Introduction to Data Mining

Lecture #19: Analysis of Large Graphs

U Kang
Seoul National University

In This Lecture

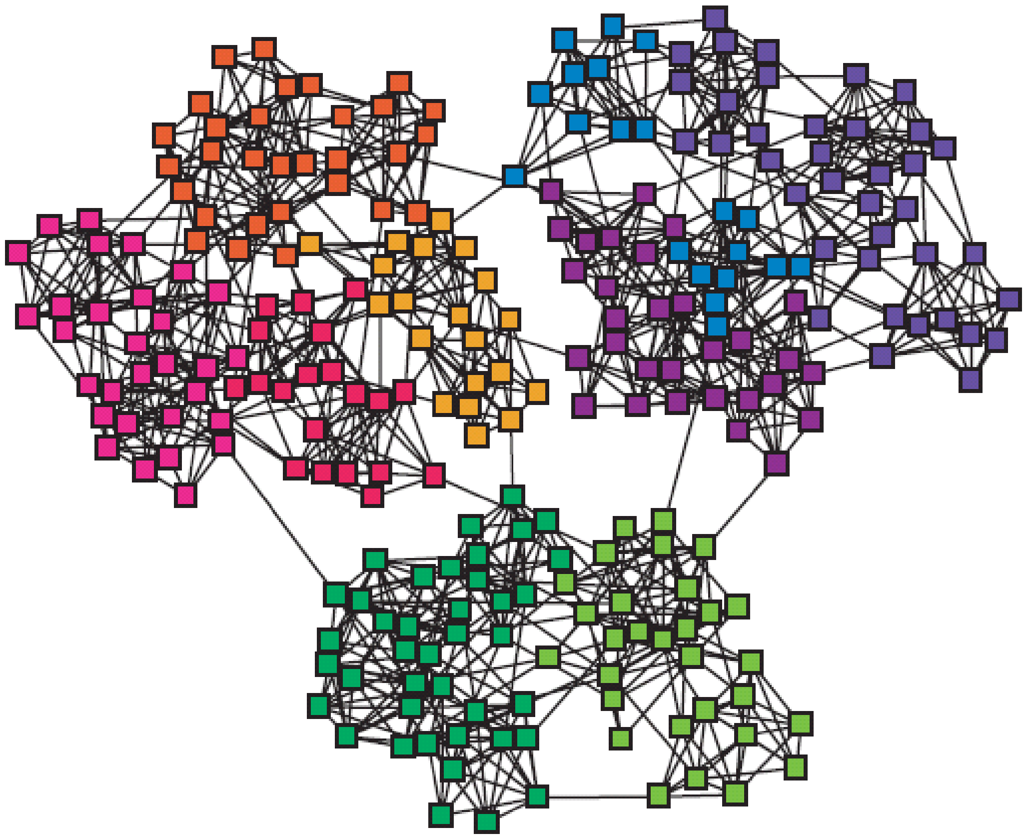
- Understand the definition and the motivation of community detection problem
- Learn the Girvan-Newman algorithm for community detection
- Learn how graph theories (connected component, betweenness, BFS, modularity) are applied for data mining

Outline

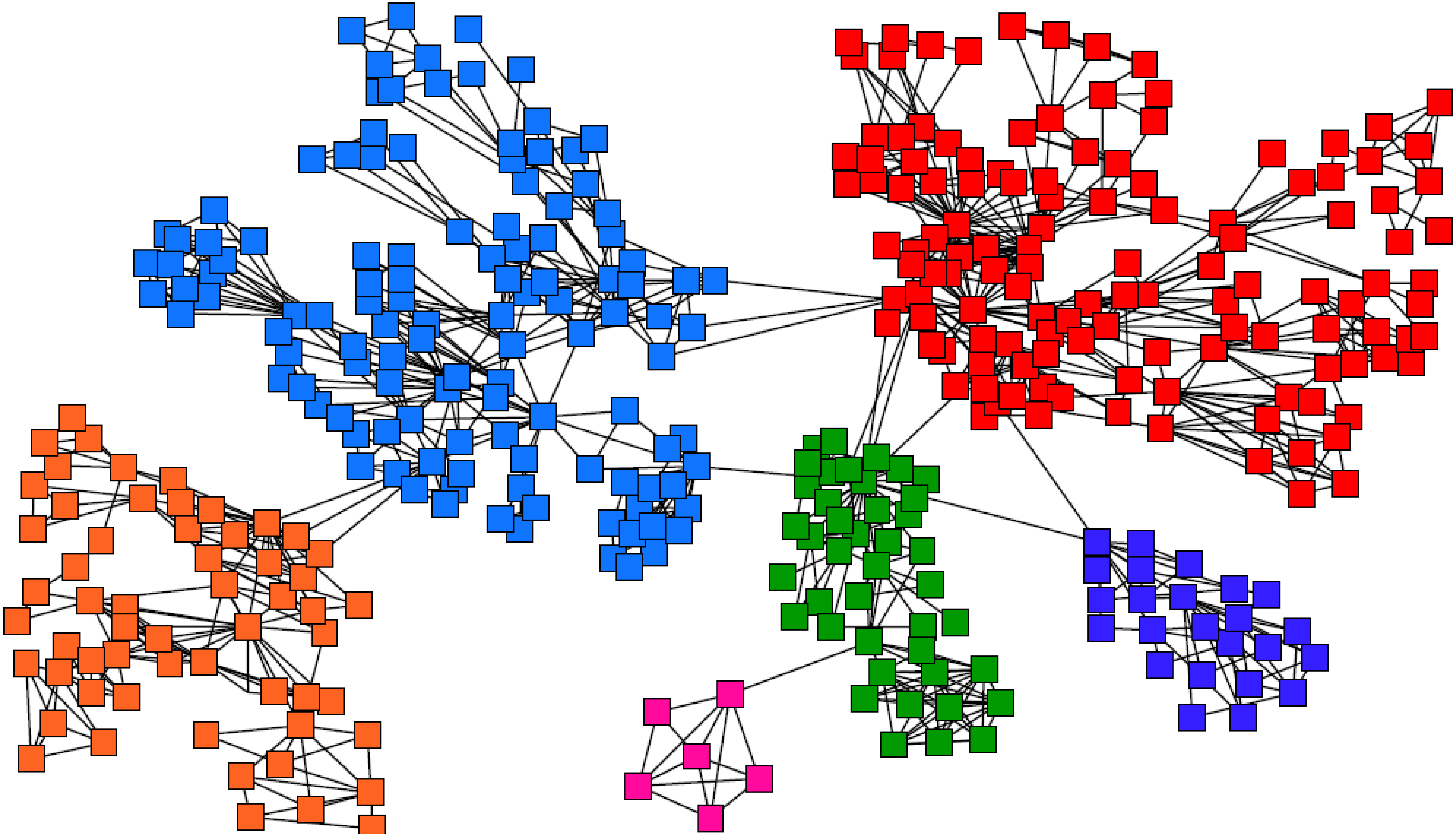
- ☐ **Overview**
- ☐ Community Detection

