Graph partitioning algorithms

I. Makarov & L.E. Zhukov

Moscow Institute of Physics and Technology

Network Science

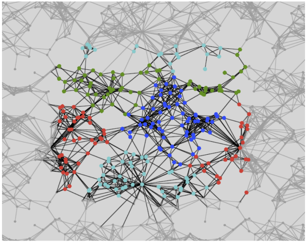
Lecture outline

1. Graph paritioning
   - Metrics
   - Algorithms
2. Spectral optimization
   - Min cut
   - Normalized cut
   - Modularity maximization
3. Multilevel spectral
4. Overlapping commmunities
   - Clique percolation method
5. Multi-level optimization
   - Fast community unfolding
6. Random walk methods
   - Walktrap

Network communities

Definition

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



- Graph partitioning problem

Graph partitioning

Combinatorial problem:

- Number of ways to divide network of n nodes in 2 groups (bi-partition):

$$\frac{n!}{n_1!n_2!}, \qquad n = n_1 + n_2$$

- Dividing into k non-empty groups (Stirling numbers of the second kind)

$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{n}(-1)^j C_k^j (k-j)^n$$

- Number of all possible partitions (n-th Bell number):

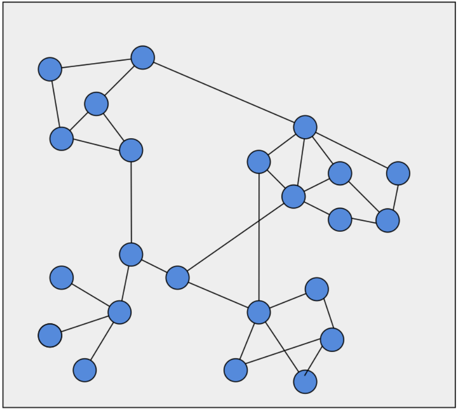
$$B_n = \sum_{k=1}^{n} S(n,k)$$

$B_{20} = 5,832,742,205,057$

Community detection

- Consider only sparse graphs $m \ll n^2$
- Each community should be connected
- Combinatorial optimization problem:
  - optimization criterion
  - optimization method
- Exact solution NP-hard
  (bi-partition: $n = n_1 + n_2$, $n!/(n_1!n_2!)$ combinations)
- Solved by greedy, approximate algorithms or heuristics
- Recursive top-down 2-way partition, multiway partiton
- Balanced class partition vs communities

Graph cut



Optimization criterion: graph cut

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):

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

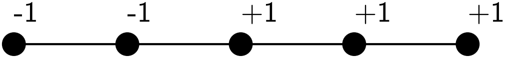
where: $Vol(V_1) = \sum_{i \in V_1, j \in V} e_{ij} = \sum_{i \in V_1} k_i$

Optimization methods

- 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]
  – Multicommodity flow [Leighton and Rao, 1988]
- Heuristics algorithms:
  – Multilevel graph partitioning (METIS) [G. Karypis, Kumar 1998]
- Randomized algorithms:
  –Randomized min cut [D. Karger, 1993]

Graph cuts

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



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

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

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

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

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

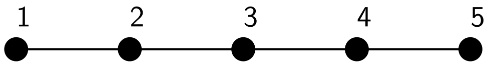
Graph cuts

- Graph Laplacian: $L_{ij} = D_{ij} - A_{ij}$, where $D_{ii} = diag(k_i)$

$$L_{ij} = \begin{cases} k(i)\,, & if\;\; i = j \\ -1\,, & if\; \exists\; e(i,j) \\ 0\,, & otherwise \end{cases}$$

- Laplacian matrix 5x5:

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



Graph cuts

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

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

- Minimal cut:

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

- 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

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

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

- Relaxation - continuous problem: find

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

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):

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

- Eigenvalue problem:

$$\mathrm{L}\mathrm{x} = \lambda\mathrm{x}, \quad \mathrm{x} \perp \mathrm{e}$$

- Solution:

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

- First (smallest) eigenvector:

$$\mathrm{L}\mathrm{e} = 0, \quad \lambda = 0, \quad \mathrm{x}_1 = \mathrm{e}$$

- Looking for the second smallest eigenvalue/eigenvector λ_2 and x_2
- Minimization of Rayleigh-Ritz quotient:

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

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 A

Output: class indicator vector s

compute $D = diag(deg(A))$;

compute $L = D - A$;

