NETWORK SCIENCE OF ONLINE INTERACTIONS

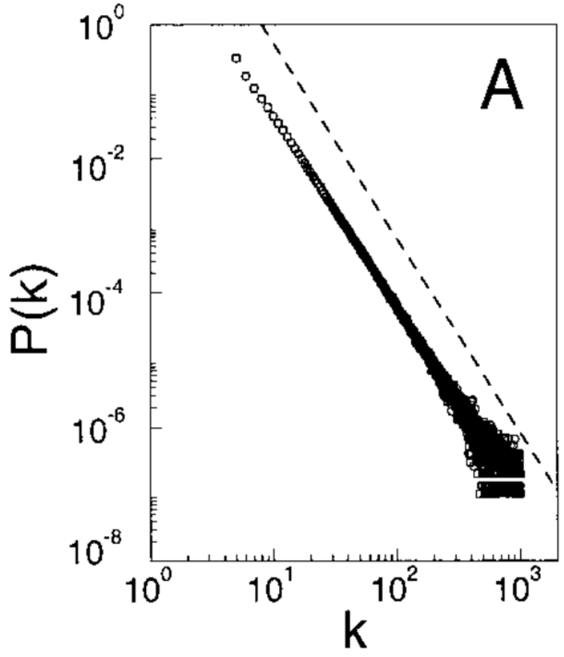
Chapter 6: Communities

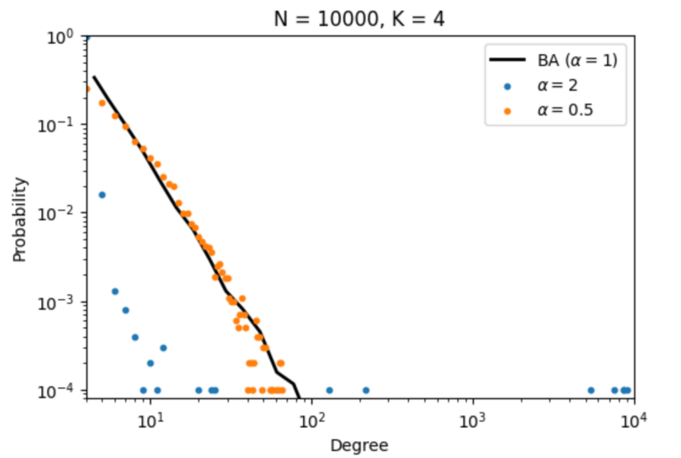
Joao Neto

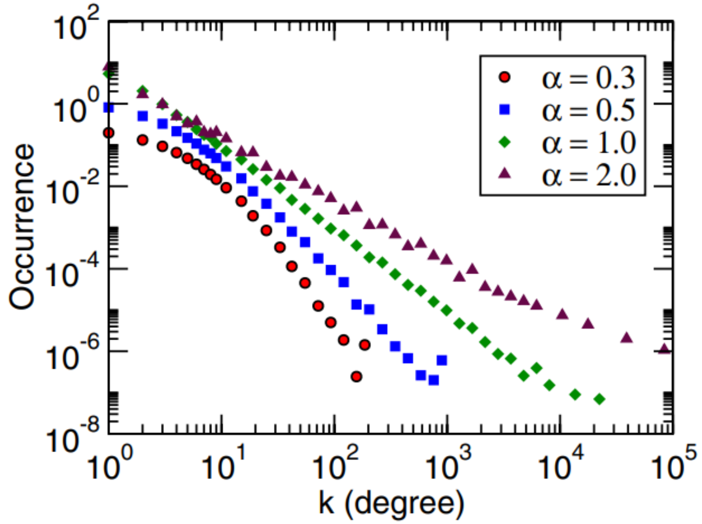
26/May/2023

SUMMARY

- Many models focusing on emulating certain properties
  - Degree distribution, clustering, triadic closure, etc
- No single "best model"
- Preferential attachment is a key mechanism
  - Can create heavy-tailed distributions
  - If unbalanced, can create hyper-concentrated hubs
- Variations of PA models can create a variety of degree distributions







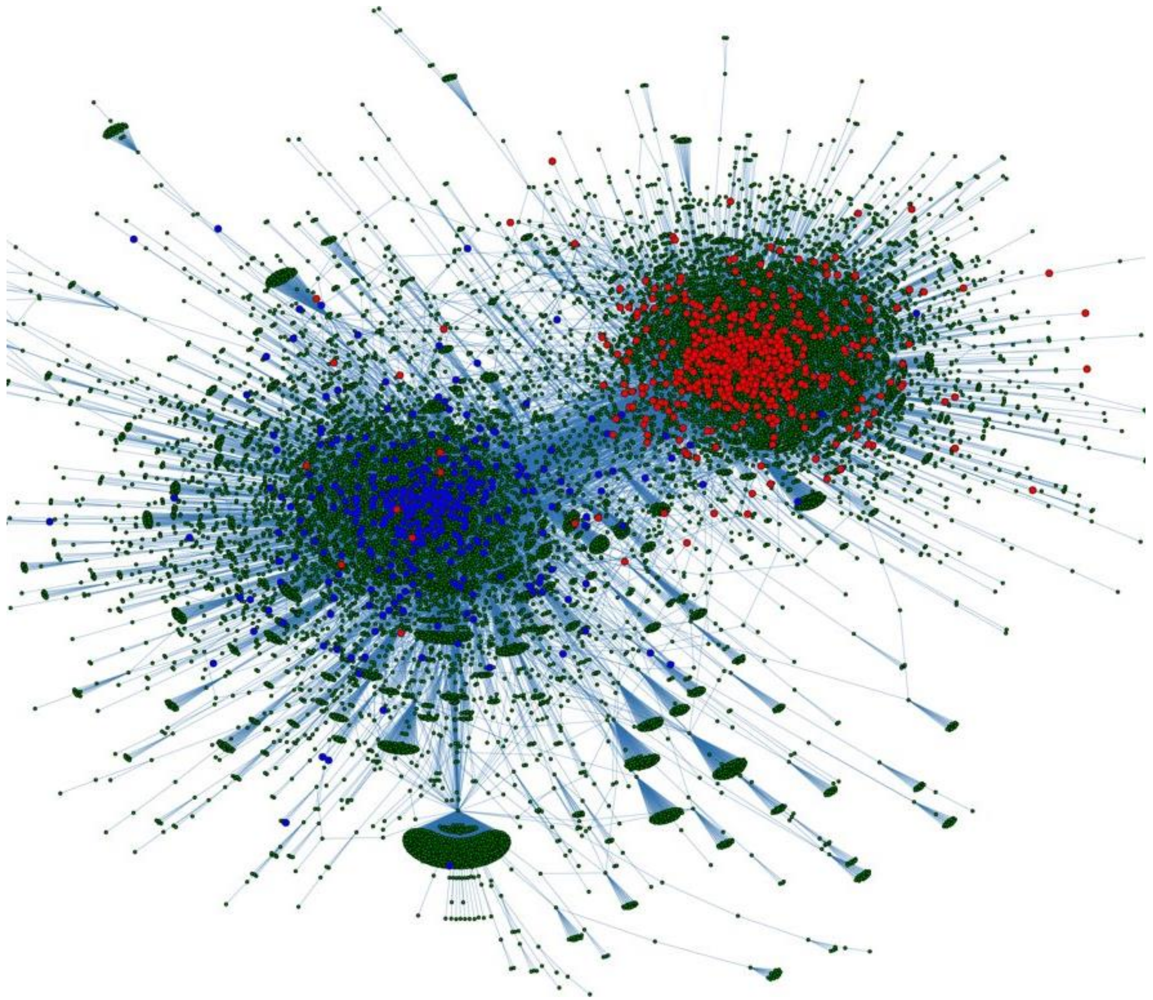
INTRODUCTION

- Community detection is one of the main fields of network theory

- Many reasons:
  - Uncover network structure
  - Identify node affiliation
  - Find missing links ↔ predict links

