Lecture 20. Community Detection

COMP90051 Statistical Machine Learning

Semester 2, 2015
Lecturer: Andrey Kan

Content is based on slides provided by Jeffrey Chan

THE UNIVERSITY OF MELBOURNE

Copyright University of Melbourne

Communities in graphs

- *Community is a group of vertices* that interact more frequently within its own group than to those outside the group
  - Families
  - Friend circles
  - Websites (communities of webpages)
  - Groups of proteins that maintain a specific function in a cell

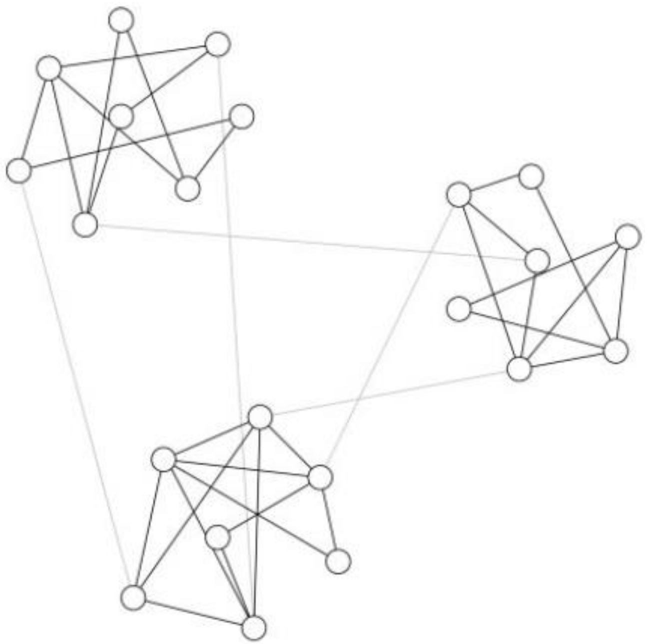


Image: Girvan and Newman, Community structure in social and biological networks, PNAS, 2002

Community detection

- Communities are natural phenomena - e.g., human societies, biological processes
- Other names for communities: modules, dense subgraph, cohesive subgraph
- Finding communities in graphs is called *community finding or detection*
- Generally unsupervised problem

Why search for communities?

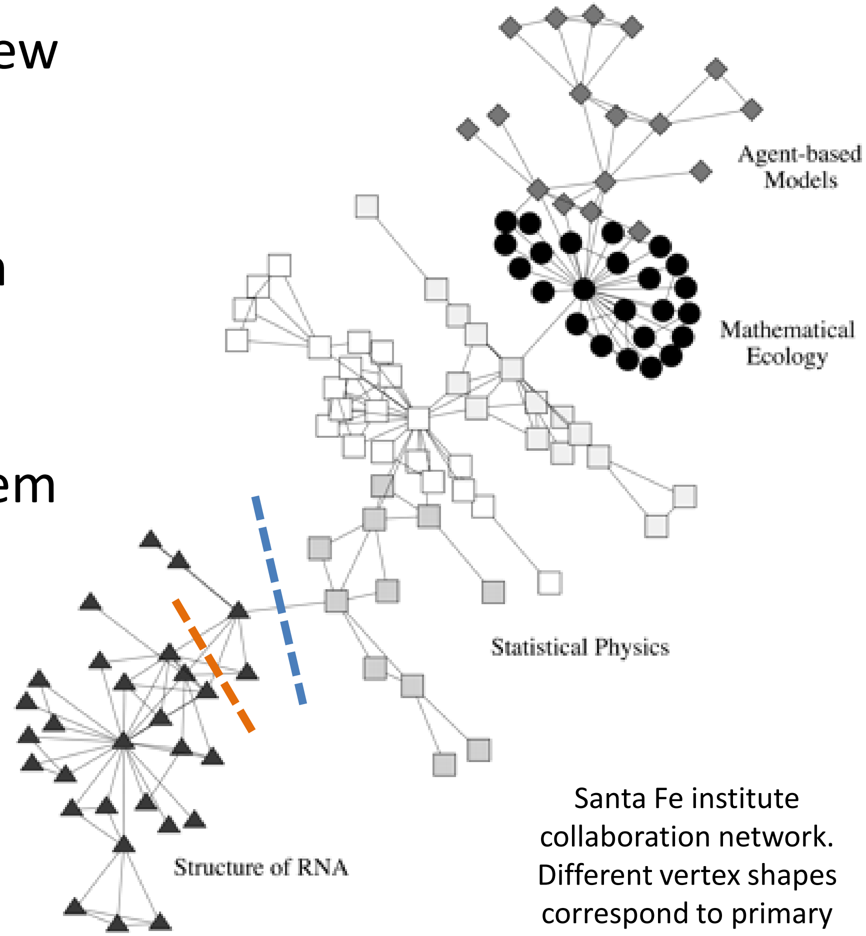
- Understanding the system behind the network
  - Social structure
  - Biological structure
- Identifying roles of vertices (e.g., hubs, mediators)
- Reduction of large graphs
  - Summary graph: vertices – communities, edges – connections between communities
- Facilitate distributed computing
  - Place data from the same community to the same server or core

