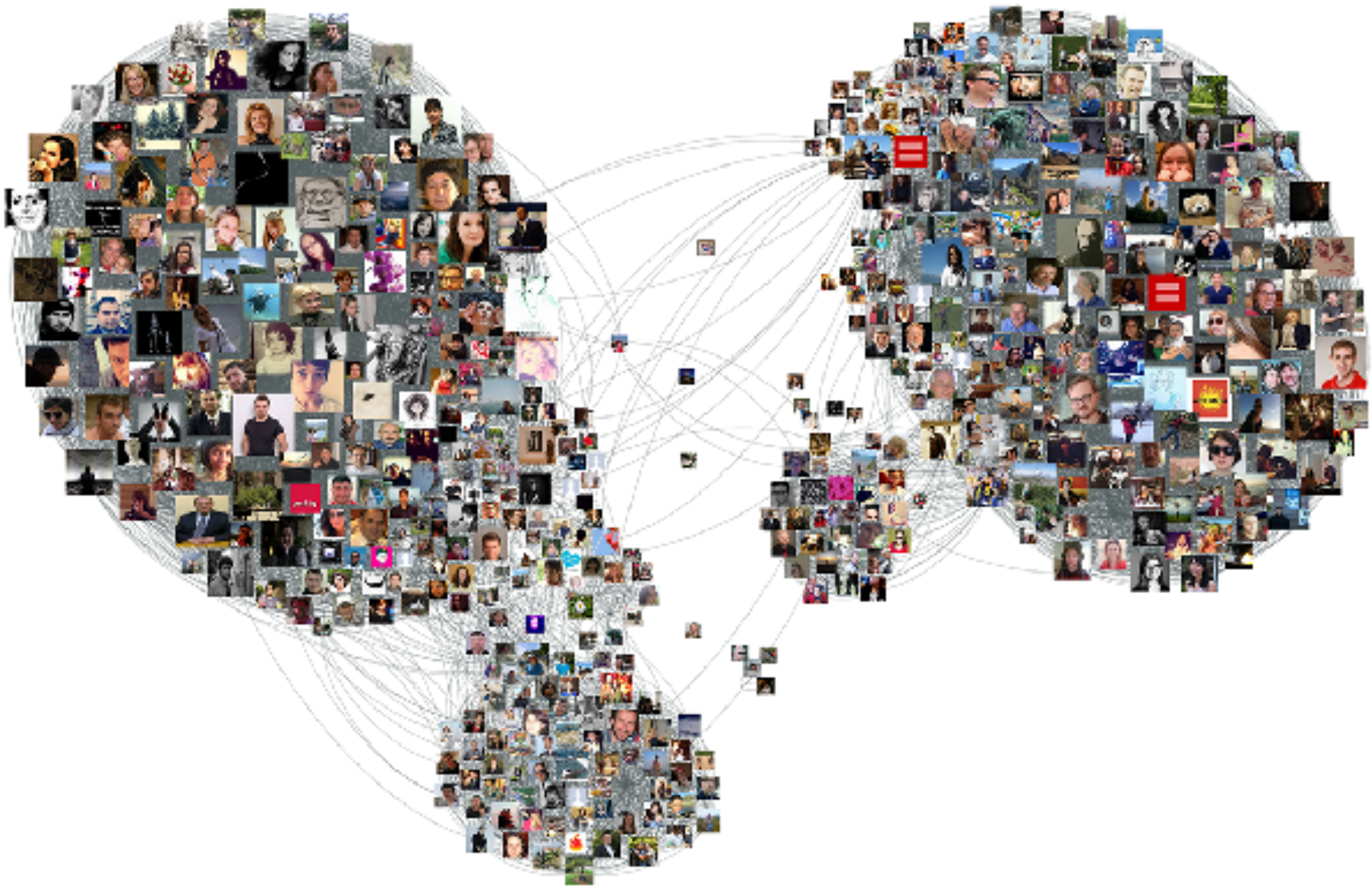
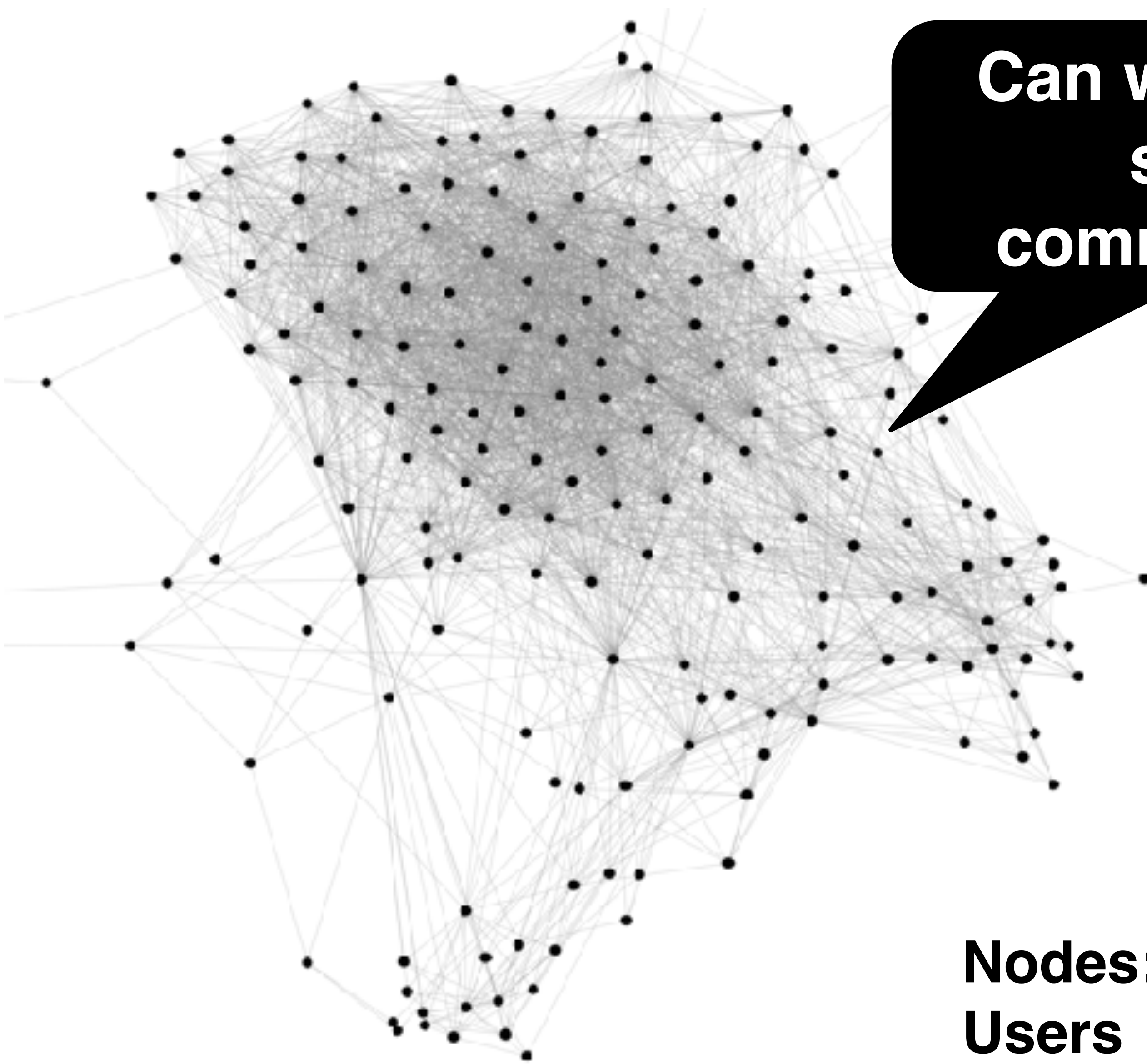
Social Network Analysis

Social Network Graph



Facebook Network

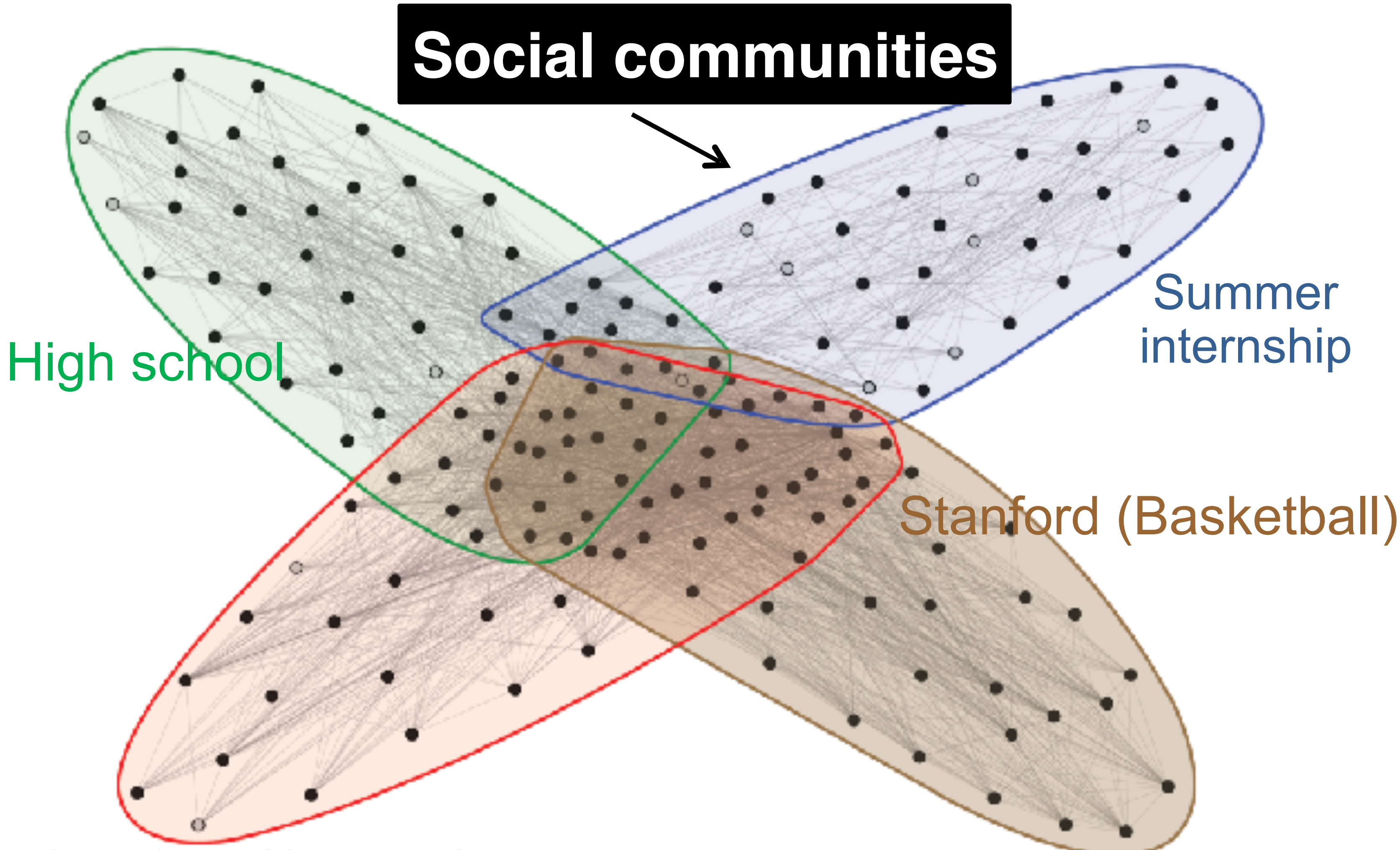


Can we identify social communities?

