Machine Learning with Graphs (MLG)

Network Community Detection (2)

Edge Removal-based Algorithms

Cheng-Te Li (李政德)

Institute of Data Science

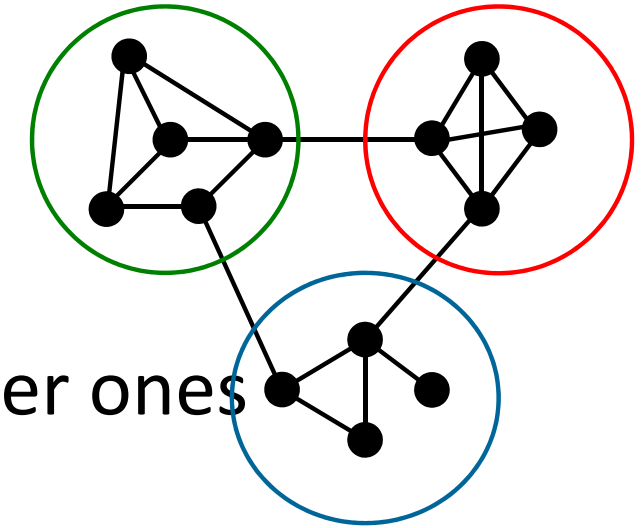
National Cheng Kung University

chengte@mail.ncku.edu.tw

Community Detection Approaches

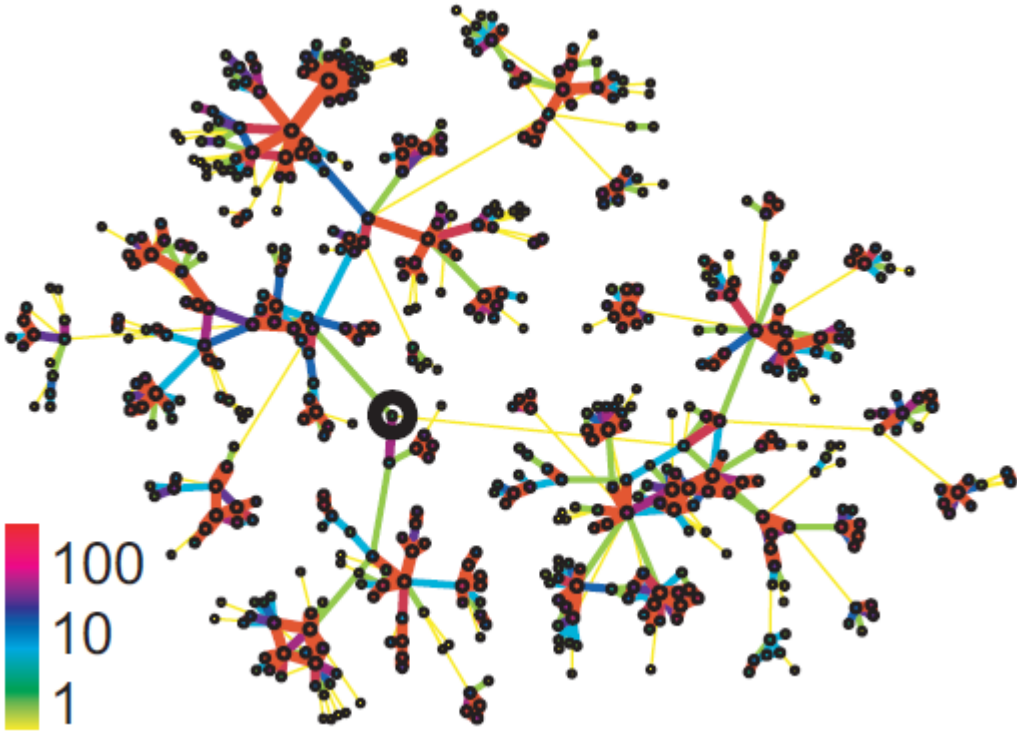
- Propagation-based Method
  - Structural Clustering Algorithm for Networks (SCAN)
- Edge-Removal
  - Girvan-Newman Algorithm (GNA)
  - Fast Newman Algorithm
- Louvain Algorithm
- Label Propagation Algorithm

Edge Removal Approach

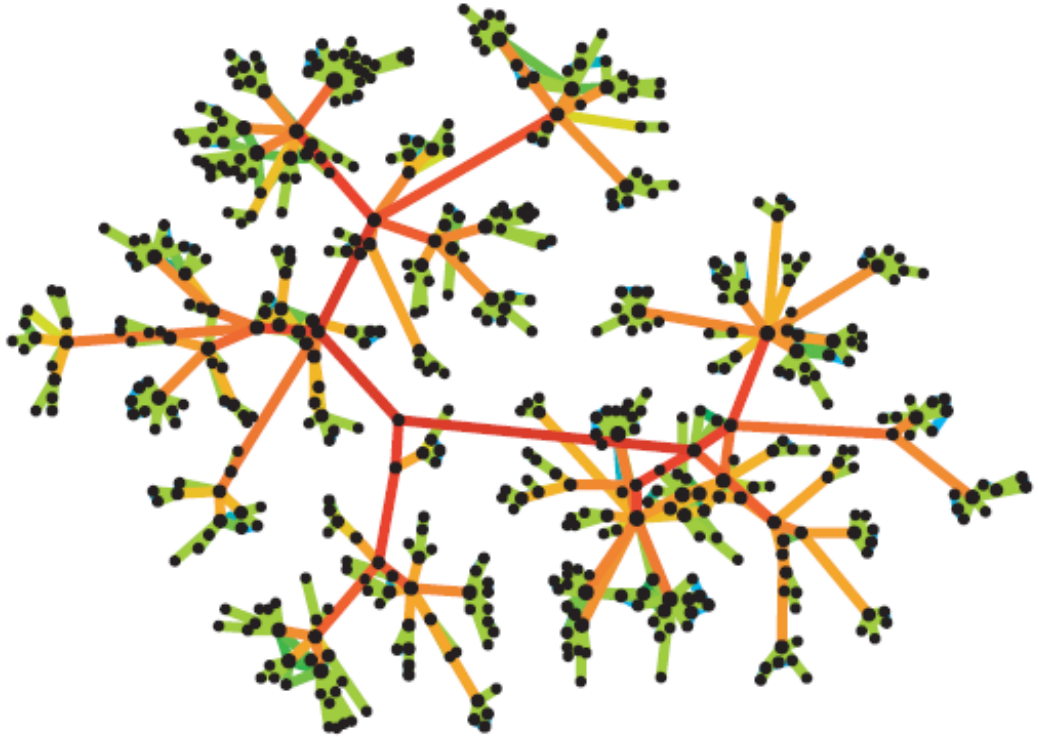


- Basic Idea
  - Partition nodes into several sets
  - Each set is further divided into smaller ones
- Recursively remove the “weakest” tie
  - Step-1: Find the edge with the least strength
  - Step-2: Remove the edge and update the strength of each edge
- Redo step 1-2 until a network is decomposed into desired number of connected components
  - Each component forms a community