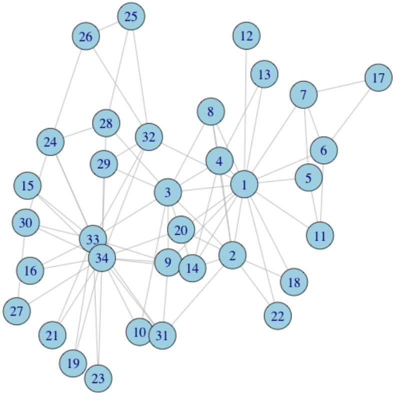
solve for second smallest eigenvector:

min cut: $Lx = \lambda x$;

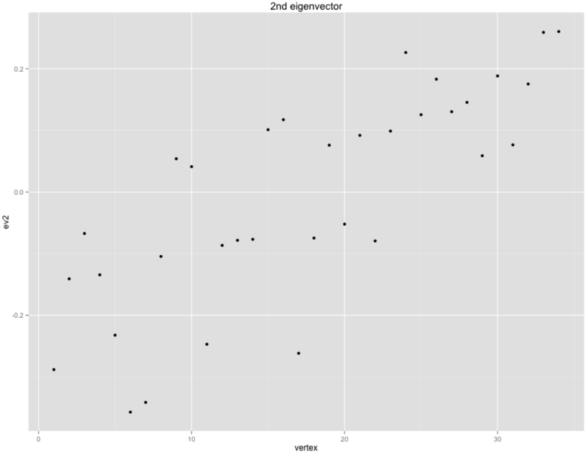
normalized cut : $Lx = \lambda Dx$;

set $s = sign(x_2)$

Example

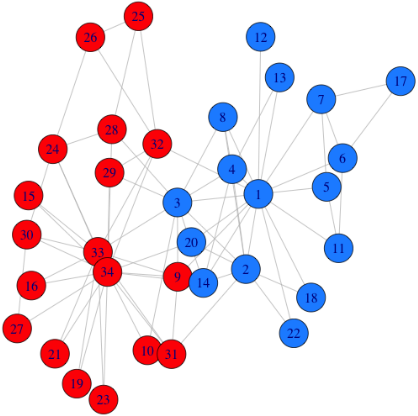

25
26
12
13
17
7
8
28
32
6
24
29
4
5
1
15
3
30
20
11
33
2
34
9
14
16
18
27
22
21
10
31
19
23

Example



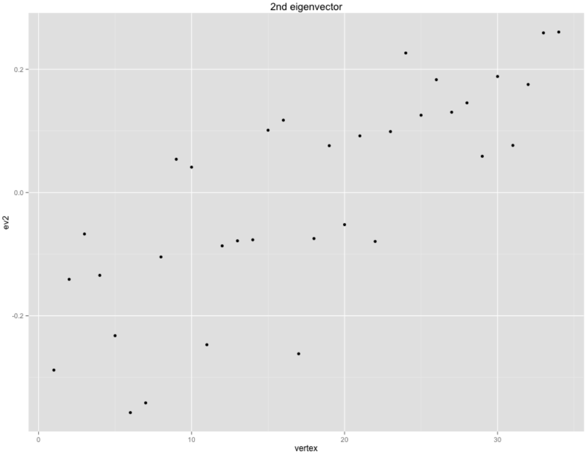
Eigenvalues: $\lambda_1 = 0$, $\lambda_2 = 0.2$, $\lambda_3 = 0.25$ ...

Example

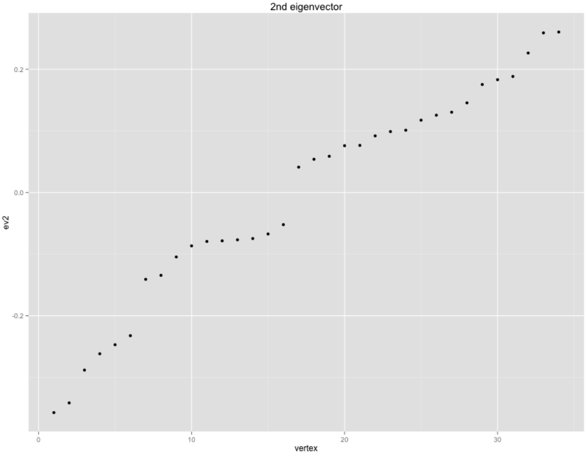


25
26
12
13
17
7
8
28
32
24
6
4
29
5
1
15
3
30
20
11
33
2
34
9
14
16
18
27
22
21
10
31
19
23

