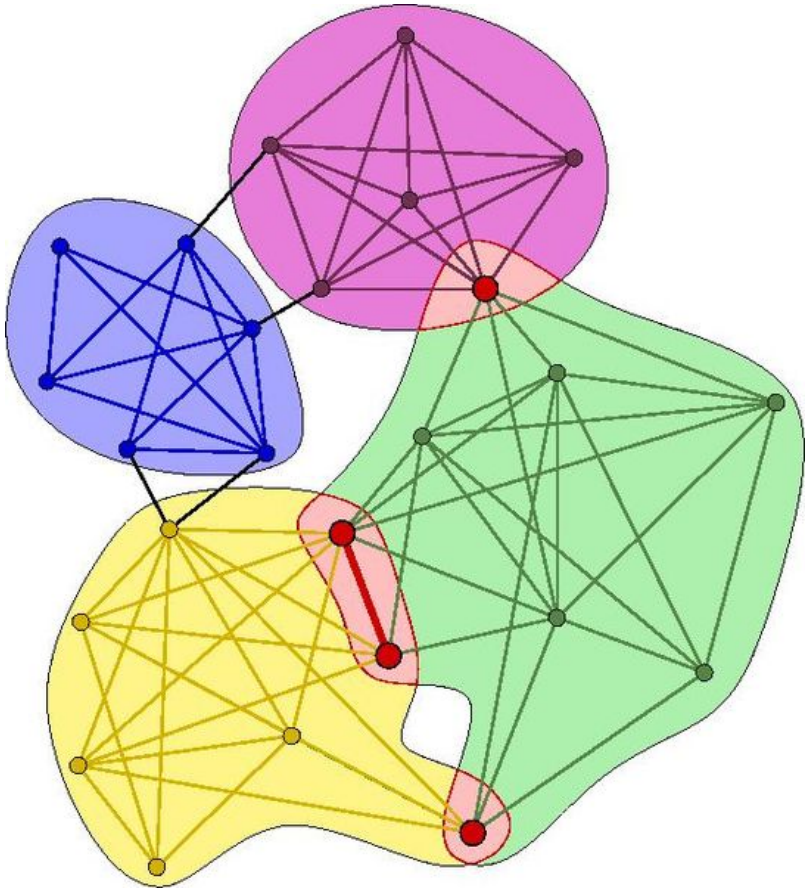



Adjacency matrix

# Overlapping Communities

- **Non-overlapping vs. overlapping communities**

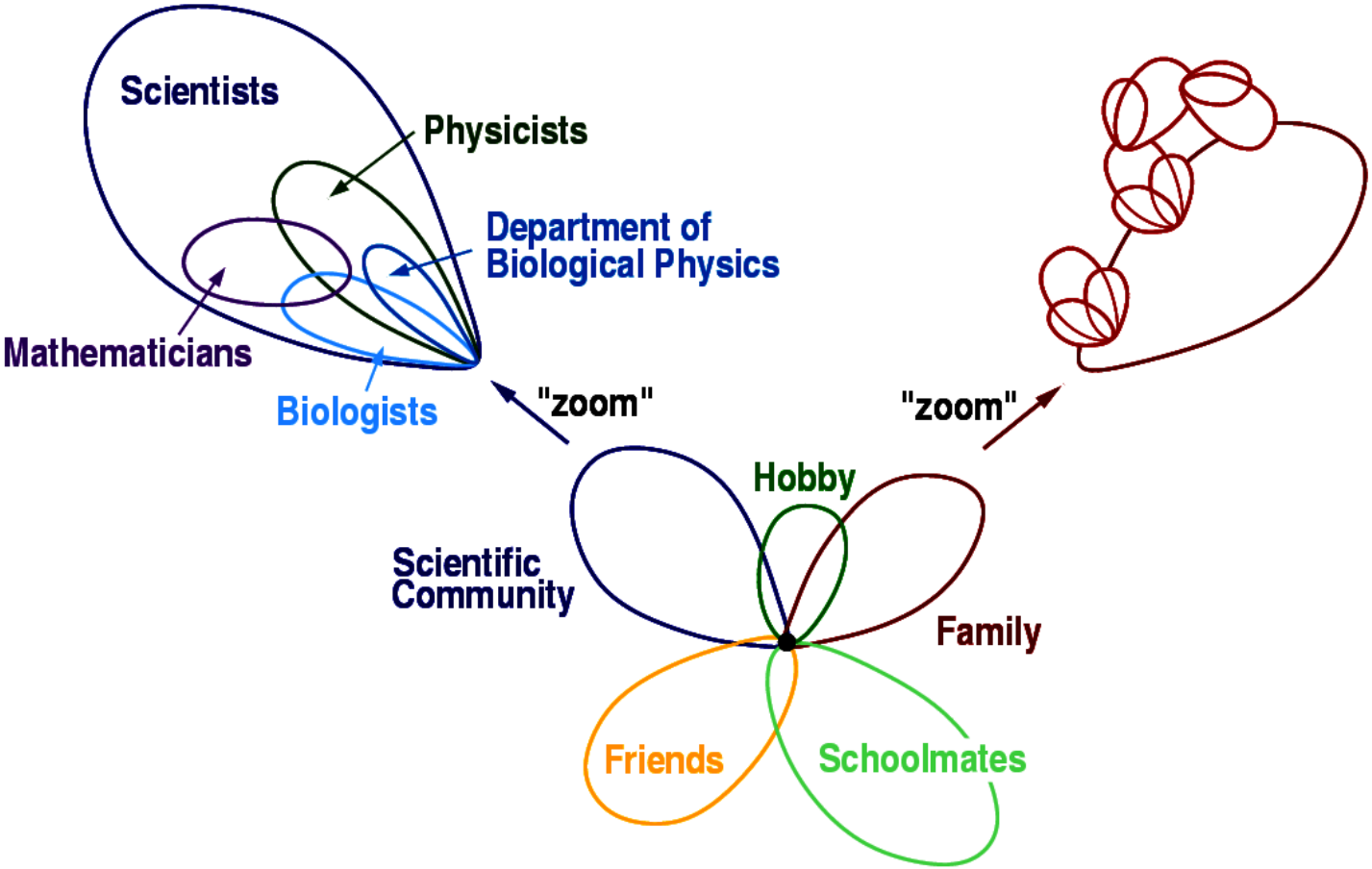




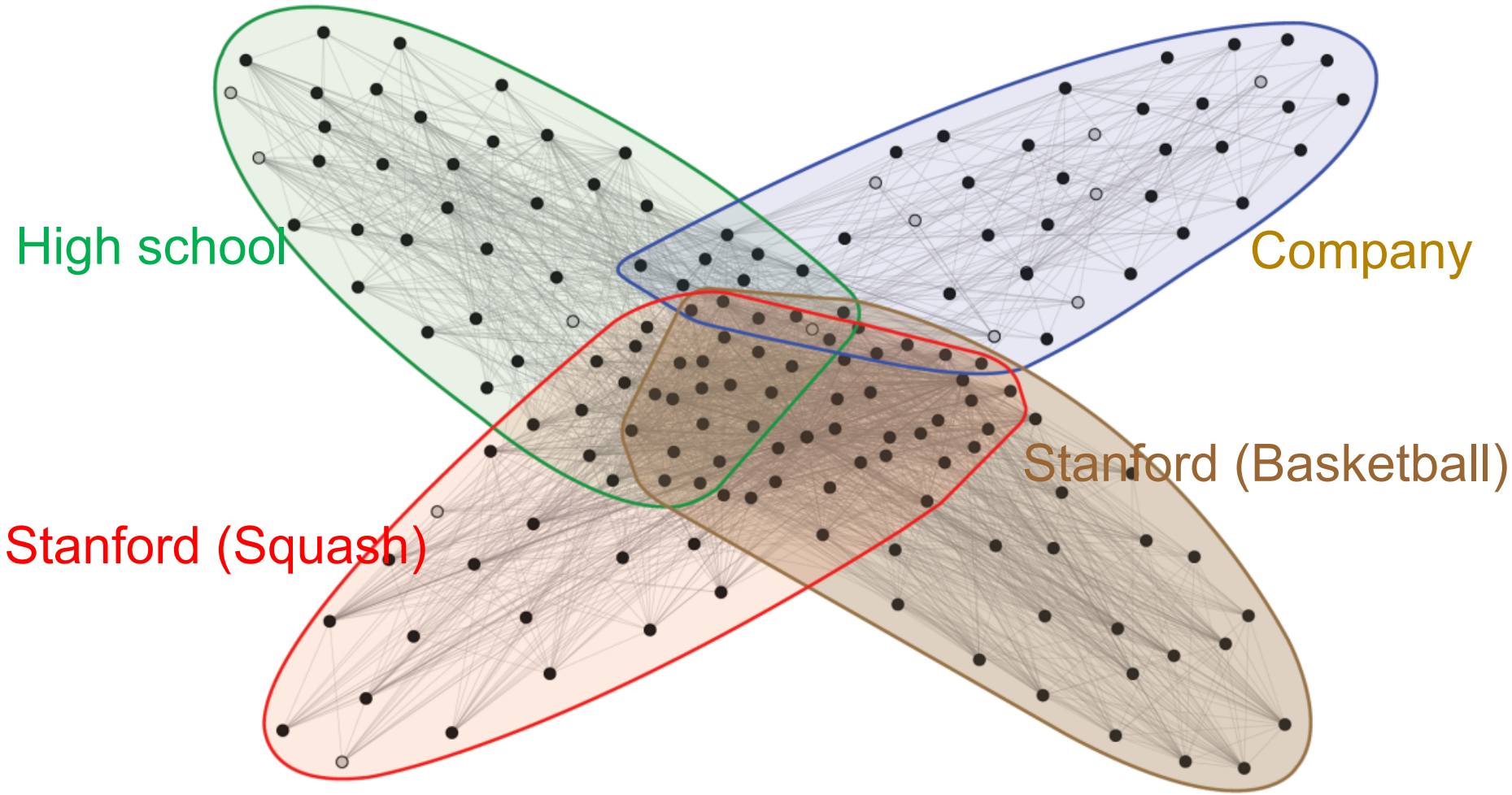
[Palla et al., '05]

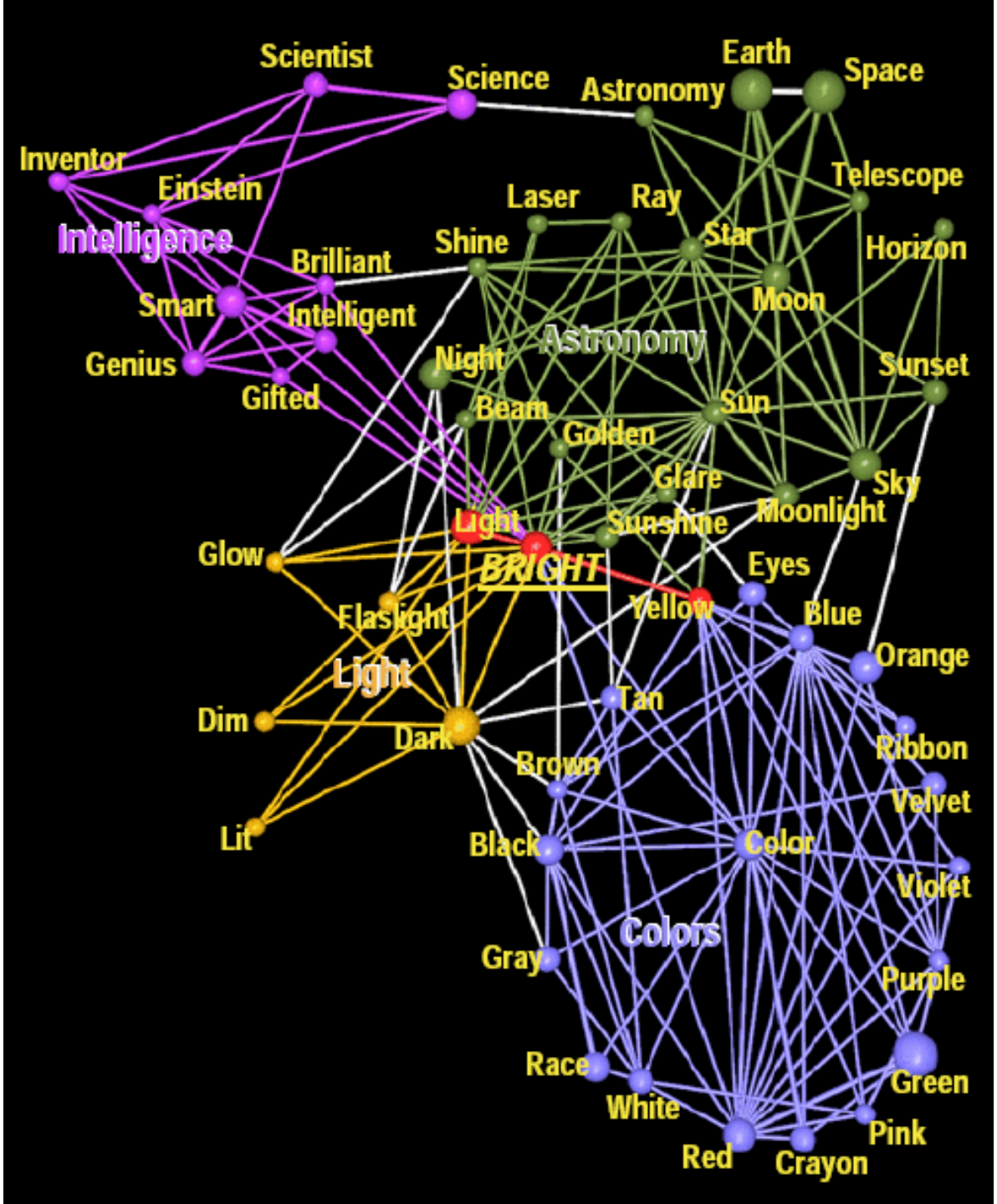
# Overlaps of Social Circles

- A node can belong to many social "circles"



# What if communities overlap?

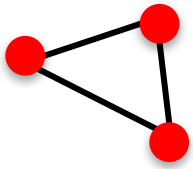



Scientist
Science
Astronomy
Earth
Space
Inventor
Einstein
Intelligence
Laser
Ray
Telescope
Star
Horizon
Shine
Brilliant
Smart
Intelligent
Moon
Genius
Gifted
Astronomy
Night
Sunset
Beam
Sun
Golden
Glare
Sky
Moonlight
Sunshine
Light
BRIGHT
Glow
Eyes
Yellow
Blue
Flashlight
Orange
Light
Tan
Dim
Dark
Ribbon
Brown
Velvet
Lit
Black
Color
Violet
Colors
Gray
Purple
Race
Green
White
Pink
Red
Crayon

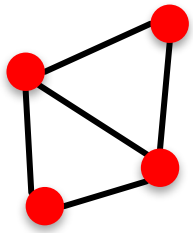
[Palla et al., '05]

# Clique Percolation Method (CPM)

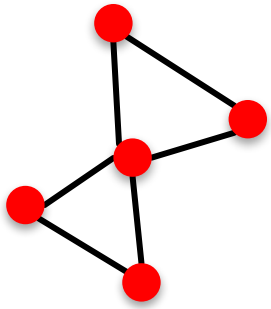
- **Two nodes belong to the same community if they can be connected through adjacent $k$-cliques:**
  - **$k$-clique:**
    - Fully connected graph on $k$ nodes
  - **Adjacent $k$-cliques:**
    - overlap in $k-1$ nodes
- **$k$-clique community**
  - Set of nodes that can be reached through a sequence of adjacent $k$-cliques



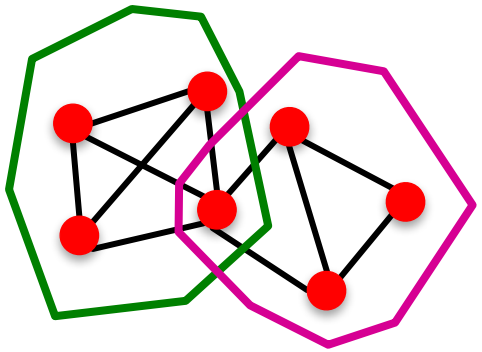
3-clique



Adjacent 3-cliques



Non-adjacent 3-cliques

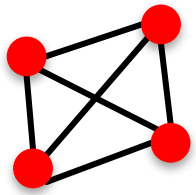


Two overlapping 3-clique communities

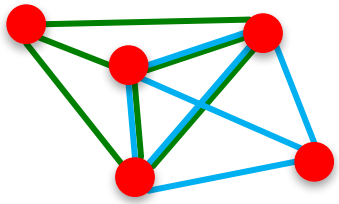
[Palla et al., '05]

# Clique Percolation Method (CPM)

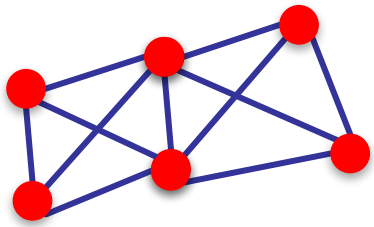
- **Two nodes belong to the same community if they can be connected through adjacent $k$-cliques:**



