Community Detection

Vasilii Latonov

Higher School of Economics

wlatonov@gmail.com

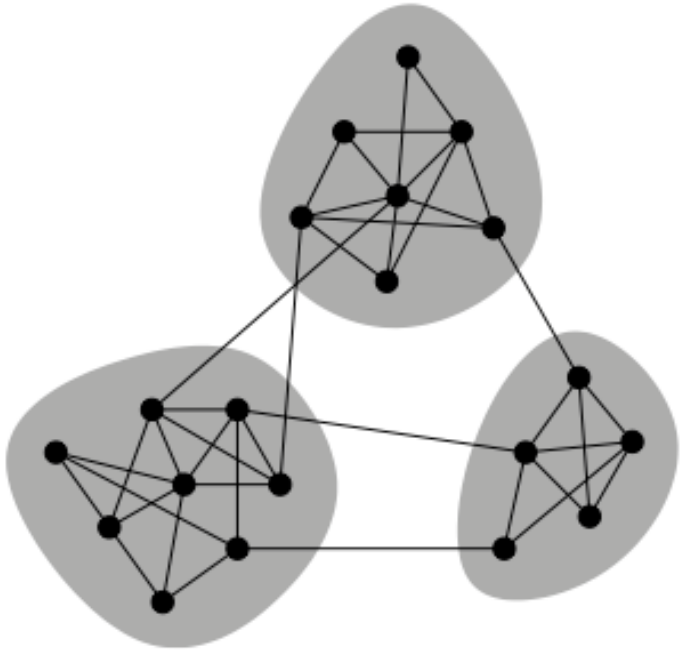
Structural Analysis and Visualization of Networks
01.03.2024

Presentation Overview

Network communities

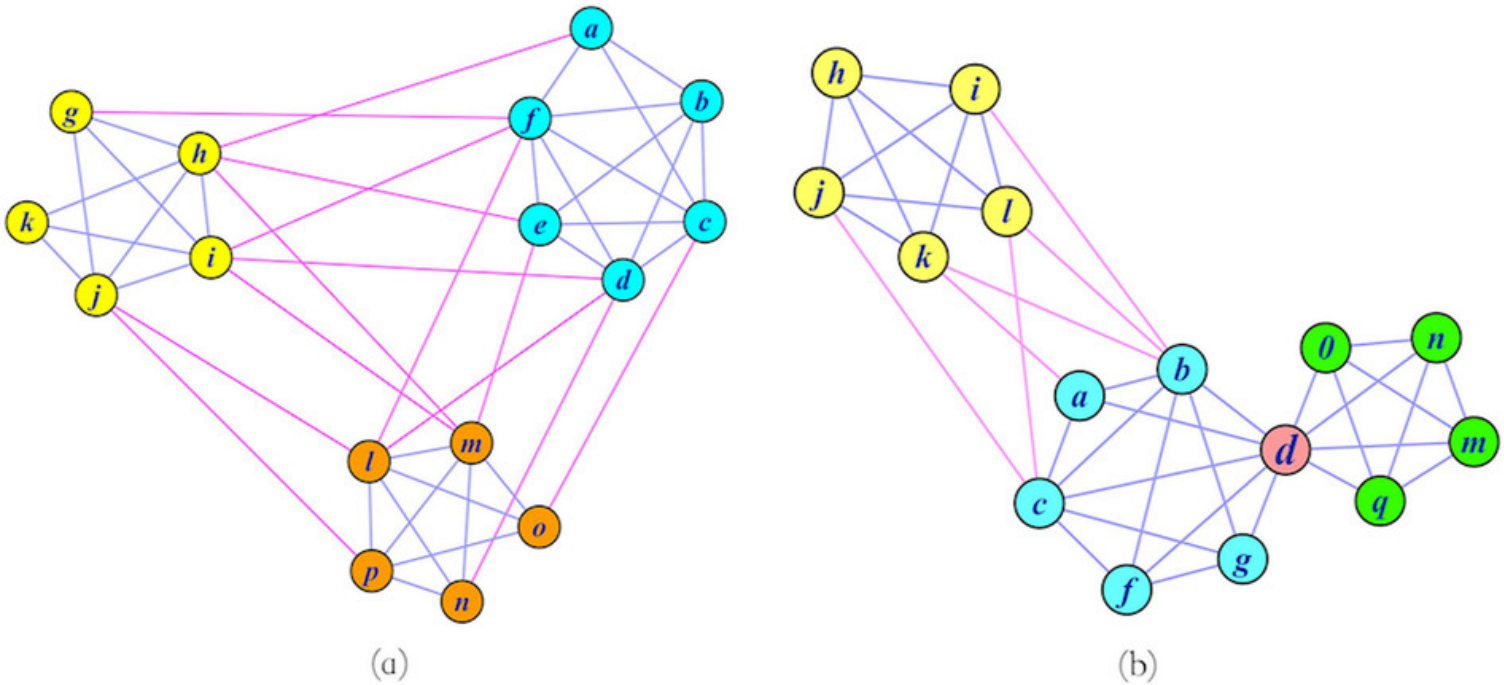
Definition

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.

Overlapping and non-overlapping communities



What makes a community?

Community is a 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

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 $= \frac{m}{n(n-1)2}$
- Community internal density $\sigma_{int} = \frac{m_s}{n_s(n_s-1)2}$
- External edges density $\sigma_{ext} = \frac{m_{ext}}{n_s(n-n_s)}$
- Community (cluster): $\sigma_{int} > p$, $\sigma_{ext} < p$

Modularity

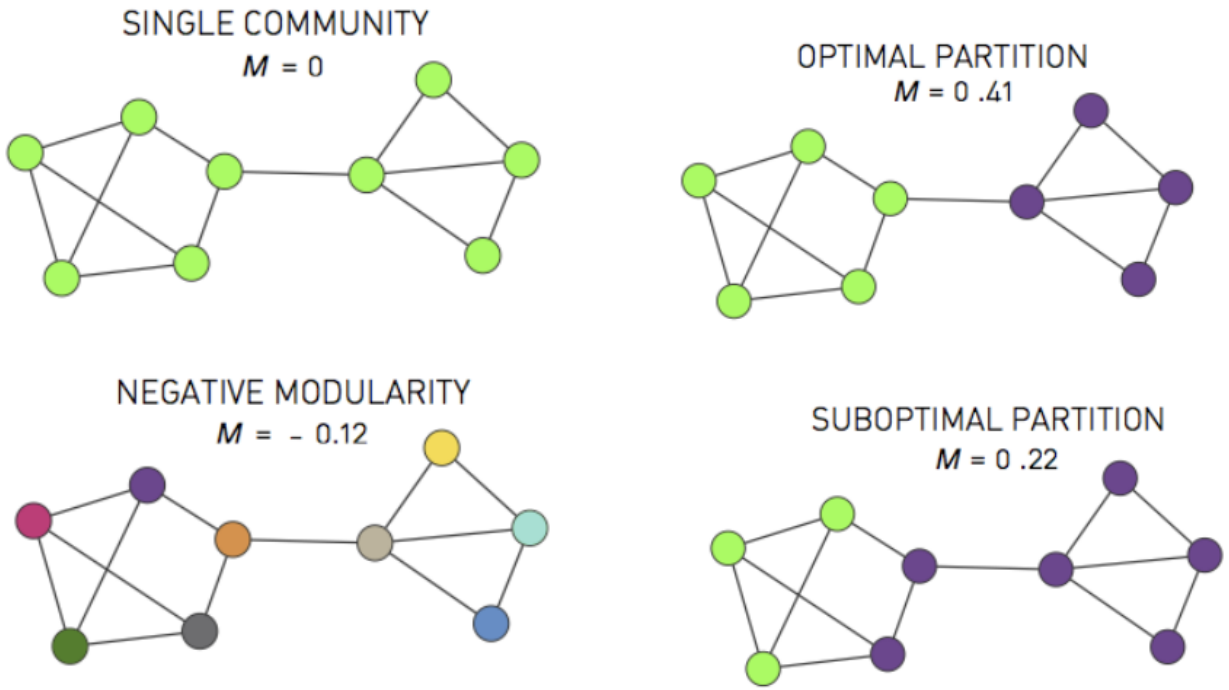
- Compare fraction of edges within the cluster to expected fraction in random graph with identical degree sequence
- Modularity score

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

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

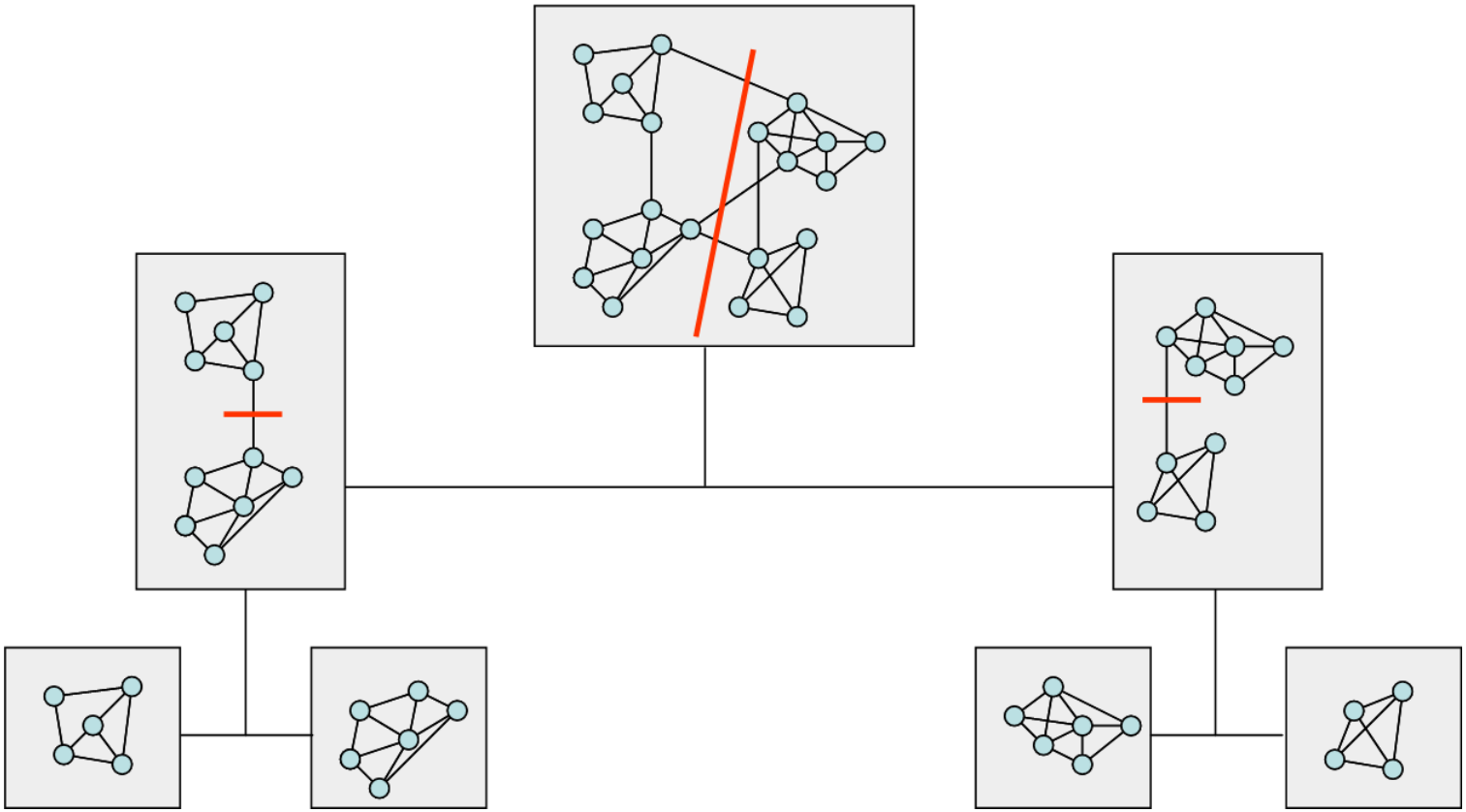
- Modularity score range Q [-1/2, 1), single community $Q = 0$

Modularity



The higher the modularity score - the better are communities

Community detection by graph partitioning



Recursive graph partitioning

Graph partitioning

Exact solution NP-hard

- Number of ways to divide network of n nodes in 2 groups (bi-partition): 2N-1 - 1 ways - distinct cuts
- Combinatorial optimization problem:
  - optimization criterion
  - optimization method
- Solved by greedy, approximate algorithms or heuristics
- Balanced vs unbalanced partition
- 2 way-partiton vs multiway parittion

Graph partitioning algorithms

- Greedy optimization:
  Local search [Kernighan and Lin, 1970], [Fidducia and Mettheyses, 1982]
- Approximate optimization:
  Spectral graph partitioning [M. Fiedler, 1972], [Pothen et al 1990], [Shi and Malik, 2000]
- Randomized algorithms:
  Randomized min cut [D. Karger, 1993]
- Heuristics algorithms:
  Multilevel graph partitioning (METIS) [G. Karypis, Kumar 1998]

Finding optimal partition

Graph $G(E, V)$ partition: $V = V_1 + V_2$

- Graph cut

$$Q = cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} e_{ij}$$

- Ratio cut:

$$Q = \frac{cut(V_1, V_2)}{||V_1||} + \frac{cut(V_1, V_2)}{||V_2||}$$

- Normalized cut:

$$Q = \frac{cut(V_1, V_2)}{Vol(V_1)} + \frac{cut(V_1, V_2)}{Vol(V_2)}$$

- Quotient cut (conductance):

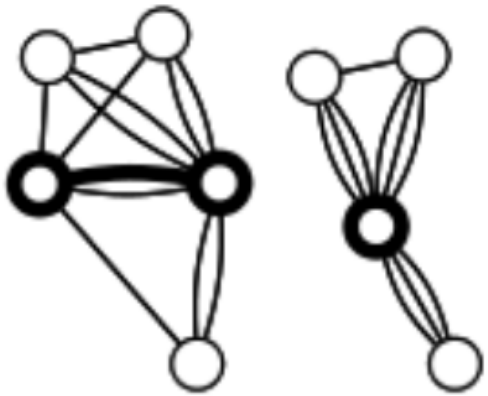
$$Q = \frac{cut(V_1, V_2)}{\min\left(Vol(V_1), Vol(V_2)\right)}$$

where: $Vol(V_1) = \sum_{i \in V_1, j \in V} e_{ij} = \sum_{i \in V_1} k_i$

Randomized min cut

Karger's algorithm for finding minimum cut

- Edge contraction - removing an edge and merging two vertices that it previously joined



- $P(\textit{final cut} = \textit{mincut}) \geq 2/n^2$
- Run it $\Omega(n^2)$ times and choose the smallest cut

Karger min cut

Input: Graph **G**

Output: class indicator vector *mincutX*

$X = \text{inf}$

for $i = 1 : N$

  $X = GuessMinCut(G)$

  if $|X| < mincutk$

  $mincutX = X, mincutk = |X|$

return *mincutX*

for $i = N$ down to 2

  pick a random edge e in G

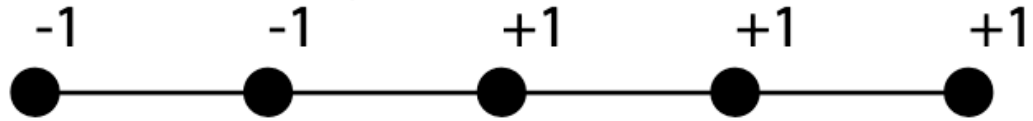
  $G = G/e$

return the only cut in G

Karger's algoirthm

Linear algebra of graph cuts

- Let $V = V^+ + V^-$ be partitioning of the nodes
- Let $\mathbf{s} = \{+1, -1, +1, \dots -1, +1\}^T$ - indicator vector



$$s(i) = \begin{cases} +1: & \text{if } v(i) \in V^+ \\ -1: & \text{if } v(i) \in V^- \end{cases}$$

- Number of edges, connecting V^+ and V^-

$$cut(V^+, V^-) = \frac{1}{4}\sum_{e(i,j)}(s(i) - s(j))^2 = \frac{1}{8}\sum_{i,j}A_{ij}(s(i) - s(j))^2 =$$

$$= \frac{1}{4}\sum_{i,j}(k_i\delta_{ij}s(i)^2 - A_{ij}s(i)s(j)) = \frac{1}{4}\sum_{i,j}(k_i\delta_{ij} - A_{ij})s(i)s(j)$$

$$cut(V^+, V^-) = \frac{1}{4}\sum_{i,j}(D_{ij} - A_{ij})s(i)s(j)$$

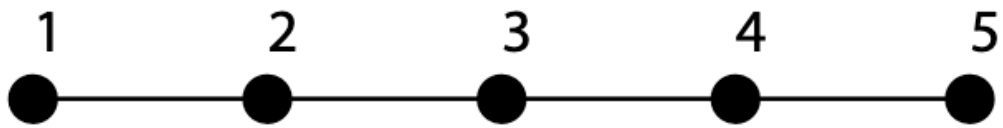
Linear algebra of graph cuts

- Graph Laplacian: $\mathbf{L}_{ij} = \mathbf{D}_{ij} - \mathbf{A}_{ij}$, where $\mathbf{D}_{ii} = diag(k_i)$

$$\mathbf{L}_{ij} = \begin{cases} k(i), & \text{if } i = j \\ -1, & \text{if } \exists\, e(i,j) \\ 0, & \text{otherwise} \end{cases}$$

- Laplacian matrix 5x5:

$$\mathbf{L} = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}$$



Graph cuts

- Graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- Graph cut:

$$Q(\mathbf{s}) = cut(V^+, V^-) = \frac{1}{4}\sum_{i,j} L_{ij}s(i)s(j) = \frac{\mathbf{s}^T\mathbf{L}\mathbf{s}}{4}$$

- Minimal cut:

$$\min_{\mathbf{s}} Q(\mathbf{s})$$

- Balanced cut constraint:

$$\sum_i s(i) = 0$$

- Integer minimization problem, exact solution is NP-hard!

