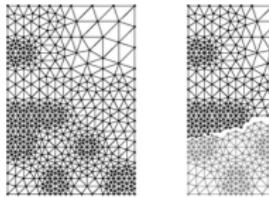


network *clustering*

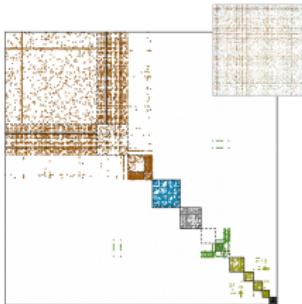
introduction to *network analysis* (*ina*)

Lovro Šubelj
University of Ljubljana
spring 2022/23

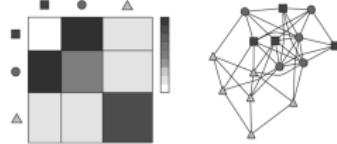
clustering *overview*



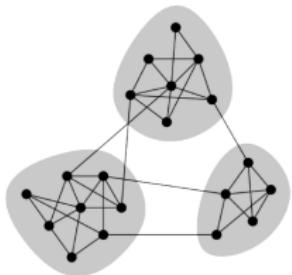
graph partitioning [KL70, Fie73]



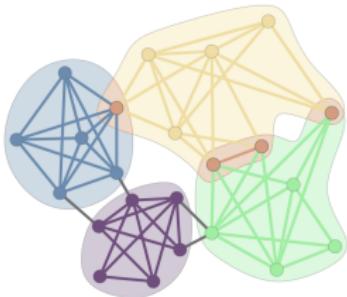
blockmodeling [LW71, WR83]



stochastic block models [Pei15]



communities [GN02]

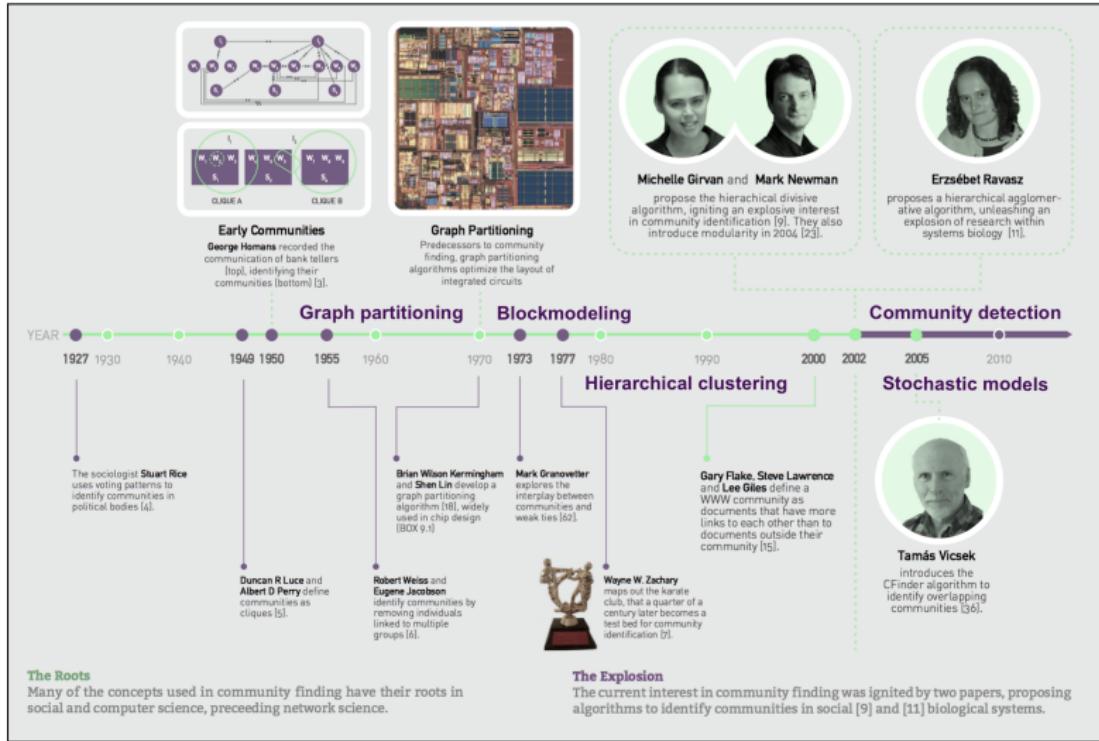


overlapping communities [PDFV05]



link communities [EL09, ABL10]

clustering *history*



graph *partitioning*

introduction to *network analysis* (*ina*)