Nodes: Facebook Users

3

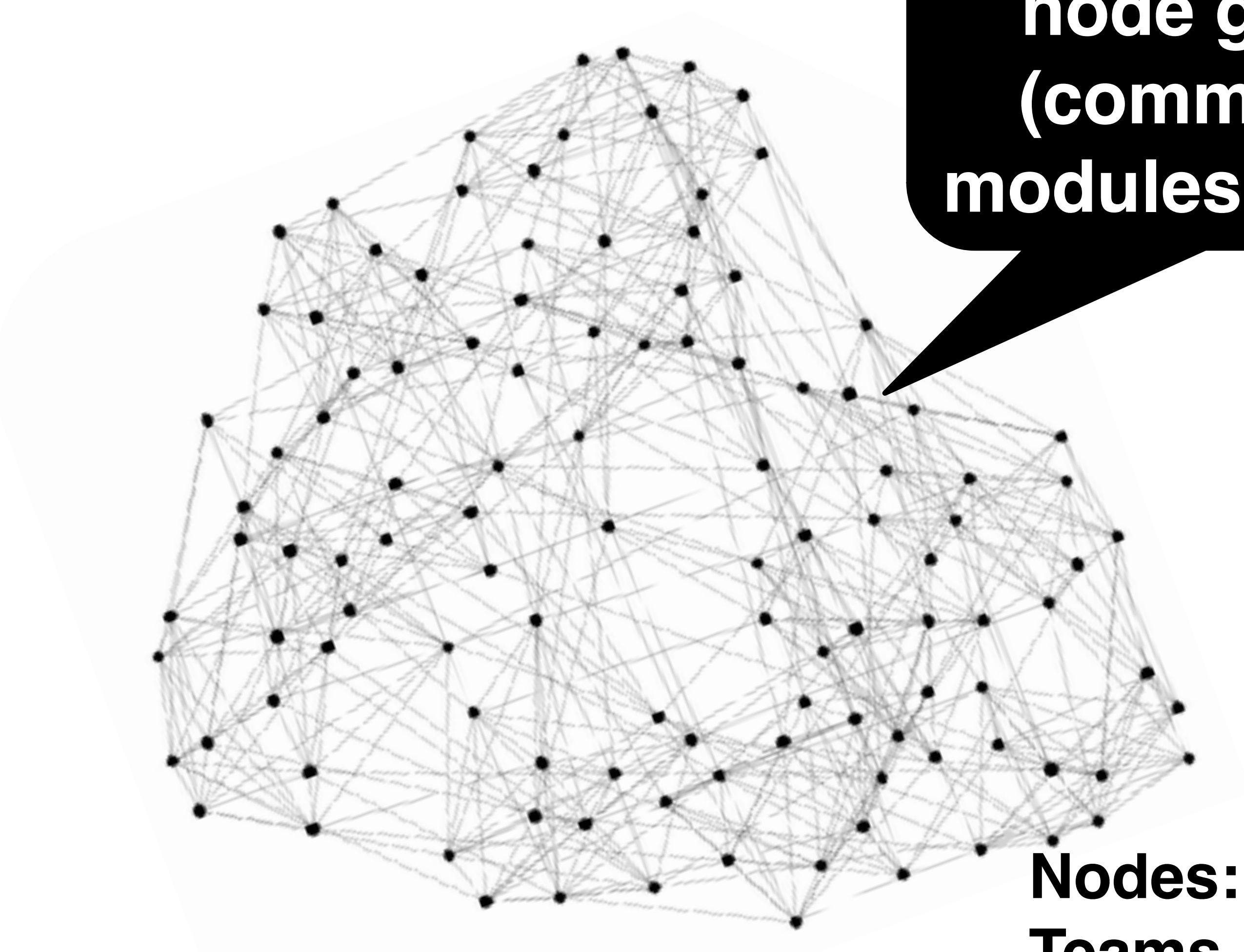
Facebook Network



Nodes: Facebook Users

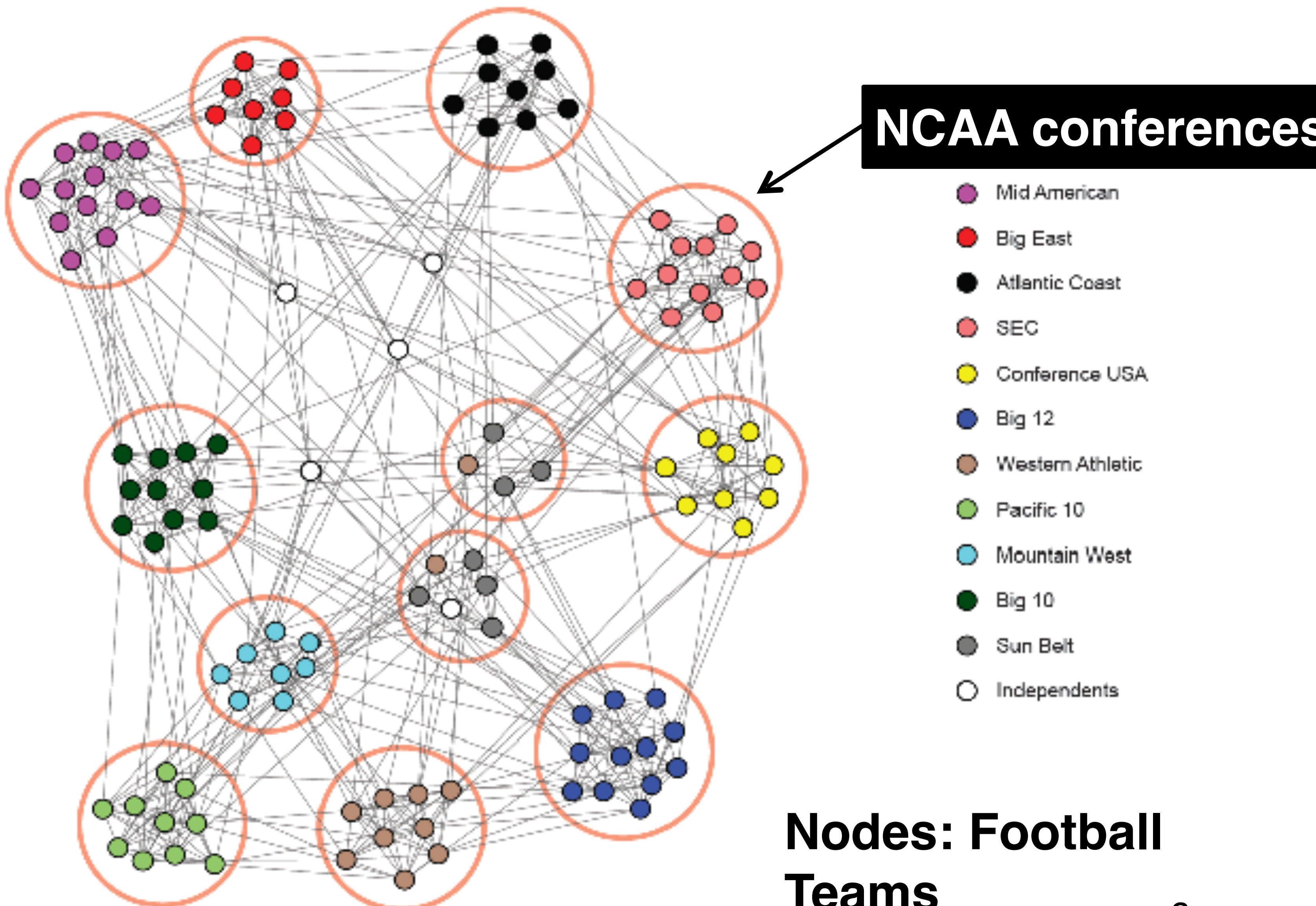
Identifying Communities

Can we identify node groups? (communities, modules, clusters)



