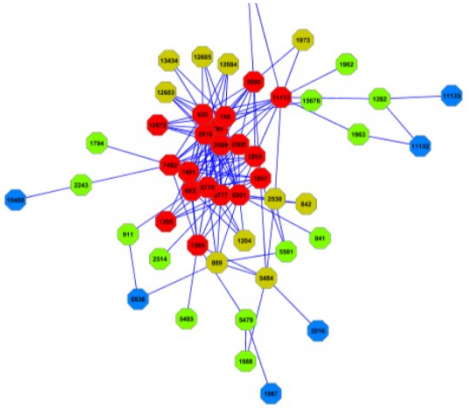
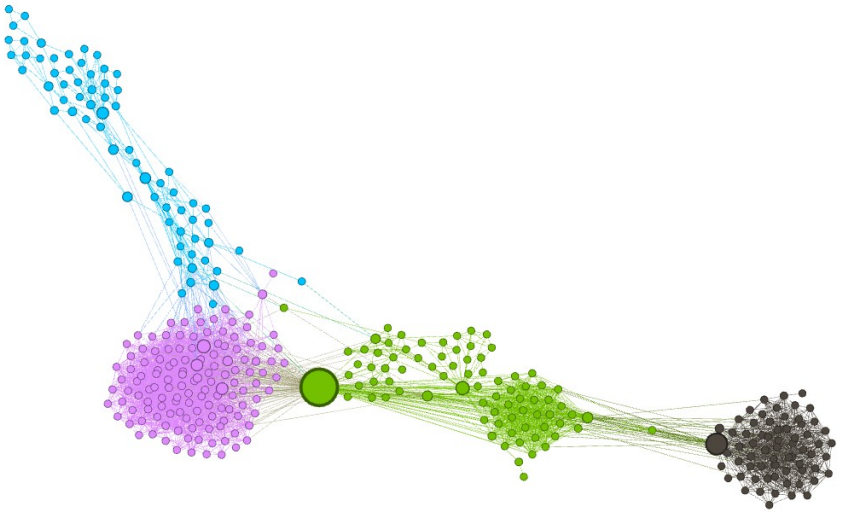
Community Structure in Networks

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Pedro Ribeiro
(DCC/FCUP & CRACS/INESC-TEC)

CRACS
INESCTEC
ASSOCIATE LABORATORY
PORTUGAL

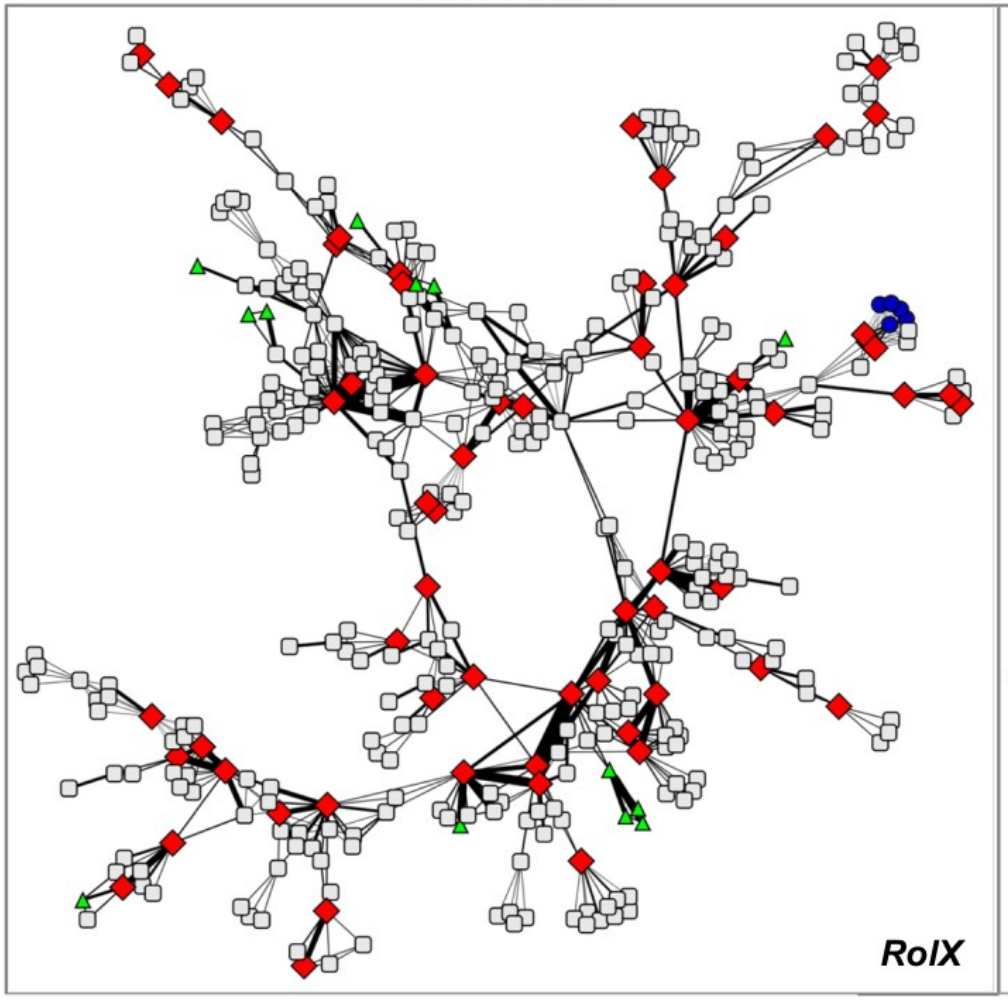




(Mainly selected slides from Jure Leskovec and Gonzalo Mateos)

Roles and Communities

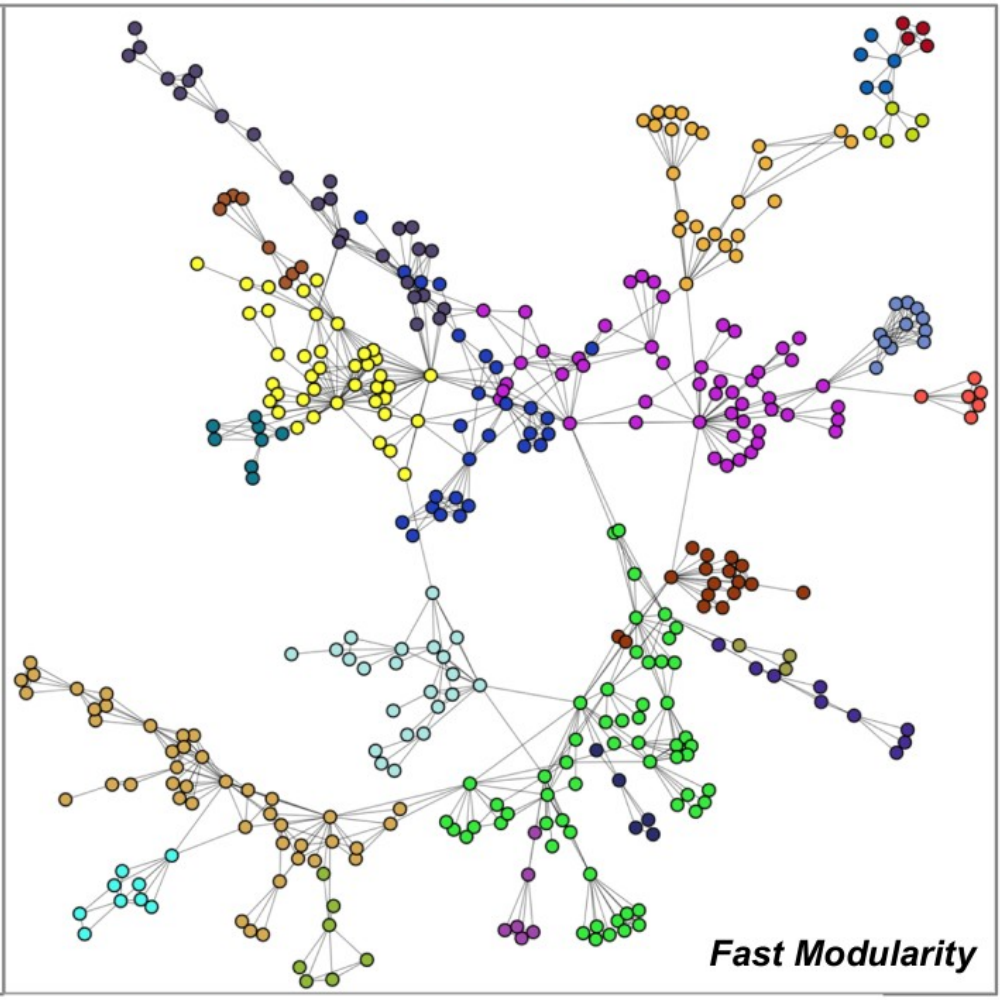
Roles



Henderson, *et al.*, KDD 2012

Nodes with different structural roles (connector node, bridge node, etc.)

Communities



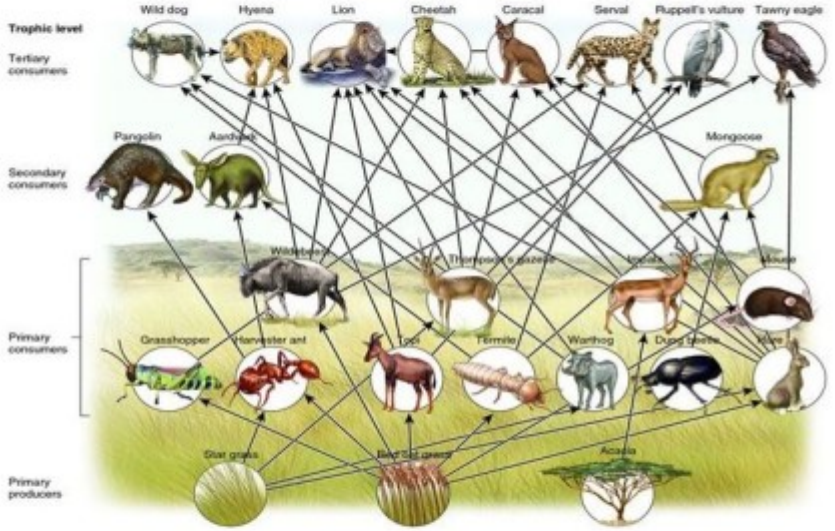
Clauset, *et al.*, Phys. Rev. E 2004

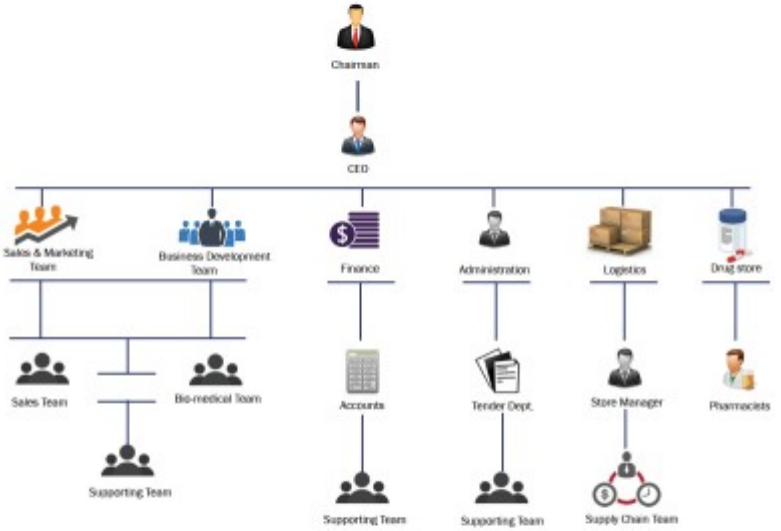
Nodes belonging to the same cluster/community

Structural Roles

What are Roles?

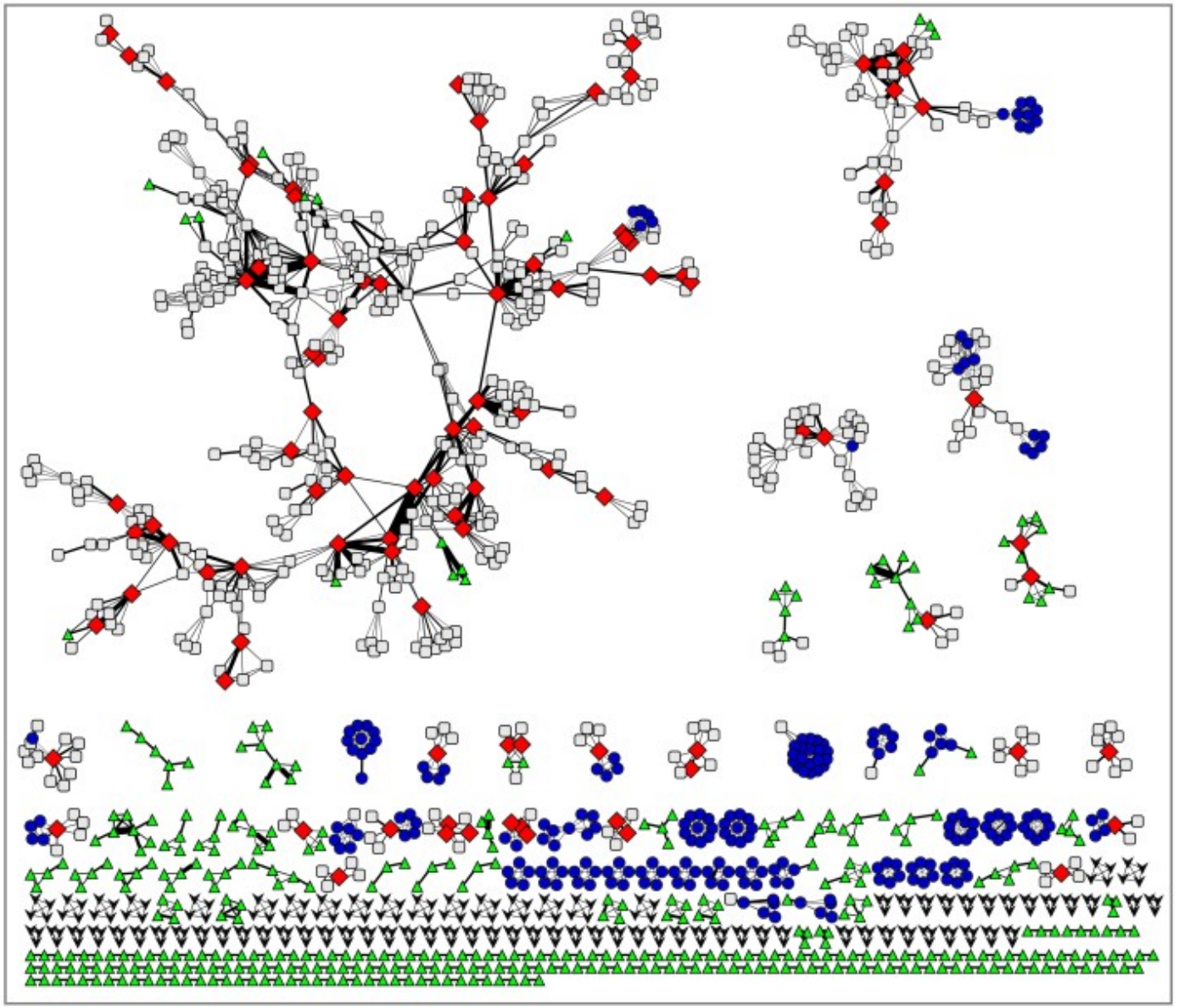
- **Roles are "functions" of nodes in a network:**
  - Roles of **species** in **ecosystems**
  - Roles of **individuals** in **companies**





- **Roles are measured by structural behaviors:**
  - Centers of stars
  - Members of cliques
  - Peripheral nodes, etc.

Examples of Roles



◆ centers of stars
○ members of cliques
▲ peripheral nodes

Network Science
Co-authorship network
[Newman 2006]

Roles vs Groups in Networks

- **Role**: A collection of nodes which have similar positions in a network:
  - Roles are based on the similarity of ties among subsets of nodes
  - Different from **community** (or cohesive subgroup)
    - Group is formed based on adjacency, proximity or reachability
    - This is typically adopted in current data mining

Nodes with the same role need not be in direct, or even indirect interaction with each other

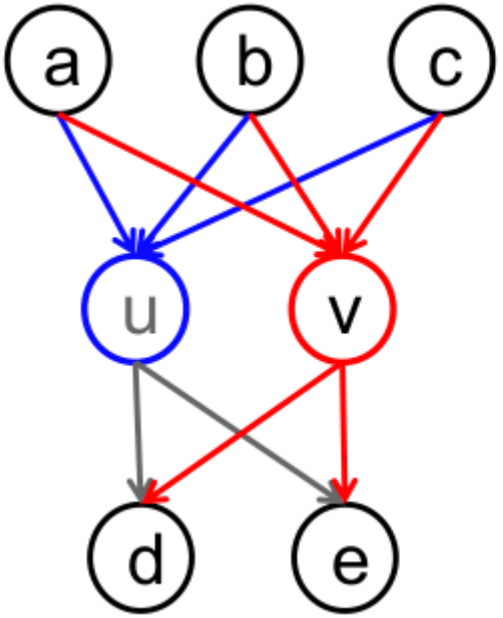
Roles and Communities

- **Roles**:
  - A group of nodes with similar structural properties
- **Communities:**
  - A group of nodes that are well-connected to each other
- Roles and communities **are complementary**

- Consider the social network of a CS Dept:
  - **Roles:** Faculty, Staff, Students
  - **Communities:** AI Lab, Info Lab, Theory Lab

Roles: More Formally

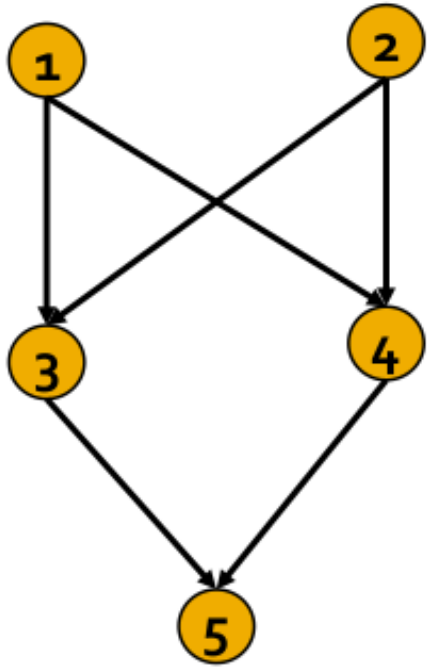
- **Structural equivalence:** Nodes u and v are structurally equivalent if they have the same relationships **to all other nodes** [Lorrain & White 1971]
  - Structurally equivalent nodes are likely to be similar in other ways – *i.e.*, friendships in social networks



Structural Equivalence

- Nodes u and v are **structurally equivalent:**
  - For all the other nodes k, node u has tie to k iff node v has tie to k

- **Example:**



Adjacency matrix

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

- *E.g.,* nodes 3 and 4 are structurally equivalent

Discovering Structural Roles

Why are roles important?

Task	Example Application
Role query	Identify individuals with similar behavior to a known target
Role outliers	Identify individuals with unusual behavior
Role dynamics	Identify unusual changes in behavior
Identity resolution	Identify/de-anonymize, individuals in a new network
Role transfer	Use knowledge of one network to make predictions in another
Network comparison	Compute similarity of networks, determine compatibility for knowledge transfer

War Story

Evolutionary Role Mining in Complex Networks by Ensemble Clustering

Sarvenaz Choobdar, Pedro Ribeiro, Fernando Silva
CRACS & INESC-TEC
DCC-FCUP, Universidade do Porto, Portugal
{sarvenaz,pribeiro,fds}@dcc.fc.up.pt

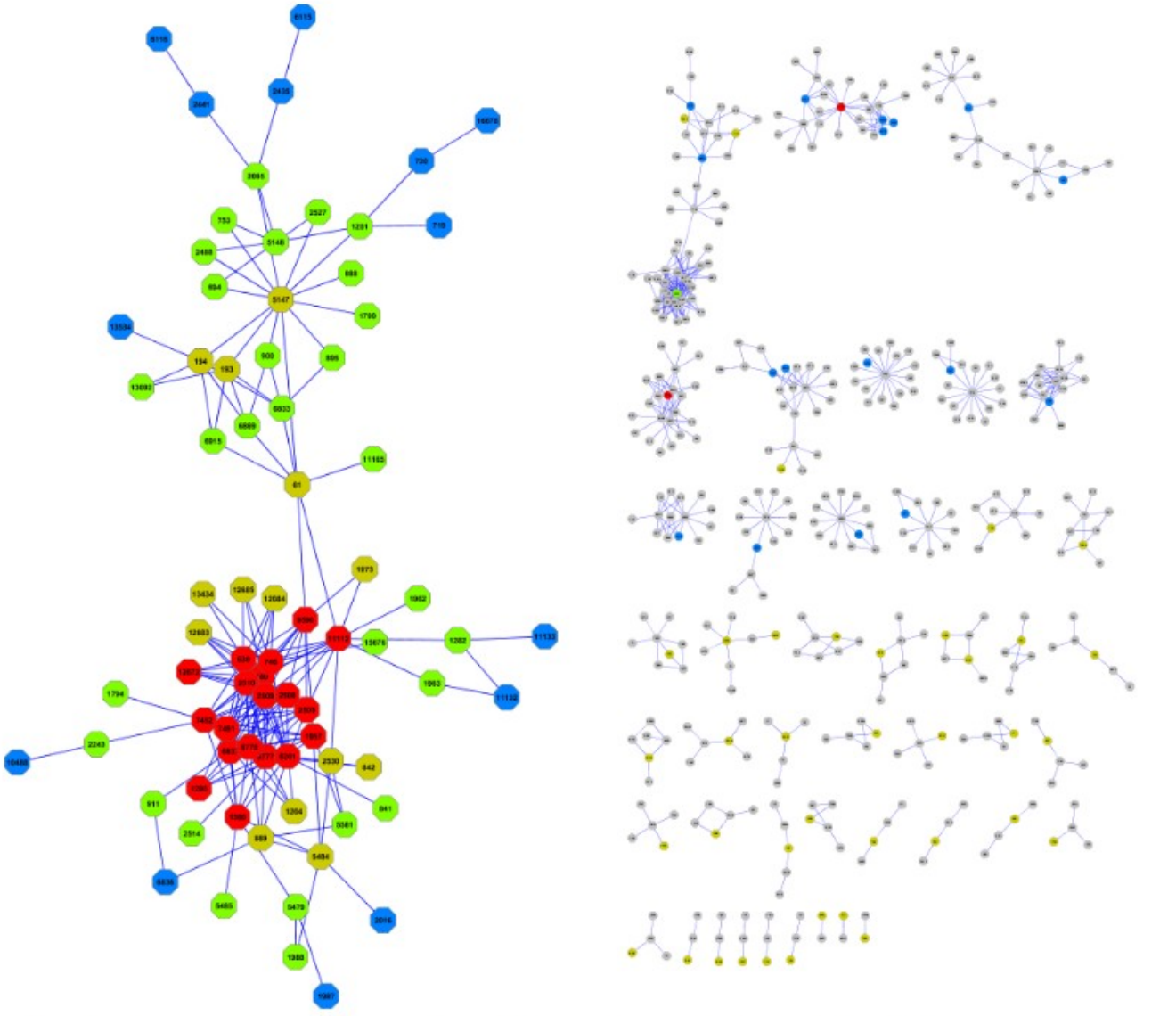
- the normalized node degree: quantifies the linkage of node i; it is the degree of node i divided by the sum of all nodes' degree in the network.
- the normalized average degree: shows the intensity of connectivity in the neighborhood of node i; it is calculated by averaging over all degree of immediate neighbors of node i.
- the coefficient variation of the degrees of the immediate neighbors of a node (cv): characterizes the coherence of the connectivity; it the standard deviation of the degrees in the neighborhood of node i.
- the clustering coefficient: quantifies the connectivity between neighbors; it is measured as the proportion of existing connections between neighbors of node i to the number of all possible links between them [25].
- the locality index: characterizes the structure of neighbors' connectivity to rest of the network; it is the ratio of links within the neighborhood to the number of links to the nodes outside of neighborhood.

