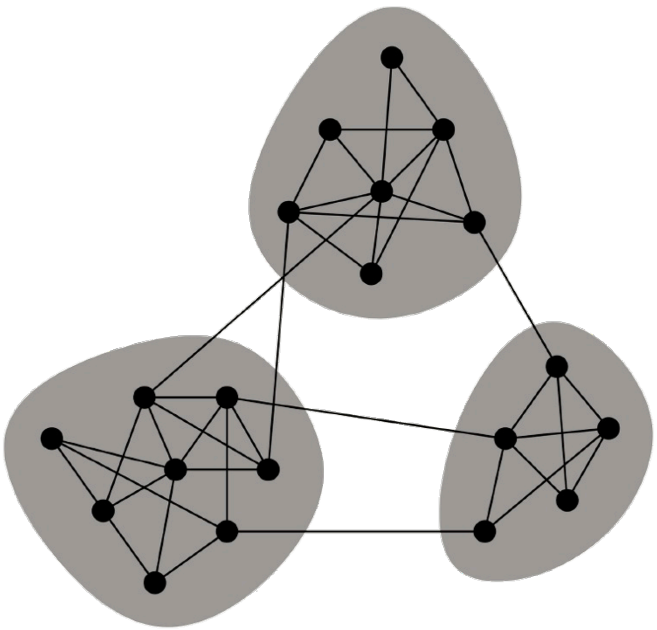
GRAPHS

SESSION 2

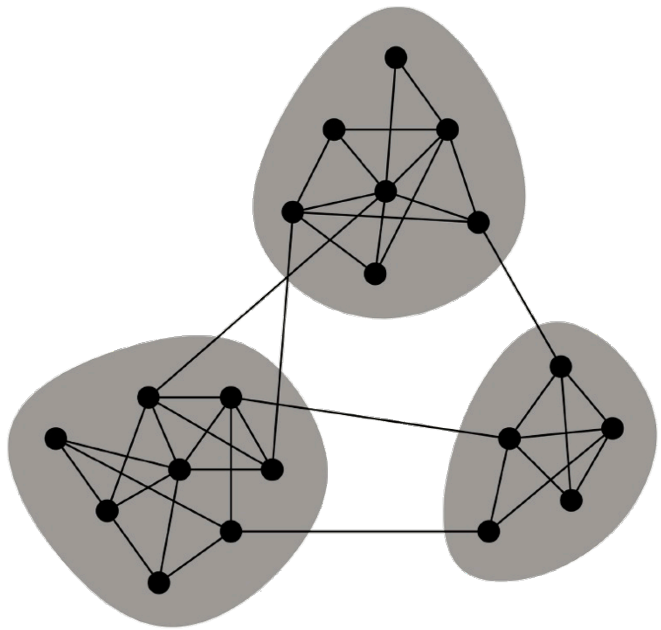
Defining communities in graphs
Finding communities

Communities



- Formally defining a measure of “community-ness” in the graph
- Finding communities

