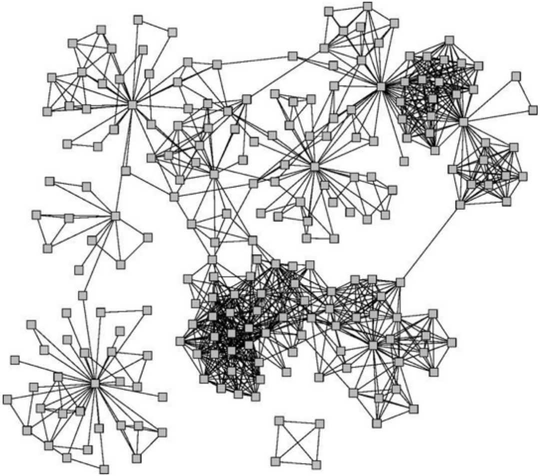
Network structure

Leonid E. Zhukov

School of Data Analysis and Artificial Intelligence
Department of Computer Science
National Research University Higher School of Economics

Network Science

Network structure



Typical network structure

Core-periphery structure of a network

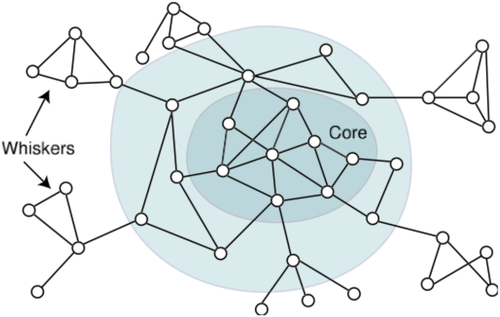
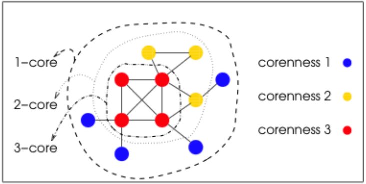
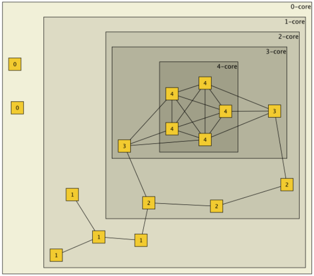


image from J. Leskovec, K. Lang, 2010

Definition

A *k-core* is the largest subgraph such that each vertex is connected to at least k others in subset



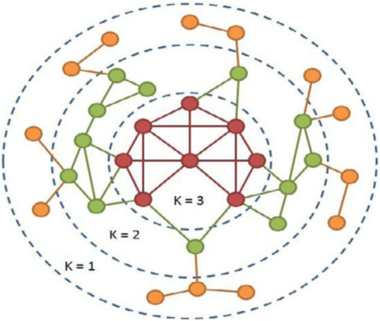


Every vertex in k-core has a degree $k_i \geq k$
$(k+1)$-core is always subgraph of k-core
The core number of a vertex is the highest order of a core that contains this vertex

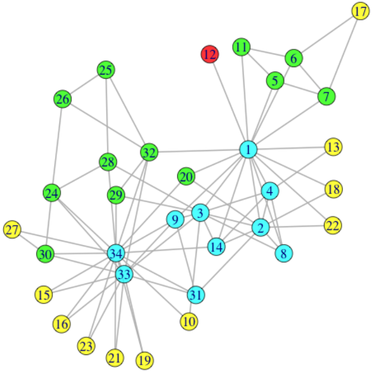
k-core decomposition

V. Batageli, M. Zaversnik, 2002

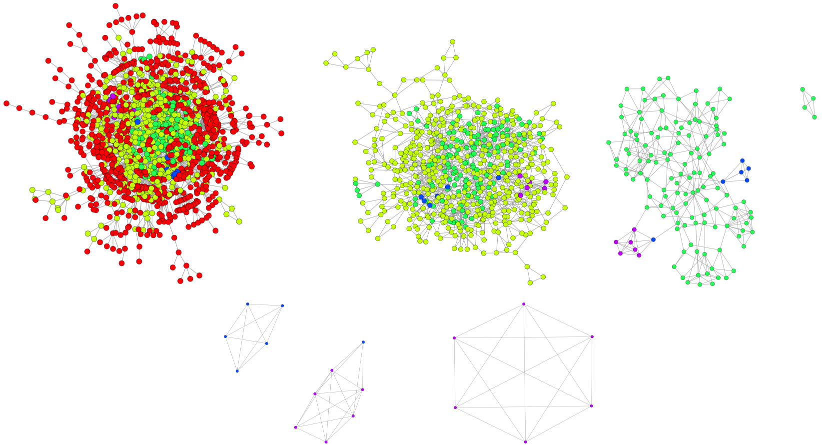
- If from a given graph $G = (V, E)$ recursively delete all vertices, and lines incident with them, of degree less than k, the remaining graph is the k-core.