Algorithm 1 Evolutionary Role Mining (ERM)

procedure ERM($G_T, K, wFun(C), clustAlgo(M, K)$)
  for t **in** $1:T$ **do**
    $X_t \leftarrow localProperties(G_t)$
    $C_t \leftarrow kmeans(X_t, K)$
    $C \leftarrow C \cup C_t$
  end for
  $\{\alpha_1, ..., \alpha_T\} \leftarrow wFun(C)$
  for t **in** $1:T$ **do**
    $M \leftarrow M + pairwiseSimilarity(G, C_t) * \alpha_t$
  end for
  $C_T \leftarrow clustAlgo(M, K)$
  return C_T
end procedure

Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Evolutionary Role Mining in Complex Networks by Ensemble Clustering. Proceedings of the 32nd ACM Symposium On Applied Computing - Social Network and Media Analysis Track (ACMSAC), pp. 1053-1060, ACM, Marrakech, Morocco, April, 2017.

War Story



(b) Color-code by role of nodes, identified by proposed method

War Story

Pairwise structural role mining for user categorization in information cascades

Sarvenaz Choobdar, Pedro Ribeiro, Fernando Silva
CRACS and INESC-TEC
University of Porto, Portugal
Email: {sarvenaz,prebeiro,fds}@dcc.fc.up.pt

***Abstract*—The tendency of users to connect with peers of similar interests and social demography (homophily) is one of the sources of information for user behavior modeling and classification. However this is yet an open question for structural roles where nodes at similar structural position in the network play the same roles: are structurally equivalent nodes more prone to have connections between themselves? In this paper, we tackle this open question by studying the patterns of homophily for structural roles. We propose a new method named SR-Diffuse to simultaneously identify structural roles in a network and to model the role membership matrix of users. In this method, we integrate pairwise role dependency alongside with structural features of users for role mining. We show that pairwise role dependency is necessary to distinguish some structural roles but it is a misleading factor for some others. We design an optimization model to capture structural roles with the guidance of pairwise dependency, and devise an iterative algorithm to learn structural roles simultaneously from structural properties and social dependency of users. We examine the efficacy of our new method in a users classification problem for information cascades. We compare the predictability of discovered roles by our method against some baseline methods for predicting social classes of users in different information cascades in two social networks, Flickr and Digg. The experimental results suggest that our method can improve the quality of roles membership of users and can better represent the profile of users in the network, hence it is a better predictor for social classes of users in an information cascade.**

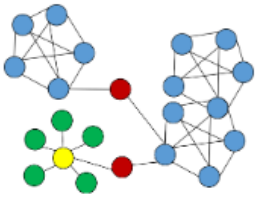


Fig. 1: Pairwise dependency across structural roles, different colors correspond to different structural roles; the pairwise role dependency exists in some structural roles such as member-of-clique (blue nodes) but it does not hold on some others such as member-of-star (green nodes).

structural position may have a tendency to have connections between themselves. Figure 1 exemplifies that, with the blue nodes (member-of-clique) having connections to other blue nodes. However, this it not the case for all types of structural roles. For instance, the green nodes (member-of-star) have no connections to other green nodes, as their structural features do not give origin to pairwise connections. In this paper, one of our main goals is to incorporate pairwise dependency of different structural roles in role mining framework. For that, we first examine how actually the pairwise relations are across

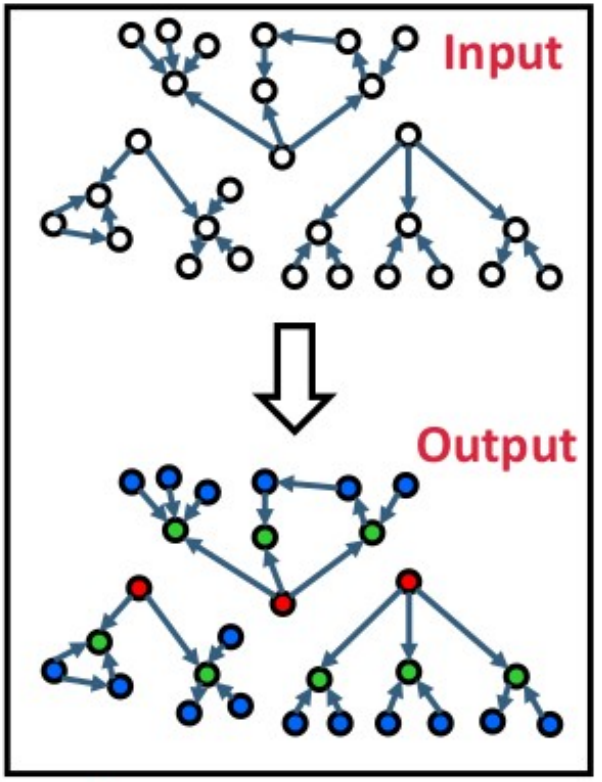
Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Pairwise structural role mining for user categorization in information cascades. Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 137-144, IEEE, Paris, France, August, 2015.

RolX

- **RolX:** Automatic discovery of nodes' structural roles in networks
  [Henderson, et al. 2011b]

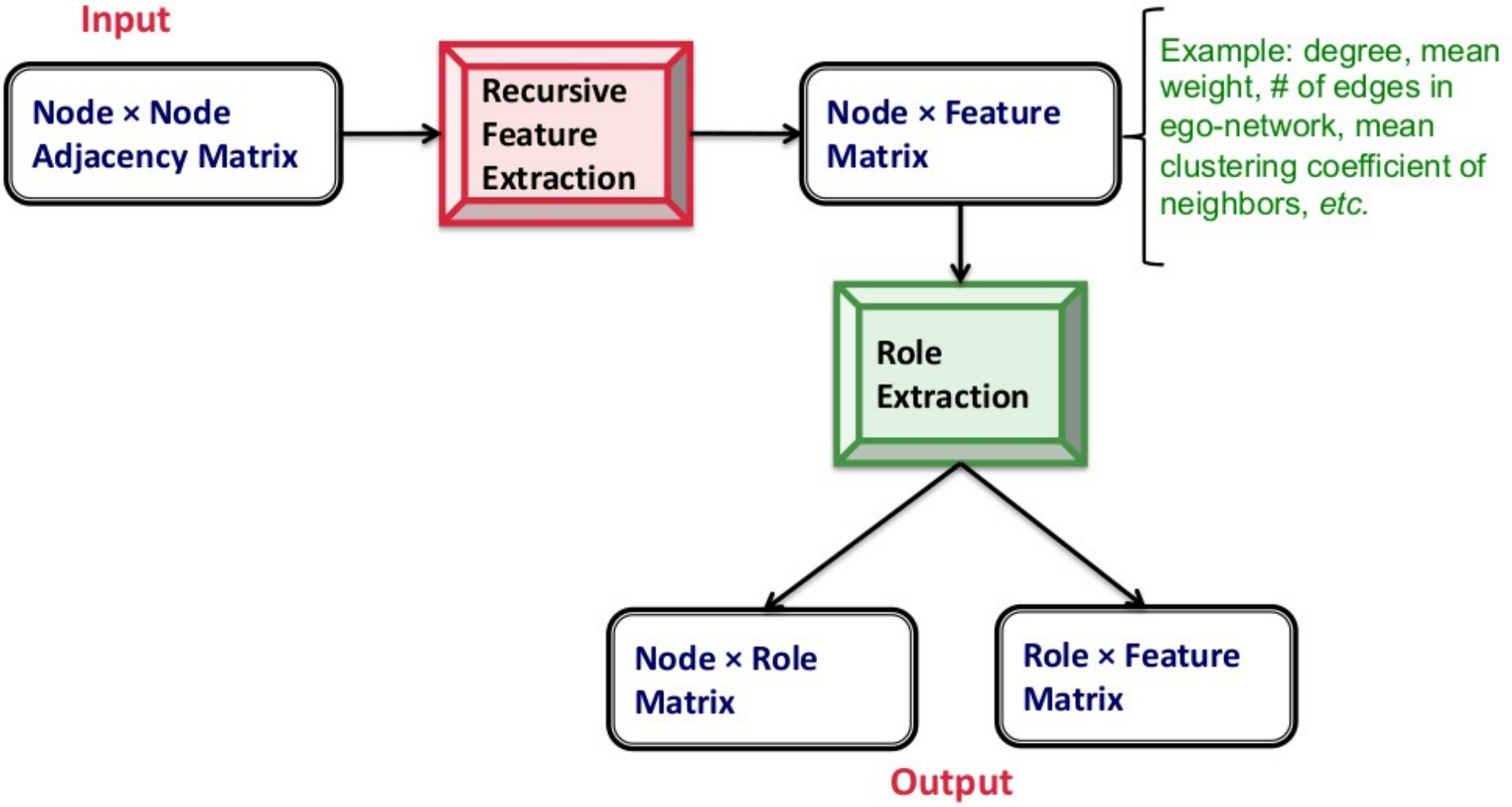
  - Unsupervised learning approach
  - No prior knowledge required
  - Assigns a mixed-membership of roles to each node
  - Scales linearly in #(edges)

Role Discovery



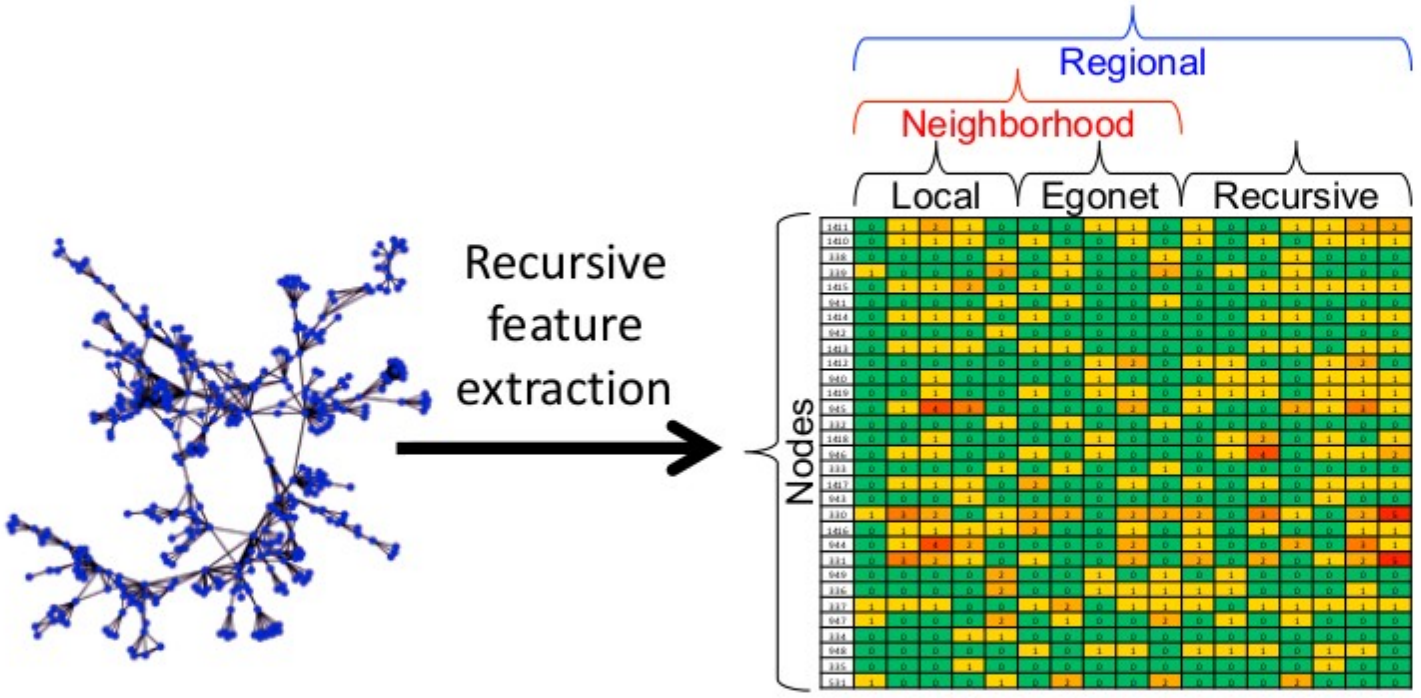
✓ Automated discovery
✓ Behavioral roles
✓ Roles generalize

RolX: Approach Overview



RolX: Recursive Feature Extraction

- **Recursive feature extraction** [Henderson, et al. 2011a] turns network connectivity into structural features

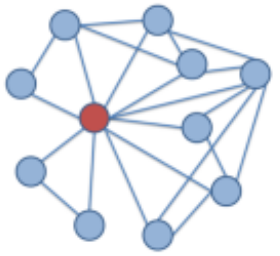


- **Neighborhood features:** What is a node's connectivity pattern?
- **Recursive features:** To what **kinds** of nodes is a node connected?

RolX: Recursive Feature Extraction

- **Idea:** Aggregate features of a node and use them to **generate new recursive features**

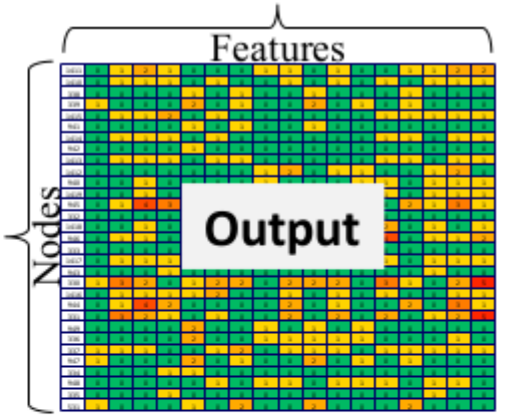
- **Base set of a node's neighborhood features:**
  - **Local features:** All measures of the node degree:
    - If network is directed, include in- and out-degree, total degree
    - If network is weighted, include weighted feature versions
  - **Egonetwork features:** Computed on the node's egonet:
    - **Egonet** includes the node, its neighbors, and any edges in the induced subgraph on these nodes
    - #(within-egonet edges), #(edges entering/leaving egonet)



Egonet for **red node**

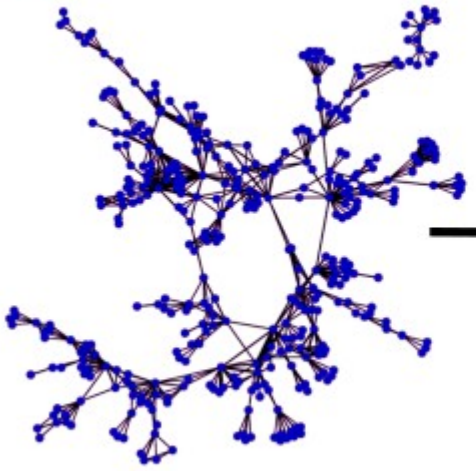
RolX: Recursive Feature Extraction

- Start with the base set of node features
- Use the **set of current node features** to generate **additional features**:
  - Two types of **aggregate functions:** means and sums
    - *E.g.,* mean value of "unweighted degree" feature among all neighbors of a node
    - Compute means and sums over all current features, including other recursive features
  - Repeat
- The number of possible recursive features **grows exponentially** with each recursive iteration:
  - Reduce the number of features using a **pruning technique:**
    - Look for pairs of features that are highly correlated
    - Eliminate one of the features whenever two features are correlated above a user-defined threshold

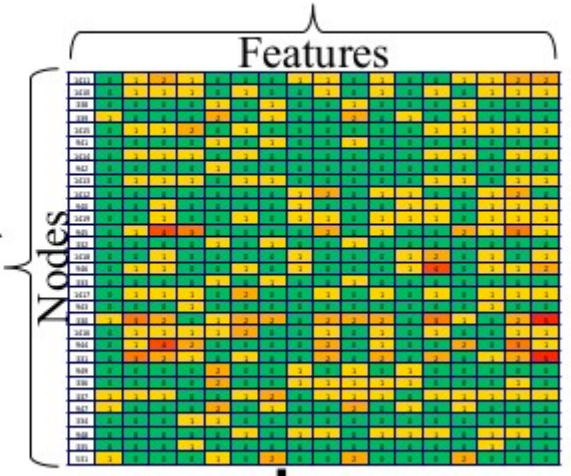


RolX: Role Extraction

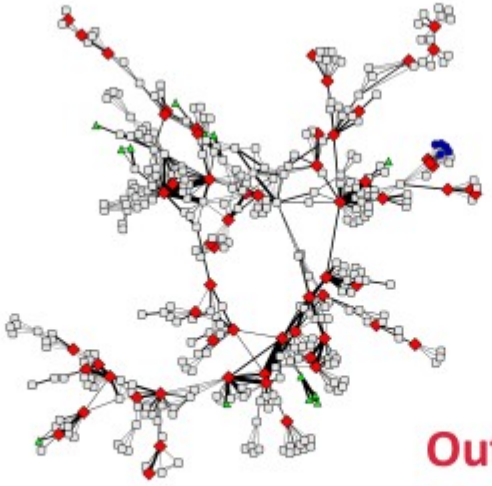
Input



Recursively extract features



1) Can compare nodes based on their structural similarity
2) Can cluster nodes to identify different structural roles



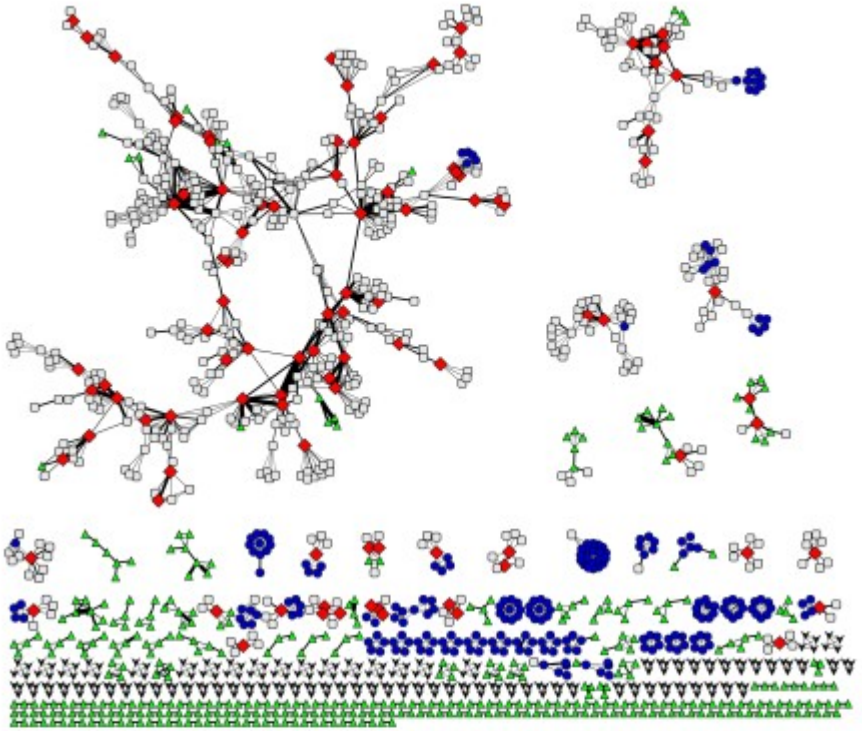
Output

e.g, RolX uses a clustering technique called non-negative matrix factorization

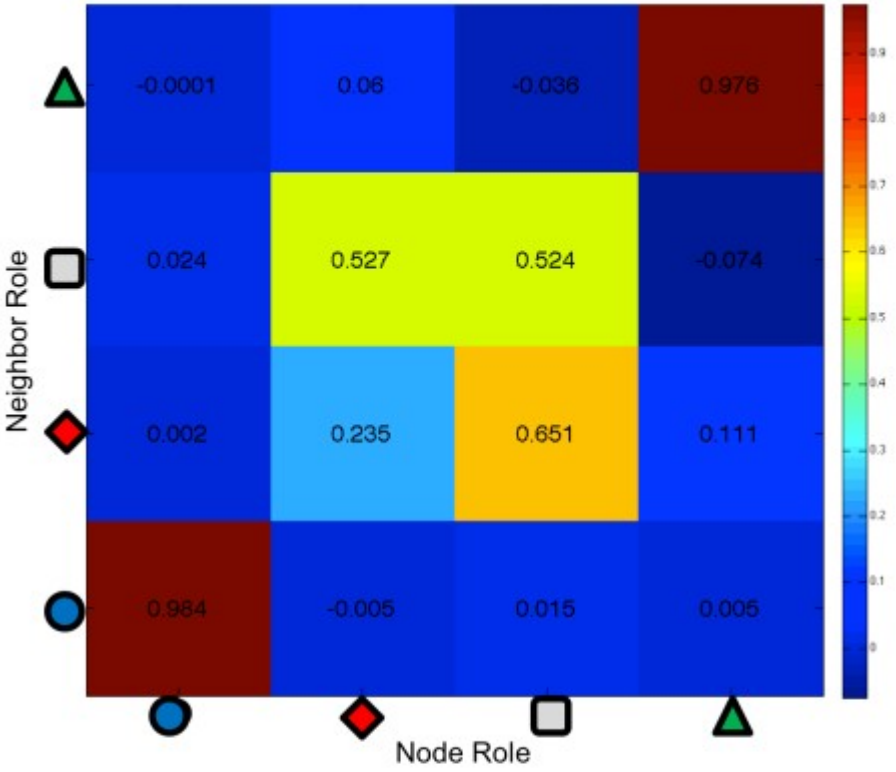
Application: Structural Similarity

- **Task:** Cluster nodes based on their structural similarity
- **Two networks:**
  - Network science co-authorship network:
    - Nodes: Network scientists; Edges: The number of co-authored papers
  - Political books co-purchasing network:
    - Nodes: Political books on Amazon; Edges: Frequent co-purchasing of books by the same buyers
- **Setup:** For each network:
  - Use RolX to assign each node a distribution over the set of discovered, structural roles
  - Determine similarity between nodes by comparing their role distributions