Spectral method - relaxation

- Discrete problem $\rightarrow$ continuous problem
- Discrete problem: find

$$\min_{\mathbf{s}}(\frac{1}{4}\mathbf{s}^T\mathbf{L}\mathbf{s})$$

under constraints: $s(i) = \pm 1, \sum_i s(i) = 0$;

- Relaxation - continuous problem: find

$$\min_{\mathbf{x}}(\frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x})$$

under constraints: $\sum_i x(i)^2 = n\,, \sum_i x(i) = 0$

- Given $x(i)$, round them up by $s(i) = sign(x(i))$
- Exact constraint satisfies relaxed equation, but not other way around!

Spectral method - computations

- Constraint optimization problem (Lagrange multipliers):

$$Q(\mathbf{x}) = \frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - n), \;\; \mathbf{x}^T\mathbf{e} = 0$$

- Eigenvalue problem:

$$\mathbf{L}\mathbf{x} = \lambda\mathbf{x}, \;\; \mathbf{x} \perp \mathbf{e}$$

- Solution:

$$Q(\mathbf{x_i}) = \frac{n}{4}\lambda_i$$

- First (smallest) eigenvector:

$$\mathbf{L}\mathbf{e} = 0, \;\; \lambda = 0, \;\; \mathbf{x_1} = \mathbf{e}$$

- Looking for the second smallest eigenvalue/eigenvector λ_2 and $\mathbf{x}_2$
- Minimization of Rayleigh-Ritz quotient:

$$\min_{\mathbf{x} \perp \mathbf{x}_1} \left(\frac{\mathbf{x}^T\mathbf{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right)$$

Spectral graph theory

- $\lambda_1 = 0$
- Number of $\lambda_i = 0$ equal to the number of connected components

- $0 \leq \lambda_2 \leq 2$
  $\lambda_2 = 0$, disconnected graph
  $\lambda_2 = 1$, totally connected

- Graph diameter (longest shortest path)

$$D(G) >= \frac{4}{n\lambda_2}$$

Spectral graph partitioning algorithm

Algorithm: Spectral graph partitioning - normalized cuts

Input: adjacency matrix $\mathbf{A}$

Output: class indicator vector $\mathbf{s}$

compute $\mathbf{D} = diag(deg(\mathbf{A}))$;

compute $\mathbf{L} = \mathbf{D} - \mathbf{A}$;

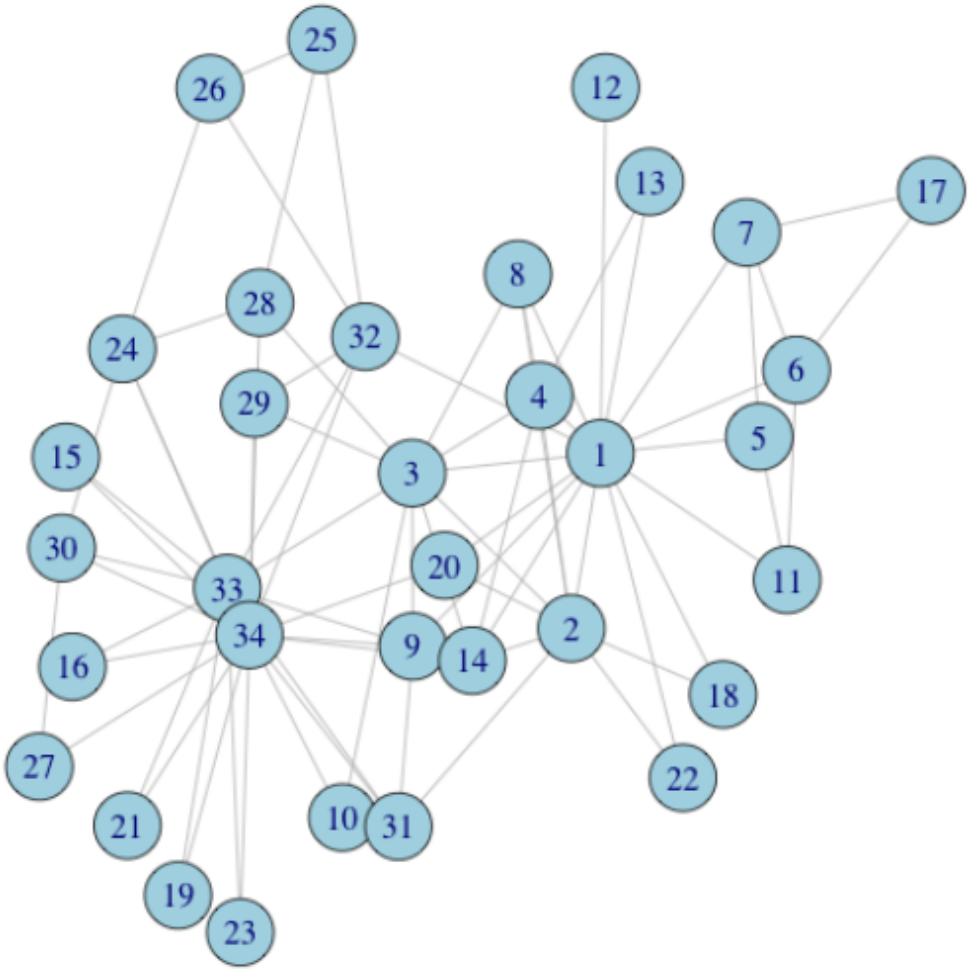
solve for second smallest eigenvector:

min cut: $\mathbf{Lx} = \lambda \mathbf{x}$;

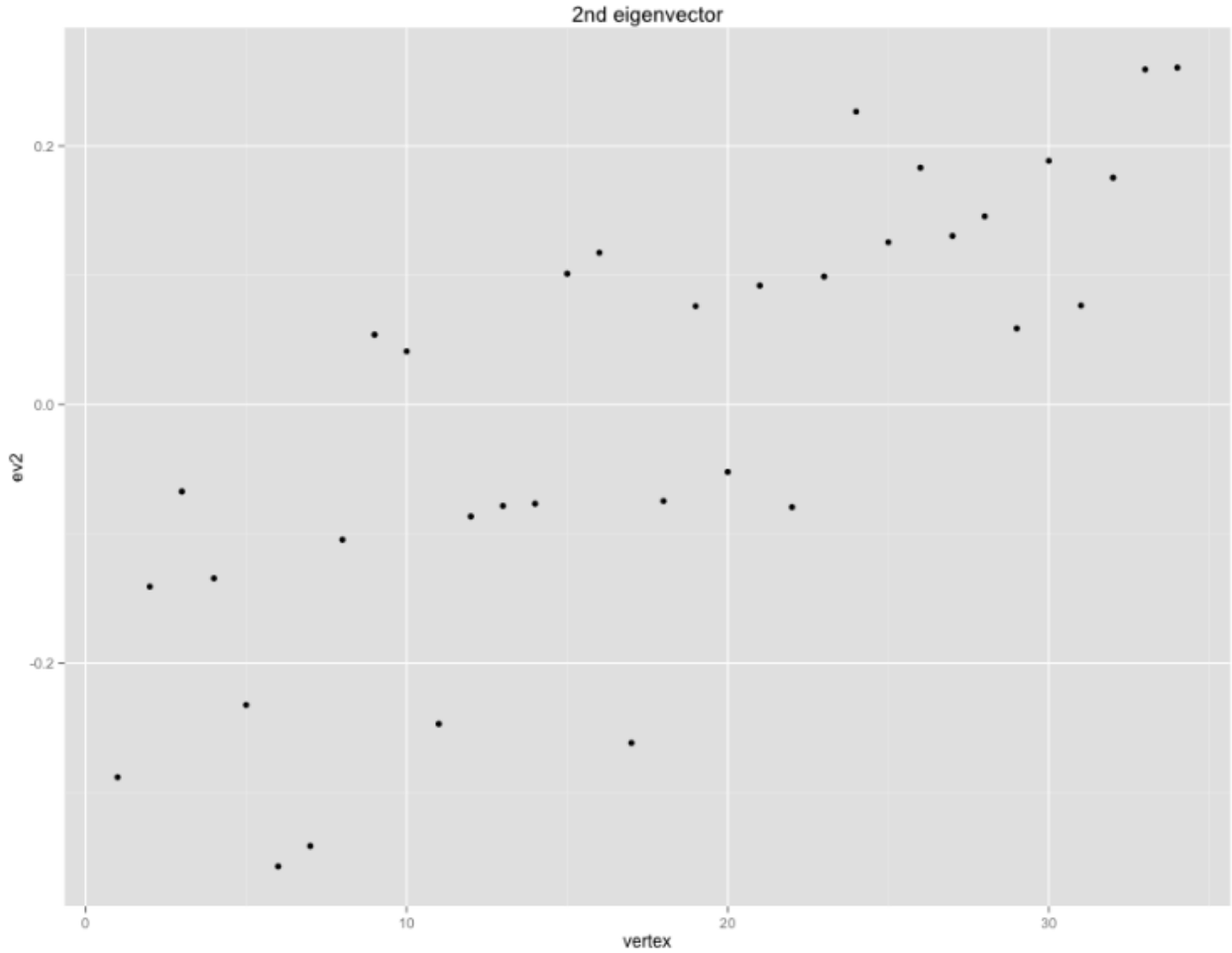
normalized cut : $\mathbf{Lx} = \lambda \mathbf{Dx}$;