Strength of Weak Ties



Edge strengths (call volume) in a real network



Edge betweenness in a real network

Edge Betweenness

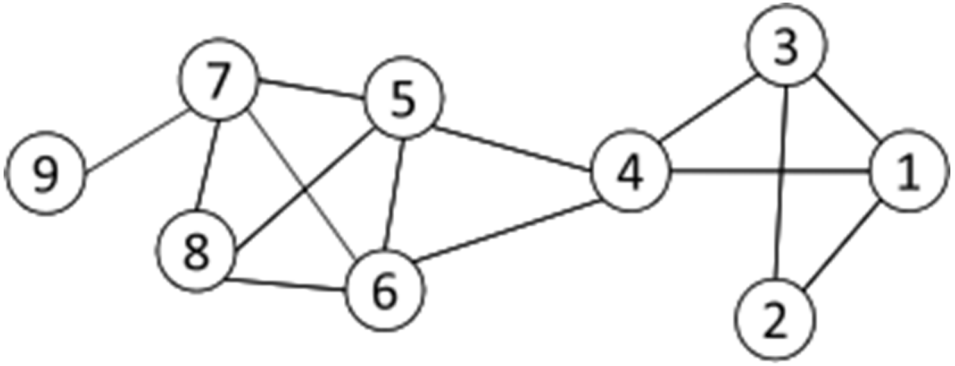
$$C_B(i) = \sum_{s,t \in V} \frac{\sigma(s,t|i)}{\sigma(s,t)}$$

$\sigma(s,t|i)$ = the number of shortest paths connecting nodes s and t passing through node i

$\sigma(s,t)$ = the total number of shortest paths connecting s and t

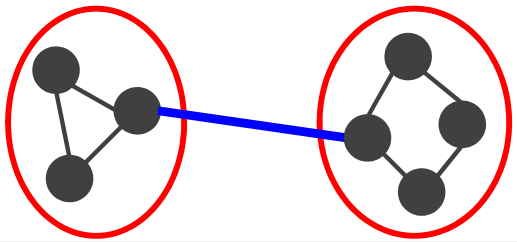
- Measure the strength of an edge
- Edge Betweenness: the number of shortest paths that pass along with the edge

$$C_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}$$



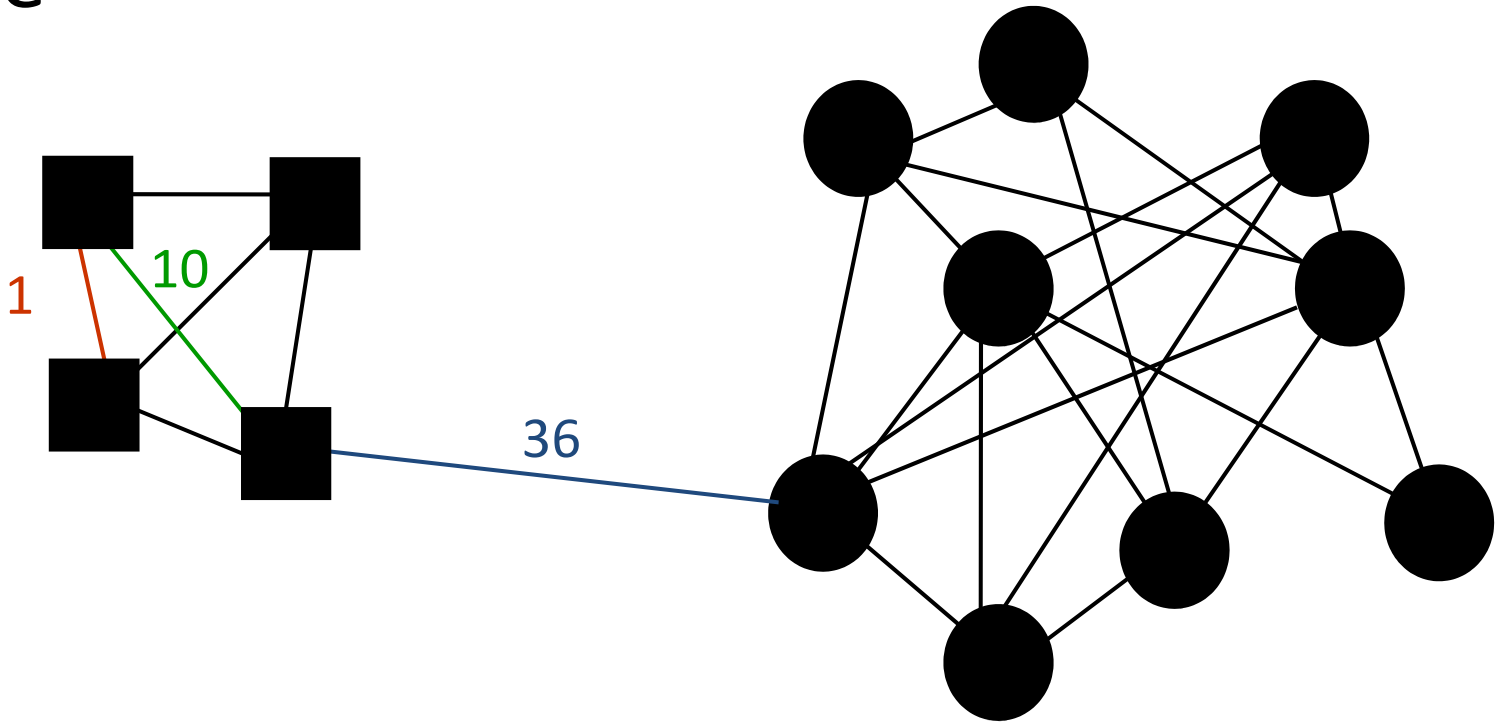
The edge betweenness of e(1, 2) is 4 (=6/2 + 1), as all the shortest paths from 2 to {4, 5, 6, 7, 8, 9} have to either pass e(1, 2) or e(2, 3), and e(1,2) is the shortest path between 1 and 2

- The edge with higher betweenness tends to be the bridge between two communities