Structural Sim: Co-Authorsip



Role-colored graph: each node is colored by the primary role that RolX finds



Role affinity heat-map

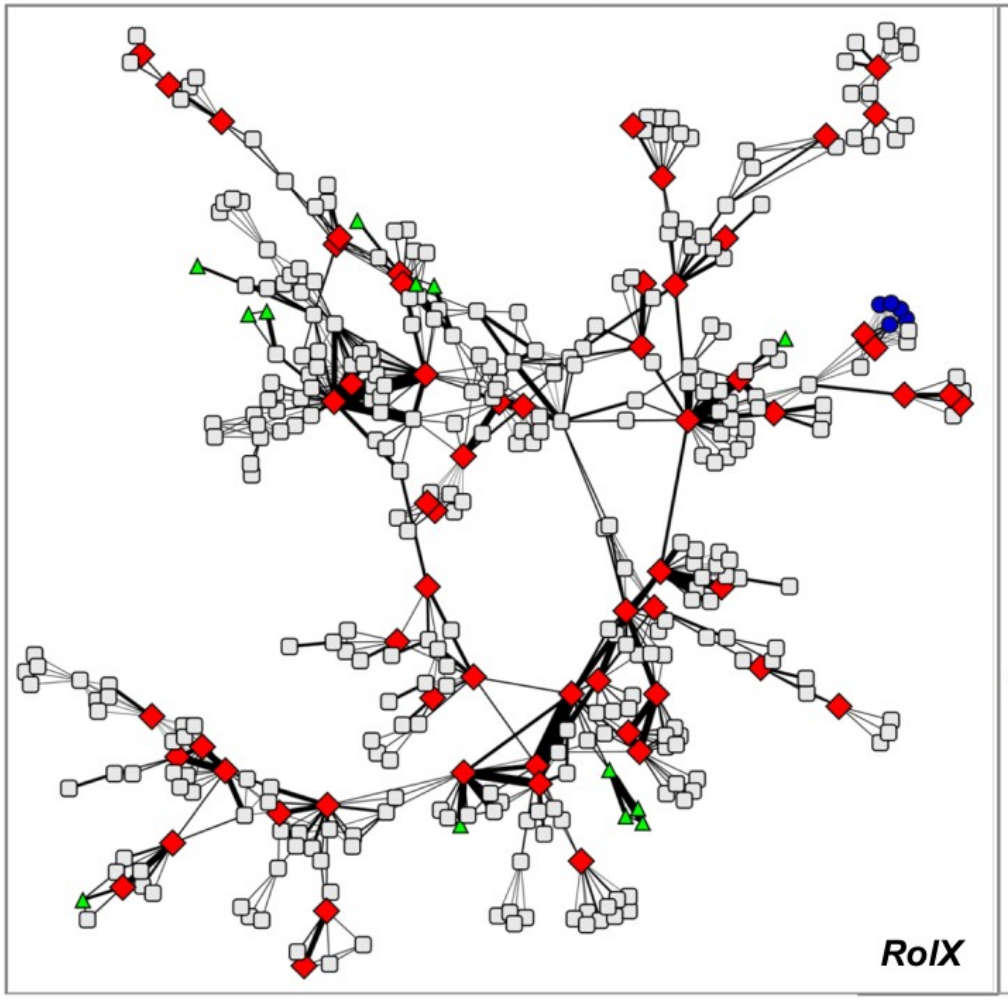
Making sense of roles:

- **Blue circle: Tightly knit**, nodes that participate in tightly-coupled groups
- **Red diamond: Bridge nodes**, that connect groups of nodes
- **Gray rectangle: Main-stream**, most of nodes, neither a clique, nor a chain
- **Green triangle: Pathy**, nodes that belong to elongated clusters

Community Structure

Roles and Communities

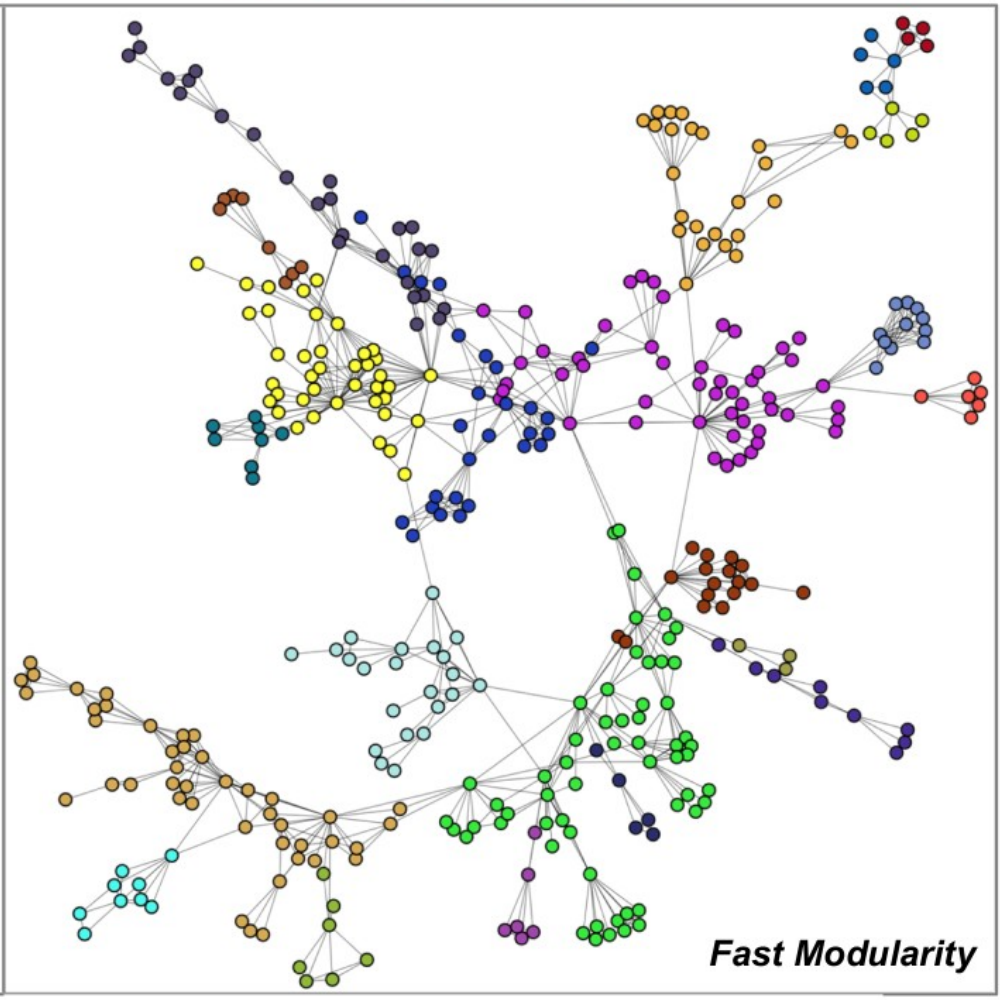
Roles



Henderson, *et al.*, KDD 2012

Nodes with different structural roles (connector node, bridge node, etc.)

Communities

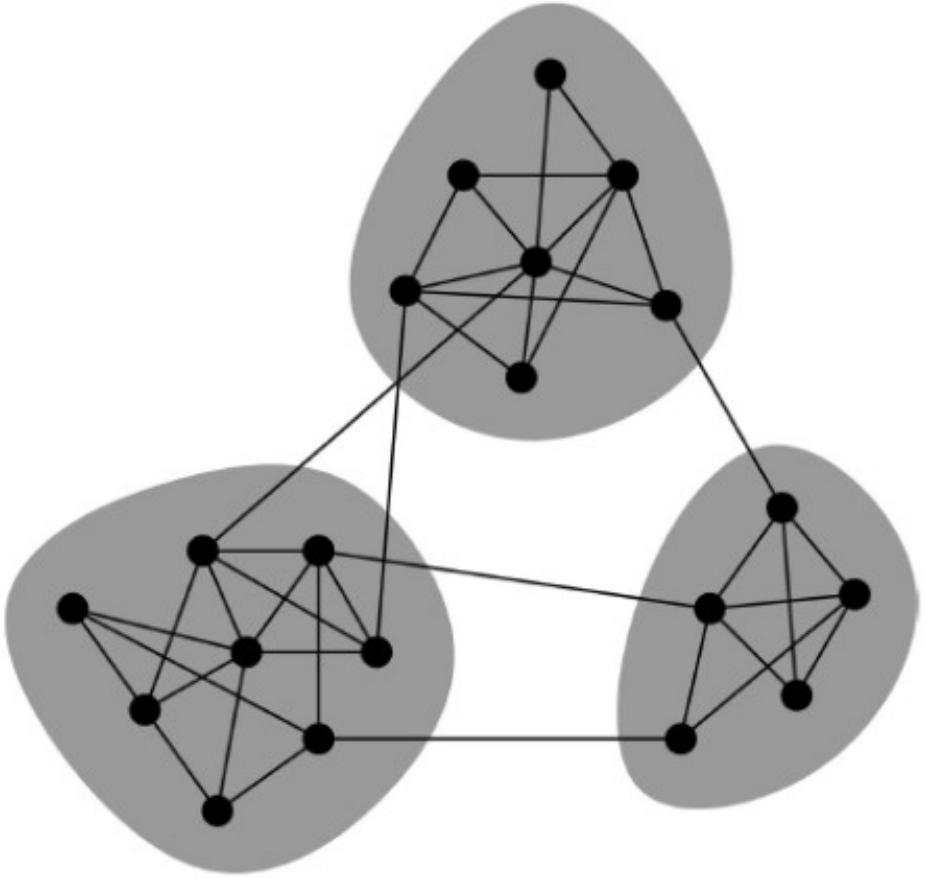


Clauset, *et al.*, Phys. Rev. E 2004

Nodes belonging to the same cluster/community

Networks and Communities

- We often think of networks "looking" like this:



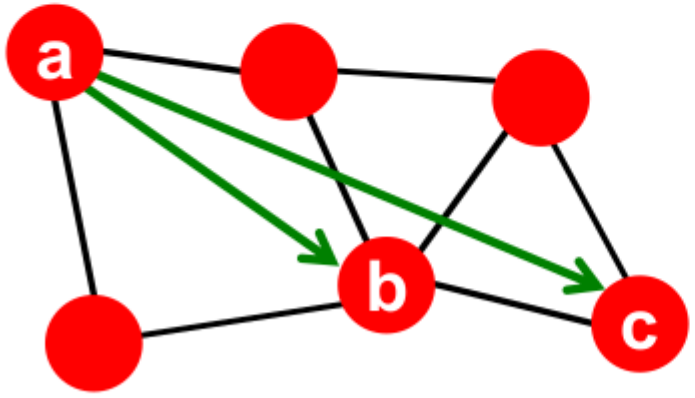
- What led to such a conceptual picture?

Networks: Flow of Information

- **How does information flow through the network?**
  - **What structurally distinct roles do nodes play?**
  - **What roles do different links ("short" vs. "long") play?**
- **How do people find out about new jobs?**
  - Mark Granovetter, part of his PhD in 1960s
  - People find the information through personal contacts
- **But:** Contacts were often **acquaintances** rather than close friends
  - **This is surprising:** One would expect your friends to help you out more than casual acquaintances
- **Why is it that acquaintances are most helpful?**

Granovetter's Answer

- **Two perspectives on friendships:**
  - **Structural:** Friendships span different parts of the network
  - **Interpersonal:** Friendship between two people is either **strong** or **weak**
- **Structural role: Triadic Closure**

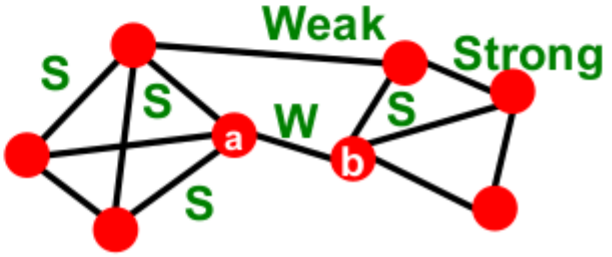


Which edge is more likely, a-b or a-c?

If two people in a network have a friend in common, then there is an increased likelihood they will become friends themselves.

Granovetter's Explanation

- **Granovetter makes a connection between social and structural role of an edge**
- **First point: Structure**
  - Structurally embedded edges are also socially strong
  - Long-range edges spanning different parts of the network are socially weak
- **Second point: Information**
  - Long-range edges allow you to gather information from different parts of the network and get a job
  - Structurally embedded edges are heavily redundant in terms of information access

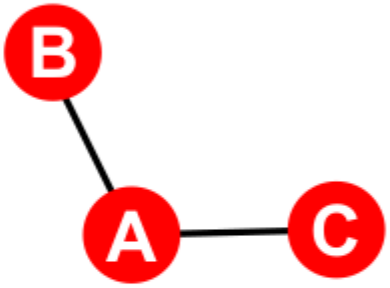


Triadic Closure

- **Triadic closure == High clustering coefficient**

Reasons for triadic closure:

- If B and C have a friend A in common, then:
  - B is **more likely to meet** C
    - (since they both spend time with A)
  - B and C **trust** each other
    - (since they have a friend in common)
  - A has **incentive** to bring B and C together
    - (since it is hard for A to maintain two disjoint relationships)



- **Empirical study by Bearman and Moody:**
  - Teenage girls with low clustering coefficient are more likely to contemplate suicide

Tie Strength in Real Data

- **For many years Granovetter's theory was not tested**
- But, today we have large who-talks-to-whom graphs:
  - Email, Messenger, Cell phones, Facebook
- **Onnela et al. 2007:**
  - Cell-phone network of 20% of country's population
  - **Edge strength:** # phone calls

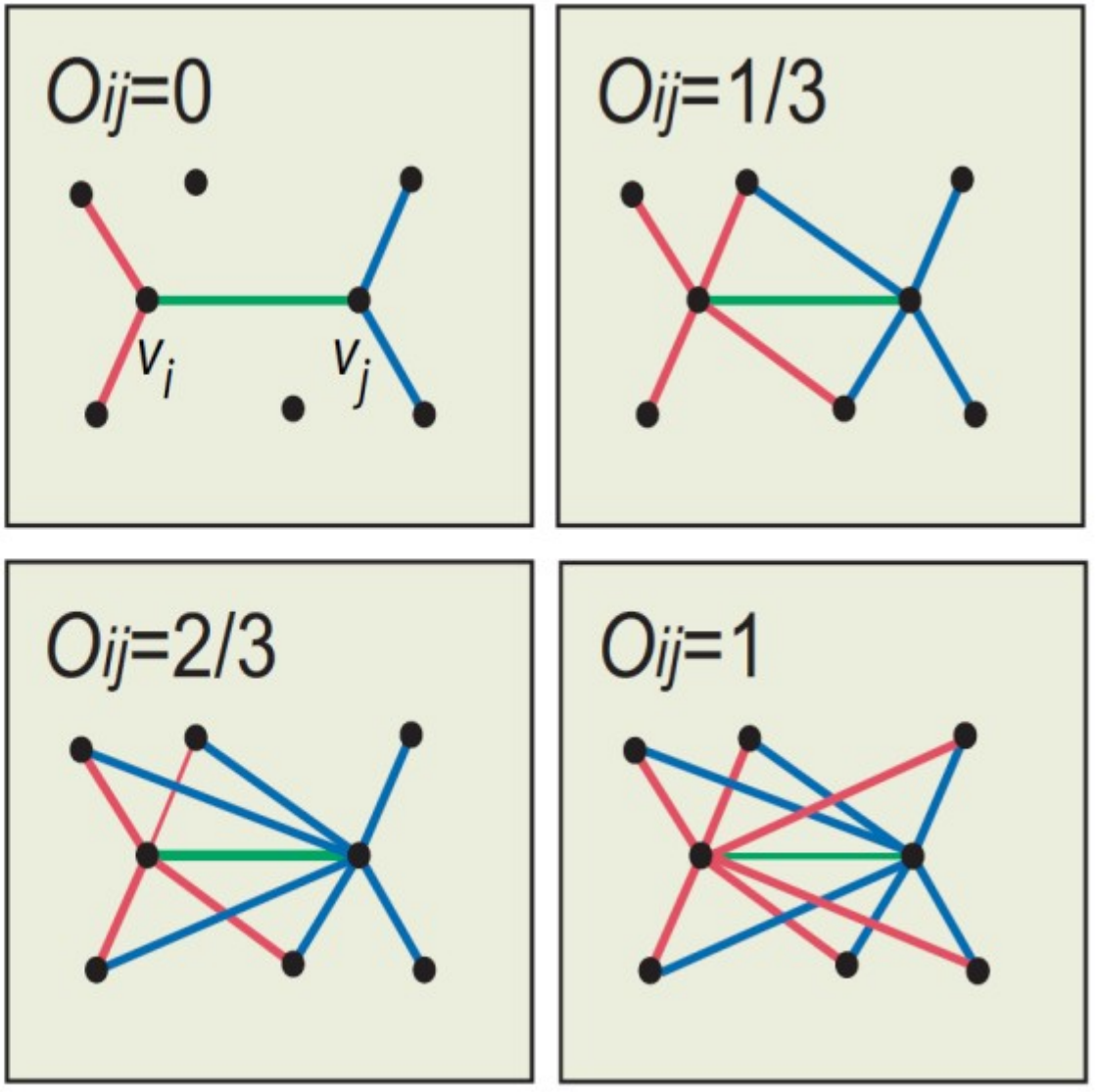
Neighborhood Overlap

- **Edge overlap:**

$$O_{ij} = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$$

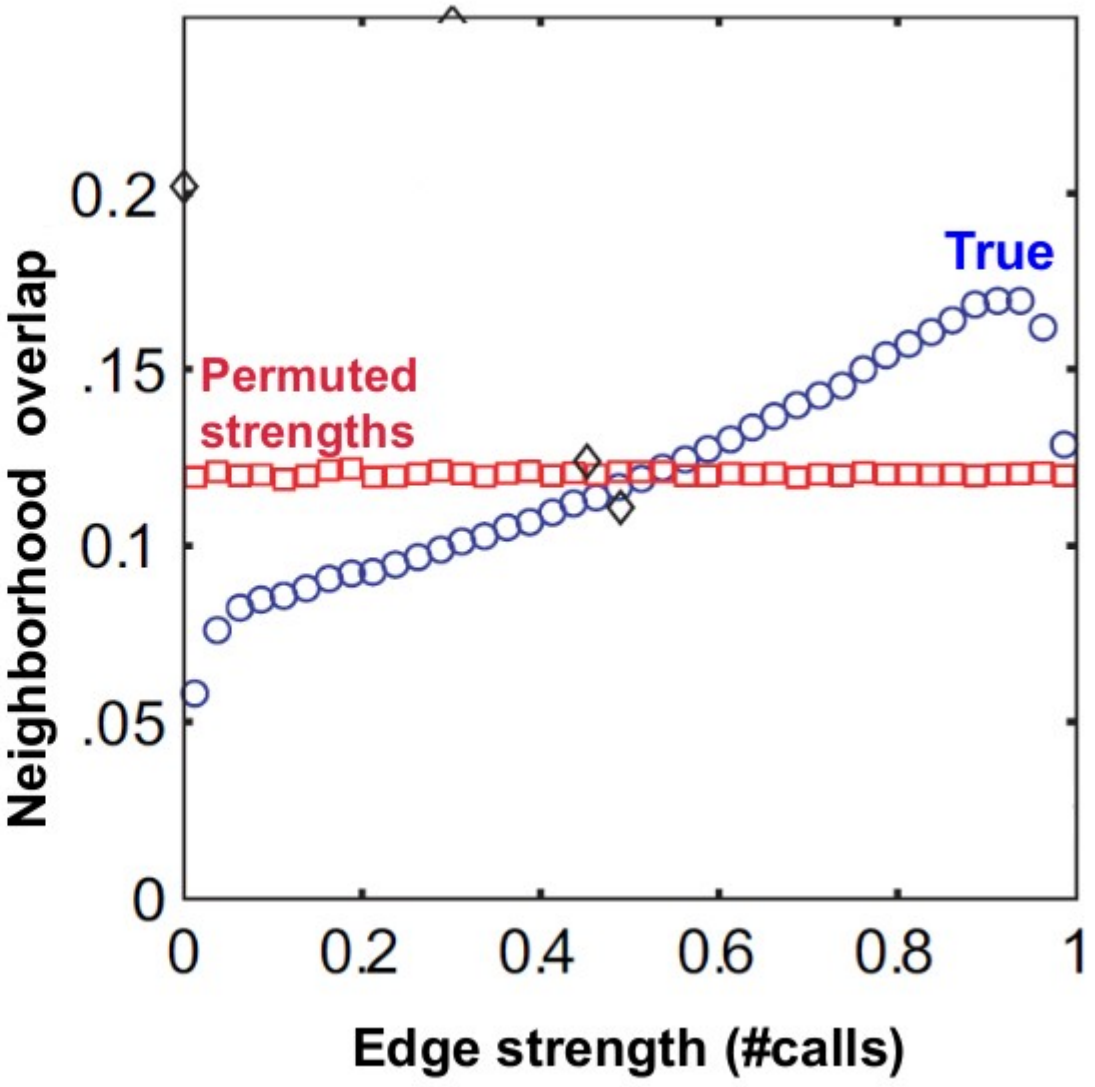
  - $N(i)$ ... a set of neighbors of node i