Community definition revisited

- *Community is a group of vertices* that interact more frequently within its own group than to those outside the group
  - But what is community exactly (mathematically)?
- No commonly accepted rigorous definition
- Many different approaches, but we focus on a few interesting ones
  - Edge betweenness
  - Modularity score
  - Clique percolation

Edge betweenness idea

- Communities are connected by a few connections, which tends to form *bridges*
- Consider all shortest path between each pair of vertices
- Bridges are edges that have many shortest paths running through them
- Related idea: minimum cuts



Santa Fe institute collaboration network. Different vertex shapes correspond to primary divisions of the institute

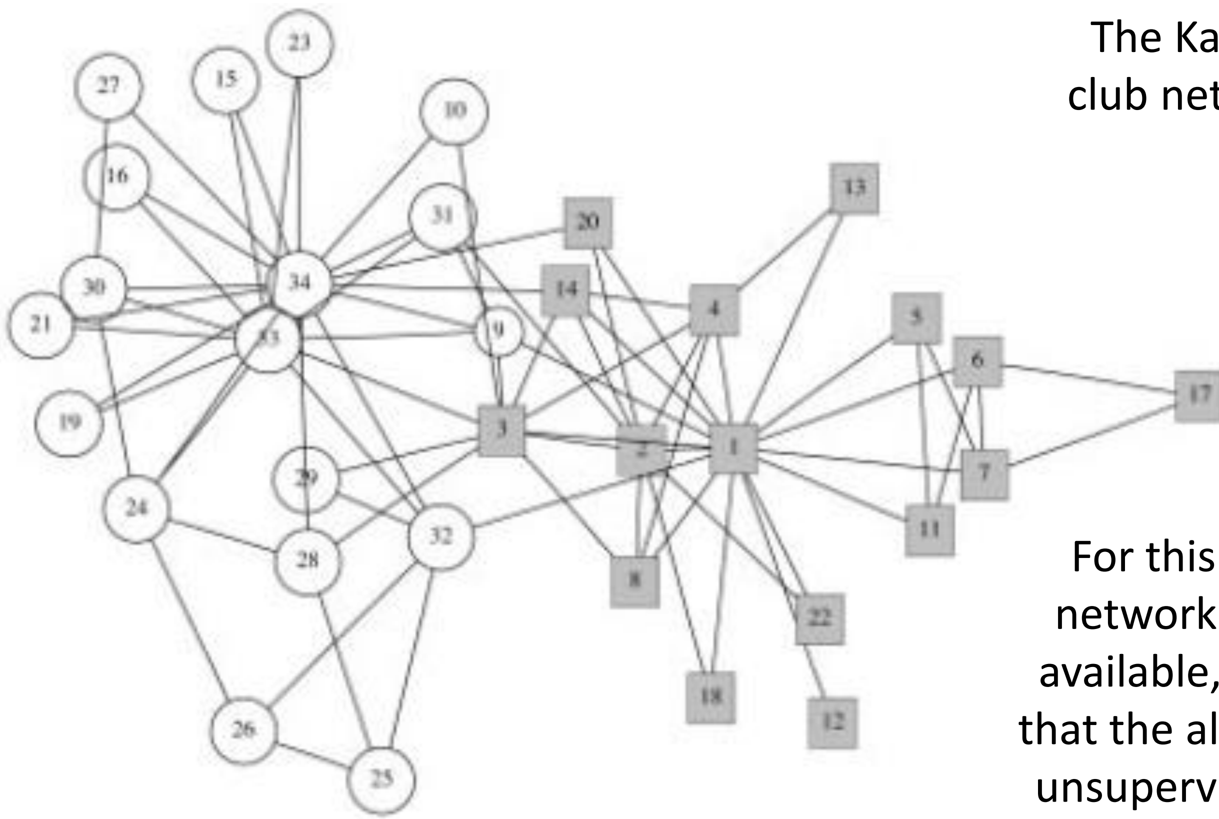
Image: Girvan and Newman, Community structure in social and biological networks, PNAS, 2002