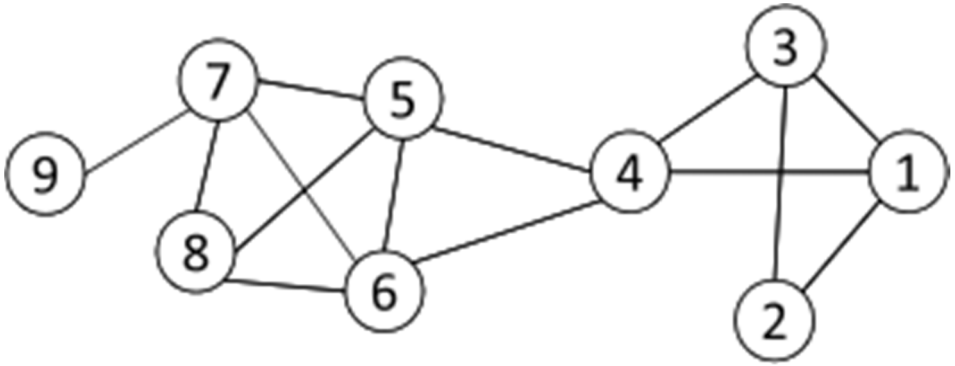


Edge Betweenness

- The times of shortest paths containing an edge



Divisive clustering based on edge betweenness



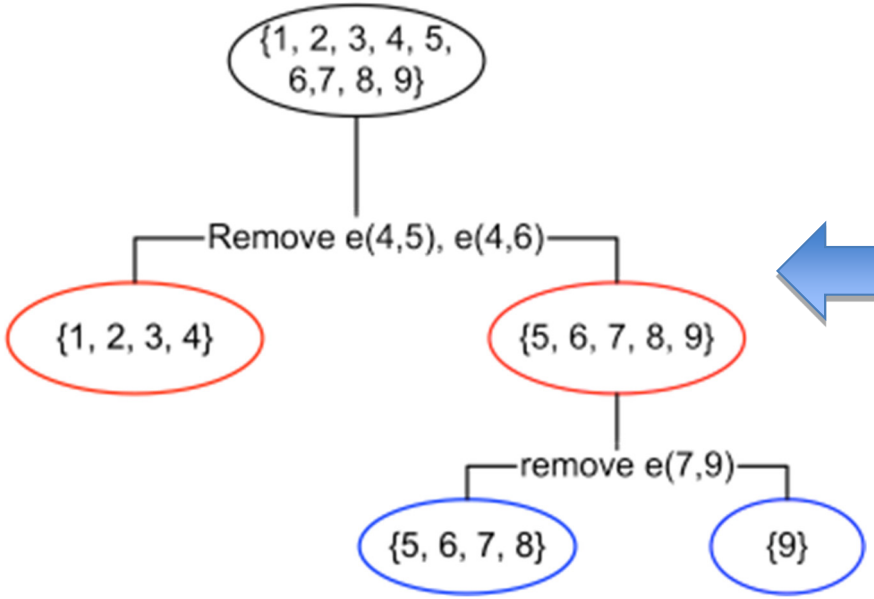
Initial betweenness value

Table 3.3: Edge Betweenness

	1	2	3	4	5	6	7	8	9
1	0	4	1	9	0	0	0	0	0
2	4	0	4	0	0	0	0	0	0
3	1	4	0	9	0	0	0	0	0
4	9	0	9	0	0	20	0	0	0
5	0	0	0	0	0	1	6	3	0
6	0	0	0	20	1	0	6	3	0
7	0	0	0	0	6	6	0	2	8
8	0	0	0	0	3	3	2	0	0
9	0	0	0	0	0	0	8	0	0

After remove e(4,5), the betweenness of e(4, 6) becomes 20, which is the highest;

After remove e(4,6), the edge e(7,9) has the highest betweenness value 4, and should be removed.