- **Overlap = 0** when an edge is a **local bridge**

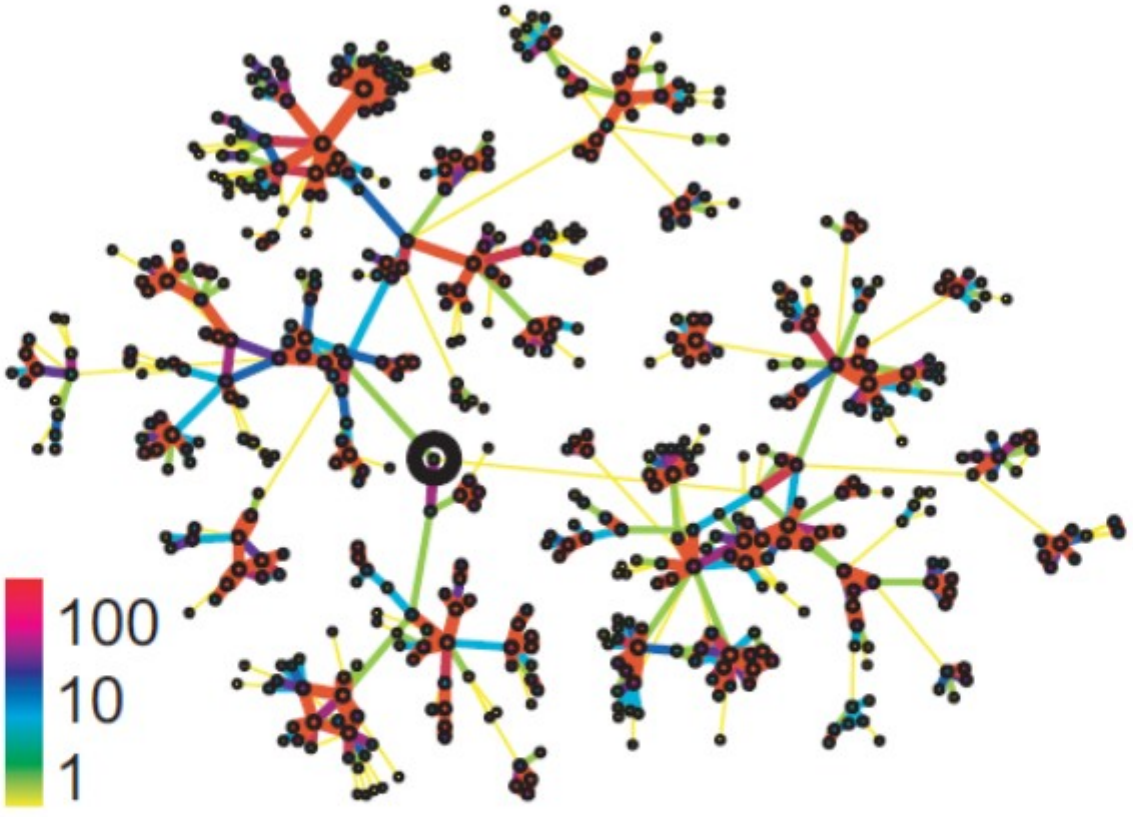


Phones: Edge Overlap vs Strength

- Cell-phone network
- Observation:
  - Highly used links have high overlap!
- Legend:
  - True: The data
  - Permuted strengths: Keep the network structure but randomly reassign edge strengths

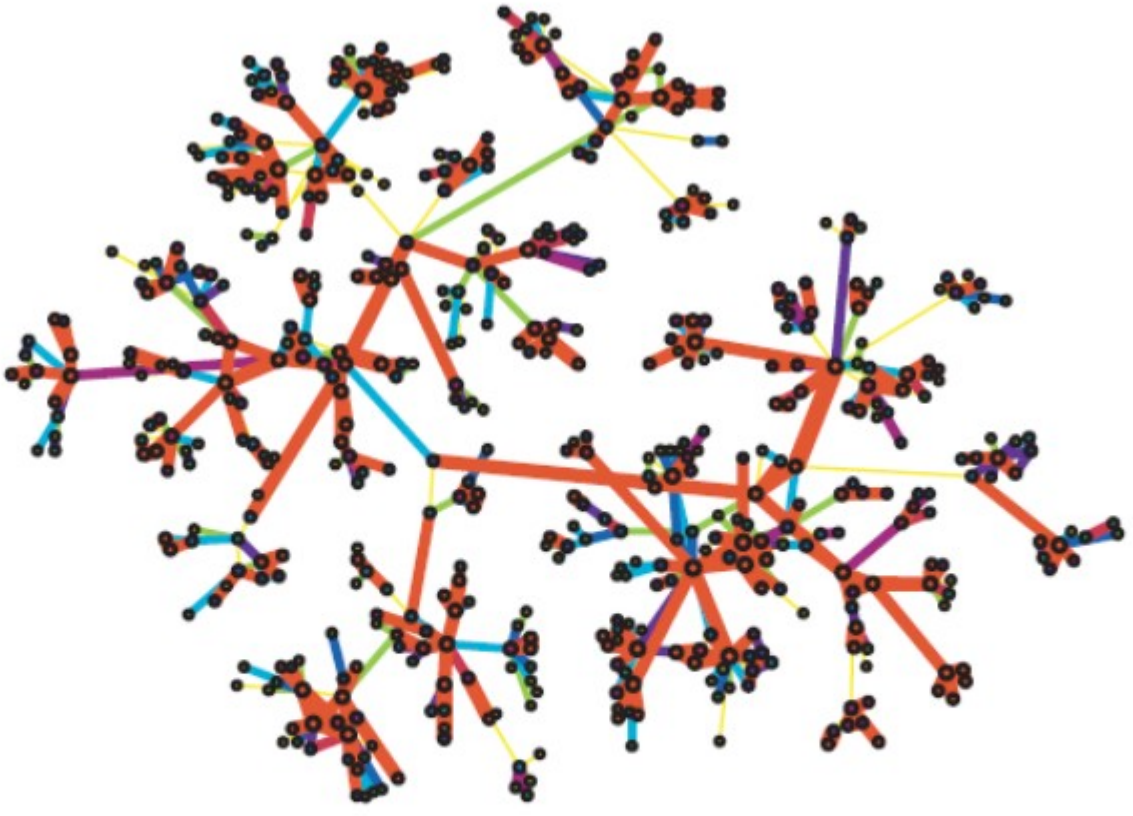


Real Net, Real Tie Strenghts



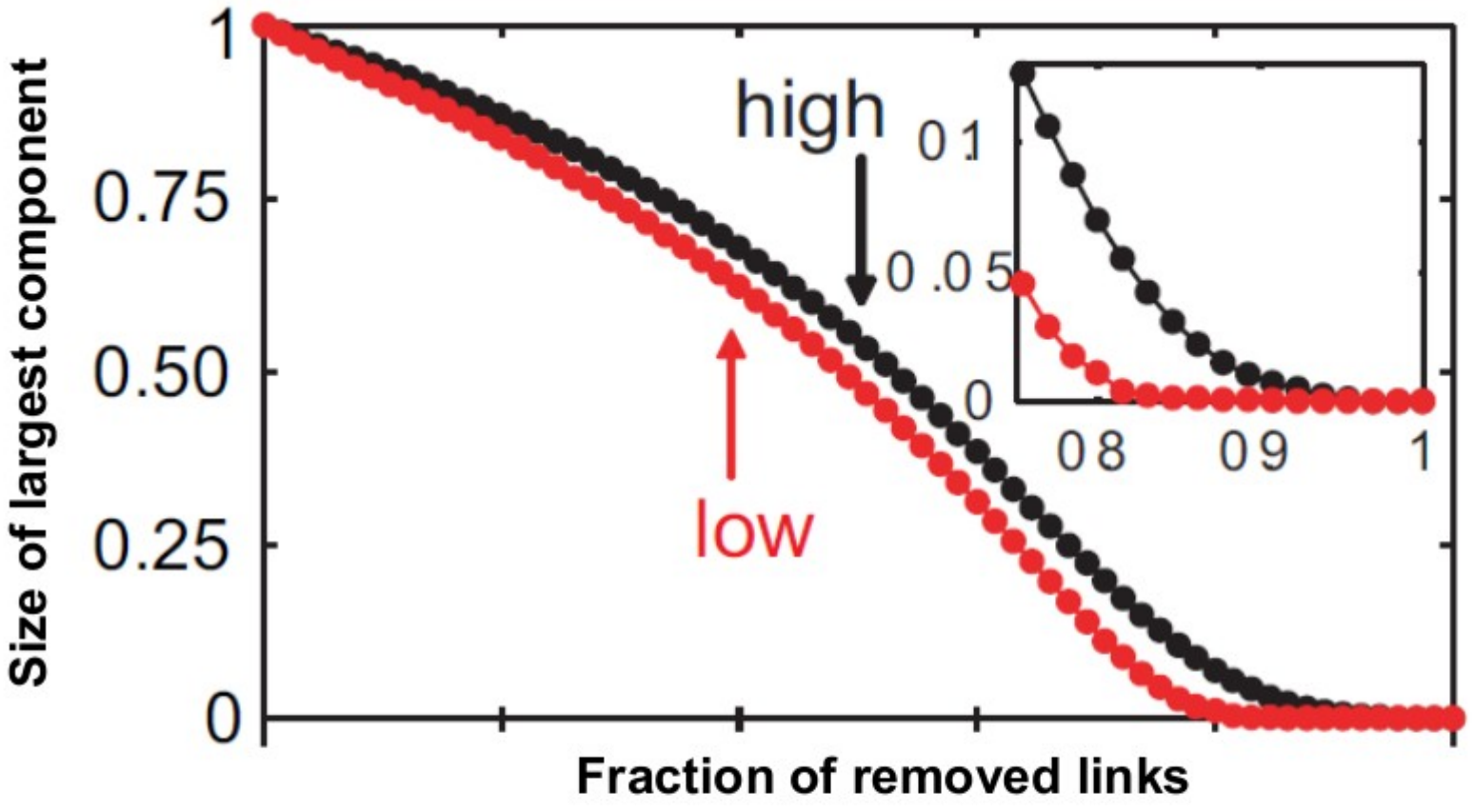
- **Real edge strengths in mobile call graph**
  - Strong ties are more embedded (have higher overlap)

Real Net, Permuted Tie Strenghts



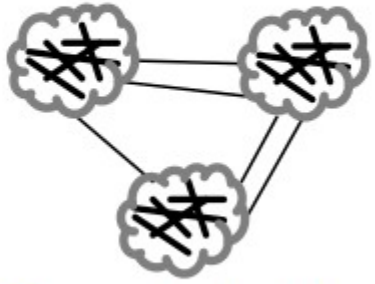
- **Same network, same set of edge strengths**
  but now **strengths are randomly shuffled**

Link Removal by Strength



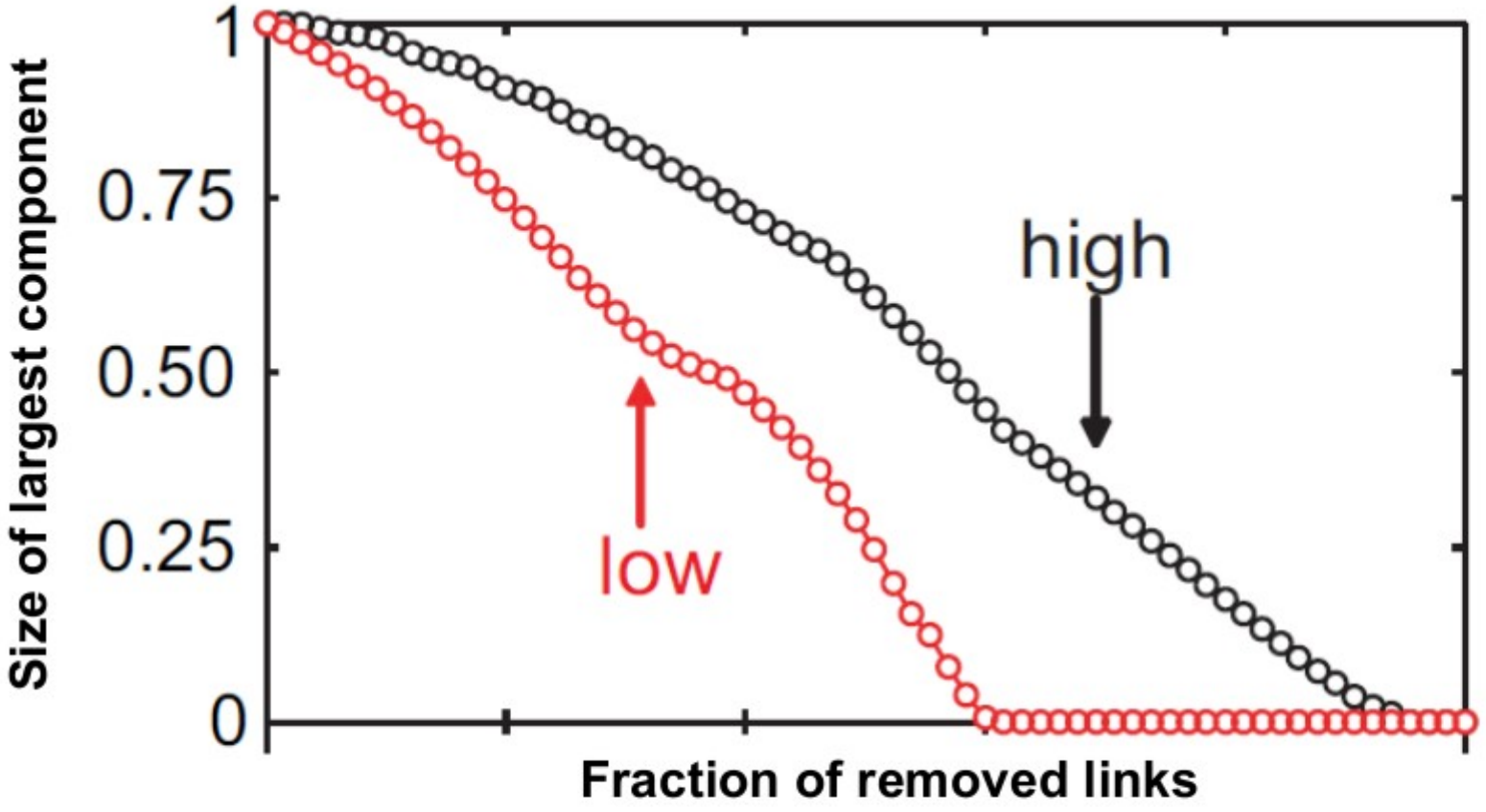
Low disconnects the network sooner

- Removing links by **strength (#calls)**
  - Low to high
  - High to low



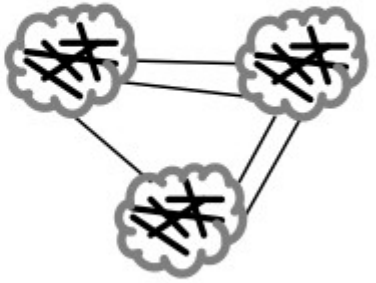
Conceptual picture of network structure

Link Removal by Overlap



Low disconnects the network sooner

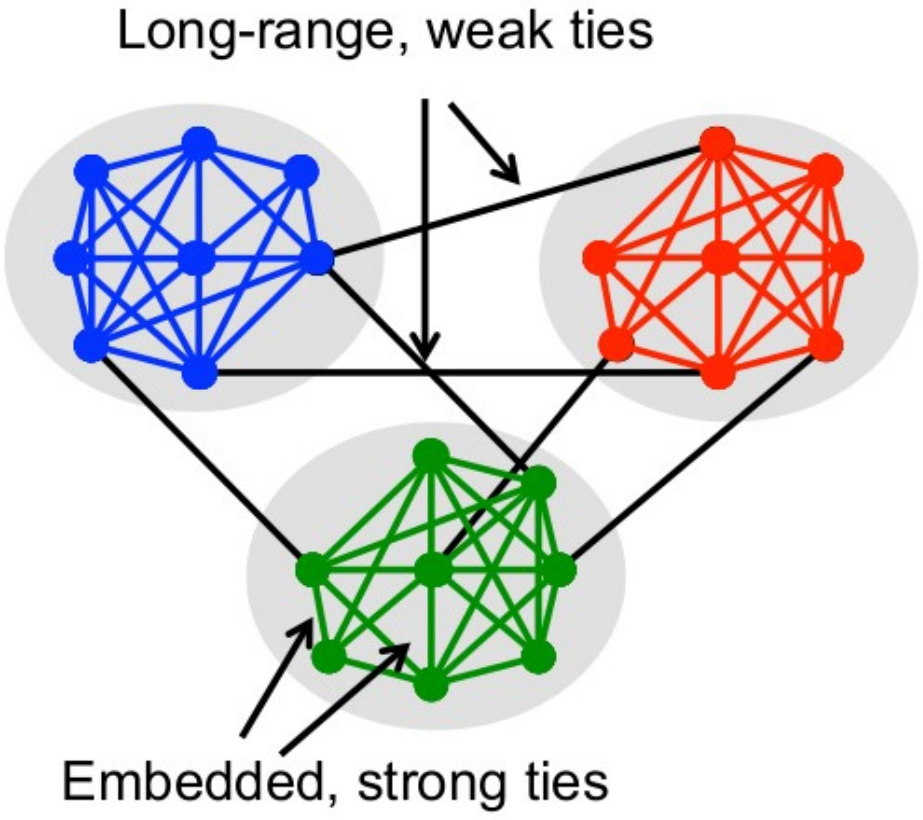
- Removing links based on **overlap**
  - Low to high
  - High to low

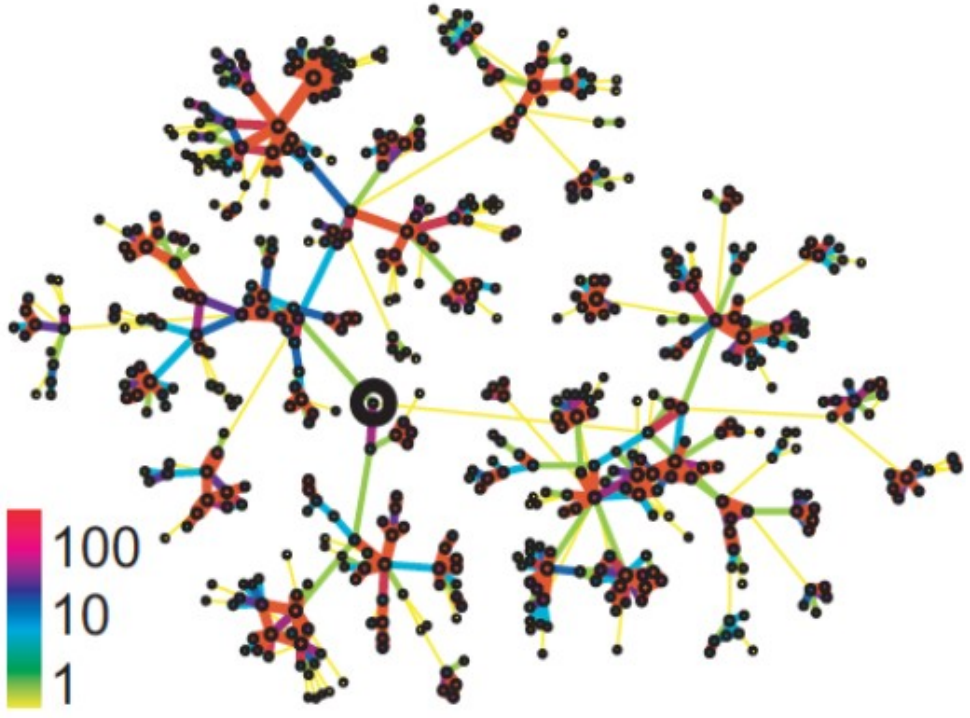


Conceptual picture of network structure

Closing the Loop

- We often think of (social) networks as having the following structure



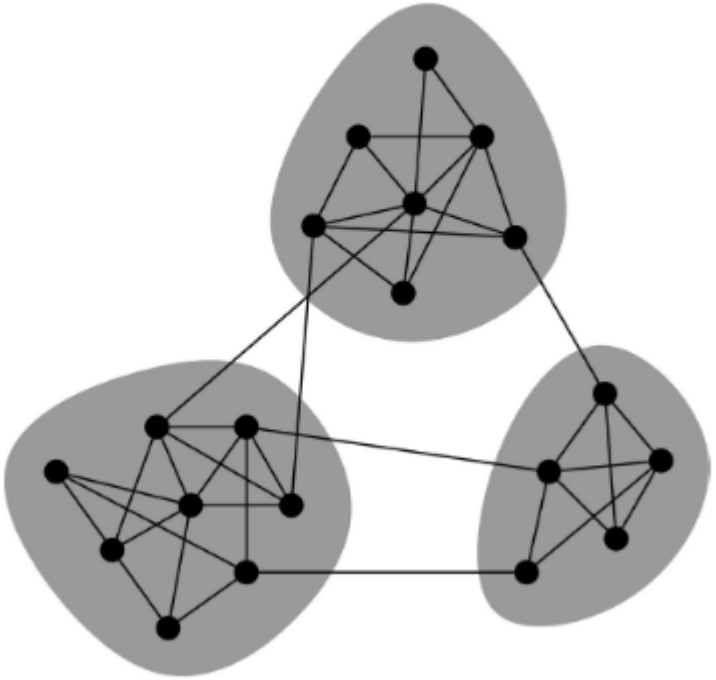


- Conceptual picture supported by Granovetter's strength of weak ties

Network Communities

Network Communities

- Granovetter's theory suggest that networks are composed of **tightly connected sets of nodes**

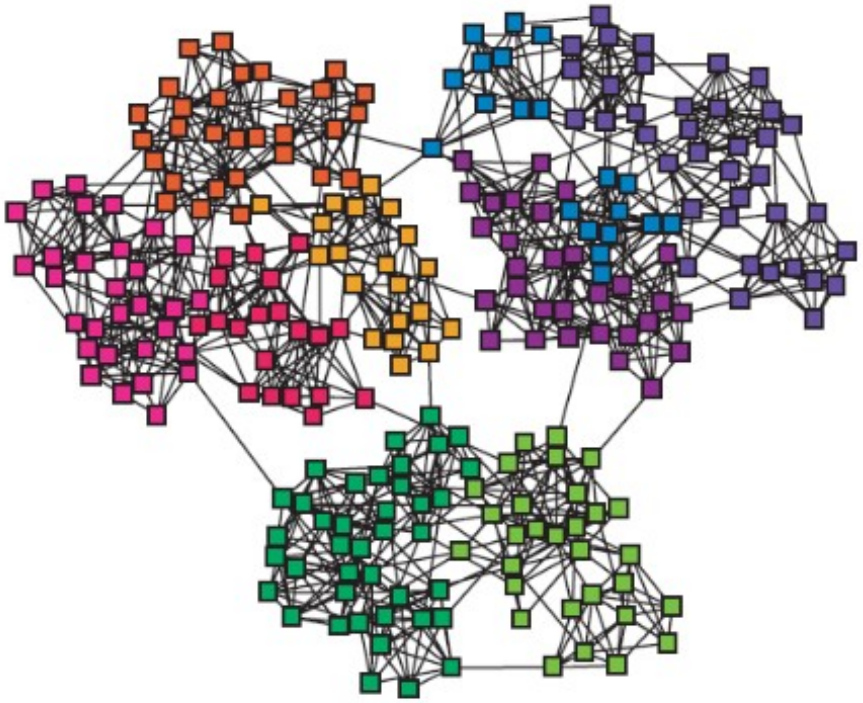


Communities, clusters, groups, modules

- **Network communities:**
  - Sets of nodes with **lots** of **internal** connections and **few external** ones (to the rest of the network).

