Community detection

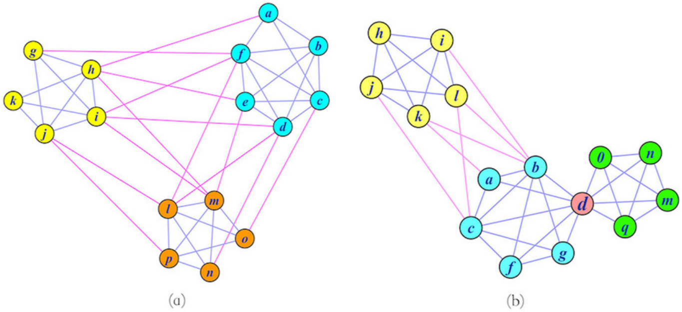
Leonid E. Zhukov

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

Structural Analysis and Visualization of Networks

Lecture outline

1. Overlapping commmunities
   - Clique percolation method
2. Heuristic methods
   - Label propagation
   - Fast community unfolding
3. Random walk methods
   - Walktrap
   - Nibble

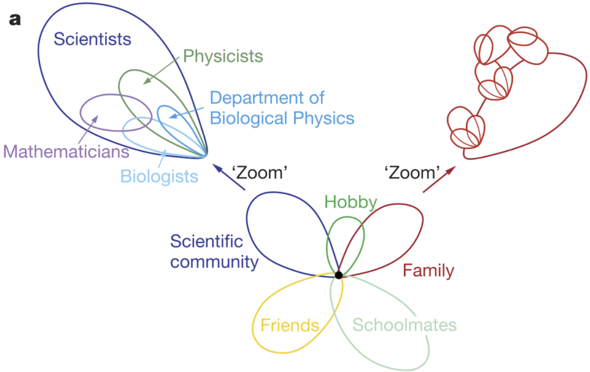


Community detection:

- Vertex clustering (vertex similarity)
- Graph partitioning (sparse cuts)

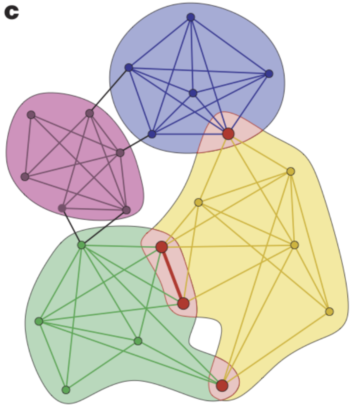
image from W. Liu , 2014

Overlapping communities



Palla, 2005

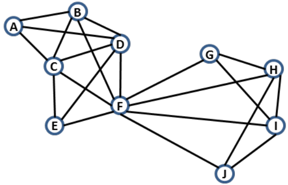
Overlapping communities

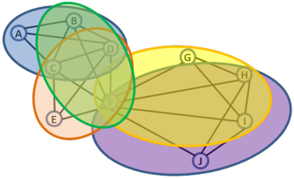


Palla, 2005

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.





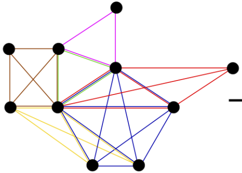
Adjacent 4-cliques

k-clique percolation

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

Palla, 2005

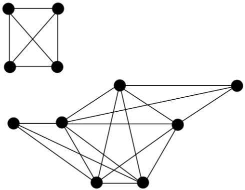
k-clique percolation



	5	3	2	1	3	1
	3	4	2	1	1	1
	2	2	3	2	1	2
	1	1	2	3	0	1
	3	1	1	0	4	2
	1	1	2	1	2	4

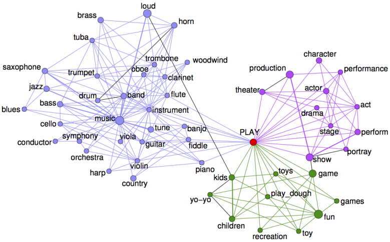
k=4

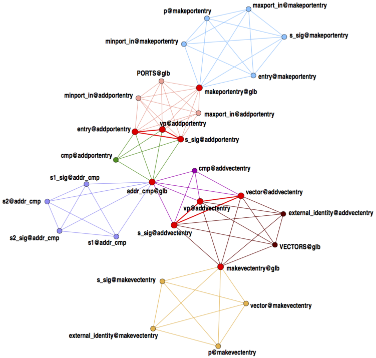
	1	1	0	0	1	0
	1	1	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	1	0
	0	0	0	0	0	1