set $\mathbf{s} = sign(\mathbf{x}_2)$

Example Graph cuts

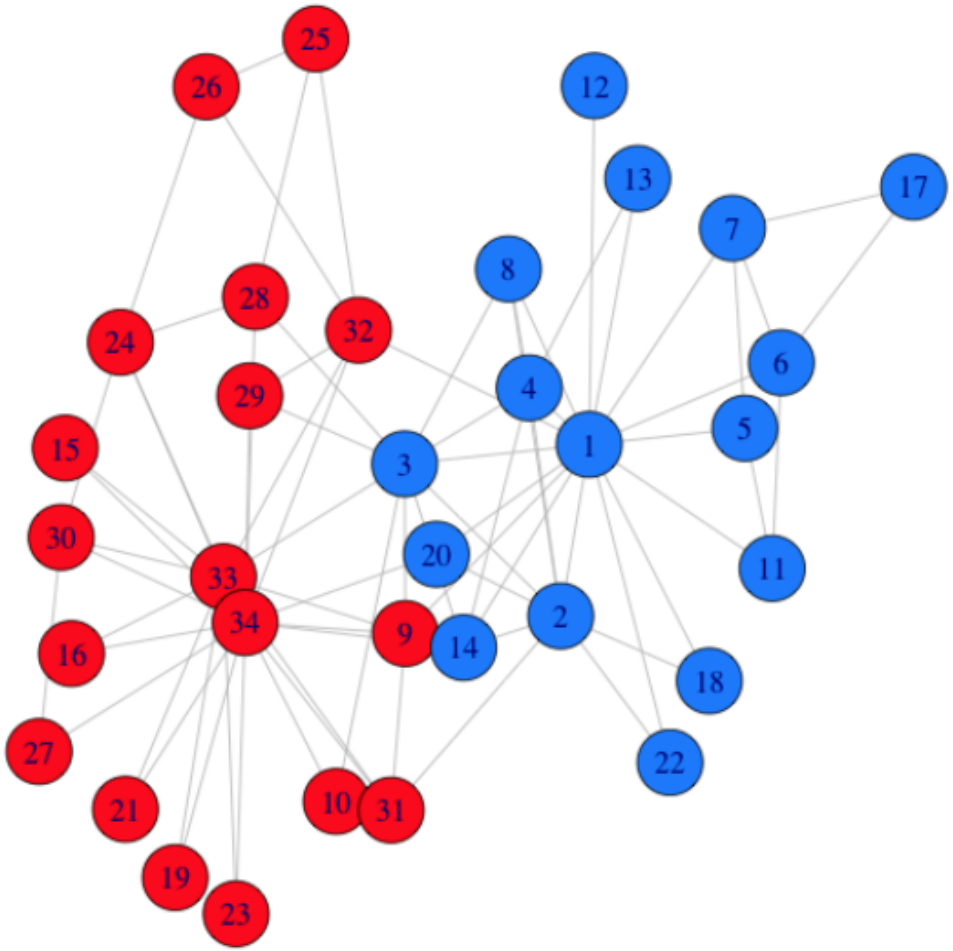


Example Graph cuts

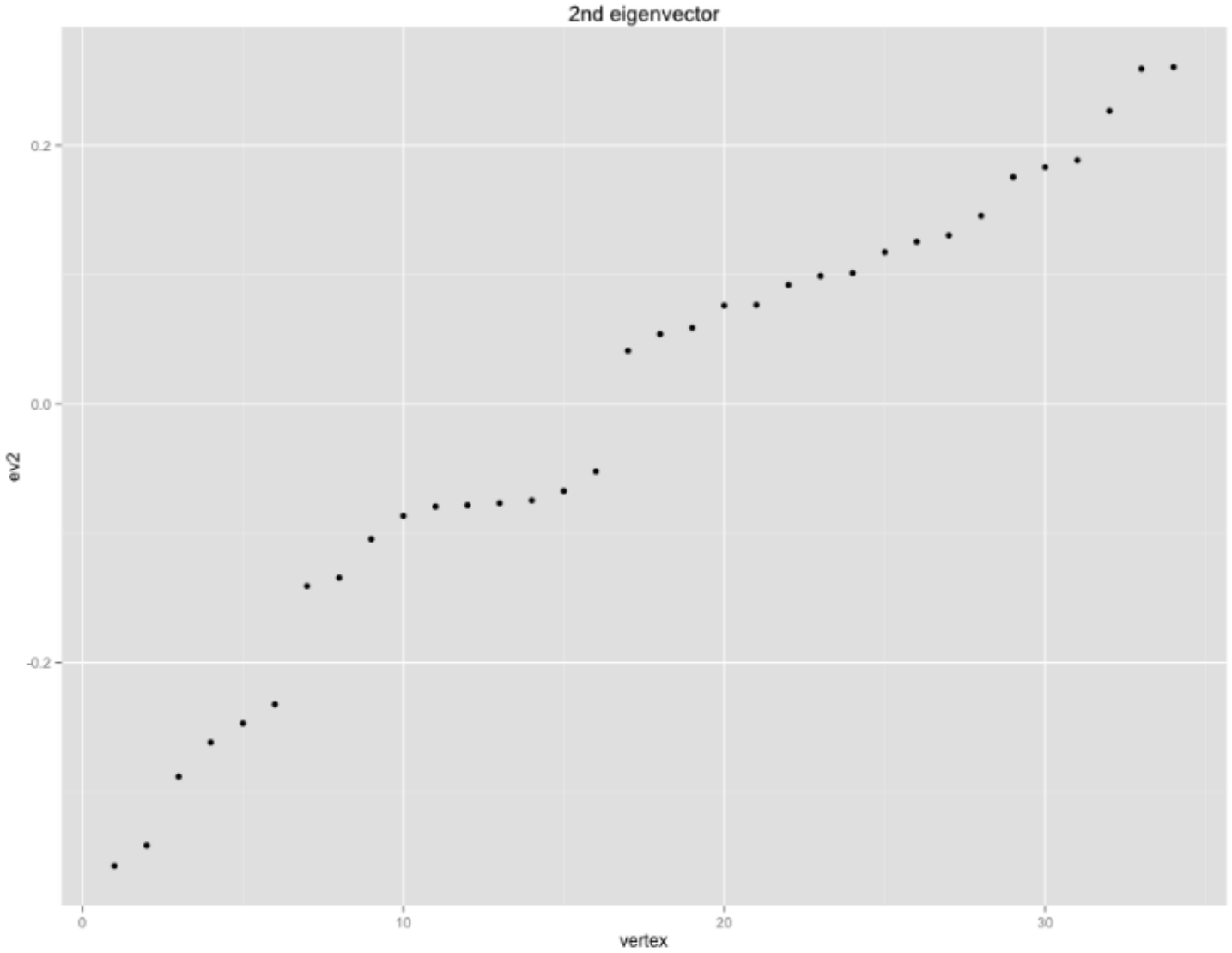


Eigenvalues: $\lambda_1 = 0, \lambda_2 = 0.2, \lambda_3 = 0.25$ ...

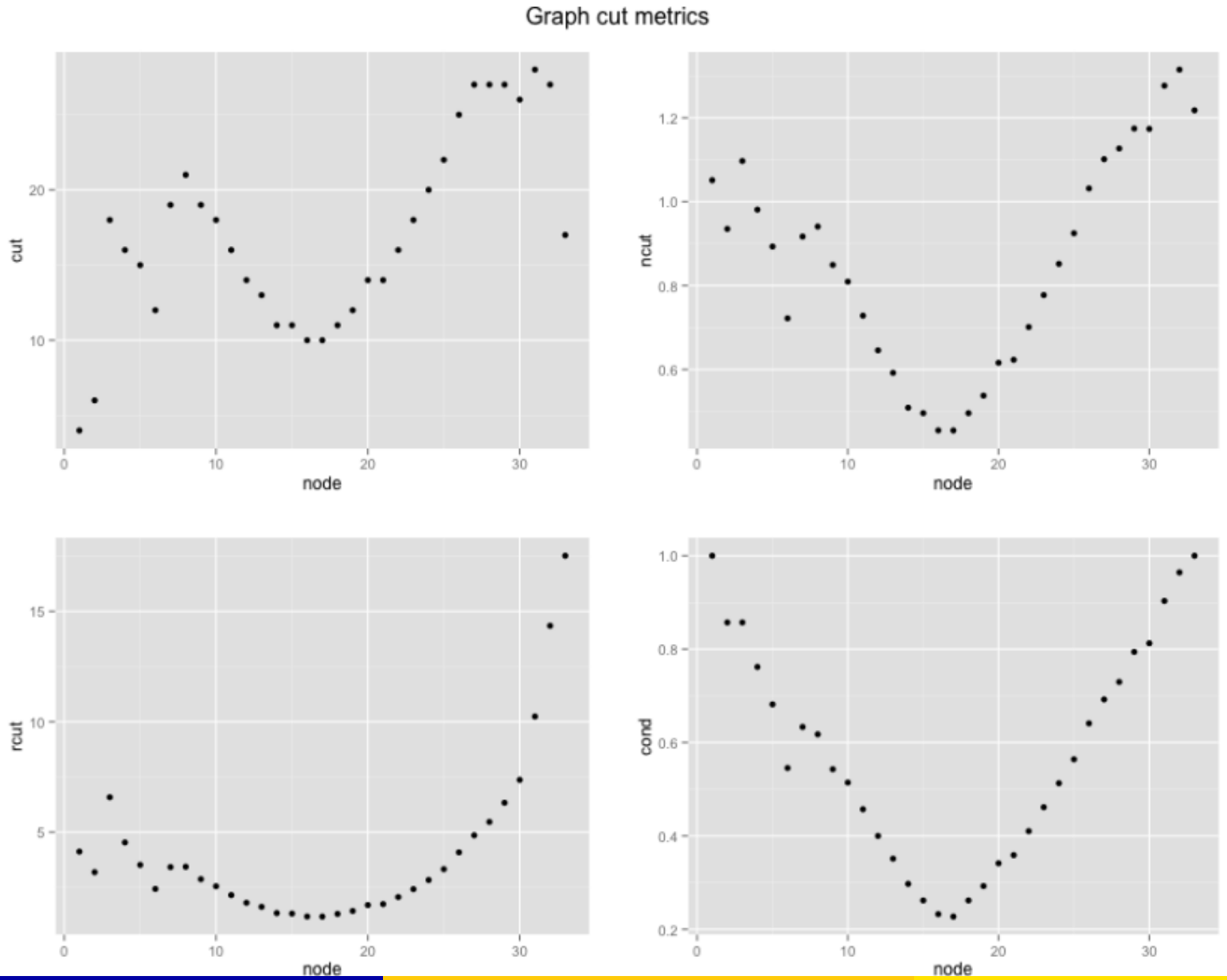
Example Normalized Graph cuts



Example Normalized Graph cuts



Cut metrics



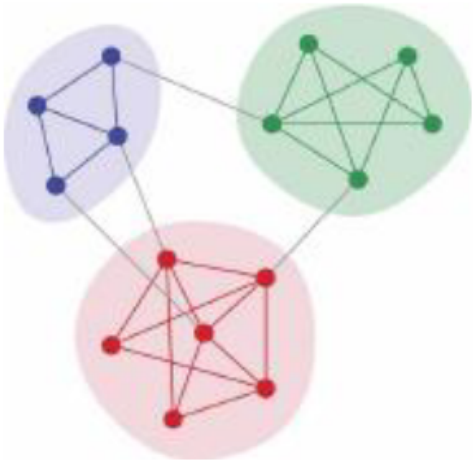
Optimization criterion: modularity

- Modularity:

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

where n_c - number of classes and

$$\delta(c_i, c_j) = \begin{cases} 1: & \text{if } c_i = c_j \\ 0: & \text{if } c_i \neq c_j \end{cases} \text{ - kronecker delta}$$



[Maximization!]

Spectral modularity maximization

- Direct modularity maximization for bi-partitioning, [Newman, 2006]
- Let two classes $c_1 = V^+$, $c_2 = V^-$, indicator variable $s = \pm 1$

$$\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1)$$

- Modularity

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{i,j} B_{ij} s_i s_j$$

where

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

M. Newman, 2006

Spectral modularity maximization

- Qudratic form:

$$Q(\mathrm{s}) = \frac{1}{4m}\mathrm{s}^T\mathrm{B}\mathrm{s}$$

- Integer optimization - NP, relaxation $s \to x$, $x \in R$
- Keep norm $||x||^2 = \sum_i x_i^2 = \mathrm{x}^T\mathrm{x} = n$
- Quadratic optimization

$$Q(\mathrm{x}) = \frac{1}{4m}\mathrm{x}^T\mathrm{B}\mathrm{x} - \lambda(\mathrm{x}^T\mathrm{x} - n)$$

- Eigenvector problem

$$\mathrm{B}\mathrm{x}_i = \lambda_i \mathrm{x}_i$$

- Approximate modularity

$$Q(\mathrm{x_i}) = \frac{n}{4m}\lambda_i$$

- Modularity maximization - largest $\lambda = \lambda_{max}$

Modularity maximization

Algorithm: Spectral modularity maximization: two-way partition

Input: adjacency matrix A

Output: class indicator vector s

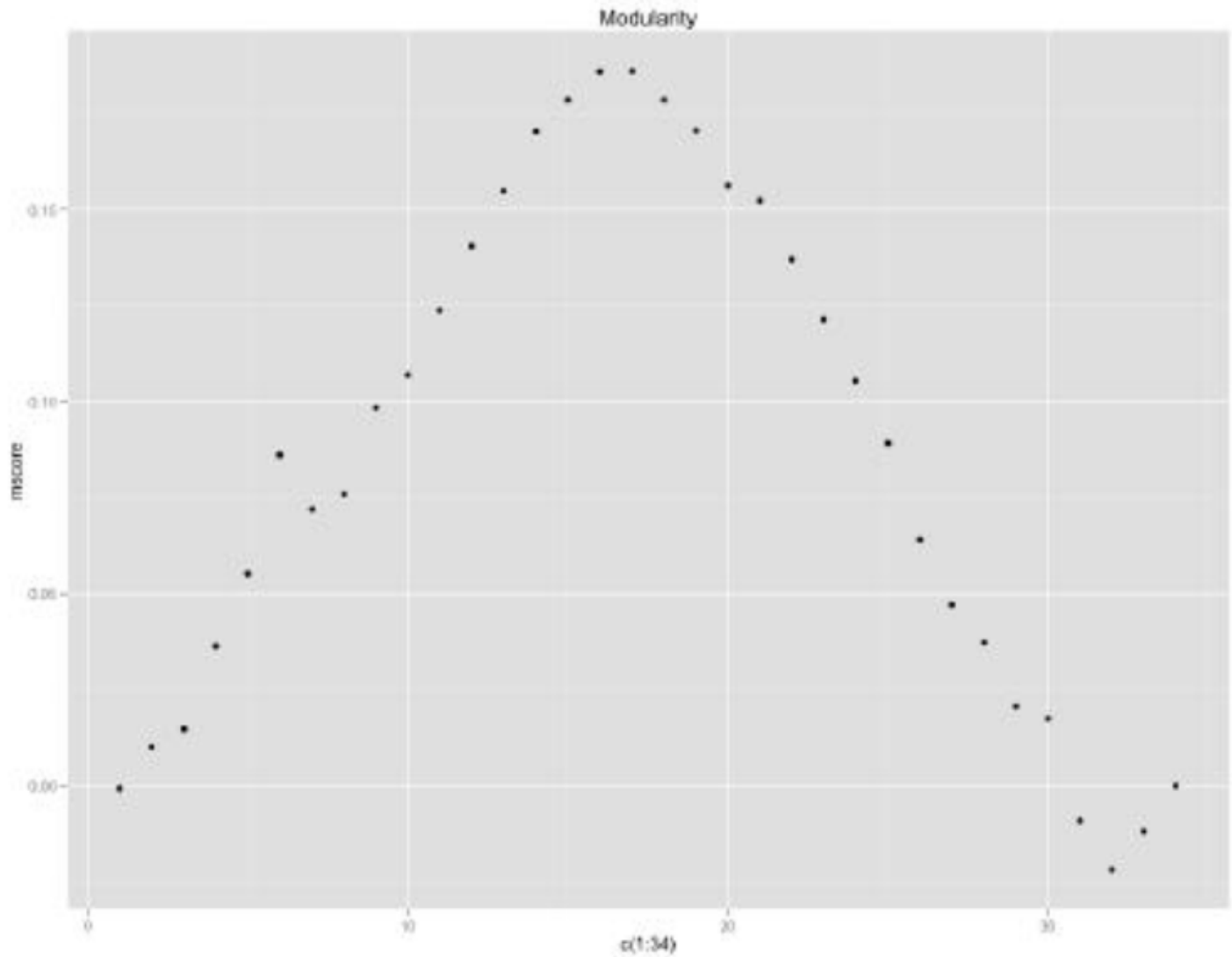
compute $\mathrm{k} = deg(\mathrm{A})$;

compute $\mathrm{B} = \mathrm{A} - \frac{1}{2m}\mathrm{k}\mathrm{k}^T$;

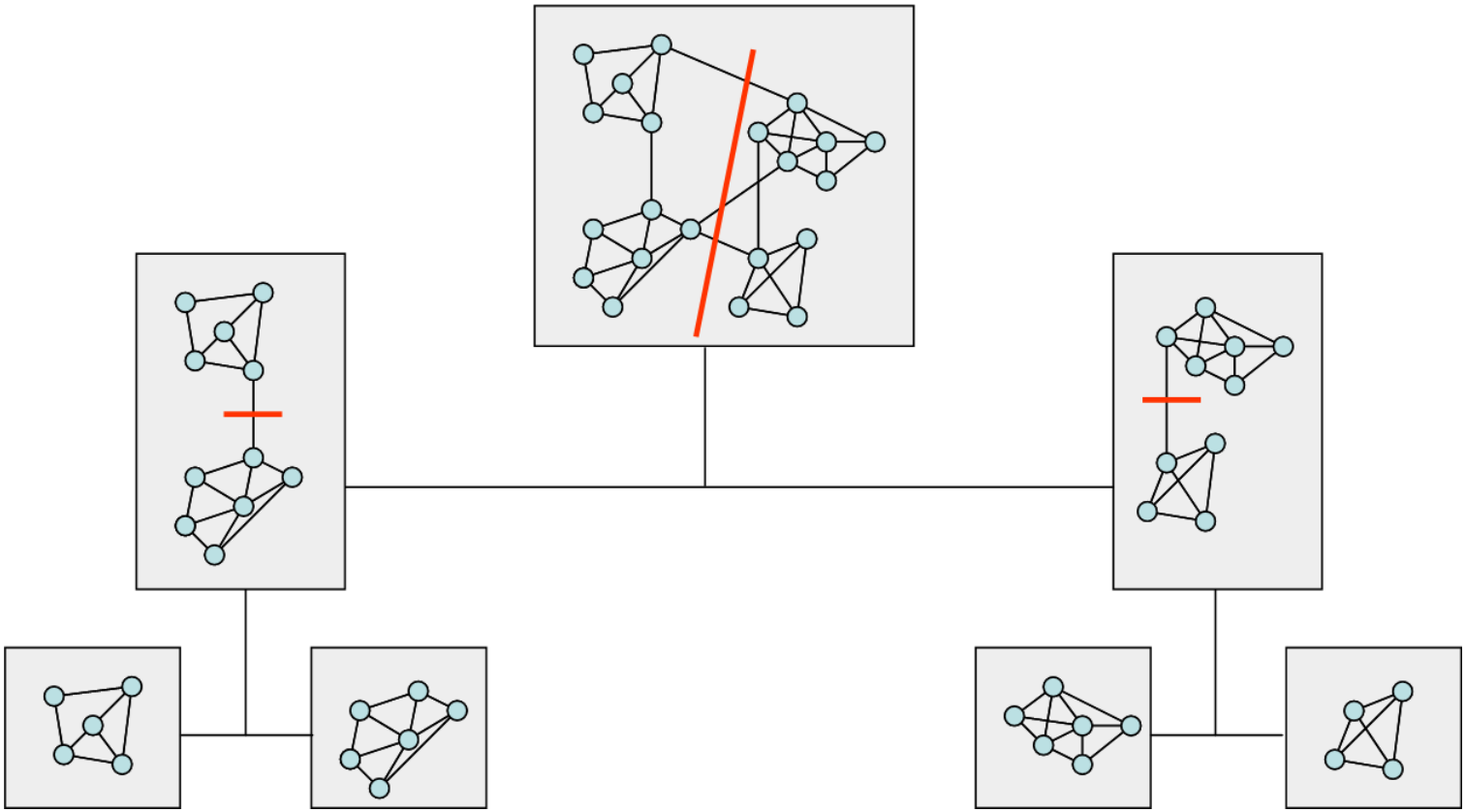
solve for maximal eigenvector $\mathrm{B}\mathrm{x} = \lambda\mathrm{x}$;