Communities, general idea



# of edges inside a community are greater than the number of edges to outside communities

Modularity of a division

Number of edges within group

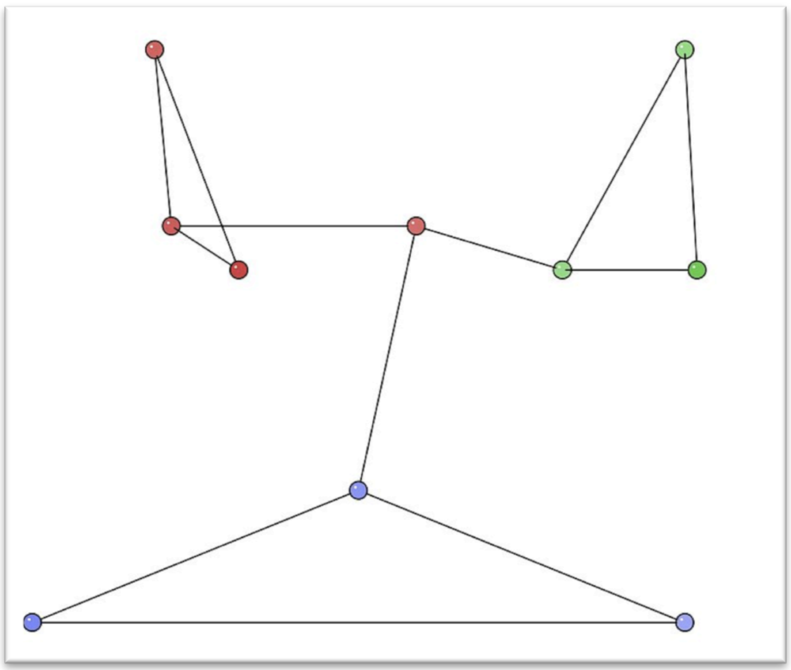
VERSUS

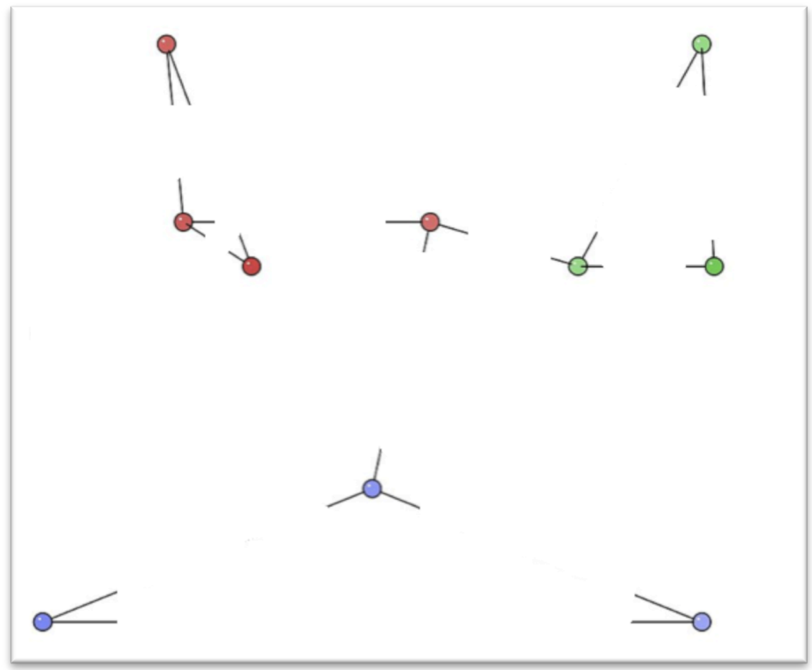
Expected value of edges within group in a RANDOM graph with same node degrees

If we were analyzing a tightly knit community the edge density seen within the community will be higher than expected at random.

Let's explore this idea with an example.

Randomized graph





Cut edges in half to end up with stubs that look like this
Then randomly wire up stubs to each other. Loops are allowed!

Randomized graph to expected values

$$\text{P(single edge stub gets connected to } j) = \frac{d(j)}{2m}$$

E(# of full edges from i to j) $= d(i) \cdot \dfrac{d(j)}{2m} = \dfrac{d(i)d(j)}{2m}$

where

$$d(i) = \text{degree of node } i$$

$$m = \text{number of edges in the graph}$$

Expected value of edges within given community

E(edges within given community) $= \displaystyle\sum_{i,j \text{ in same community}} \frac{d(i)d(j)}{2m}$

Modularity

$$\text{modularity}(G,\mathcal{C}) = \frac{1}{2m} \overbrace{\sum_{C\in\mathcal{C}} \sum_{i,j\in C} \overbrace{\overset{\text{Actual Edges}}{A_{ij}} - \overset{\text{Expected edges if random}}{\frac{d(i)d(j)}{2m}}}^{\text{In a given community}}}^{\text{Over all communities}}$$

where

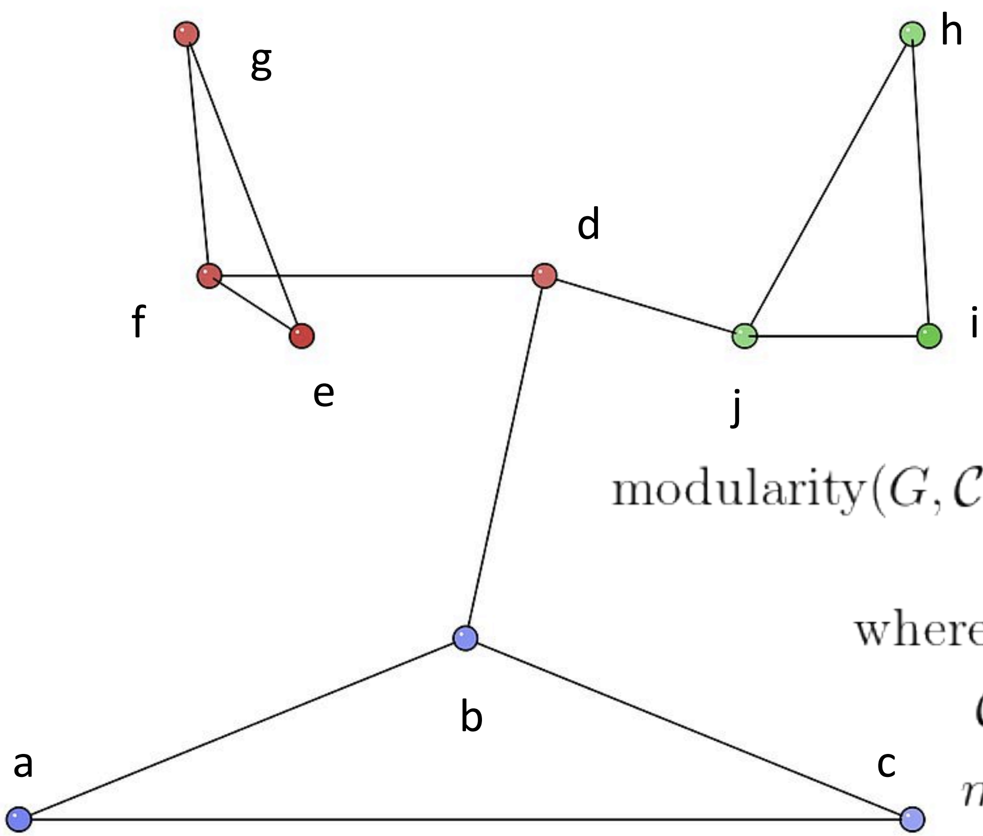
$$\mathcal{C} = \text{the collection of communities}$$