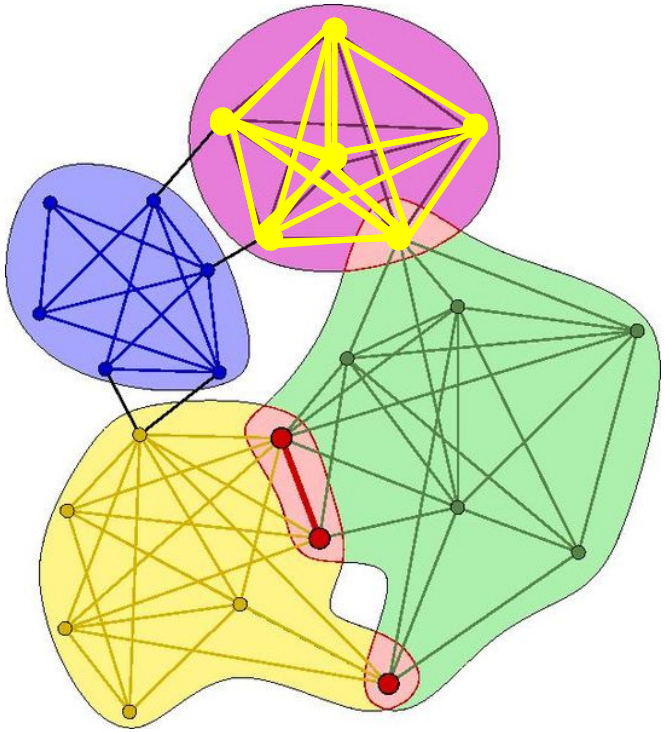
4-clique



Adjacent 4-cliques



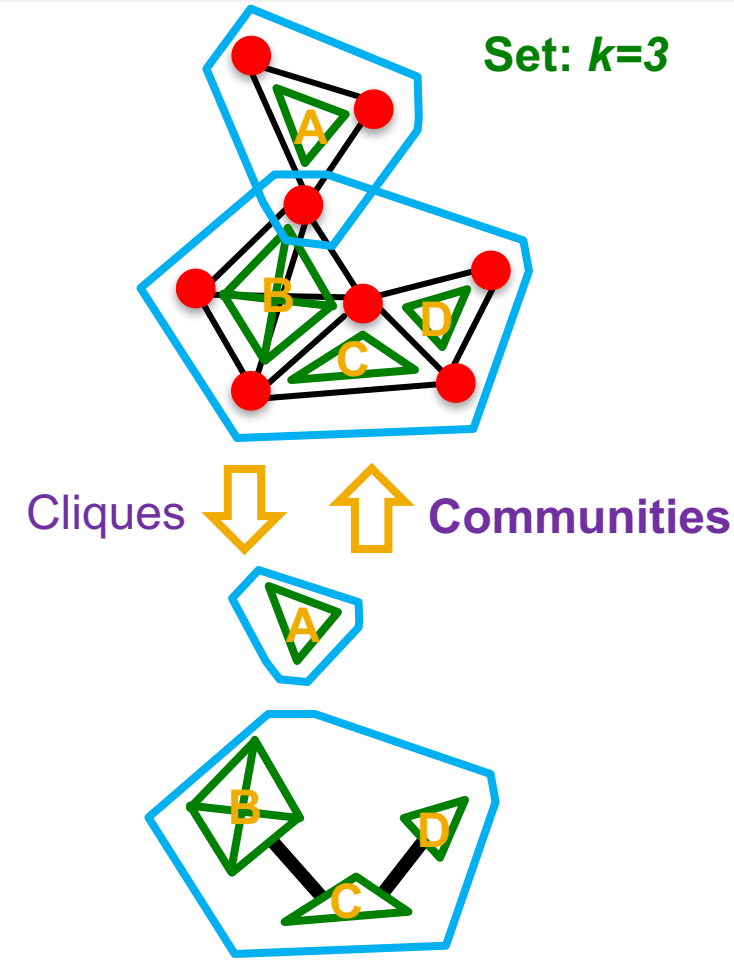
Non-adjacent 4-cliques



Communities for k=4

# CPM: Steps

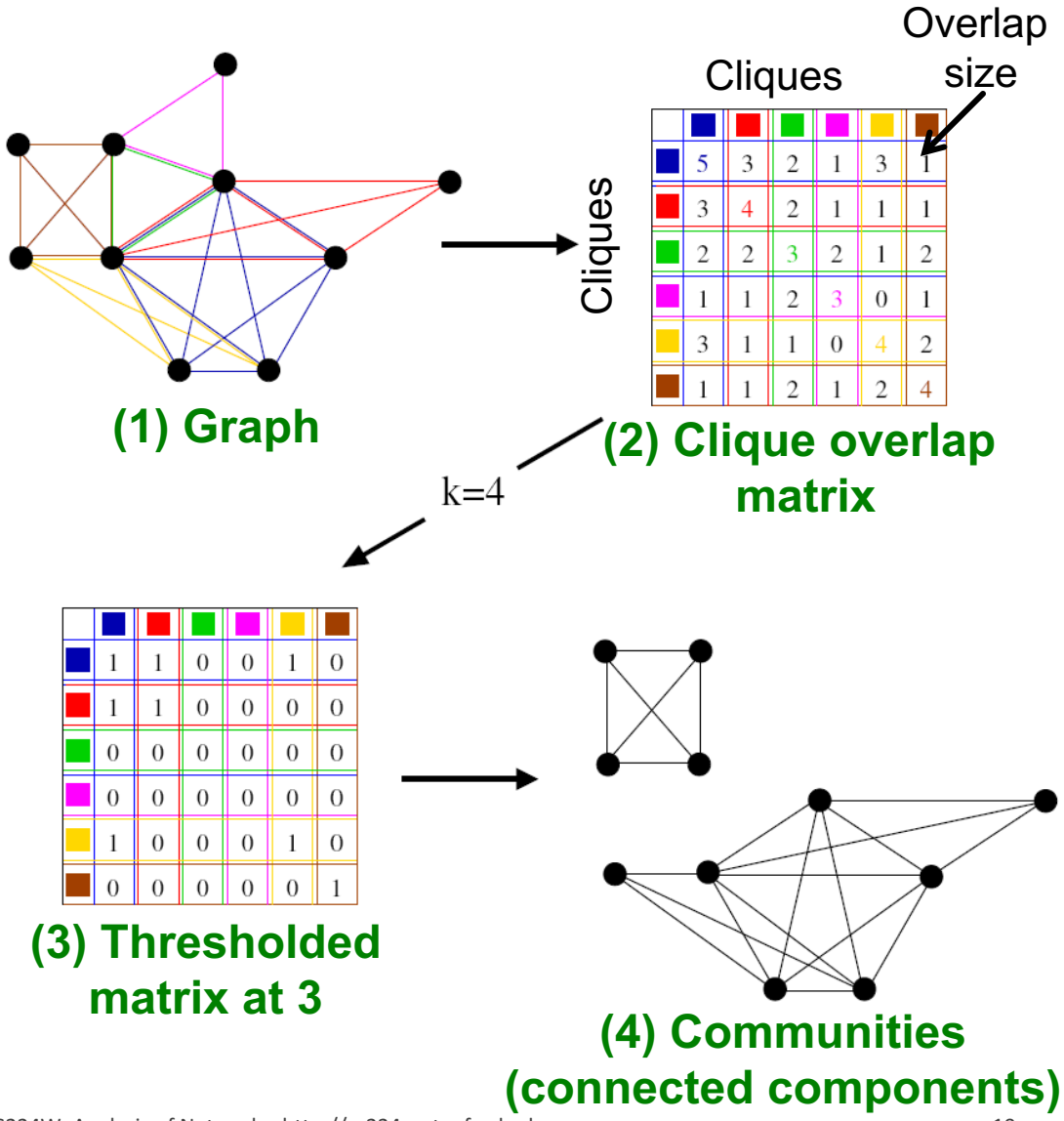
- **Clique Percolation Method:**
  - **Find maximal-cliques**
    - Def: Clique is maximal if no superset is a clique
  - **Clique overlap super-graph:**
    - Each clique is a super-node
    - Connect two cliques if they overlap in at least $k-1$ nodes
  - **Communities:**
    - Connected components of the clique overlap matrix




- **How to set $k$?**
  - Set $k$ so that we get the "richest" (most widely distributed cluster sizes) community structure

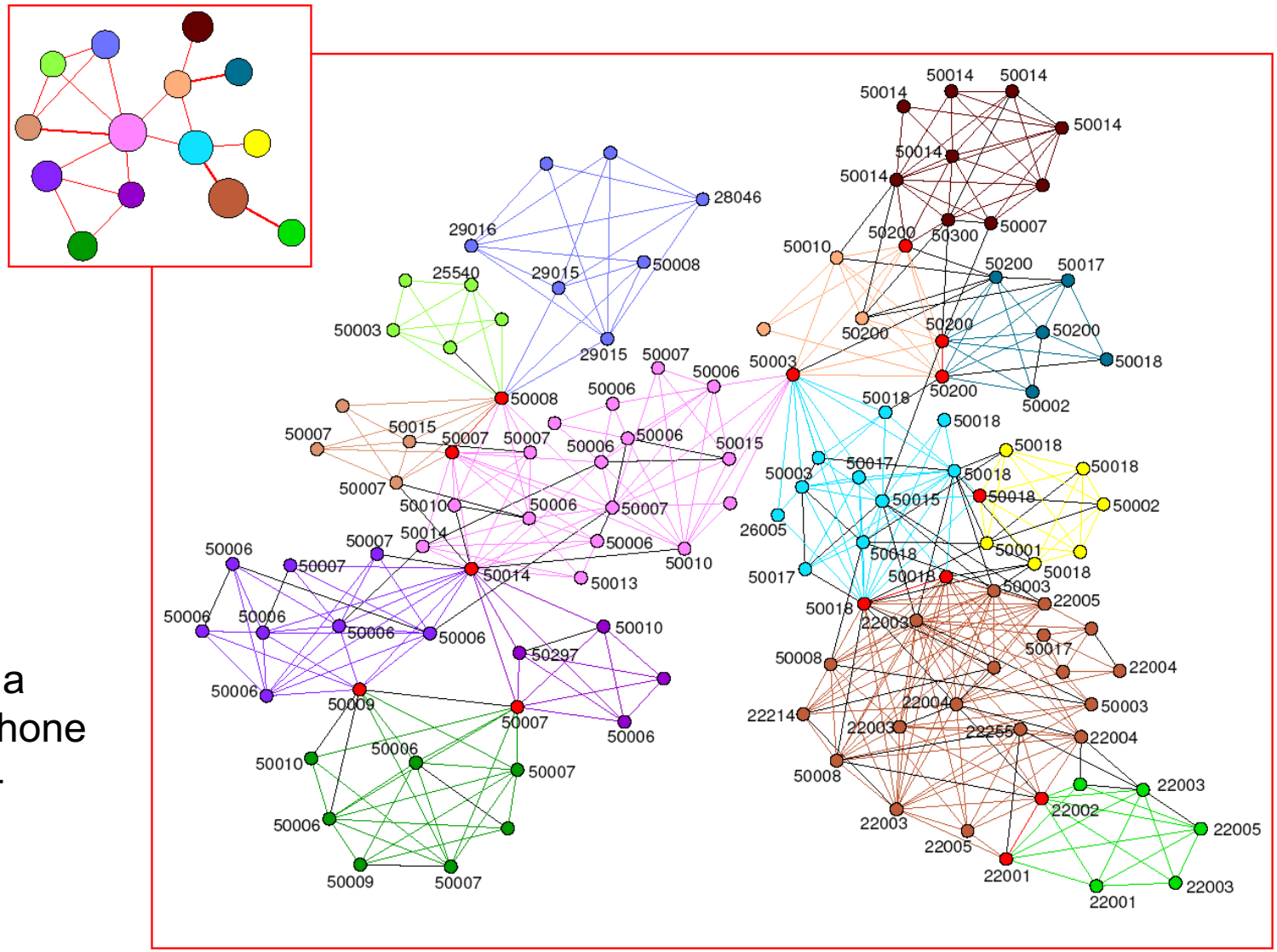
# CPM method: Example

- Start with graph
- Find maximal cliques
- Create clique overlap matrix $A$
  - Rows/Cols are max-cliques, entry is number of nodes in common
- Threshold the matrix at value ***k-1***
  - If $a_{ij} < k - 1$ *set* $0$
- Communities are the connected components of the thresholded matrix



[Palla et al., '07]

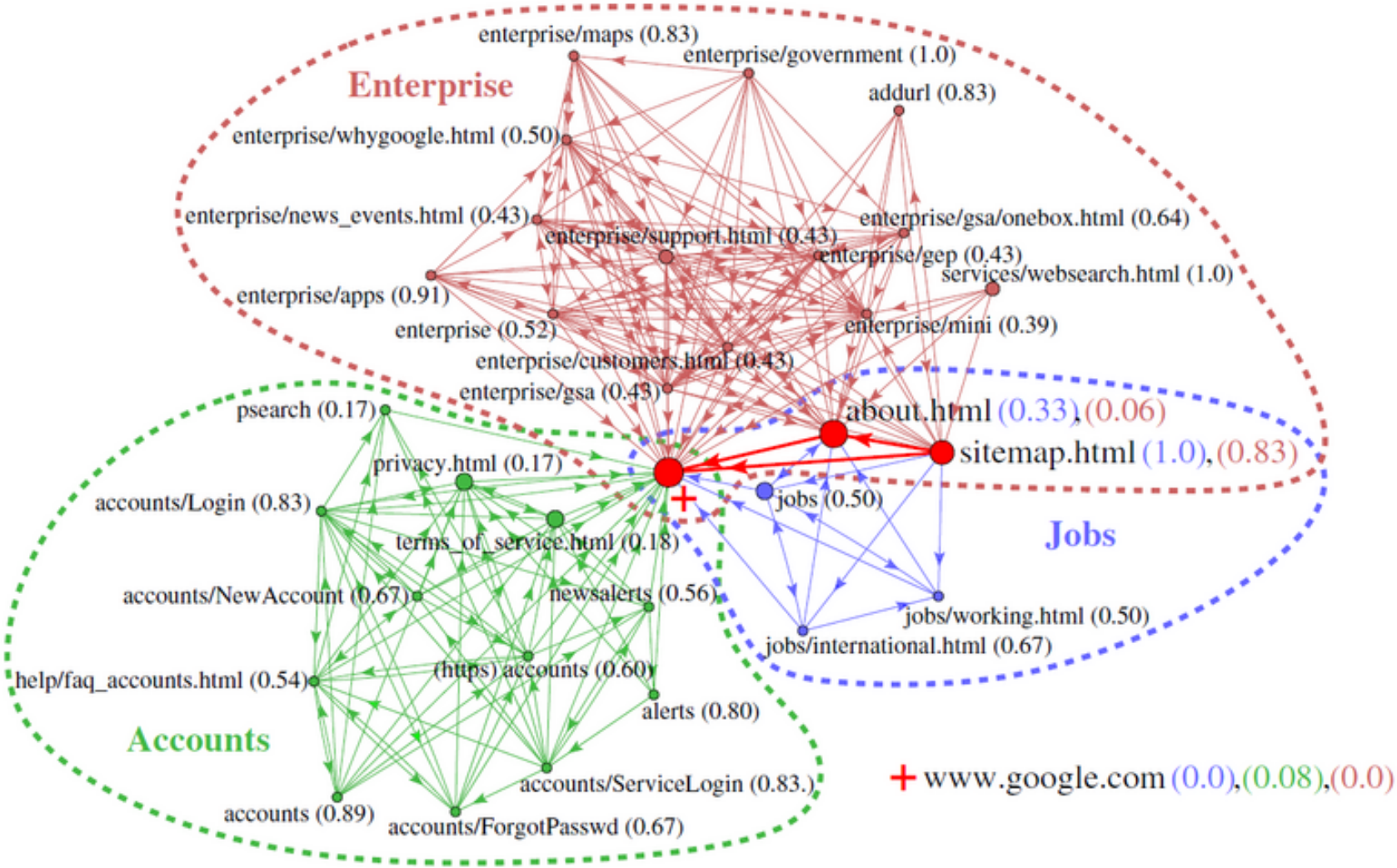
# Example: Phone-Call Network




Communities in a "tiny" part of a phone call network of 4 million users
[Palla et al., '07]

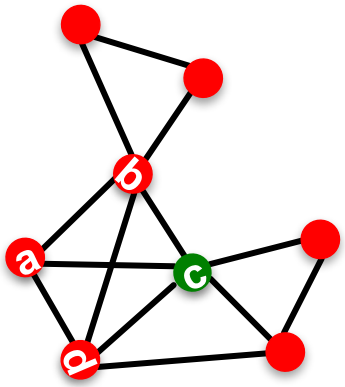
[Farkas et. al. 07]

# Example: Website



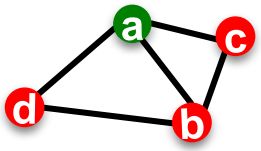
# How to Find Maximal Cliques?

- **No nice way, hard combinatorial problem**
- **Maximal clique:** Clique that can't be extended
  - $\{a, b, c\}$ is a clique but not maximal clique
  - $\{a, b, c, d\}$ is maximal clique
- **Algorithm:** Sketch
  - Start with a seed node
  - Expand the clique around the seed
  - Once the clique cannot be further expanded we found the maximal clique
  - **Note:**
    - This method will generate the same clique multiple times



# How to Find Maximal Cliques?

- Start with a seed vertex $a$
- **Goal:** Find the max clique $Q$ that $a$ belongs to
  - **Observation:**
    - If some $x$ belongs to $Q$ then it is a neighbor of $a$
      - **Why?** If $a, x \in Q$ but edge $(a, x)$ does not exist, $Q$ is not a clique!
- **Recursive algorithm:**
  - $Q$ … current clique
  - $R$ … candidate vertices to expand the clique to
- **Example:** Start with $a$ and expand around it



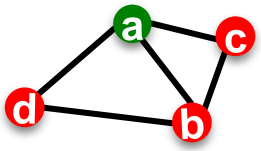

**Q=**
**R=**


Steps of the recursive algorithm

$\Gamma(u)$…neighbor set of *u*

# How to Find Maximal Cliques?

- Start with a seed vertex $a$
- **Goal:** Find the max clique $Q$ that $a$ belongs to
  - **Observation:**
    - If some $x$ belongs to $Q$ then it is a neighbor of $a$
      - **Why?** If $a, x \in Q$ but edge $(a, x)$ does not exist, $Q$ is not a clique!
- **Recursive algorithm:**
  - $Q$ … current clique
  - $R$ … candidate vertices to expand the clique to
- **Example:** Start with $a$ and expand around it




| | | | | | |
|---|---|---|---|---|---|
| **Q=** | {a} | {a,b} | {a,b,c} | bktrack | {a,b,d} |
| **R=** | {b,c,d} | {b,c,d} ∩Γ(b)={c,d} | {d}∩Γ(c)={} | | {c}∩Γ(d)={} |

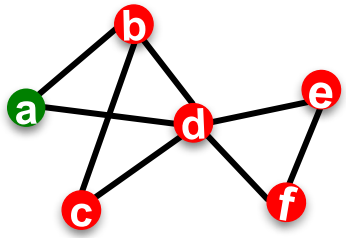
Steps of the recursive algorithm

Γ(u)…neighbor set of *u*

# How to Find Maximal Cliques?

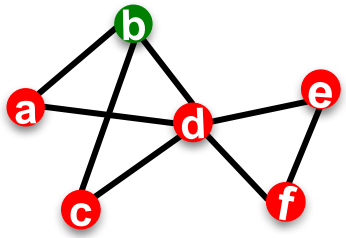
- $Q$ … current clique
- $R$ … candidate vertices

- **Expand(R,Q)**
  - **while** R ≠ {}
    - p = vertex in R
    - $Q_p$ = Q ∪ {p}
    - $R_p$ = R ∩ Γ(p)
    - **if** $R_p$ ≠ {}: Expand($R_p$,$Q_p$)
      **else:** output $Q_p$
    - R = R − {p}



# How to Find Maximal Cliques?

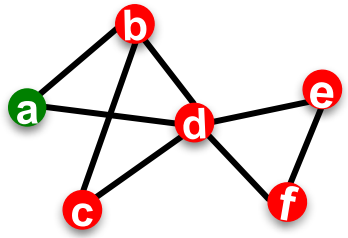
- $Q$ … current clique
- $R$ … candidate vertices
- **Expand(R,Q)**
  - **while** R ≠ { }
    - p = vertex in R
    - $Q_p$ = Q ∪ {p}
    - $R_p$ = R ∩ Γ(p)
    - **if** $R_p$ ≠ { }: Expand($R_p$,$Q_p$)
      **else:** output $Q_p$
    - R = R − {p}




**Start: Expand(V, {})**
R={a,…f}, Q={}
p = {b}
$Q_p$ = {b}
$R_p$ = {a,c,d}
**Expand($R_p$, Q):**
R = {a,c,d}, Q={b}
p = {a}
$Q_p$ = {b,a}
$R_p$ = {d}
**Expand($R_p$, Q):**
R = {d}, Q={b,a}
p = {d}
$Q_p$ = {b,a,d}
$R_p$ = {} : **output {b,a,d}**
p = {c}
$Q_p$ = {b,c}
$R_p$ = {d}
**Expand($R_p$, Q):**
R = {d}, Q={b,c}
p = {d}
$Q_p$ = {b,c,d}
$R_p$ = {} : **output {b,c,d}**

# How to Find Maximal Cliques?

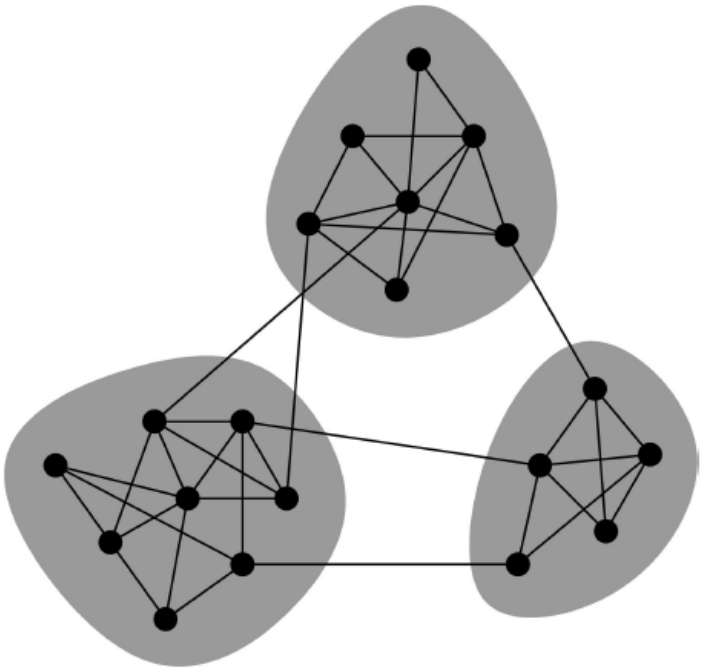
- **How to prevent maximal cliques from being generated multiple times?**
  - Only output cliques that are lexicographically minimum
    - $\{a, b, c\} < \{b, a, c\}$
  - **Even better:** Only expand to the nodes higher in the lexicographical order