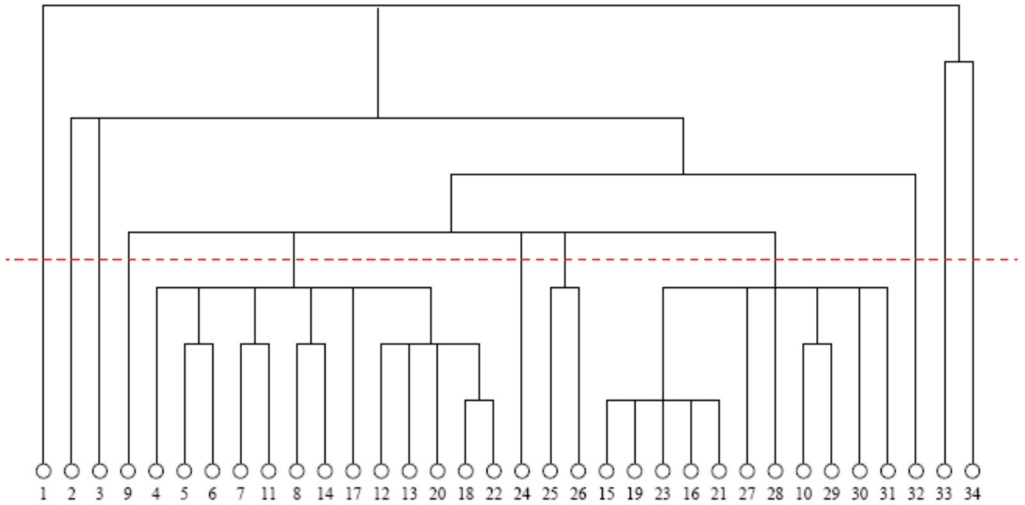


Idea: progressively removing edges with the highest betweenness

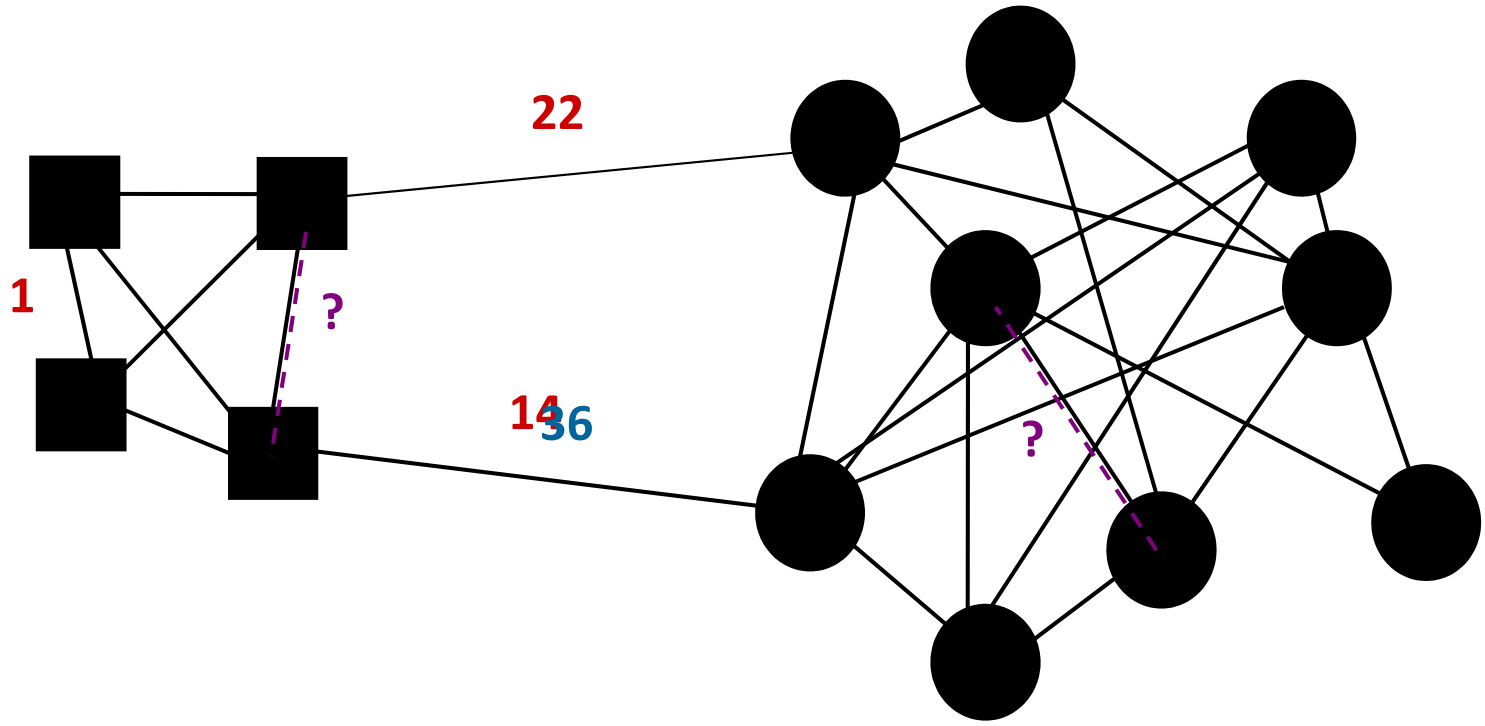
Girvan-Newman Algorithm

1. Calculate betweenness for all existing edges
2. The edges with the highest betweenness are removed
3. **Recalculate** betweenness for edges affected by removals
4. Repeat step 2. and step 3. until no edges remain
5. Cut down the **dendrogram**
   Based on Normalized Cut or **Modularity**

- By removing these edges, we separate groups from one another as components



Girvan-Newman Algorithm



If we don’t stop the process…

Girvan-Newman Algorithm

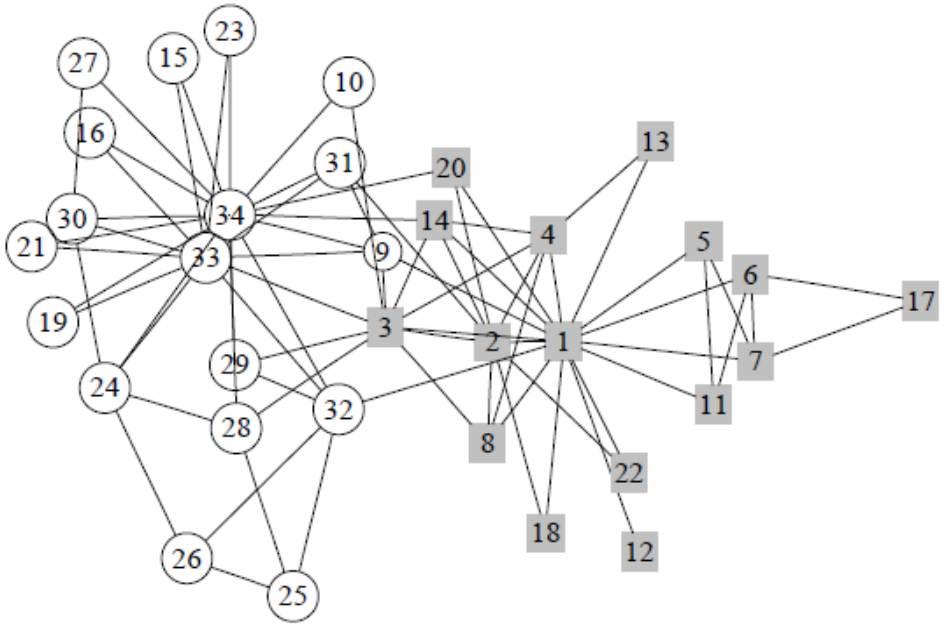
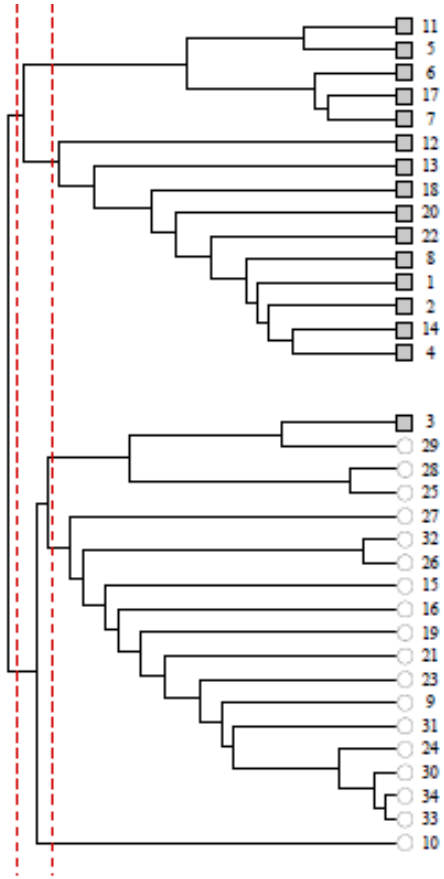
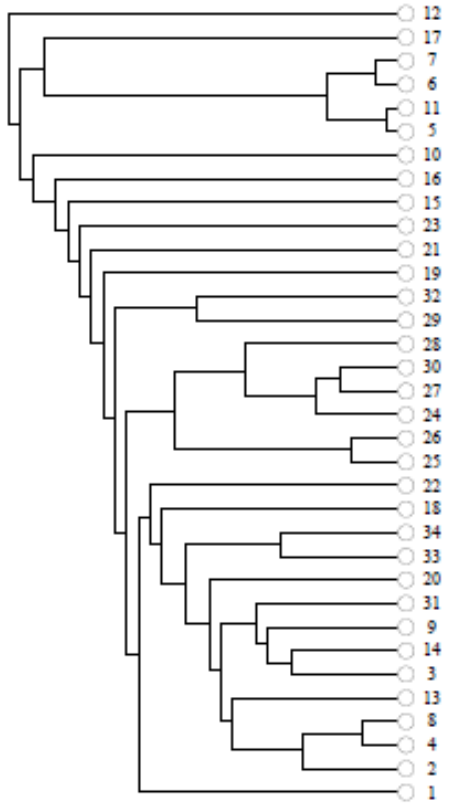


FIG. 8: The network of friendships between individuals in the karate club study of Zachary [35]. The administrator and the instructor are represented by nodes 1 and 33 respectively. Shaded squares represent individuals to who ended up aligning with the club's administrator after the fission of the club, open circles those who aligned with the instructor.