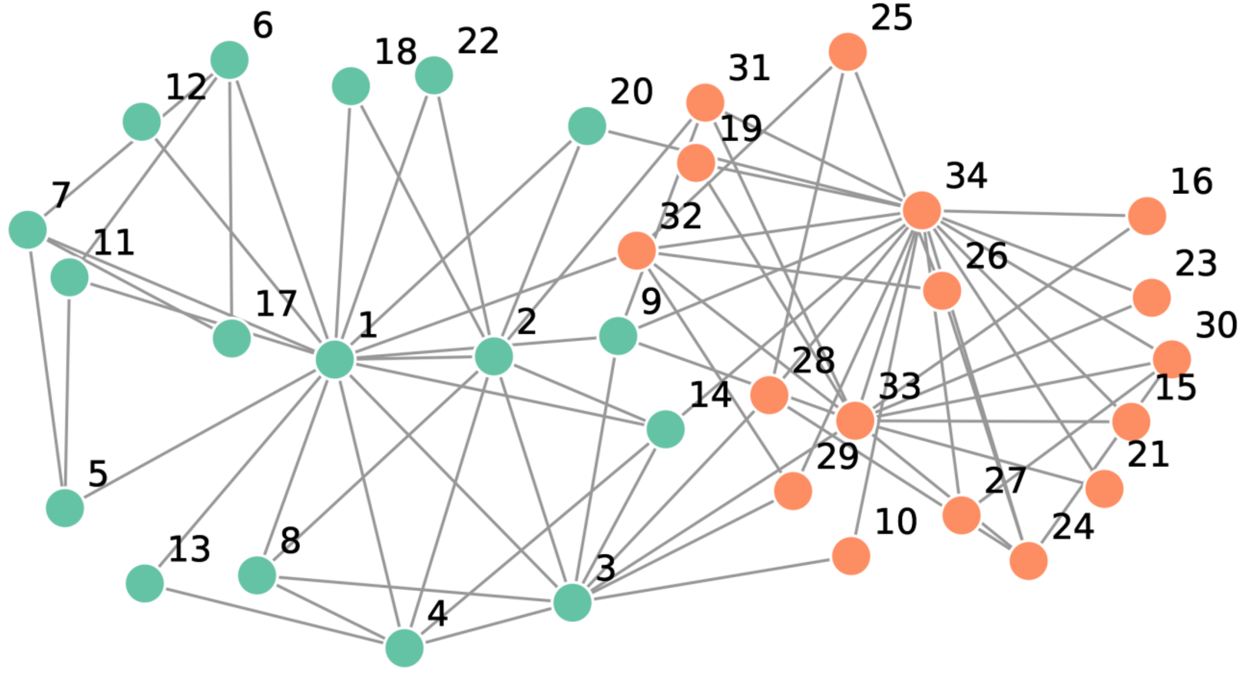
- Different algorithms for different definitions of a community

- Two categories
  - Descriptive algorithms
  - Inferential algorithms



INTRODUCTION

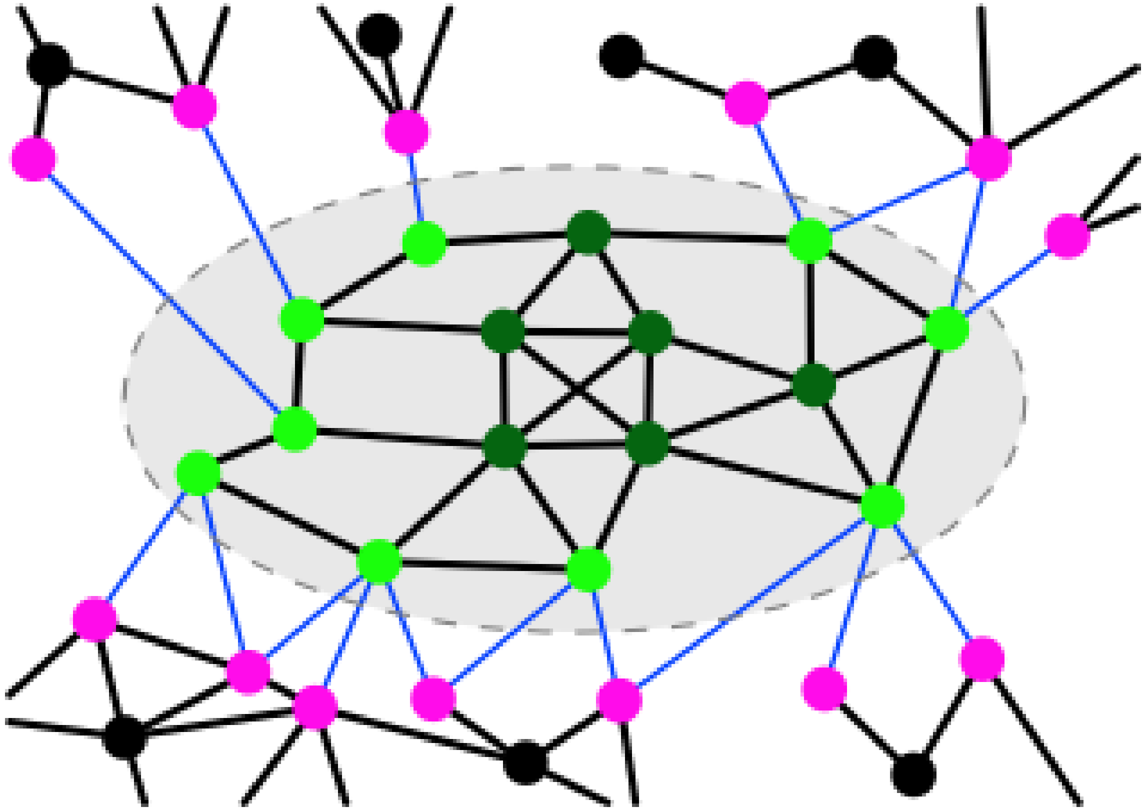
- Basic example: Zachary's Karate club
  - 34 individuals
  - Mapped external relationships
  - Disagreement between 2 instructors
  - Break in two groups
  - Standard test of community detection methods
    - Find two communities
    - Assign nodes to the correct community



5.1 BASIC DEFINITIONS

- Community variables
  - Node degree
    - Internal links k_i^{int} (black)
    - External links k_i^{ext} (blue)
  - Internal links L_C, internal nodes N_C
  - Internal link density $\delta_C^{int} = 2L_C/N_C(N_C-1)$
  - Community degree $k_C = \sum_{j \in C}\left(k_j^{int} + k_j^{ext}\right)$

- Valid for undirected, unweighted networks
  - Undirected, weighted: degree → strength
  - Directed: split in in- and out-links, harder to interpret