Networks & Communities

- **We often think of networks being organized into modules, cluster, communities:**

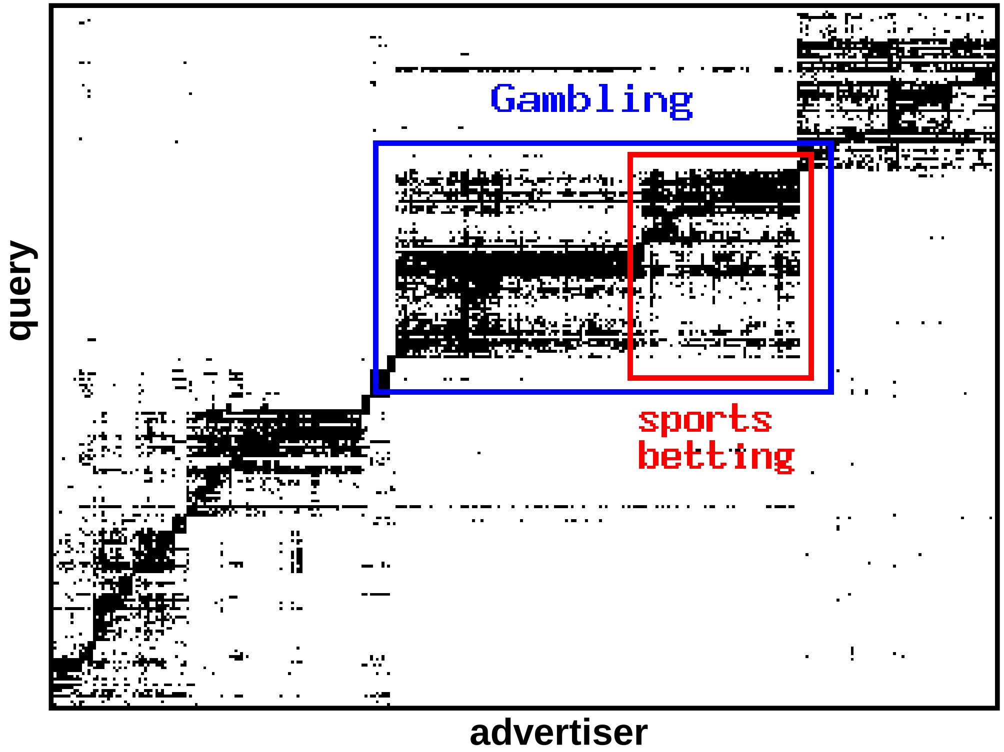


Goal: Find Densely Linked Clusters



Micro-Markets in Sponsored Search

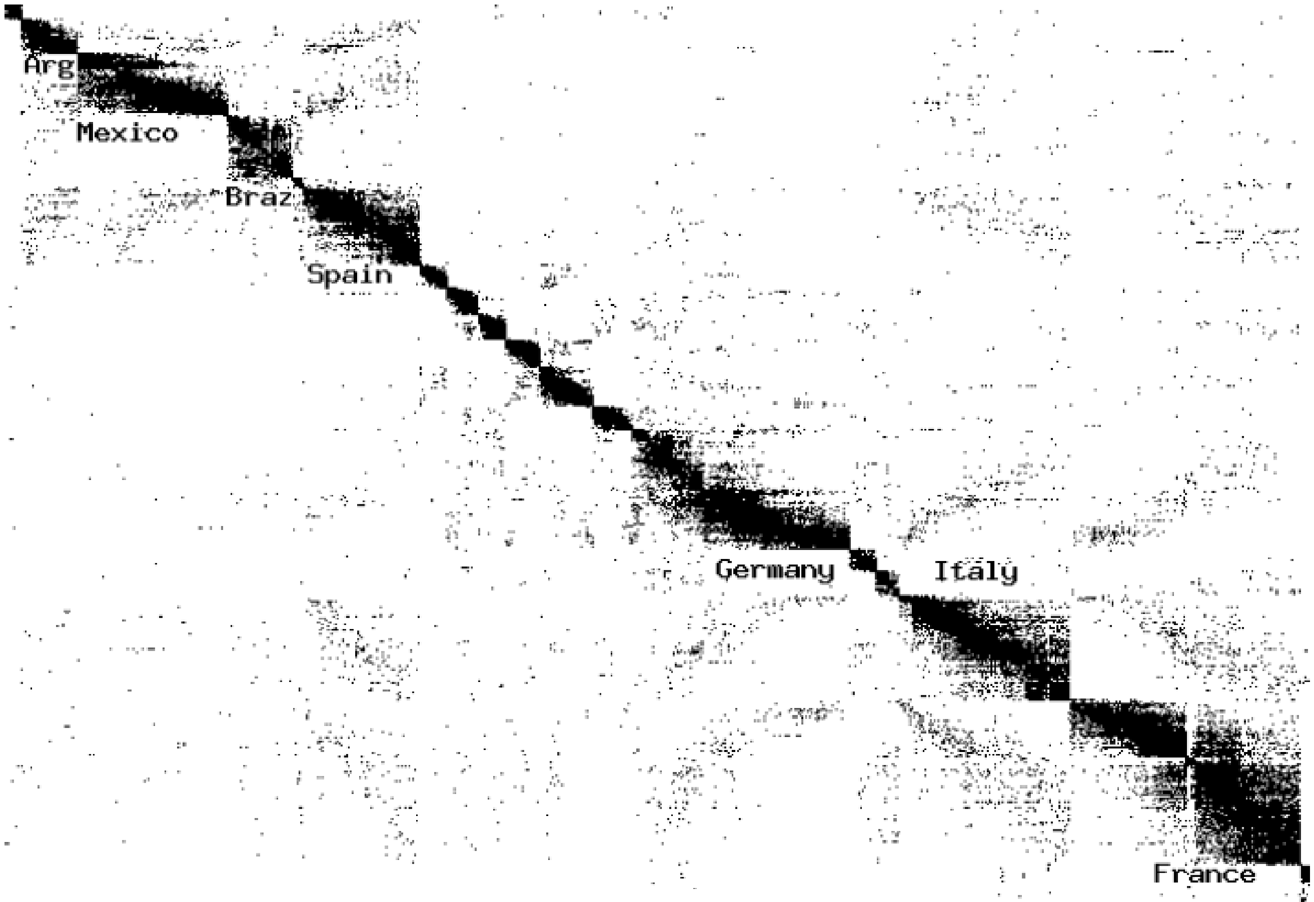
- **Find micro-markets by partitioning the query-to-advertiser graph:**