Lovro Šubelj
University of Ljubljana
spring 2022/23

partitioning *bisection*

— Kernighan-Lin *graph bisection* [KL70]

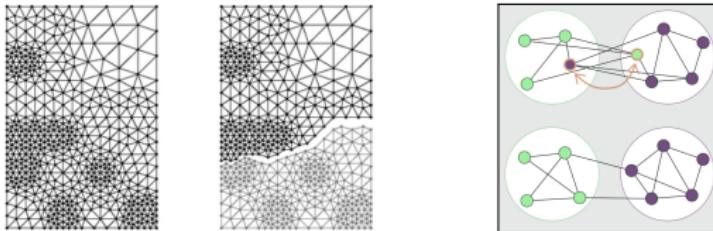
- define *bisection quality* as *cut size*

$$R = \frac{1}{2} \sum_{ij} A_{ij}(1 - \delta_{c_i c_j}) \quad \forall i : c_i = \pm 1$$

1. swap nodes by minimizing cut size $\mathcal{O}(cn^2 m)$

$$\Delta R_{ij} = k_i^{\text{ext}} - k_i^{\text{in}} + k_j^{\text{ext}} - k_j^{\text{in}} - 2A_{ij}$$

2. repeat 1. until $\min(n_1, n_2)$ nodes swapped
3. return bisection minimizing cut size



* example mesh bisection with cut size equal to 40

partitioning *spectral*

— Fiedler *graph bisection* [Fie73]

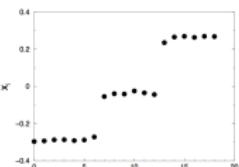
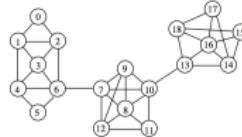
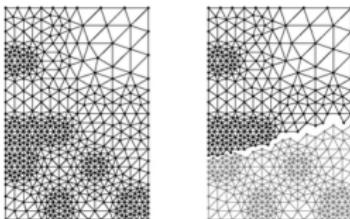
- define *bisection quality* as *cut size*

$$R = \frac{1}{4} \sum_{ij} A_{ij}(1 - s_i s_j) \quad \forall i : s_i = \delta_{c_i c_1} - \delta_{c_i c_2}$$

- formulate *eigenvector problem* of *graph Laplacian*

$$R = \frac{1}{4} \sum_i k_i s_i^2 - \frac{1}{4} \sum_{ij} A_{ij} s_i s_j = \frac{1}{4} \sum_{ij} (k_i \delta_{ij} - A_{ij}) s_i s_j = \frac{1}{4} s^T L s \simeq \frac{1}{4} v^T L v = \frac{n_1 n_2}{n} \lambda$$

1. find *eigenvector* v_2 of L with *algebraic connectivity* λ_2 $\mathcal{O}(nm)$
2. assign n_1 nodes with *largest/smallest* v_2 to C_1
3. return *bisection minimizing cut size*



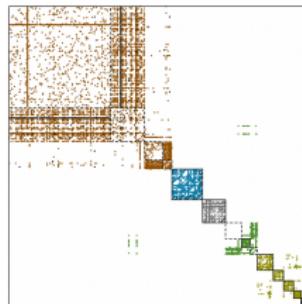
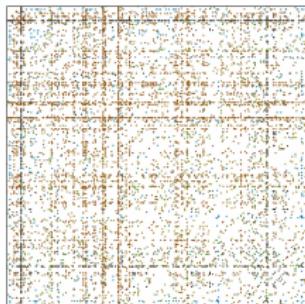
see *graclus* and *metis* implementations

†

example mesh bisection with cut size equal to 46

partitioning *blockmodeling*

- standard *equivalence blockmodeling* [DBF05]
 - define *node similarity* as (*structural*) *equivalence*
$$s_{ij} \sim |\Gamma_i \cap \Gamma_j|$$
 - 1. *blockmodeling* by (*hierarchical*) *clustering* $\mathcal{O}(n^2)$
 - 2. return *block model* at desired *clustering resolution*



see **catrge** implementation



`javax.swing`, `javax.management`, `javax.naming`, `javax.print`, `javax.xml`, `javax.lang` etc.

community detection

introduction to *network analysis* (*ina*)