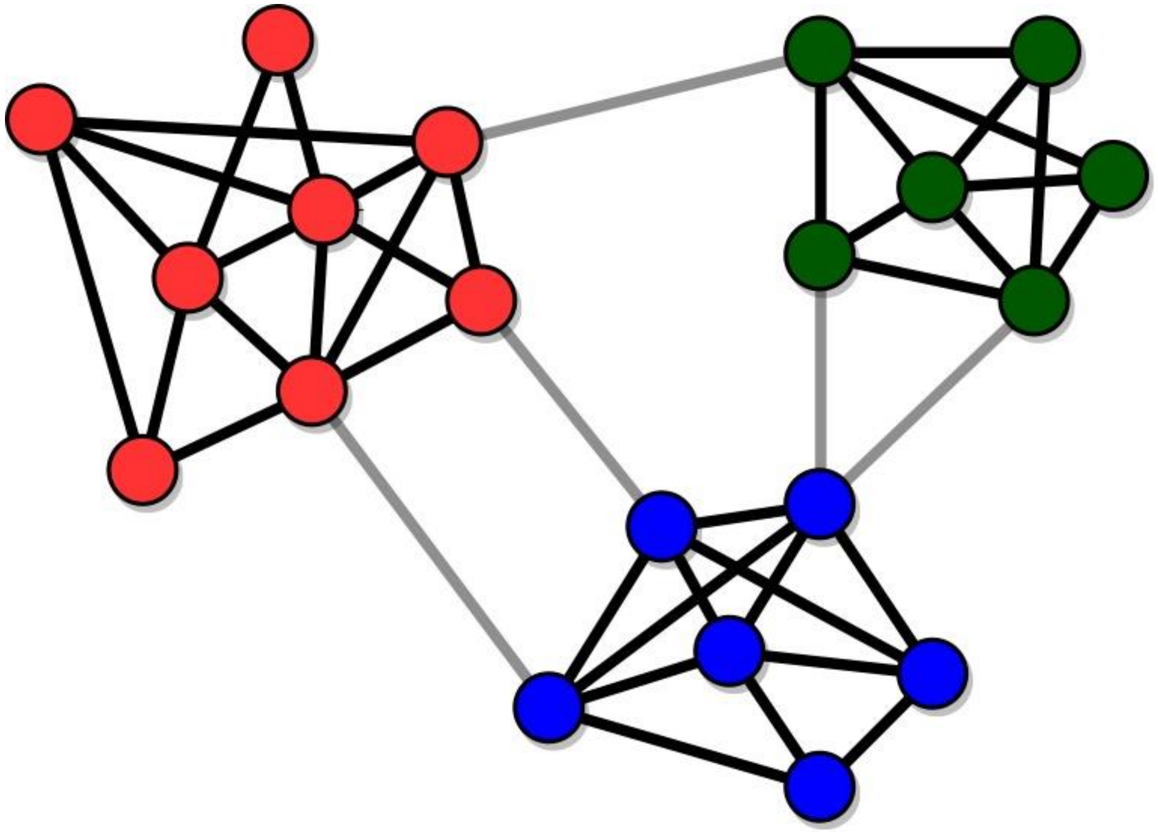


5.1 BASIC DEFINITIONS

- Intuitive idea:
  - Many internal links (cohesion)
  - Few external links (separation)

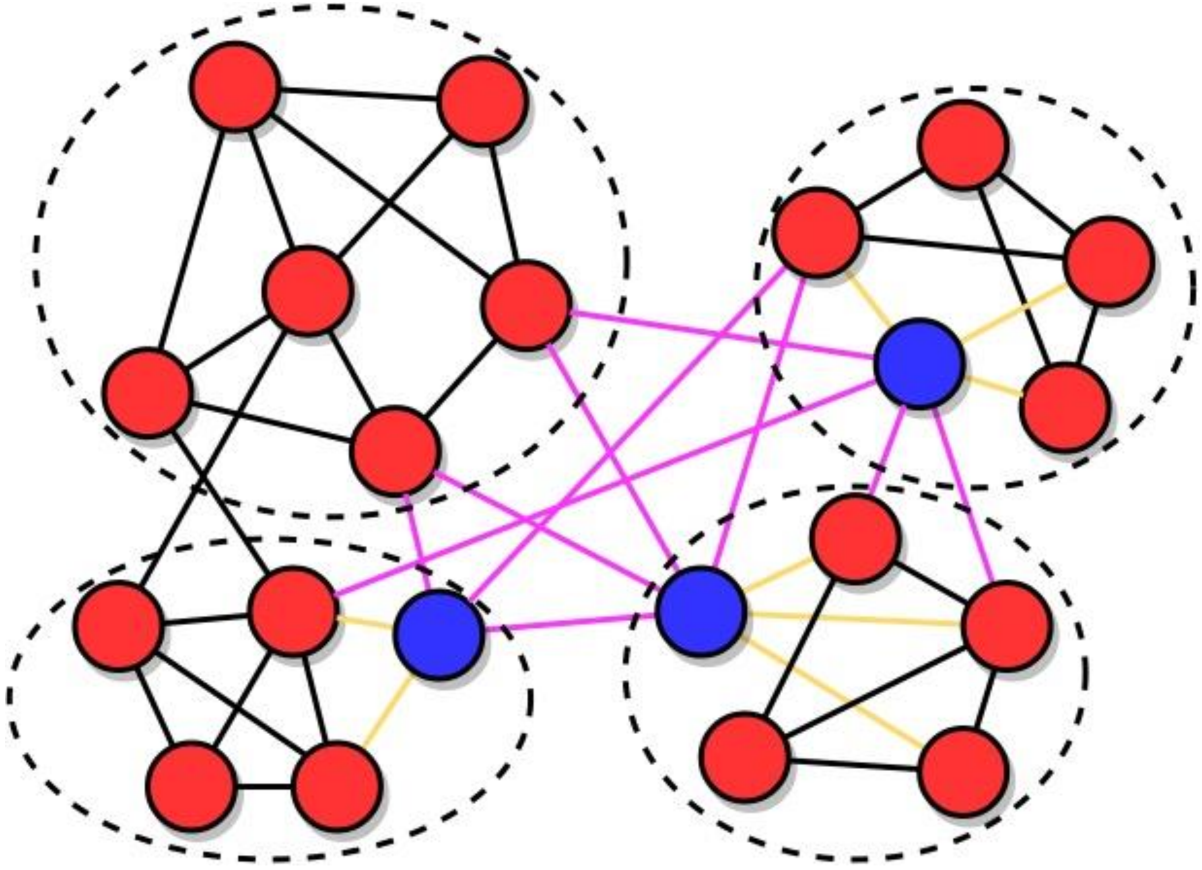
- A community should have more internal than external links

- Definition #1
  - **Strong community**: subnetwork where $k_i^{int} > k_i^{out}$ for *each node*
  - **Weak community**: $\sum_i k_i^{int} > \sum_i k_i^{out}$
  - Problem: compares the community to the entire network



5.1 BASIC DEFINITIONS

- Definition #2
  - Compares communities to *other communities*
  - Strong community: internal degree of each node higher than to any other single community (red)
  - Weak community: sum of internal degrees greater than sum of external to any other community (red + blue)
- Current definitions: counting links
  - Bias against small communities
- Alternative: compare link probabilities
  - Requires a network model
  - Inferential methods (more later)