Finding Network Communities

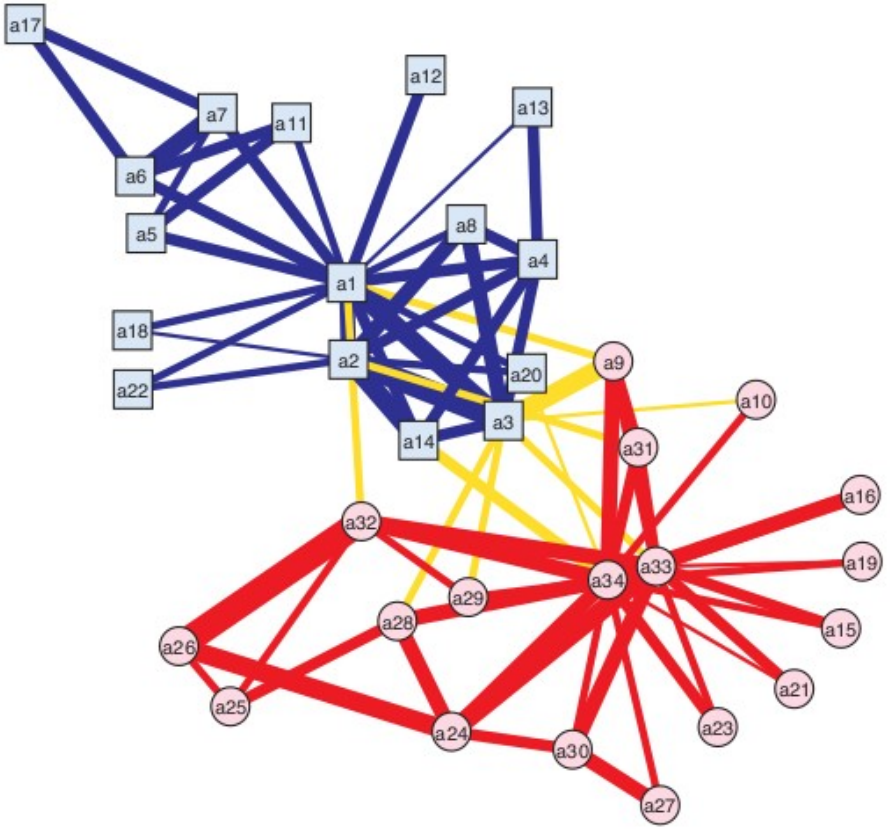
- **How to automatically find such densely connected groups of nodes?**
- Ideally such automatically detected clusters would then correspond to real groups
- **For example:**



Communities, clusters, groups, modules

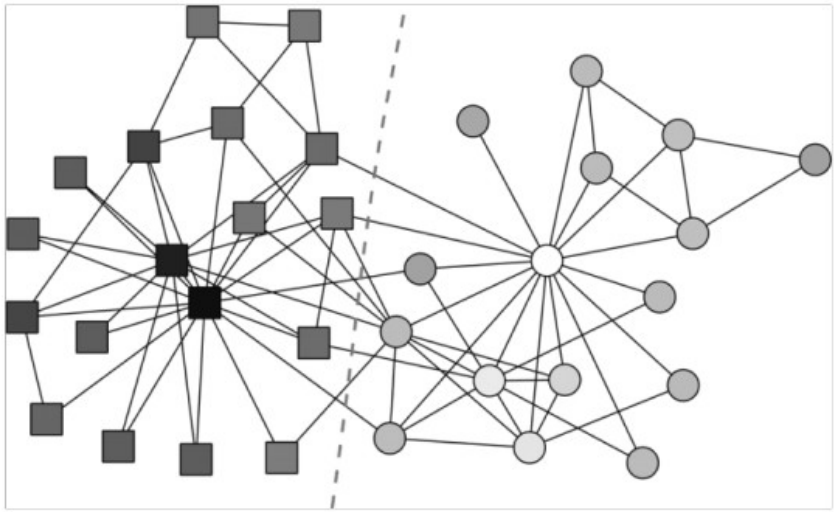
Zachary's karate club

- Social interactions among members of a karate club in the 70s

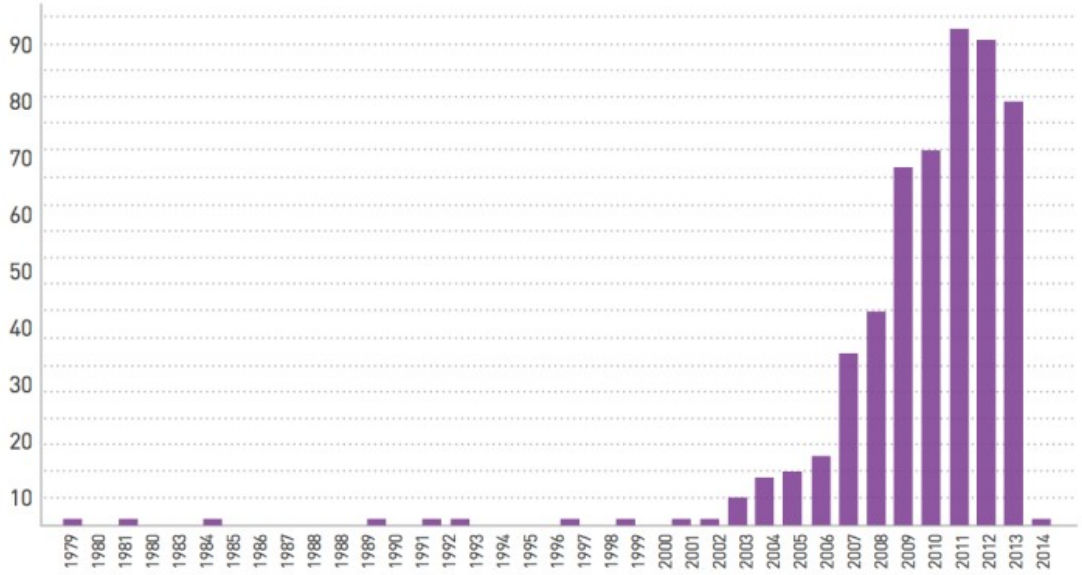


- Zachary witnessed the club split in two during his study
  - ⇒ Toy network, yet canonical for community detection algorithms
  - ⇒ Offers "ground truth" community membership (a rare luxury)

Zachary's karate club



Citation history
of the Zachary's Karate club paper



Zachary’s karate club Club!

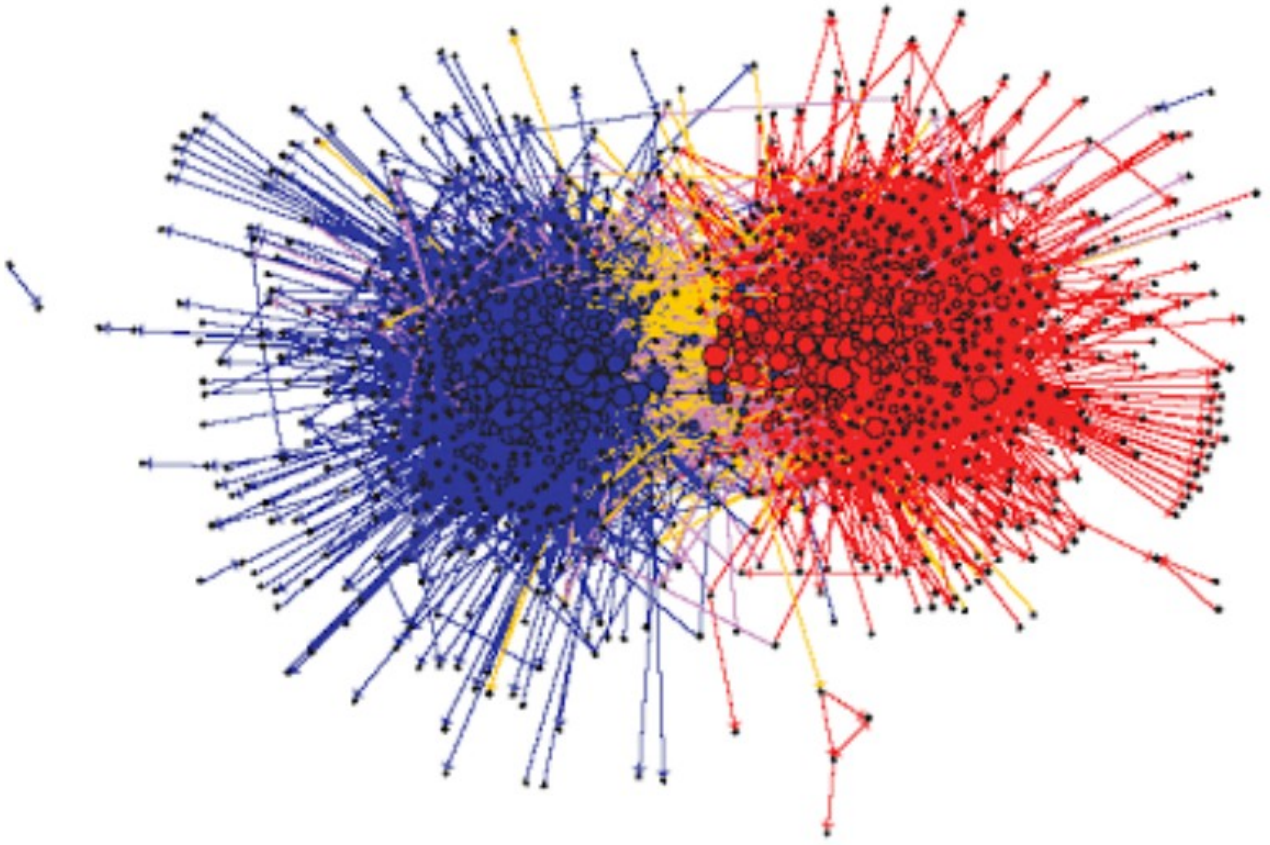
The first scientist at any conference on networks who uses Zachary's karate club as an example is inducted into the Zachary Karate Club Club, and awarded a prize.

Chris Moore (9 May 2013).
Mason Porter (NetSci, June 2013).
Yong-Year Ahn (Oxford University, July 2013)
Marián Boguñá (ECCS, September 2013).
Mark Newman (Netsci, June 2014)

http://networkkarate.tumblr.com/)

Political blogs

- The political blogosphere for the US 2004 presidential election



- Community structure of liberal and conservative blogs is apparent
  - $\Rightarrow$ People have a stronger tendency to interact with "equals"

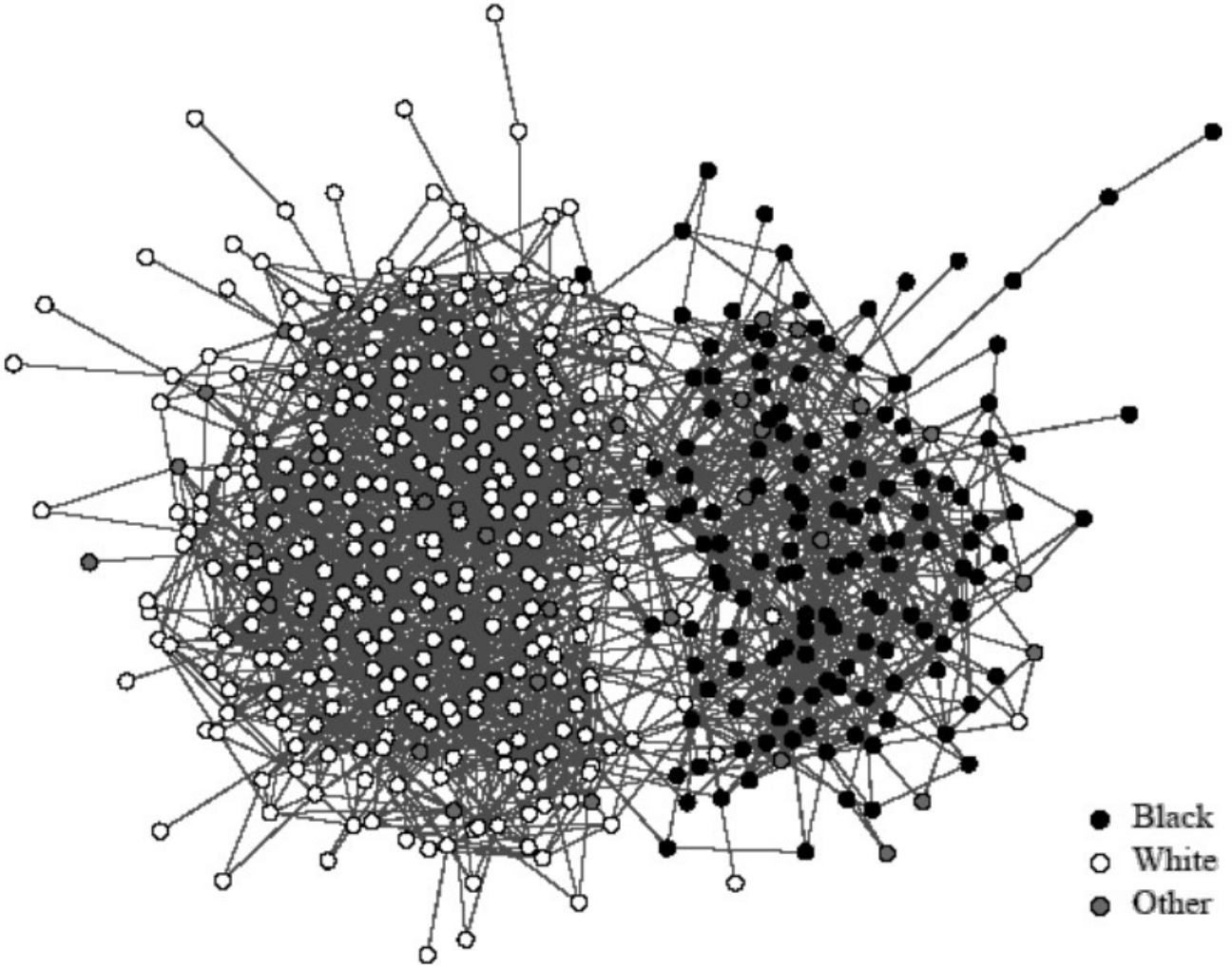
Electrical power grid

- Split power network into areas with minimum inter-area interactions

- **Applications:**
  - Decide control areas for distributed power system state estimation
  - Parallel computation of power flow
  - Controlled islanding to prevent spreading of blackouts

High-school students

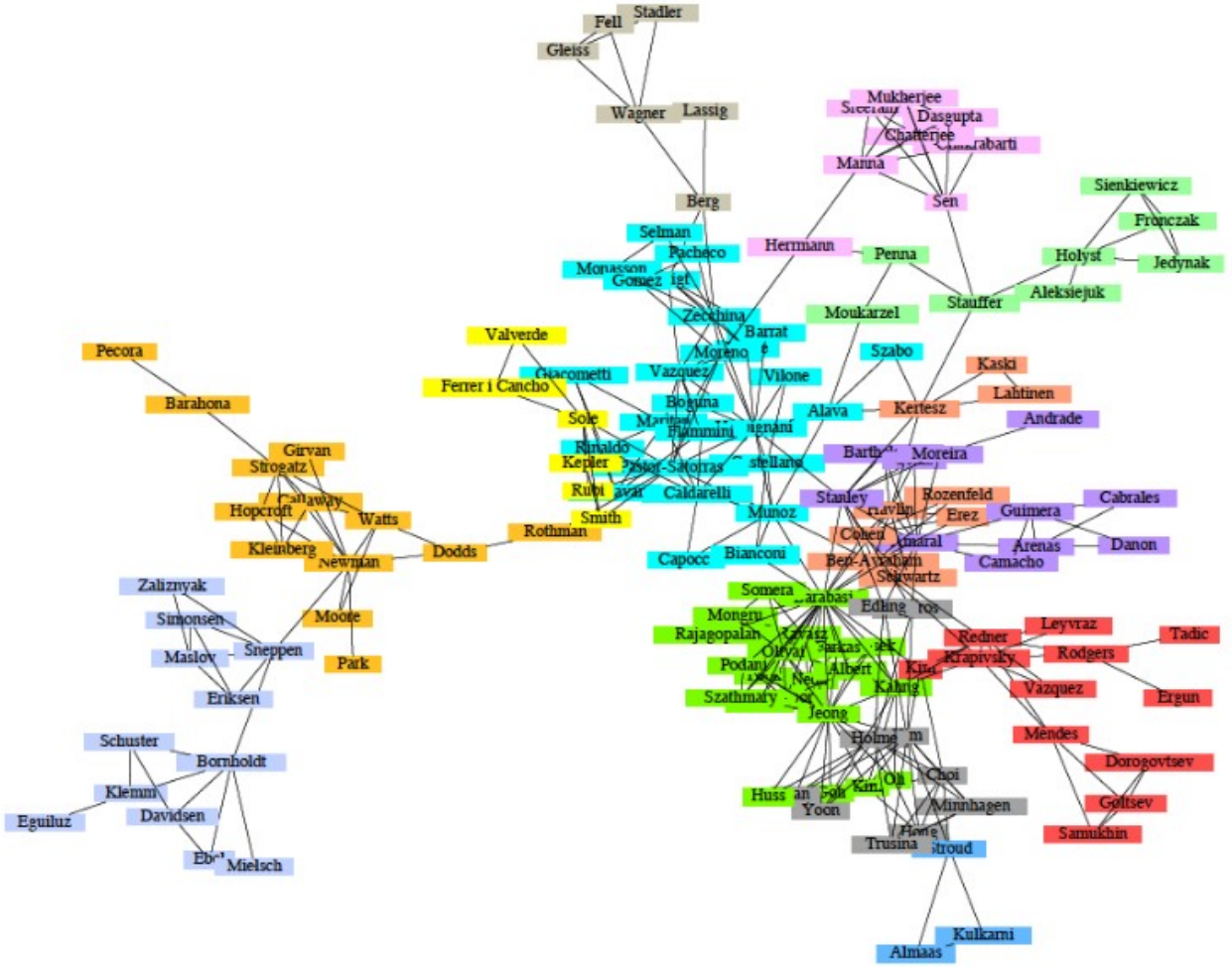
- Network of social interactions among high-school students



- Strong assortative mixing, with race as latent characteristic

Physicists working on NetSci

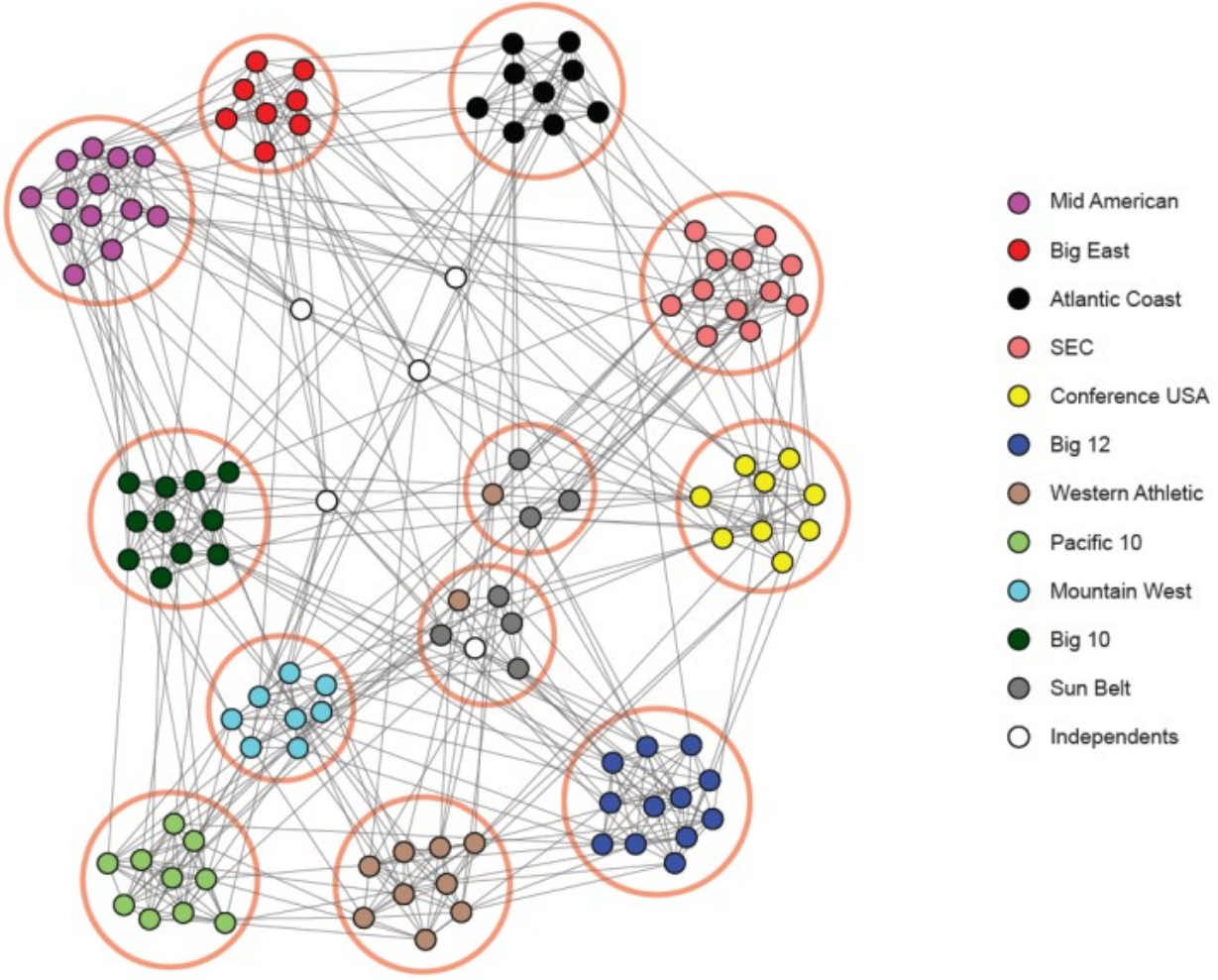
- Coauthorship network of physicists publishing networks' research