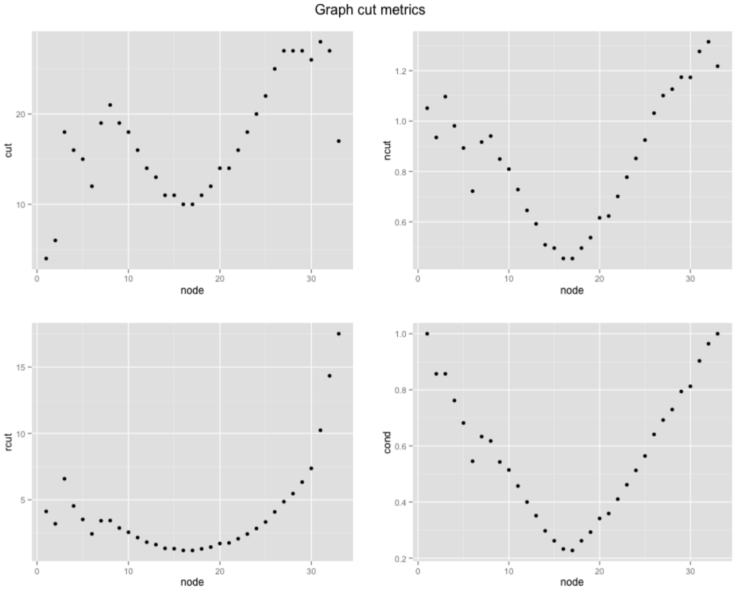
Example



Spectral ordering



Cut metrics



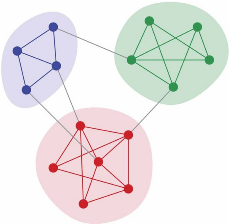
Optimization criterion: modularity

- 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

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

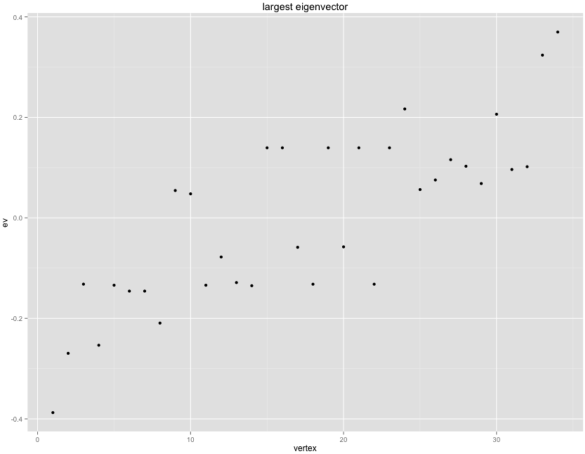
compute $\mathrm{k} = deg(\mathrm{A})$;

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

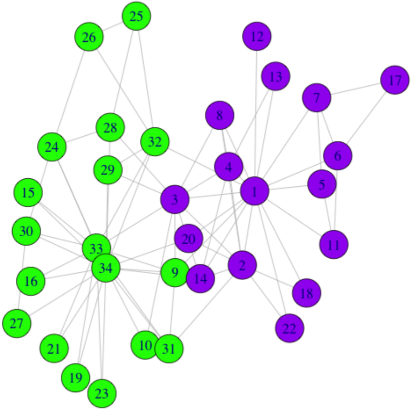
solve for maximal eigenvector $\mathrm{Bx} = \lambda \mathrm{x}$;