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

Movies and Actors

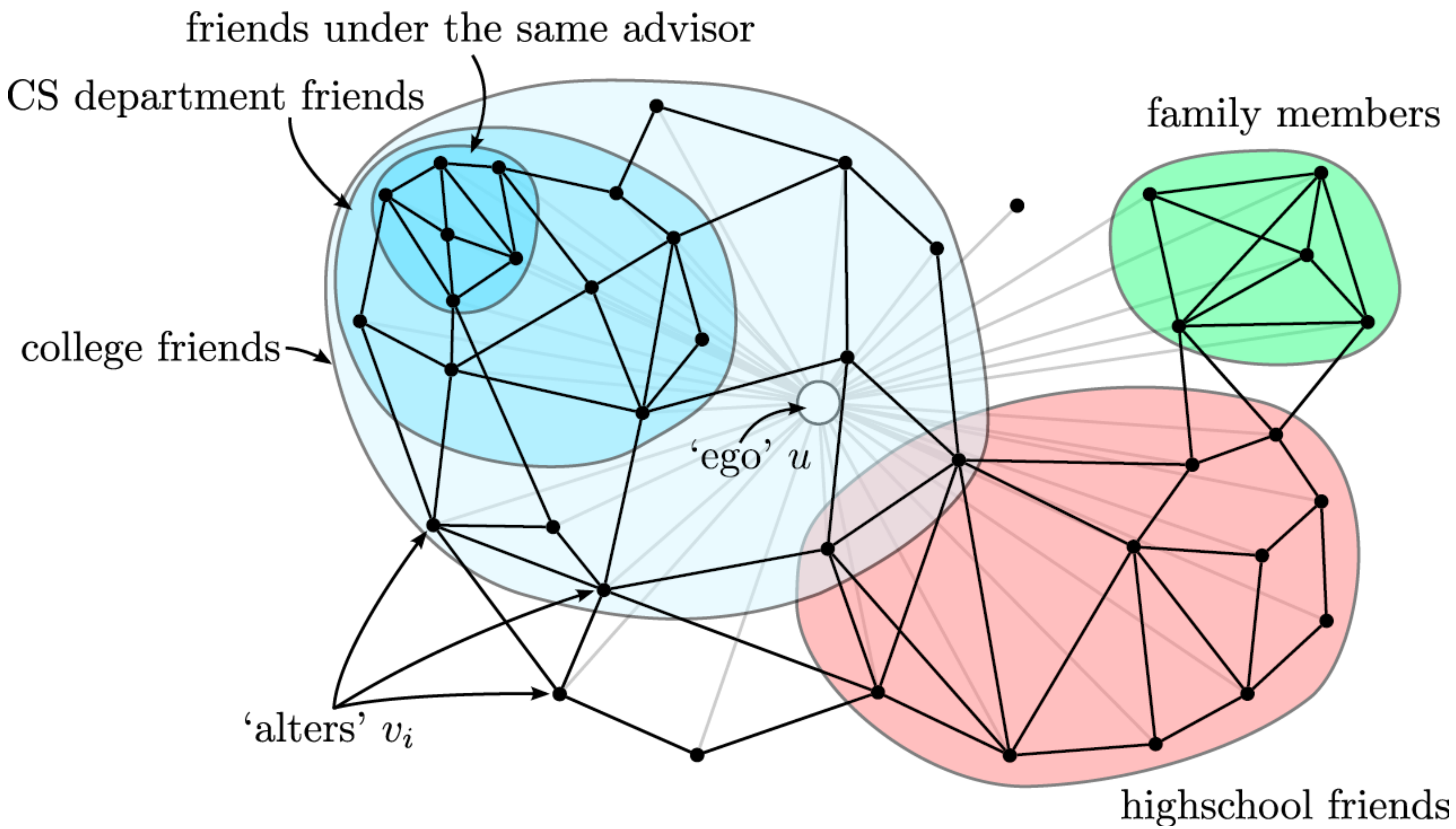
- **Clusters in Movies-to-Actors graph:**



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

Twitter & Facebook

- **Discovering social circles, circles of trust:**



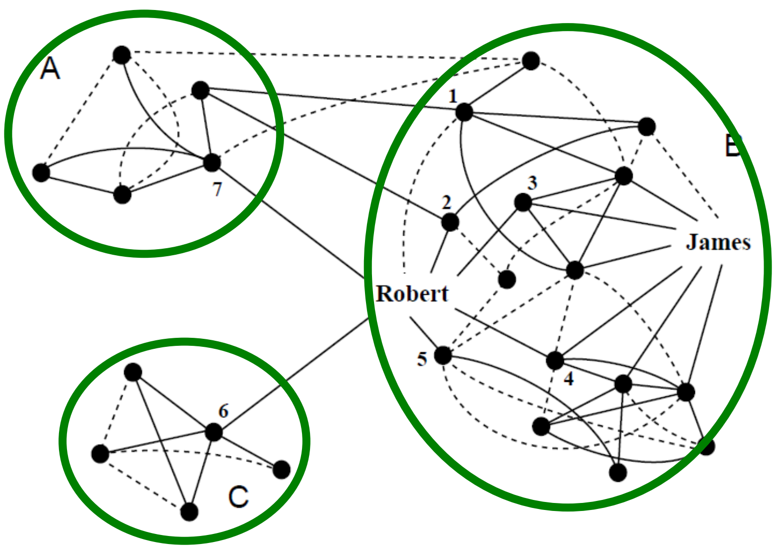
[McAuley, Leskovec: Discovering social circles in ego networks, 2012]