$$m = \text{number of edges in the graph}$$

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is an edge} \\ 0 & \text{if } (i,j) \text{ is not an edge} \end{cases}$$

$$d(i) = \text{degree of node } i$$

Modularity calculation toy example (Q = 0.296875)



$$\text{modularity}(G, \mathcal{C}) = \frac{1}{2m} \sum_{C \in \mathcal{C}} \sum_{i,j \in C} A_{ij} - \frac{d(i)d(j)}{2m}$$

where

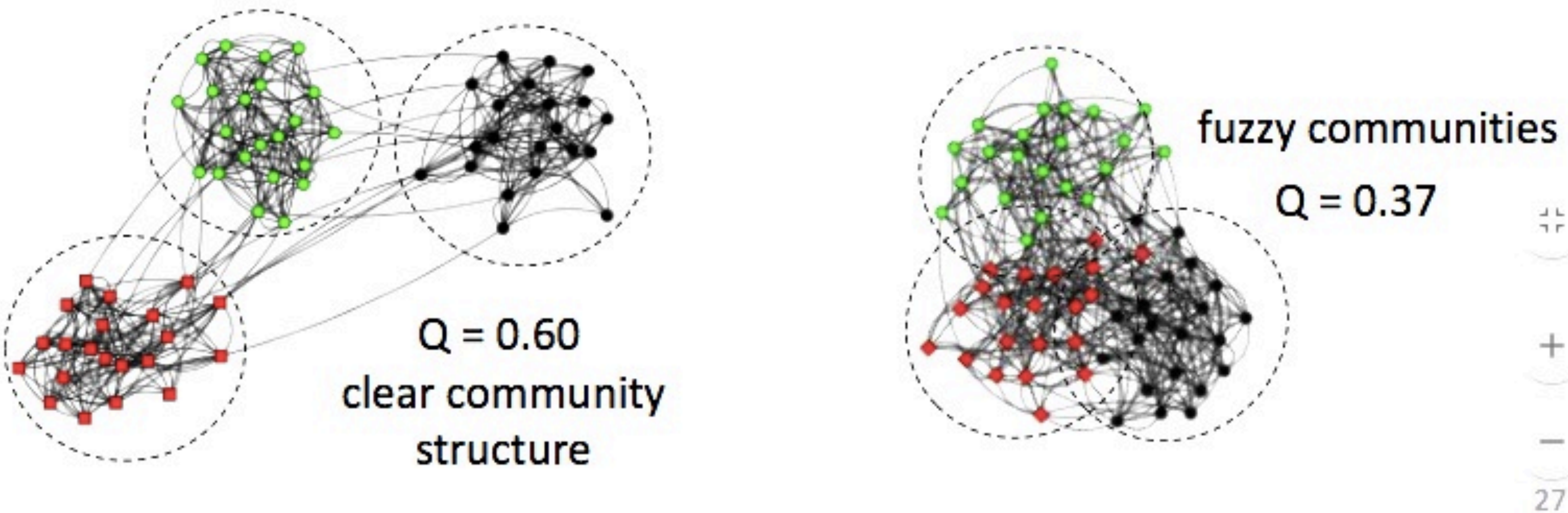
$$\mathcal{C} = \text{the collection of communities}$$

$$m = \text{number of edges in the graph}$$

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \text{ is an edge} \\ 0 & \text{if } (i, j) \text{ is not an edge} \end{cases}$$