set $\mathrm{s} = sign(\mathrm{x}_{max})$

Example

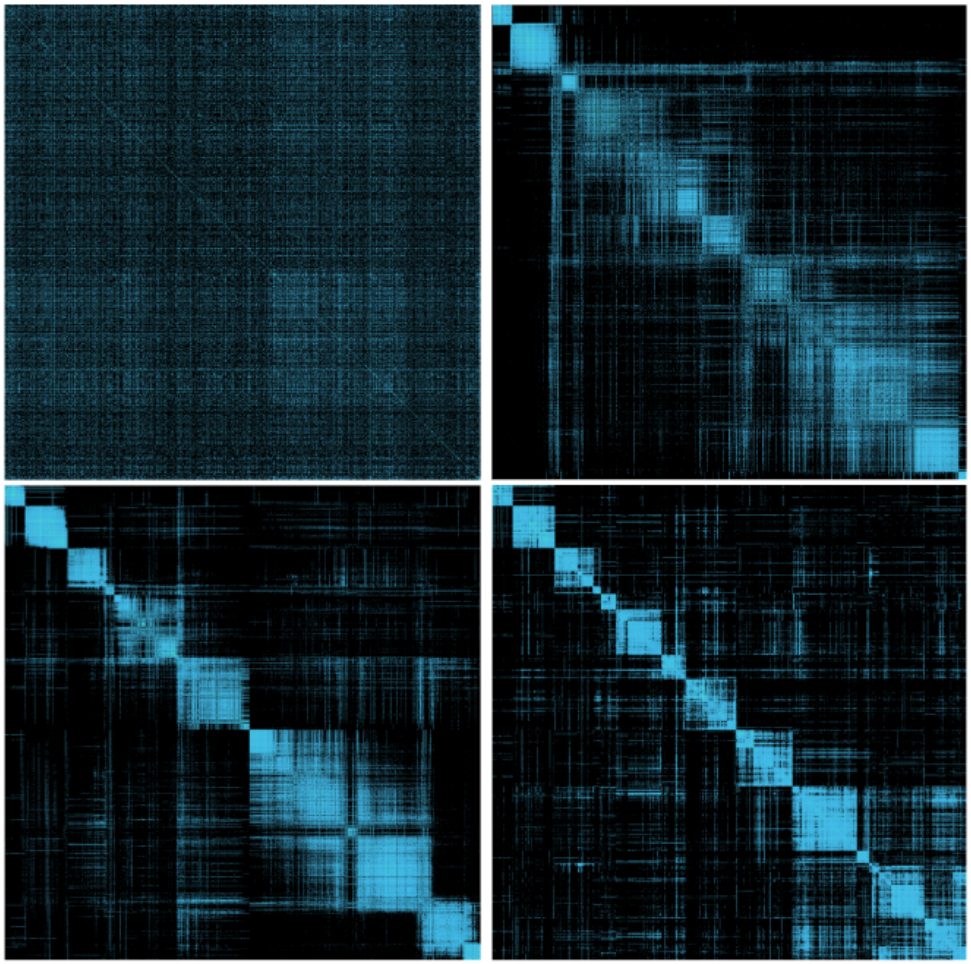


Multilevel spectral



Recursive graph partitioning

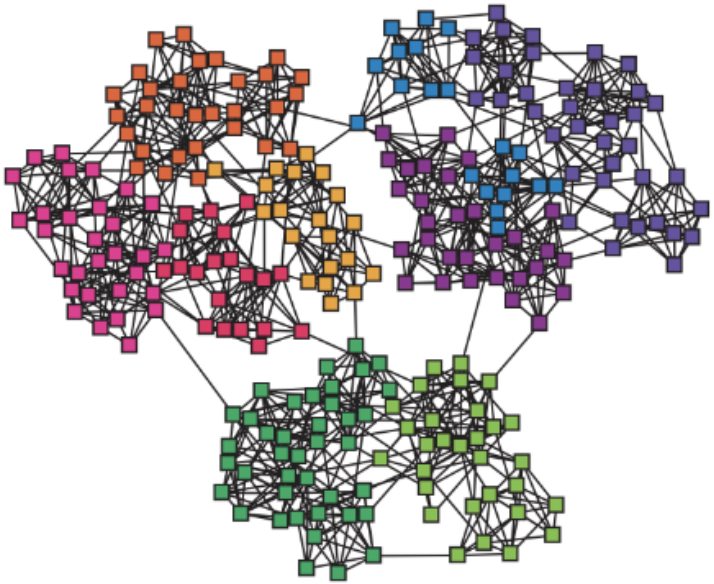
Multilevel spectral



Heuristic approach

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

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



Recover communities by progressively removing edges

Edge Betweenness: Girvan–Newman algorithm

Newman-Girvan, 2004

Algorithm: Edge Betweenness

Input: graph G(V,E)

Output: Dendrogram

repeat

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

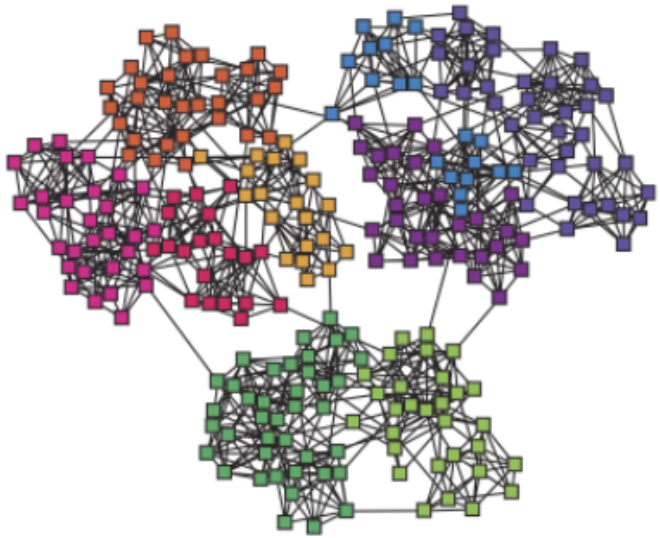
remove edge e_i with largest $C_B(e_i)$;

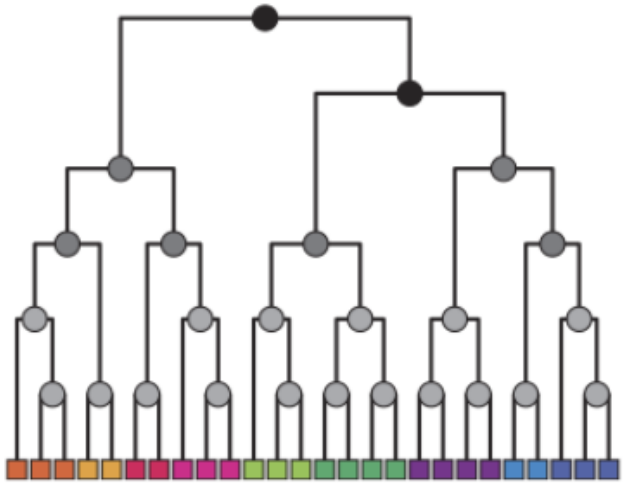
until *edges left*;