Lovro Šubelj
University of Ljubljana
spring 2022/23

community *agglomerative*

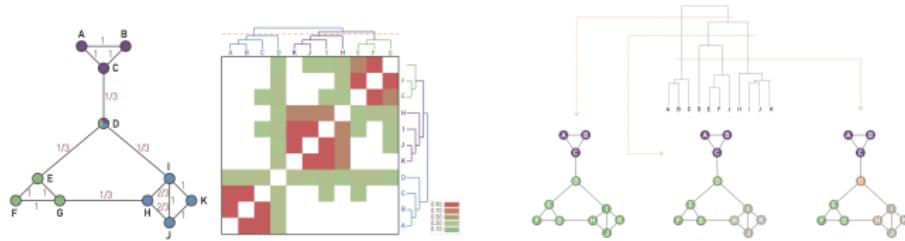
- Ravasz *hierarchical clustering* [RSM⁺02]
 - define *node similarity* as *topological overlap*

$$s_{ij} = \frac{|\Gamma_i \cap \Gamma_j| + A_{ij}}{\min(k_i, k_j)}$$

- define *cluster similarity* as *average linkage*

$$S_{ij} = \frac{1}{n_i n_j} \sum_{xy} s_{xy} \delta_{c_x c_i} \delta_{c_y c_j}$$

1. bottom-up *agglomerative hierarchical clustering* $\mathcal{O}(n^2)$
2. cut *cluster dendrogram* at desired *clustering resolution*



community *divisive*

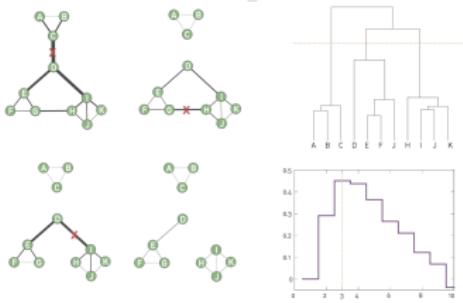
— Girvan-Newman *hierarchical clustering* [GN02]

– define *node dissimilarity* as *link betweenness*

$$\sigma_{ij} = \sum_{st \notin \{i,j\}} \frac{g_{st}^{ij}}{g_{st}}$$

1. top-down *divisive hierarchical clustering* $\mathcal{O}(nm^2)$
2. cut *cluster dendrogram* at *maximum modularity*

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



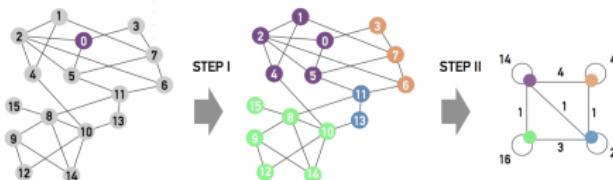
community *modularity*

— Louvain *modularity optimization* [BGLL08]

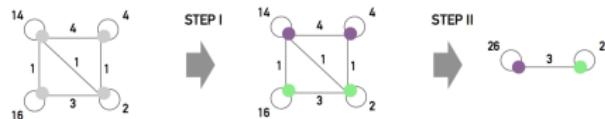
1. set *node community* by *modularity optimization* $\mathcal{O}(cm)$
2. *aggregate community nodes into supernodes* and repeat 1.
3. return *community structure maximizing modularity*

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

1ST PASS



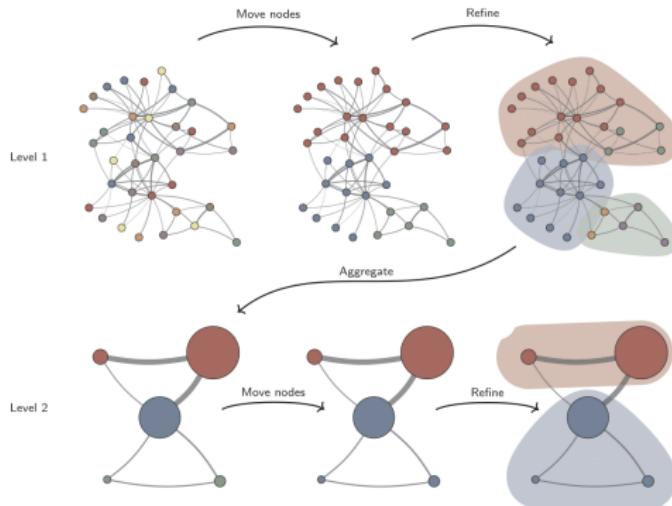
2ND PASS



see louvain implementation

community *modularity*

- Leiden *modularity optimization* [TWVE19]
 - ×. *improved Louvain algorithm* with *quality guarantees*