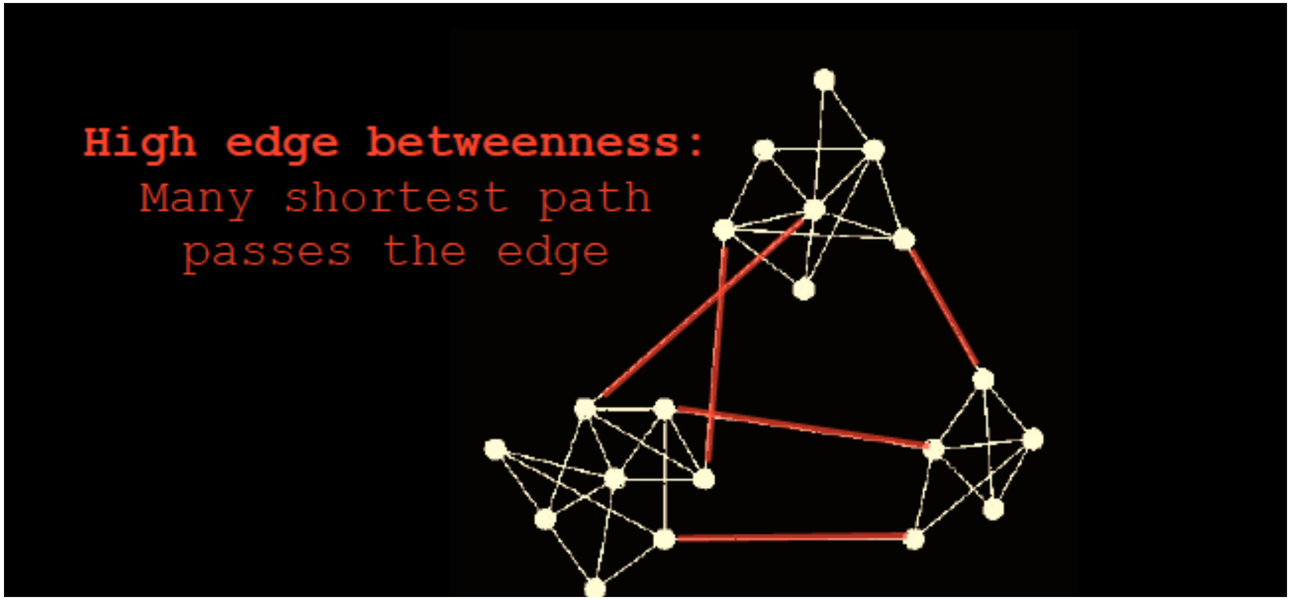
$$d(i) = \text{degree of node } i$$

Exploring Q



Typically in non-random graphs modularity takes values between 0.3 and 0.7
Greater than 0.7 is considered a high level of modularity.

Girvan-Newman



We iteratively remove the edge with the highest *edge betweenness*. The *edge betweenness* is a measure of how many paths an *edge* is part of.

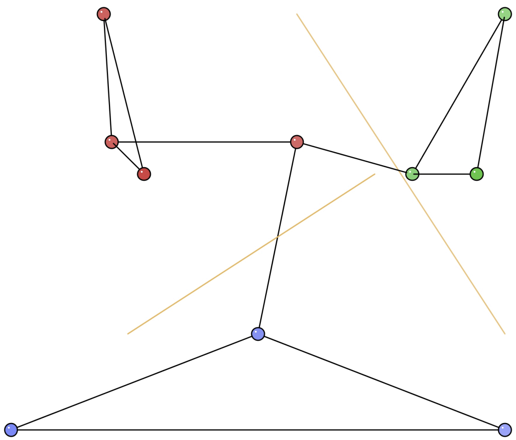
Edge Betweenness

$$
\begin{aligned}
\text{betweenness}(e) &= \sum_{s \neq v \neq t} \text{percent of shortest paths from } s \text{ to } t \text{ which pass through } e \\
&= \sum_{s \neq v \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}
\end{aligned}
$$