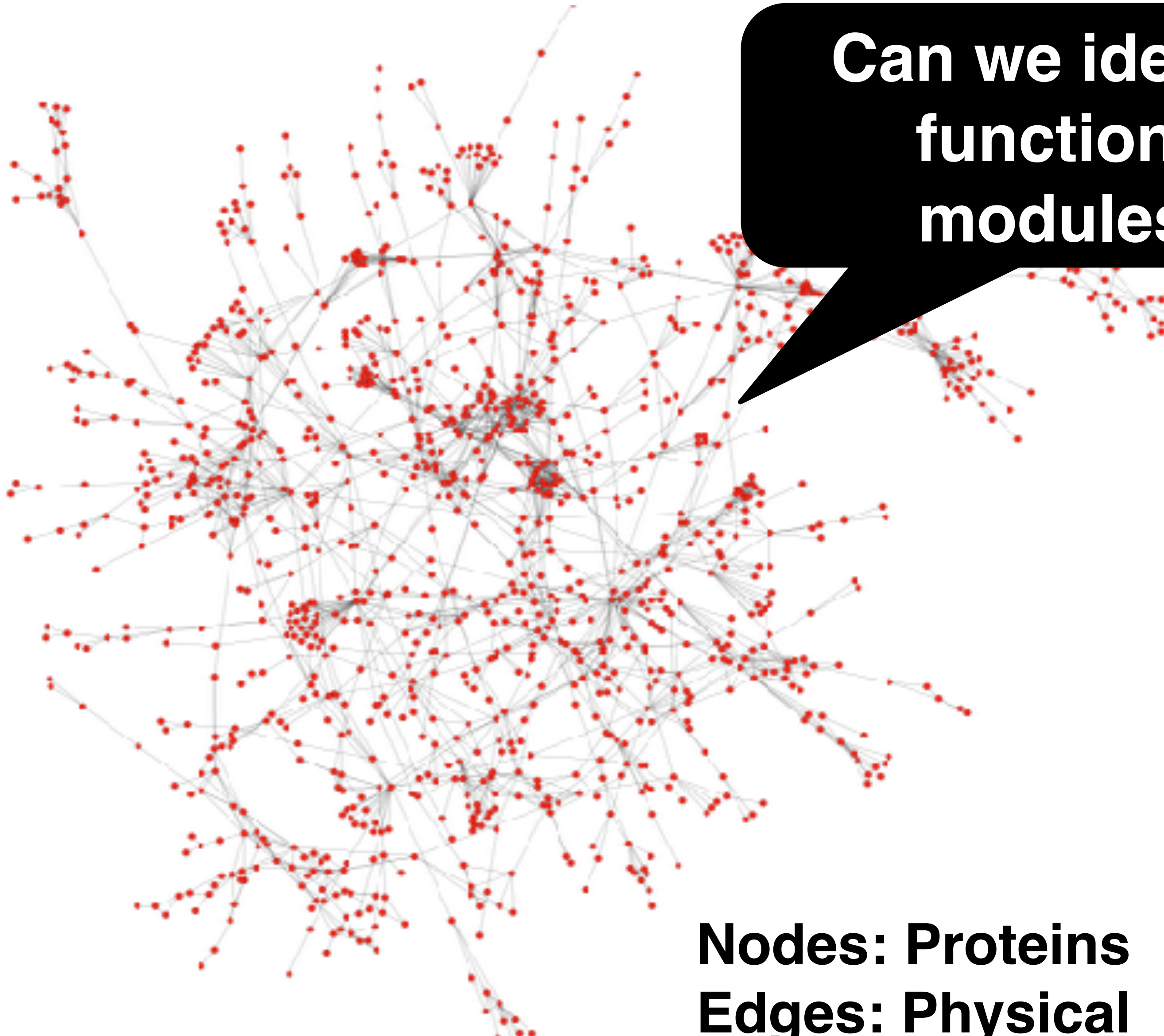
Nodes: Football Teams

NCAA Football Network



Nodes: Football Teams

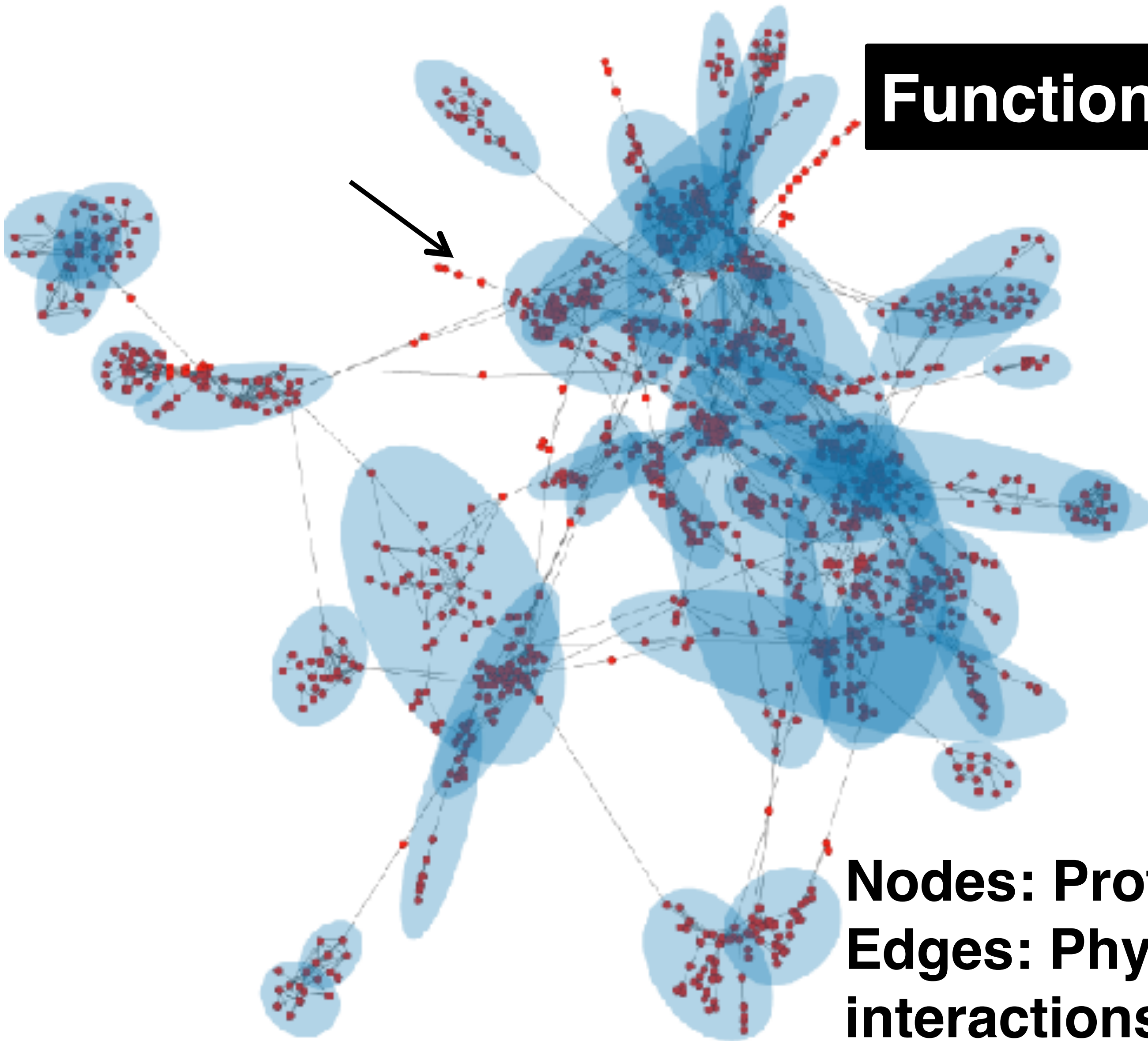
Other: Protein-Protein Interactions



Can we identify functional modules?