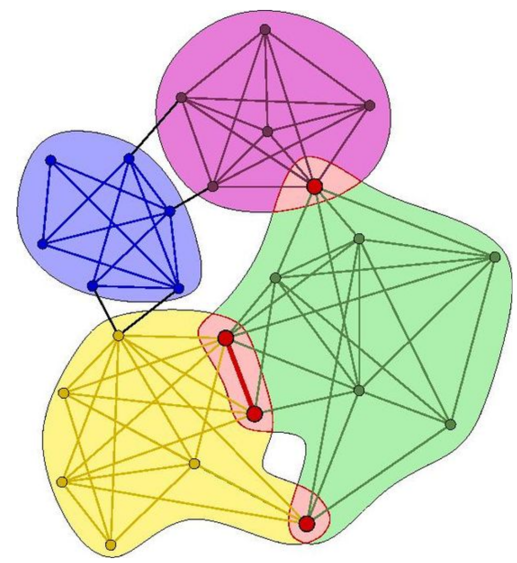
Outline

- ☑ Overview
- ☐ **Community Detection**

Community Detection

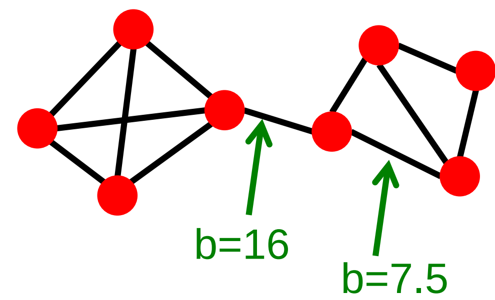
- **How to find communities?**



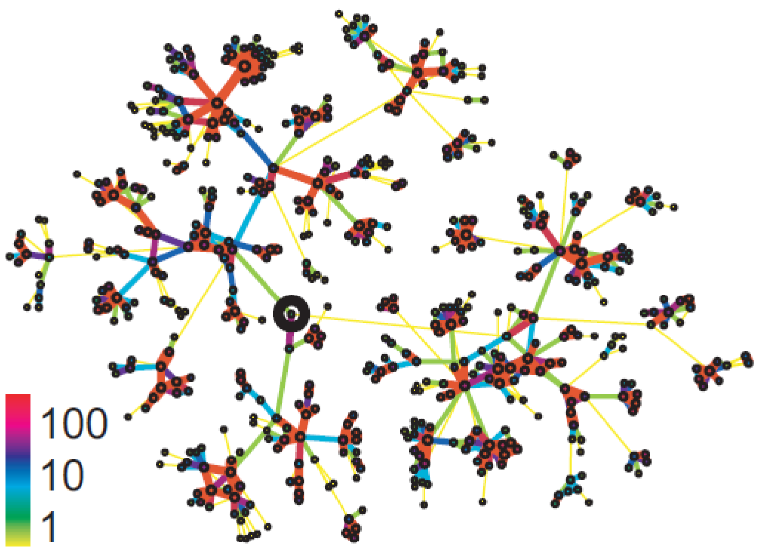


Method 1: Strength of Weak Ties

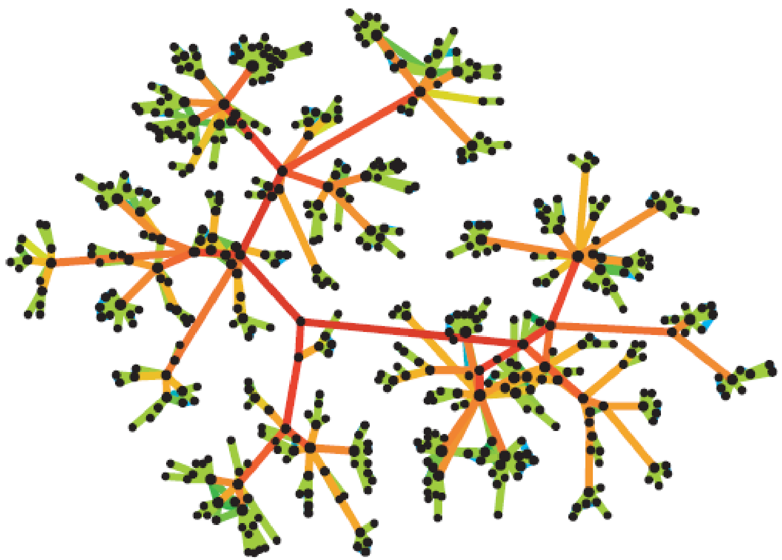
- **Edge betweenness: Number of shortest paths passing over the edge**