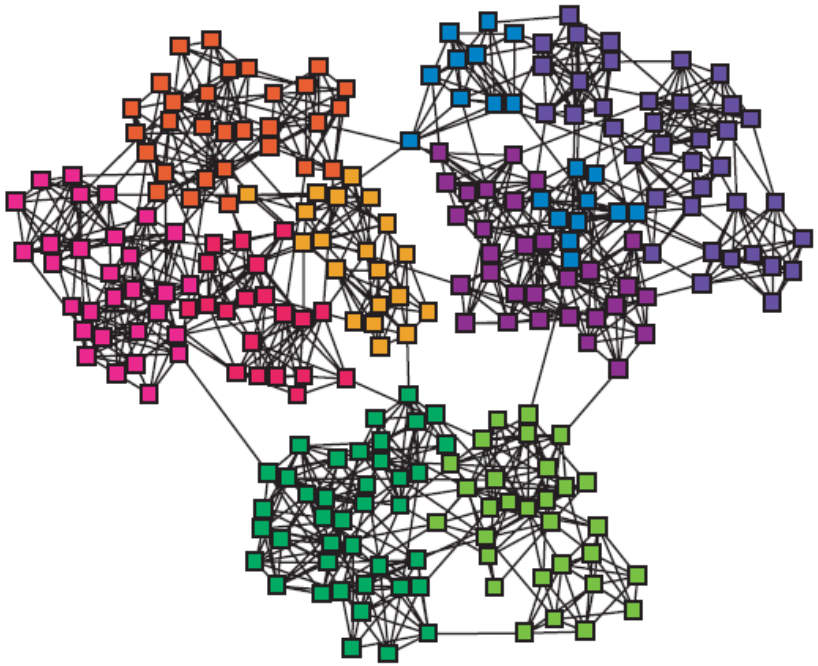


# How to Model Networks with Communities?

# Network and Communities

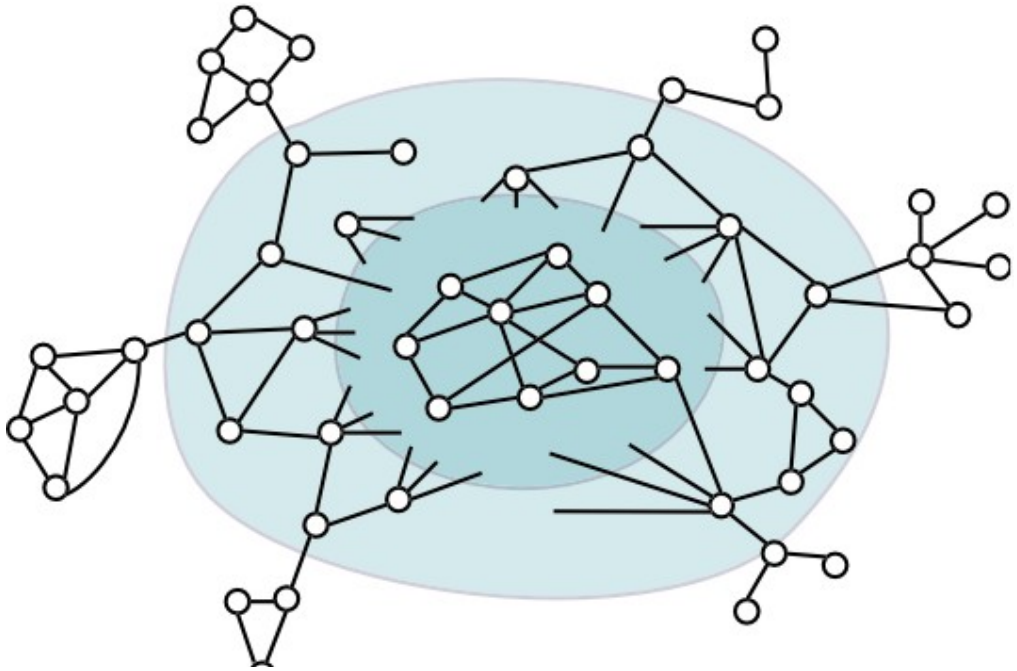
- **How should we think about large scale organization of clusters in networks?**
  - **Finding:** Community Structure





# Network and Communities

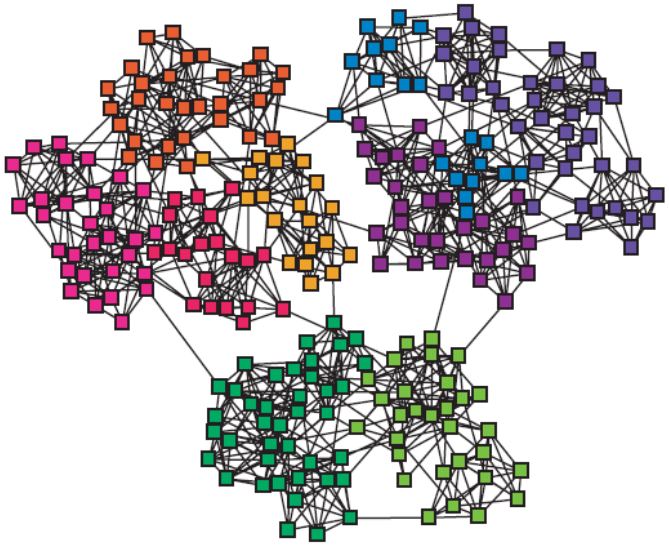
- **How should we think about large scale organization of clusters in networks?**
  - **Finding:** Core-periphery structure



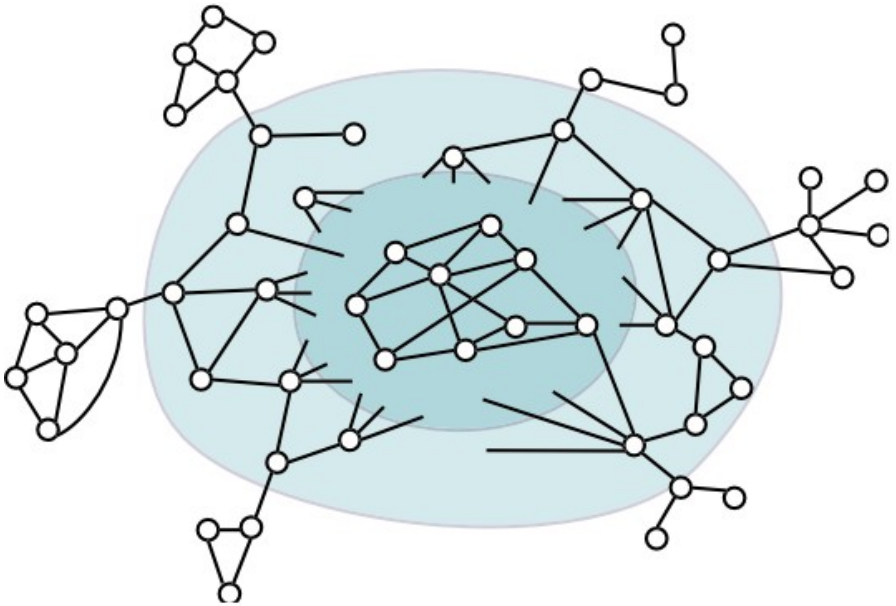
**Nested Core-Periphery**

# Network and Communities

- **How do we reconcile these two views?**
  **(and still do community detection)**



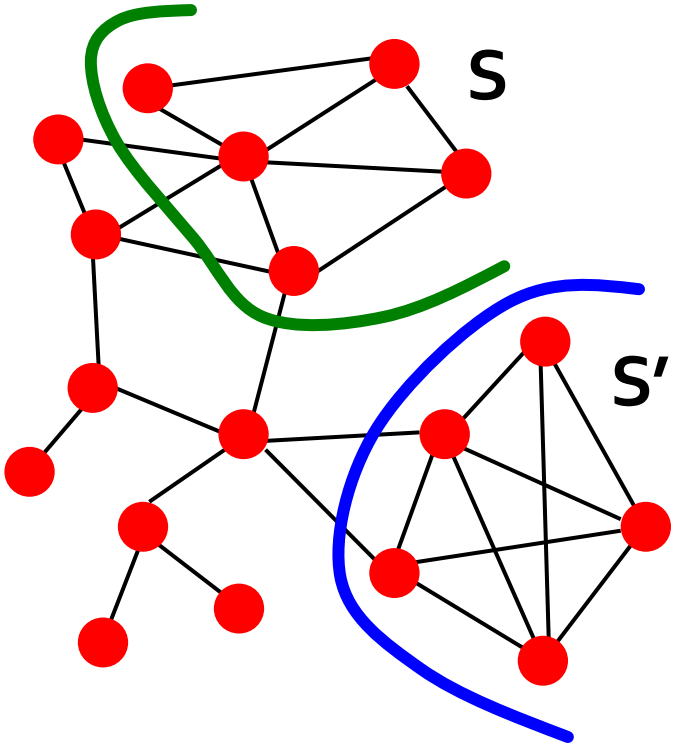
**vs.**



**Community structure** **Core-periphery**

# Community Score

- **How community-like is a set of nodes?**
- **A good cluster *S* has**
  - Many edges internally
  - Few edges pointing outside
- What's a good metric:
  **Conductance**




$$\phi(S) = \frac{|\{(i,j) \in E; i \in S, j \notin S\}|}{\sum_{s \in S} d_s}$$

**Small conductance** corresponds to good clusters

**Note:** We are assuming $|S| < |V|/2$, $d_s$ degree of node s

# Network Community Profile Plot

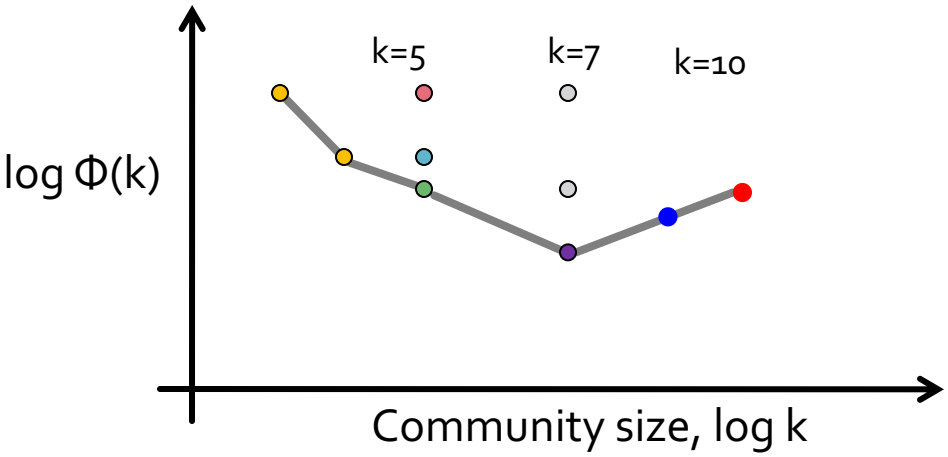
(Note |S| < |V|/2)

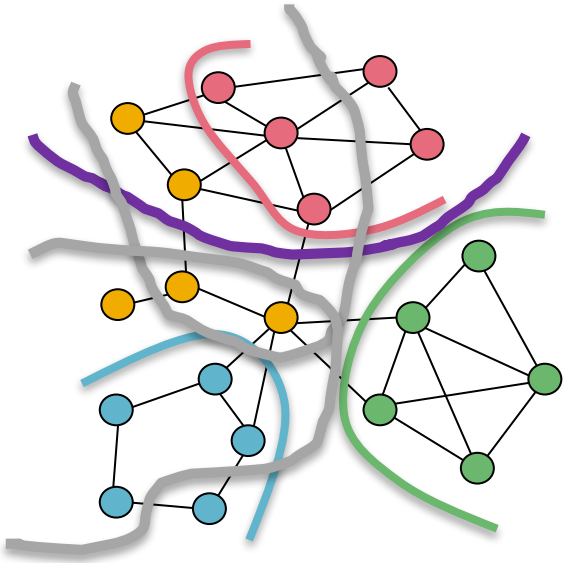
- Define:

  Network community profile (**NCP**) plot

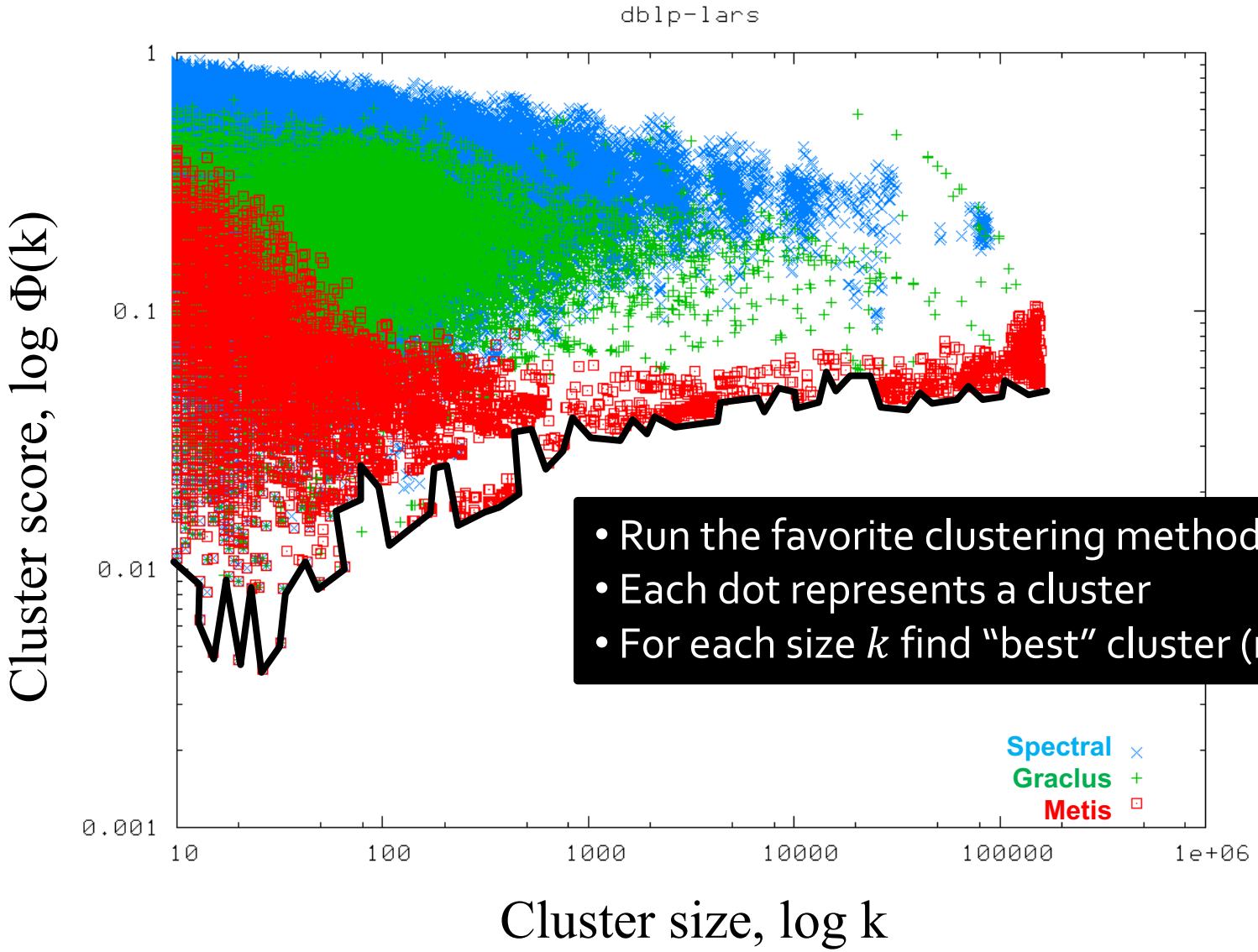
  Plot the score of **best** community of size *k*

$$\Phi(k) = \min_{S \subset V, |S| = k} \phi(S)$$



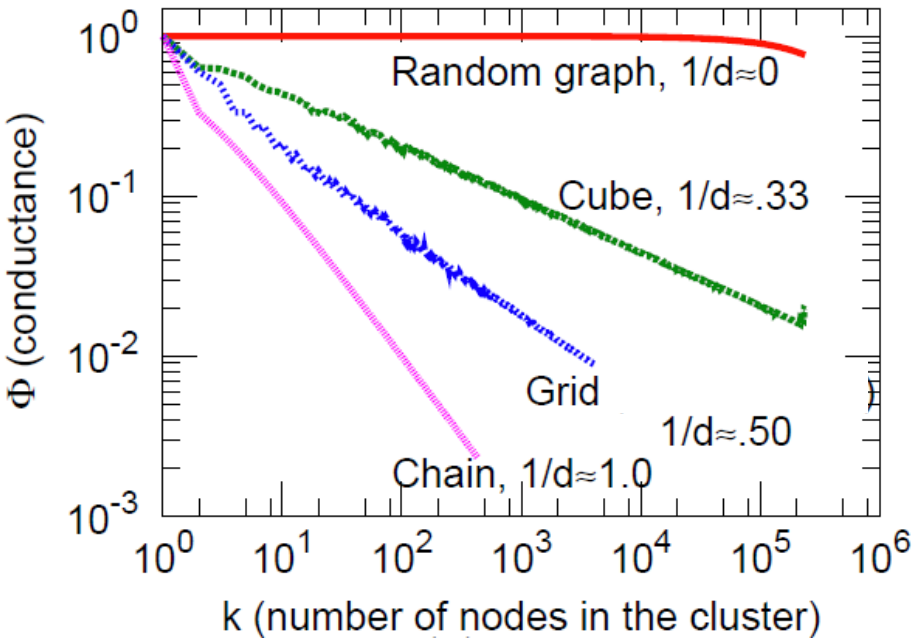
# How to (Really) Compute NCP?