Zachary karate club: 1,2,3,4 - cores



k-cores



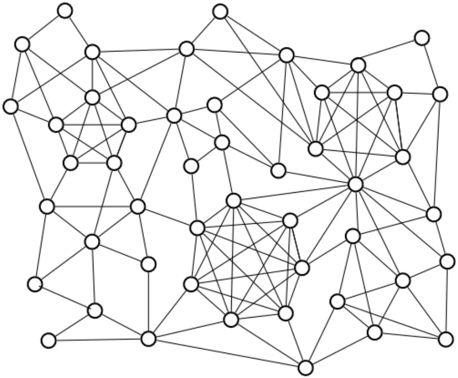
k-cores: 1:1458, 2:594, 3:142, 4:12, 5:6

k-shells: 1:864-red, 2:452-pale green, 3:130-green, 5:6-blue, 6:6-purple

Graph cliques

Definition

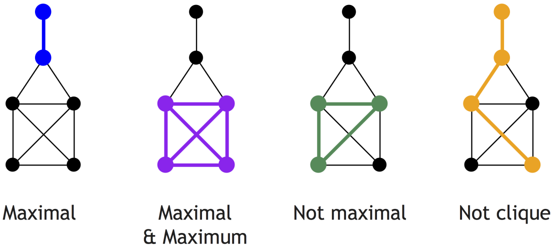
A *clique* is a complete (fully connected) subgraph, i.e. a set of vertices where each pair of vertices is connected.



Cliques can overlap

Graph cliques

- A **maximal clique** is a clique that cannot be extended by including one more adjacent vertex (not included in larger one)
- A **maximum clique** is a clique of the largest possible size in a given graph

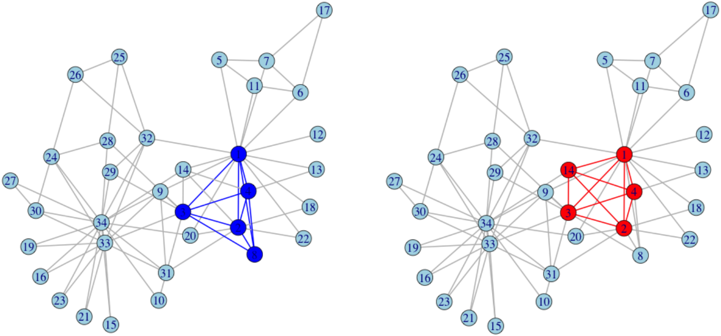


- Graph clique number is the size of the maximum clique

image from D. Eppstein

Graph cliques

Maximum cliques



Maximal cliques:

Clique size:	2	3	4	5
Number of cliques:	11	21	2	2

Zachary, 1977

Graph cliques

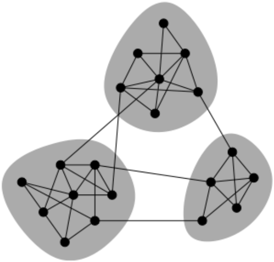
Computational issues:

- Finding click of fixed given size k - $O(n^k k^2)$
- Finding maximum clique $O(3^{n/3})$
- But in sparse graphs...

Network communities

Definition

Network communities are groups of vertices such that vertices inside the group connected with many more edges than between groups.



- Community detection is an assignment of vertices to communities.
- Will consider non-overlapping communities
- Graph partitioning problem

Network communities

What makes a community (cohesive subgroup):

- Mutuality of ties. Almost everyone in the group has ties (edges) to one another
- Compactness. Closeness or reachability of group members in small number of steps, not necessarily adjacency
- Density of edges. High frequency of ties within the group
- Separation. Higher frequency of ties among group members compared to non-members