where

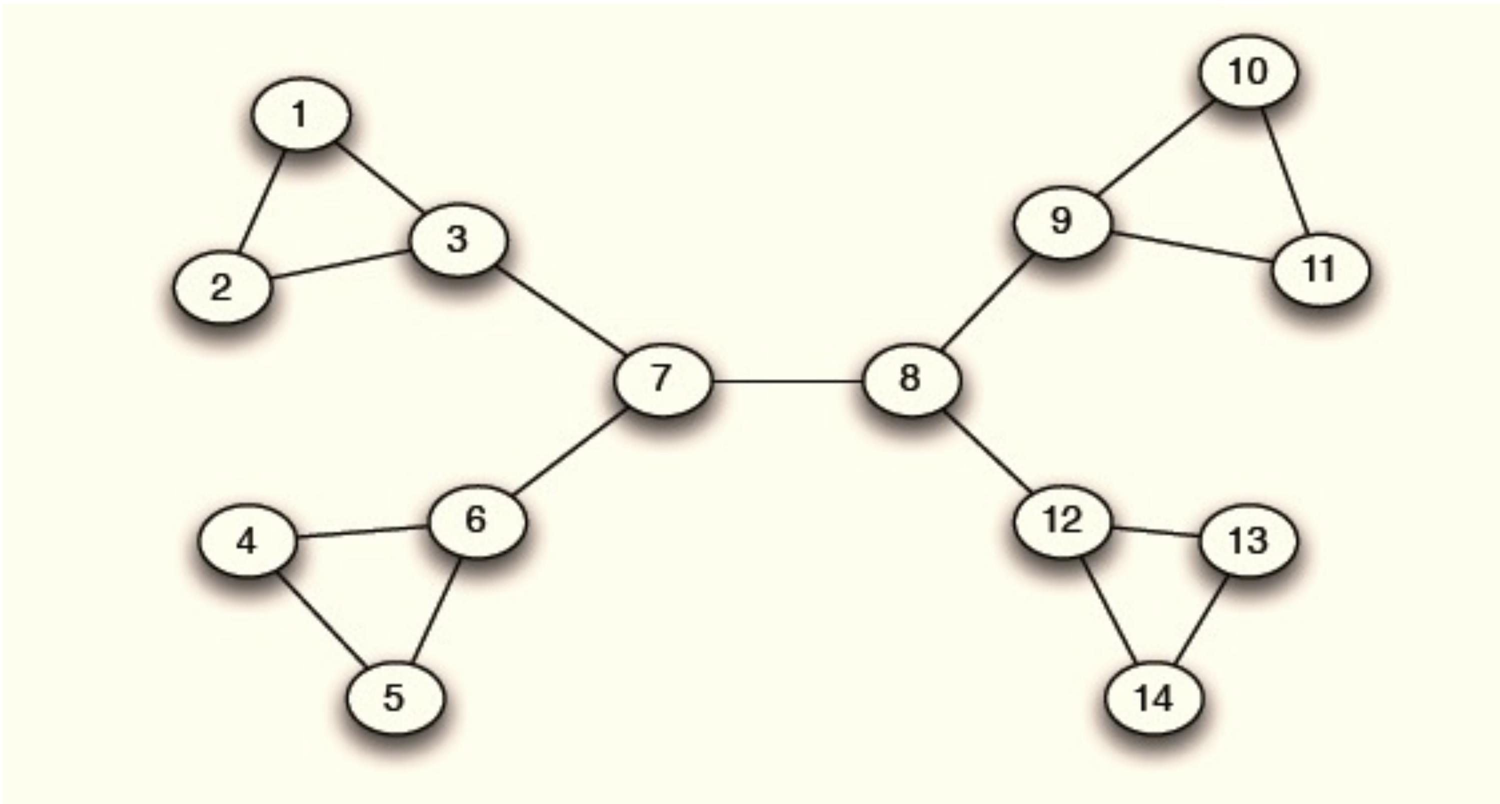
$$
\begin{aligned}
\sigma_{st}(e) &= \#\text{ of shortest paths from } s \text{ to } t \text{ which pass through } e \\
\sigma_{st} &= \#\text{ of shortest paths from } s \text{ to } t
\end{aligned}
$$

Girvan-Newman Pseudocode



```
function GirvanNewman:
    repeat:
        repeat until a new connected component is created:
            calculate the edge betweenness centralities for all the edges
            remove the edge with the highest betweenness
```