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

Example

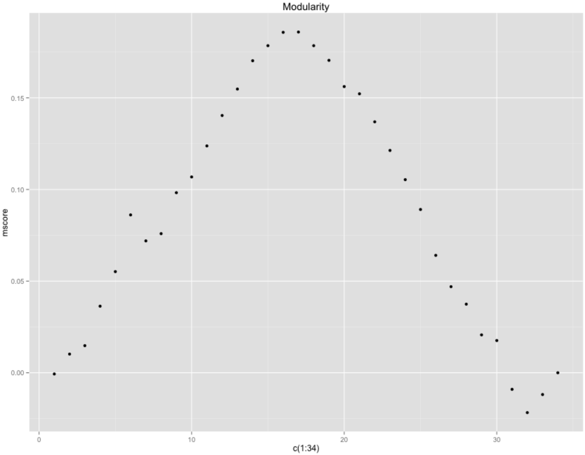


Example

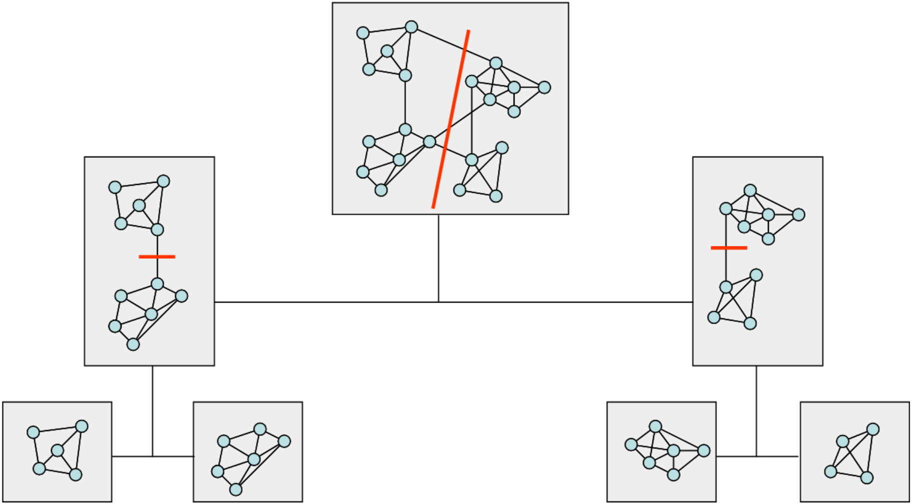


25
26
12
13
17
7
8
28
32
24
6
4
29
5
1
15
3
20
30
11
33
2
34
9
14
18
16
22
27
10
31
21
19
23

Example

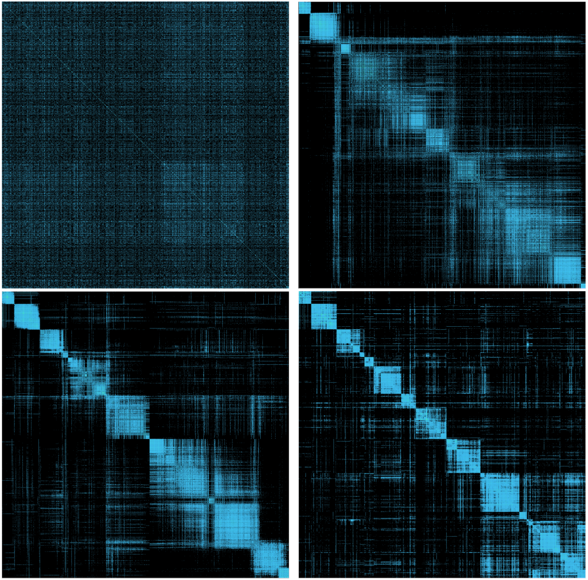


Multilevel spectral



recursive partitioning

Multilevel spectral



Lecture outline

1. Graph paritioning
   - Metrics
   - Algorithms
2. Spectral optimization
   - Min cut
   - Normalized cut
   - Modularity maximization
3. Multilevel spectral
4. Overlapping commmunities
   - Clique percolation method
5. Multi-level optimization
   - Fast community unfolding
6. Random walk methods
   - Walktrap

Community detection

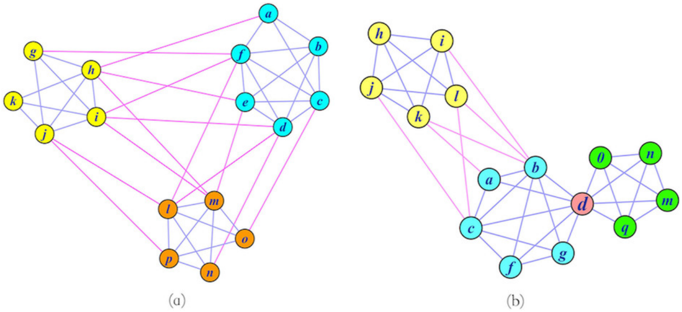
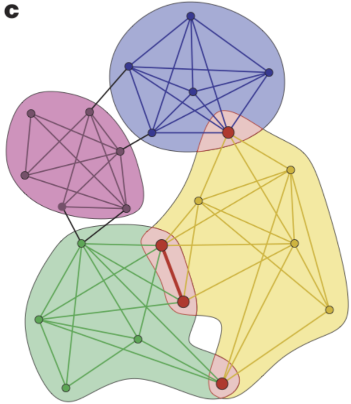