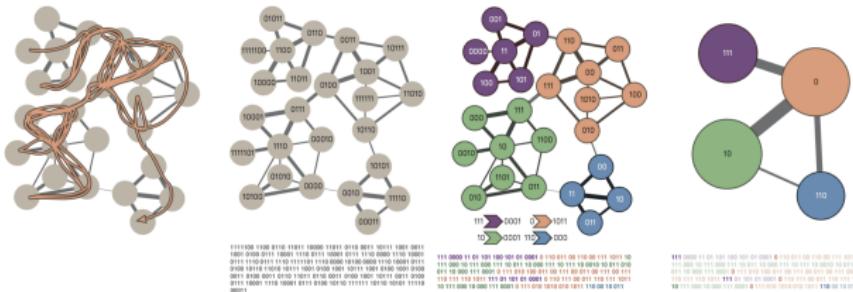
see `leidenalg` implementation

community *map equation*

— Infomap *map equation compression* [RB08]

1. set node community by optimal coding $\mathcal{O}(m \log m)$
 2. compress community nodes into supernodes and repeat 1.
 3. return community structure maximizing map equation

$$\mathcal{L} = \sum_i p_{i \rightsquigarrow} H(\tilde{\mathcal{C}}) + \sum_i p_{i \leftarrow} H(\mathcal{C}_i)$$



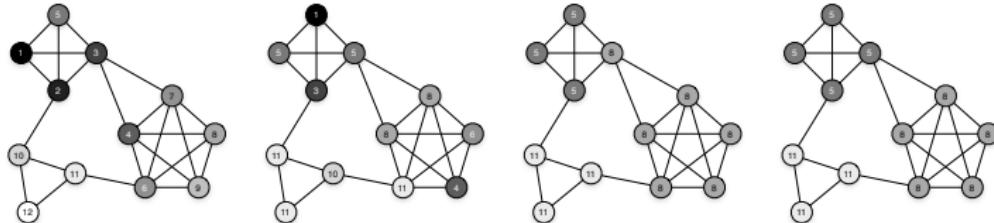
see `mapequation` implementation

community *propagation*

— Raghavan *label propagation* [RAK07, ŠB11]

1. set *node community* by *neighbors frequency* $\mathcal{O}(cm)$
2. *randomly shuffle nodes* and repeat 1. *until convergence*
3. return *community structure connected components*

$$\forall i : c_i = \arg \max_c \sum_j A_{ij} \delta_{c_j c}$$

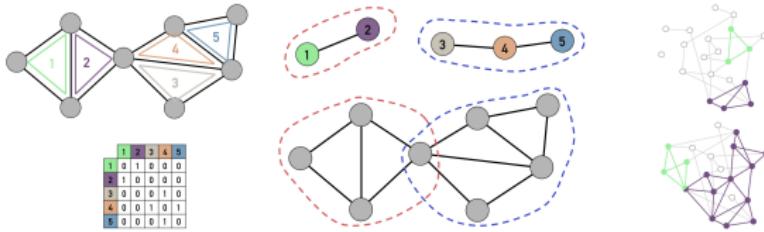


see **balanced** implementation

community *percolation*

— Palla *clique percolation* [PDFV05, KKKS08]

1. find *k-node cliques* by *sequential enumeration* $\mathcal{O}(n_k)$
2. *merge clique nodes into supernodes* and *link adjacent*
adjacent *k-node cliques* share $k - 1$ nodes
3. *return clique structure connected components*
clique percolation at $(kn - n)^{\frac{1}{1-k}}$



see **kclique** implementation

community *links*

- Ahn *link clustering* [EL09, ABL10]

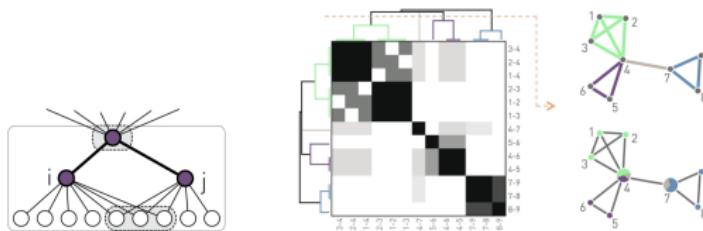
- define *link similarity* as *neighbors index*

$$\forall ij \in \Gamma_x : s_{ij}^x = \frac{|\Gamma_i^+ \cap \Gamma_j^+|}{|\Gamma_i^+ \cup \Gamma_j^+|}$$