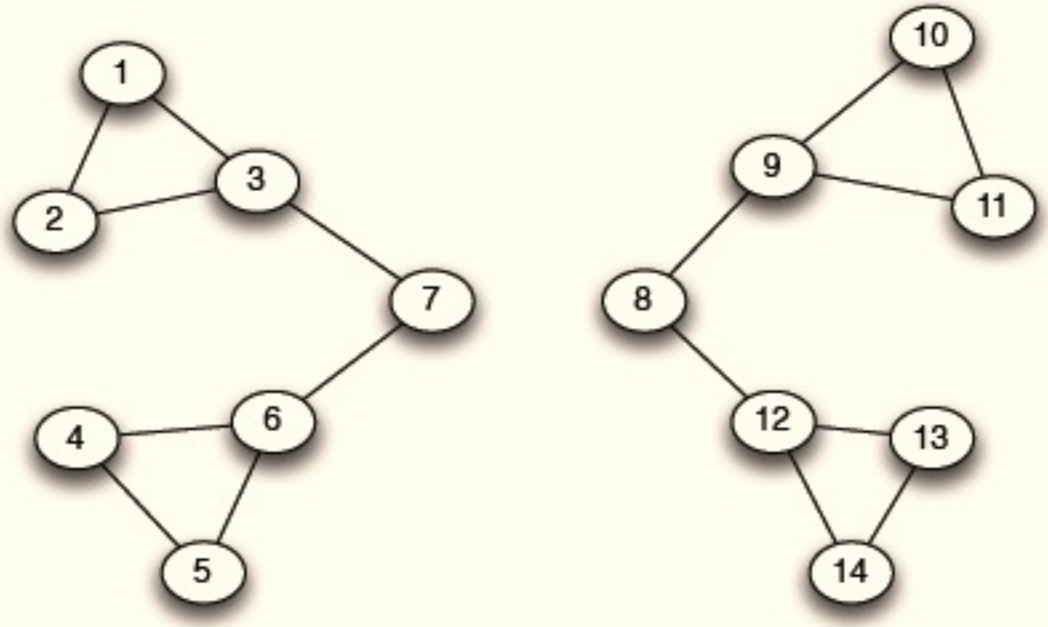
Girvan Newman method: An example



Betweenness(7-8)= 7x7 = 49 Betweenness(1-3) = 1X12=12

Betweenness(3-7)=betweenness(6-7)=betweenness(8-9) = betweenness(8-12)= 3X11=33

Girvan Newman method: An example

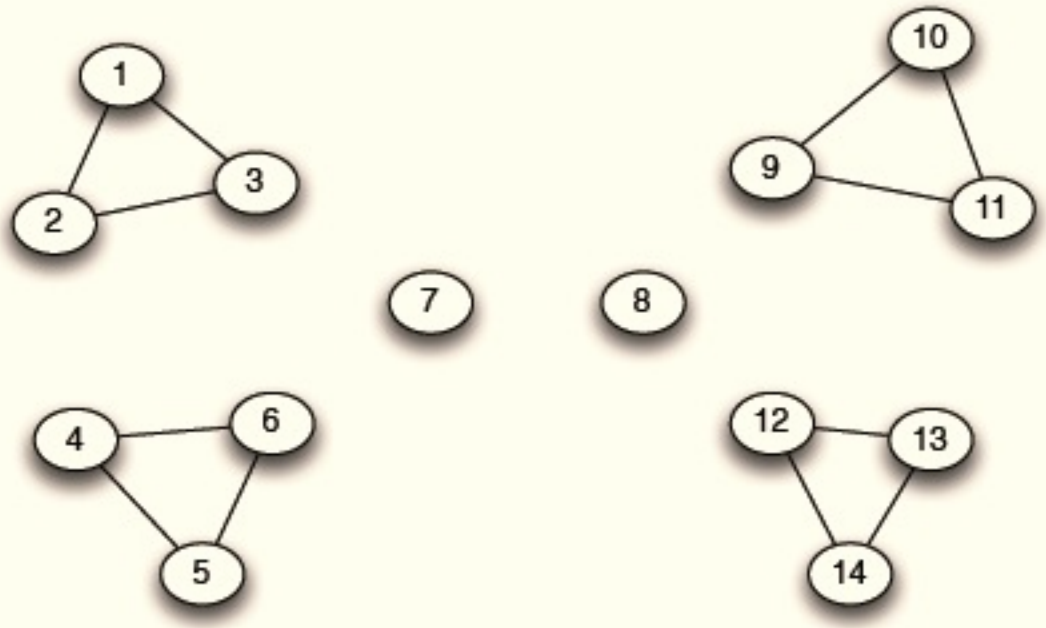


(a) *Step 1*

Betweenness(1-3) = 1X5=5

Betweenness(3-7)=betweenness(6-7)=betweenness(8-9) = betweenness(8-12)= 3X4=12

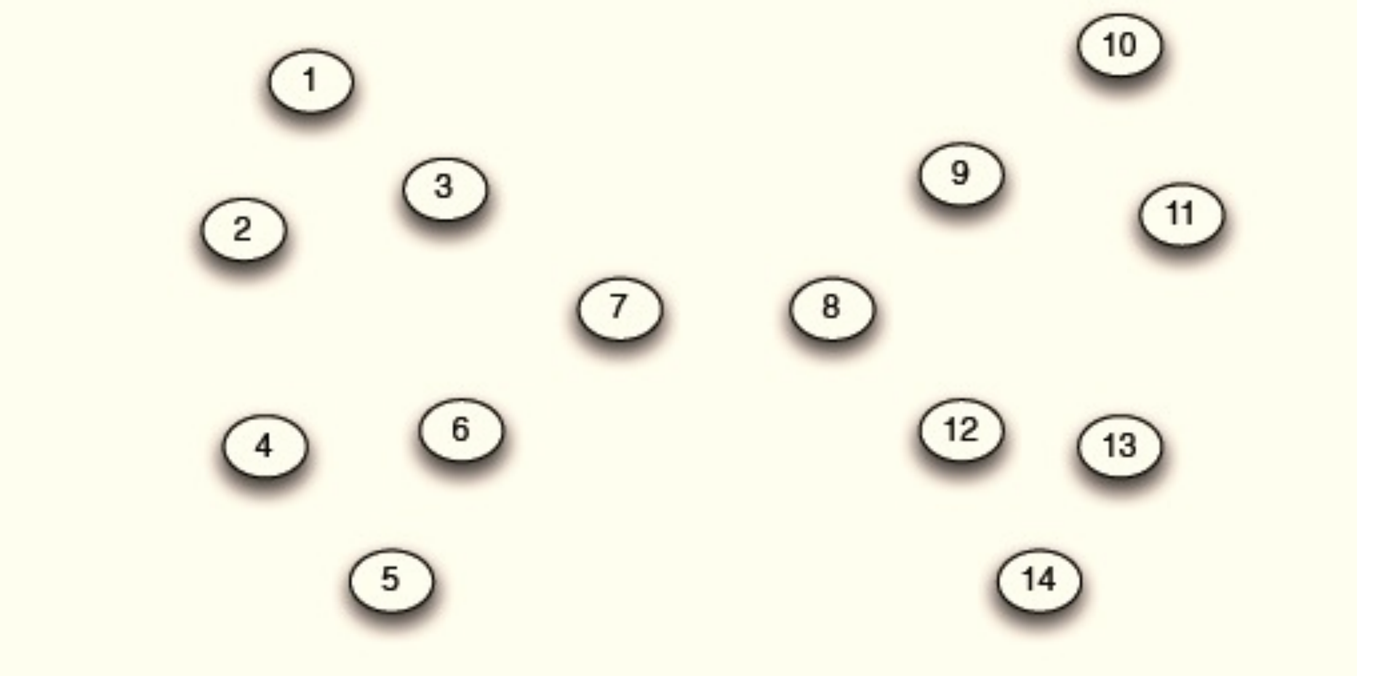