- Tightly-knit subgroups are evident from the network structure

College football

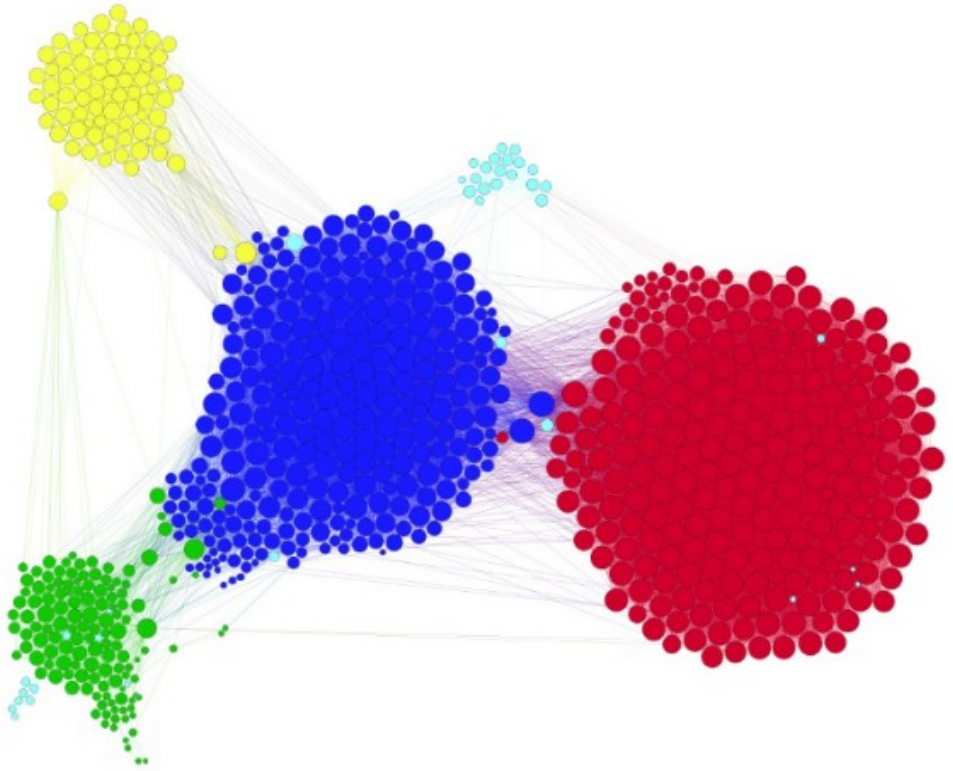
- Vertices are NCAA football teams, edges are games during Fall'00



- Communities are the NCAA conferences and independent teams

Facebook friendships

- Facebook egonet with 744 vertices and 30K edges

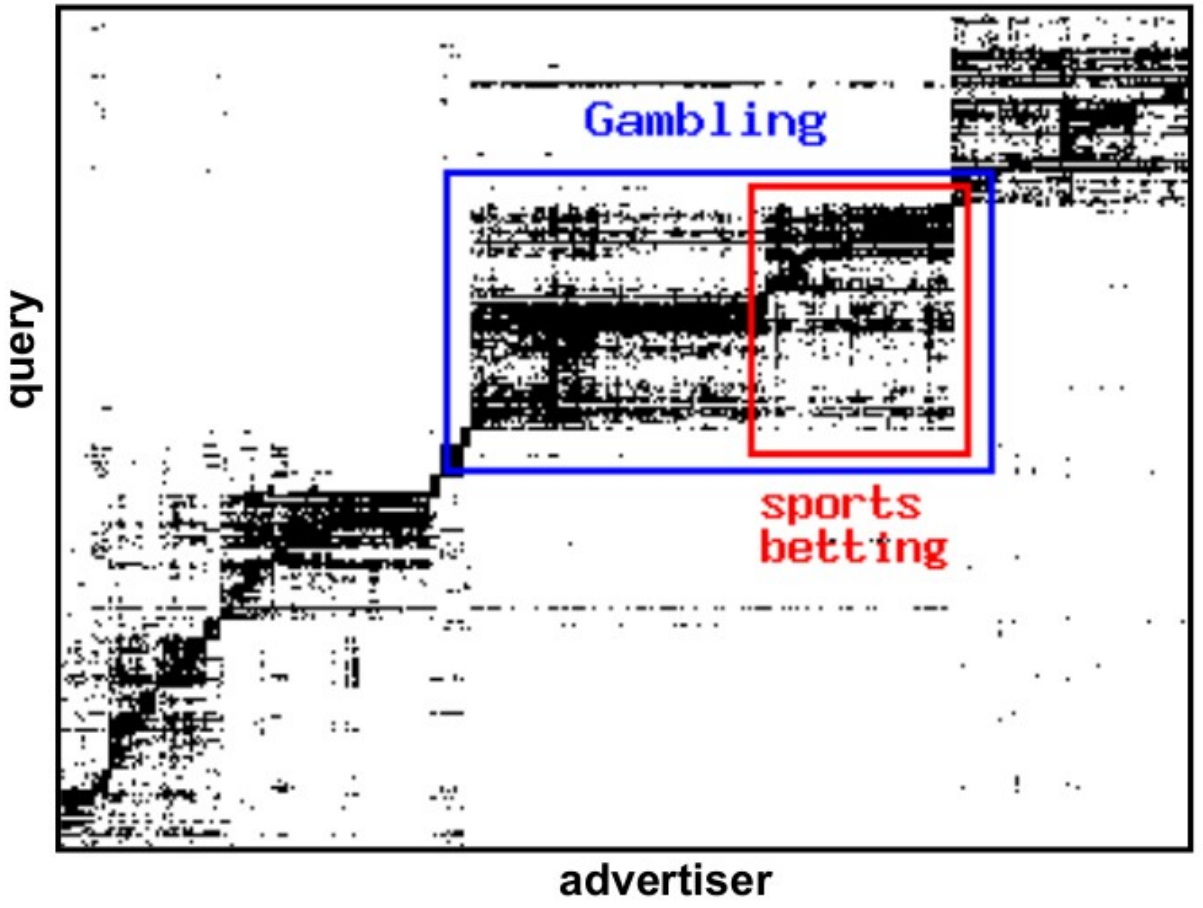


- Asked "ego" to identify social circles to which friends belong

  $\Rightarrow$ Company, high-school, basketball club, squash club, family

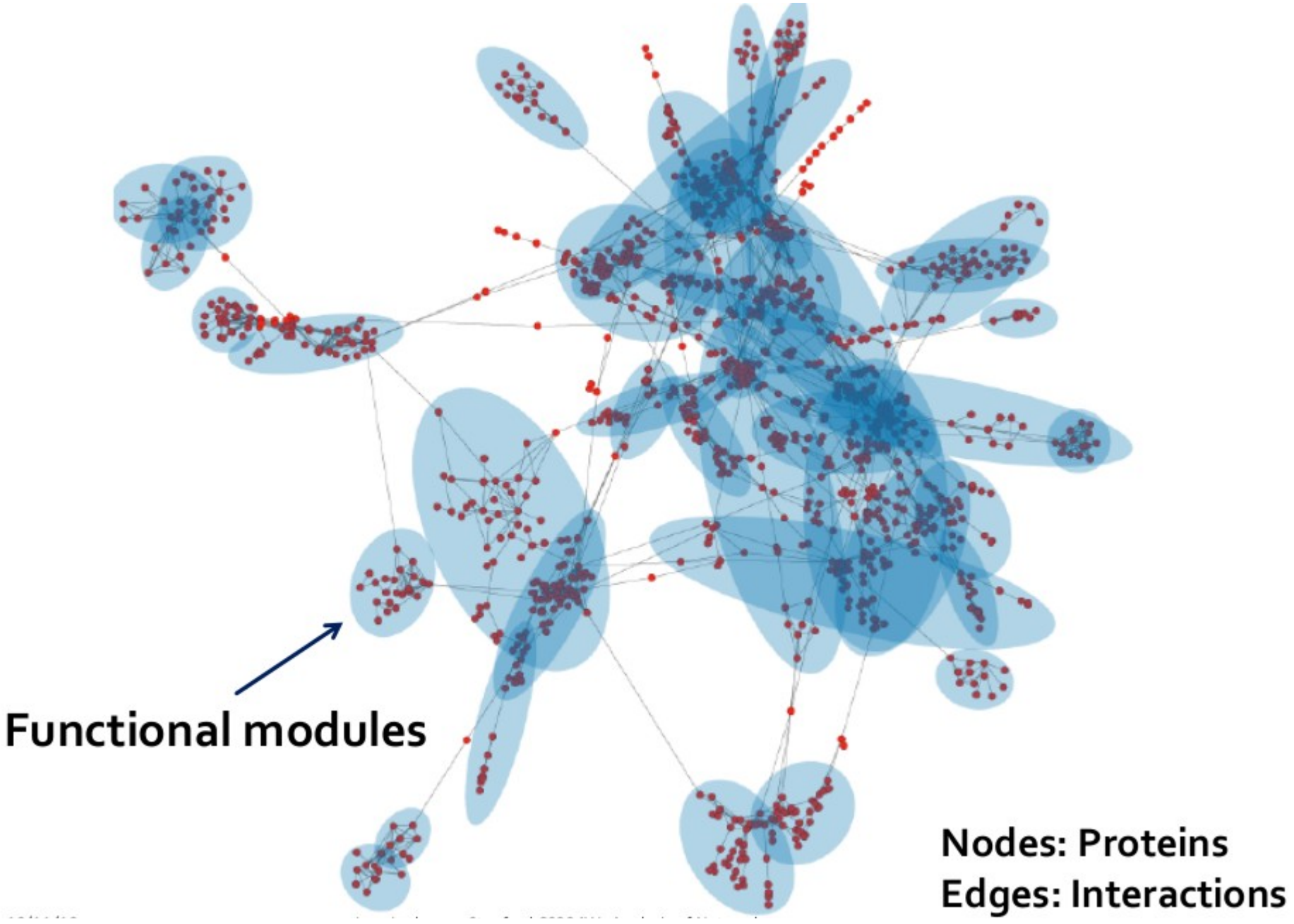
Micro-Markets in Sponsored Search

Find micro-markets by partitioning the "query-to-advertiser" graph in web search:

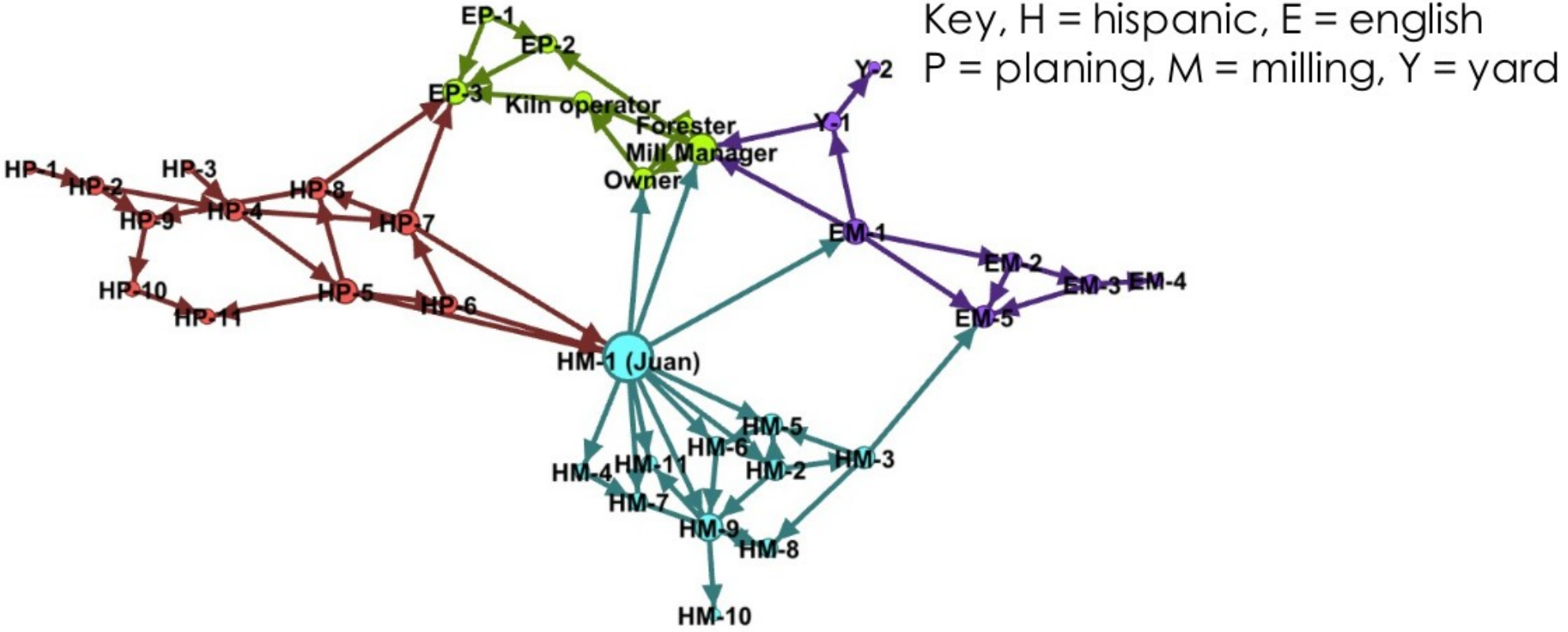


Nodes: advertisers and queries/keywords; Edges: Advertiser advertising on a keyword.

Protein-Protein Interaction



Why look for community structure?

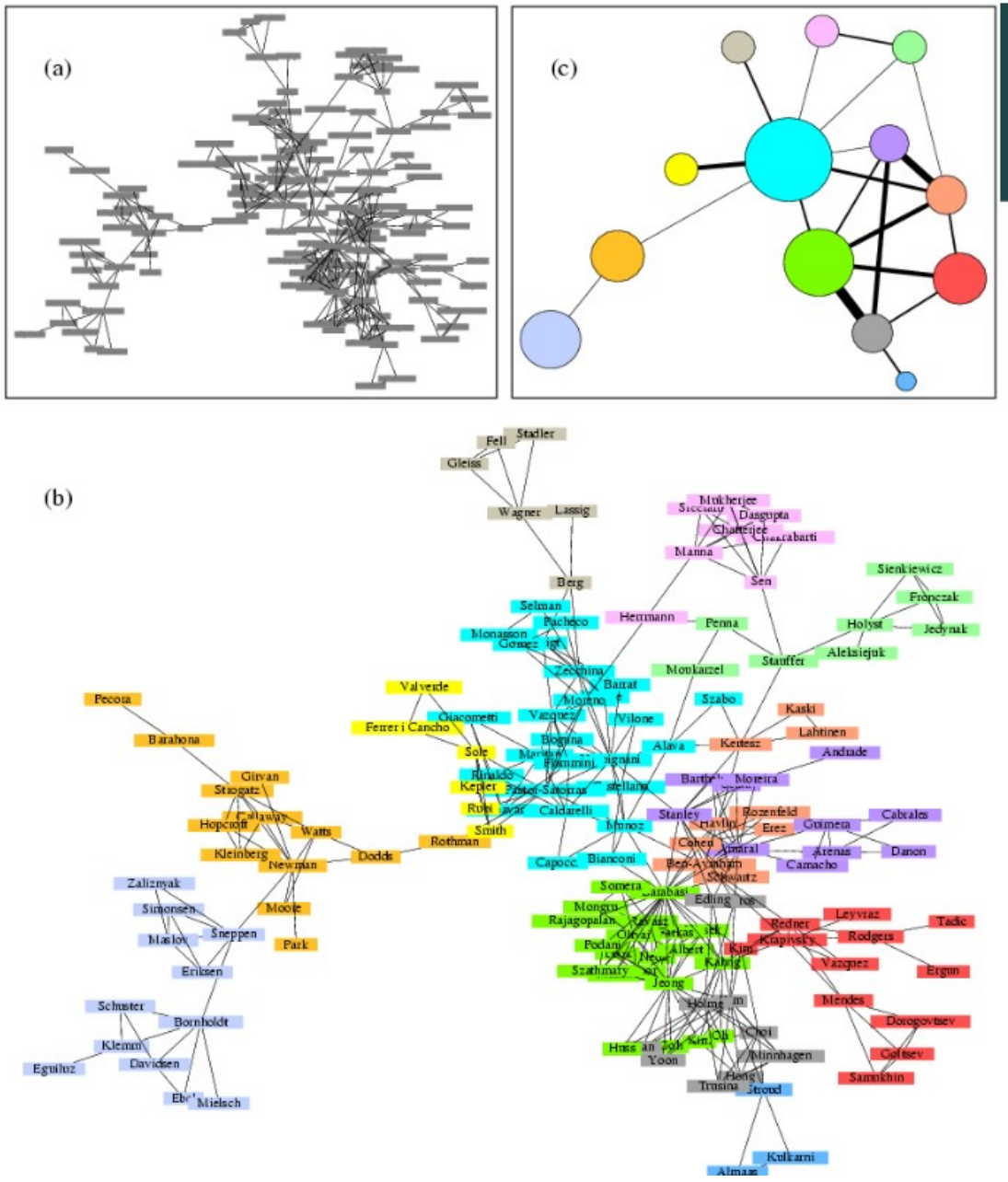


- The management at the sawmill was having difficulty persuading the workers to adopt a new plan, even though everyone would benefit. In particular the Hispanic workers (H) were reluctant to agree. The management called in a sociologist who mapped out who talked to whom regularly. Then they suggested that the management talk to Juan and have him talk to the Hispanic workers. It was a success, promptly everyone was on board with the new plan. Why?

Why: gain understanding

- Gain understanding of networks
  - Discover communities of practice
  - Measure isolation of groups
  - Understand opinion dynamics / adoption

Why: Visualize



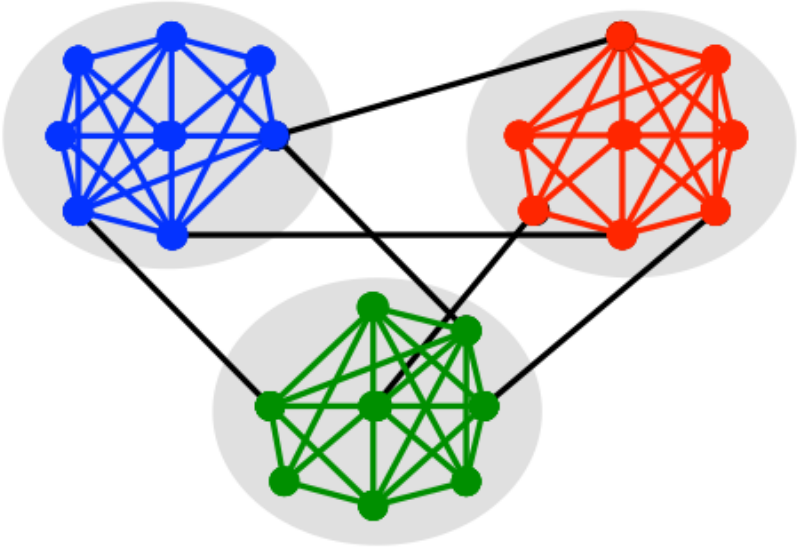
Why do it: visualize

- Communities help to "aggregate" network data

Unveiling network communities

- Nodes in real-world networks organize into communities

  Ex: families, clubs, political organizations, proteins by function, ...



- Community (a.k.a. group, cluster, module) members are:
  - $\Rightarrow$ Well connected among themselves
  - $\Rightarrow$ Relatively well separated from the rest
- Exhibit high cohesiveness w.r.t. the underlying relational patterns
- Q: How can we automatically identify such cohesive subgroups?

Community detection and graph partitioning

- Community detection is a challenging clustering problem
  - C1) No consensus on the structural definition of community
  - C2) Node subset selection often intractable
  - C3) Lack of ground-truth for validation

- Useful for exploratory analysis of network data

  Ex: clues about social interactions, content-related web pages

Graph partitioning

Split V into given number of non-overlapping groups of given sizes

- **Criterion:** number of edges between groups is minimized (more soon)

  Ex: task-processor assignment for load balancing

- Number and sizes of groups unspecified in community detection

  $\Rightarrow$ Identify the natural fault lines along which a network separates

Graph partitioning is hard

- Ex: Graph bisection problem, i.e., partition V into two groups
  - Suppose the groups V_1 and V_2 are non-overlapping
  - Suppose groups have equal size, i.e., $|V_1| = |V_2| = N_v/2$
  - Minimize edges running between vertices in different groups

- Simple problem to describe, but hard to solve

$$\text{Number of ways to partition } V: \quad \binom{N_v}{N_v/2} \approx \frac{2^{N_v}}{\sqrt{N_v}}$$

$\Rightarrow$ Used Stirling's formula $N_v! \approx \sqrt{2\pi N_v}(N_v/e)^{N_v}$

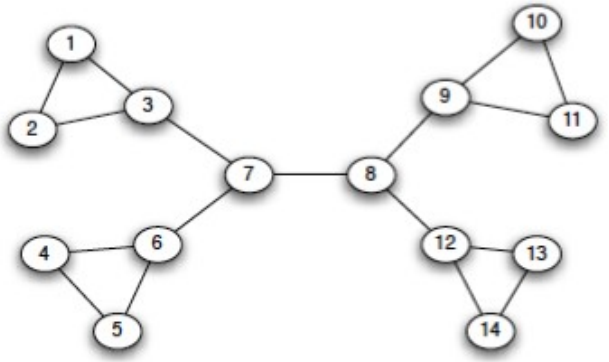
$\Rightarrow$ Exhaustive search intractable beyond toy small-sized networks

- No smart (i.e., polynomial time) algorithm, NP-hard problem

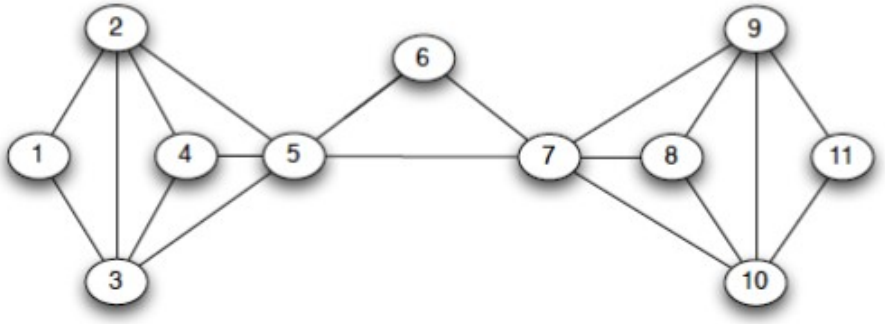
$\Rightarrow$ Seek good heuristics, e.g., relaxations of natural criteria

Strength of weak ties motivation

- Local bridges connect weakly interacting parts of the network



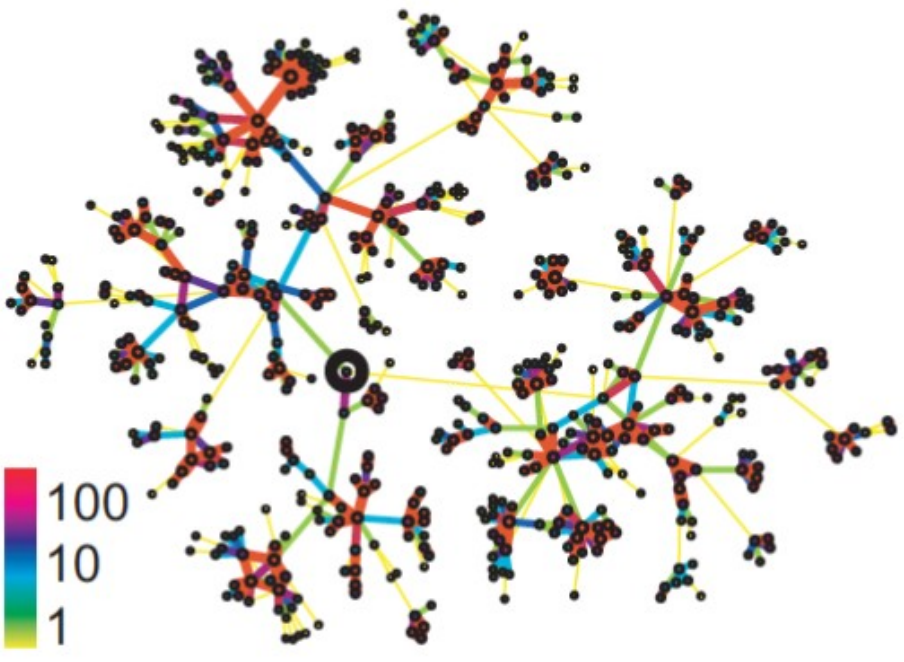
- Q: What about removing those to reveal communities?