Wasserman and Faust

Community density

- Graph $G(V, E)$, $n = |V|$, $m = |E|$
- Community - set of nodes S
  n_s-number of nodes in S, m_s - number of edges in S
- Graph density

$$\rho = \frac{m}{n(n-1)/2}$$

- community internal density

$$\delta_{int} = \frac{m_s}{n_s(n_s-1)/2}$$

- external edges density

$$\delta_{ext} = \frac{m_{ext}}{n_s(n-n_s)}$$

- community (cluster): $\delta_{int} > \rho$, $\delta_{ext} < \rho$

Modularity

- Compare fraction of edges within the cluster to expected fraction in random graph with identical degree sequence

$$Q = \frac{1}{4}(m_s - E(m_s))$$

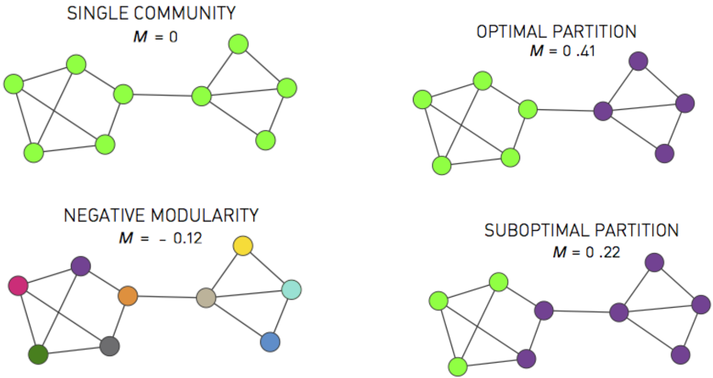
- Modularity score

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j), = \sum_u \left(\frac{m_u}{m} - \left(\frac{k_u}{2m}\right)^2\right)$$

m_u - number of internal edges in a community u,
k_u - sum of node degrees within a community

- Modularity score range $Q \in [-1/2, 1)$, single community $Q = 0$

Modularity



- The higher the modularity score - the better are communities

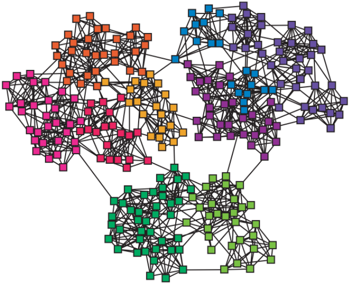
from A.L. Barabasi 2016

Heuristic approach

Focus on edges that connect communities.

Edge betweenness -number of shortest paths $\sigma_{st}(e)$ going through edge e

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$



Construct communities by progressively removing edges

Edge betweenness

Newman-Girvan, 2004

Algorithm: Edge Betweenness

Input: graph G(V,E)

Output: Dendrogram

repeat

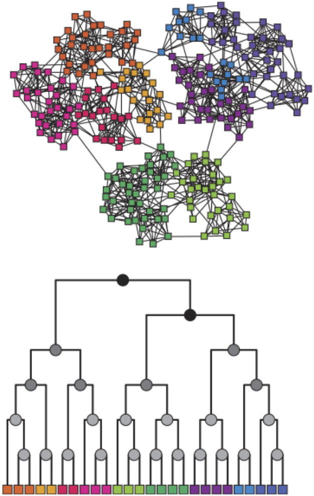
  For all $e \in E$ compute edge betweenness $C_B(e)$;

  remove edge e_i with largest $C_B(e_i)$;

until *edges left*;

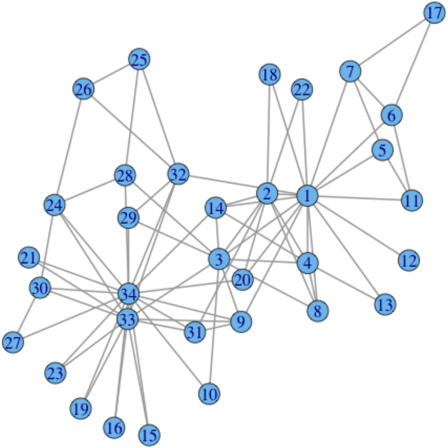
If bi-partition, then stop when graph splits in two components (check for connectedness)