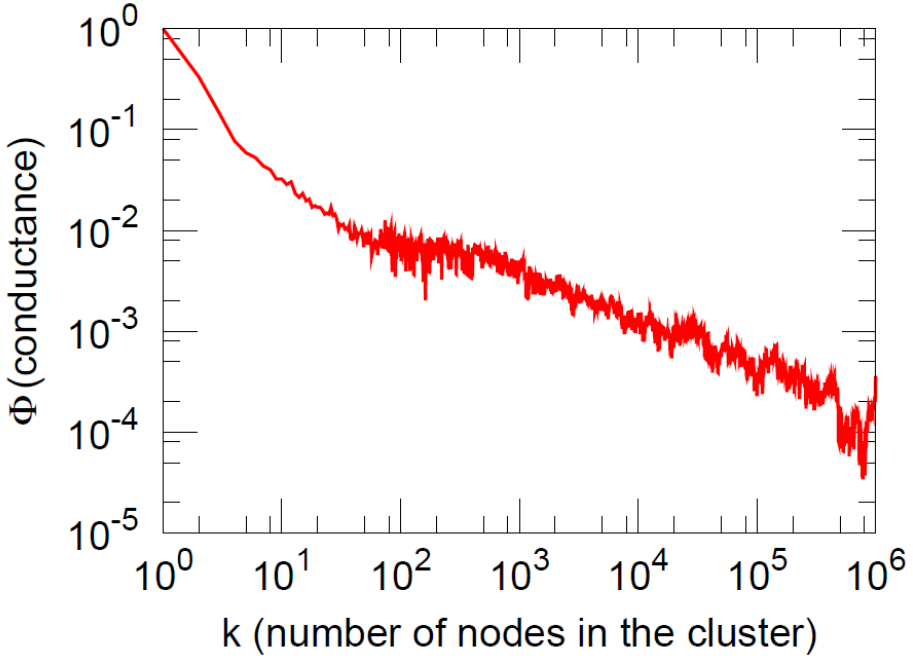

- Run the favorite clustering method(s)
- Each dot represents a cluster
- For each size $k$ find "best" cluster (min $\Phi(k)$)

# NCP Plot: Meshes

- **Meshes, grids, dense random graphs:**




d-dimensional meshes

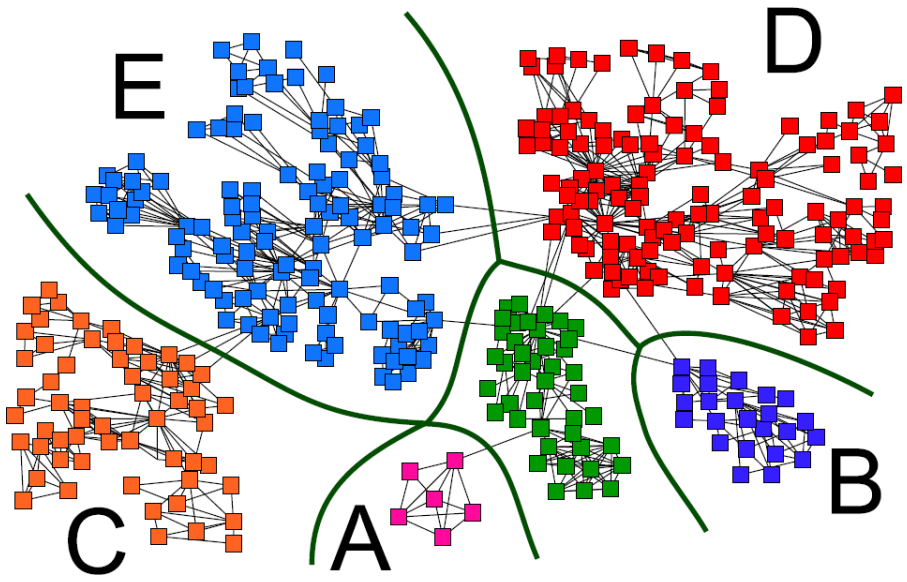


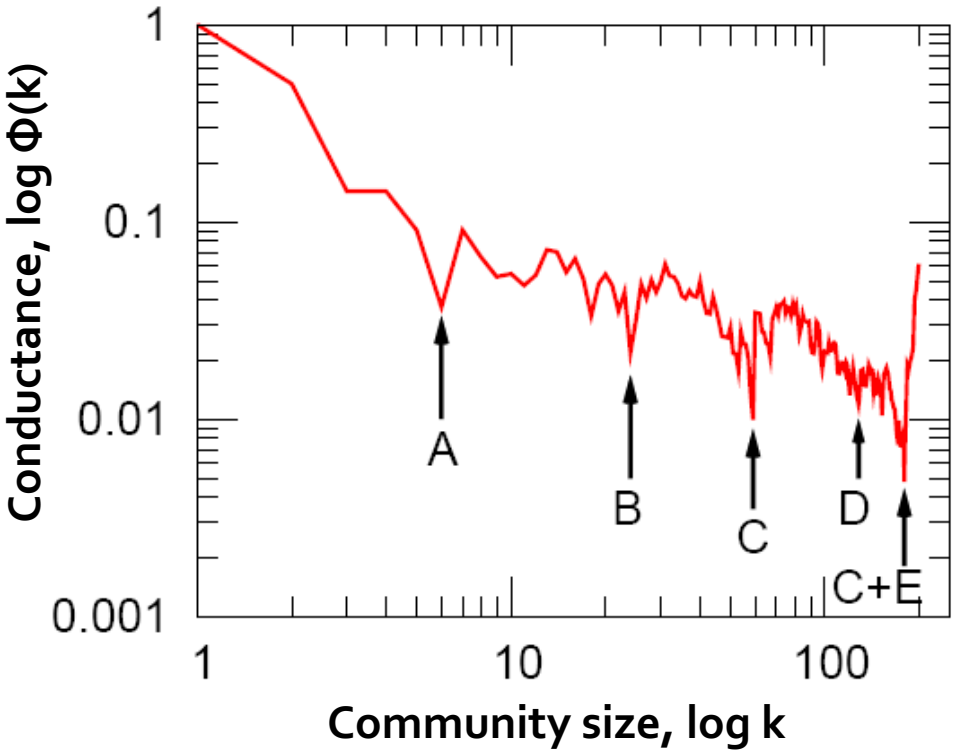

California road network

[WWW '08]

# NCP plot: Network Science

- **Collaborations between scientists in networks**
  [Newman, 2005]



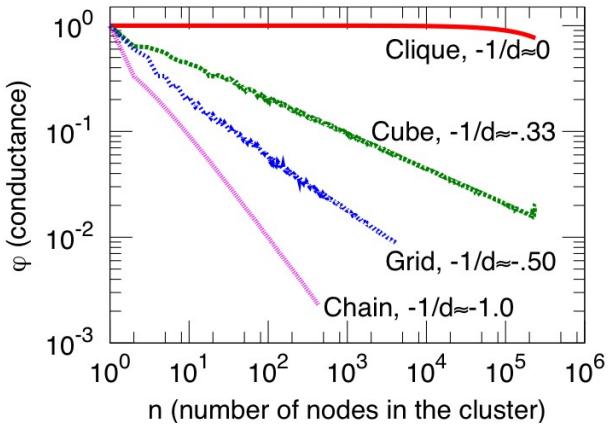

**Dips in the conductance graph correspond to the "good" clusters we can visually detect**

[Internet Mathematics '09]

# Natural Hypothesis

**Natural hypothesis about NCP:**

- NCP of real networks slopes downward
- Slope of the NCP corresponds to the "dimensionality" of the network



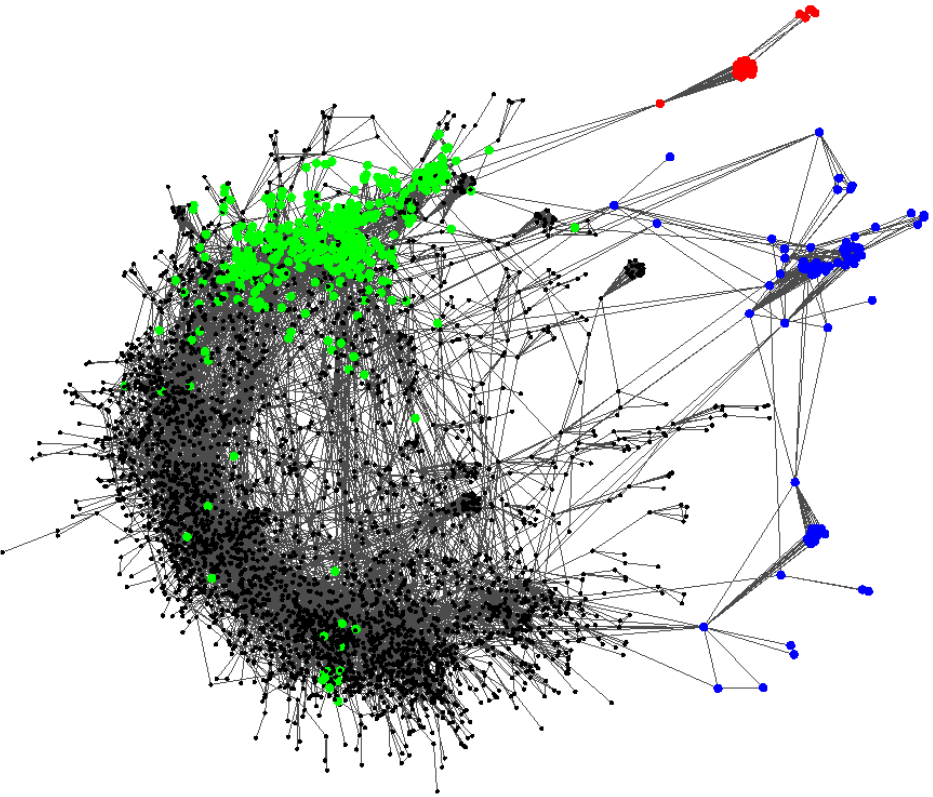

**What about large networks?**

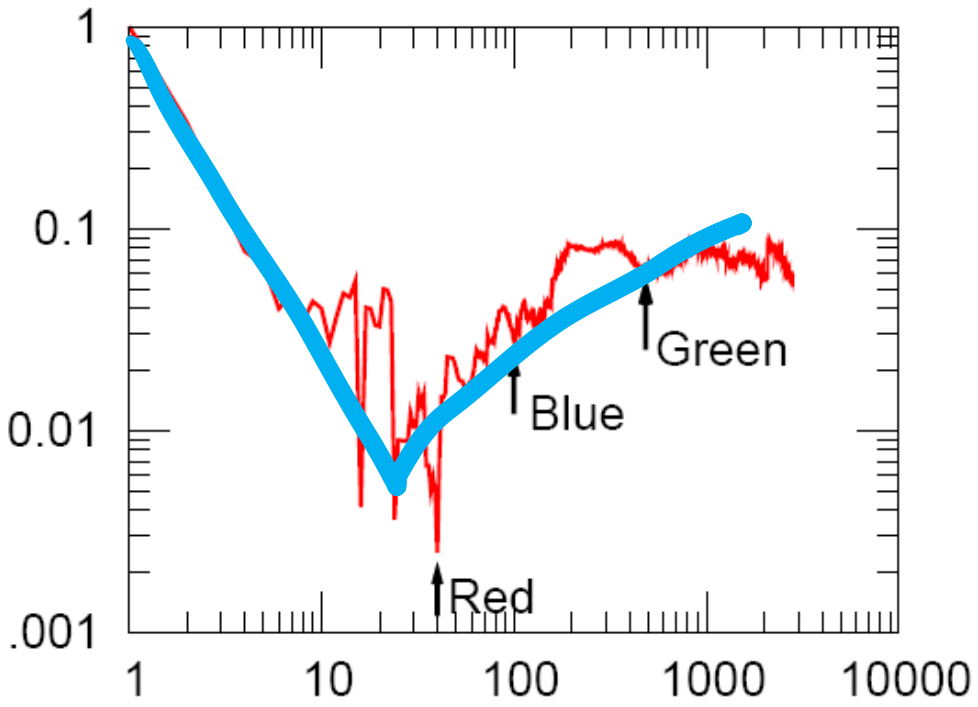
| • Social nets | Nodes | Edges | Description |
|---|---|---|---|
| LIVEJOURNAL | 4,843,953 | 42,845,684 | Blog friendships [5] |
| EPINIONS | 75,877 | 405,739 | Trust network [28] |
| CA-DBLP | 317,080 | 1,049,866 | Co-authorship [5] |
| • Information (citation) networks | | | |
| CIT-HEP-TH | 27,400 | 352,021 | Arxiv hep-th [14] |
| AMAZONPROD | 524,371 | 1,491,793 | Amazon products [8] |
| • Web graphs | | | |
| WEB-GOOGLE | 855,802 | 4,291,352 | Google web graph |
| WEB-WT10G | 1,458,316 | 6,225,033 | TREC WT10G |
| • Bipartite affiliation (authors-to-papers) networks | | | |
| ATP-DBLP | 615,678 | 944,456 | DBLP [21] |
| ATM-IMDB | 2,076,978 | 5,847,693 | Actors-to-movies |
| • Internet networks | | | |
| ASSKITTER | 1,719,037 | 12,814,089 | Autonom. sys. |
| GNUTELLA | 62,561 | 147,878 | P2P network [29] |

[Internet Mathematics '09]

# Large Networks: Very Different

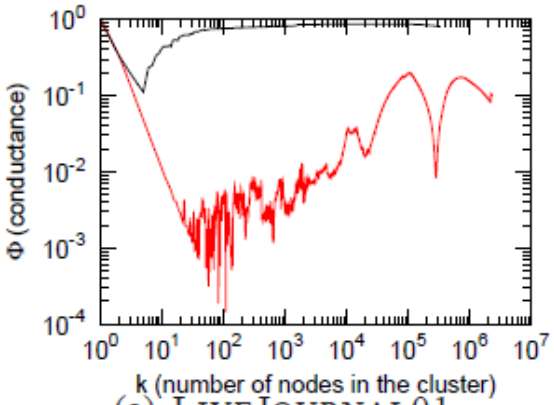
**Typical example:** General Relativity collaborations (n=4,158, m=13,422)



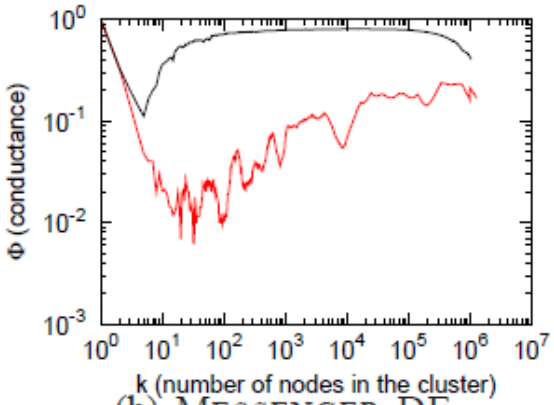


[Internet Mathematics '09]