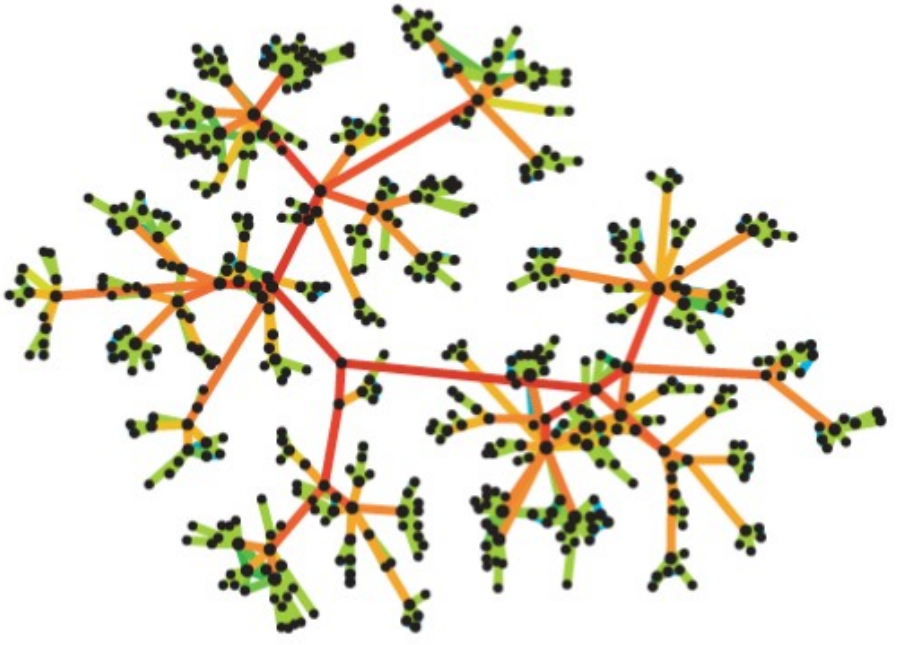
- **Challenges**
  - Multiple local bridges. Some better that others? Which one first?
  - There might be no local bridge, yet an apparent natural division

Edge betweenness centrality

- Idea: high edge betweenness centrality to identify weak ties
  - High $c_{Be}(e)$ edges carry large traffic volume over shortest paths
  - Position at the interface between tightly-knit groups
- Ex: cell-phone network with colored edge strength and betwenness



Edge strength

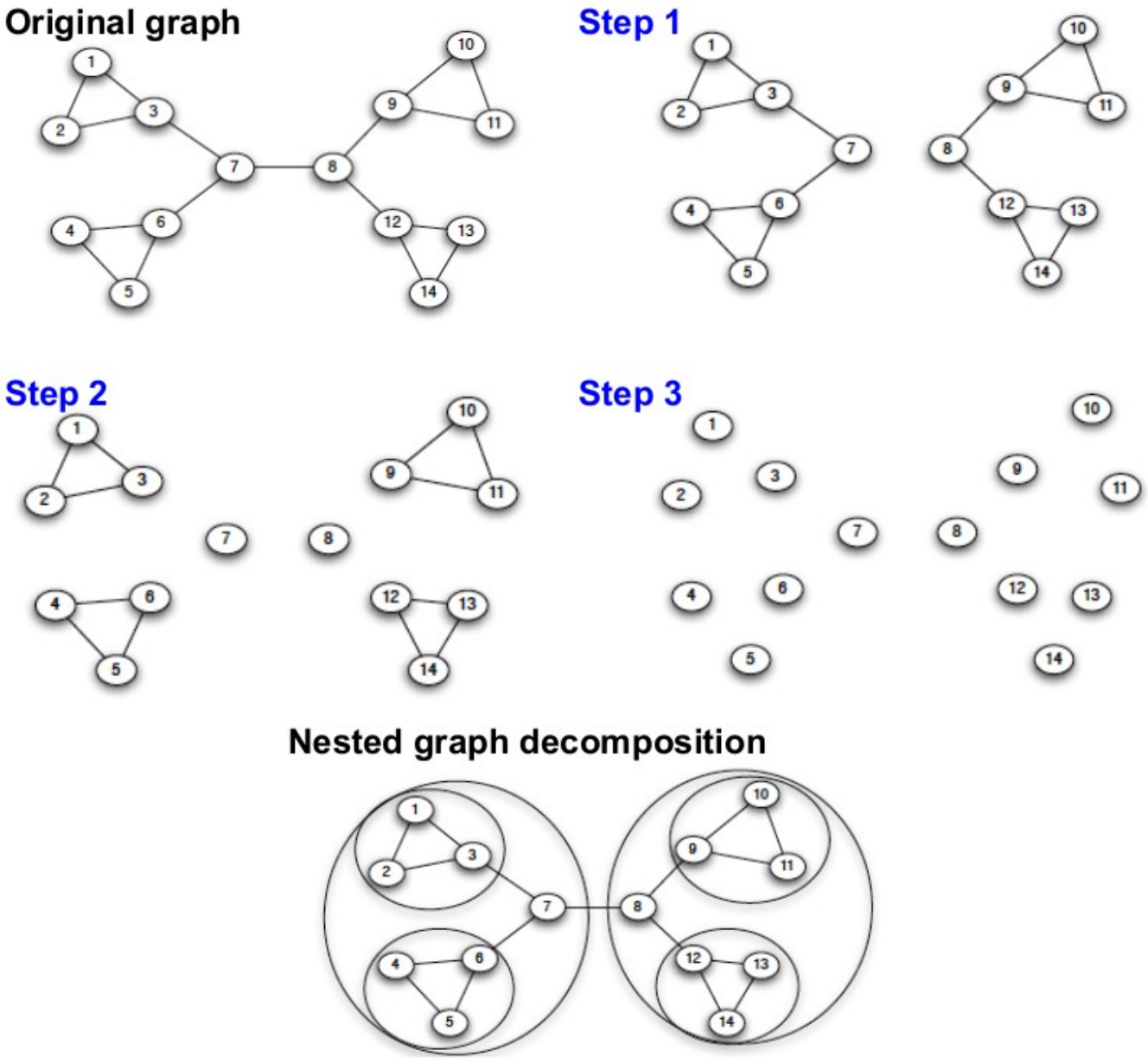


Edge betweenness

Girvan-Newman's method

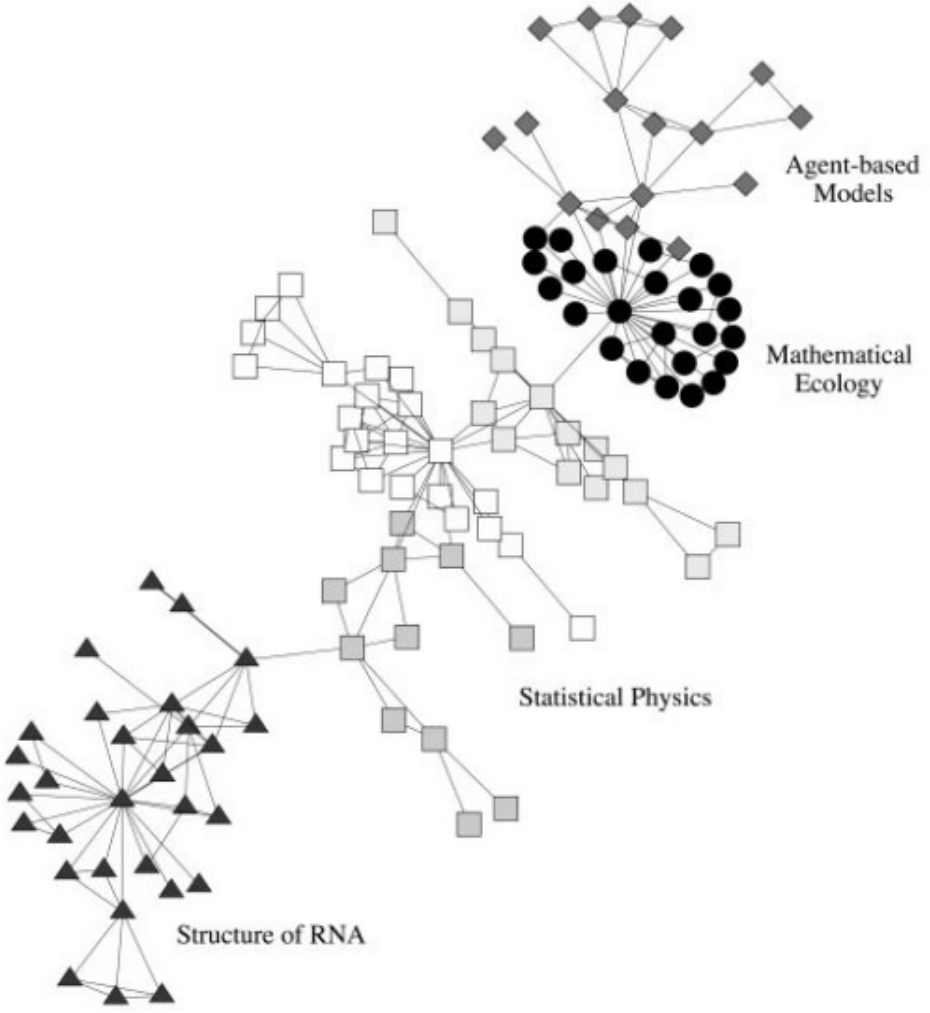
- Girvan-Newmann's method extremely simple conceptually
  - $\Rightarrow$ Find and remove "spanning links" between cohesive subgroups
- **Algorithm:** Repeat until there are no edges left
  - $\Rightarrow$ Calculate the betweenness centrality $c_{Be}(e)$ of all edges
  - $\Rightarrow$ Remove edge(s) with highest $c_{Be}(e)$
- Connected components are the communities identified
  - Divisive method: network falls apart into pieces as we go
  - Nested partition: larger communities potentially host denser groups
  - Recompute edge betweenness in $O(N_v N_e)$-time per step
- M. Girvan and M. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, pp. 7821-7826, 2002

Example: The algorithm in action



Scientific collaboration network

- Ex: Coauthorship network of scientists at the Santa Fe Institute



- Communities found can be traced to different disciplines

Hierarchical clustering

- Greedy approach to iteratively modify successive candidate partitions
  - Agglomerative: successive coarsening of partitions through merging
  - Divisive: successive refinement of partitions through splitting
- Per step, partitions are modified in a way that minimizes a cost
  - Measures of (dis)similarity x_{ij} between pairs of vertices v_i and v_j
  - Ex: Euclidean distance dissimilarity

$$x_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}$$

- Method returns an entire hierarchy of nested partitions of the graph

  $\Rightarrow$ Can range fully from $\{\{v_1\}, \ldots, \{v_{N_v}\}\}$ to V

Agglomerative clustering

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

  S4: Compute the dissimilarity between the new group and all others

  S5: Repeat from S3 until all vertices belong to a single group

- Need to define group dissimilarity from pairwise vertex counterparts
  - Single linkage: group dissimilarity $x^{SL}_{G_i,G_j}$ follows single most dissimilar pair

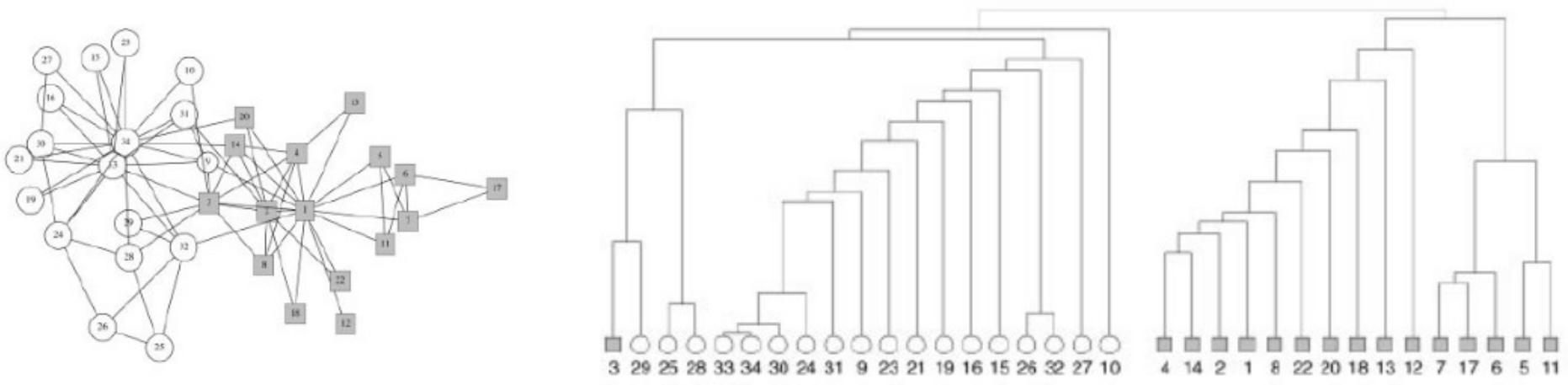
$$x^{SL}_{G_i,G_j} = \max_{u \in G_i, v \in G_j} x_{uv}$$

  - Complete linkage: every vertex pair highly dissimilar to have high $x^{CL}_{G_i,G_j}$

$$x^{CL}_{G_i,G_j} = \min_{u \in G_i, v \in G_j} x_{uv}$$

Dendrogram

- Hierarchical partitions often represented with a dendrogram
- Shows groups found in the network at all algorithmic steps
  $\Rightarrow$ Split the network at different resolutions
- Ex: Girvan-Newman's algorithm for the Zachary's karate club



- Q: Which of the divisions is the most useful/optimal in some sense?
- A: Need to define metrics of graph clustering quality

Modularity

- Size of communities typically unknown $\Rightarrow$ Identify automatically
- Modularity measures how well a network is partitioned in communities
  - Intuition: density of edges in communities higher than expected
- Consider a graph G and a partition into groups $s \in S$. Modularity:

$$Q(G,S) \propto \sum_{s \in S} [(\#\text{ of edges within group } s) - \mathbb{E}[\#\text{ of such edges}]]$$

- Formally, after normalization such that $Q(G,S) \in [-1,1]$

$$Q(G,S) = \frac{1}{2N_e} \sum_{s \in S} \sum_{i,j \in s} \left[A_{ij} - \frac{d_i d_j}{2N_e} \right]$$

$\Rightarrow$ Null model: randomize edges, preserving degree distribution

Expected connectivity among nodes

- Null model: randomize edges preserving degree distribution in G
  - $\Rightarrow$ Random variable $A_{ij} := \mathbb{I}\{(i,j) \in E\}$
  - $\Rightarrow$ Expectation is $\mathbb{E}[A_{ij}] = \mathrm{P}((i,j) \in E)$
- Suppose node i has degree d_i, node j has degree d_j
  - $\Rightarrow$ Degree is "# of spokes" per node, $2N_e$ spokes in G

- Probability spoke i_k connected to j is $\frac{d_j}{2N_e-1} \approx \frac{d_j}{2N_e}$, hence

$$\mathrm{P}((i,j) \in E) = \mathrm{P}\left(\bigcup_{i_k=1}^{d_i} \{\text{spoke } i_k \text{ connected to } j\}\right)$$

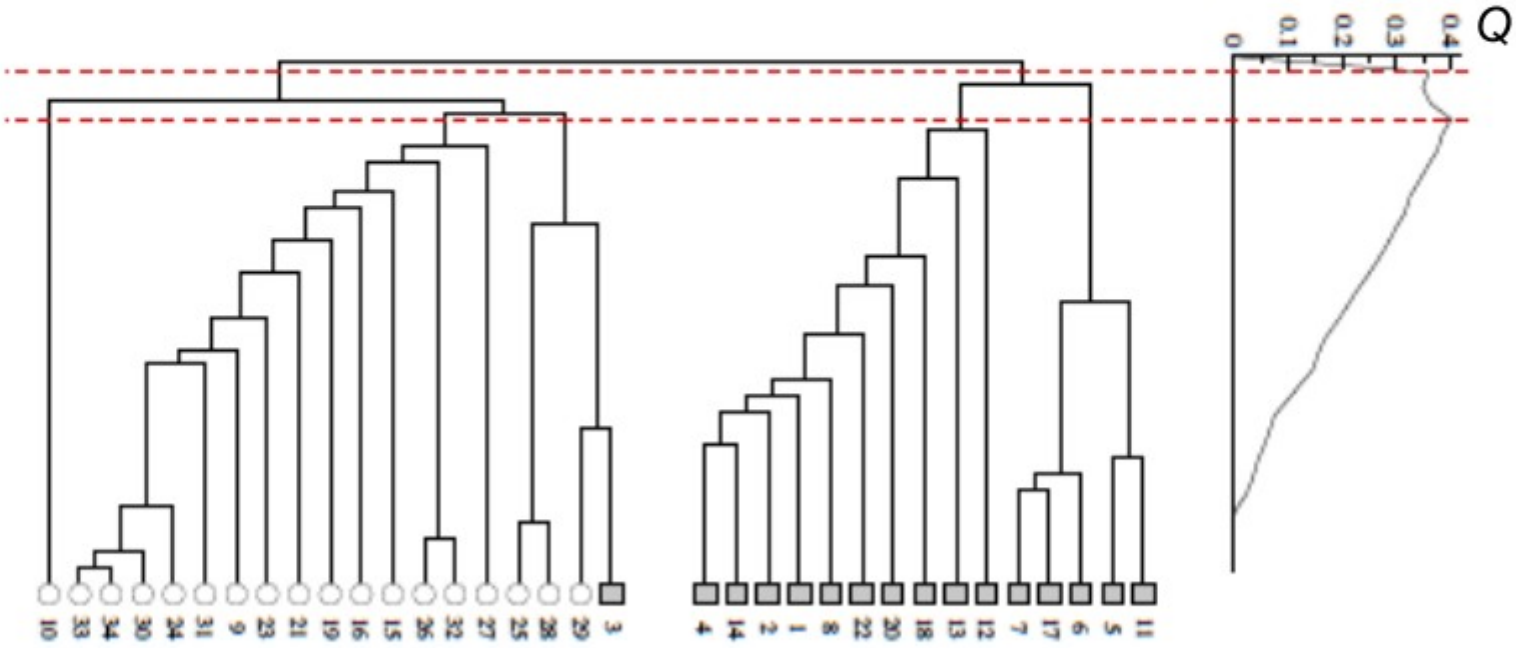
$$= \sum_{i_k=1}^{d_i} \mathrm{P}(\text{spoke } i_k \text{ connected to } j) = \frac{d_i d_j}{2N_e}$$

Assessing clustering quality

- Can evaluate the modularity of each partition in a dendrogram

  $\Rightarrow$ Maximum value gives the "best" community structure

- Ex: Girvan-Newman's algorithm for the Zachary's karate club



- Q: Why not optimize $Q(G, S)$ directly over possible partitions S?