- **Intuition:**



Edge strengths (call volume) in a real network



Edge betweenness in a real network

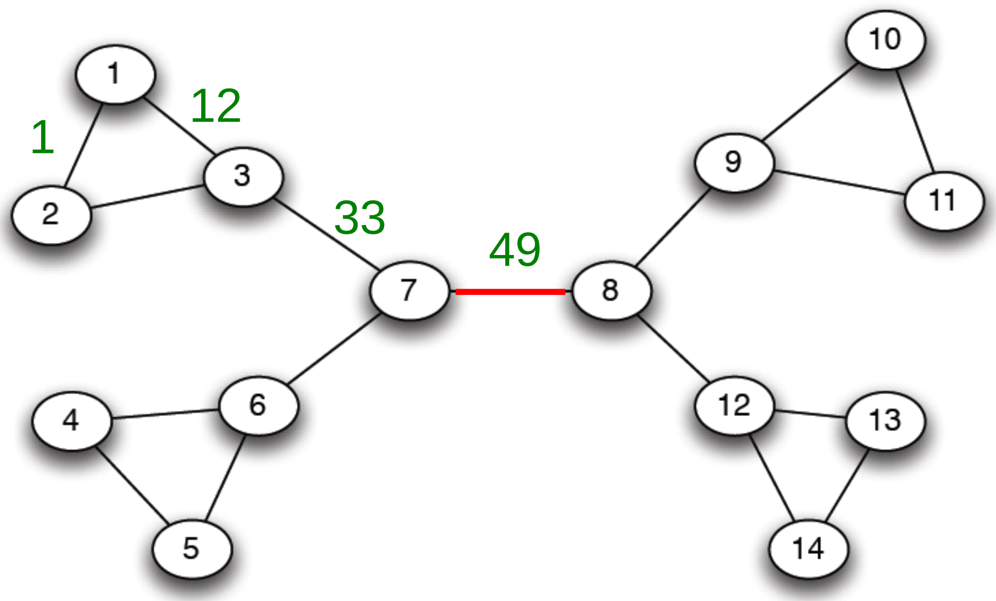
Method 1: Girvan-Newman

- 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

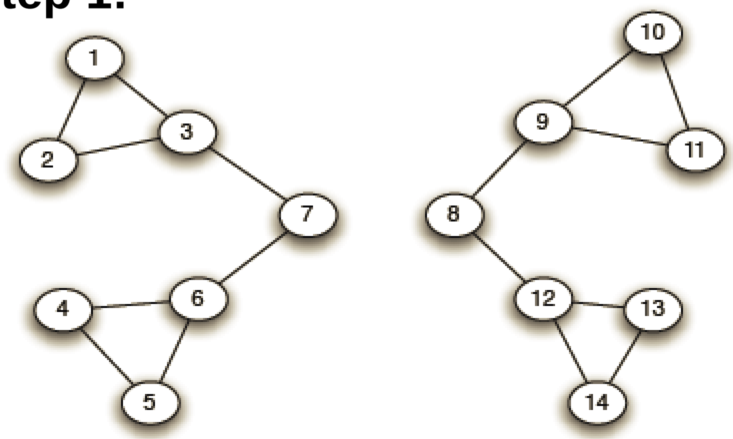
Girvan-Newman: Example



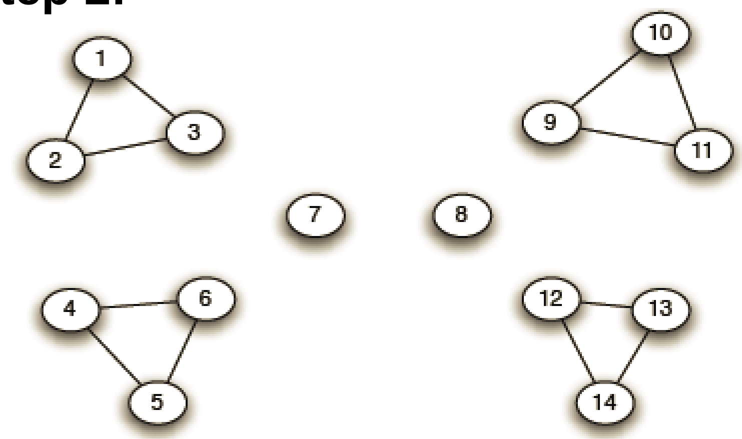
Need to re-compute betweenness at every step

Girvan-Newman: Example

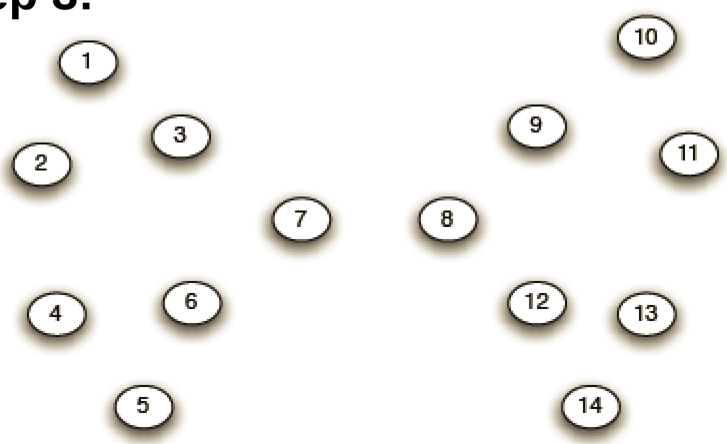
Step 1:



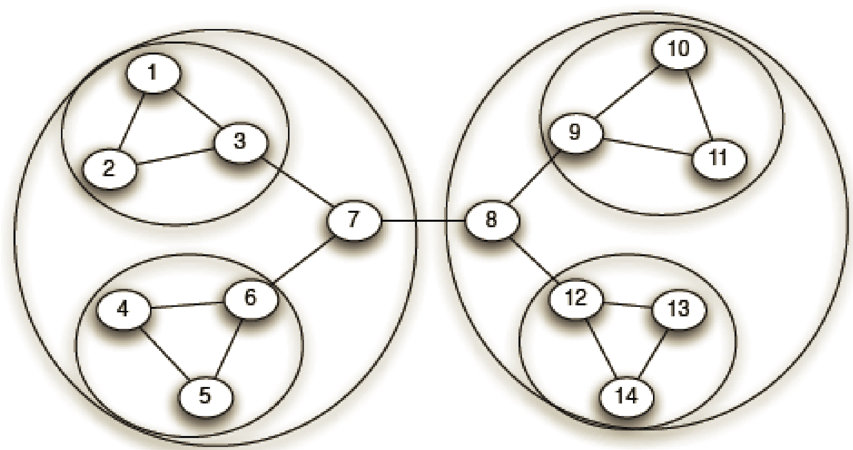
Step 2:



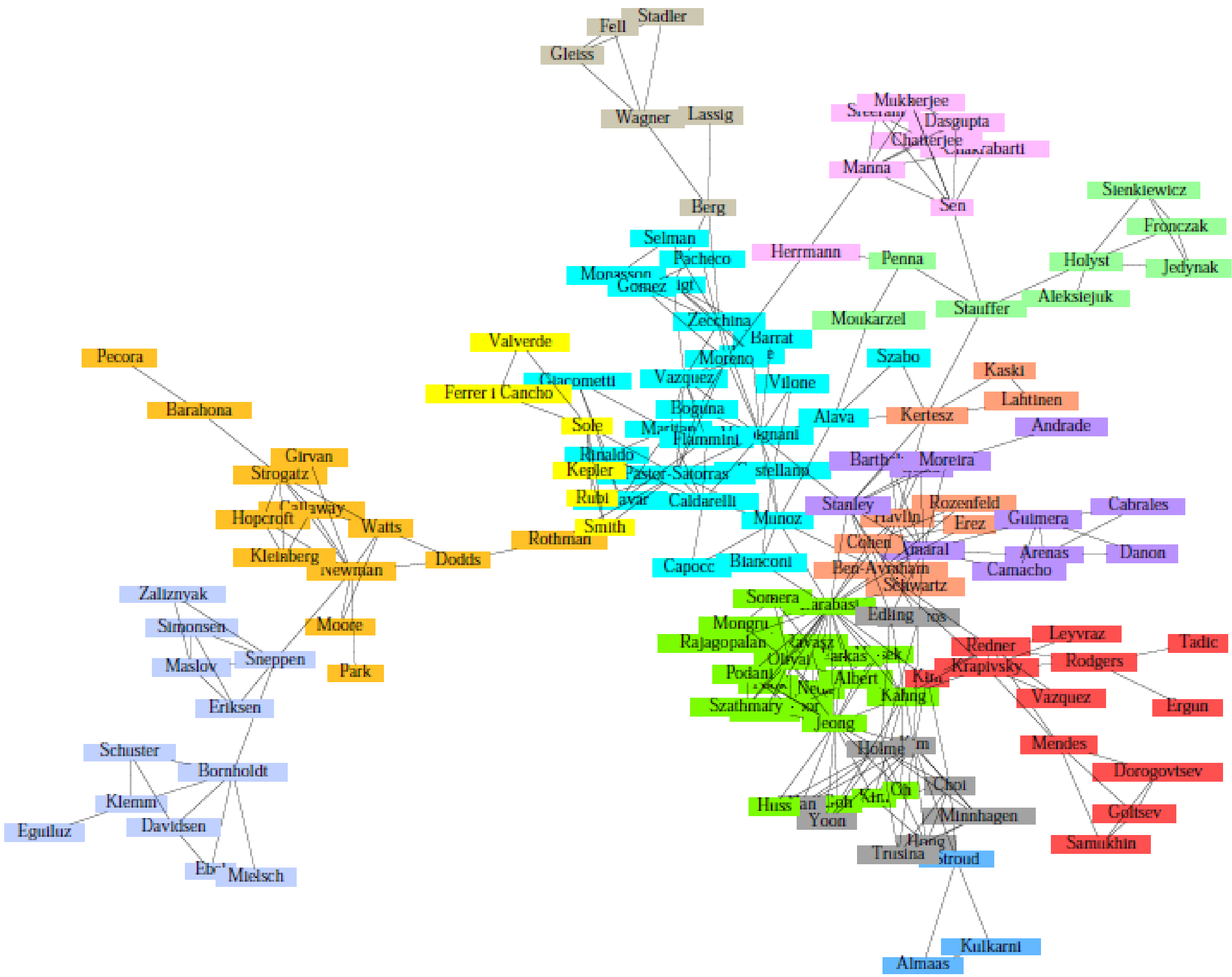
Step 3:



Hierarchical network decomposition:



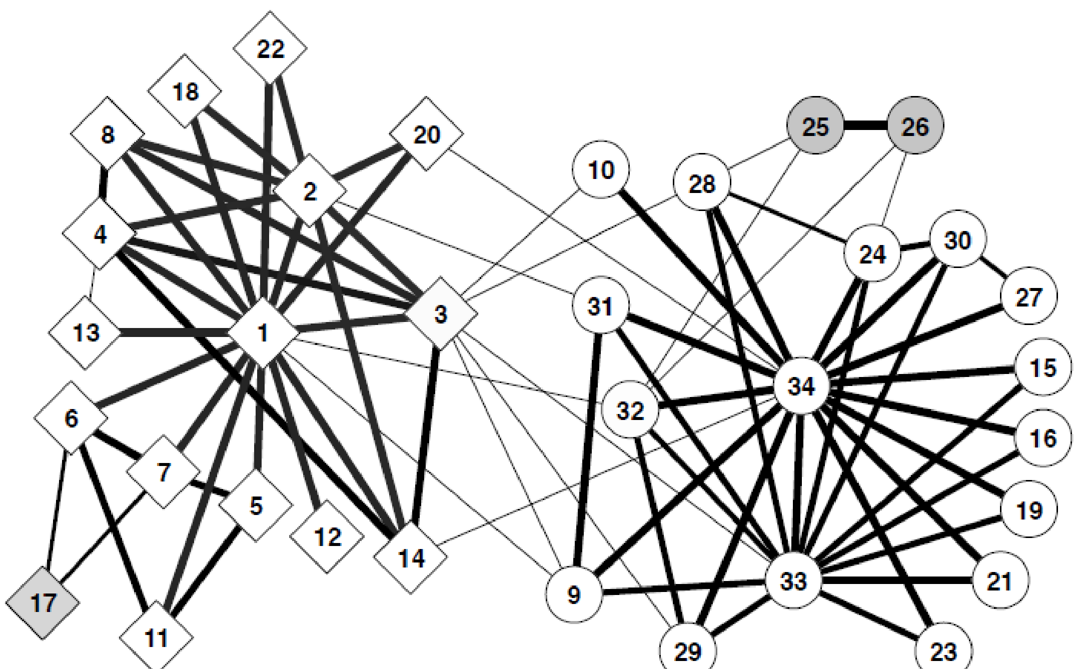
Girvan-Newman: Results

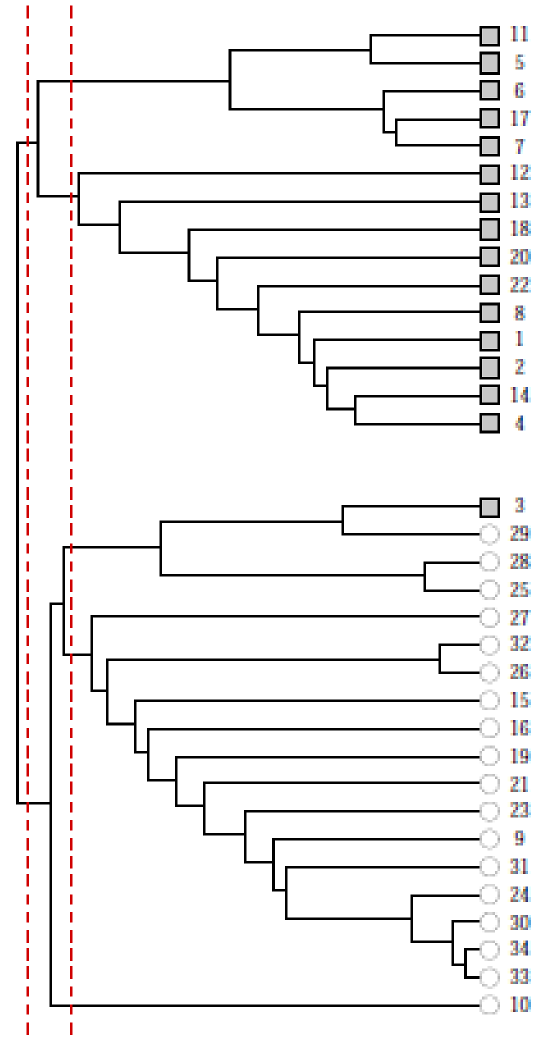


Communities in physics collaborations

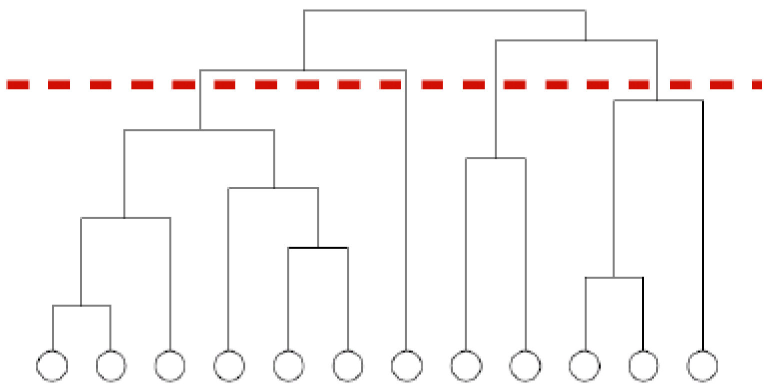
Girvan-Newman: Results

- **Zachary's Karate club:** Hierarchical decomposition





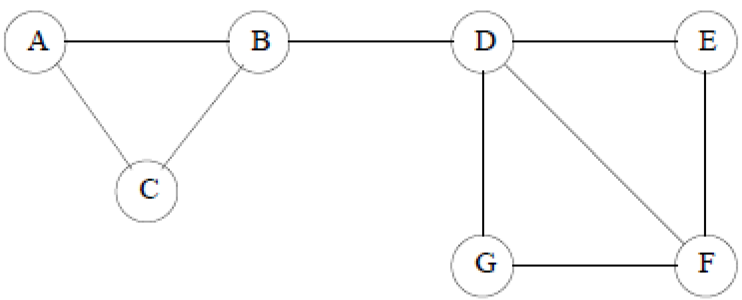
WE NEED TO RESOLVE 2 QUESTIONS



1. **How to compute betweenness?**
2. **How to select the number of clusters?**

How to Compute Betweenness?