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

Edge Betweenness

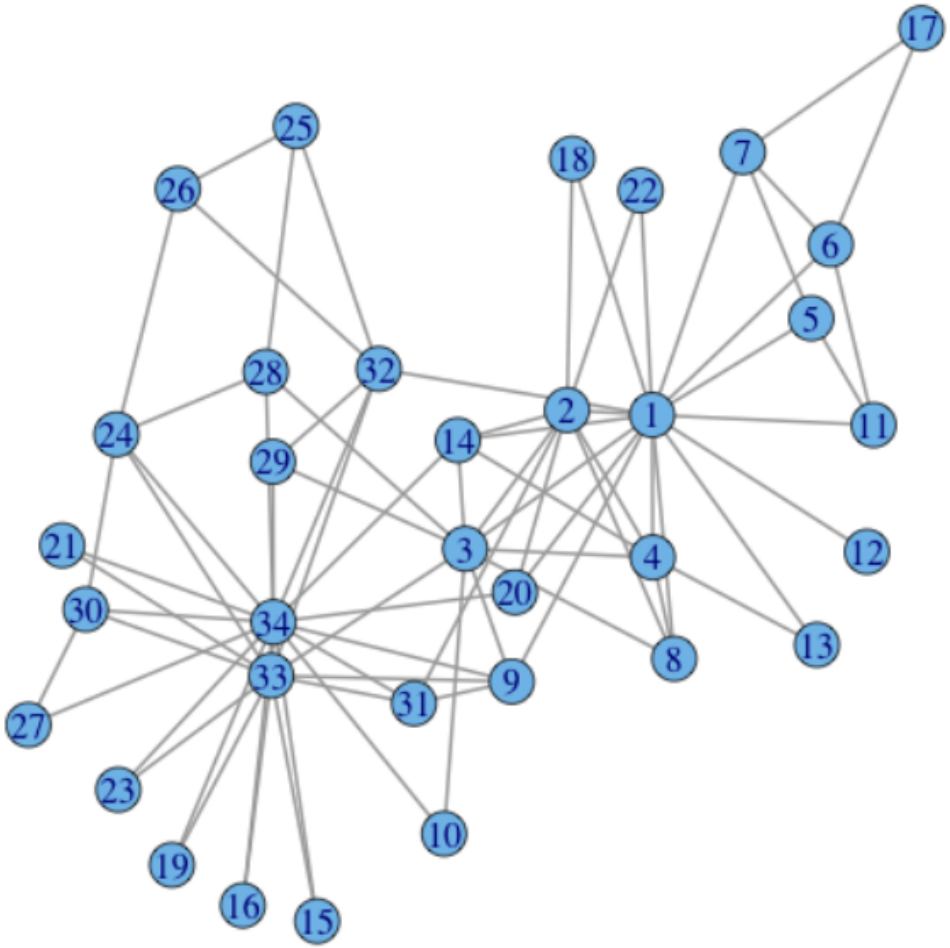
Hierarchical algorithm, dendrogram





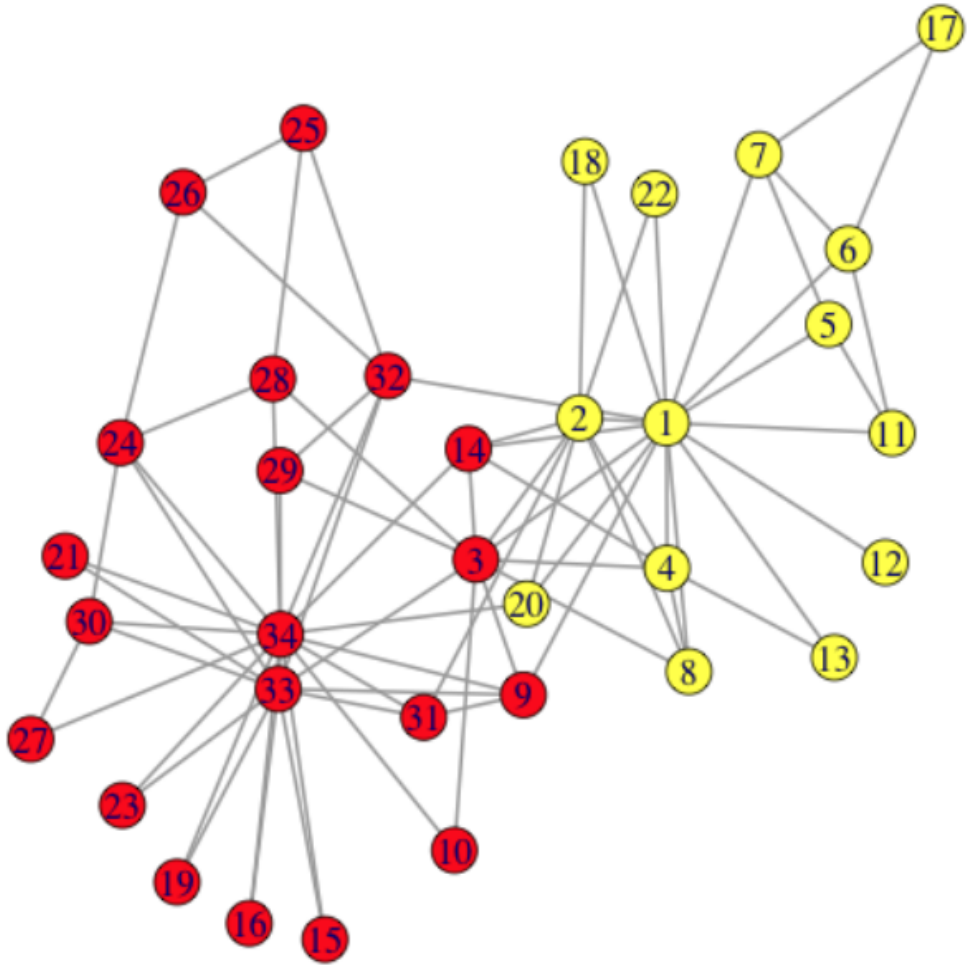
Edge betweenness: Zachary karate club

Zachary karate club



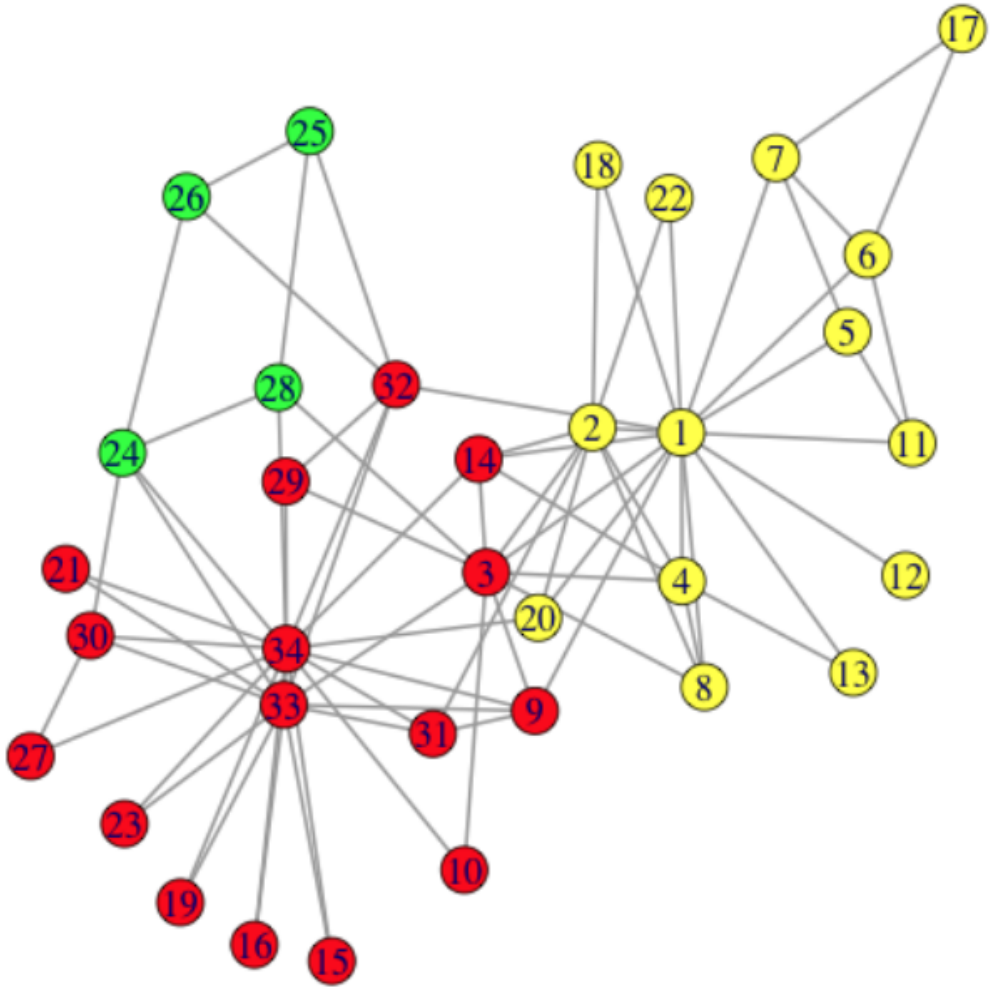
Edge betweenness: Zachary karate club

Zachary karate club

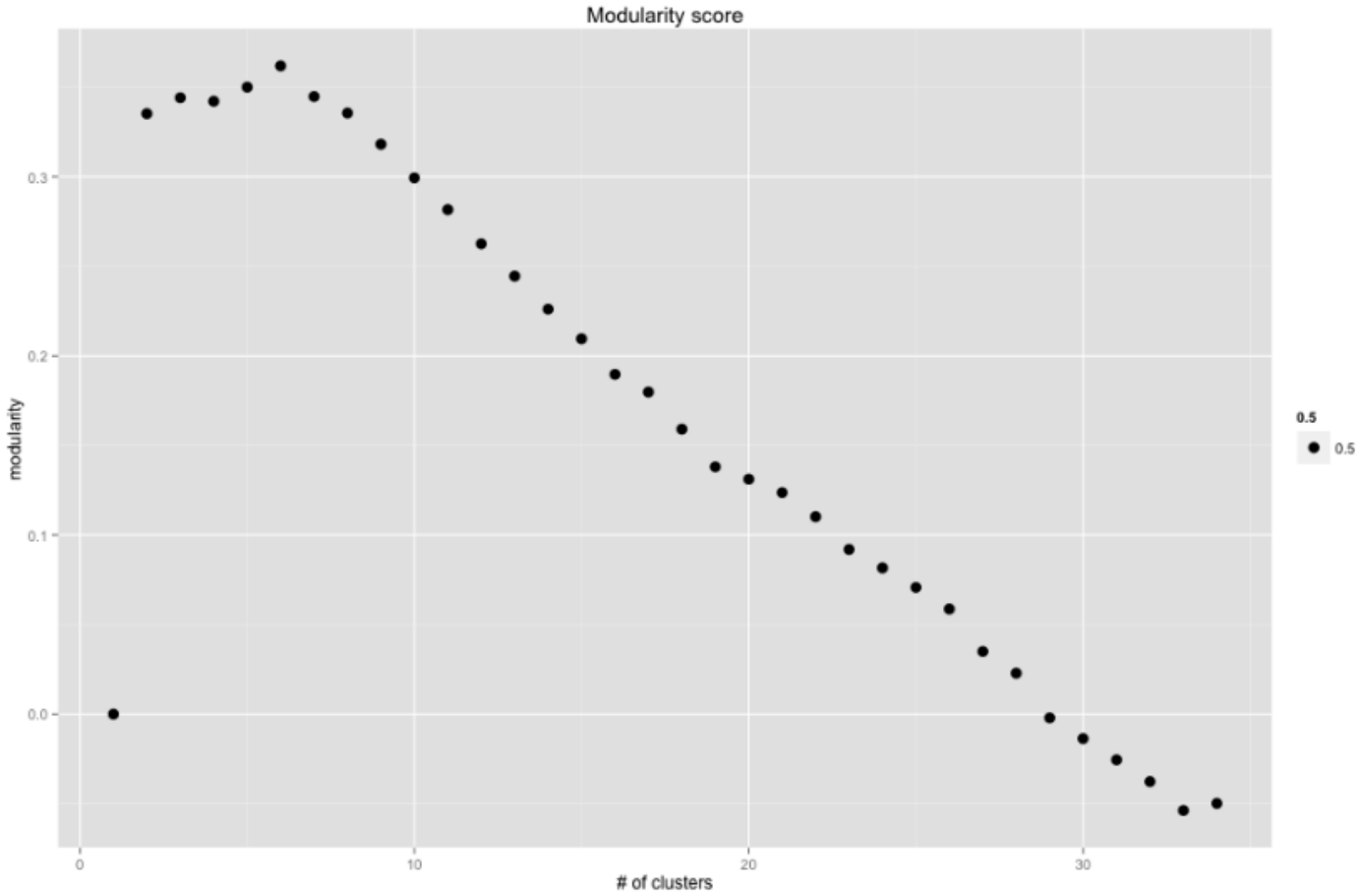


Edge betweenness: Zachary karate club

Zachary karate club

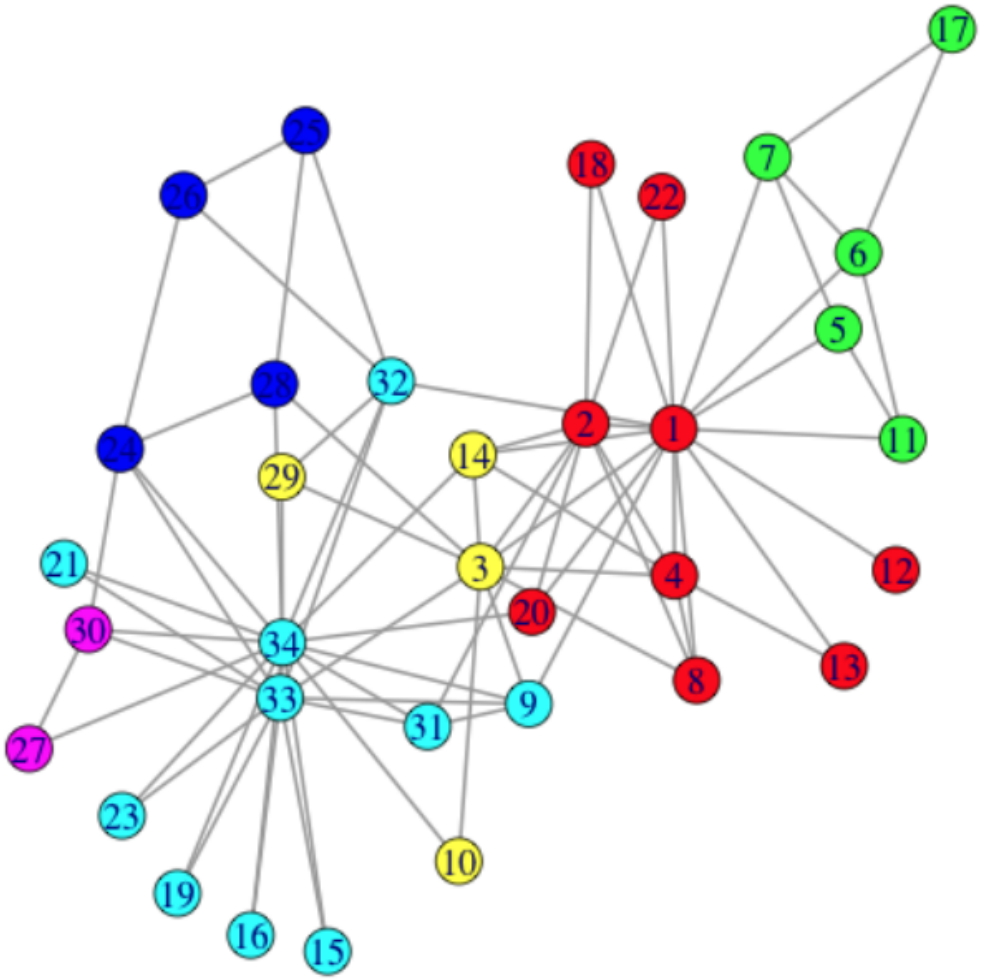


Edge betweenness: Zachary karate club

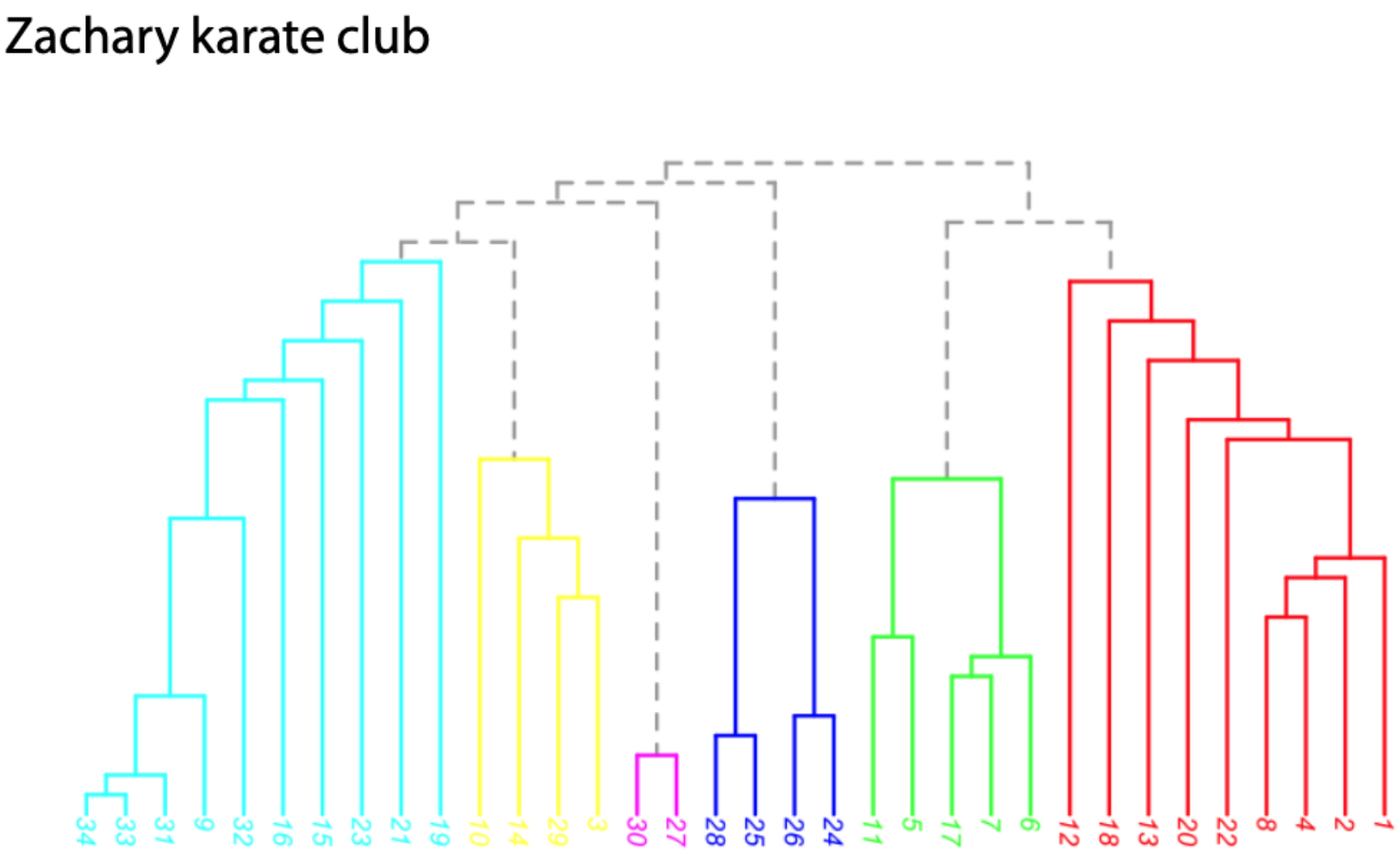


Edge betweenness: Zachary karate club

best: clusters = 6, modularity = 0.345

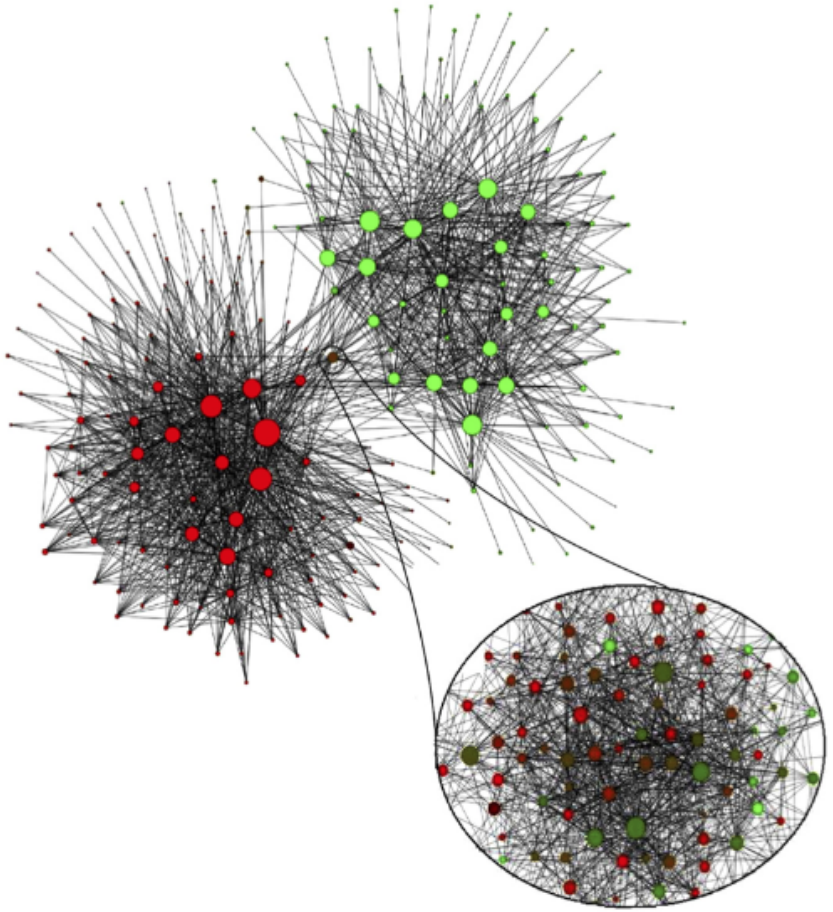


Edge betweenness: Zachary karate club



Louvain Algorithm: Fast community unfolding

Multi-resolution scalable method



2 mln mobile phone network

V. Blondel et.al., 2008

Louvain Algorithm: Fast community unfolding

"The Louvain method"

- Heuristic method for greedy modularity optimization
- Find partitions with high modularity
- Multi-level (multi-resolution) hierarchical scheme
- Scalable

Modularity:

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

V. Blondel et.al., 2008

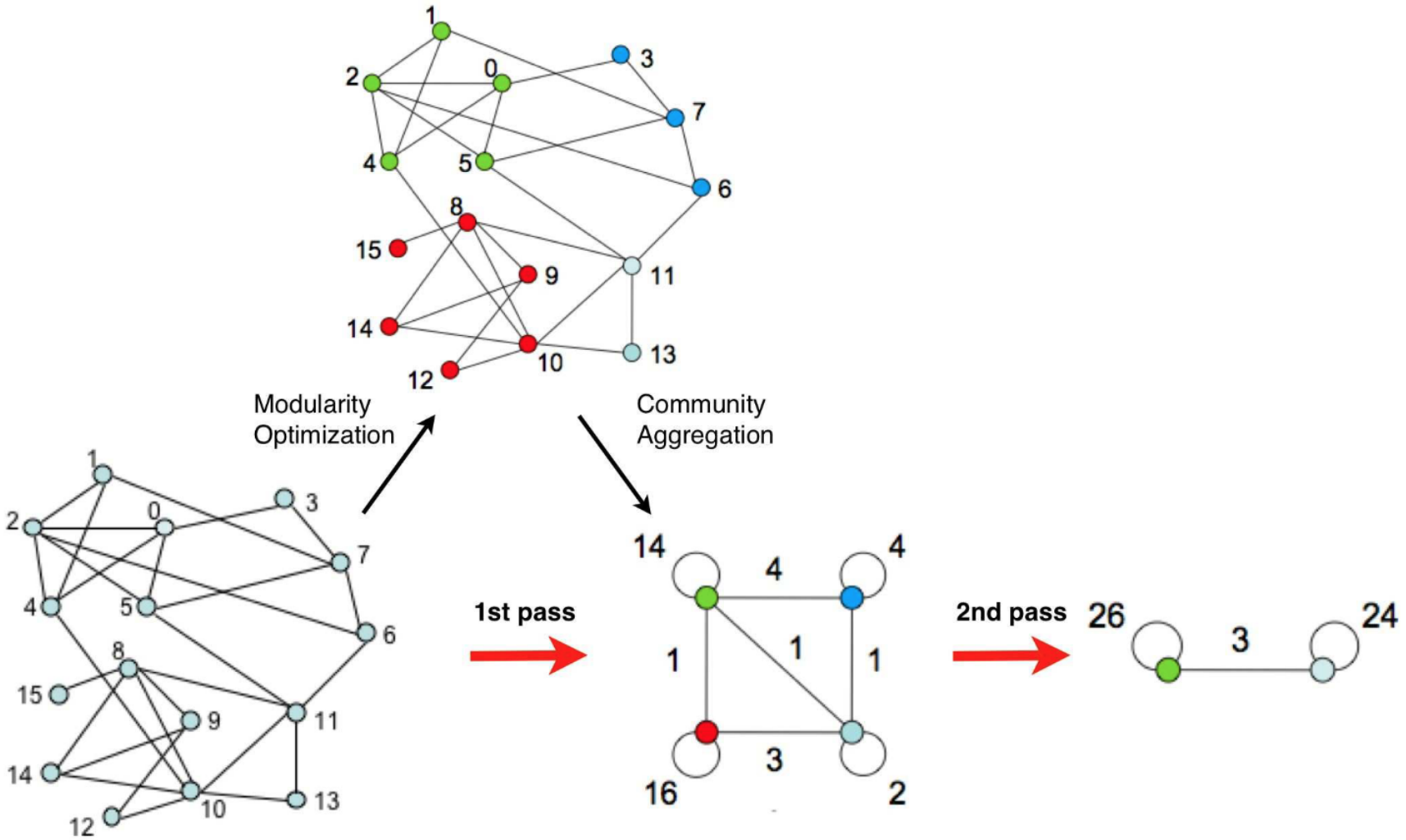
Louvain Algorithm: Fast community unfolding

Algorithm

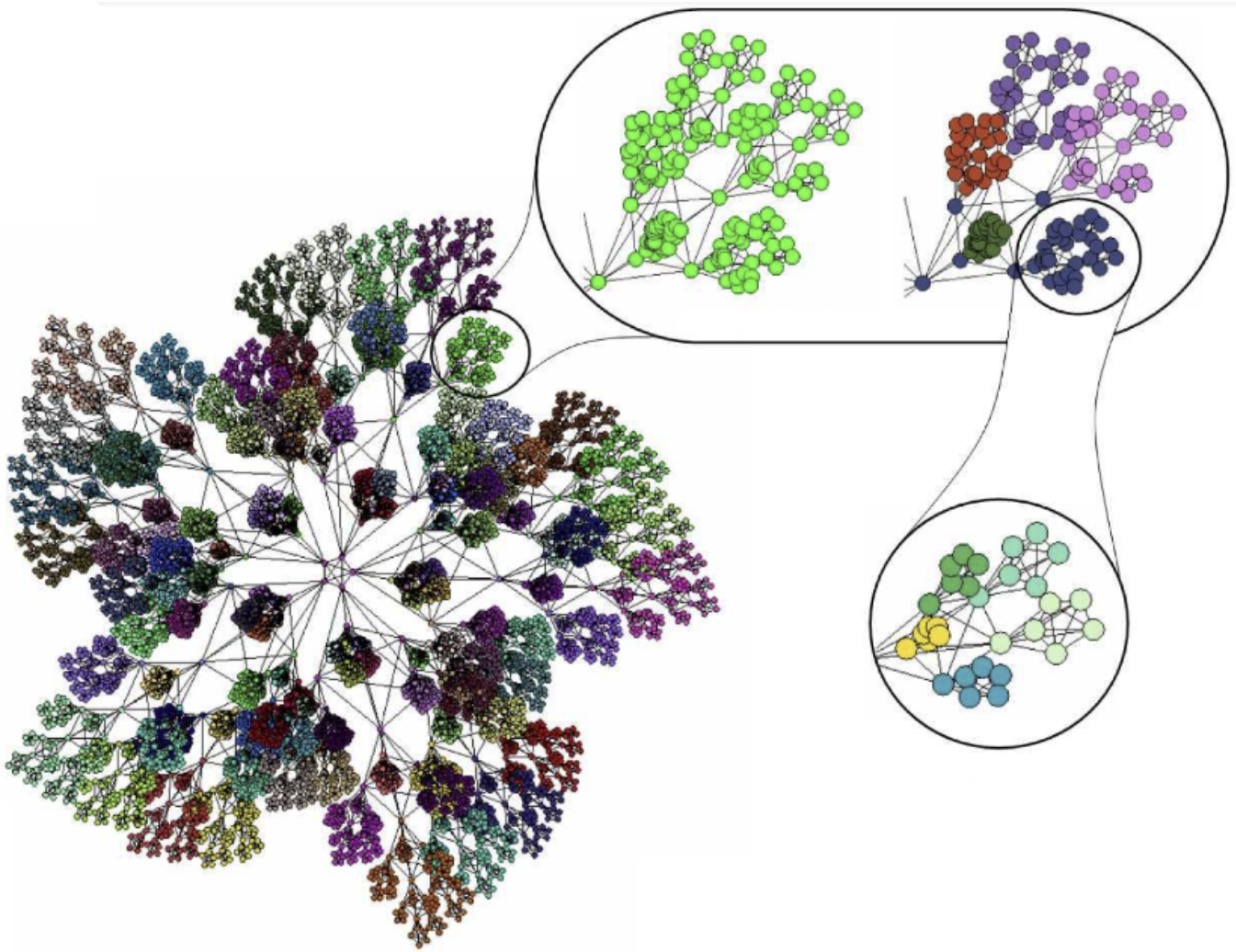
- Assign every node to its own community
- Phase I
  - For every node evaluate modularity gain from removing node from its community and placing it in the community of its neighbor
  - Place node in the community maximizing modularity gain
  - repeat until no more improvement (local max of modularity)
- Phase II
  - Nodes from communities merged into "super nodes"
  - Weight on the links added up
- Repeat until no more changes (max modularity)

V. Blondel et.al., 2008

Louvain Algorithm: Fast community unfolding



Louvain Algorithm: Fast community unfolding

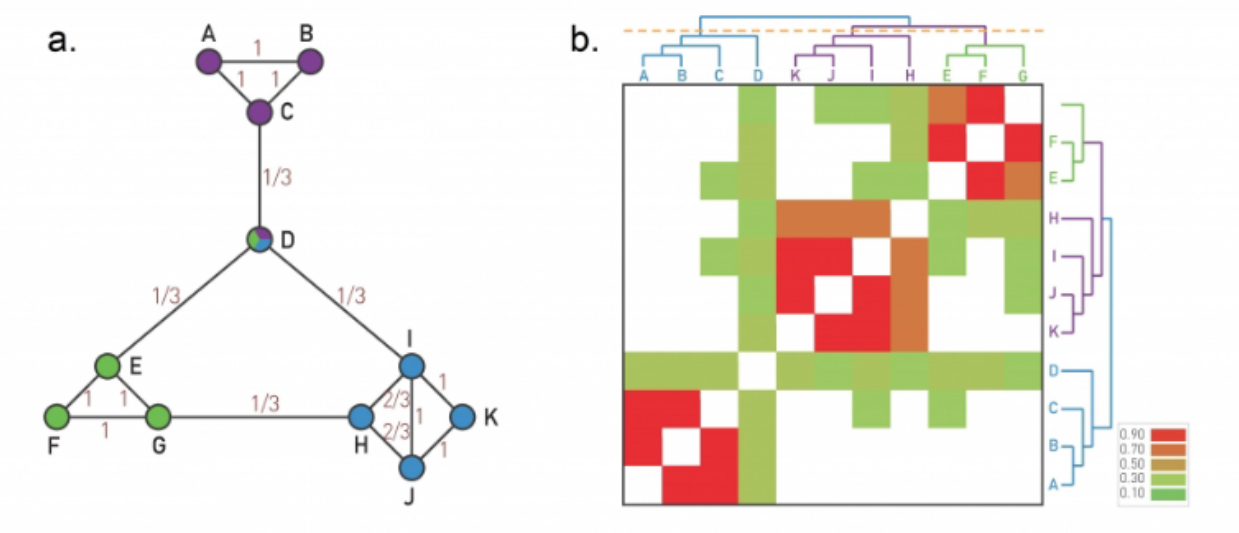


Agglomerative hierarchical clustering

Similarity matrix

$$x_{ij} = \frac{N(i,j)}{\min(k_i, k_j) + 1 - \theta(A_{i,j})}$$

- N_{ij} - number of common neighbors
- $\theta()$ - step function, 0 for $x \leq 0$ and 1 for $x > 0$



E. Ravasz et.al., 2002

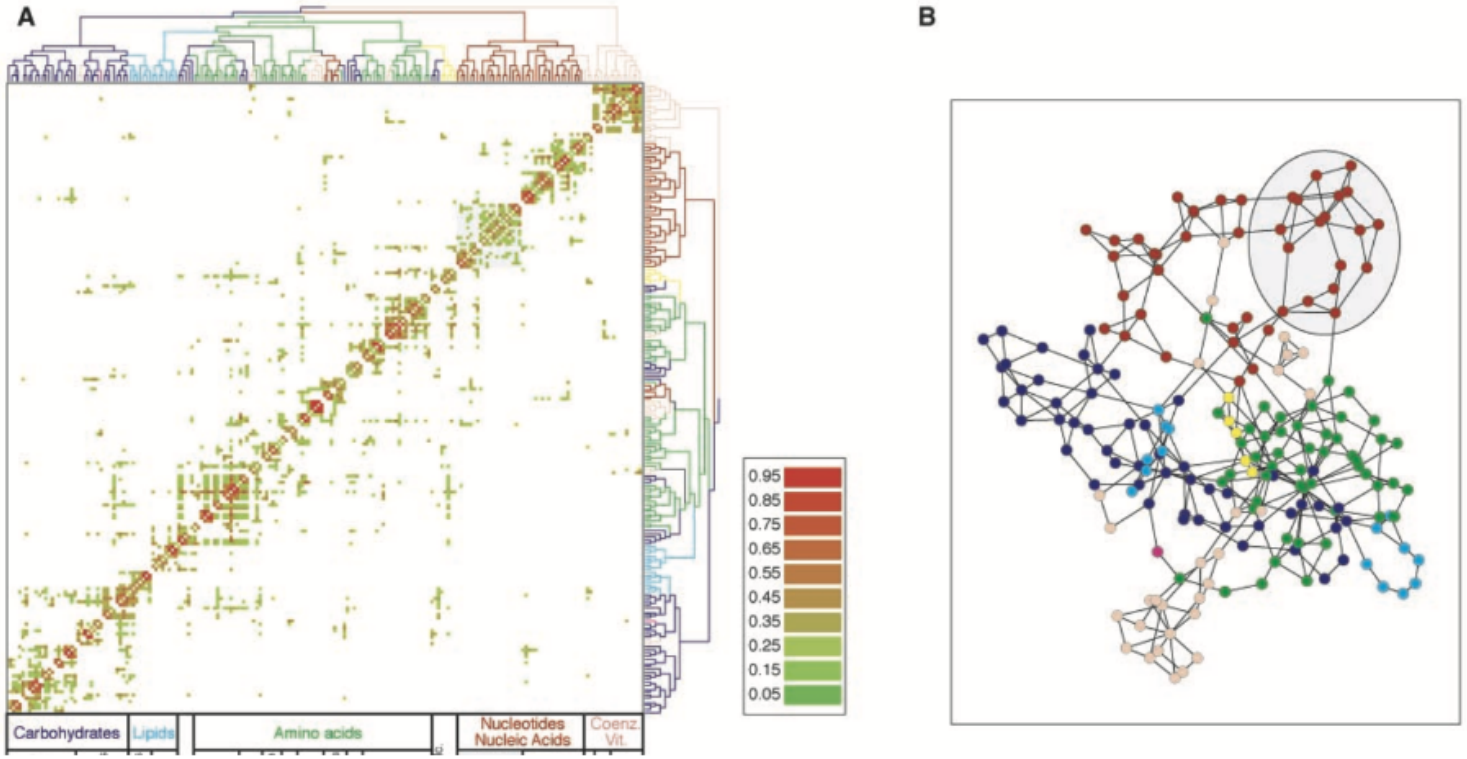
Agglomerative hierarchical clustering

The Ravasz algorithm:

- Assign each node to a community of its own and evaluate x_{ij} for all node pairs.
- Find the node (community) pair with the highest similarity and merge them into a single community.
- Calculate the similarity between the new community and all other communities.
- Repeat Steps 2 and 3 until all nodes form a single community.
- Find the optimal cut of the dendrogram

E. Ravasz et.al., 2002

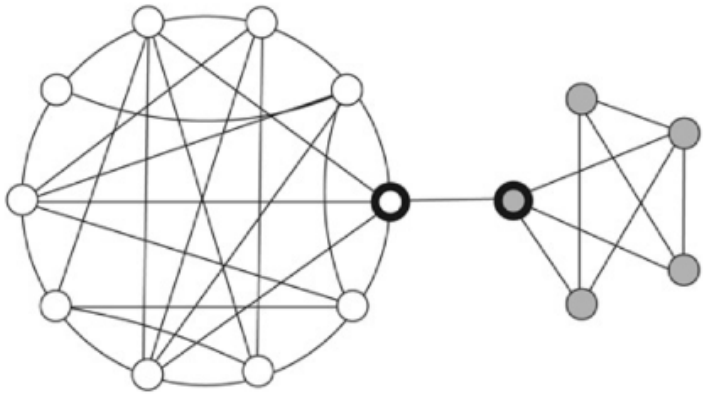
Agglomerative hierarchical clustering



Reduced E.coli metabolic network

E. Ravasz et.al., 2002

Communities and random walks



- Random walks on a graph tend to get trapped into densely connected parts corresponding to communities.