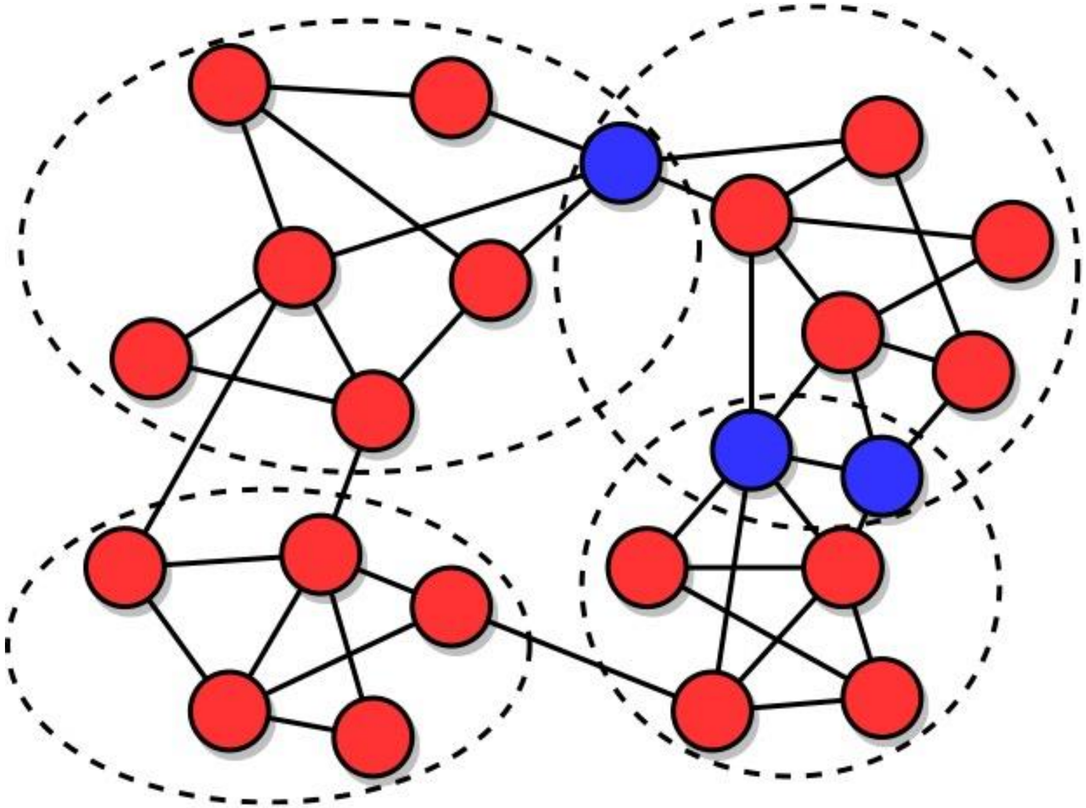


5.1 BASIC DEFINITIONS

- Communities can be
  - Non-overlapping: **partition**
  - Overlapping: **cover**

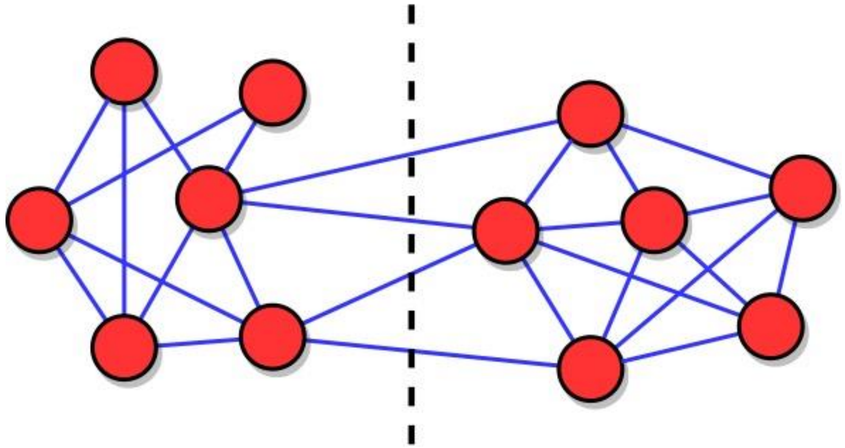
- Number of communities
  - Number of possible partitions grows super-exponentially (Bell's number)
  - $N = 15 \rightarrow$ 1.3 billion possible partitions
  - Number of possible covers is even worse
  - Need for a heuristic algorithm to detect communities

- Graph partitioning is well-studied
  - Community detection in networks
  - Task parallelization in computer science



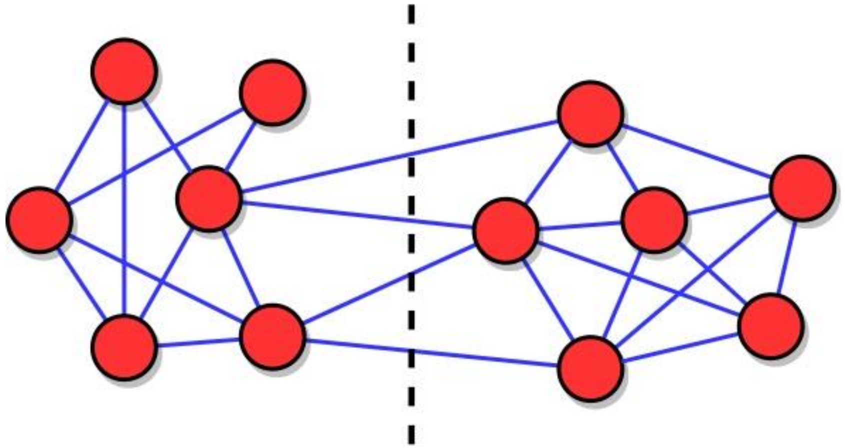
6.2 RELATED PROBLEMS

- Network partitioning with fixed group size
  - **Cut size**: number of links between partitions
  - Good partitions: minimum cut size
  - **Graph bisection**: two partitions with equal size
    - How to do it?
- **Kerninghan-Lin** algorithm
  - The idea: minimize cut size
  - The algorithm
    1. Start from a random bisection
    2. For each pair of nodes from the two groups, compute the swap that would result in the largest decrease in cut size
    3. Swap those nodes and **lock** them in place
    4. Repeat from 2 until cut size cannot be decreased



6.2 RELATED PROBLEMS

- **Kerninghan-Lin** algorithm
  - Greedy algorithm: always tries to minimize/maximize by step
  - May get stuck in local minima/maxima
    - Variants implement random changes to avoid this problem
  - Depends on initial condition
    - Run many in parallel and choose the one with lowest cut size
  - Widely used as post-processing to improve partitioning from other methods (e.g. expert knowledge)
- Limitations of partitioning
  - Partitioning maximizes separation, not internal link density: **not necessarily good communities**
  - Requires giving the number of communities

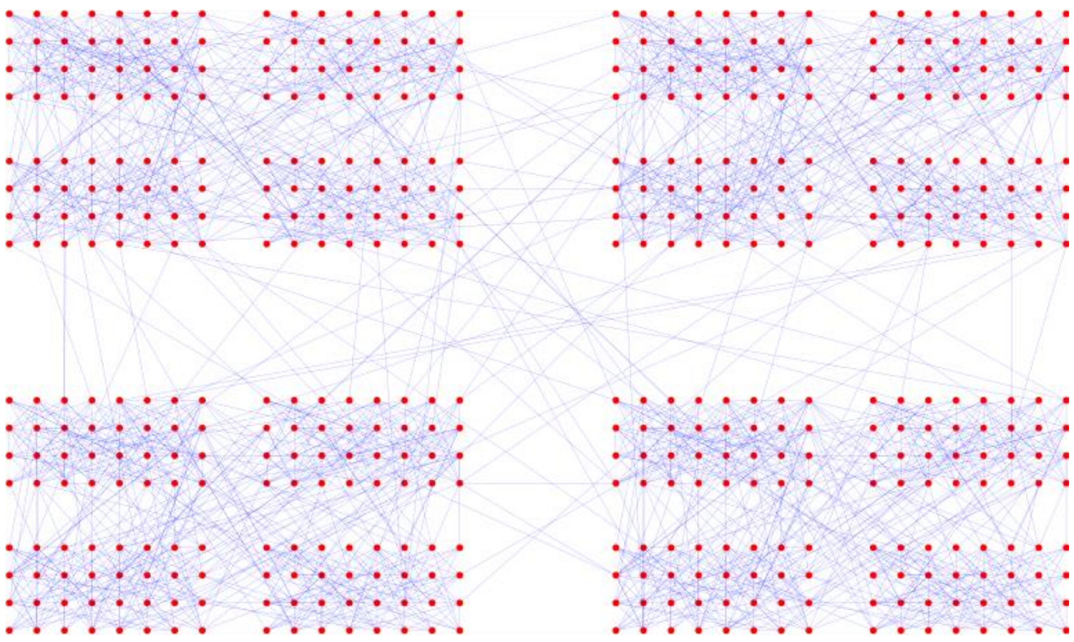


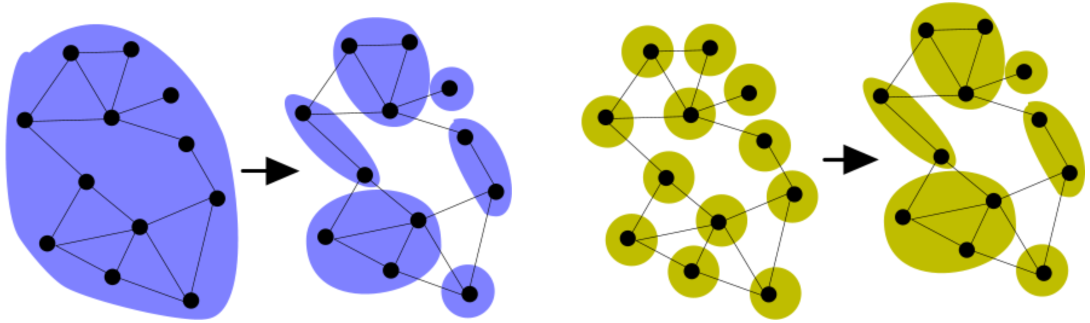
```
partition = nx.community.kernighan_lin_bisection(G)
```

6.2 RELATED PROBLEMS

- Partitions can be hierarchical
  - Subdivisions in companies
  - Classes in school
  - How to detected them?