Girvan Newman method: An example



(b) *Step 2*

Betweenness of every edge = 1

Girvan Newman method: An example



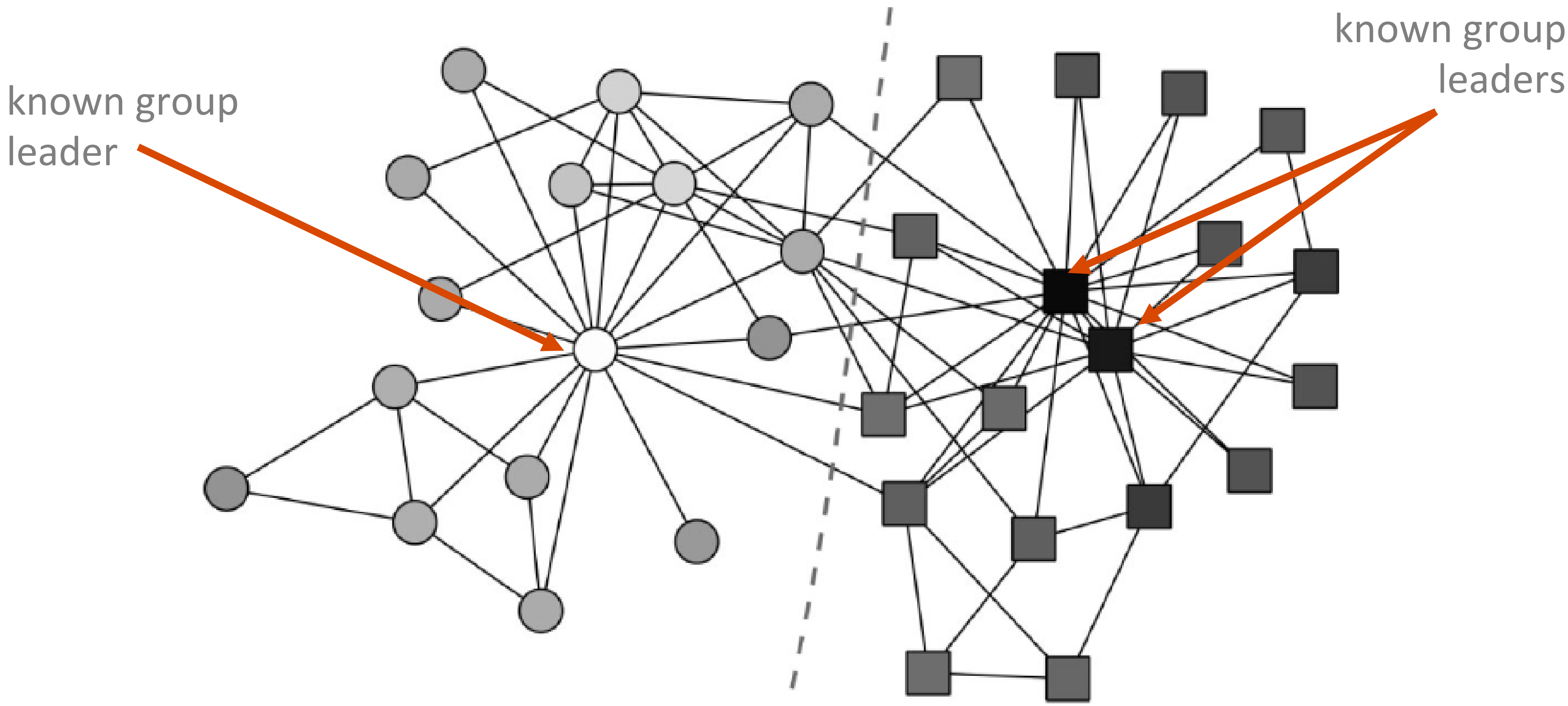
Girvan-Newman

- This will iteratively create new communities.
- But how do we determine the appropriate number of communities?

  We calculate the modularity for each set of communities and pick the one with the maximum modularity.

A real world example next!