Edge betweenness community detection

- Look for bridges, remove them. Eventually remove all edges between the natural communities.
  - Essentially divisive hierarchical clustering

1. Start with a single community with all vertices
2. While there are edges in graph
   a) (Re-)calculate betweenness for all edges
   b) Remove edge with the highest betweenness
   c) When current graph falls apart in two connected components record these as communities

Edge betweenness algorithm in action

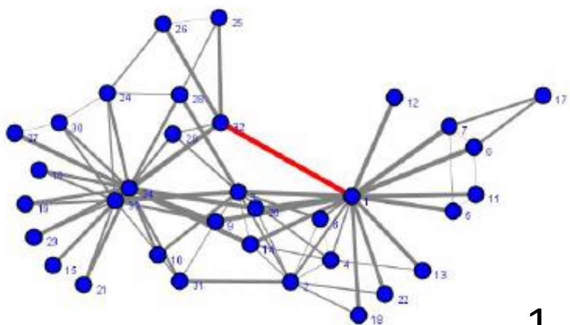


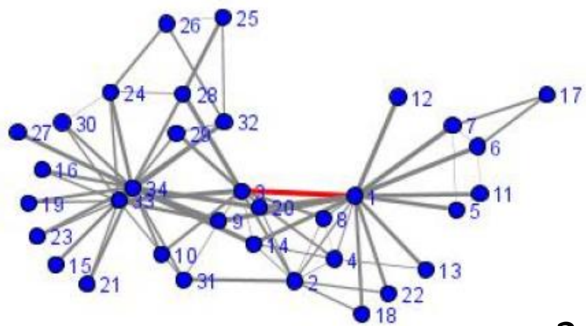
The Karate club network

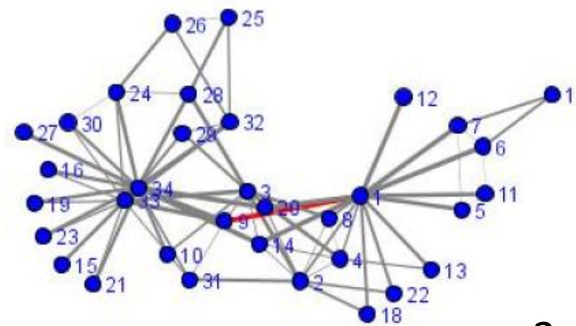
For this particular network labels are available, but recall that the algorithm is unsupervised (does not use the labels)