- **Want to compute betweenness of paths starting at node E**



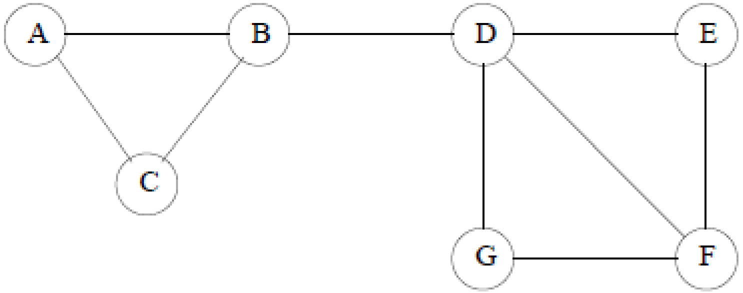
Note: if we do this for all the starting nodes, and divide by 2, we get the betweenness

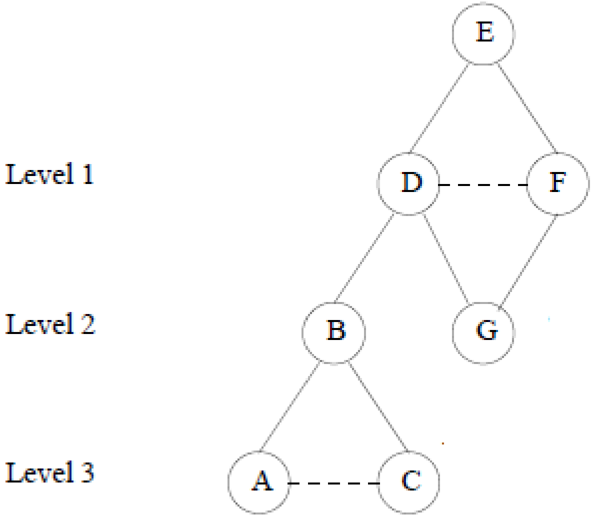
- **Overview**
  - **Step 1: construct the BFS tree starting from E**
  - **Step 2: label each node by the # of shortest paths from the root E**
  - **Step 3: for each edge, calculate the sum (over all nodes Y) of the fraction of shortest paths from E to Y**

(BFS: Breadth First Search)

How to Compute Betweenness?

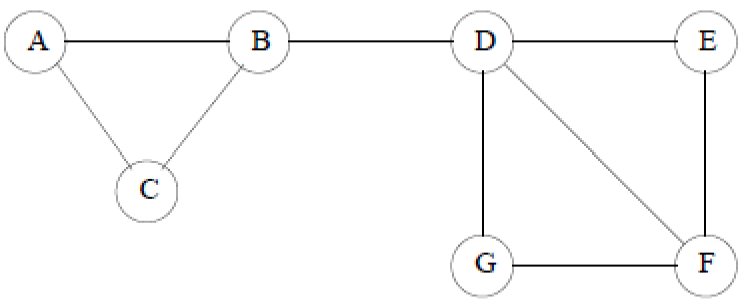
- **Step 1: construct the BFS (breadth-first-search) tree starting from E**

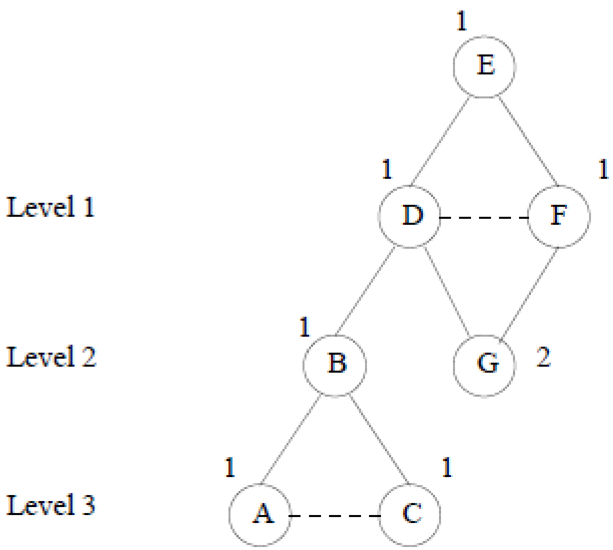




How to Compute Betweenness?

- **Step 2: label each node by the # of shortest paths from the root E**

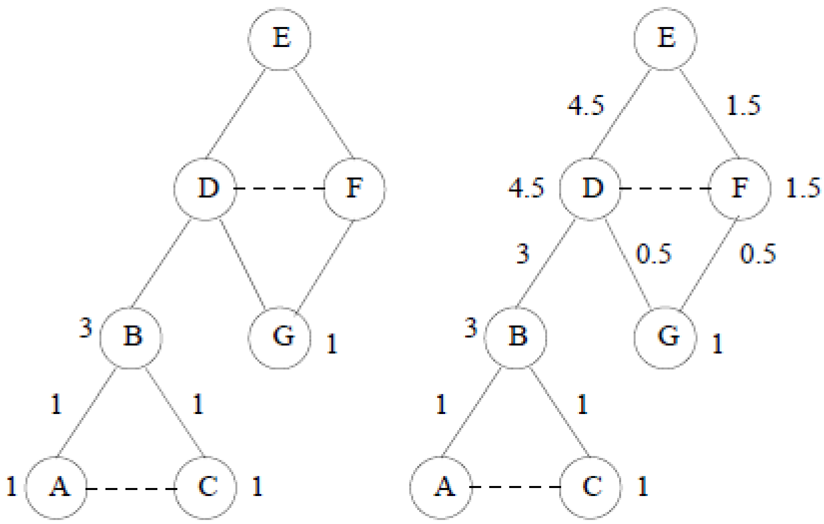




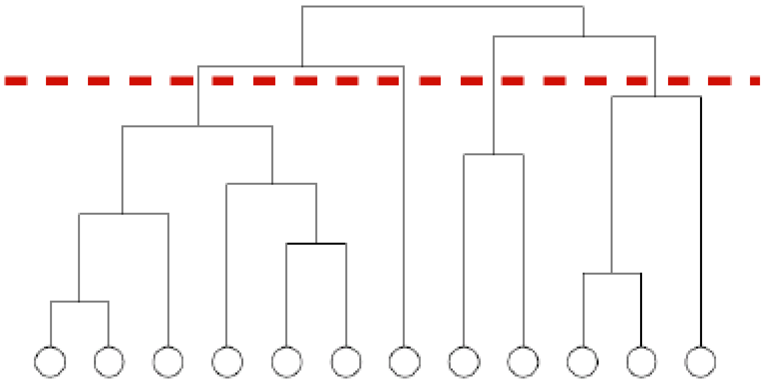
How to Compute Betweenness?