- **Hierarchical clustering**
  - Main ingredient: **similarity measure**
  - Approaches:
    - Agglomerative hierarchical clustering
    - Divisive hierarchical clustering





6.2 RELATED PROBLEMS

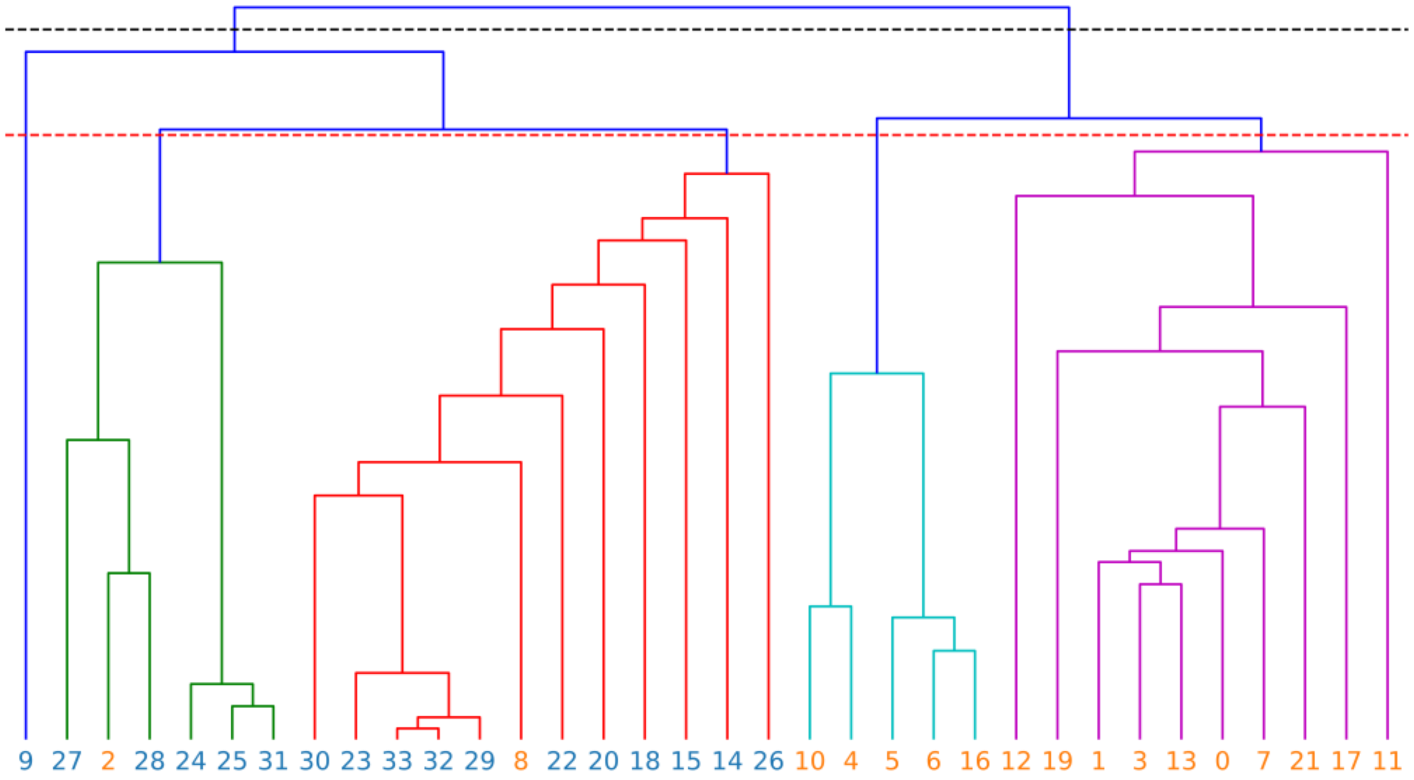
- Example similarity: **structural equivalence**
  - Idea: nodes are similar if their neighbours are similar

$$S_{ij}^{SE} = \frac{\text{number of neighbors shared by } i \text{ and } j}{\text{total number of nodes neighboring only } i\text{, only } j\text{, or both}}$$

- How do define similarity between *partitions*?
  - **Single linkage**: take the maximum pairwise similarity $S_{G_1,G_2} = \max_{i,j} S_{i,j}$
  - **Complete linkage**: take the minimum pairwise similarity $S_{G_1,G_2} = \min_{i,j} S_{i,j}$
  - **Average linkage**: take the average pairwise similarity $S_{G_1,G_2} = \langle S_{ij} \rangle_{i,j}$

6.2 RELATED PROBLEMS

- Result of clustering: **dendrogram**
  - Summary of similarity between nodes
  - Shows the partitioning for each value of number of partitions (horizontal cuts)
  - Each partition includes the partitions lower in the hierarchy