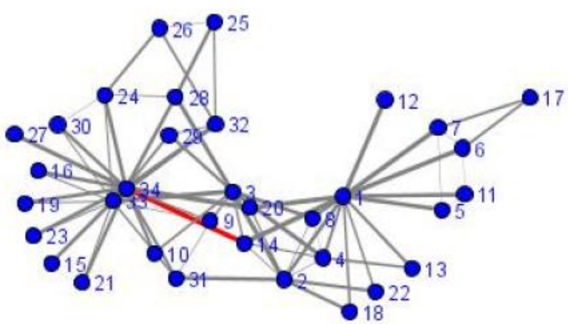
Image: Girvan and Newman, Community structure in social and biological networks, PNAS, 2002

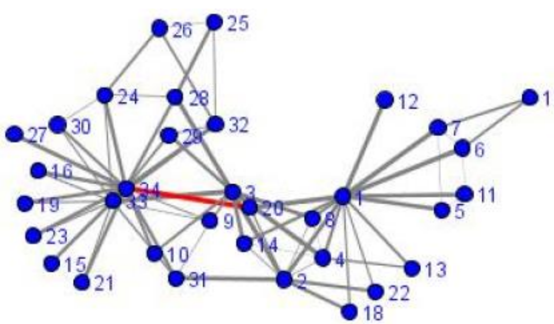
Edge betweenness algorithm in action

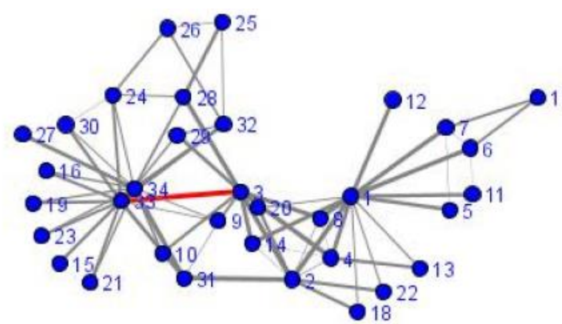

1


2

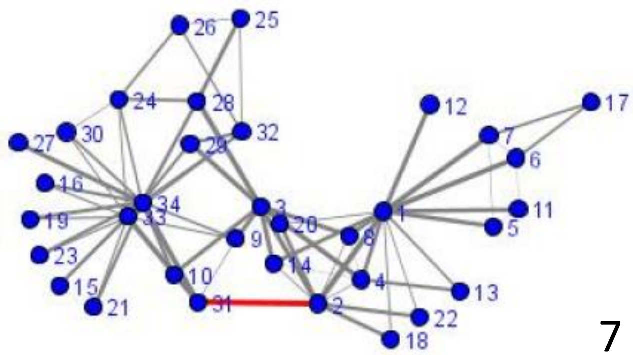

3


4

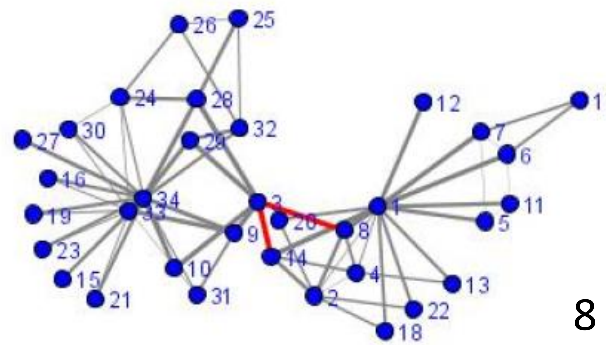

5


6

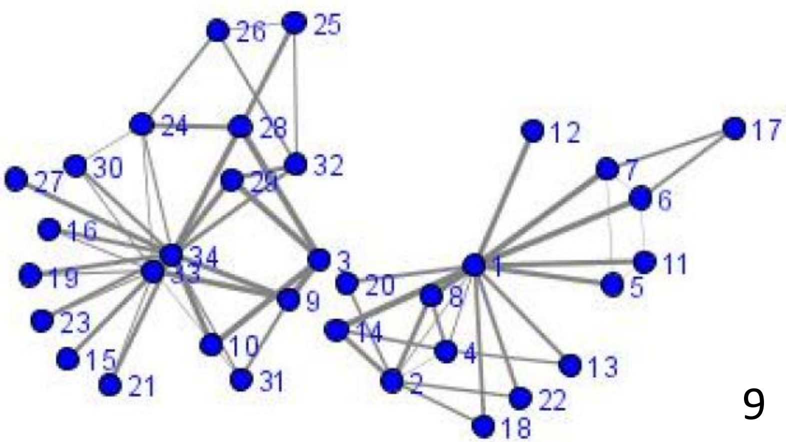
Edge betweenness algorithm in action



7

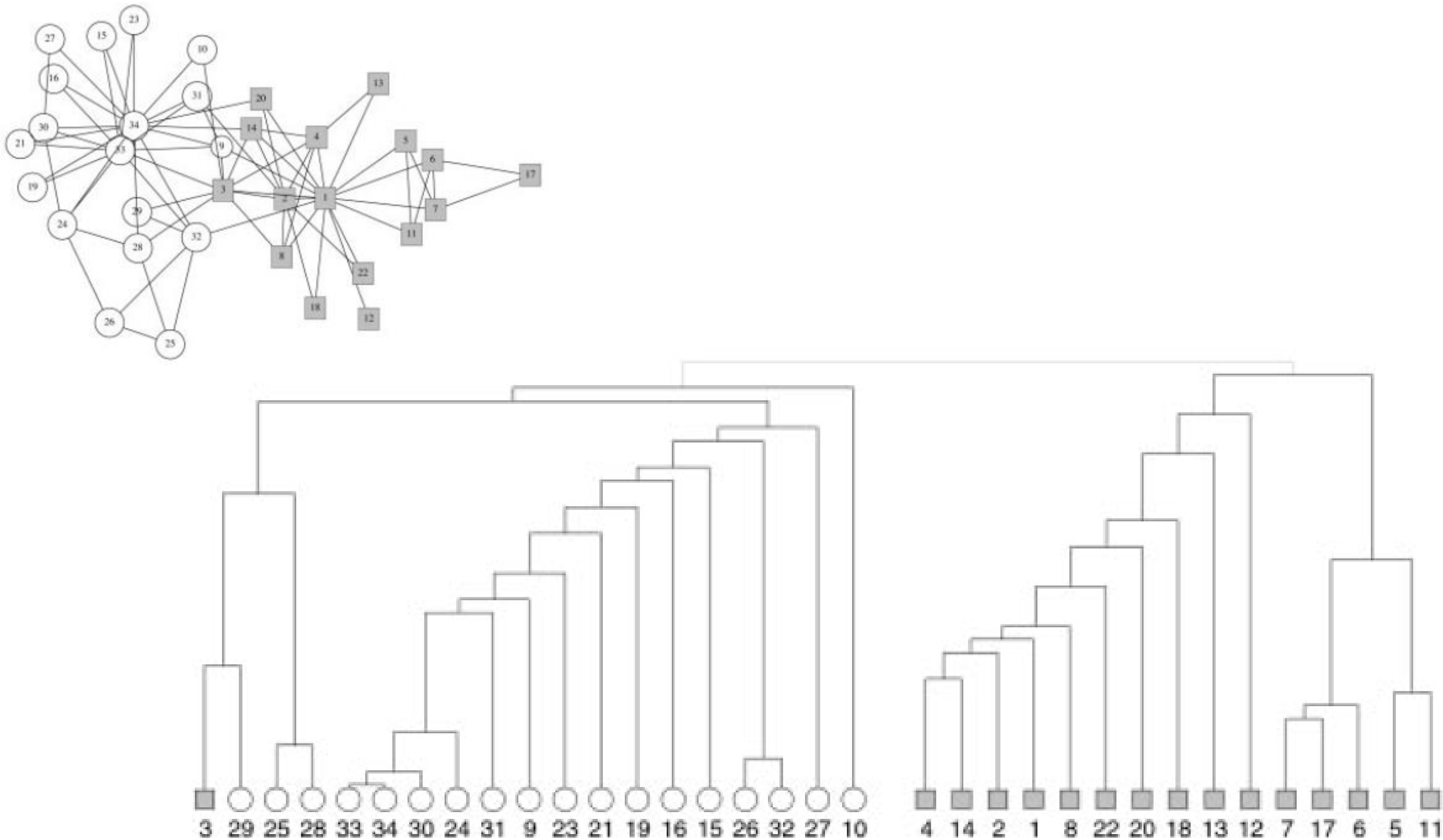


8



9

Edge betweenness algorithm result



Images: Girvan and Newman, Community structure in social and biological networks, PNAS, 2002

Modularity based community detection

- *Modularity* is a measure that indicates how unexpected a set of communities are
  - The more unexpected, the more likely those communities are inherent ones
- Note that any random arrangement of graph will result in some form of communities