Nodes: Proteins
Edges: Physical

Other: Protein-Protein Interactions

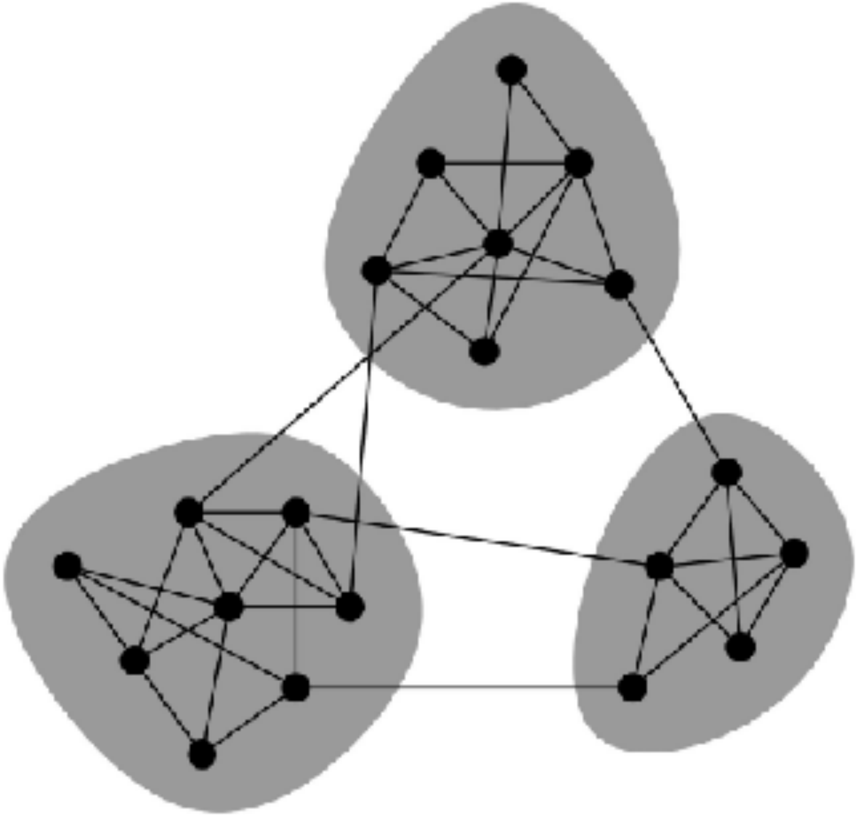


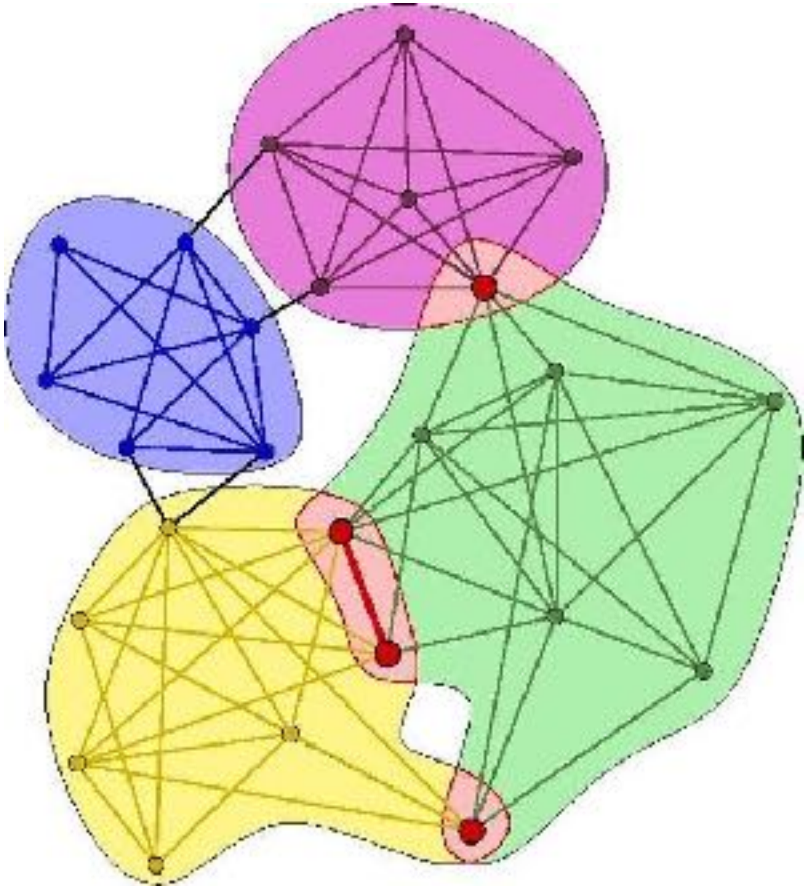
Functional modules

Nodes: Proteins
Edges: Physical interactions

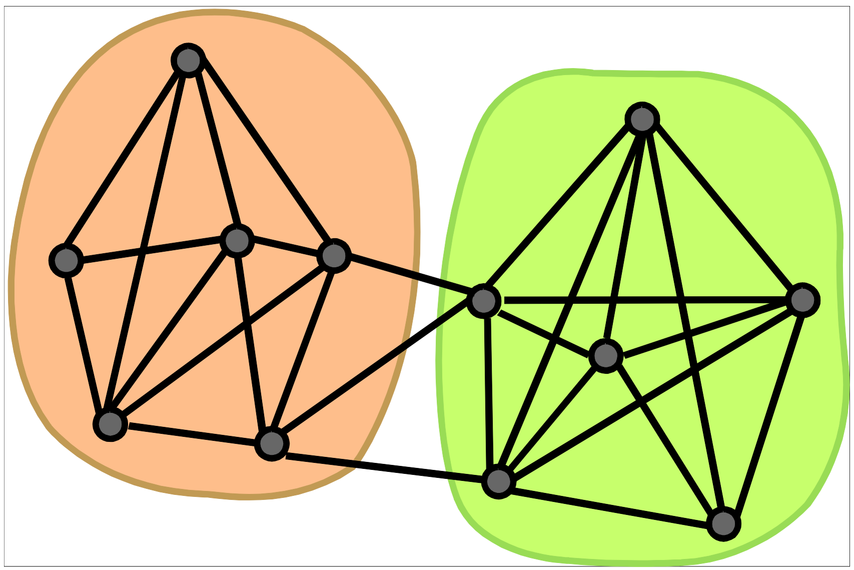
Overlapping Communities

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

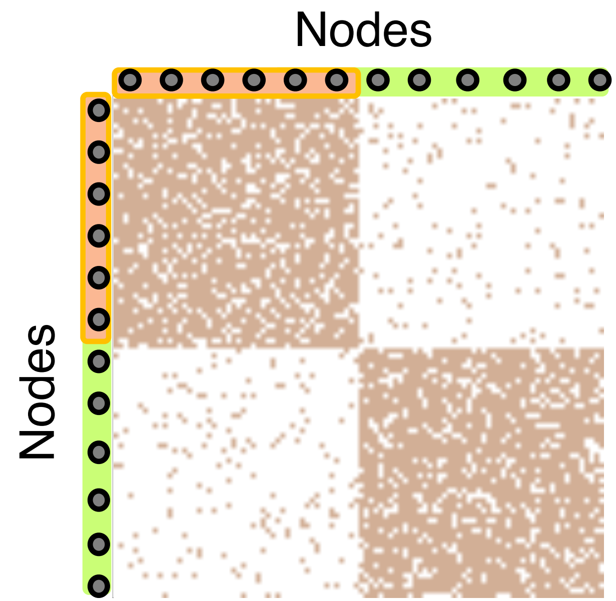




Non-overlapping Communities



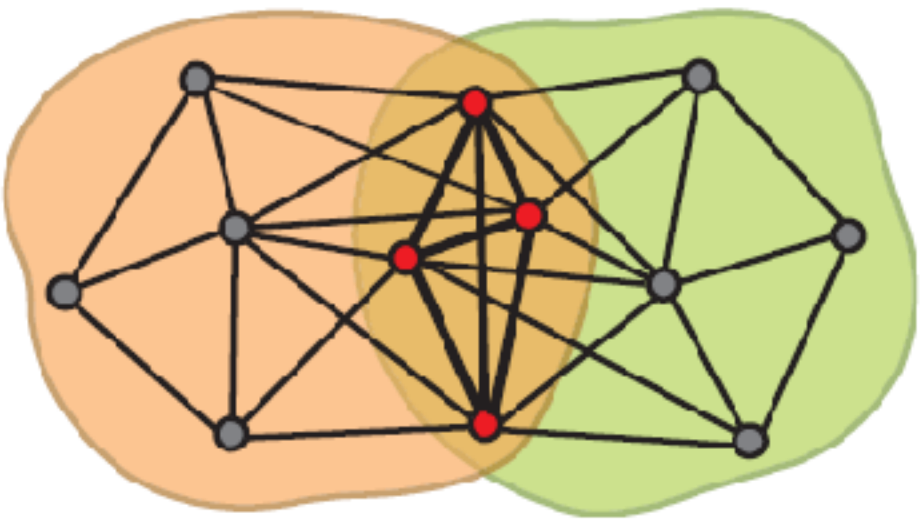
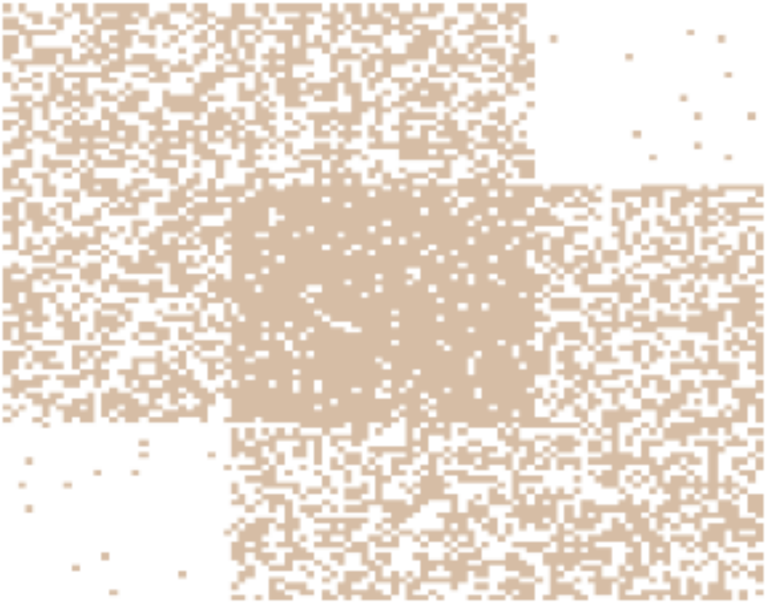
Network



Adjacency matrix

Communities as Tiles!

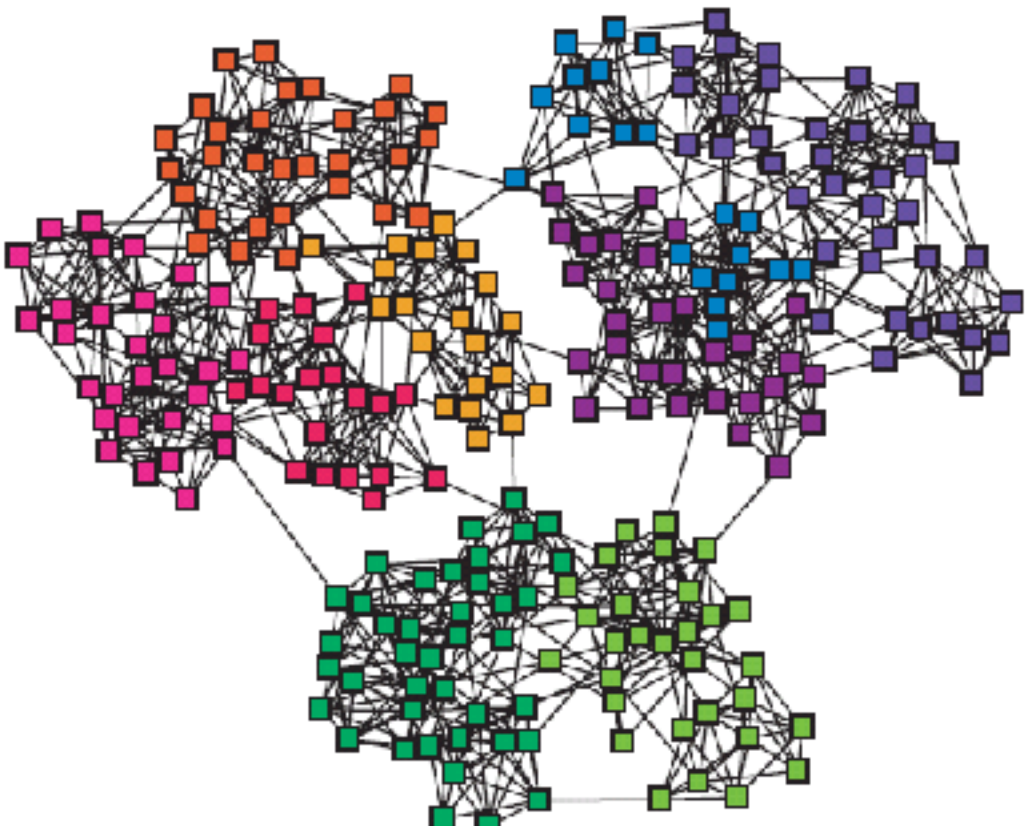
- What is the structure of community overlaps: Edge density in the overlaps is higher!

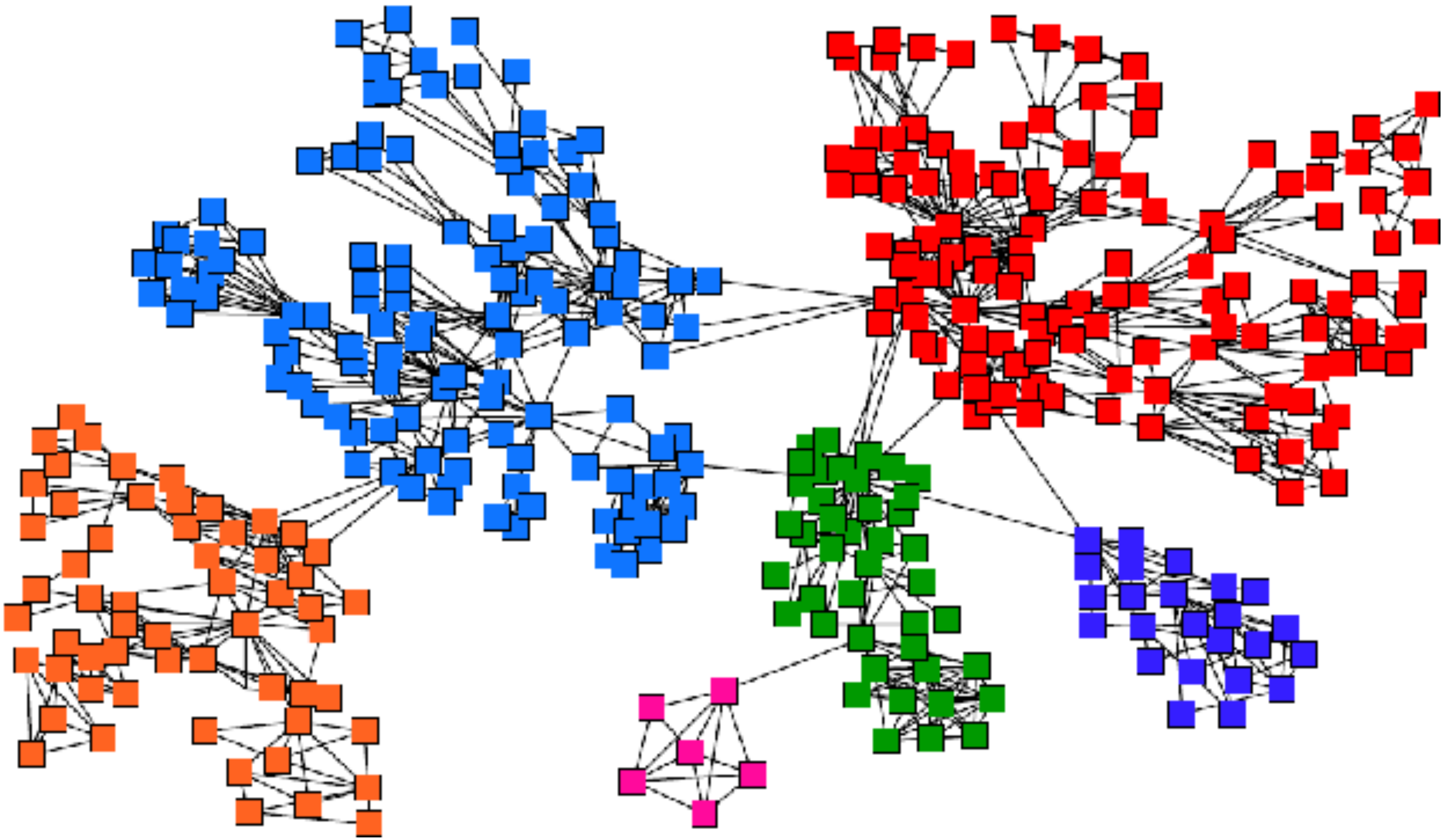
Communities as "tiles"