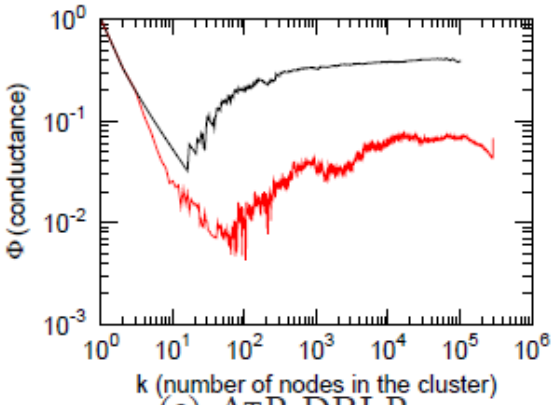
# More NCP Plots of Networks



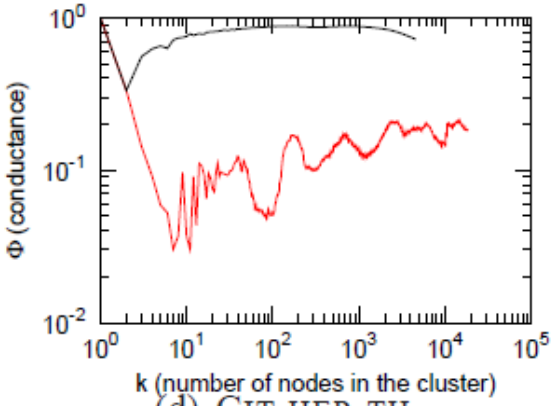
(a) LIVEJOURNAL01



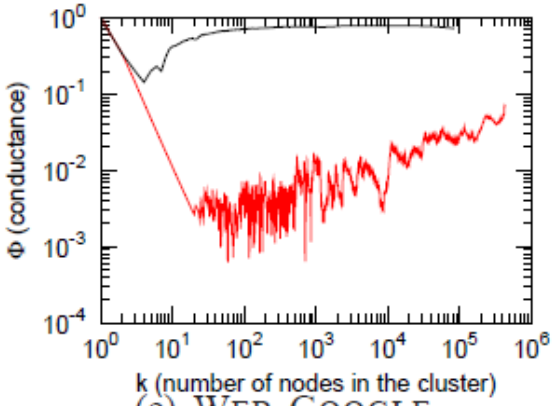
(b) MESSENGER-DE



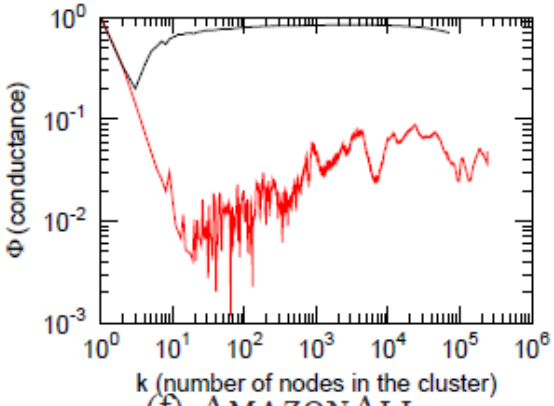
(c) ATP-DBLP



(d) CIT-HEP-TH



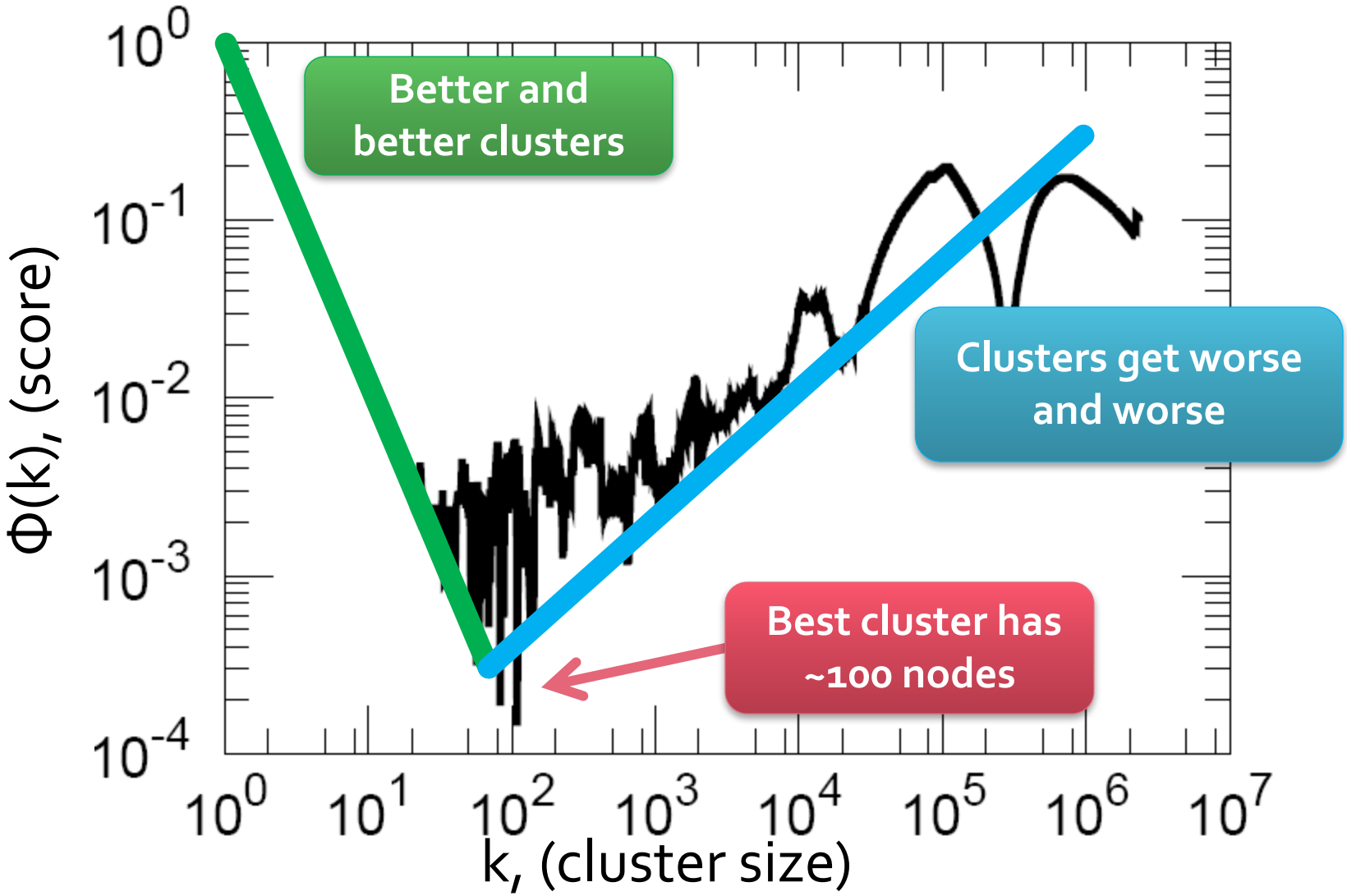
(e) WEB-GOOGLE



(f) AMAZONALL

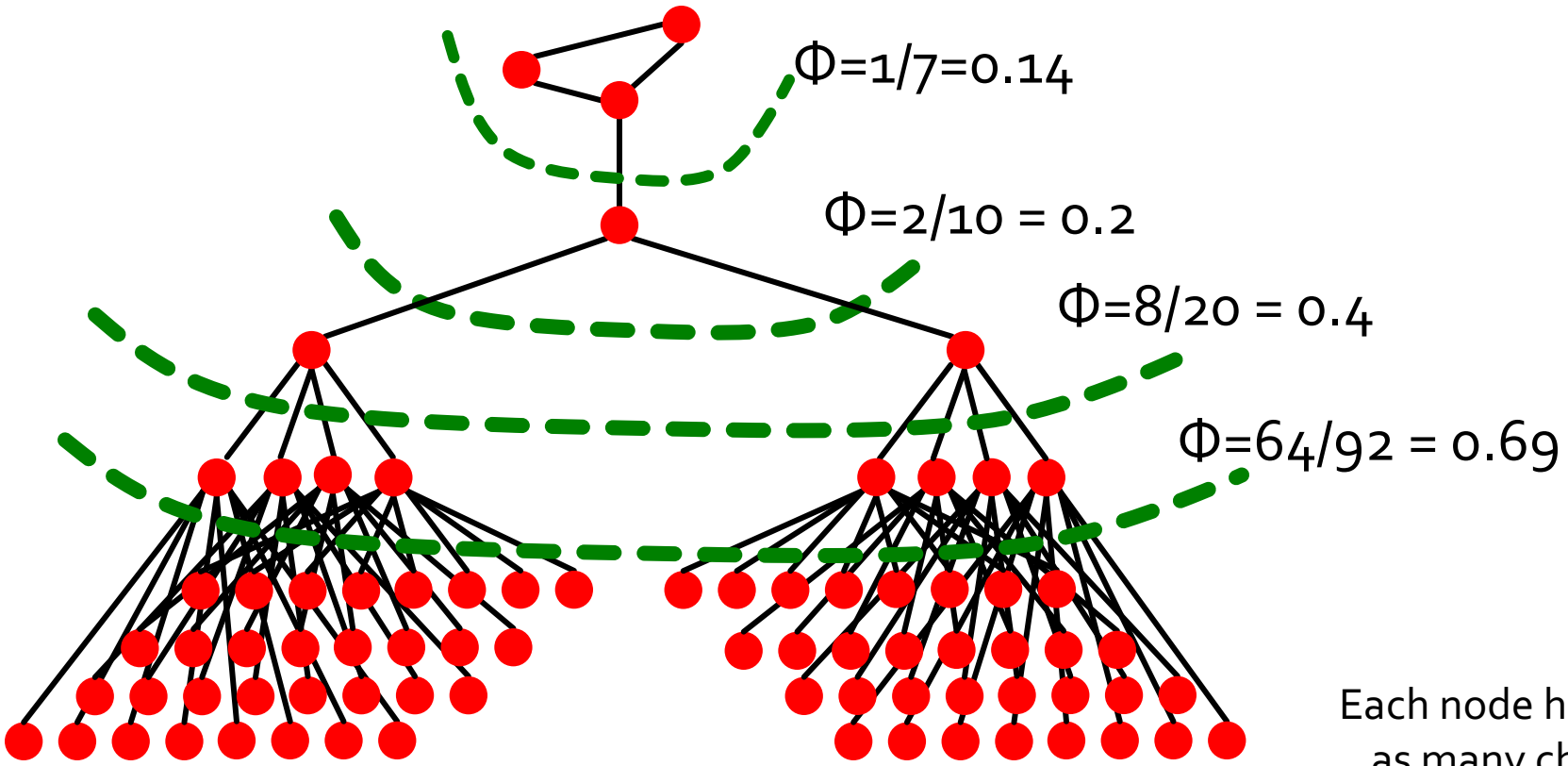
-- Rewired graph
-- Real graph

# NCP: LiveJournal (n=5m, m=42m)



# Explanation: The Upward Part

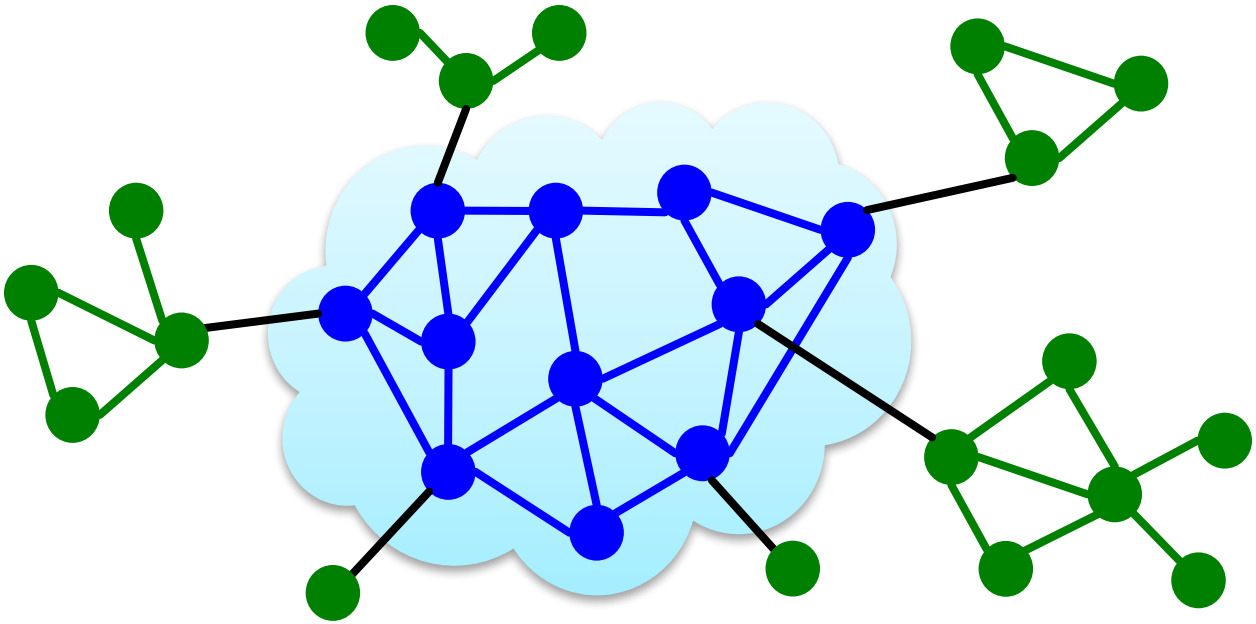
- As clusters grow the number of edges inside grows **slower** that the number crossing

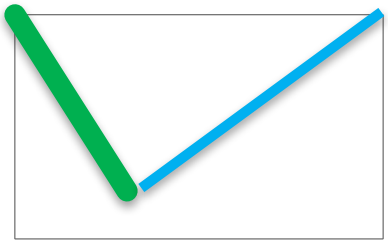



Each node has twice as many children

# Explanation: Downward Part

- Empirically we note that **best clusters** (corresponding to **green nodes "whiskers"**) are **barely connected** to the network

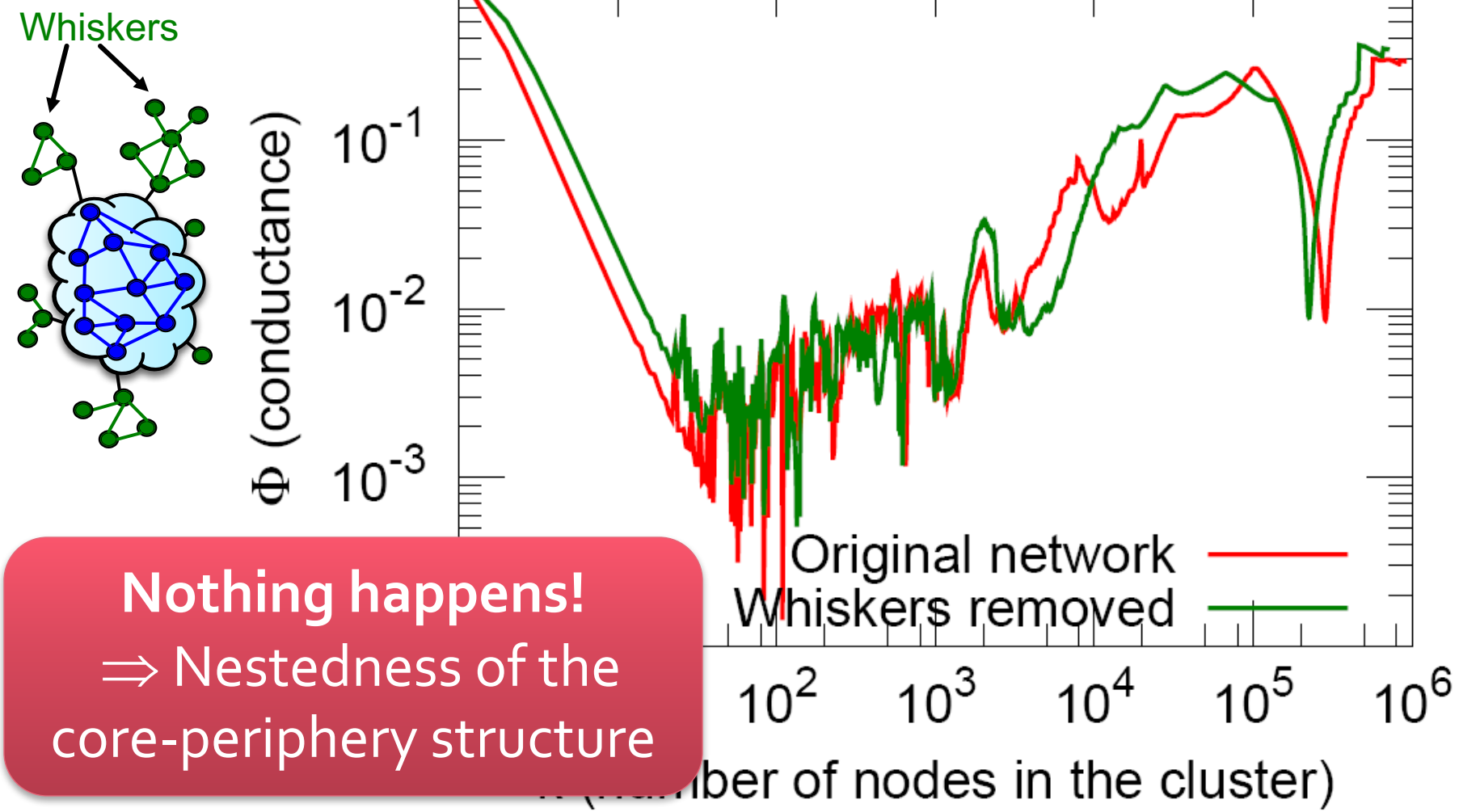




NCP plot

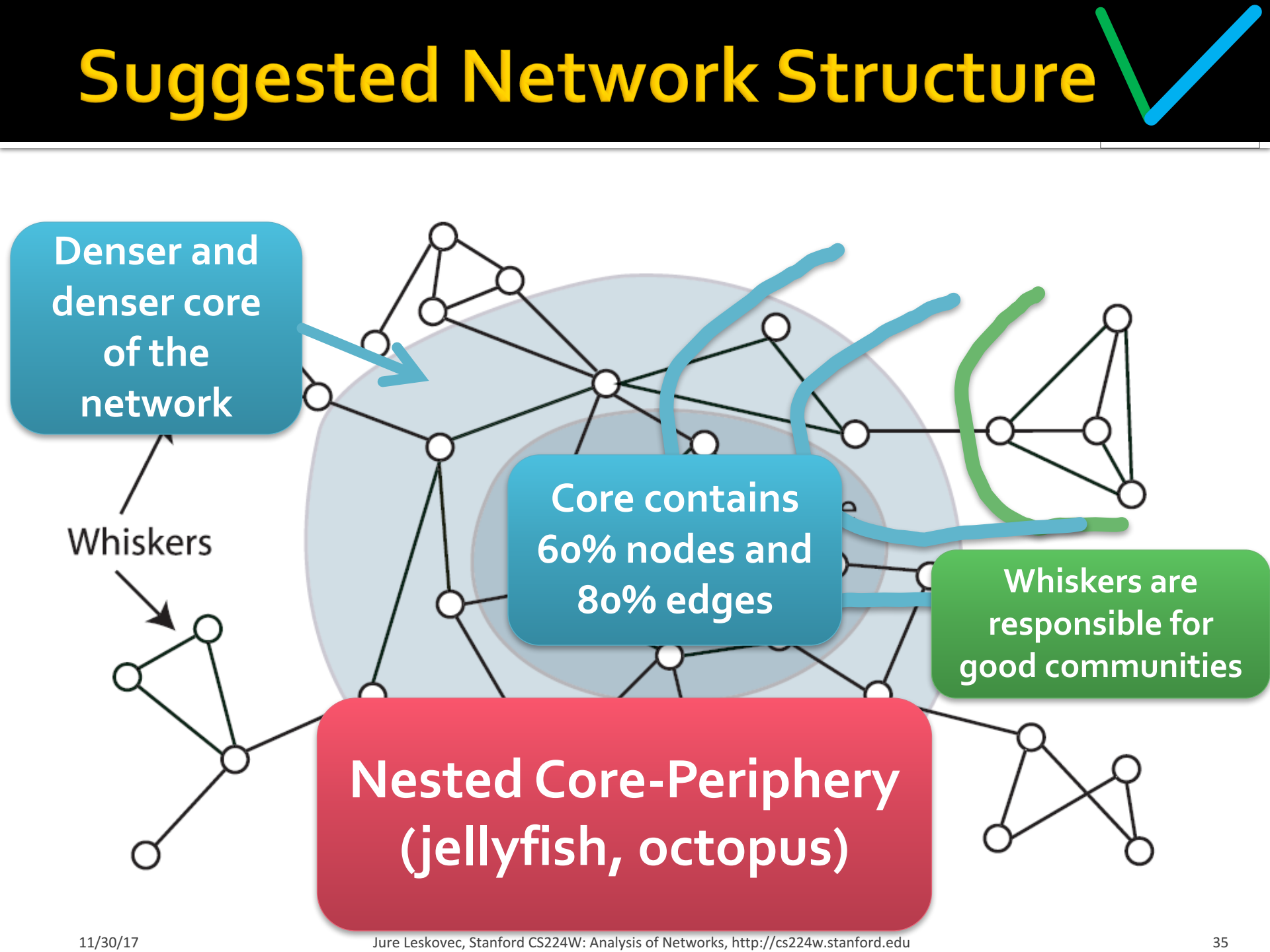
**⇒ Core-periphery structure**

# What If We Remove Good Clusters?

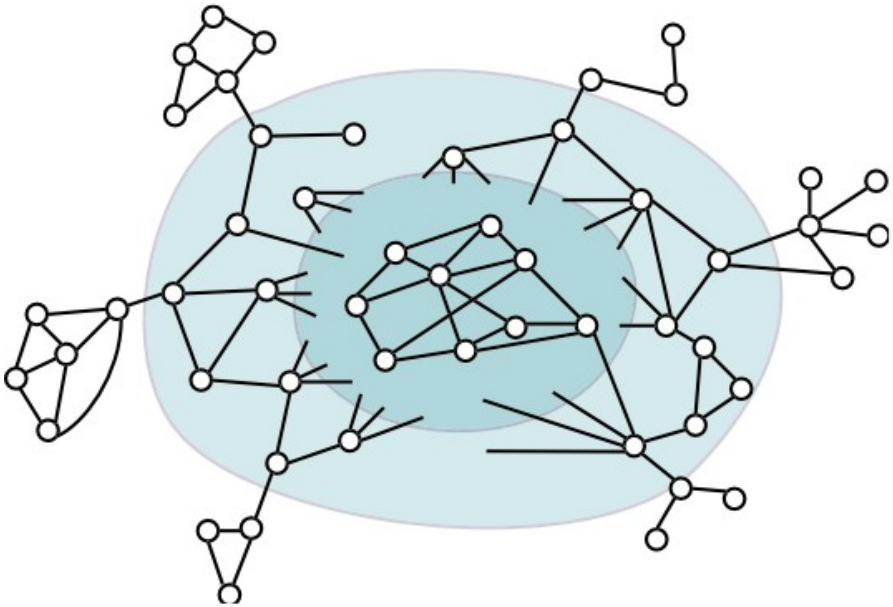



**Nothing happens!**
⇒ Nestedness of the core-periphery structure

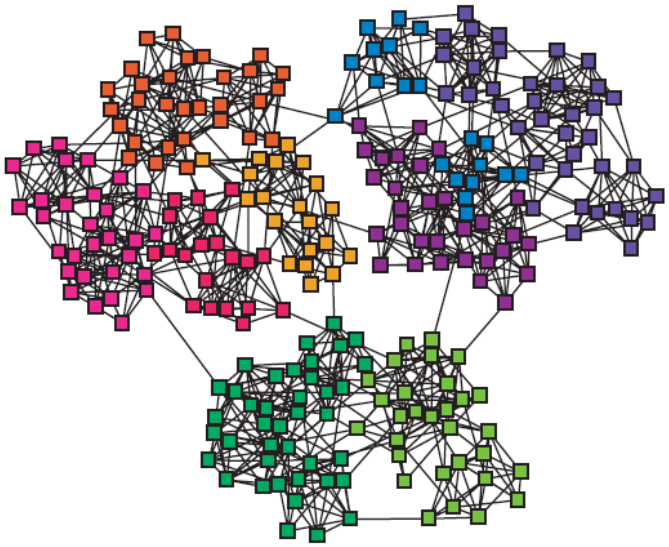
# Suggested Network Structure



# Part 2: Explanation



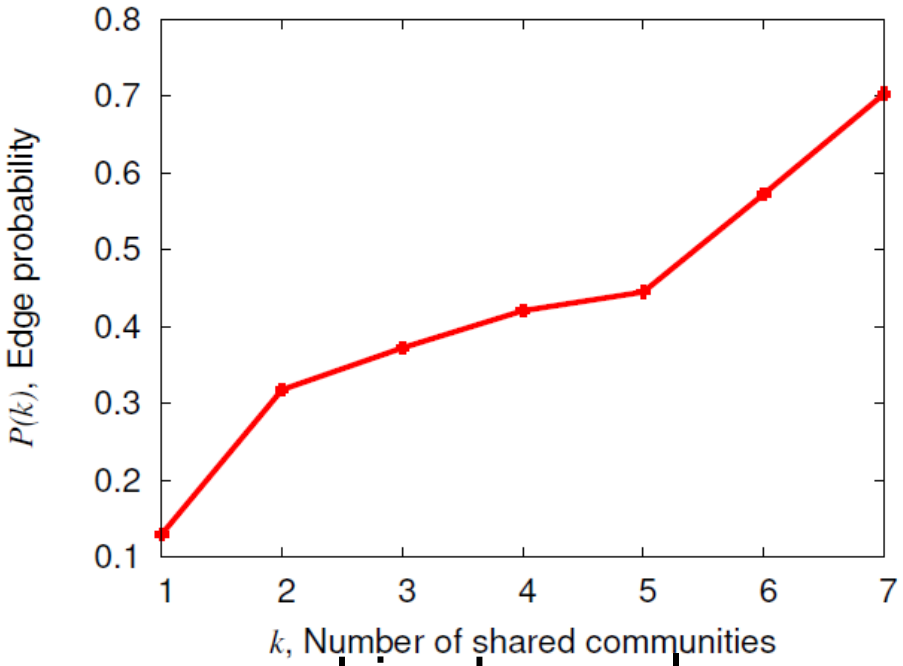
vs.