Removal based on edge betweenness

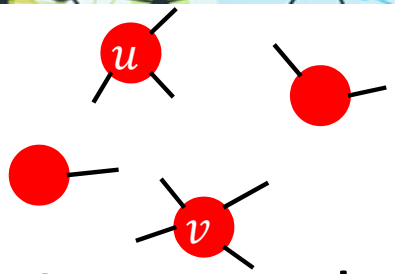


Removal based on edge betweenness **without re-calculation**

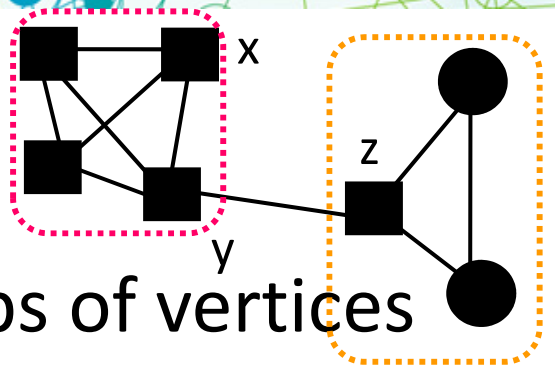
Two Disadvantages Currently

1. It provides no guide to how many communities a network should be split into
   - Need to estimate the quality of communities
   - Modularity was introduced

2. It is slow
   - *m edge removals* X *edge betweenness time*
   - Need improvement



Modularity



- Structural Definition of Community: Groups of vertices with many interior edges and few exterior edges

$$Q = \frac{1}{2m}\sum_{uv}\left(A_{uv} - \frac{k_u k_v}{2m}\right)\delta(x_u, x_v)$$

$\delta(x_u, x_v)$: Used to select only pairs belonging to the same community

sum over all pairs of nodes

$\frac{A_{uv}}{2m}$: the observed fraction of all edges between u and v

$\frac{k_u}{2m} \times \frac{k_v}{2m}$: the expected fraction of edges between u and v

$Q \in [-1, 1]$

- A: the adjacency matrix of the graph
- k_u: the degree of node u
- m: the total number of edges
- x_u: the community label of node u
- $\delta(x_u, x_v)$ = 1, if $x_u = x_v$; 0, otherwise

$$\frac{A_{xy}}{2m} = \frac{1}{2 \times 10} = \frac{20}{400}$$

x,y屬相同community, 但**有**彼此相鄰

$$\frac{k_x}{2m} \times \frac{k_y}{2m} = \frac{3}{2 \times 10} \times \frac{4}{2 \times 10} = \frac{12}{400}$$

$$\frac{A_{xz}}{2m} = \frac{0}{2 \times 10} = \frac{0}{400}$$

x,z屬相同community, 但**沒有**彼此相鄰

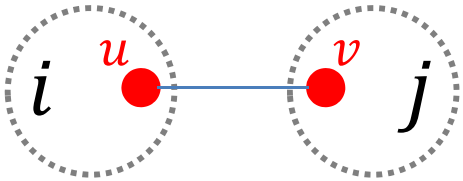
$$\frac{k_x}{2m} \times \frac{k_z}{2m} = \frac{3}{2 \times 10} \times \frac{3}{2 \times 10} = \frac{9}{400}$$

Modularity

$$Q = \frac{1}{2m}\sum_{uv}\left(A_{uv} - \frac{k_u k_v}{2m}\right)\delta(x_u, x_v)$$

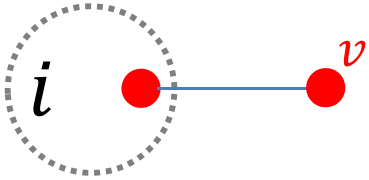
Summation is over pairs of nodes belonging to same communities
→ rewrite and simplify the equation by dropping the many zeros

Let $e_{ij} = \frac{1}{2m}\sum_{uv} A_{uv}\delta(x_u, i)\,\delta(x_v, j)$



be the fraction of edges that join nodes with community i
to nodes with community j

Let $a_i = \frac{1}{2m}\sum_{v} k_v\delta(x_v, i)$



be the fraction of ends of edges that are attached
to nodes with community i

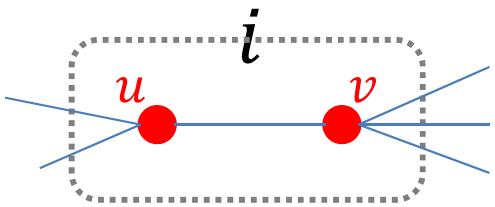
Modularity

$$Q = \frac{1}{2m}\sum_{uv}\left(A_{uv} - \frac{k_u k_v}{2m}\right)\delta(x_u, x_v)$$

$$Q = \frac{1}{2m}\left(\sum_{uv}\left(A_{uv} - \frac{k_u k_v}{2m}\right)\sum_{i}\delta(x_u, i)\delta(x_v, i)\right)$$

$$Q = \sum_{i}\left(\frac{1}{2m}\sum_{uv}A_{uv}\delta(x_u, i)\,\delta(x_v, i) - \frac{1}{2m}\sum_{u}k_u\delta(x_u, i)\frac{1}{2m}\sum_{v}k_v\delta(x_v, i)\right)$$

$$Q = \sum_{i} e_{ii} - a_i^2$$



$$e_{ij} = \frac{1}{2m}\sum_{uv}A_{uv}\delta(x_u, i)\,\delta(x_v, j) \qquad a_i = \frac{1}{2m}\sum_{v}k_v\delta(x_v, i)$$

Modularity

- Modularity
  - A numerical index of how good a particular division is

$$Q = \sum_i (e_{ii} - a_i^2)$$

(# of edges within groups) – (*expected* # of edges within groups)