Networks & Communities

- **Often think networks organized into modules, cluster, communities:**

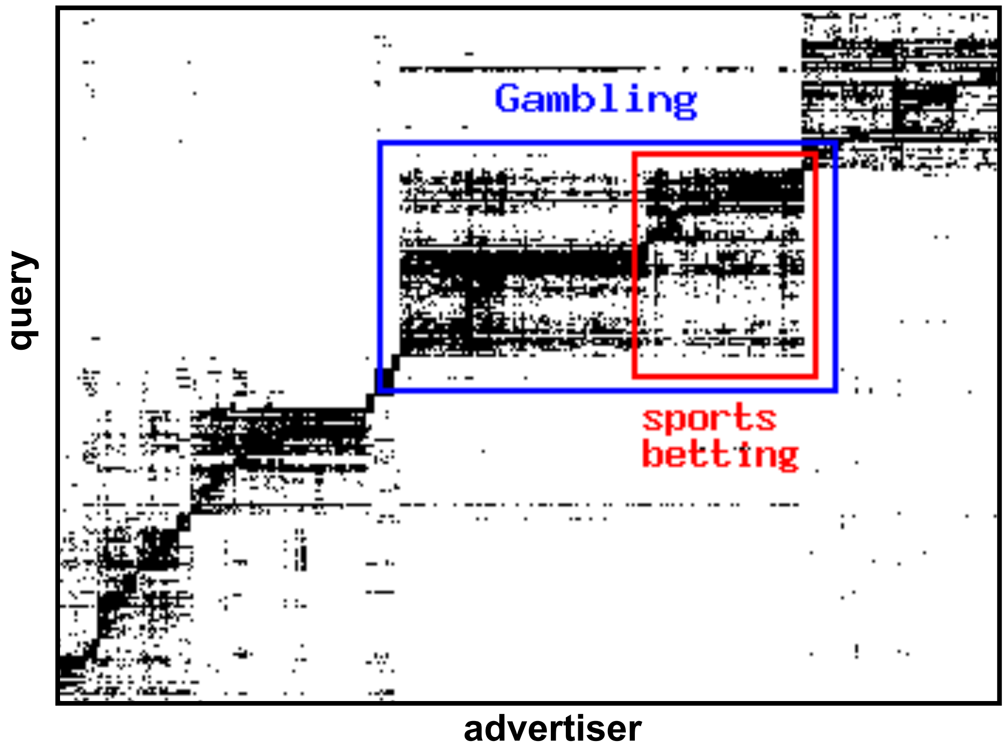


Goal: Find Densely Linked Clusters



Micro-Markets in Sponsored Search

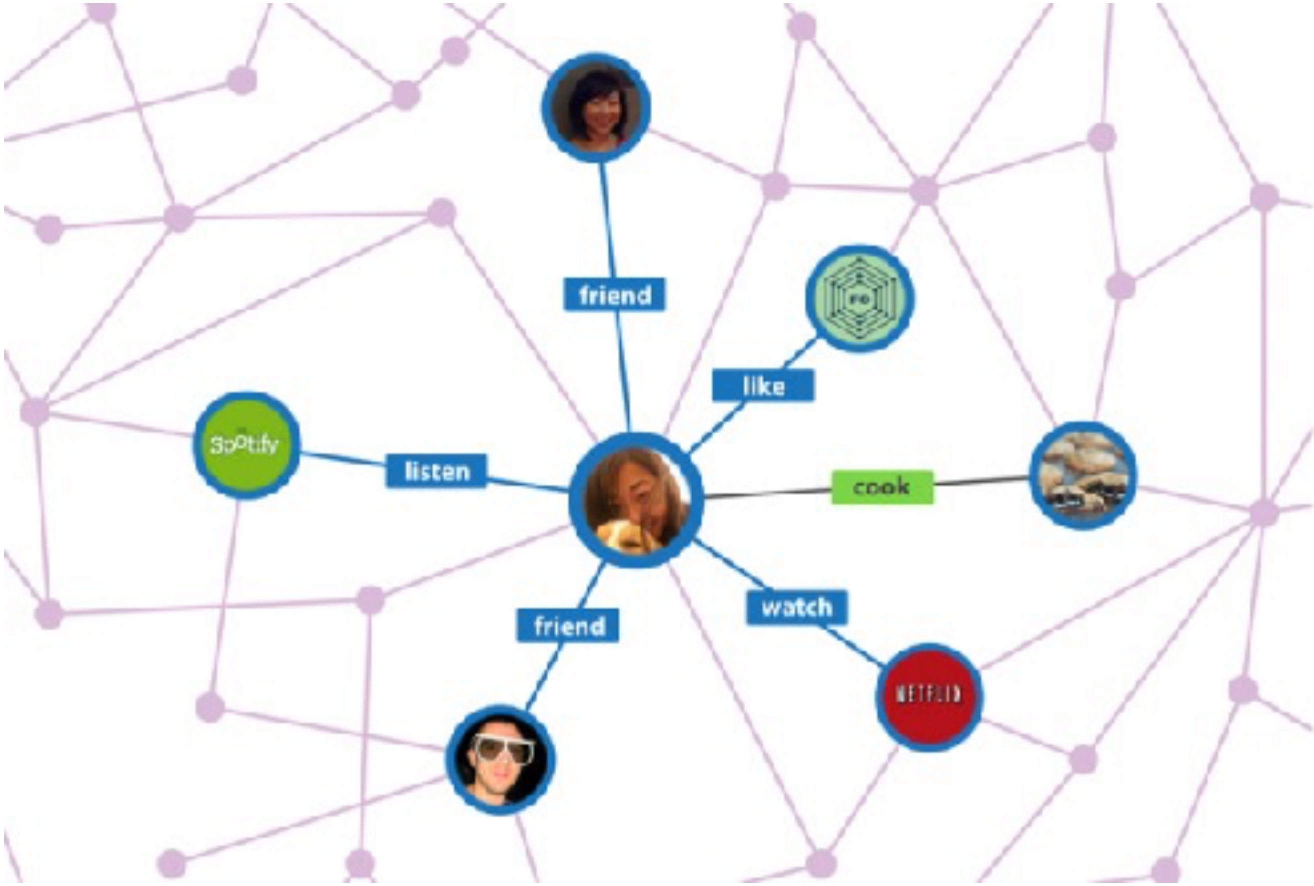
- Find micro-markets by partitioning the query-to-advertiser graph:



[Andersen, Lang: Communities from seed sets, 2006]

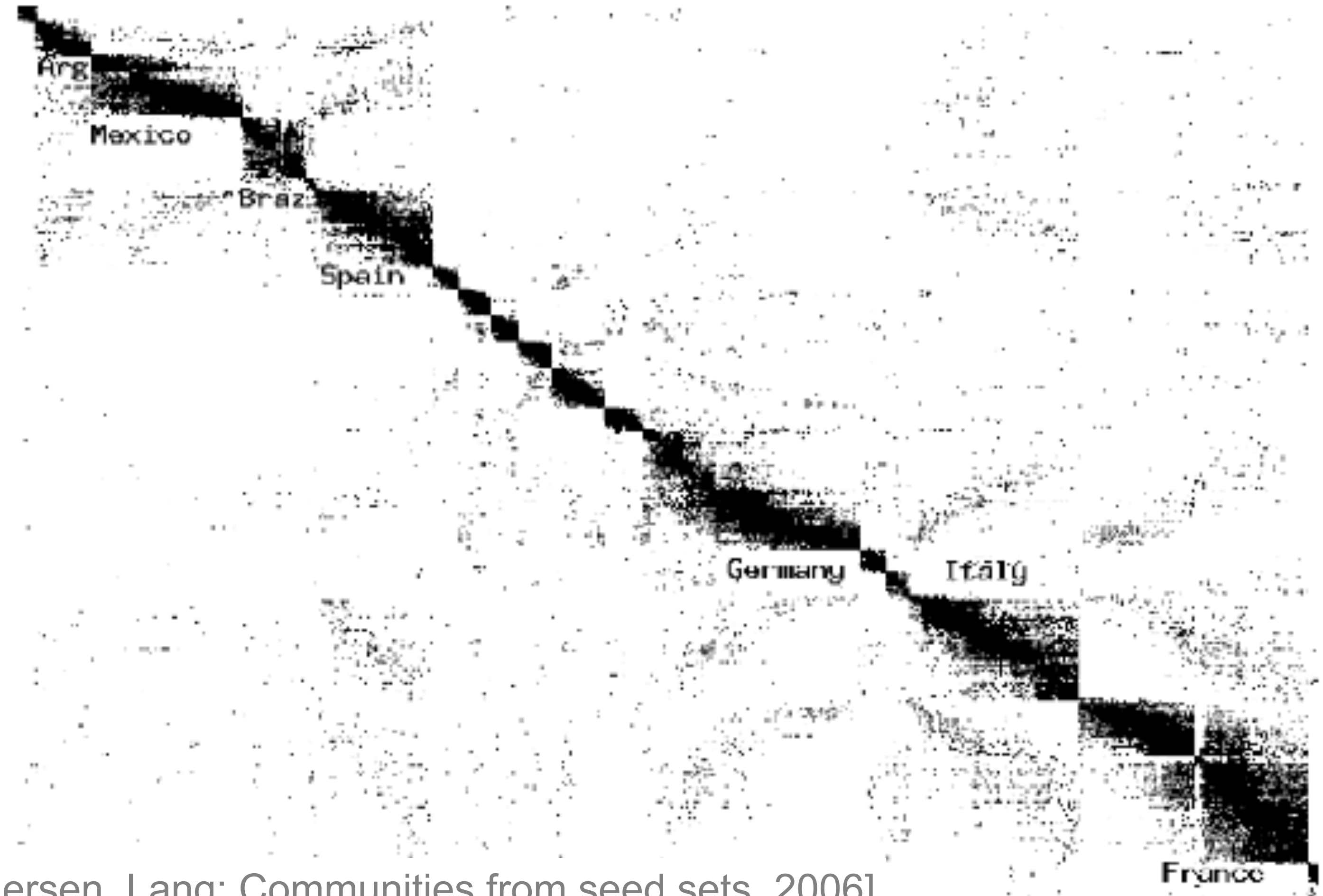
For Recommender Systems

- Find what a person likes



Movies and Actors

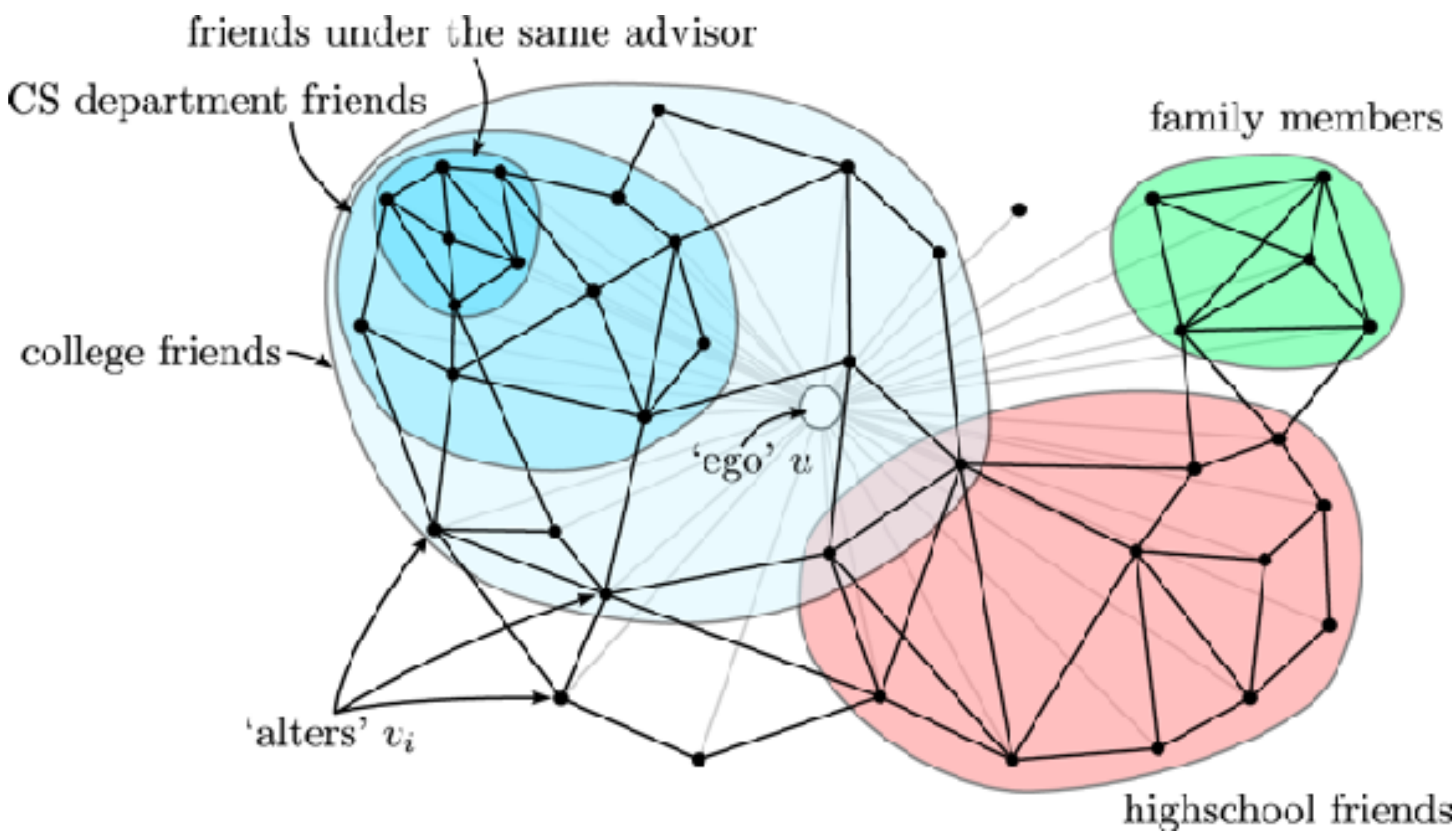
- Clusters in Movies-to-Actors graph:



[Andersen, Lang: Communities from seed sets, 2006]

Twitter & Facebook

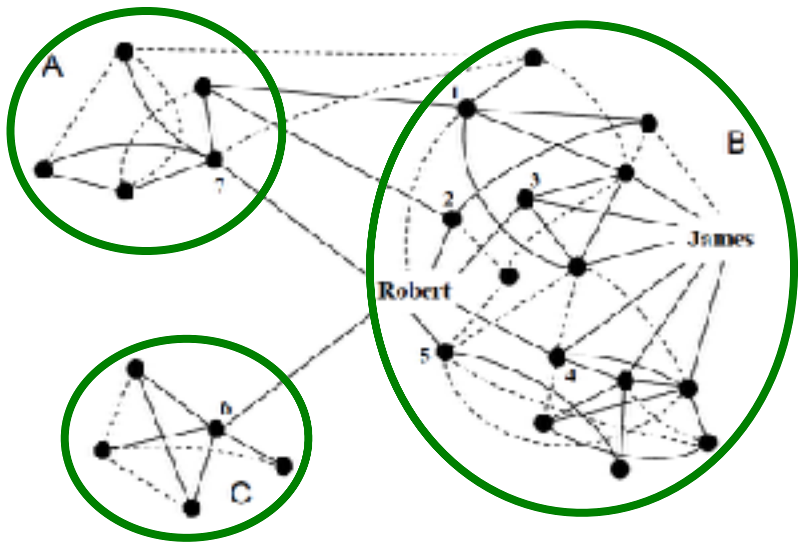
- Discovering social circles, circles of trust:

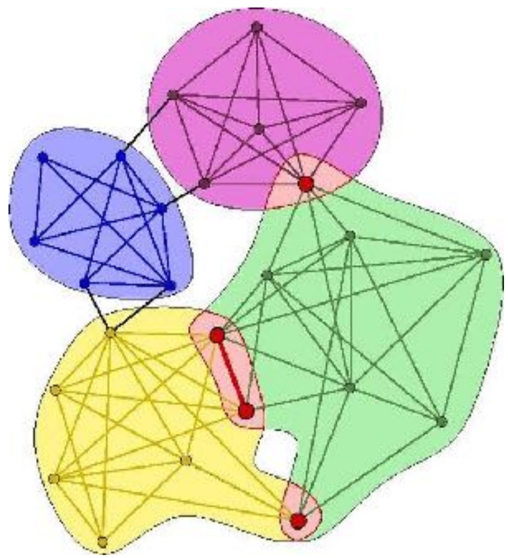


[McAuley, Leskovec: Discovering social circles in ego networks, 2012]

COMMUNITY DETECTION

How to find communities?

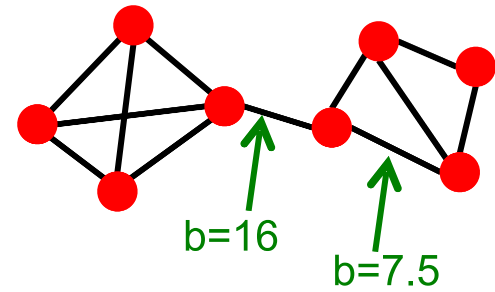


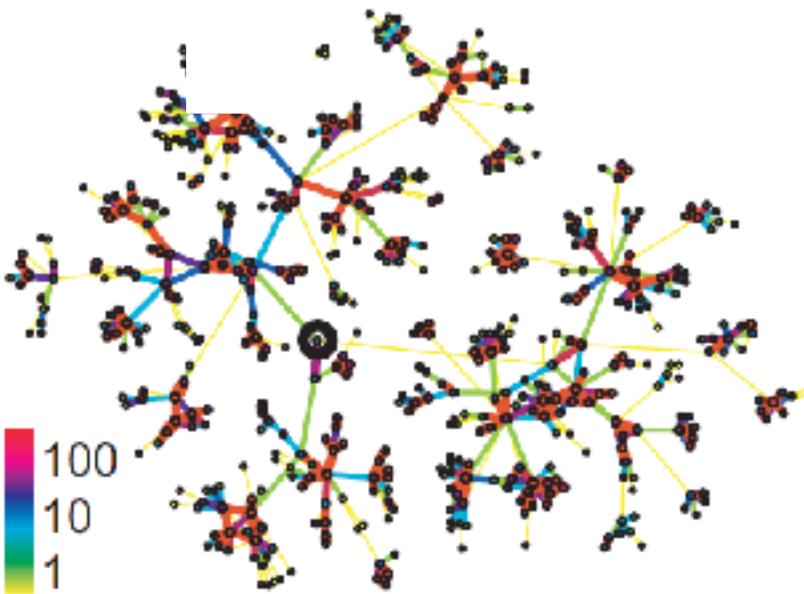


We will work with **undirected** (unweighted) networks

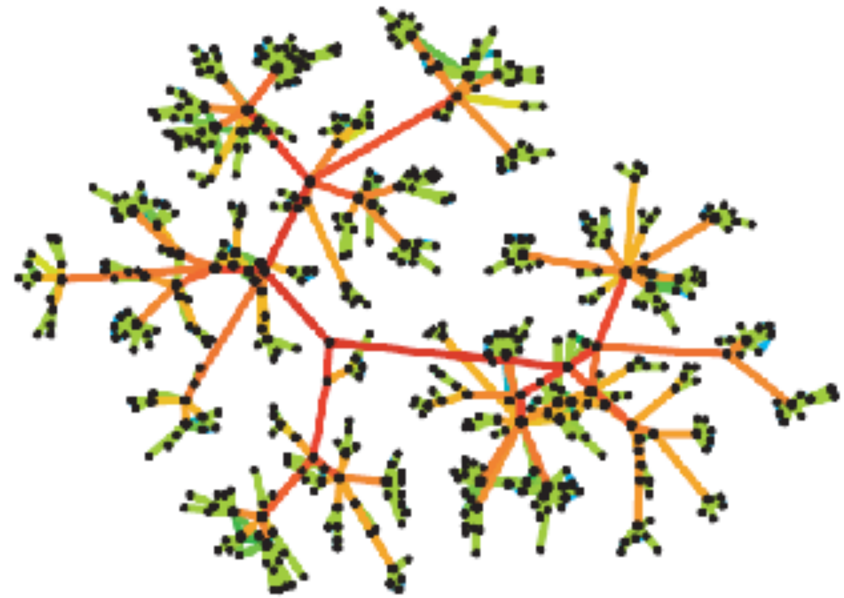
Method 1: Strength of Weak Ties

- **Edge betweenness:** Number of shortest paths passing over the edge
- Intuition:





Edge strengths (call volume) in a real network



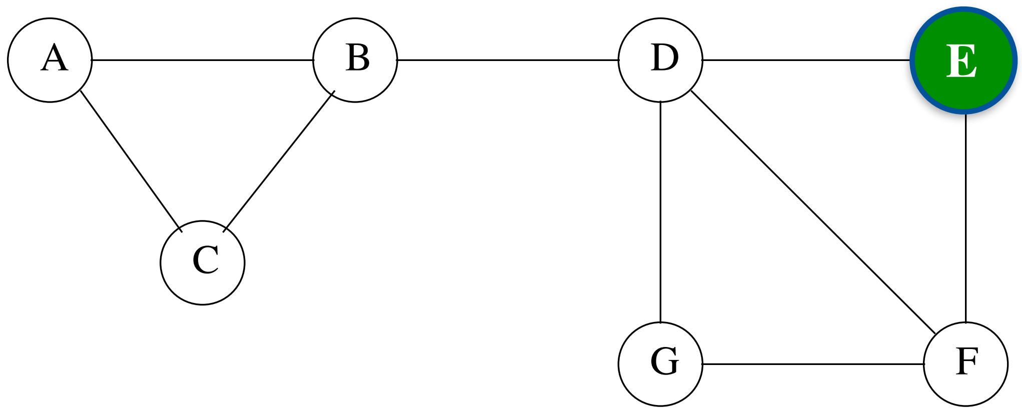
Edge betweenness in a real network

Method 1: Girvan-Newman

- Divisive hierarchical clustering based on the notion of **edge betweenness**:
  – Number of shortest paths passing through the edge
- **Girvan-Newman Algorithm**:

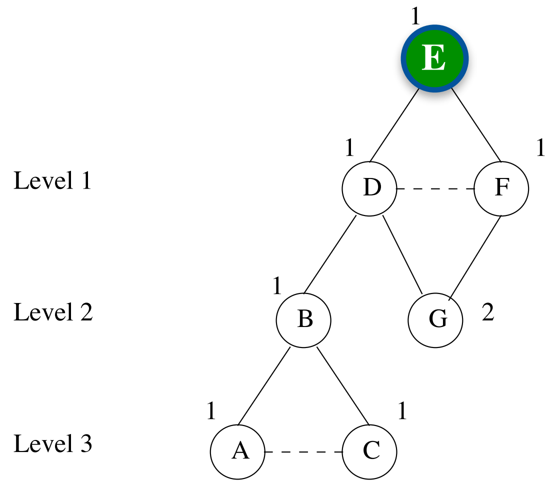
  Undirected unweighted networks
  – Repeat until no edges are left:
    - Calculate betweenness of edges
    - Remove edges with highest betweenness
  – Connected components are communities
  – Gives a hierarchical decomposition of the network

Girvan-Newman: Example

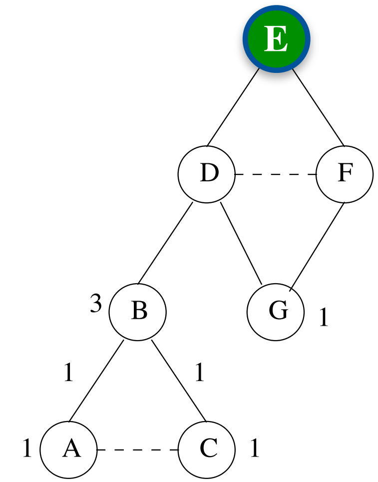


- Repeat until no edges are left:
  - **Calculate betweenness of edges**
  - Remove edges with highest betweenness
- Connected components are communities
- Gives a hierarchical decomposition of the network