- **Step 3: for each edge, calculate the sum (over all nodes Y) of the fraction of shortest paths from E to Y**

  - **Leaf node gets credit 1**
  - **Each non-leaf node gets a credit (1 + sum of child edges' credits)**
  - **Each node's credit is sent to the edges above it (divide by labels from step 2)**



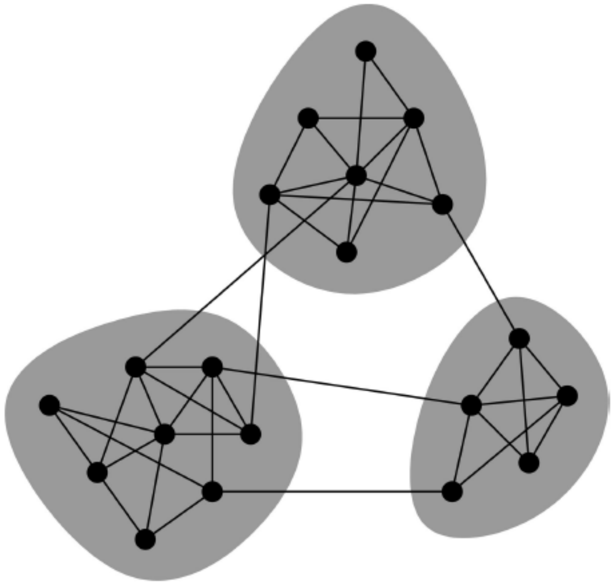
WE NEED TO RESOLVE 2 QUESTIONS



1. **How to compute betweenness?**
2. **How to select the number of clusters?**

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 $s \in S$:

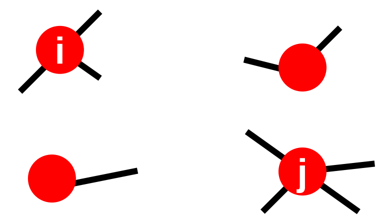


$$Q \propto \sum_{s \in S} [\ (\#\ \text{edges within group } s) - (\text{expected } \#\ \text{edges within group } s)\ ]$$

Need a null model!

Null Model: Configuration Model

- **Given real G on n nodes and m edges, construct rewired network G'**
  - Same degree distribution but random connections
  - Consider G' as a multigraph
  - **The expected number of edges between nodes i and j** of degrees k_i and k_j equals to: $k_i \cdot \frac{k_j}{2m} = \frac{k_i k_j}{2m}$
    - Check - the expected number of edges in (multigraph) **G'**:
      - $= \frac{1}{2}\sum_{i \in N}\sum_{j \in N}\frac{k_i k_j}{2m} = \frac{1}{2} \cdot \frac{1}{2m}\sum_{i \in N} k_i \left(\sum_{j \in N} k_j\right) =$
      - $= \frac{1}{4m} 2m \cdot 2m = m$



Note:
$\sum_{u \in N} k_u = 2m$

Modularity

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

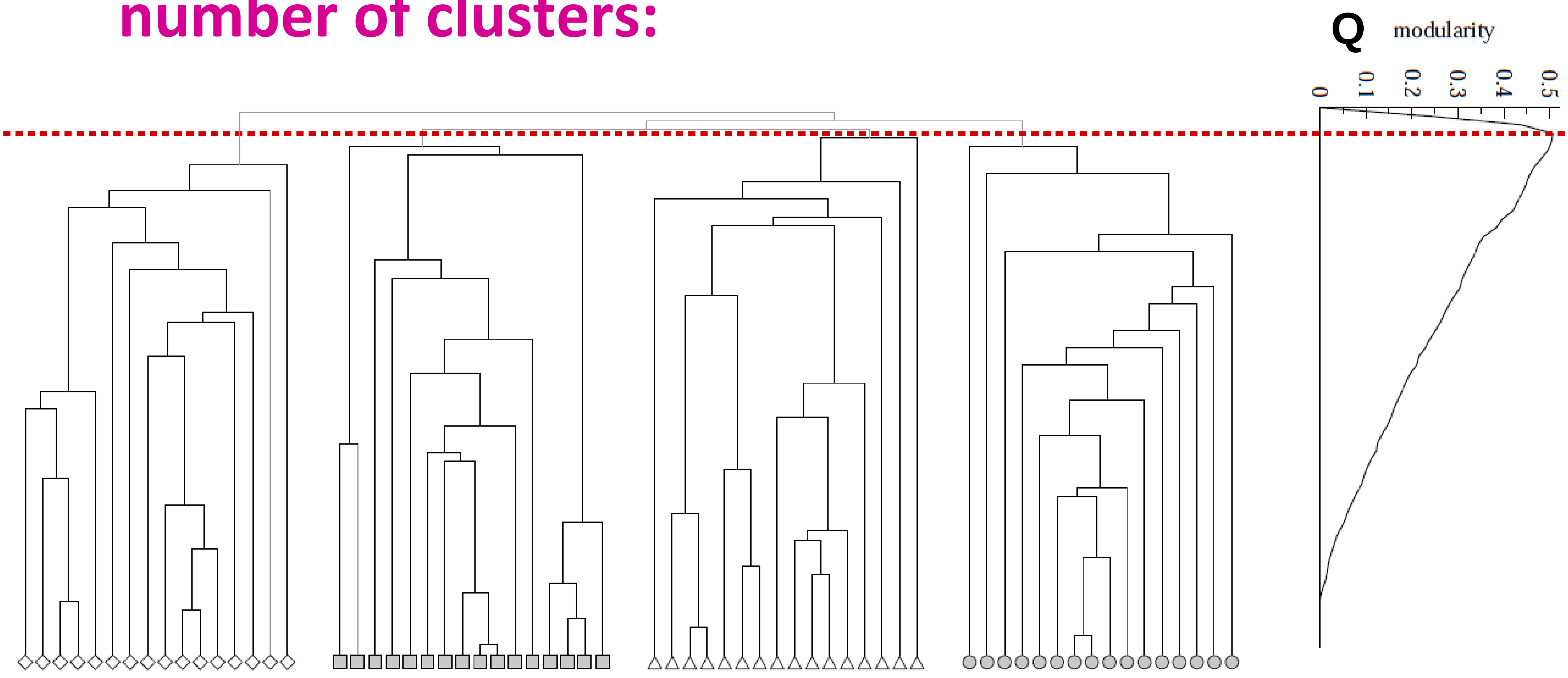
Normalizing cost.: -1<Q<1

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
  - **0.3-0.7<Q** means significant community structure

Modularity: Number of clusters

- **Modularity is useful for selecting the number of clusters:**



Questions?