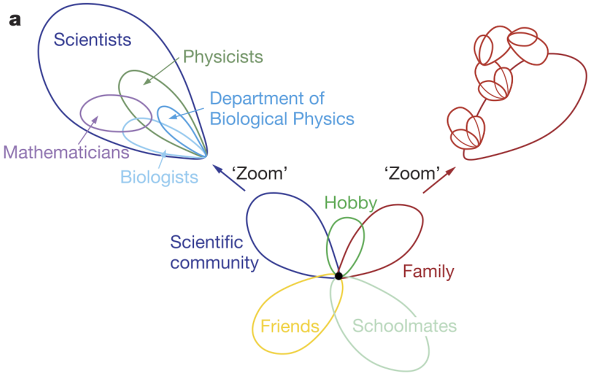
image from W. Liu , 2014

Overlapping communities



Palla, 2005

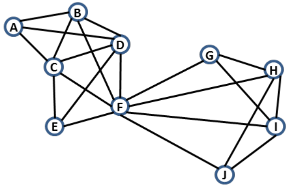
Overlapping communities

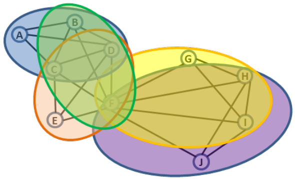


Palla, 2005

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.





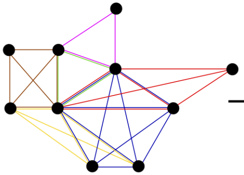
Adjacent 4-cliques

k-clique percolation

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

Palla, 2005

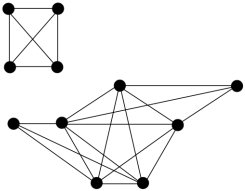
k-clique percolation



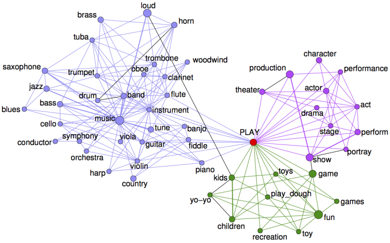
	5	3	2	1	3	1
	3	4	2	1	1	1
	2	2	3	2	1	2
	1	1	2	3	0	1
	3	1	1	0	4	2
	1	1	2	1	2	4

k=4

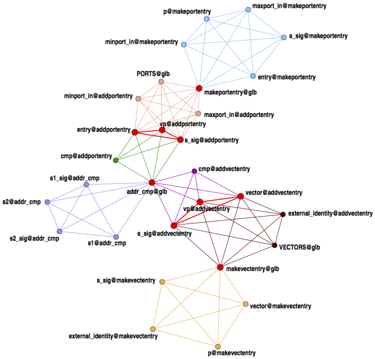
	1	1	0	0	1	0
	1	1	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	1	0
	0	0	0	0	0	1