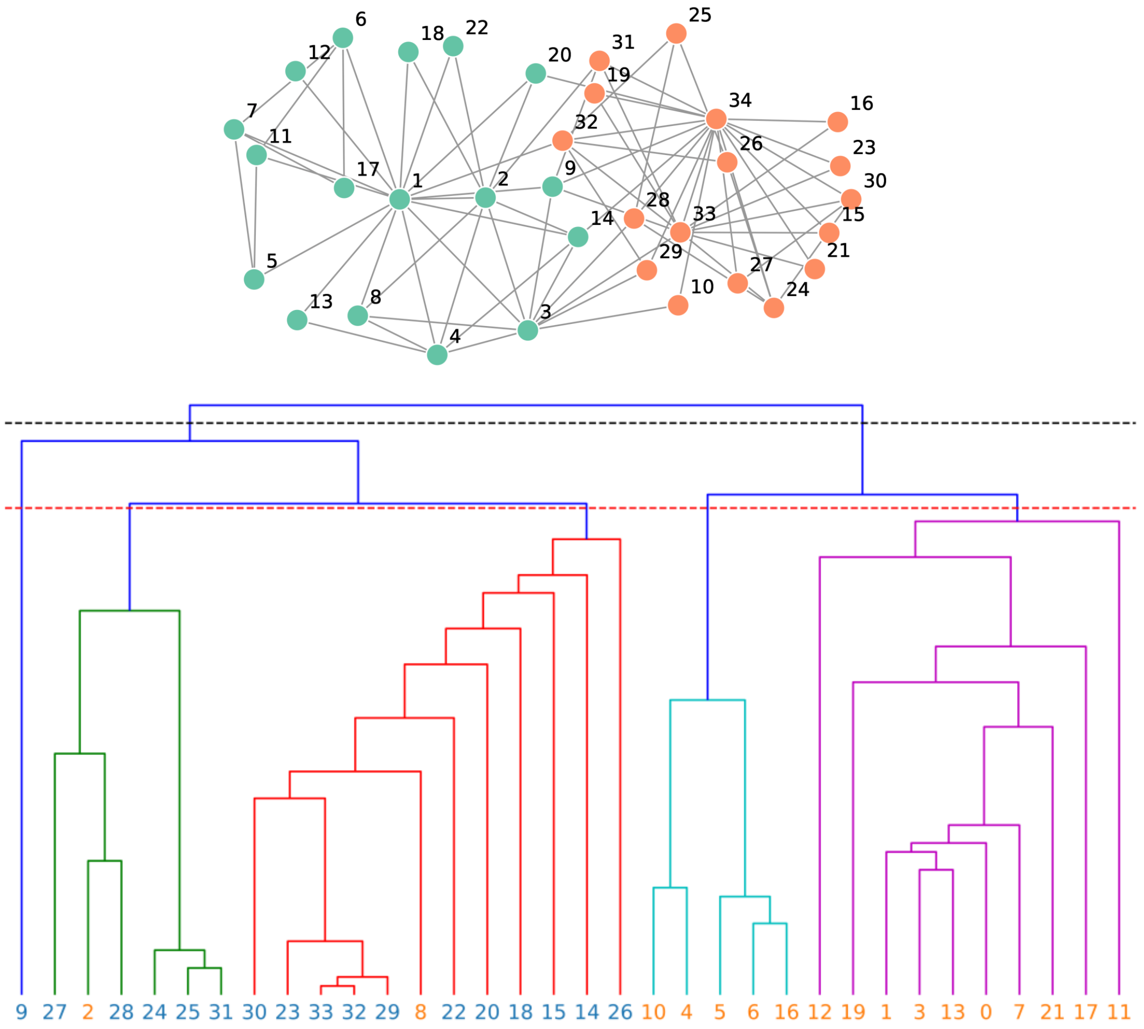


6.2 RELATED PROBLEMS

- Result of clustering: **dendrogram**

- Benefits
  - Clear picture of node affiliation at different levels
  - Different view from the graph

- Caveats
  - No criteria to select ideal partitioning
  - Depends heavily on similarity measure
  - Rather useless for large networks (slow, too many lines)

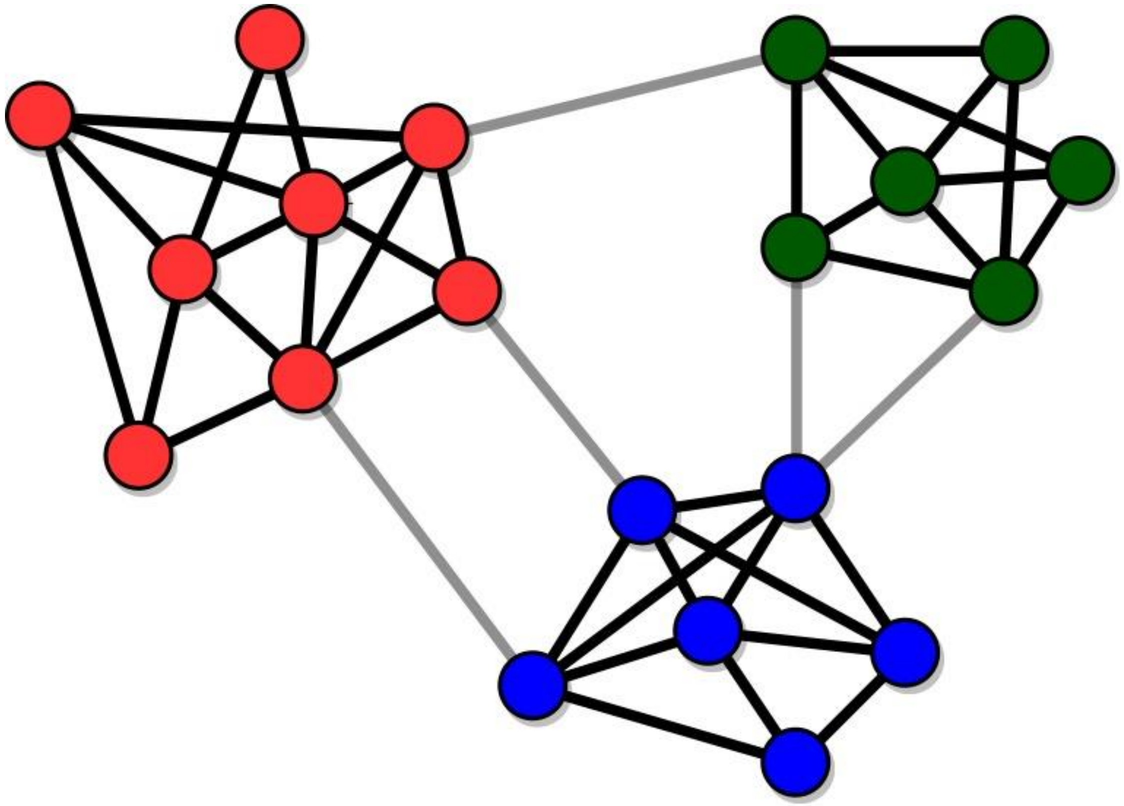


6.3 COMMUNITY DETECTION

- Many different methods

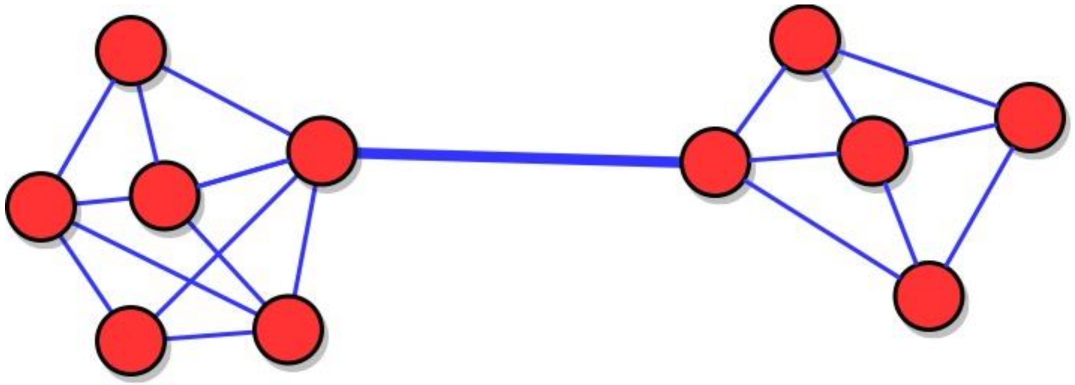
- Today
  - Bridge removal
  - Modularity maximization