Hierarchical algorithm, dendrogram



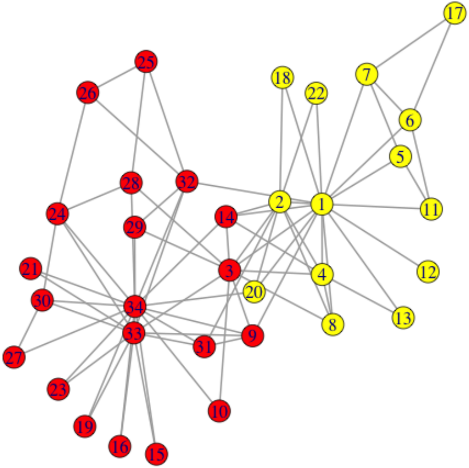
Edge betweenness

Zachary karate club



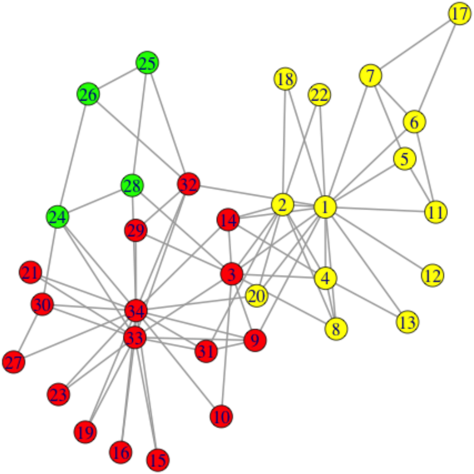
Edge betweenness

Zachary karate club

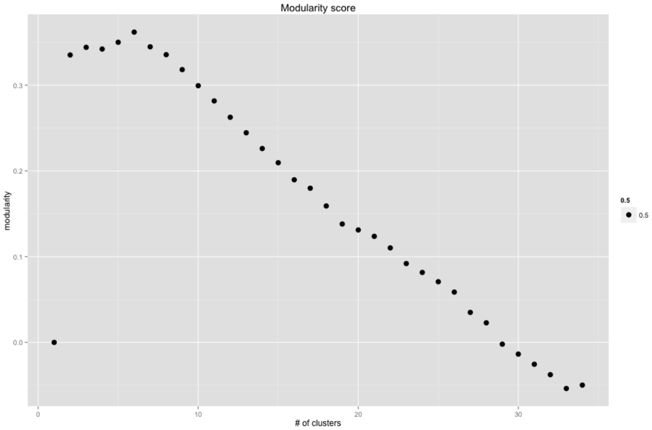


Edge betweenness

Zachary karate club

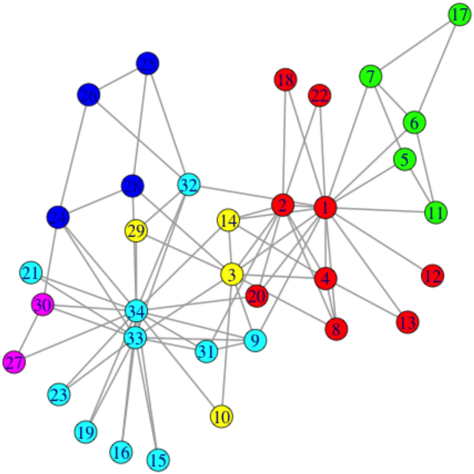


Edge betweenness



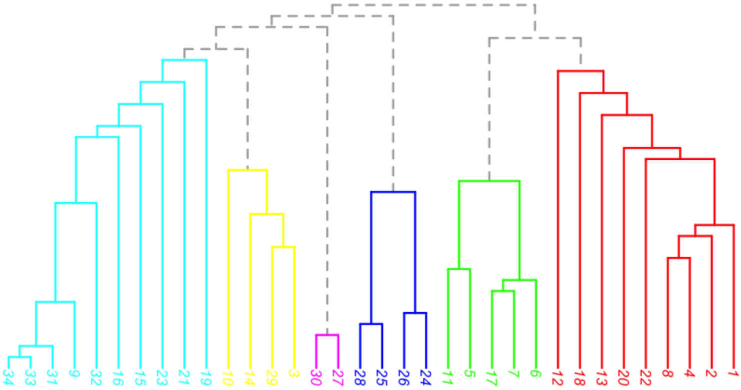
Edge betweenness

best: clusters = 6, modularity = 0.345



Edge betweenness

Zachary karate club



References

- Finding and evaluating community structure in networks, M.E.J. Newman, M. Girvan, Phys. Rev E, 69, 2004
- Modularity and community structure in networks, M.E.J. Newman, PNAS, vol 103, no 26, pp 8577-8582, 2006
- S. E. Schaeffer. Graph clustering. Computer Science Review, 1(1):2764, 2007.
- S. Fortunato. Community detection in graphs, Physics Reports, Vol. 486, Iss. 35, pp 75-174, 2010