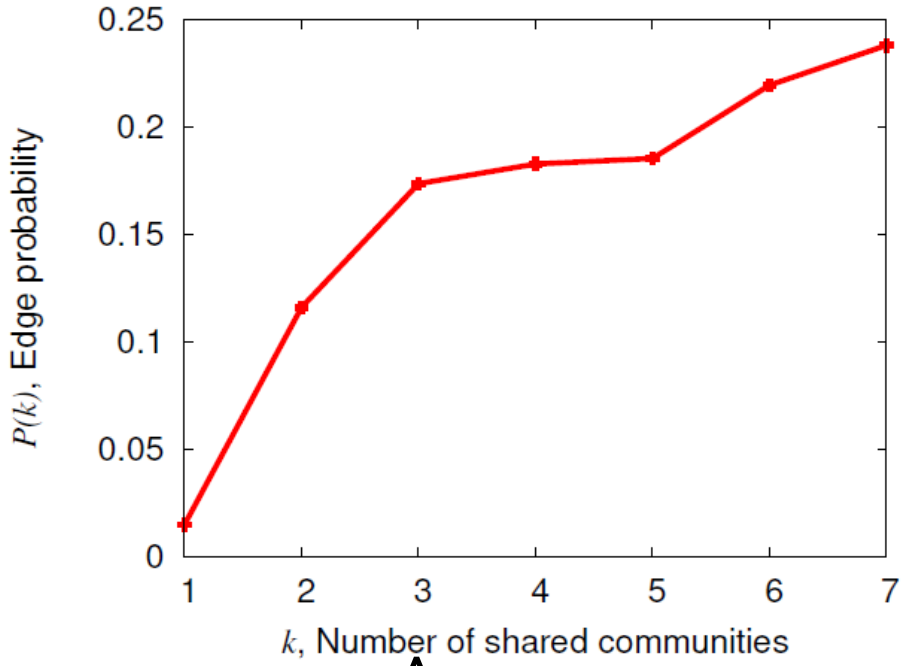
**How do we reconcile these two views?**

# Ground-truth Communities

- **Basic question: nodes $u$, $v$ share $k$ communities**
- **What's the edge probability?**
  - **Look at networks with ground-truth communities**



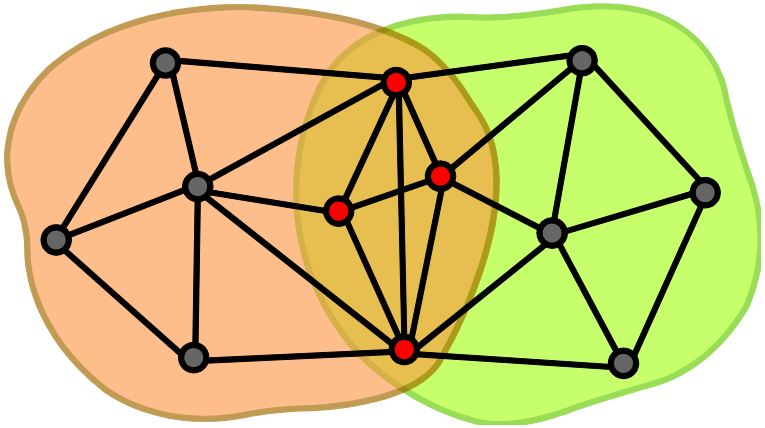

LiveJournal social network

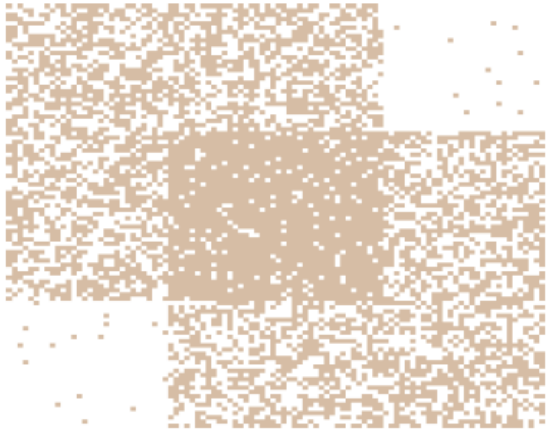



Amazon product network

# Communities as Tiles!

- **Edge density in the overlaps is higher!**

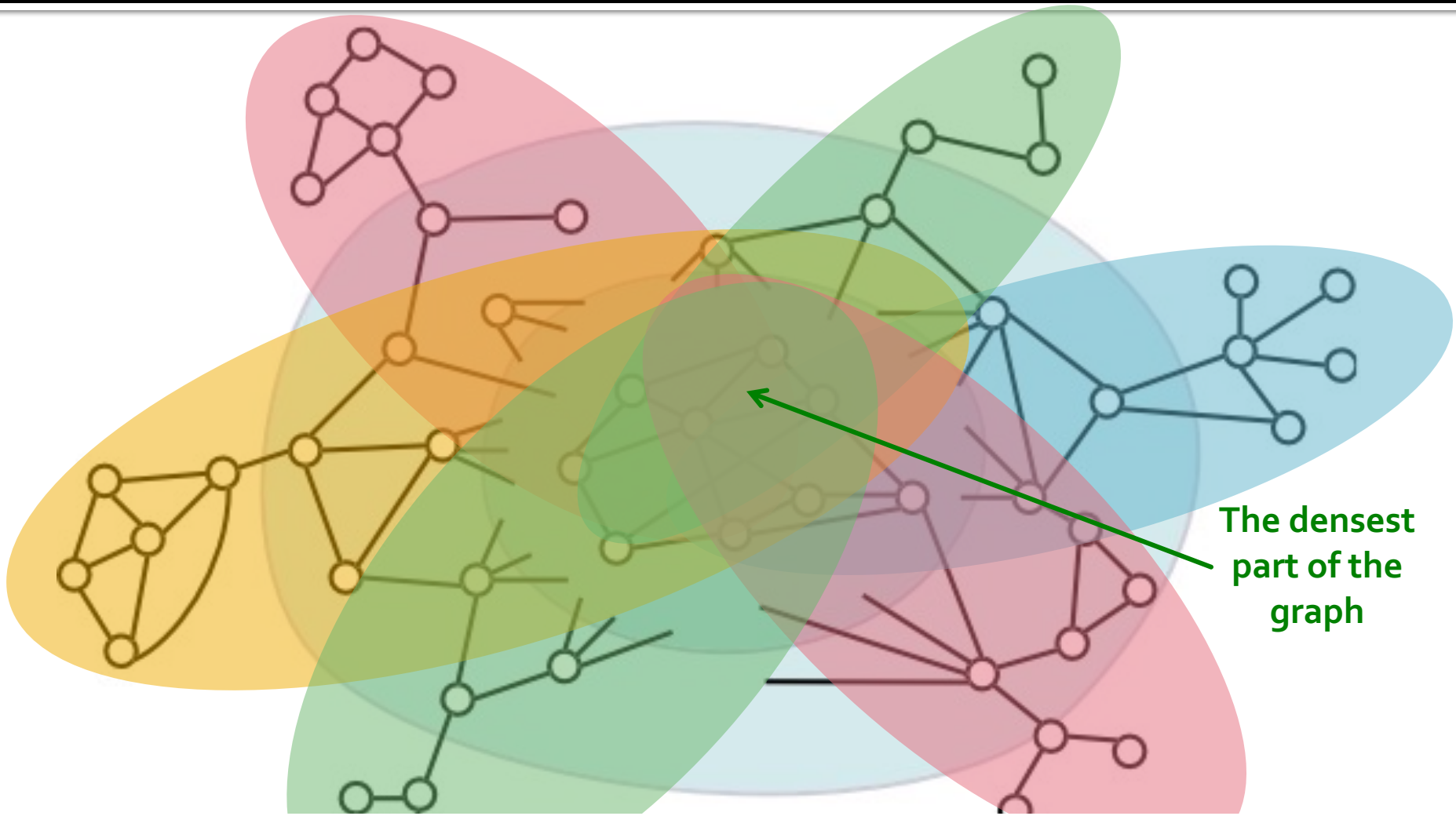




*"The more different foci (communities) that two individuals share, the more likely it is that they will be tied"* - S. Feld, 1981

**Communities as "tiles"**

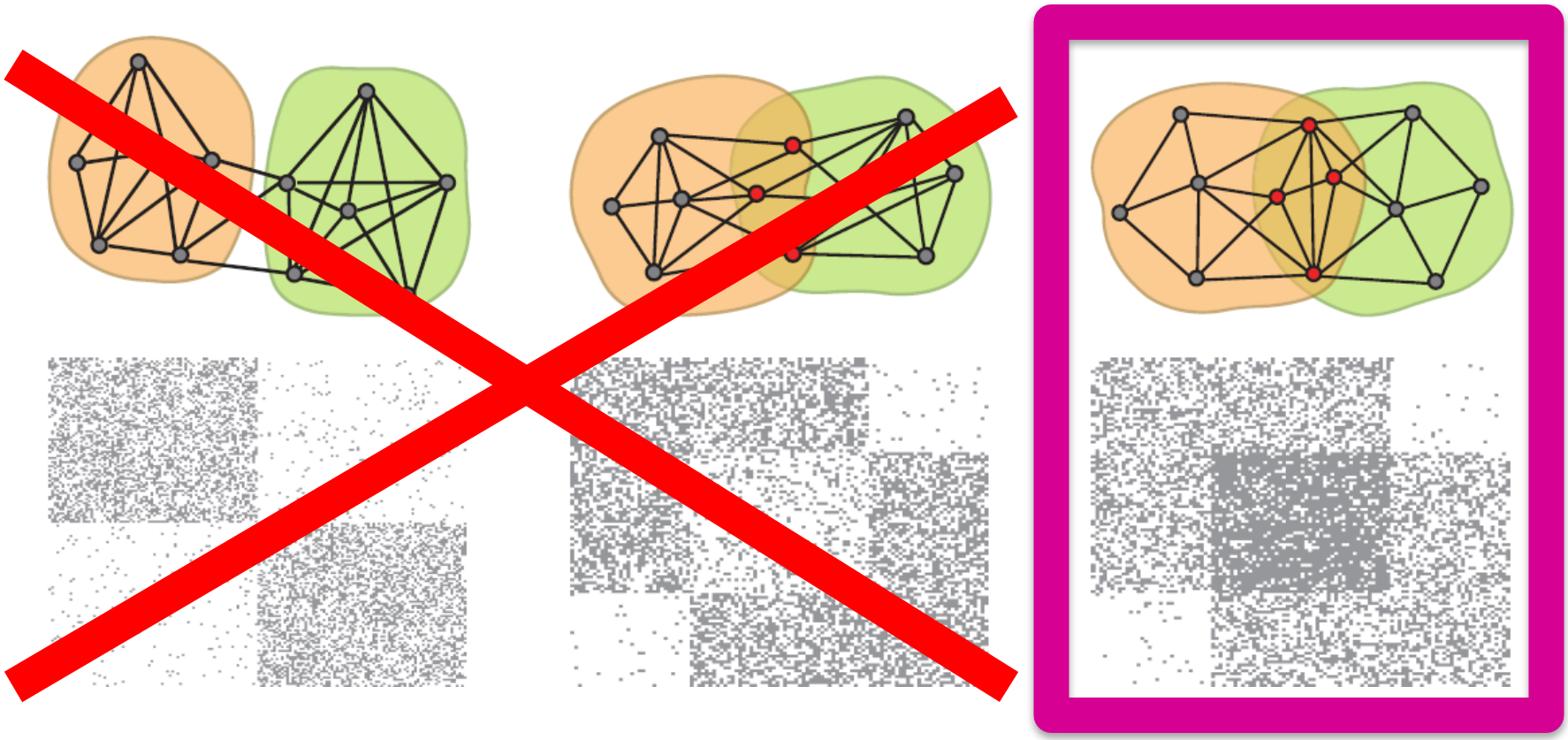
# Communities as Tiles/Circles



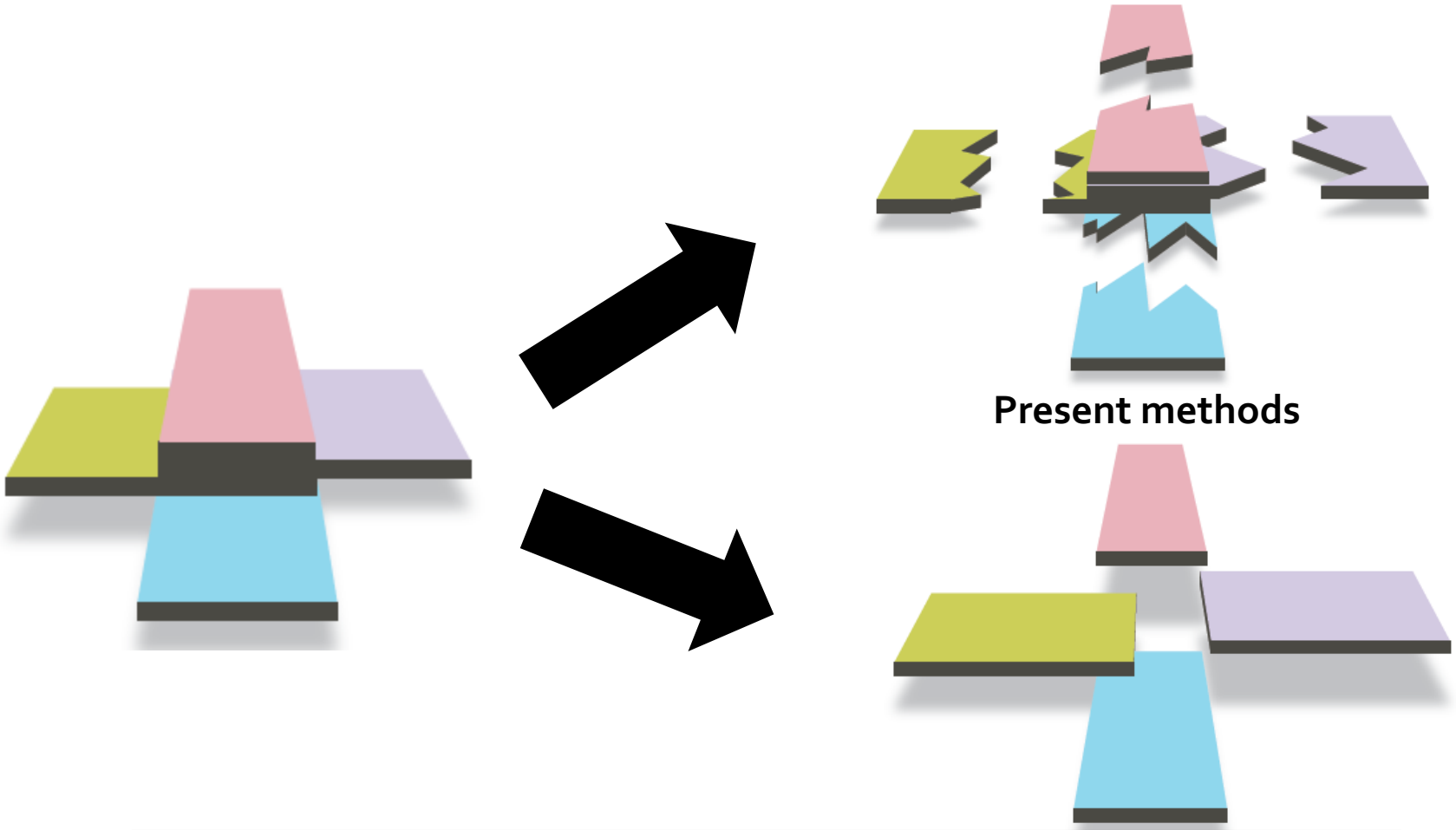

# Communities as overlapping tiles

# Communities in Networks

## What does this mean?

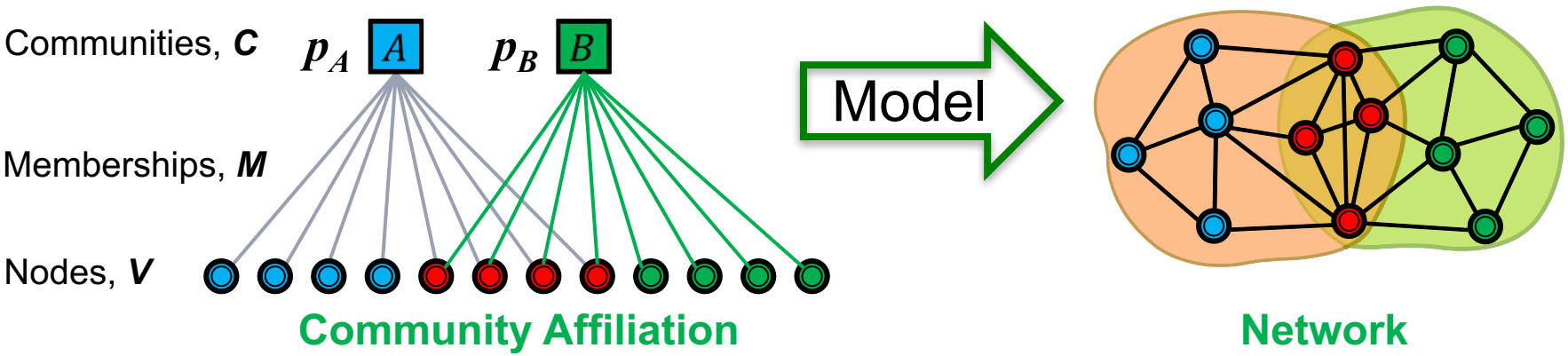


# From Networks to Communities



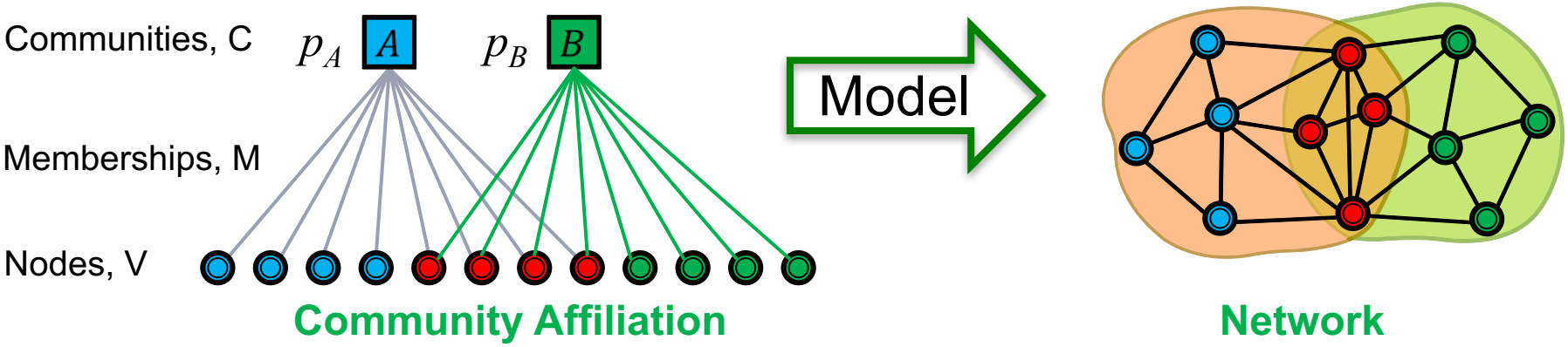

**How do we detect communities if they overlap as tiles?**

# Community-Affiliation Graph Model (AGM)




- **Generative model: How is a network generated from community affiliations?**
- **Model parameters:**
  - Nodes ***V***, Communities ***C***, Memberships ***M***
  - Each community $c$ has a single probability $p_c$