Example: a 2-division of a social network

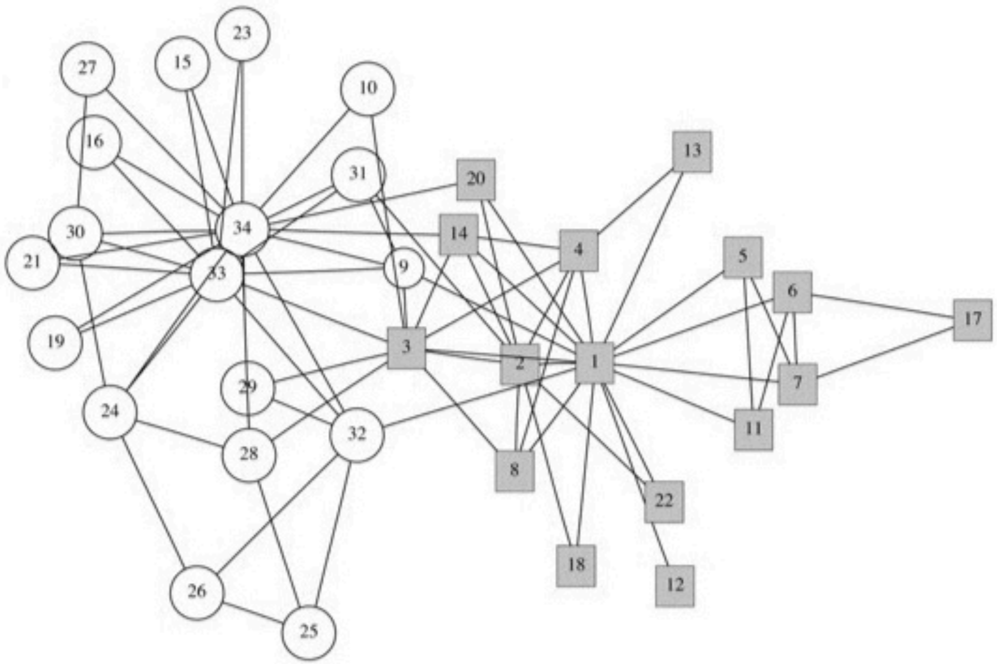


Zach's Karate Club

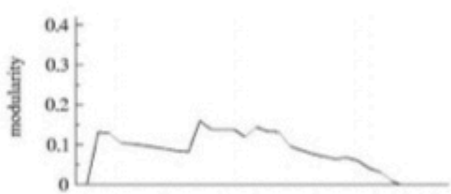
A network showing relationships between people in a karate club which eventually split into 2.

The division algorithm predicts exactly the two groups after the split

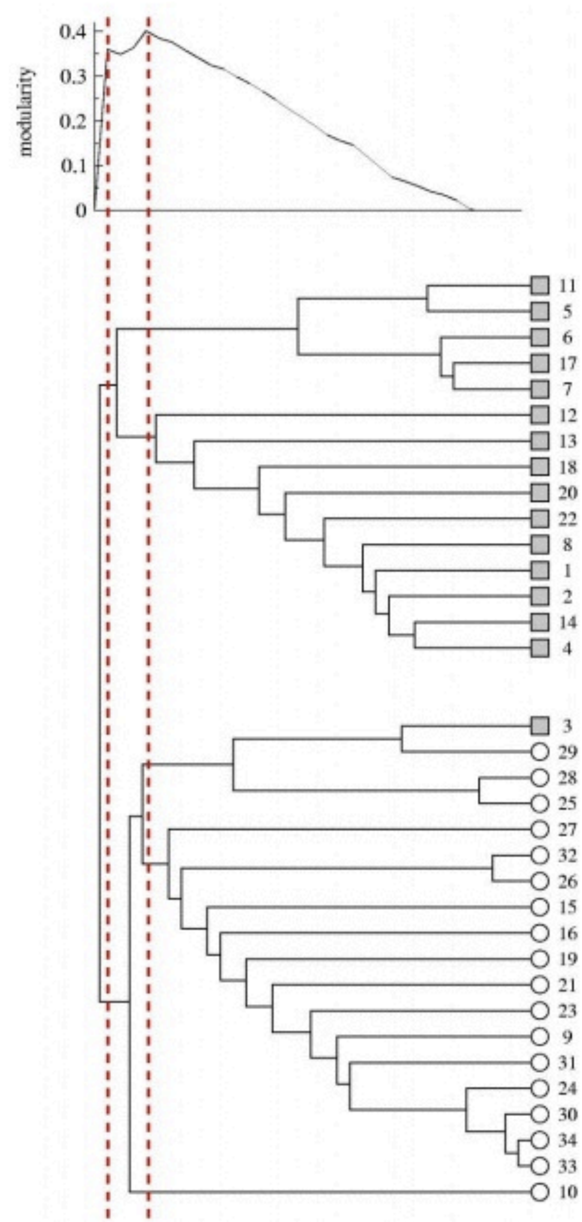
Girvan-Newman example



Optimal community structure for Zachary's karate club.



Modularity without recalculation



Session Summary and SPRINT

- Modularity – gives a measure of “community”ness of a graph and it’s communities
- Community detection using Girvan-Newman algorithm
  - Uses edge betweenness

SPRINT

- Finding communities in the IMBD database!