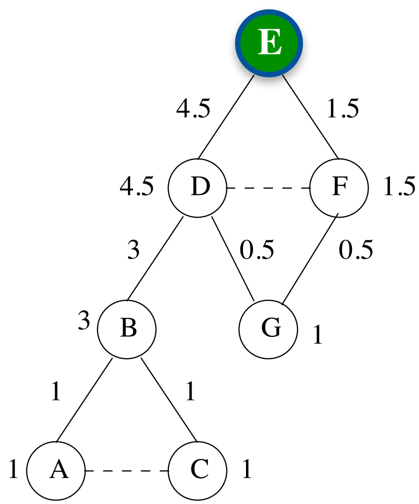
Girvan-Newman: Example



Step 1



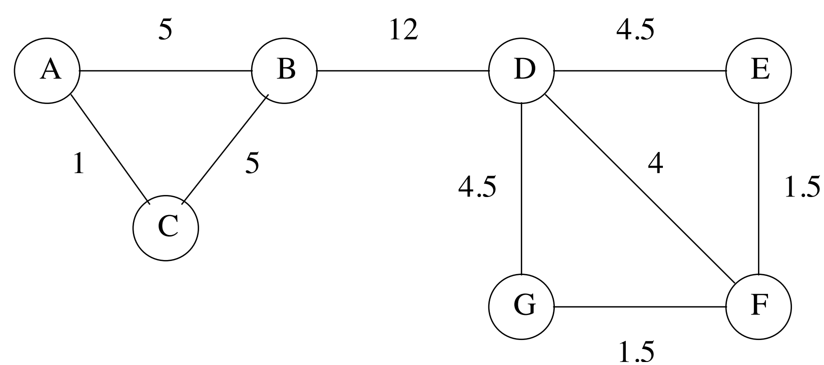
Step 2



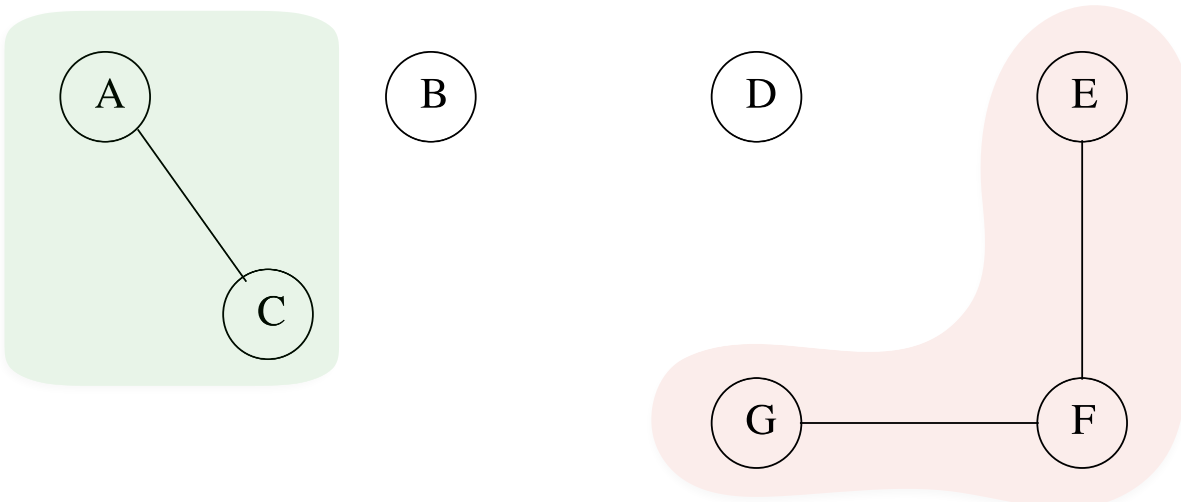
Step 3

…

Re-compute betweenness at every step

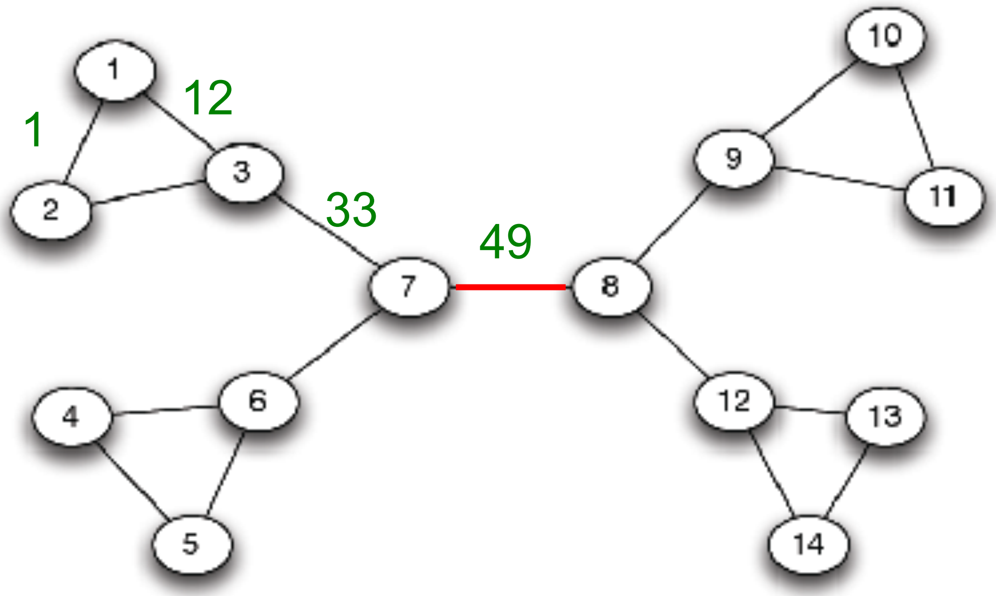


Girvan-Newman: Example



- Repeat until no edges are left:
  - Calculate betweenness of edges
  - **Remove edges with highest betweenness**
- **Connected components are communities**
- Gives a hierarchical decomposition of the network

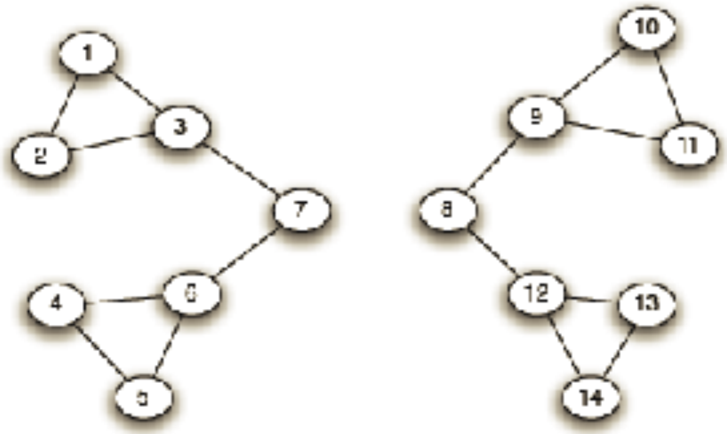
Girvan-Newman: Example



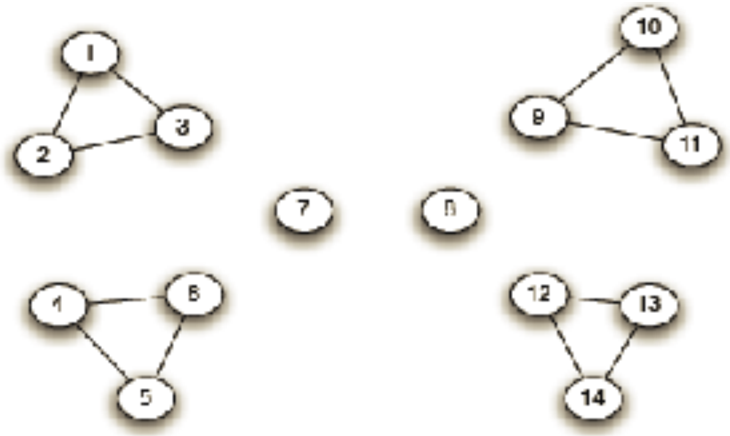
Need to re-compute betweenness at every step

Girvan-Newman: Example

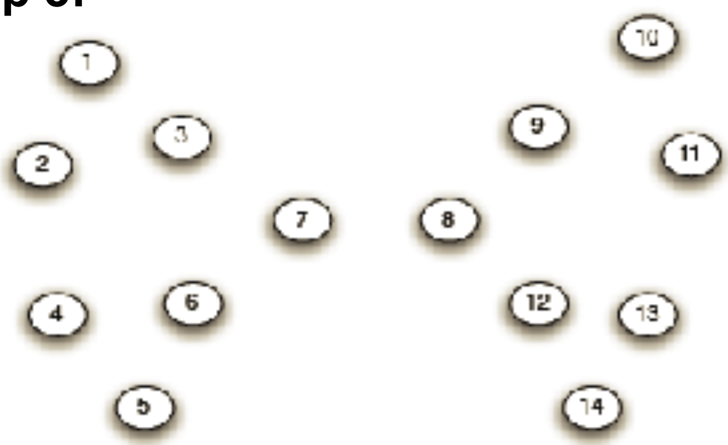
Step 1:



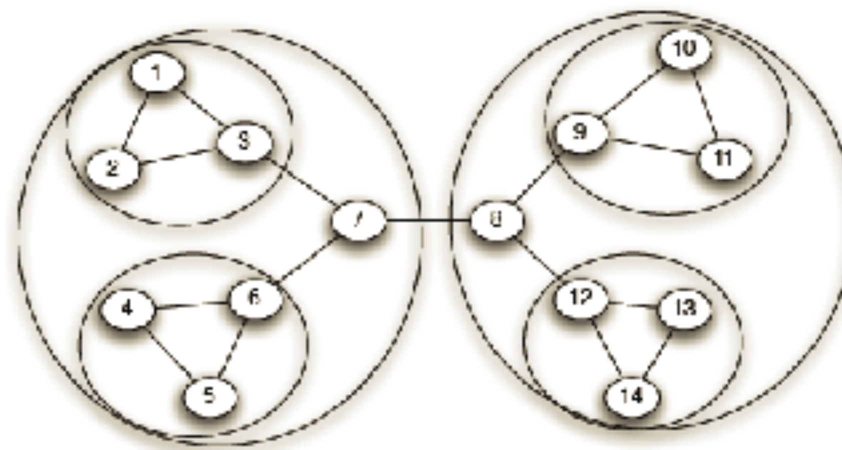
Step 2:



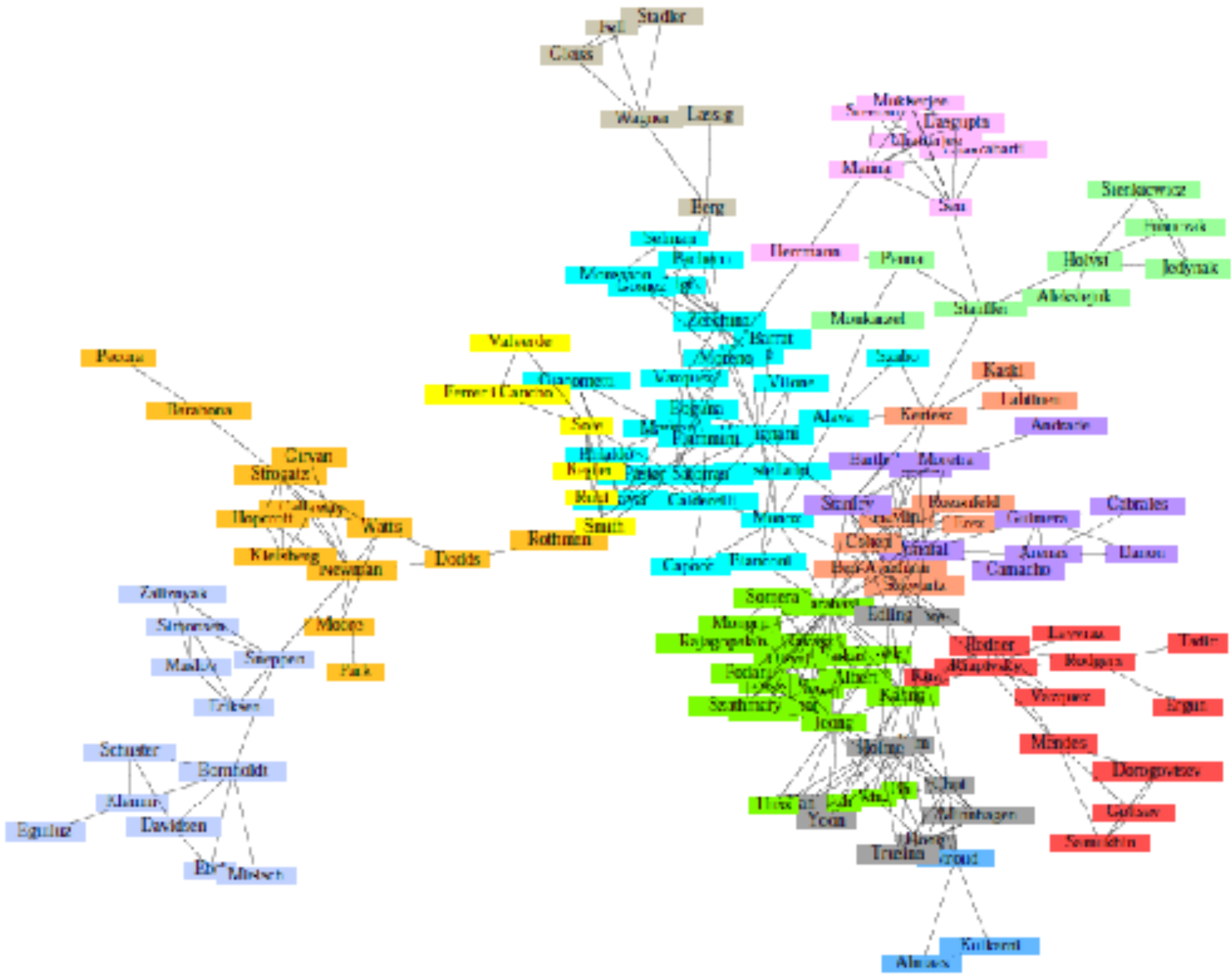
Step 3:



Hierarchical network decomposition:



Girvan-Newman: Results

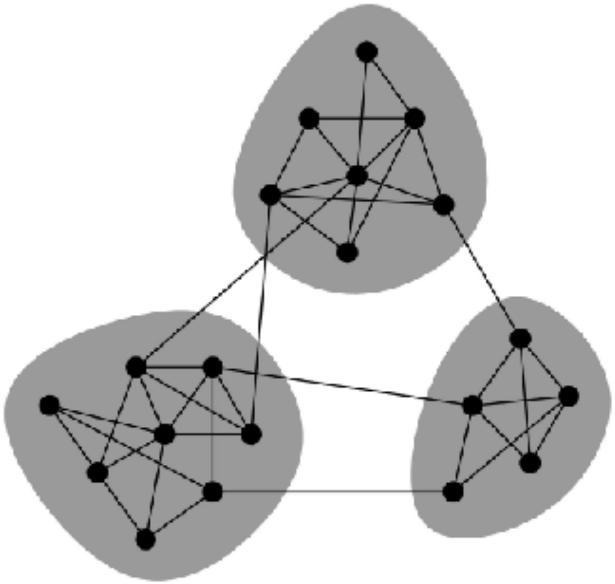


Communities in physics collaborations

Network Communities

- **Communities:** sets of **tightly connected nodes**
- Define: **Modularity *Q***
  - A measure of how well a network is partitioned into communities
  - Given a partitioning of the network into groups $s \in S$:

$$Q \propto \sum_{s \in S} [\,(\#\text{ edges within group } s) - (\text{expected } \#\text{ edges within group } s)\,]$$

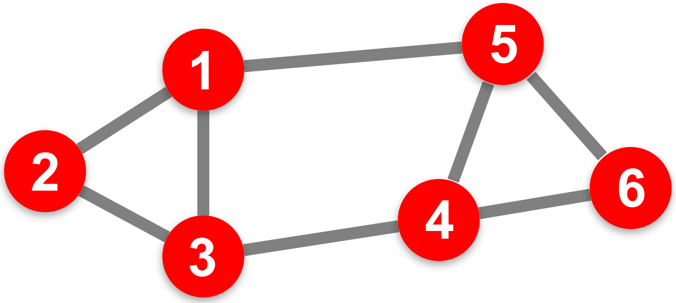


Need a null model!

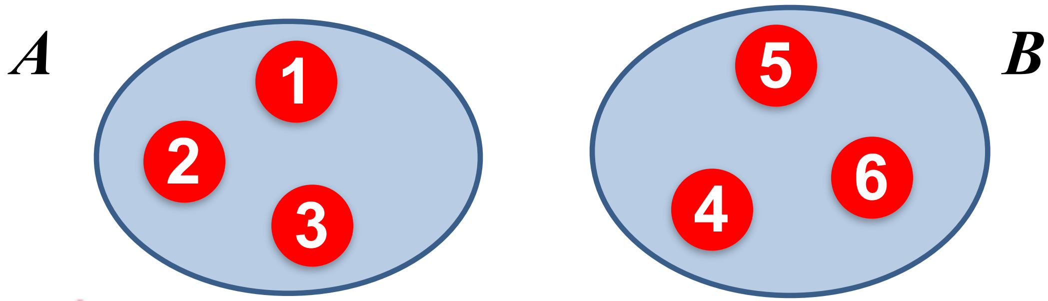
Spectral Clustering

Graph Partitioning

- Undirected graph $G(V, E)$:



- **Bi-partitioning task:**
  - Divide vertices into two disjoint groups A, B

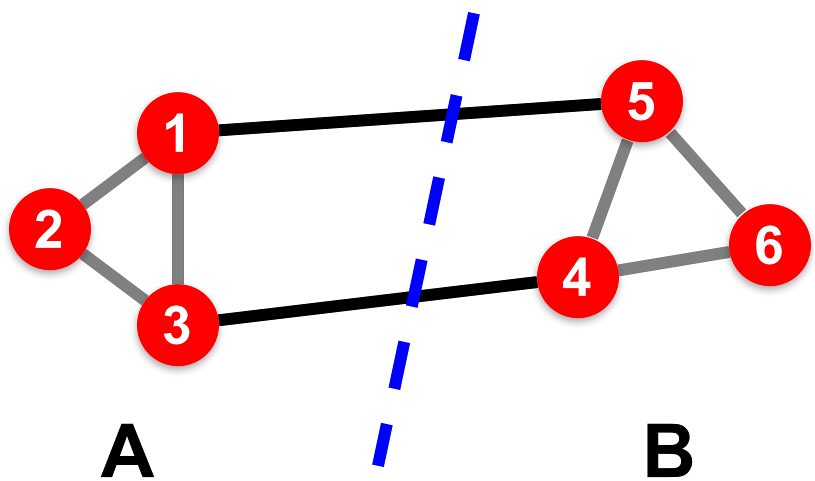


- **Questions:**
  - How can we define a "good" partition of G?
  - How can we efficiently identify such a partition?

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Graph Partitioning

- **What makes a good partition?**
  - Maximize the number of within-group connections
  - Minimize the number of between-group connections

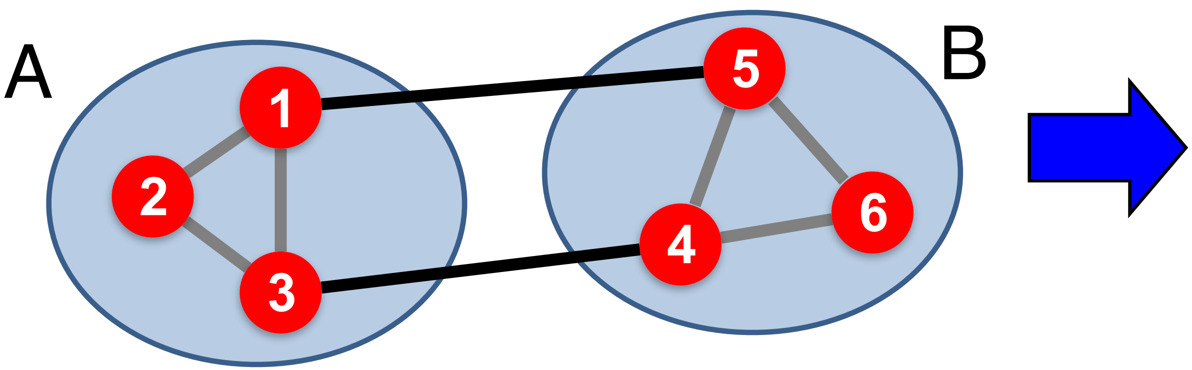


J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Graph Cuts

- Express partitioning objectives as a function of the “edge cut” of the partition
- Cut: Set of edges with only one vertex in a group:

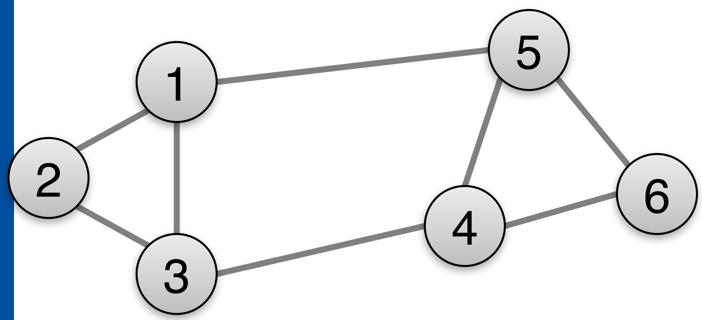
$$cut(A,B) = \sum_{i \in A,\, j \in B} w_{ij}$$



$cut(A,B) = 2$

Matrix Representations

- **Adjacency matrix (A):**
  - $n \times n$ matrix
  - $A=[a_{ij}]$, $a_{ij}=1$ if edge between node i and j



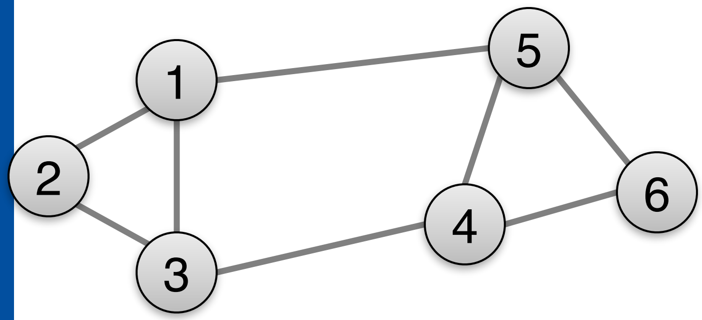
	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

- **Important properties:**
  - Symmetric matrix
  - Eigenvectors are real and orthogonal

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Matrix Representations

- **Degree matrix (D):**
  - $n \times n$ diagonal matrix
  - $D=[d_{ii}]$, $d_{ii}=$ degree of node i

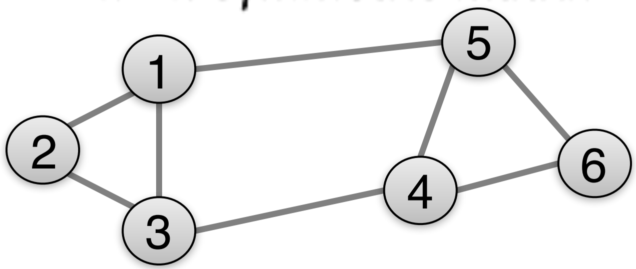


	1	2	3	4	5	6
1	3	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	3	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	2

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Matrix Representations

- **Laplacian matrix (L):**
  - $n \times n$ symmetric matrix



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

- **What is trivial eigenpair?**
  - $\boldsymbol{x} = (1, \dots, 1)$ then $\boldsymbol{L} \cdot \boldsymbol{x} = \boldsymbol{0}$ and so $\lambda = \lambda_1 = 0$
- **Important properties:**
  - **Eigenvalues** are non-negative real numbers
  - **Eigenvectors** are real and orthogonal

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org