Palla, 2005

k-clique percolation

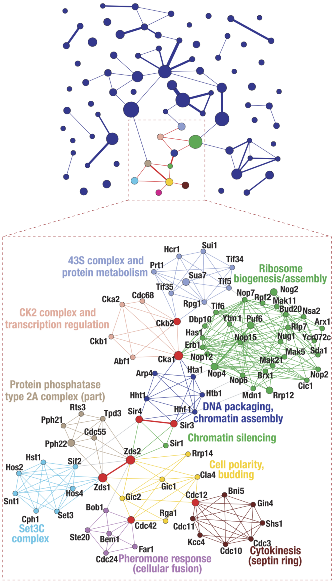




$k = 4$ $\qquad$ $k = 5$

Palla, 2005

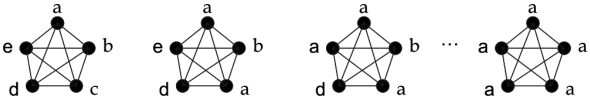
k-clique percolation



Label propagation

Algorithm:

- Initialize labels on all nodes
- Randomized node order
- For every node replace its label with occurring with the highest frequency among neighbors (ties are broken uniformly randomly).
- If every node has a label that the maximum number of their neighbors have, then stop the algorithm



Raghavan, 2007

Label propagation

image from Lab41 blog

Fast community unfolding

"The Louvain method"

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

V. Blondel et.al., 2008

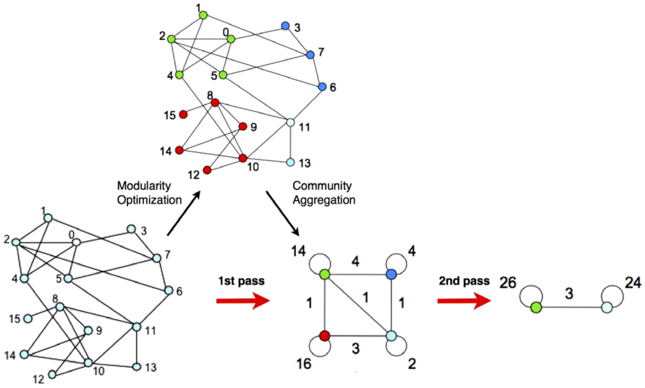
Fast community unfolding

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)

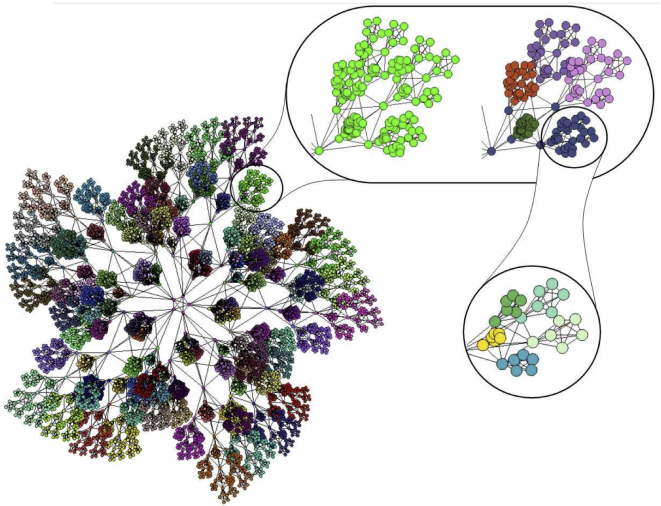
V. Blondel et.al., 2008

Fast community unfolding



V. Blondel et.al., 2008

Fast community unfolding



V. Blondel et.al., 2008

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: $P^t_{ij} = (P^t)_{ij}$
- 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

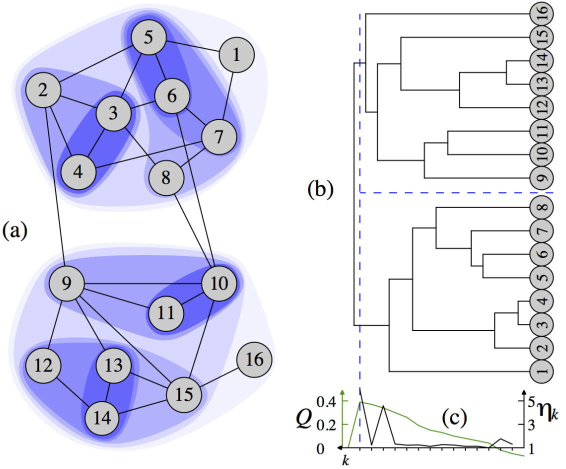
Algorithm (hierachical clustering)

- Assign each vertex to its own community
- Compute distance between adjacent vertices
- Choose two "closest"communities and merge them
- update distance between communities

After $n-1$ steps finish with one community

P. Pons and M. Latapy, 2006

Walktrap



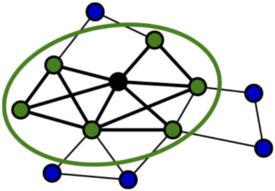
P. Pons and M. Latapy, 2006

Local clustering algorithm

- Conductance of a vertex set S

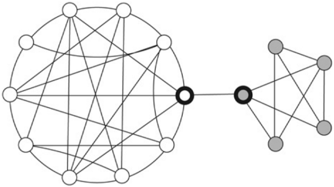
$$\phi(S) = \frac{cut(S, V \backslash S)}{\min(vol(S), vol(S \backslash V))}$$

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



- Example: $cut(S) = 7$, $vol(S) = 33$, $vol(V \backslash S) = 11$, $\phi(S) = 7/11$

Local clustering algorithm



- The probability that one-step random walk starting in the cluster will leave the cluster = conductance of the set
(it is a probability of picking up an edge from the smaller set that crosses the cut.)