- Wednesday:
  - Stochastic block modelling
  - Community validation



6.3 COMMUNITY DETECTION

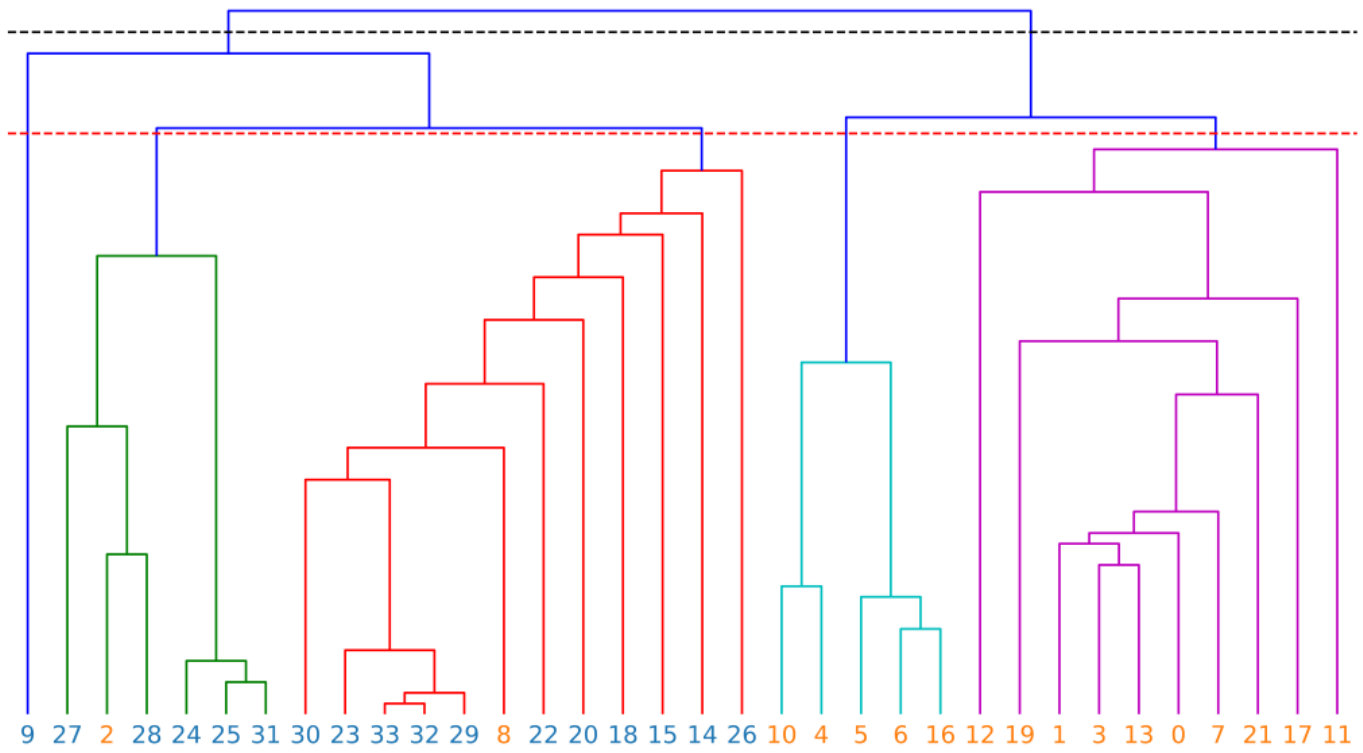
- **Bridge removal**
  - The idea: detects bridges between clusters, removes them until clusters are disconnected
  - Those clusters are communities
  - What is a good bridge metric?



- **Girvan-Newman** algorithm
  - Measure: **link betweenness centrality**
  - Algorithm
    1. Compute betweenness for all links
    2. Removes the link with largest betweenness, recomputes betweenness for all links
    3. Iterate until all links are removed

6.3 COMMUNITY DETECTION

- **Girvan-Newman:** the bad
  - Recalculate betweenness for *all links at each step*: very slow
  - Not feasible for large networks (say, > 10000 nodes)
- **Girvan-Newman:** the good
  - Dendogram
  - If strong community structure, nodes quickly get disconnected
  - Faster variants
    - Computing similarity over a sample
    - Faster similarity than betweenness




```
partition = nx.community.girvan_newman(G)
```

6.3 COMMUNITY DETECTION

- Method finds a partition: how good is it?
  - Quality function
  - Assumption: random networks have no communities
  - Must distinguish between real communities and random fluctuations

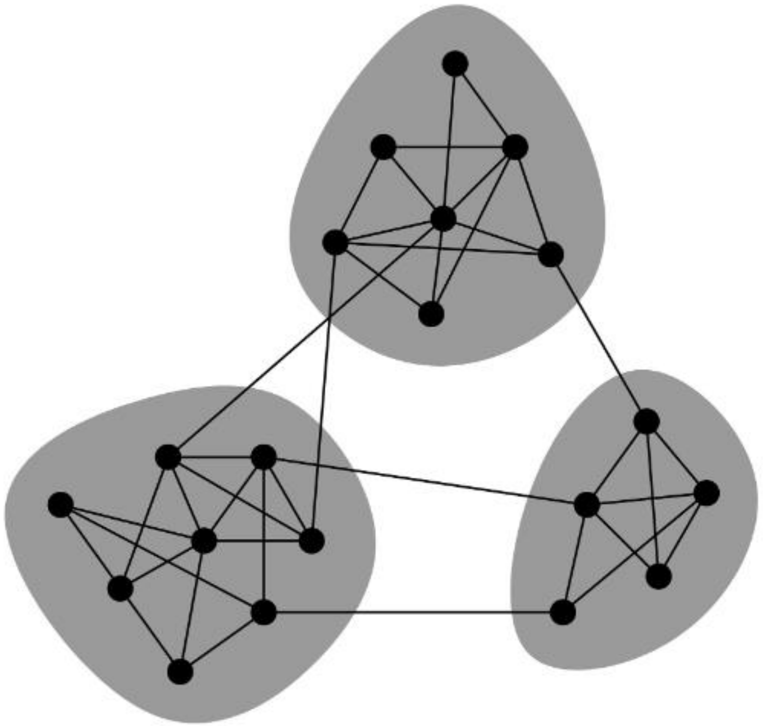
- Most famous quality function: **modularity**
  - The idea: compares community structure against randomized networks with equal degree
  - For each community, computes the **difference in number of internal links** between the network and the ensemble of random networks
  - Value $Q \in [-0.5,1]$

Modularity and community structure in networks

M. E. J. Newman*

Department of Physics and Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109

Edited by Brian Skyrms, University of California, Irvine, CA, and approved April 19, 2006 (received for review February 26, 2006)

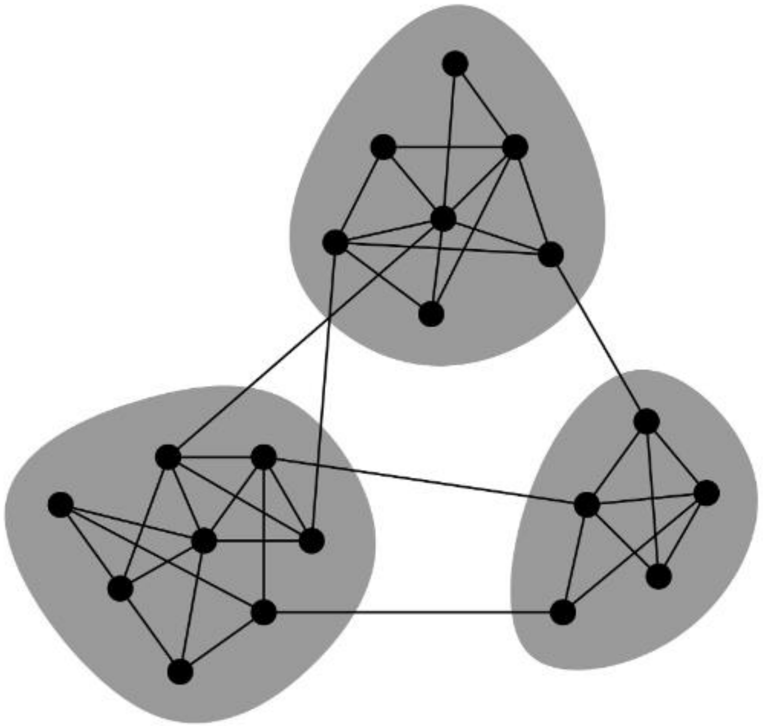


6.3 COMMUNITY DETECTION

- Modularity definition:

$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right)$$

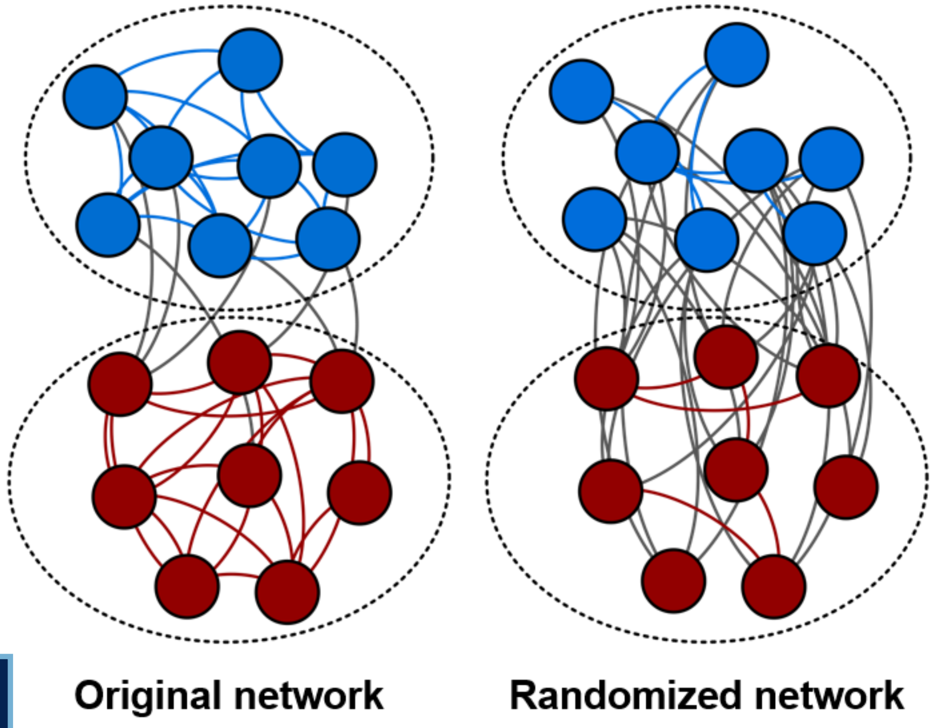
- where
  - L: number of links in the network
  - L_C: number of internal links in community C
  - k_C: degree of community C
  - $k_C^2/4L$: expected number of internal links in community C



6.3 COMMUNITY DETECTION

- Explaining $k_C^2/4L$:
  - Random links by pairing stubs (configuration model)
  - Total number of stubs in C is k_C
  - Probability of selecting a stub is $k_C/2L$
  - Probability of link is $p_C = k_C^2/4L^2$
  - Expected number of internal links is $Lp_C = k_C^2/4L$
- Properties of Q
  - $Q > 0$: some structure
  - $Q = 0$: no structure
  - $Q < 0$ if particularly bad partitioning

$$Q = \frac{1}{L}\sum_C \left(L_C - \frac{k_C^2}{4L}\right)$$




```
modularity = nx.community.quality.modularity(G,partition)
```

6.3 COMMUNITY DETECTION

- Modularity has straightforward extensions
- Directed:

$$Q_d = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^{in} k_C^{out}}{L} \right)$$

- Weighted:

$$Q_w = \frac{1}{W} \sum_C \left(W_C - \frac{s_C^2}{4W} \right)$$

- Directed and weighted:

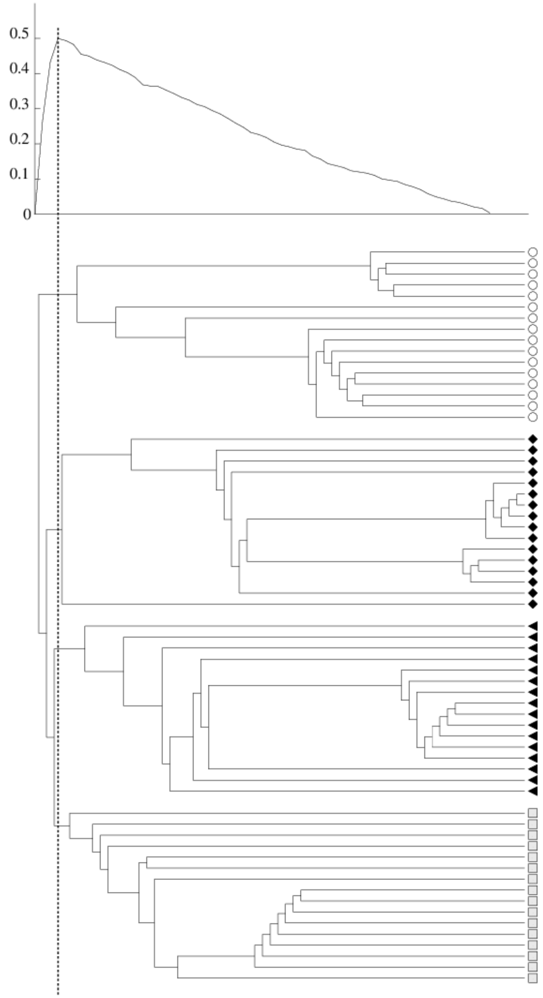
$$Q_{dw} = \frac{1}{W} \sum_C \left(W_C - \frac{s_C^{in} s_C^{out}}{W} \right)$$

6.3 COMMUNITY DETECTION

- Modularity maximization
  - Finding partitioning that maximizes Q

- **Newman's greedy algorithm**
  1. Each node in its community (agglomerative)
  2. Merge nodes that yield the largest increase in Q
  3. Continue until single community
  4. Pick partition with largest Q

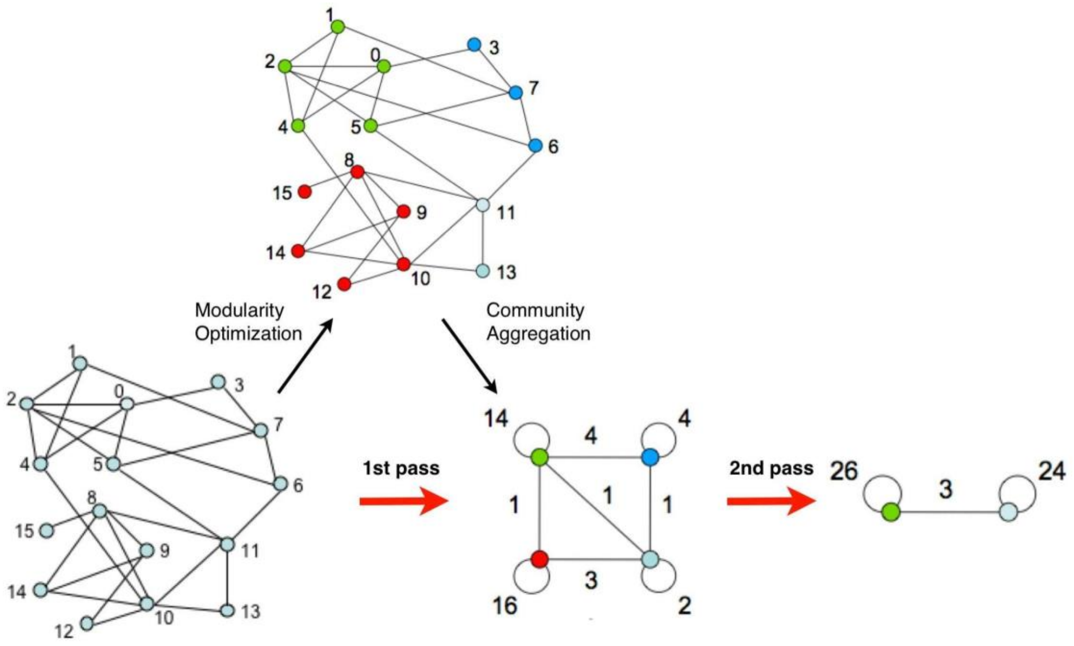
```
partition = nx.community.greedy_modularity_communities(G)
```



pip install python-louvain

6.3 COMMUNITY DETECTION

- Newman's greedy algorithm limitations
  - greedy: gets stuck on local minima
  - tends to generate unbalanced networks
  - Slow
- Standard modularity algorithm: **Louvain algorithm**
  - Similar to Newman's
  - **Very fast**
  - In python




```
#pip install python-Louvain
partition = community.community_louvain.best_partition(G)
```

6.3 COMMUNITY DETECTION

- Modularity maximization is widely used
- Has a few problems
  - **Resolution limit**: small communities are undetected
  - Fails the random network test
- Solution: stochastic blockmodels

SUMMARY

- Community detection is a huge part of network theory
- Best method depends on community definition
- Dendogram is useful for small networks
- Modularity is very popular, but comes with caveats