- Modularity measures the extent of deviation from randomness

Modularity score

- Consider an undirected unweighted graph $G = \{V, E\}$ without self-edges and multi-edges
- Denote adjacency matrix of this graph as A_{ij}, and the number of edges $m \stackrel{\text{def}}{=} |E|$
  - Then the number of non-zero elements in A_{ij} is $2m$
- Next consider a candidate partitioning into communities C
- Let $\delta(i, j|C)$ be 1 if vertices i and j belong to the same community, and 0 otherwise

Modularity score

- The modularity of a partitioning is defined as

$$Q(C) = \frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i, j | C)$$

  * Here, k_i, k_j are degrees for vertices i and j, respectively

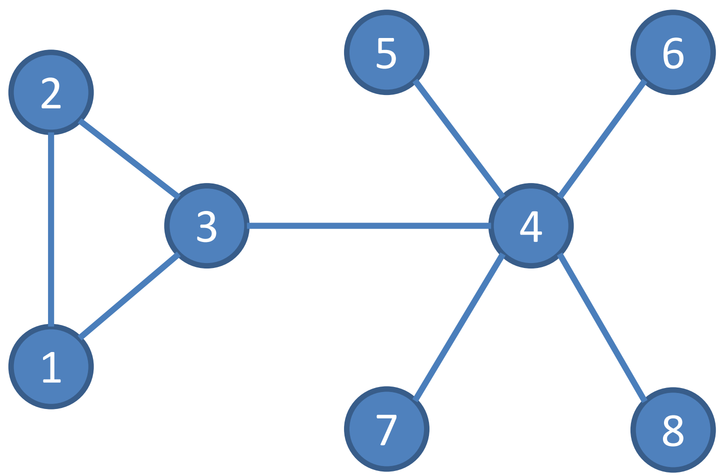
- Quantity $\frac{k_i k_j}{2m}$ is the expected number of edges in a random graph, given vertex degrees and number of edges

- Hence $Q(C)$ measures the deviation from the random graph (the difference between observed and expected)
  * Note that for a random graph $Q(C) = 0$, as desired

Side note: Chung-Lu random graph

- Expected vertex degrees are given $k_1, k_2, \dots, k_n$, where $n = |V|$ is the number of vertices
- The vertices are connected at random
- The probability of an edge between vertices i and j is taken $\frac{k_i k_j}{2m}$, where $m = |E|$ is the number of edges

Modularity score exercise



For $(3{,}4)$ it is
$$\left(1-\frac{3\cdot 5}{2\cdot 8}\right)=\frac{1}{16}$$

For $(2{,}3)$ it is
$$\left(1-\frac{2\cdot 3}{2\cdot 8}\right)=\frac{5}{8}$$

For $(4{,}5)$ it is
$$\left(1-\frac{5\cdot 1}{2\cdot 8}\right)=\frac{3}{8}$$

- Compute $\left(A_{ij}-\frac{k_i k_j}{2m}\right)$ for vertex pairs $(3{,}4)$, $(2{,}3)$, and $(4{,}5)$