# AGM: Generative Process

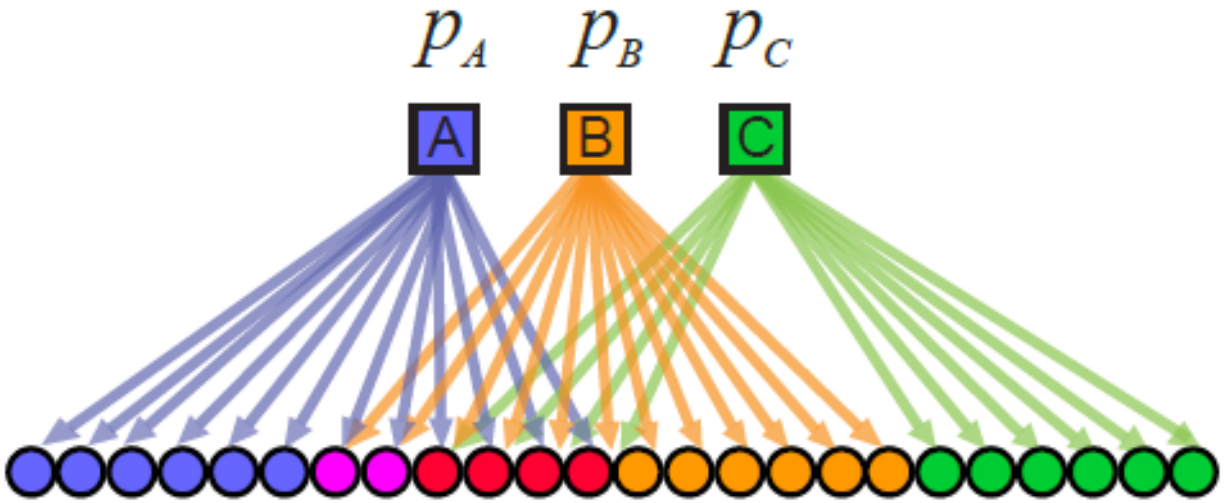



- **Given parameters ($V$, $C$, $M$, $\{p_c\}$)**
  - Nodes in community $c$ connect to each other by flipping a coin with probability $p_c$
  - **Nodes that belong to multiple communities have multiple coin flips: Dense community overlaps**
    - If they "miss" the first time, they get another chance through the next community"

$$p(u,v) = 1 - \prod_{c \in M_u \cap M_v} (1 - p_c)$$

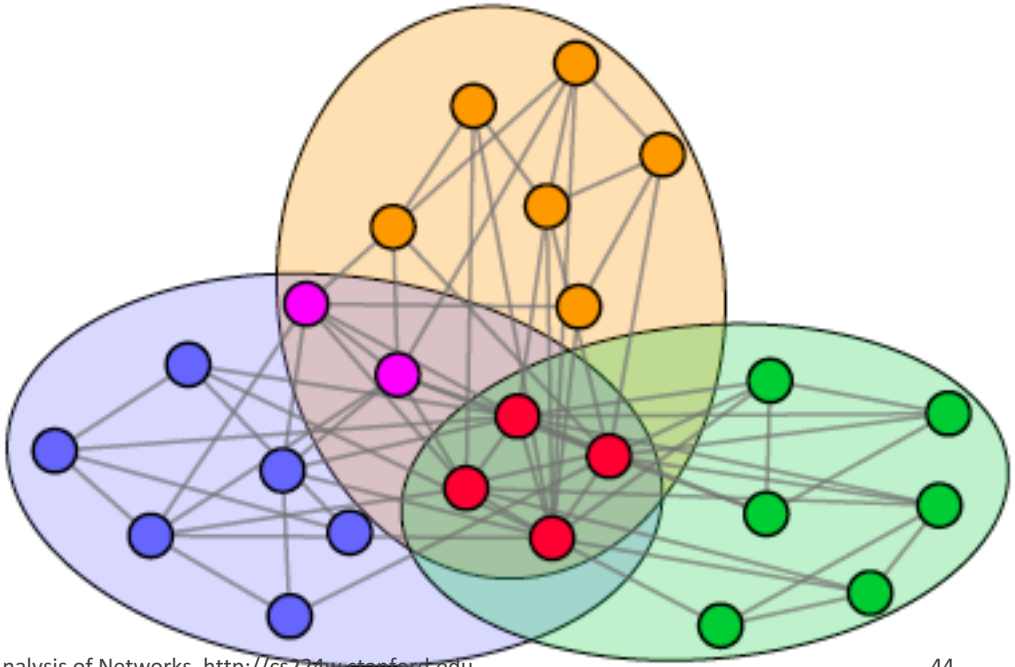
Note: If two nodes u and v have no communities in common, then p(u,v)=0. We resolve this by having a "background" community that every node is a member of.

# AGM: Dense Overlaps




Model

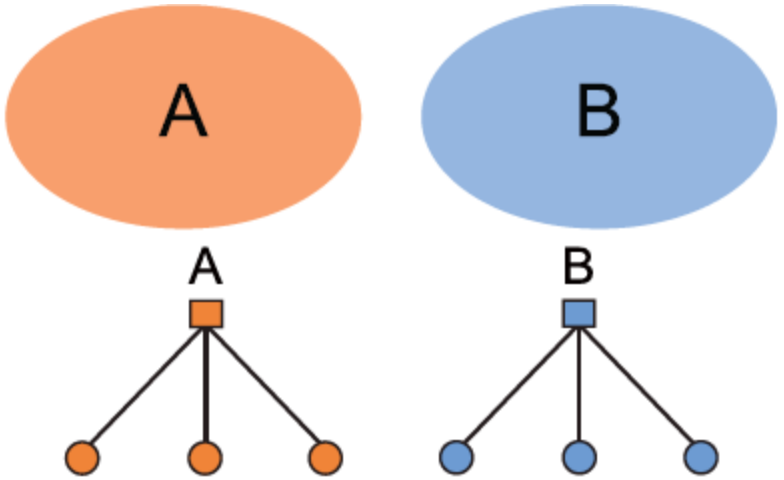
Network

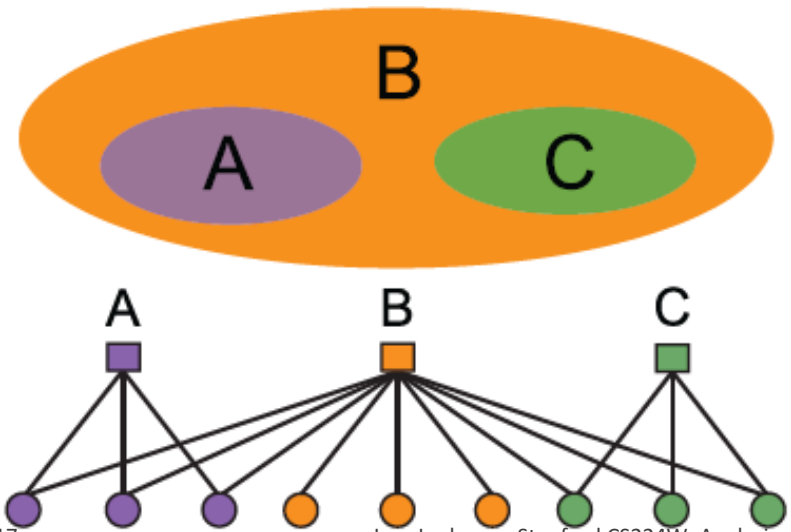


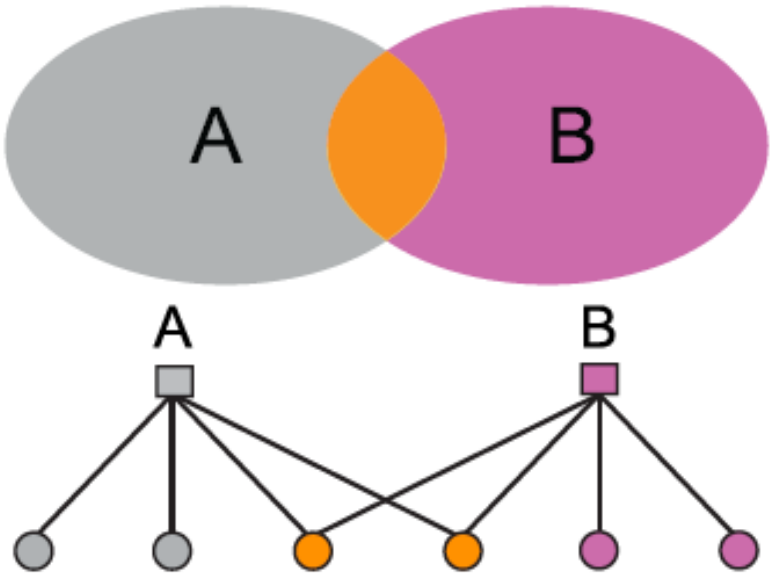
# AGM: Flexibility

[icdm '12]

- **AGM can express a variety of community structures:**
  Non-overlapping,
  Overlapping, Nested

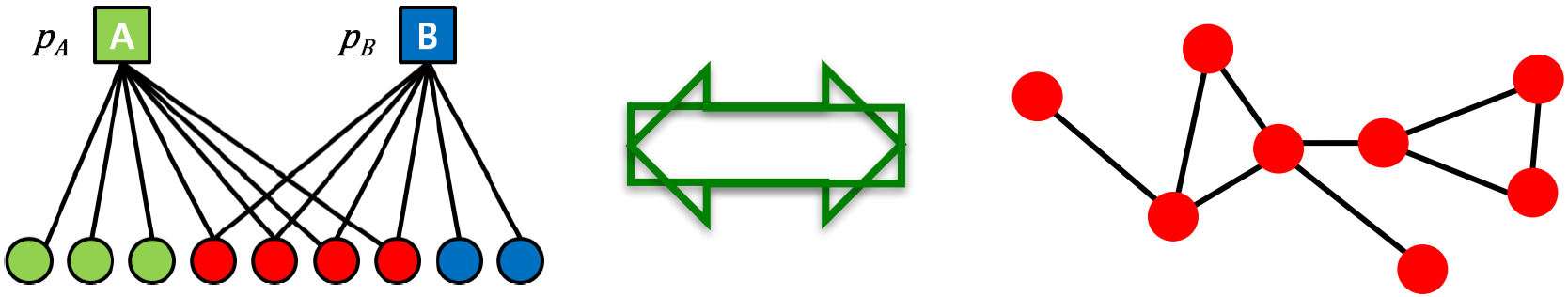


# Detecting Communities

- **Detecting communities with AGM:**



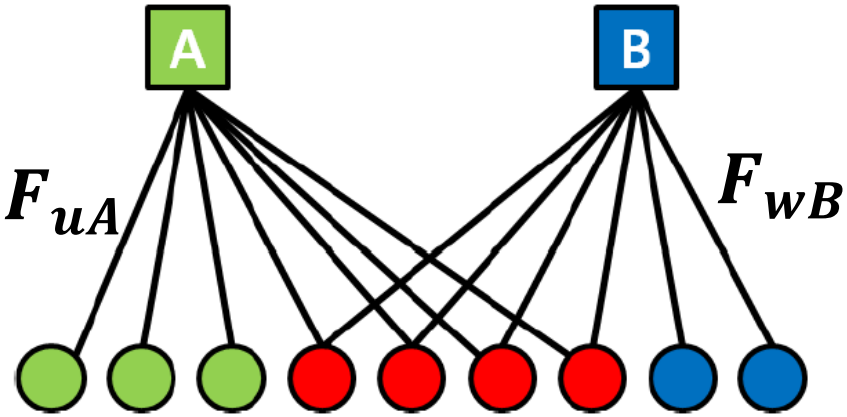

## Given a Graph, find the Model

1) Affiliation graph ***M***
2) Number of communities **C**
3) Parameters $\boldsymbol{p_c}$

# "Relaxing" AGM

[wsdm '13]

- **"Relax" the AGM: Memberships have strengths**




- $F_{uA}$: The membership strength of node $u$ to community $A$ ($F_{uA} = 0$: no membership)

# BigCLAM Model

[wsdm '13]

- **Prob. of nodes linking is proportional to the strengths of shared memberships:**

$$P(u,v) = 1 - \exp(-F_u \cdot F_v^T)$$

- **Now, given a network, we estimate $F$**

$$l(F) = \sum_{(u,v)\in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v)\notin E} F_u F_v^T$$

- **Non-negative matrix factorization:**
  - Update $F_{uC}$ for node $u$ while fixing the memberships of all other nodes
  - Updating takes linear time in the degree of $u$

# BigCLAM Model

[wsdm '13]

- **Apply block coordinate gradient ascent**

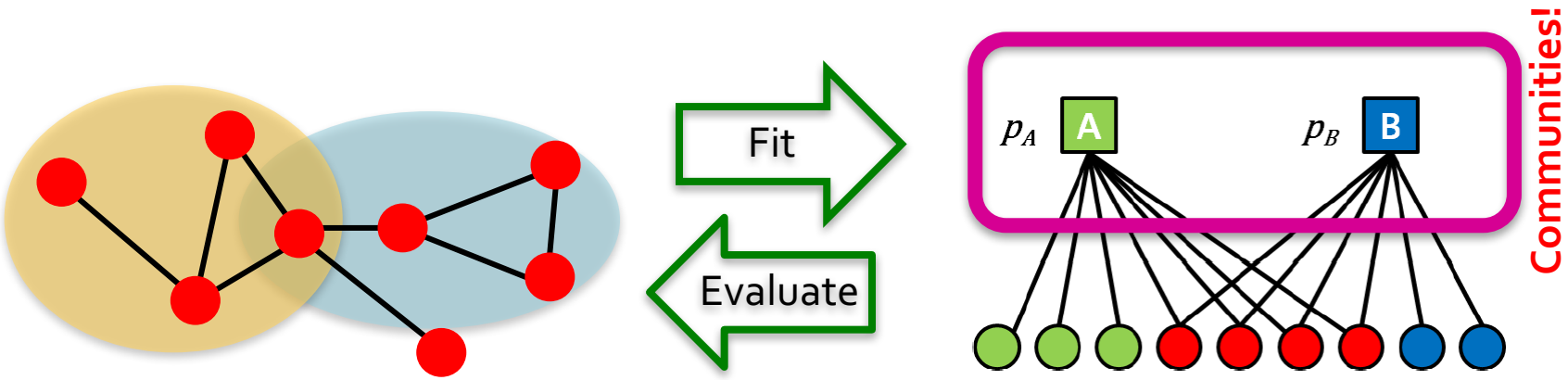
$$\nabla l(F_u) = \sum_{v \in \mathcal{N}(u)} F_v \frac{\exp(-F_u F_v^T)}{1 - \exp(-F_u F_v^T)} - \sum_{v \notin \mathcal{N}(u)} F_v$$

  - Step size: backtracking line search
  - Project $\boldsymbol{F_u}$ back to a non-negative vector
- Pure gradient ascent is slow! **However:**

$$\sum_{v \notin \mathcal{N}(u)} F_v = (\sum_{v} F_v - F_u - \sum_{v \in \mathcal{N}(u)} F_v)$$

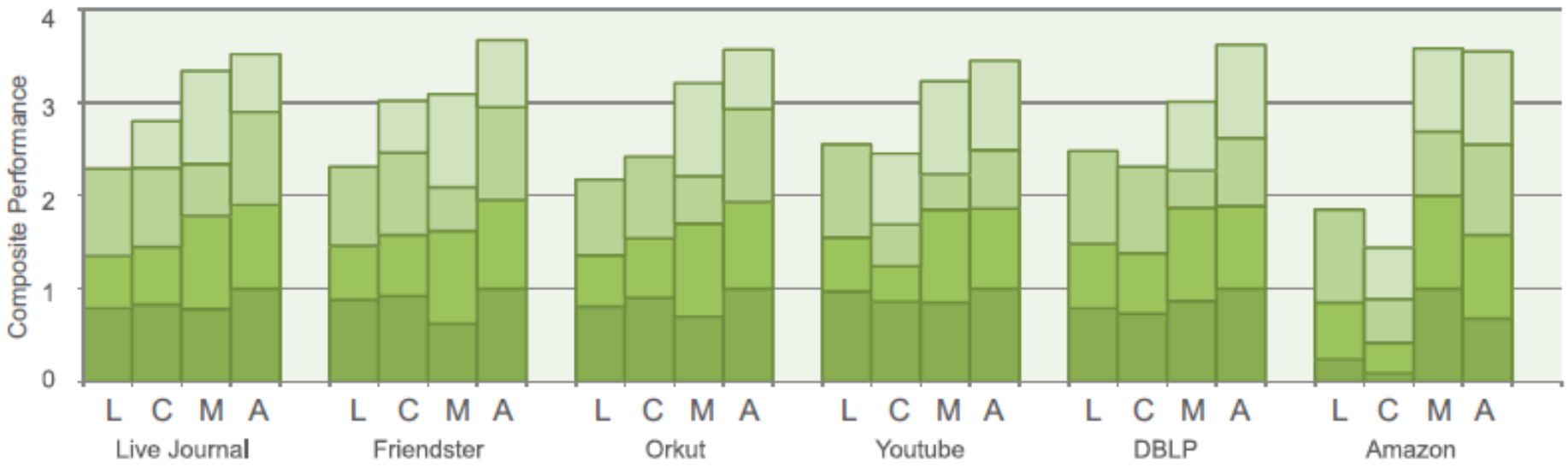
  - By caching $\boldsymbol{F_v}$ the gradient step takes **linear time** in the degree of $\boldsymbol{u}$

# Experimental Setup




- **How well do inferred communities correspond to ground-truth?**
  - **F1 score, Ω-index, Mutual Information**
- We can rank algorithms based on their ability to detect ground-truth

# Experiments: Ground-truth



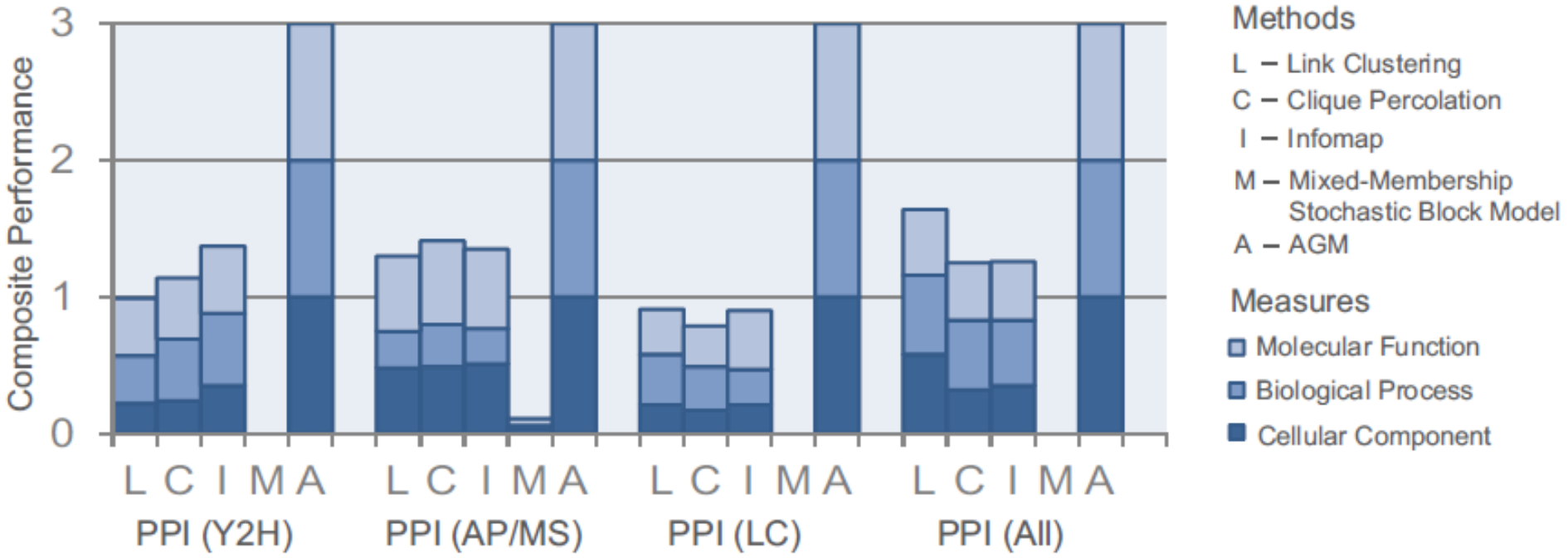

- **BigClam improves:**
  - 79% over Link clustering
  - 48% over CPM
  - 15% over MMSB

  (while being orders of magnitude faster)