Palla, 2005



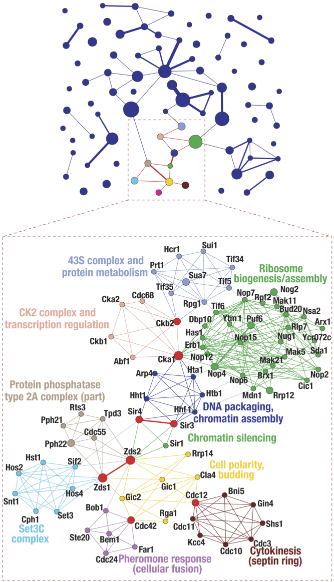
$k = 4$



$k = 5$

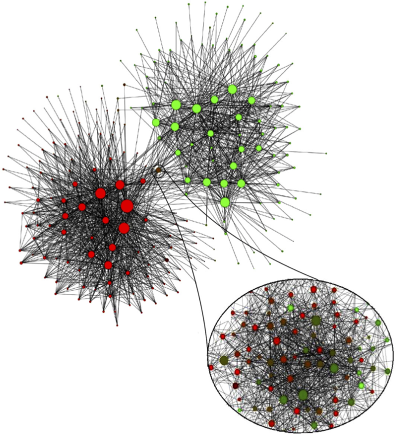
Palla, 2005

k-clique percolation



Fast community unfolding

Multi-resolution scalable method



2 mln mobile phone network
V. Blondel et.al., 2008

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

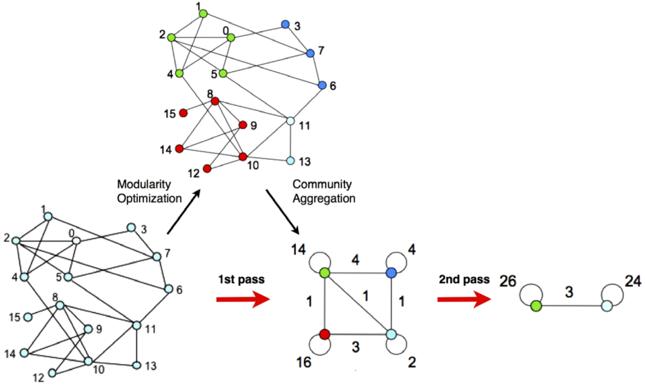
Fast community unfolding

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)

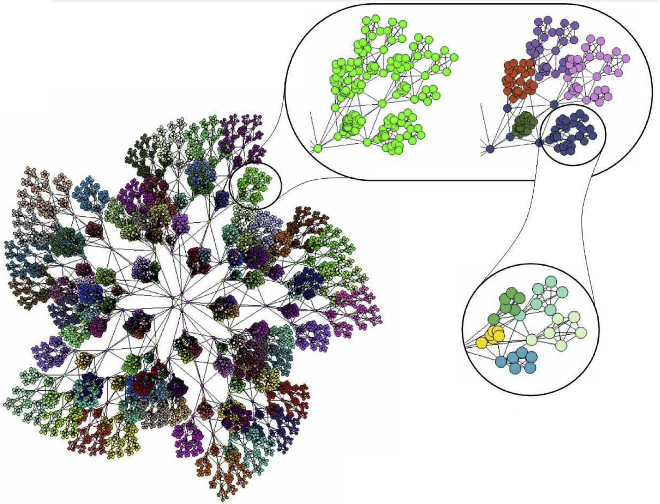
V. Blondel et.al., 2008

Fast community unfolding



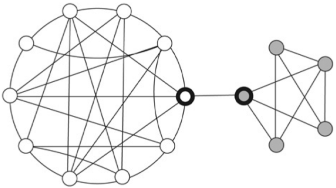
V. Blondel et.al., 2008

Fast community unfolding



V. Blondel et.al., 2008

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):

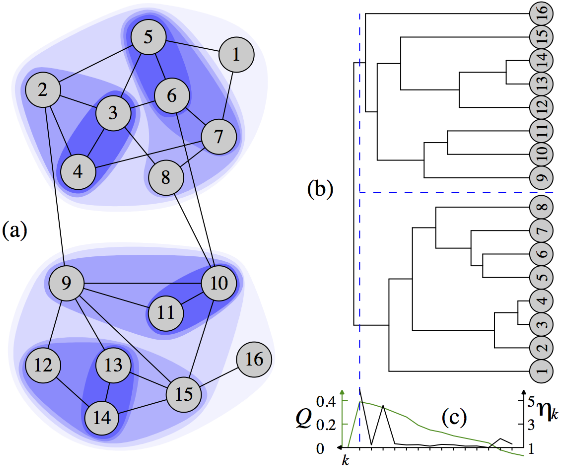
$$\Delta\sigma(C_1, C_2) = \frac{1}{n}\left(\sum_{i \in C_3} r_{iC_3}^2 - \sum_{i \in C_1} r_{iC_1}^2 - \sum_{i \in C_2} r_{iC_2}^2\right)$$

  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

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.
- 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.

Summary

Lectures 1-8

- 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 1-8

- Structural equivalence
  – Vertex equivalence
  – Vertex similarity
- Assortative mixing
  – Assortative and disassortative networks
  – Mixing by node degree
  – Modularity
- Network structures:
  – Cliques
  – k-cores
- Network communities:
  – Graph partitioning
  – Overlapping communities
  – Heuristic methods
  – Random walk based methods

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
- M.E.J. Newman. Modularity and community structure in networks. PNAS Vol. 103, N 23, pp 8577-8582, 2006
- B. Good, Y.-A. de Montjoye, A. Clauset. Performance of modularity maximization in practical contexts, Physical Review E 81, 046106, 2010

References

- 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