Agglomerative hierarchical clustering

1. Initialisation: each vertex forms a separate community
2. Update:
   a) Choose *a pair of communities*, such that their merging maximizes ΔQ
   b) Merge the two clusters

   (the number of clusters decreases by one in each step)
3. Termination: stop when all vertices fall into a single community
4. Go to Step 2

Limitations of modularity based method

- Resolution limit: the random graph model assumes that each vertex “sees” any other vertex. This assumption is often violated in real-world networks
- Inter-community edges can have large positive modularity values, and communities can incorrectly merged

Alternative view of the modulariy score

- By definition

$$Q(C) = \frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i,j|C)$$

- This can be rewritten as

$$Q(C) = \sum_{c=1}^{n_c} \left(\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right)$$

  * Here n_c is the number of communities, l_c is the number of edges between vertices in community c, and d_c is the sum of the degrees in vertices from community c
  * This equation makes use of the fact $\left(\sum_s^{n_c} k_s\right)^2 = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} k_i k_j$, where n_c is the number of vertices in community c

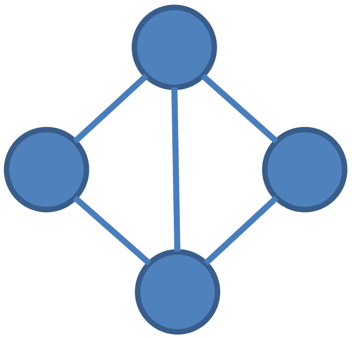
- Here $\frac{l_c}{m}$ is the *observed* fraction of edges a community, and $\left(\frac{d_c}{2m}\right)^2$ is the *expected* fraction of edges for a community with given vertex degrees

Clique-centric community detection

- Intuitively a community is a strongly connected subgraph
- Recall that a clique is a completely connected subgraph
- However, requiring each community to be a clique can be too stringent a requirement
- Also identifying clique-like subgraphs in itself doesn’t consider the way how these subgraphs are connected