Methods
L – Link Clustering
C – Clique Percolation
M – Mixed-Membership Stochastic Block Model
A – AGM

Measures
Number of Communities
Normalized Mutual Information
F1-score
Ω-index

# Experiments: PPI Nets




- **Protein-Protein interaction networks:**
  Gene Ontology based quality of detected communities

# Communities:
# Issues and Questions

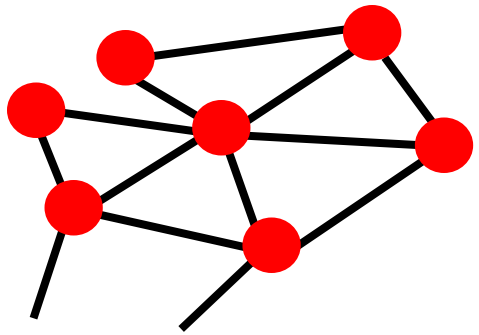
# Communities: Issues and Questions

- **Some issues with community detection:**
  - Many different formalizations of clustering objective functions
  - Objectives are NP-hard to optimize exactly
  - Methods can find clusters that are systematically "biased"
- **Questions:**
  - **How well do algorithms optimize objectives?**
  - **What clusters do different methods find?**

[WWW '09]

# Many Different Objective Functions

- **Single-criterion:**
  - Modularity: $m\text{-}E(m)$
  - Edges cut: $c$
- **Multi-criterion:**
  - Conductance: $c/(2m+c)$
  - Expansion: $c/n$
  - Density: $1\text{-}m/n^2$
  - CutRatio: $c/n(N\text{-}n)$
  - Normalized Cut: $c/(2m+c) + c/2(M\text{-}m)+c$
  - Flake-ODF: *frac. of nodes with more than ½ edges pointing outside S*




$n$: nodes in S
$m$: edges in S
$c$: edges pointing outside S

[WWW '09]

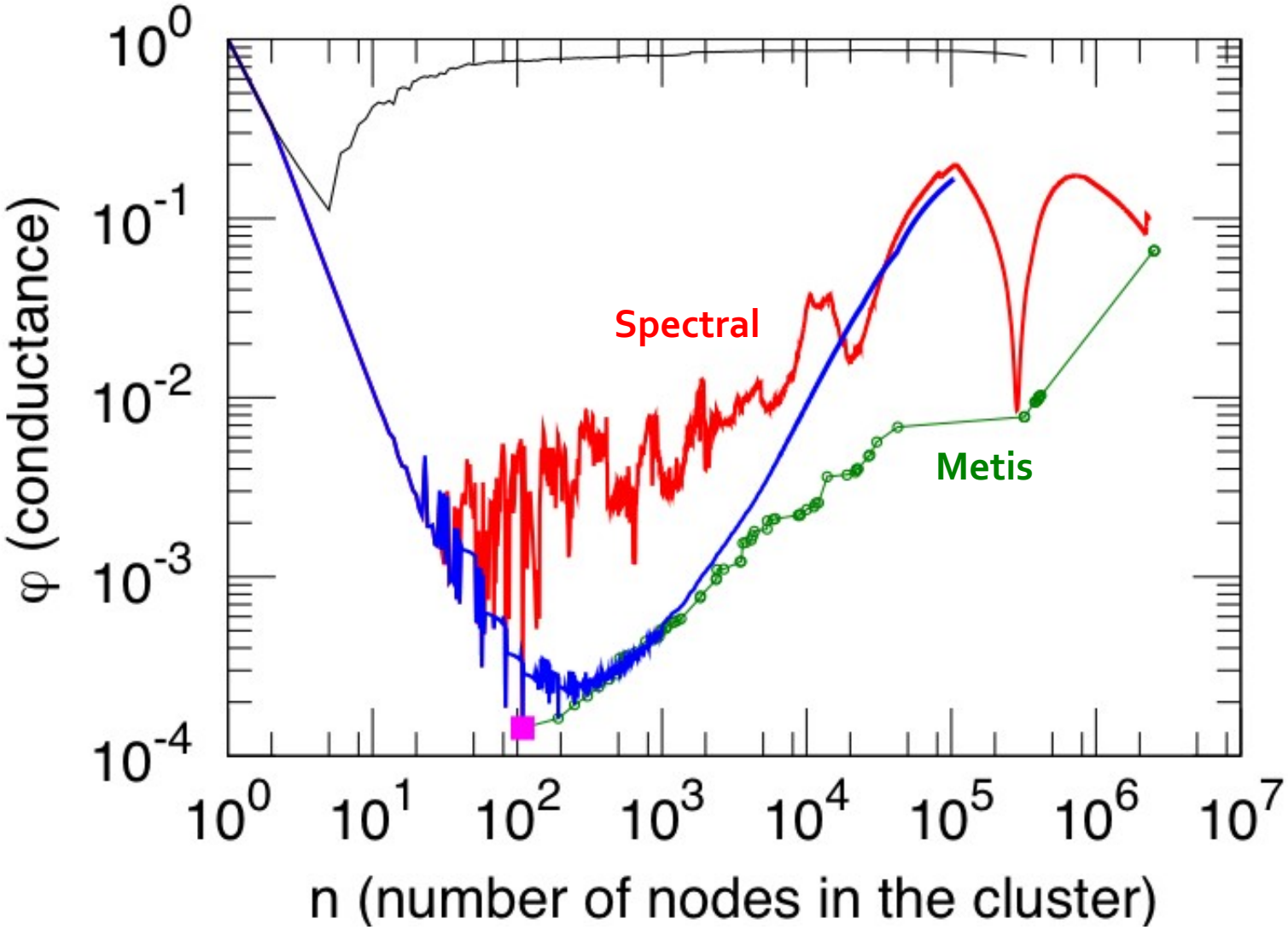
# Many Classes of Algorithms

**Many algorithms that implicitly or explicitly optimize objectives and extract communities:**

- **Heuristics:**
  - Girvan-Newman, Modularity optimization: popular heuristics
  - Metis: multi-resolution heuristic [Karypis-Kumar '98]
- **Theoretical approximation algorithms:**
  - Spectral partitioning

[WWW '09]

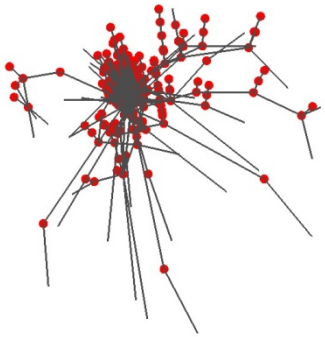
# NCP: Live Journal

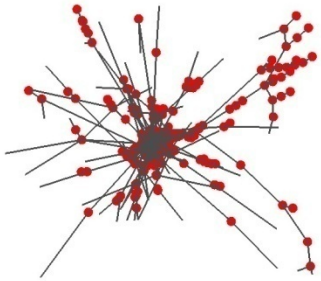


[WWW '09]

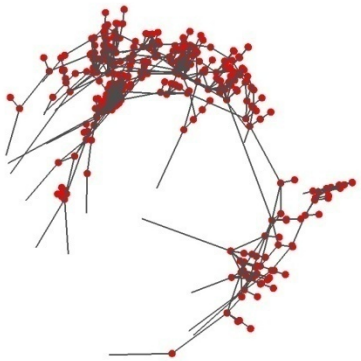
# Properties of Clusters (1)

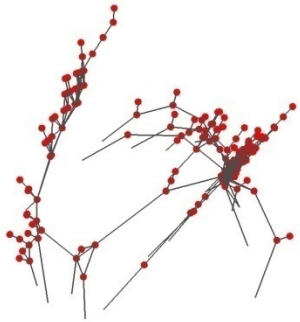
**500 node communities from Spectral:**





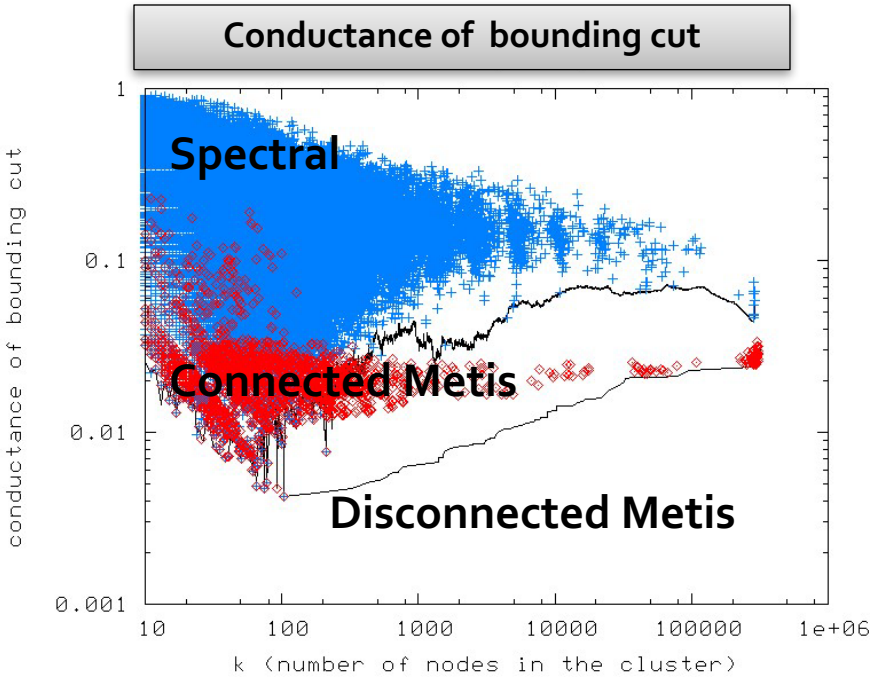
**500 node communities from Metis:**

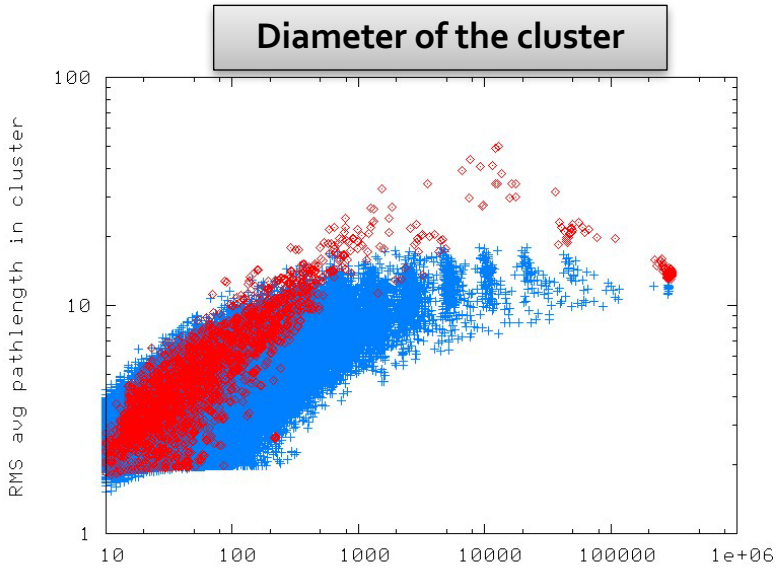


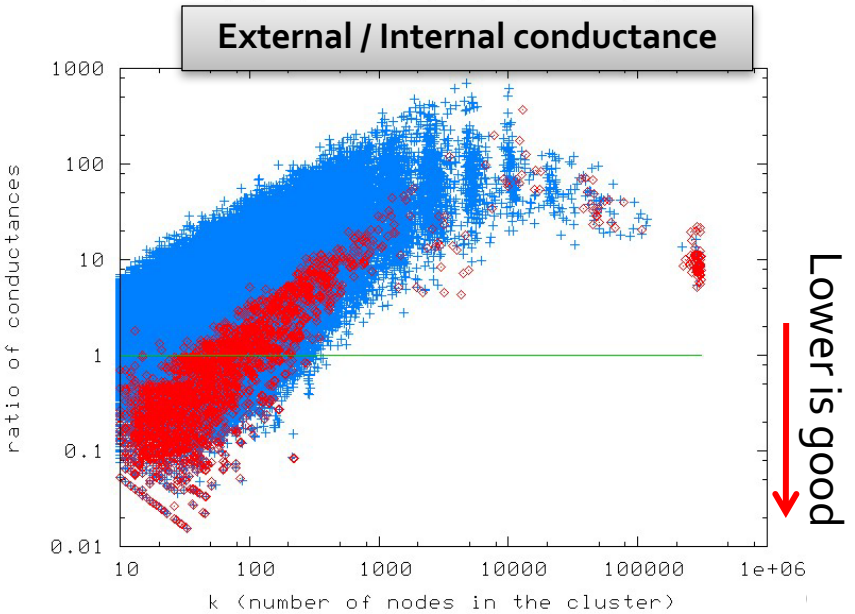


[WWW '09]

# Properties of Clusters (2)



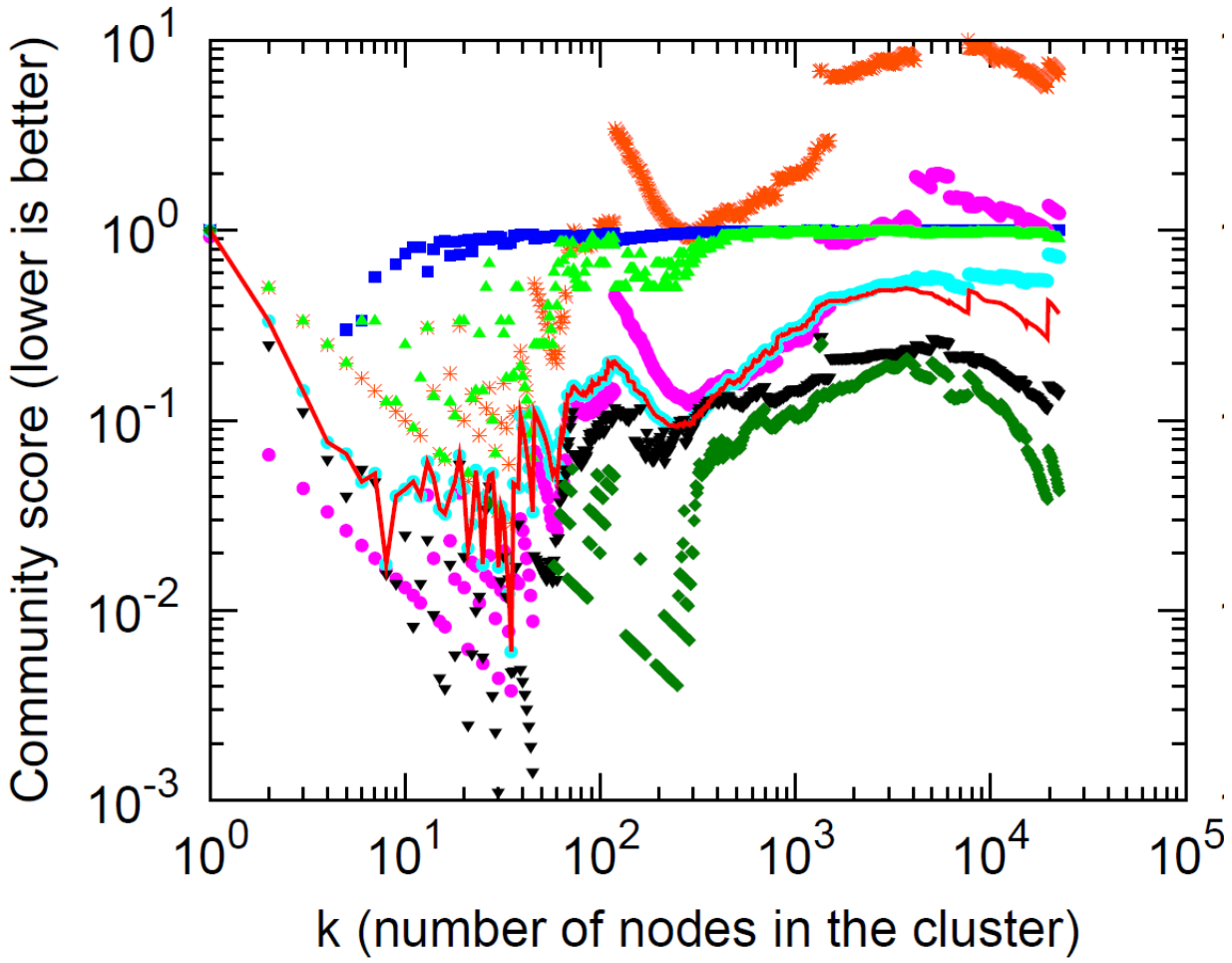




- Metis gives sets with better conductance
- Spectral gives tighter and more well-rounded sets

[WWW '09]

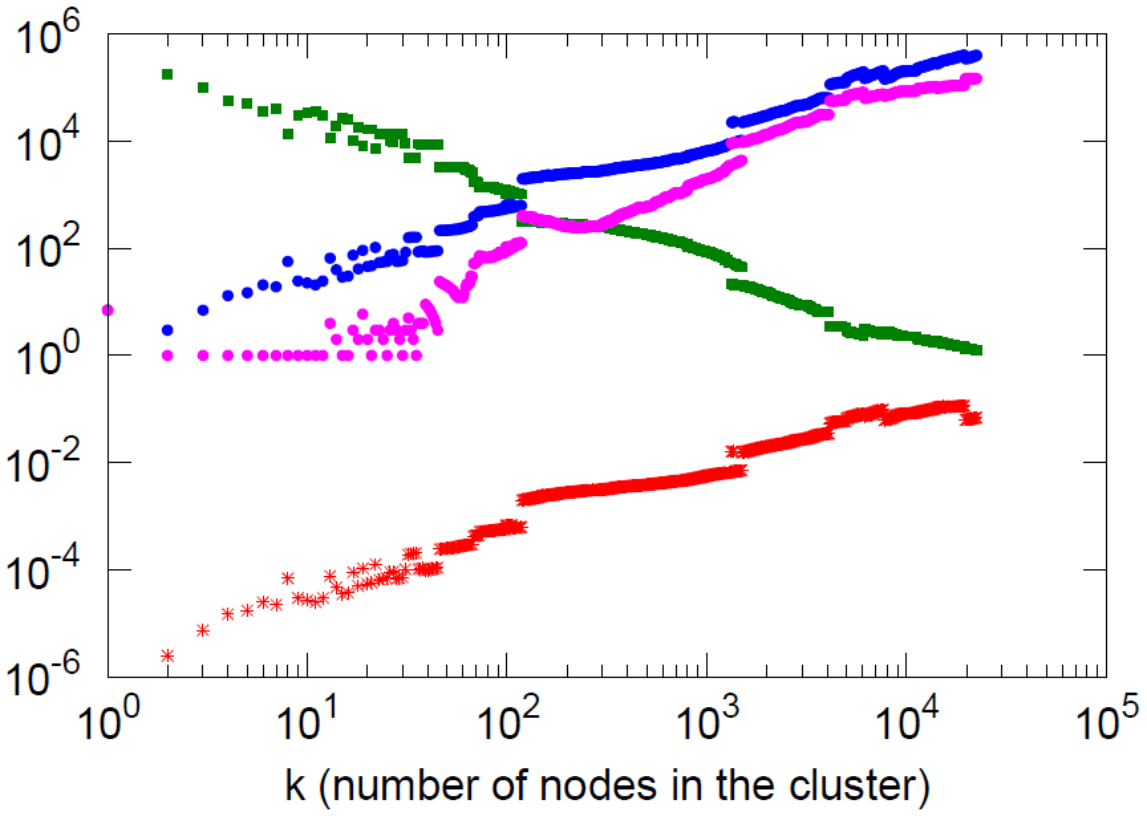
# Multi-criterion Objectives




- **All qualitatively similar**
- **Observations:**
  - Conductance, Expansion, Norm-cut, Cut-ratio are similar
  - Flake-ODF prefers larger clusters
  - Density is bad
  - Cut-ratio has high variance

[WWW '09]

# Single-criterion Objectives




**Observations:**

- All measures are monotonic
- **Modularity**
  - prefers large clusters
  - Ignores small clusters