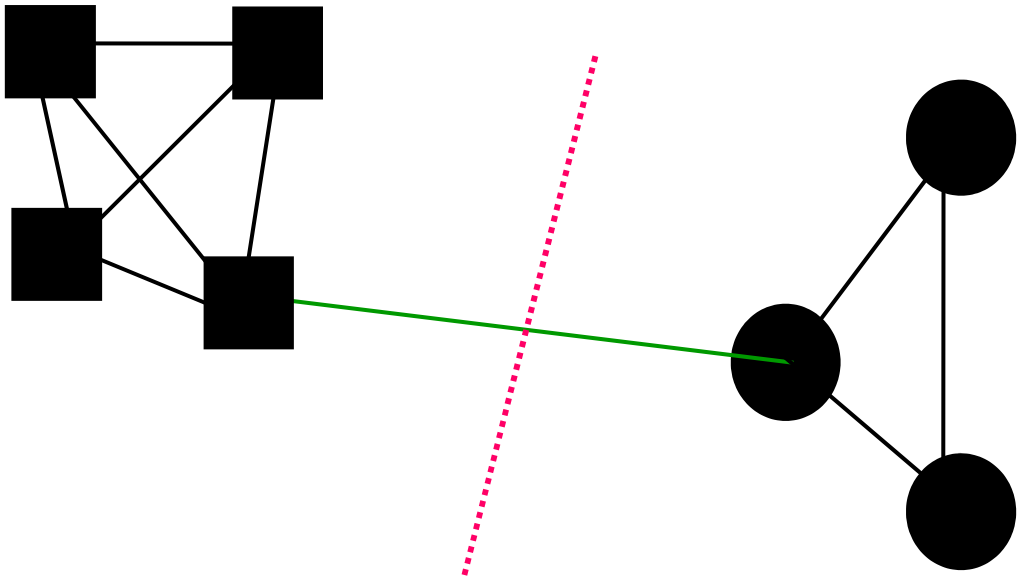
Fraction of edges within community i

Fraction of edges that involve in community i

- Consider a particular division of k communities
  - Define a $\boldsymbol{k \times k}$ symmetric matrix $\boldsymbol{E}$ (whose element is e_{ij})
  - e_{ij} = fraction of all edges in the network that link vertices in community i to vertices in community j
  - Let $a_i = \sum_j e_{ij}$ be the fraction of edges that connect to vertices in community i

Modularity (cont.)

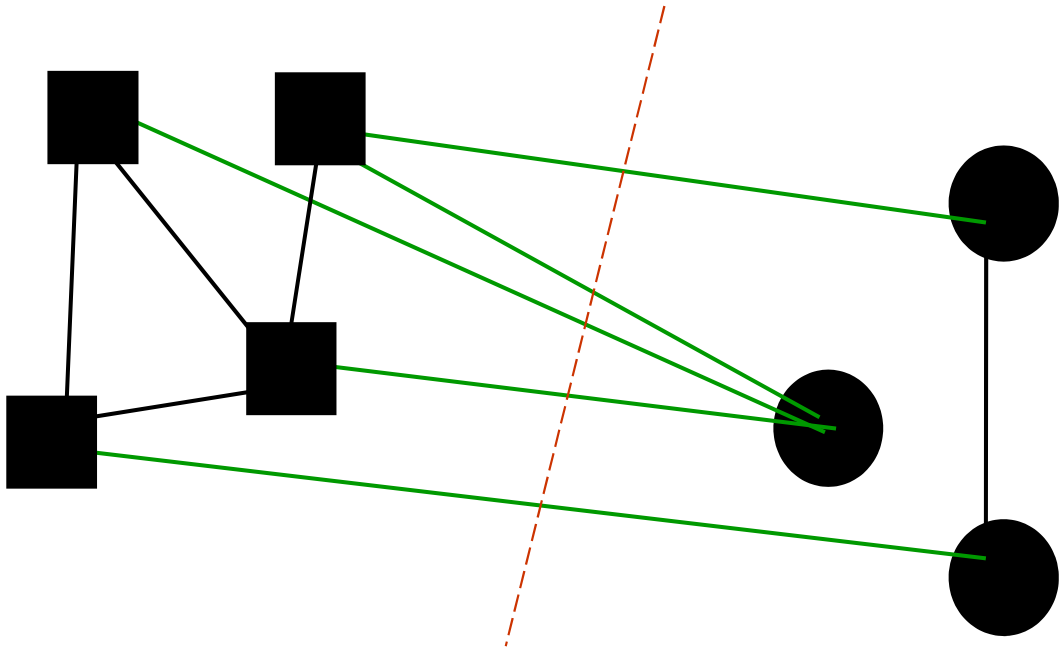
- Example, (good division)



$$e = \begin{bmatrix} \frac{6}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{3}{10} \end{bmatrix} \qquad Q = \left[\frac{6}{10} - \left(\frac{7}{10}\right)^2\right] + \left[\frac{3}{10} - \left(\frac{4}{10}\right)^2\right] = 0.25$$

Modularity (cont.)

- Example, (bad division)



$$e = \begin{bmatrix} \frac{4}{10} & \frac{5}{10} \\ \frac{5}{10} & \frac{1}{10} \end{bmatrix}$$

$$Q = \left[\frac{4}{10} - \left(\frac{9}{10}\right)^2\right] + \left[\frac{1}{10} - \left(\frac{6}{10}\right)^2\right] = -0.67$$

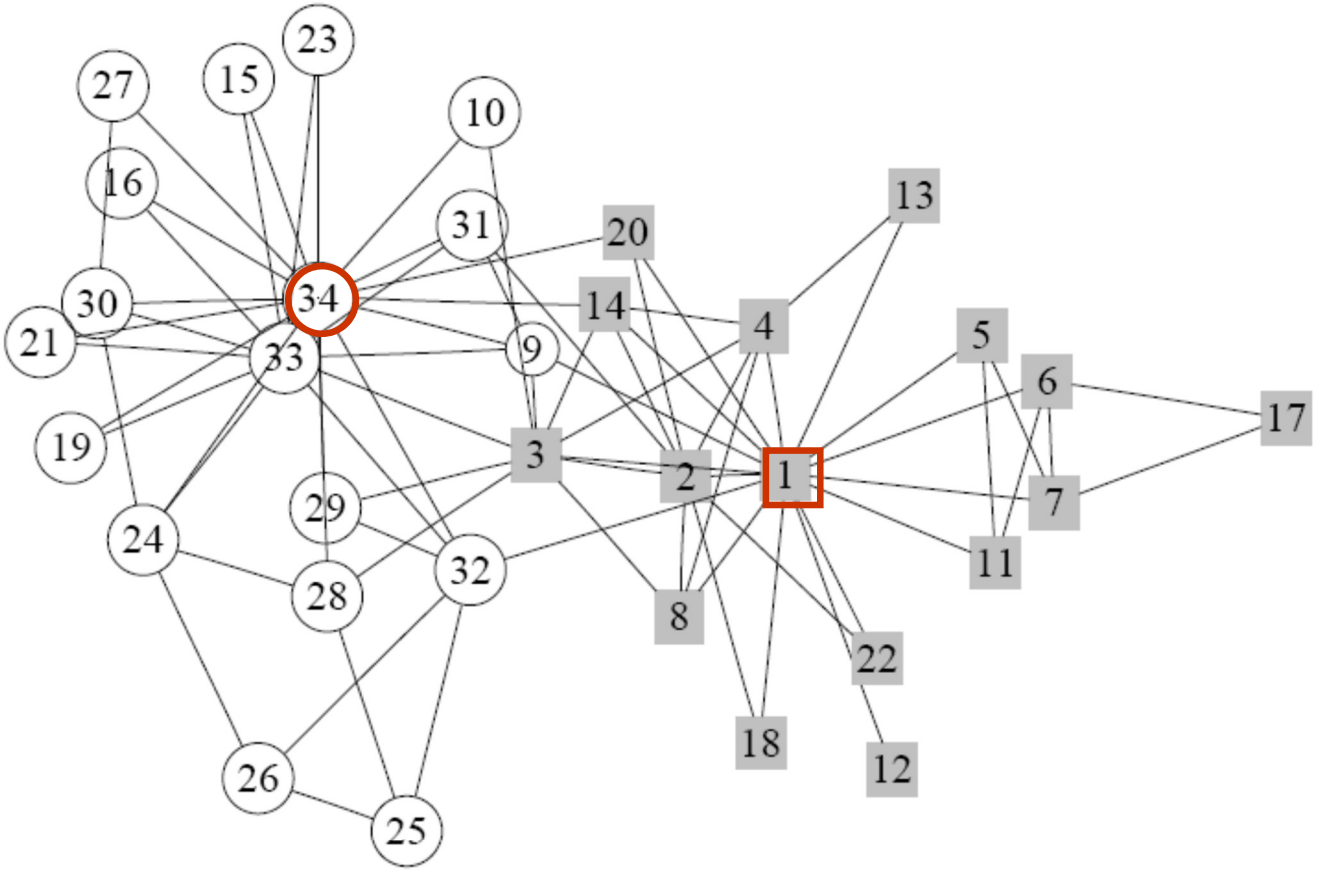
Modularity (cont.)