Modularity: another look

- **Modularity of partitioning S of graph G:**
  - $\mathbf{Q \propto \sum_{s \in S}}$ **[(# edges within group *s*) – (expected # edges within group *s*)]**
  - $$Q(G,S) = \underbrace{\frac{1}{2m}}\sum_{s \in S}\sum_{i \in s}\sum_{j \in s}\left(A_{ij} - \frac{k_i k_j}{2m}\right)$$

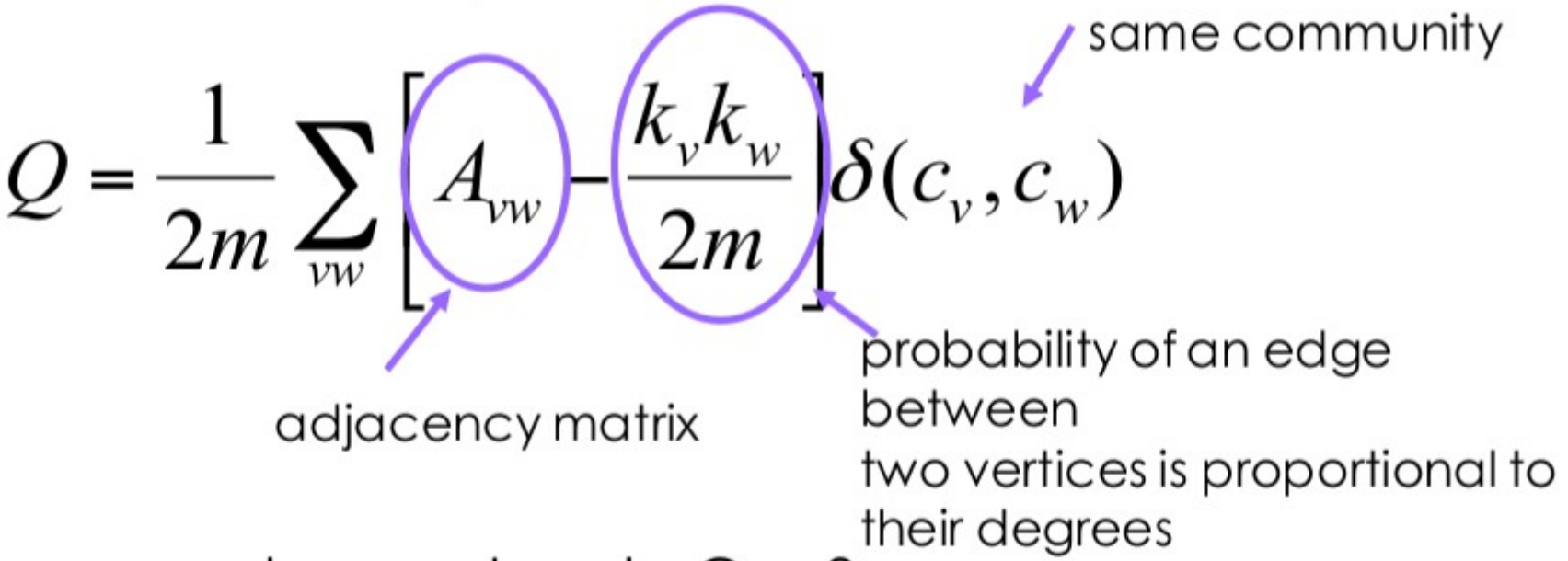
    Normalizing const.: -1<Q<1

    A_{ij} = 1 if i→j, 0 else

- **Modularity values take range [–1,1]**
  - It is positive if the number of edges within groups exceeds the expected number
  - **Q** greater than **0.3-0.7** means **significant community structure**

Modularity: another look

- Consider edges that fall within a community or between a community and the rest of the network
- Define modularity:



- For a random network, Q = 0
  - the number of edges within a community is no different from what you would expect

Modularity: another look

$$Q(G,S) = \frac{1}{2m}\sum_{s\in S}\sum_{i\in s}\sum_{j\in s}\left(A_{ij} - \frac{k_i k_j}{2m}\right)$$

Equivalently modularity can be written as:

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

- A_{ij} represents the edge weight between nodes i and j;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively;
- $2m$ is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is an indicator function

Idea: We can identify communities by maximizing modularity

Louvain Algorithm

Louvain Algorithm

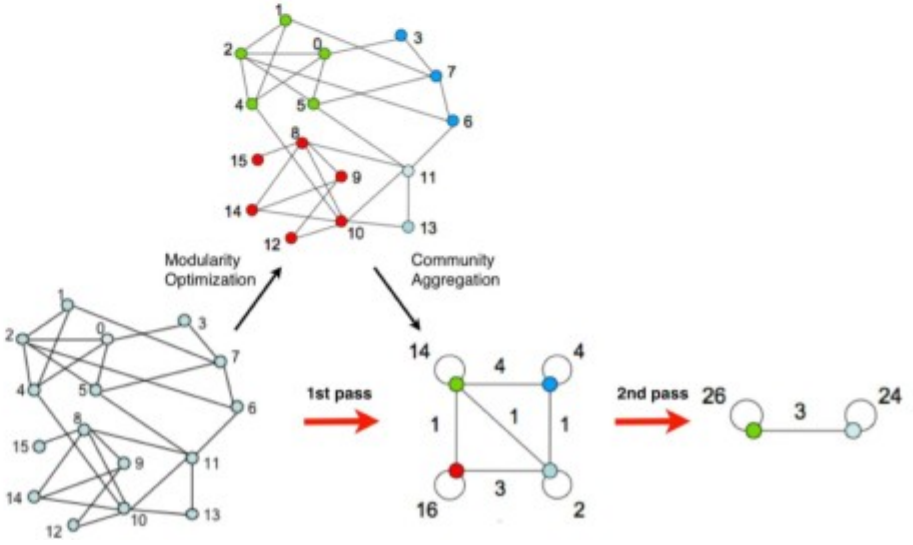
- **Greedy algorithm** for community detection
  - O(n log n) run time
- Supports weighted graphs
- Provides hierarchical partitions
- Widely utilized to **study large networks** because:
  - Fast
  - Rapid convergence properties
  - High modularity output (i.e., “better communities”)

“Fast unfolding of communities in large networks” Blondel et al. (2008)

Louvain Algorithm: at high level

- Louvain algorithm **greedily maximizes** modularity
- **Each pass is made of 2 phases:**
  - **Phase 1:** Modularity is optimized by allowing only local changes of communities
  - **Phase 2:** The identified communities are aggregated in order to build a new network of communities
  - **Goto Phase 1**

The passes are repeated **iteratively** until no increase of modularity is possible!



Louvain: 1st phase (partitioning)

- Put each node in a graph into a distinct community (one node per community)
- For each node *i*, the algorithm performs two calculations:
  - Compute the modularity gain (ΔQ) when putting node i into the community of some neighbor j
  - Move i to a community of node j that yields the largest gain ΔQ
- The loop runs until no movement yields a gain

Louvain: Modularity Gain

What is ΔQ if we move node i to community C?

$$\Delta Q(i \to C) = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

- where:
  - Σ_{in}... sum of link weights between nodes in C
  - Σ_{tot}... sum of all link weights of nodes in C
  - $k_{i,in}$... sum of link weights between node i and C
  - k_i... sum of all link weights (i.e., degree) of node i

Σ_{in}:

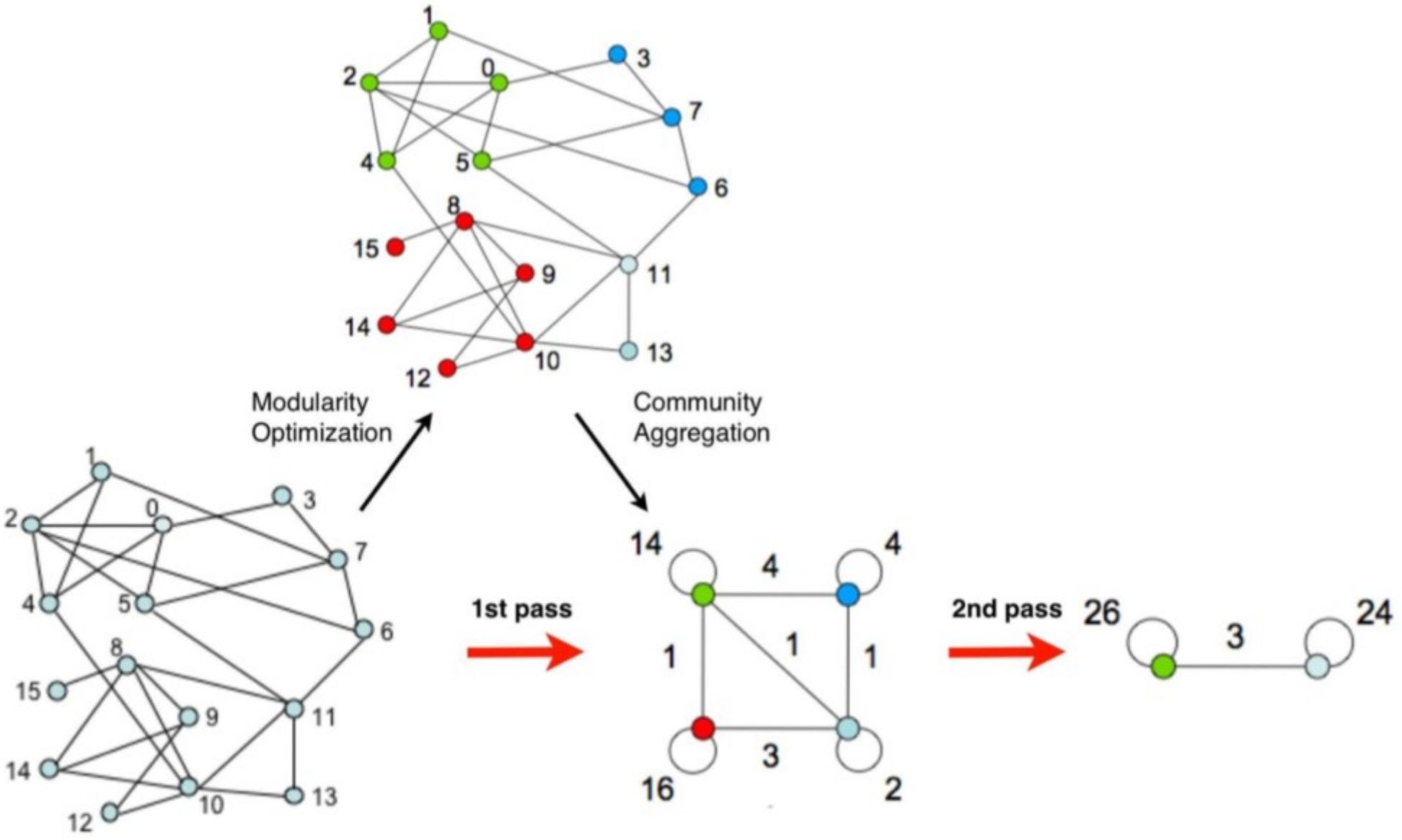
Σ_{tot}:

- Also need to derive $\Delta Q(D \to i)$ of taking node i out of community D.
- And then: $\Delta Q = \Delta Q(i \to C) + \Delta Q(D \to i)$

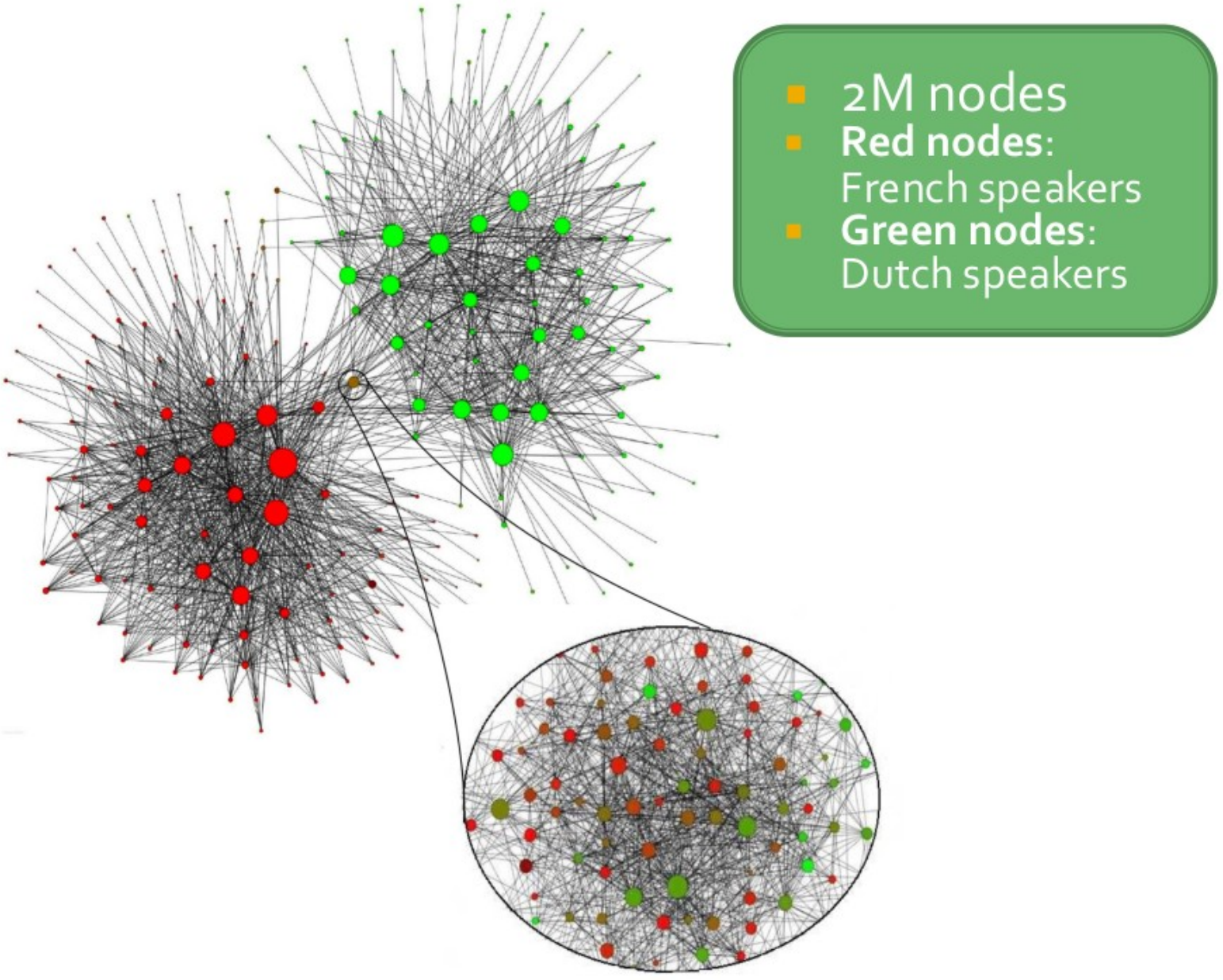
Louvain: 2nd phase (restructuring)

- The partitions obtained in the first phase are contracted into **super-nodes**, and the network is created accordingly
  - Super-nodes are connected if there is at least one edge between nodes of the corresponding partitions
  - The weight of the edge between the two super-nodes is the sum of the weights from all edges between their corresponding partitions
- **The loop runs until the community configuration does not change anymore**

Louvain Algorithm Overview



Loouvain: Belgian Phone Network



There are many algorithms

Community detection in graphs

Santo Fortunato*

Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I-ITALY.

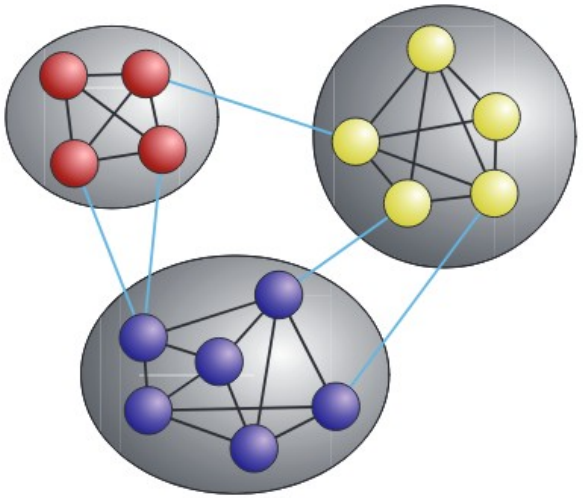
Contents

*Electronic address: fortunato@isi.it

There are many algorithms

Community detection in networks: A user guide

Santo Fortunato*



Contents

There are many “score” functions

Defining and Evaluating Network Communities based on Ground-truth

Jaewon Yang
Stanford University
crucis@stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

Dataset	N	E	C	S	A
LiveJournal	4.0M	34.9M	311,782	40.06	3.09
Friendster	117.7M	2,586.1M	1,449,666	26.72	0.32
Orkut	3.0M	117.2M	8,455,253	34.86	95.9
Ning (225 nets)	7.0M	35.5M	137,177	46.89	0.92
Amazon	0.33M	0.92M	49,732	99.86	14.83
DBLP	0.42M	1.34M	2,547	429.79	2.56

Table I

230 SOCIAL, COLLABORATION AND INFORMATION NETWORKS WITH EXPLICIT GROUND-TRUTH COMMUNITIES. N: NUMBER OF NODES, E: NUMBER OF EDGES, C: NUMBER OF COMMUNITIES, S: AVERAGE COMMUNITY SIZE, A: COMMUNITY MEMBERSHIPS PER NODE. NING STATISTICS ARE AGGREGATED OVER 225 DIFFERENT SUBNETWORKS.

(A) Scoring functions based on internal connectivity:

- **Internal density:** $f(S) = \frac{m_S}{n_S(n_S-1)/2}$ is the internal edge density of the node set S [24].
- **Edges inside:** $f(S) = m_S$ is the number of edges between the members of S [24].
- **Average degree:** $f(S) = \frac{2m_S}{n_S}$ is the average internal degree of the members of S [24].
- **Fraction over median degree (FOMD):** $f(S) = \frac{|\{u:u\in S, |\{(u,v):v\in S\}|>d_m\}|}{n_S}$ is the fraction of nodes of S that have internal degree higher than d_m, where d_m is the median value of $d(u)$ in V.
- **Triangle Participation Ratio (TPR):** $f(S) = \frac{|\{u:u\in S, \{(v,w):v,w\in S, (u,v)\in E, (u,w)\in E, (v,w)\in E\}\neq\emptyset\}|}{n_S}$ is the fraction of nodes in S that belong to a triad.

(B) Scoring functions based on external connectivity:

- **Expansion** measures the number of edges per node that point outside the cluster: $f(S) = \frac{c_S}{n_S}$ [24].
- **Cut Ratio** is the fraction of existing edges (out of all possible edges) leaving the cluster: $f(S) = \frac{c_S}{n_S(n-n_S)}$ [9].

(C) Scoring functions that combine internal and external connectivity:

- **Conductance:** $f(S) = \frac{c_S}{2m_S+c_S}$ measures the fraction of total edge volume that points outside the cluster [27].
- **Normalized Cut:** $f(S) = \frac{c_S}{2m_S+c_S} + \frac{c_S}{2(m-m_S)+c_S}$ [27].
- **Maximum-ODF (Out Degree Fraction):** $f(S) = \max_{u\in S} \frac{|\{(u,v)\in E:v\notin S\}|}{d(u)}$ is the maximum fraction of edges of a node in S that point outside S [8].
- **Average-ODF:** $f(S) = \frac{1}{n_S}\sum_{u\in S} \frac{|\{(u,v)\in E:v\notin S\}|}{d(u)}$) is the average fraction of edges of nodes in S that point out of S [8].
- **Flake-ODF:** $f(S) = \frac{|\{u:u\in S, |\{(u,v)\in E:v\in S\}|<d(u)/2\}|}{n_S}$ is the fraction of nodes in S that have fewer edges pointing inside than to the outside of the cluster [8].

(D) Scoring function based on a network model:

- **Modularity:** $f(S) = \frac{1}{4}(m_S - E(m_S))$ is the difference between m_S, the number of edges between nodes in S and $E(m_S)$, the expected number of such edges in a random graph with identical degree sequence [21].

War Story

FastStep: Scalable Boolean Matrix Decomposition

Miguel Araujo[1,2], Pedro Ribeiro[1], and Christos Faloutsos[2]

[1] Cracs/INESC-TEC and University of Porto, Porto, Portugal
pribeiro@dcc.fc.up.pt
[2] Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
{maraujo,christos}@cs.cmu.edu

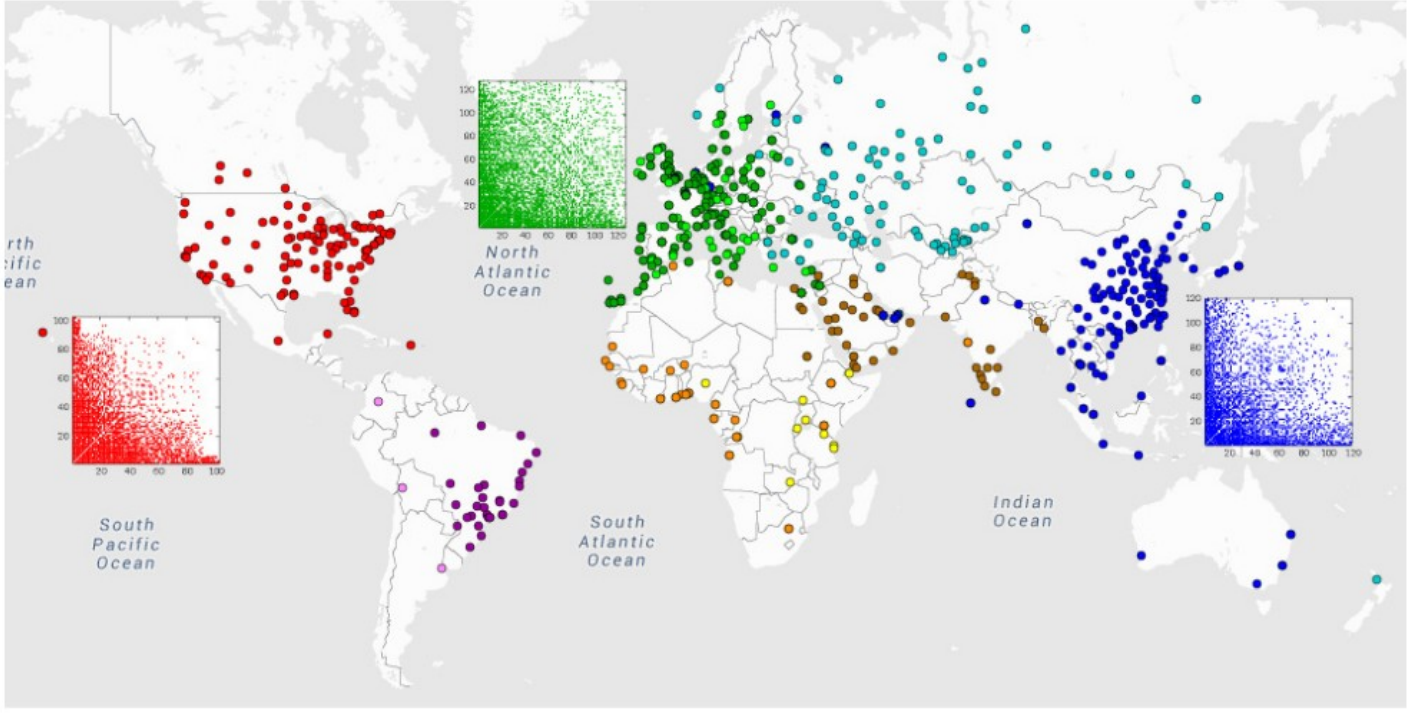


Table 1: Comparison of decomposition m
pretability and beyond block struct

	FastStep	SVD					
Scalability	✓	✓					
Overlapping	✓	✓					
Beyond blocks	✓	✓	✓			✓	✓
Boolean Reconstruction	✓			✓	✓	✓	✓
Arbitrary Marginals	✓	✓	✓				✓
Interpretability	✓			✓	✓	✓	