Walktrap community

Walktrap

- Consider random walk on graph
- At each time step walk moves to NN uniformly at random
  $P_{ij} = \frac{A_{ij}}{d(i)}, P = D^{-1}A, D_{ii} = diag(d(i))$
- P^t_{ij} - probability to get from i to j in t steps, $t \ll t_{mixing}$
- Assumptions: for two i and j in the same community P^t_{ij} is high
- if i and j are in the same community, then $\forall k, P^t_{ik} \approx P^t_{jk}$
- Distance between nodes:

$$r_{ij}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P^t_{ik} - P^t_{jk})^2}{d(k)}} = ||D^{-1/2}P^t_i - D^{-1/2}P^t_j||$$

P. Pons and M. Latapy, 2006

Walktrap

Computing node distance r_{ij}

- Direct (exact) computation: $P^t_{ij} = (P^t)_{ij}$ or $P^t_i = P^t p^0_i$, $p^0_i(k) = \delta_{ik}$
- Approximate computation (simulation):
  - Compute K random walks of length t starting form node i
  - Approximate $P^t_{ik} \approx \frac{N_{ik}}{K}$, number of walks end up on k

Distance between communities:

$$P^t_{Cj} = \frac{1}{|C|} \sum_{i \in C} P^t_{ij}$$

$$r_{C_1 C_2}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P^t_{C_1 k} - P^t_{C_2 k})^2}{d(k)}} = ||D^{-1/2} P^t_{C_1} - D^{-1/2} P^t_{C_2}||$$

P. Pons and M. Latapy, 2006

Walktrap

Algorithm (hierarchical clustering)

- Assign each vertex to its own community $S_1 = \{\{v\}, v \in V\}$
- Compute distance between all adjacent communities $r_{C_i C_j}$
- Choose two "closest" communities that minimizes (Ward's methods):

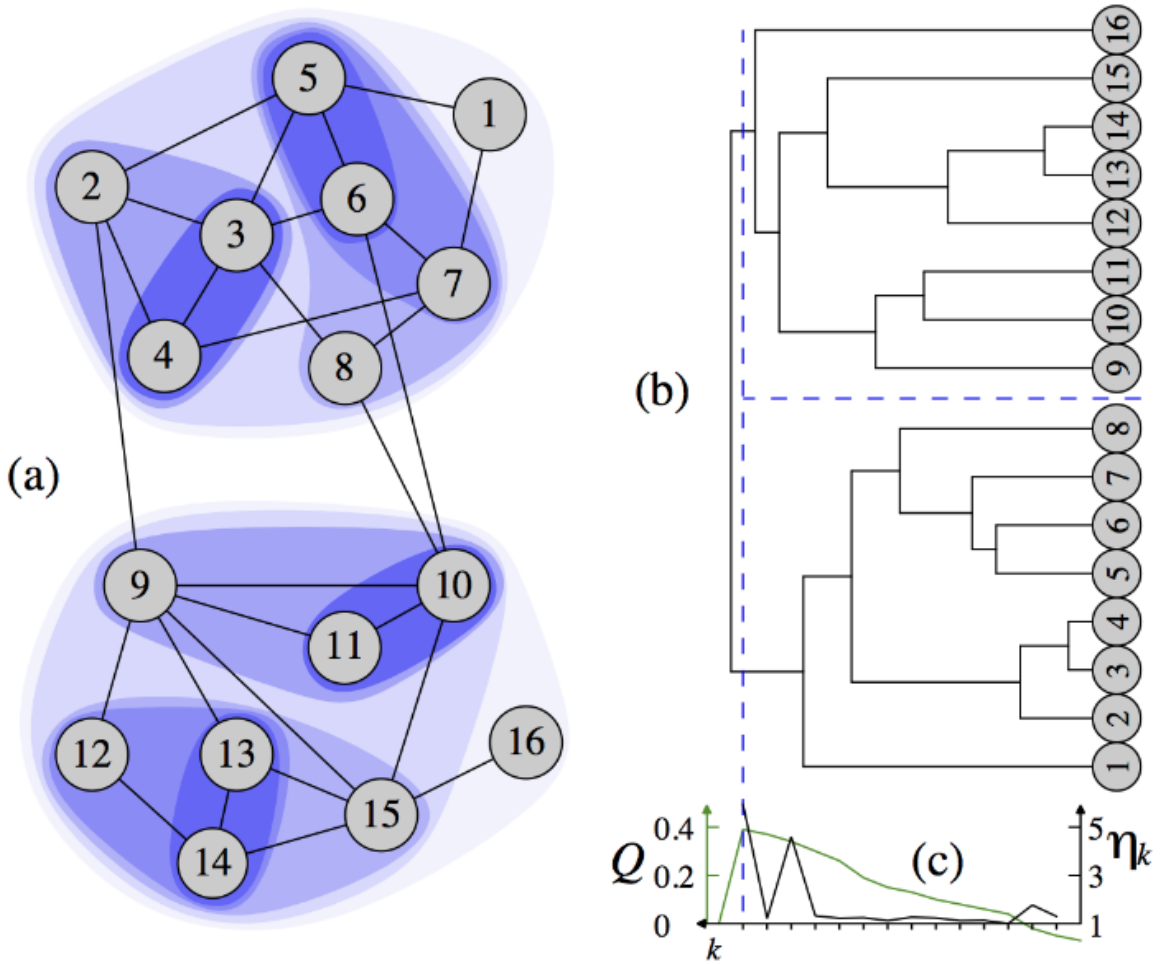
$$\Delta\sigma(C_1, C_2) = \frac{1}{n}\left(\sum_{i \in C_3} r_{iC_3}^2 - \sum_{i \in C_1} r_{iC_1}^2 - \sum_{i \in C_2} r_{iC_2}^2\right)$$

  and merge them $S_{k+1} = (S_k \backslash \{C_1, C_2\}) \cup C_3, C_3 = C_1 \cup C_2$
- update distance between communities

After $n - 1$ steps finish with one community $S_n = \{V\}$

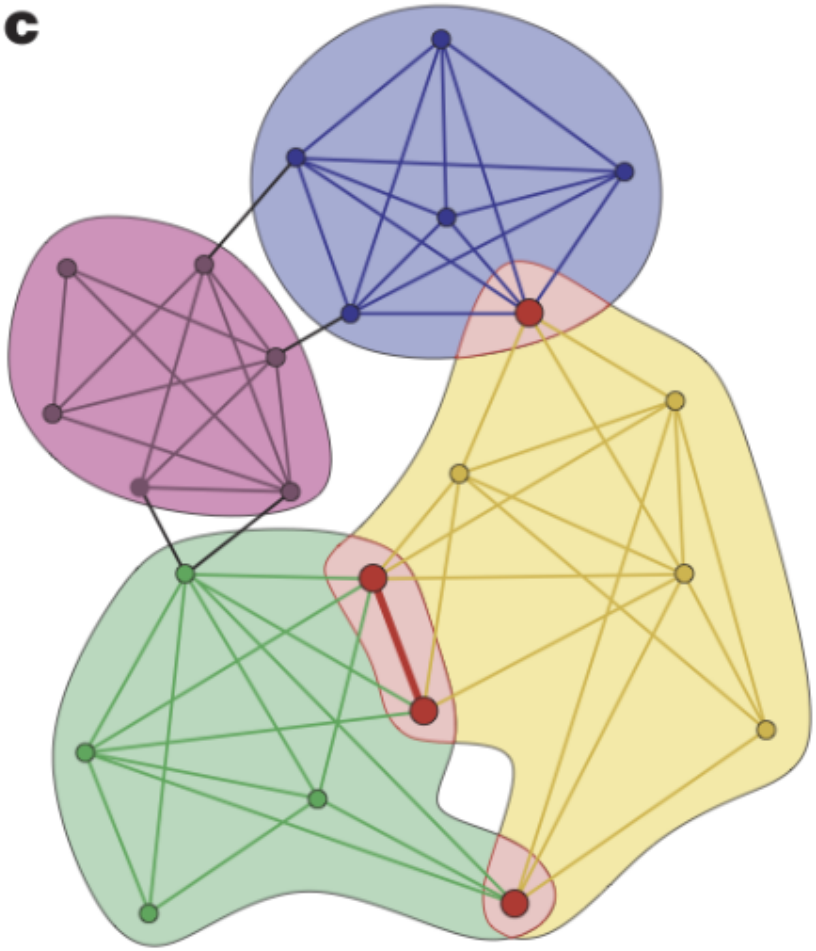
P. Pons and M. Latapy, 2006

Walktrap

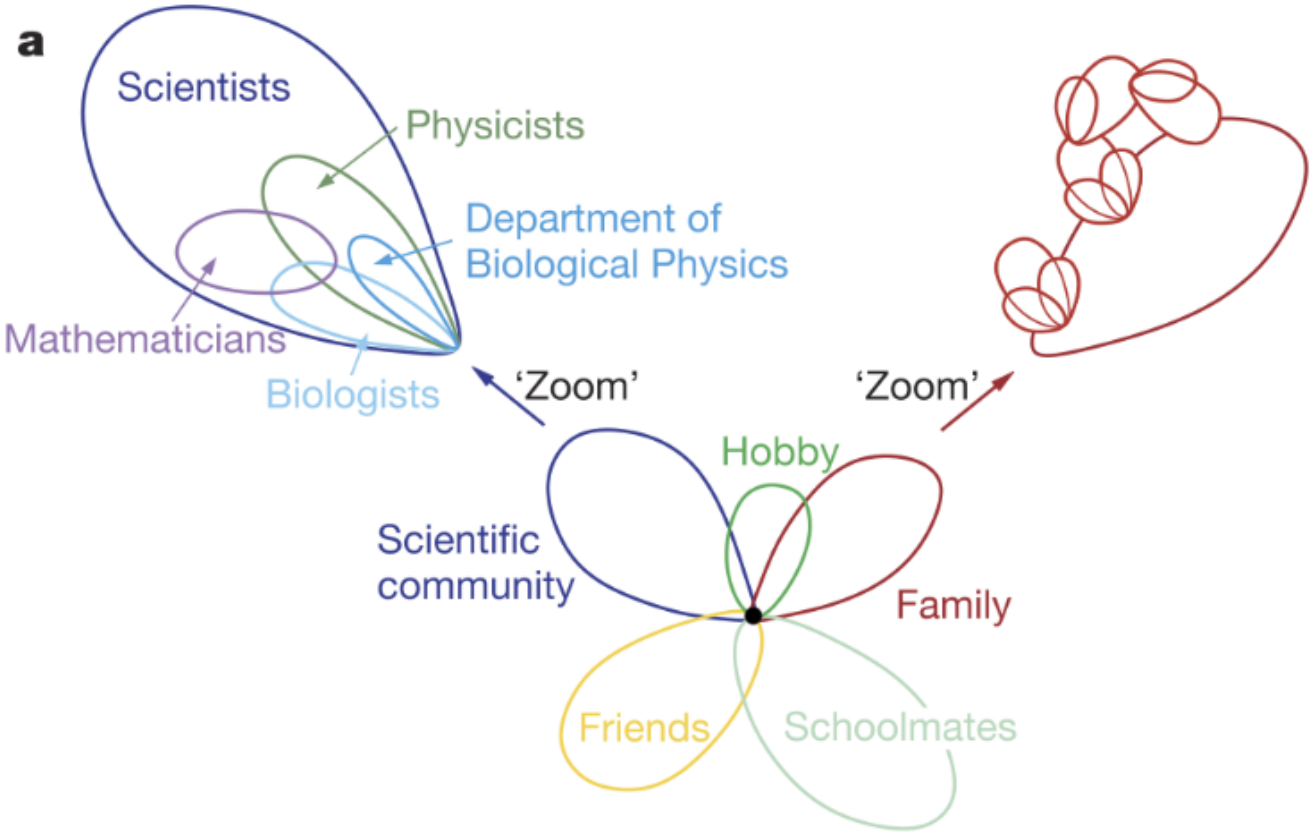


P. Pons and M. Latapy, 2006

Overlapping communities



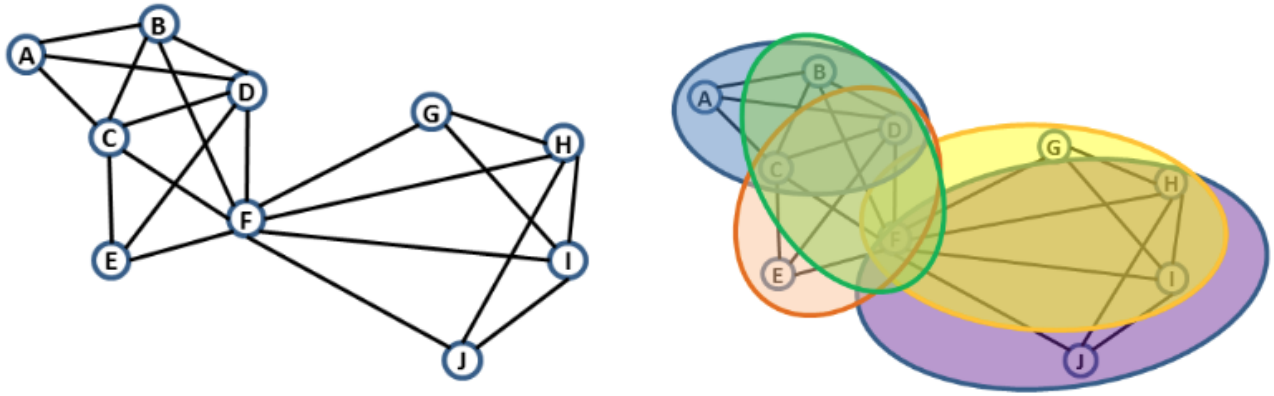
Overlapping communities



;

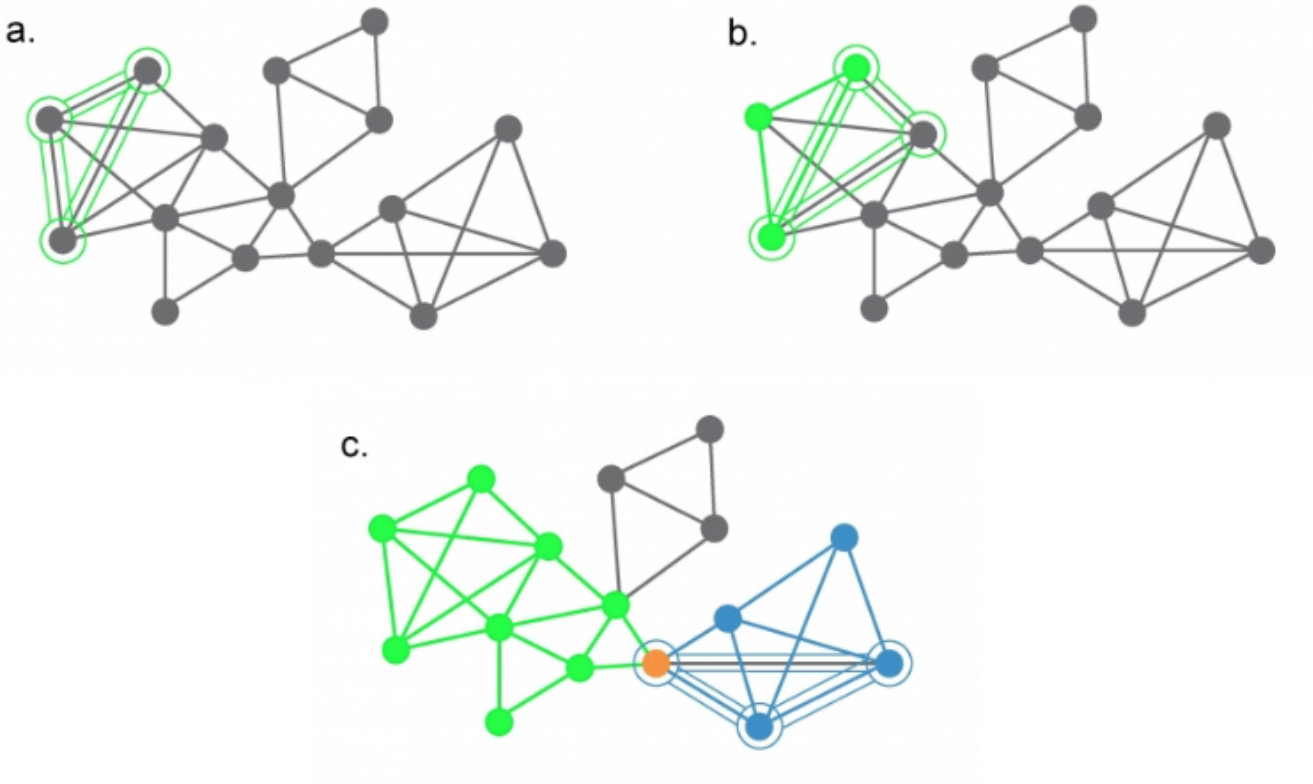
k-clique community

- *k*-clique is a clique (complete subgraph) with *k* nodes
- *k*-clique community a union of all *k*-cliques that can be reached from each other through a series of adjacent *k*-cliques
- two k-cliques are said to be adjacent if they share $k-1$ nodes.