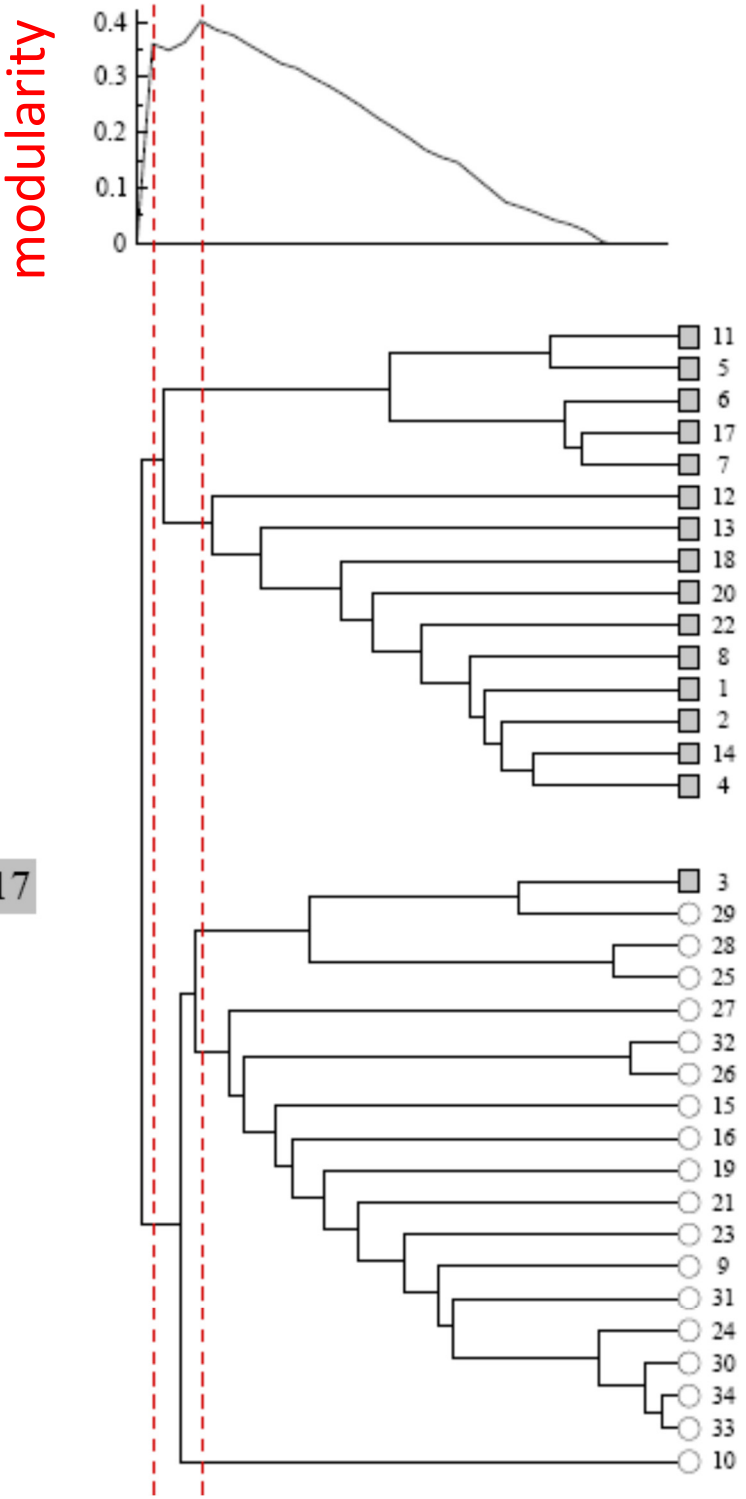
$$Q = \sum_i (e_{ii} - a_i^2)$$

- It keeps some properties ...
  - $\boldsymbol{Q = 0}$: no community structure
  - $\boldsymbol{Q \sim 1}$: perfect division
  - $\boldsymbol{0.3 < Q < 0.7}$: significant community structure (local peak)
  - $\boldsymbol{-1 < Q < 0}$: the minimum value could be