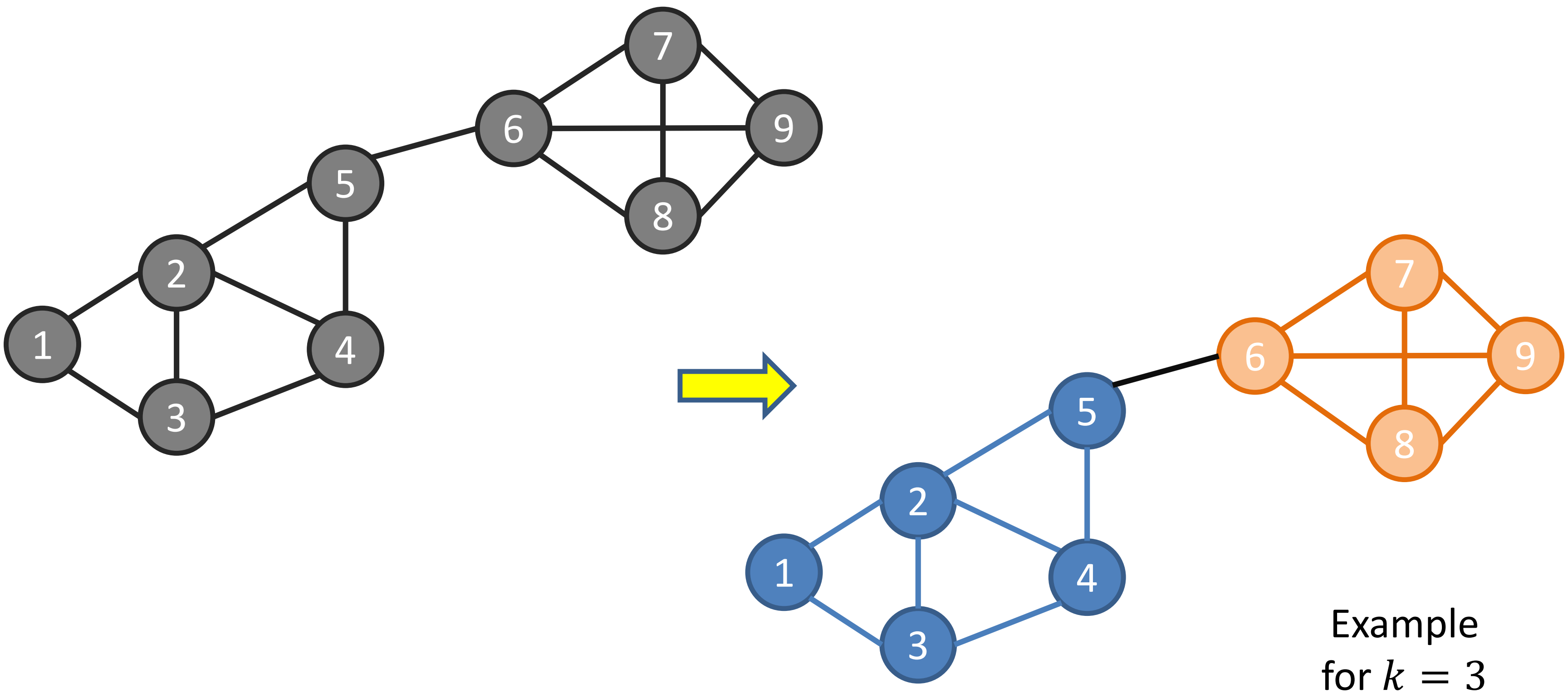
k-clique

- The idea is to look for *adjacent k-cliques*
- A *k-clique* is a clique of size k (with k vertices)
- *Adjacency* for k-cliques means sharing $k-1$ vertices
- Community can be viewed as a maximum set of adjacent k-cliques
  - Maximum means that no more k-cliques form the graph can be added



Example: two adjacent 3-cliques

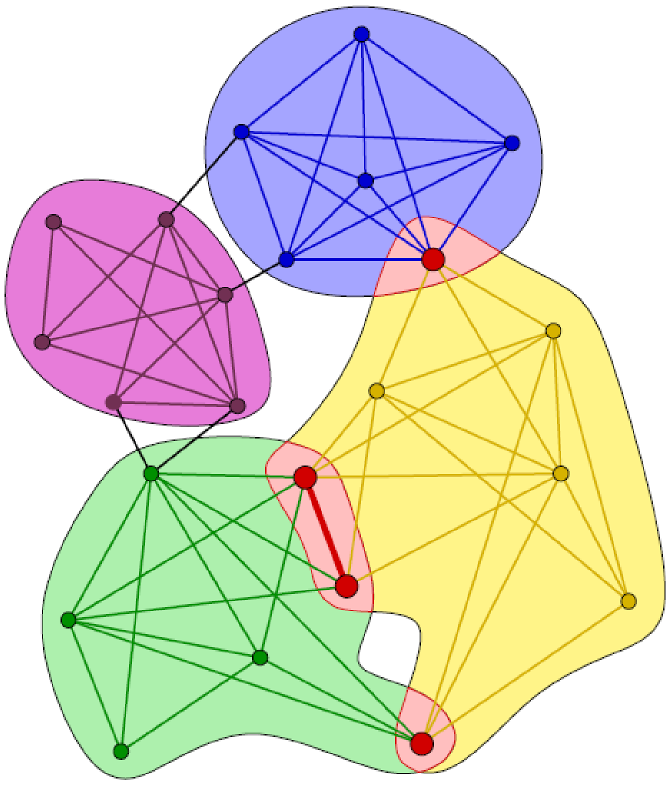
Clique percolation method (CPM)



Example
for $k = 3$

- What about overlapping communities?

CPM can detect overlapping communities



Example
for $k = 4$

Image: Palla et al., Uncovering the overlapping community structure of complex networks in nature and society, Nature, 2005

Summary

- Why are there so many definitions of a community in graphs?
- What are some approaches to finding communities?
- What are their strengths and weaknesses?