Local clustering algorithm

- Given a vertex find a small cluster around the vertex in time proportional to the size of the cluster
- Short random walks t - steps
- "Lazy" random walk operator:

$$M = (AD^{-1} + I)/2, \quad D = diag(d(i))$$

- Distribution of random walk:

$$p(t) = M^t p(0)$$

D. Spielman et.al, 2008

Local clustering algorihtm

Spielman, 2003/2008

Algorithm: Nibble

Input: Graph $G, q_0(v_0), \phi_0$

Output: Graph partition S

for $t = 1 : t_m$ **do**

  $q_t = Mr_{t-1}$;

  $r_t(i) = q_t(i) \;\; \textit{if} \;\; q_t(i)/d(i) > \epsilon, \textit{else } 0$;

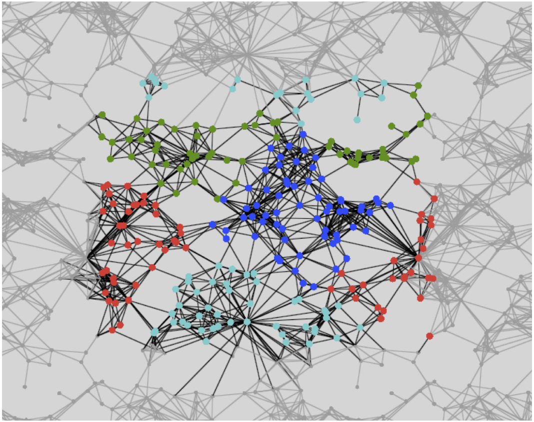
order i from large to small based on $q_{t_m}(i)/d(i)$;

Compute conductance, sweep $\phi(S\{i = 1..j\})$;

If there is $j : \phi(S_j) < \phi_0$, return S

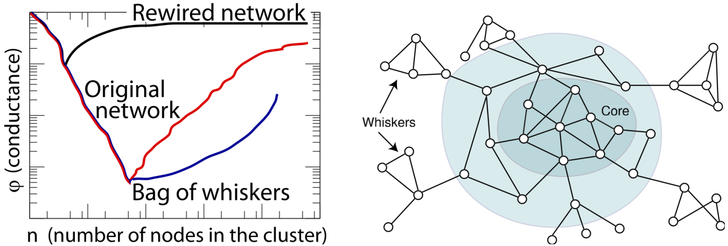
D. Spielman et al, 2008

Multiple clusters



D. Gleich, 2013

Real world communities



(a) Typical NCP plot

(b) Caricature of network structure

J. Leskovec, K. Lang, 2010

Community detection algorithms

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	$O(n^2 m)$
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	$O(m^3 n)$
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	$O(m^\beta \log n)$, $\beta \sim 1.3$

Fortunato, 2010

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.
- U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
- 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).
- Daniel A. Spielman, Shang-Hua Teng. A Local Clustering Algorithm for Massive Graphs and Its Application to Nearly Linear Time Graph Partitioning. SIAM J. Comput. 42(1): 1-26 (2013)

References

- 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.
- M.A Porter, J-P Onella, P.J. Mucha. Communities in Networks, Notices of the American Mathematical Society, Vol. 56, No. 9, 2009
- S. E. Schaeffer. Graph clustering. Computer Science Review, 1(1), pp 27-64, 2007.
- S. Fortunato. Community detection in graphs, Physics Reports, Vol. 486, Iss. 3-5, pp 75-174, 2010

Summary

Lectures 1-10

- Network characteristics:
  - Power law node degree distribution
  - Small diameter
  - High clustering coefficient (transitivity)
- Network models:
  - Random graphs
  - Preferential attachement
  - Small world
- Centrality measures:
  - Degree centrality
  - Closeness centrality
  - Betweenness centrality
- Link analysis:
  - Page rank
  - HITS

Summary

Lectures 1-10

- Structural equivalence
  - Vertex equivalence
  - Vertex similarity
- Assortative mixing
  - Assortative and disassortative networks
  - Mixing by node degree
  - Modularity
- Network structures:
  - Cliques
  - k-cores
- Network communities:
  - Similarity (vertex) clustering
  - Graph partitioning
  - Overlapping communities
  - Heuristic and random walk methods