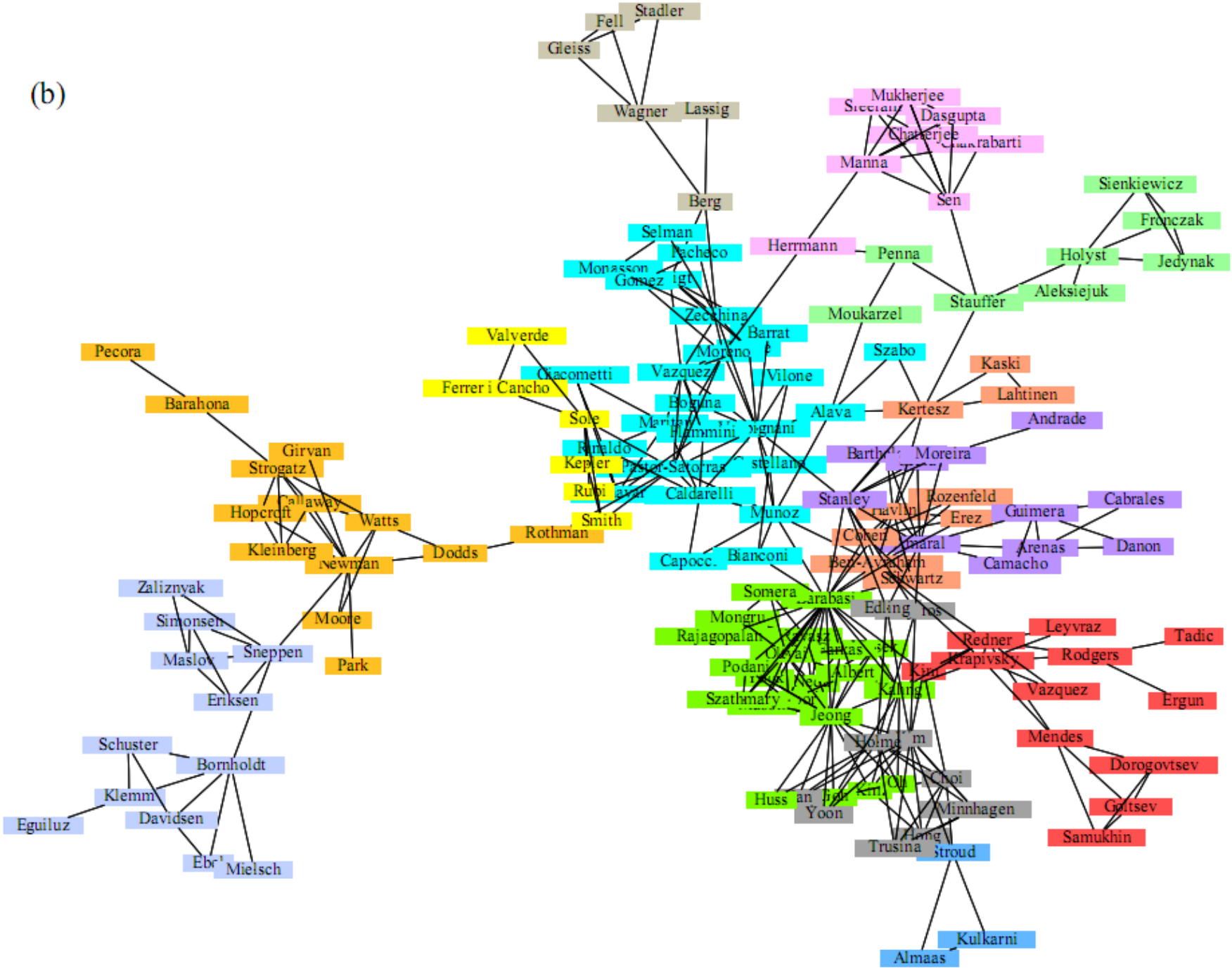
Modularity-based Partitioning

- Zachary's karate club network

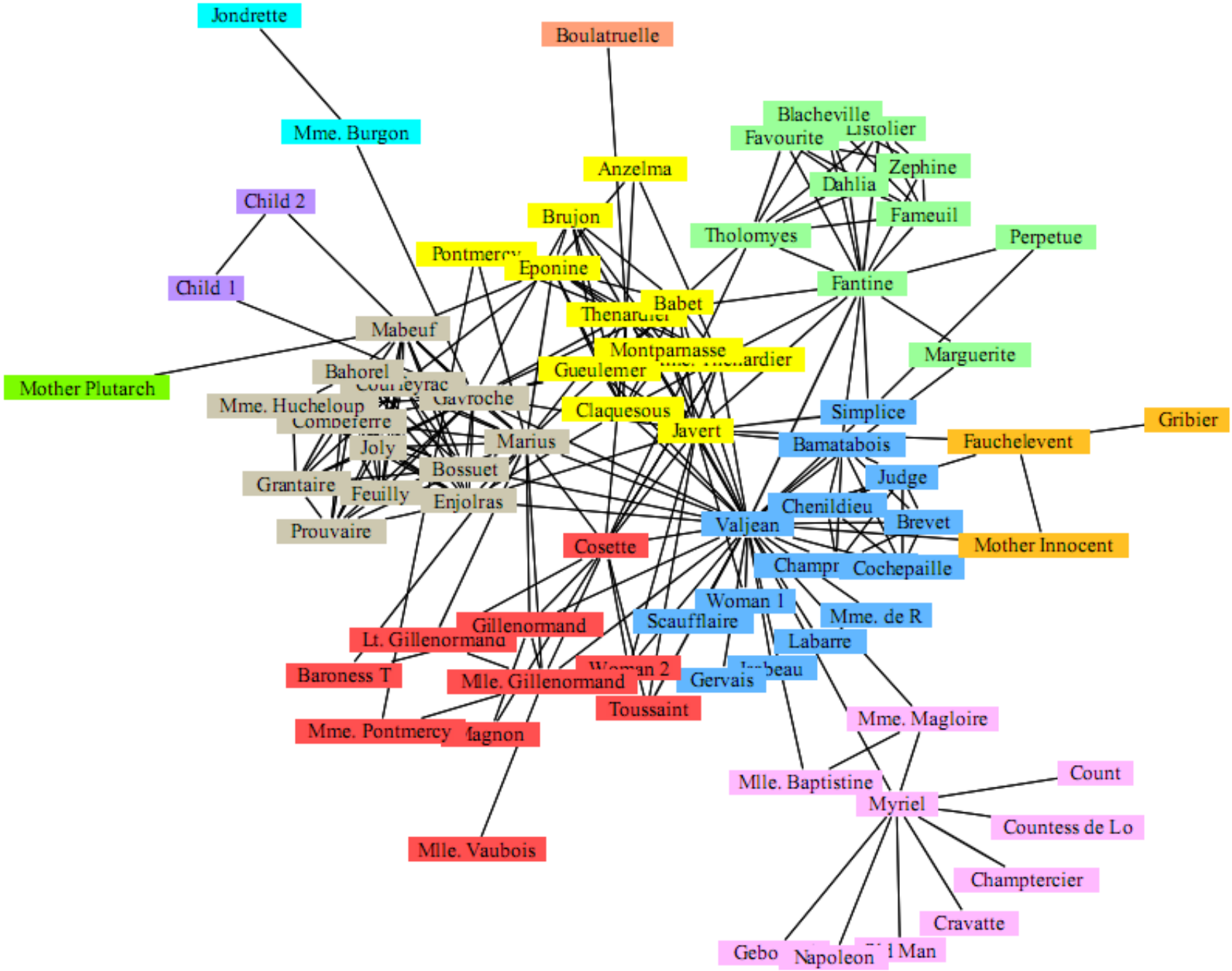




On A Real-world Co-authorship Data (Q=0.72)



On the Les Miserables Data (Q=0.54)



* M. E. J. Newman. “Fast Algorithm for Detecting Community Structure.” *Physics Review E*, 2004. (5316 cites)

Newman Fast Algorithm

- **Agglomerative** approach
  1. Separate each vertex solely into n communities
  2. **Calculate the difference of modularity score ΔQ for all connected community pairs**
  3. Join a community pair (u, v) according to
     - Largest increase or, smallest decrease of modularity
  4. Repeat step 2. and 3. until all communities merged
  5. Cross cut the dendrogram where Q is maximum