Adjacent 4-cliques

k-clique community



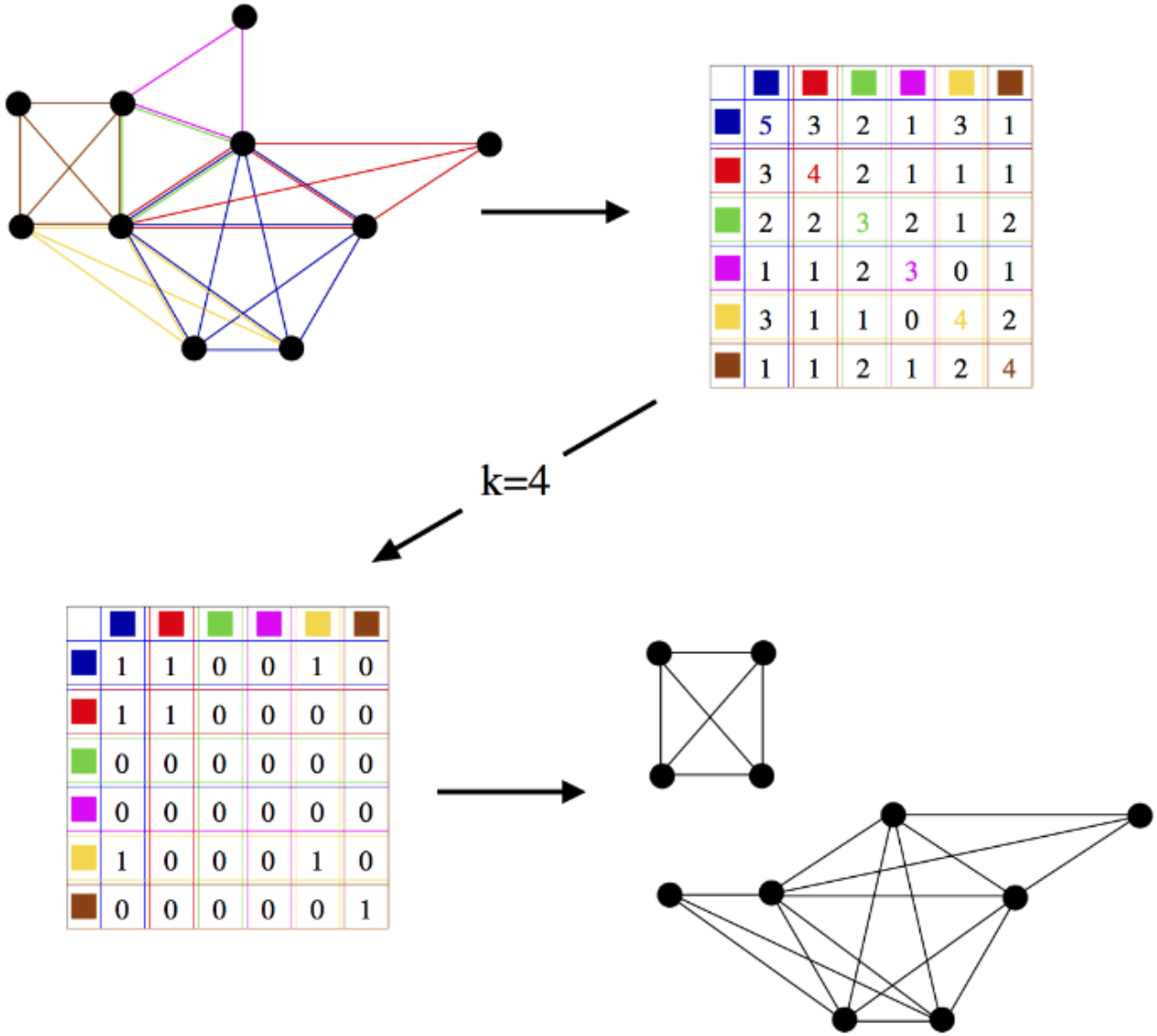
Palla, 2005

Algorithm CFinder

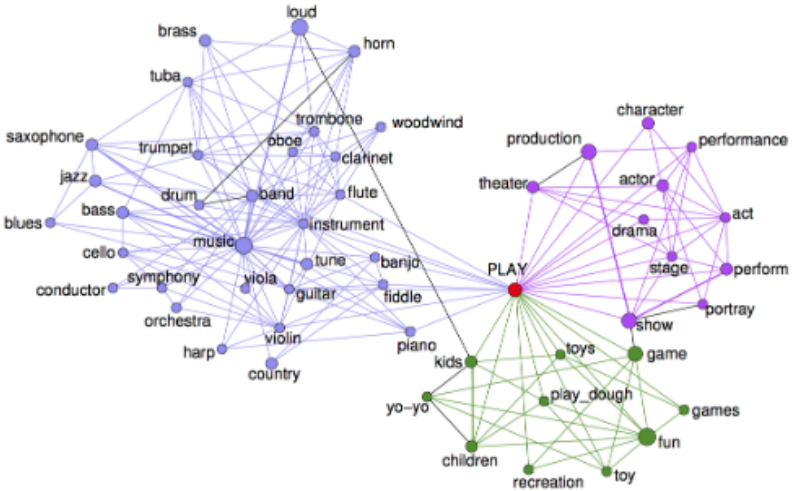
Algorithm CFinder

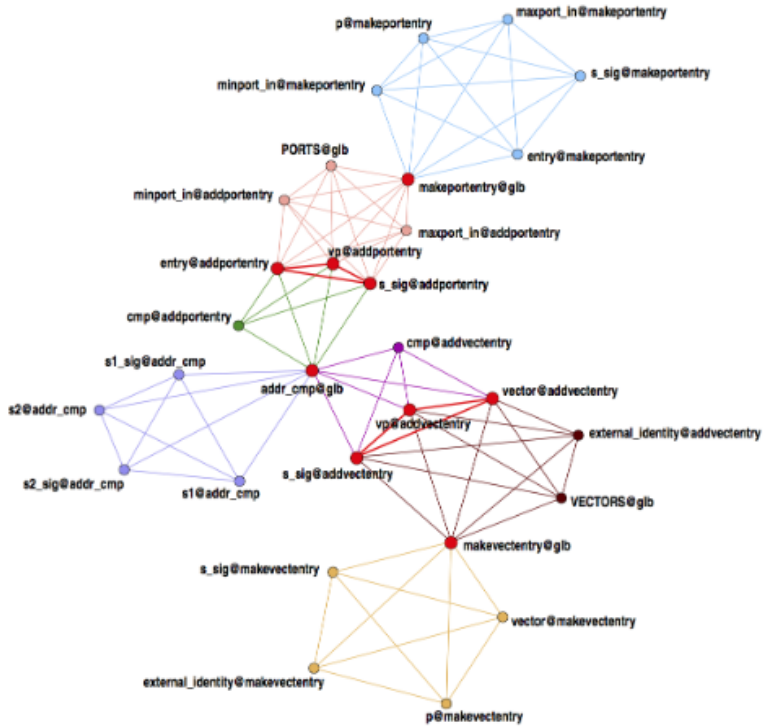
- Find all maximal cliques
- Create clique overlap matrix
- Threshold matrix at value k - 1
- Communities = connected components

k-clique percolation



k-clique percolation

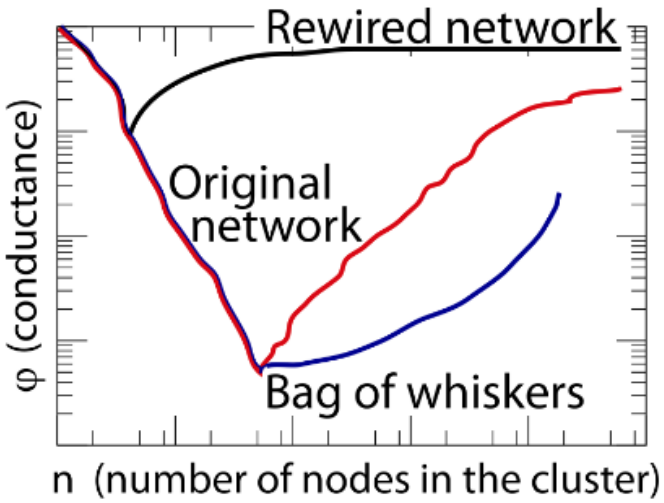




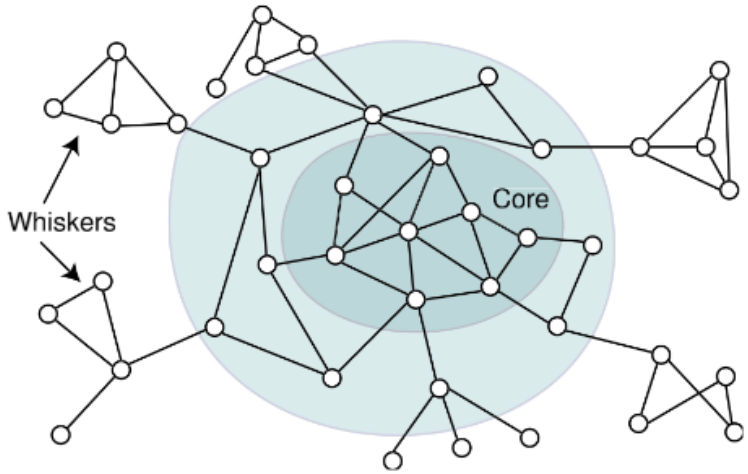
$k = 4$ $k = 5$

Palla, 2005

Real world communities



(a) Typical NCP plot



(b) Caricature of network structure

Best conductance of a vertex set S of size k:

$$\Phi(k) = \min_{S \in V, |S|=k} \phi(S), \quad \phi(S) = \frac{cut(S, V \setminus S)}{\min(vol(S), vol(S \setminus V))}$$

where $vol(S) = \sum_{i \in S} k_i$ - sum of all node degrees in the set

J. Leskovec, K. Lang, 2010

Real world communities

Author	Ref.	Label	Order
Eckmann & Moses	(Eckmann and Moses, 2002)	EM	$O(m\langle k^2\rangle)$
Zhou & Lipowsky	(Zhou and Lipowsky, 2004)	ZL	$O(n^3)$
Latapy & Pons	(Latapy and Pons, 2005)	LP	$O(n^3)$
Clauset et al.	(Clauset *et al.*, 2004)	NF	$O(n\log^2 n)$
Newman & Girvan	(Newman and Girvan, 2004)	NG	$O(nm^2)$
Girvan & Newman	(Girvan and Newman, 2002)	GN	$O(n^2m)$
Guimerà et al.	(Guimerà and Amaral, 2005; Guimerà *et al.*, 2004)	SA	parameter dependent
Duch & Arenas	(Duch and Arenas, 2005)	DA	$O(n^2\log n)$
Fortunato et al.	(Fortunato *et al.*, 2004)	FLM	$O(m^3n)$
Radicchi et al.	(Radicchi *et al.*, 2004)	RCCLP	$O(m^4/n^2)$
Donetti & Muñoz	(Donetti and Muñoz, 2004, 2005)	DM/DMN	$O(n^3)$
Bagrow & Bollt	(Bagrow and Bollt, 2005)	BB	$O(n^3)$
Capocci et al.	(Capocci *et al.*, 2005)	CSCC	$O(n^2)$
Wu & Huberman	(Wu and Huberman, 2004)	WH	$O(n+m)$
Palla et al.	(Palla *et al.*, 2005)	PK	$O(\exp(n))$
Reichardt & Bornholdt	(Reichardt and Bornholdt, 2004)	RB	parameter dependent

Author	Ref.	Label	Order
Girvan & Newman	(Girvan and Newman, 2002; Newman and Girvan, 2004)	GN	$O(nm^2)$
Clauset et al.	(Clauset *et al.*, 2004)	Clauset et al.	$O(n\log^2 n)$
Blondel et al.	(Blondel *et al.*, 2008)	Blondel et al.	$O(m)$
Guimerà et al.	(Guimerà and Amaral, 2005; Guimerà *et al.*, 2004)	Sim. Ann.	parameter dependent
Radicchi et al.	(Radicchi *et al.*, 2004)	Radicchi et al.	$O(m^4/n^2)$
Palla et al.	(Palla *et al.*, 2005)	Cfinder	$O(\exp(n))$
Van Dongen	(Dongen, 2000a)	MCL	$O(nk^2)$, $k < n$ parameter
Rosvall & Bergstrom	(Rosvall and Bergstrom, 2007)	Infomod	parameter dependent
Rosvall & Bergstrom	(Rosvall and Bergstrom, 2008)	Infomap	$O(m)$
Donetti & Muñoz	(Donetti and Muñoz, 2004, 2005)	DM	$O(n^3)$
Newman & Leicht	(Newman and Leicht, 2007)	EM	parameter dependent
Ronhovde & Nussinov	(Ronhovde and Nussinov, 2009)	RN	$O(m^\beta\log n)$, $\beta\sim 1.3$

Practical Applications of Community Detection

- Criminology
- Public Health
- Politics
- Customer Segmentation, Smart Advertising and Targeted Marketing
- Recommendation Systems
- Social Network Analysis
- Network Summarization and Privacy
- Community Evolution Prediction
- and etc

Application Areas paper

Citing References

- M. Fiedler. Algebraic connectivity of graphs, Czech. Math. J, 23, pp 298-305, 1973
- A. Pothen, H. Simon and K. Liou. Partitioning sparse matrices with eigenvectors of graphs, SIAM Journal of Matrix Analysis, 11, pp 430-452, 1990
- Bruce Hendrickson and Robert Leland. A Multilevel Algorithm for Partitioning Graphs, Sandia National Laboratories, 1995
- Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, N 8, pp 888-905, 2000
- Karger, David (1993). ”Global Min-cuts in RNC and Other Ramifications of a Simple Mincut Algorithm”. Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms.

Citing References

- M.A Porter, J-P Onella, P.J. Mucha. Communities in Networks, Notices of the American Mathematical Society, Vol. 56, No. 9, 2009
- 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):27–64, 2007.
- S. Fortunato. Community detection in graphs, Physics Reports, Vol. 486, Iss. 3–5, pp 75-174, 2010

Citing References

- G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814?818.
- P. Pons and M. Latapy, Computing communities in large networks using random walks, Journal of Graph Algorithms and Applications, 10 (2006), 191-218.
- V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. P10008 (2008).
- J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Statistical properties of community structure in large social and information networks. In WWW 08: Procs. of the 17th Int. Conf. on World Wide Web, pages 695-704, 2008.