- define *cluster similarity* as *single linkage*

$$S_{ij} = \max_{xy \in \Gamma_z} (s_{xy}^z \delta_{c_{xz} c_i} \delta_{c_{yz} c_j})$$

1. bottom-up *agglomerative hierarchical clustering* $\mathcal{O}(m^2)$
2. cut *dendrogram* at desired *clustering resolution*



see [linkcomm](#) implementation

community *measures*

- degree K , expansion E and Flake F [FLG00, RCC⁺04] of $\{C\}$

$$K = \frac{1}{n} \sum_{ij} A_{ij} \delta_{c_i c_j} = \langle k \rangle - E \quad F = \frac{|\{i : \sum_j A_{ij} \delta_{c_i c_j} < k_i/2\}|}{n}$$

- normalized mutual information NMI [DDGDA05] of $\{C\}, \{D\}$

- p_c & p_{cd} are *standard* & *joint distributions* of $\{C\}, \{D\}$
- $H(C)$ & $H(C|D)$ are *standard* & *conditional entropies*
- MI & VI are *mutual* & *variation of information*

$$NMI = \frac{2MI(C,D)}{H(C)+H(D)} = \frac{2H(C)-2H(C|D)}{H(C)+H(D)} = \frac{2H(C)+2\sum_{CD} p_{cd} \log \frac{p_{cd}}{p_d}}{-\sum_C p_c \log p_c + H(D)}$$

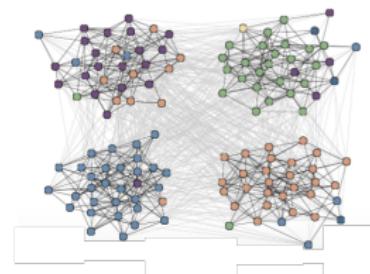
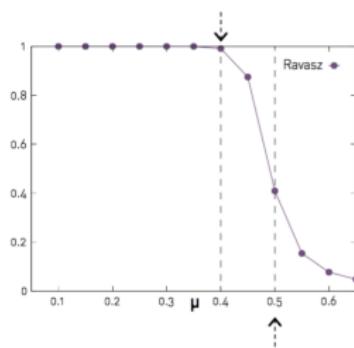
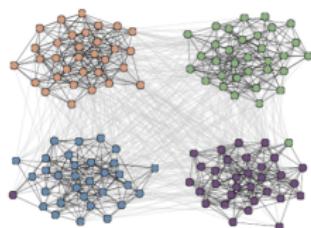
- normalized variation of information NVI [Mei07, KLN08]

$$NVI = \frac{VI(C,D)}{\log n} = \frac{H(C|D)+H(D|C)}{\log n}$$

community *benchmarks*

- Girvan-Newman *synthetic graphs* [GN02]
- *planted partition* controlled by *mixing parameter* μ

$$n = 128 \quad \langle k \rangle = \langle k^{\text{int}} \rangle + \langle k^{\text{ext}} \rangle = 16 \quad \mu = \frac{\langle k^{\text{ext}} \rangle}{\langle k \rangle}$$



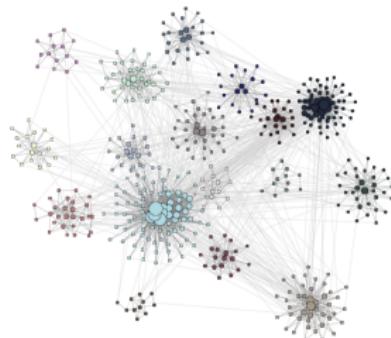
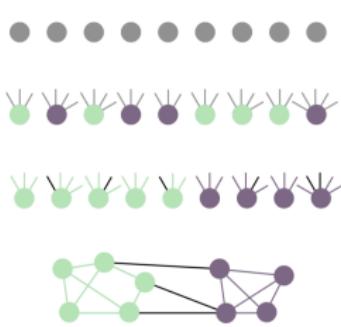
community *benchmarks*

- Lanchichinetti *synthetic graphs* [LFR08]
- *power-law distributions* $p_k \sim k^{-\gamma_k}$ & $p_s \sim s^{-\gamma_s}$
- *planted communities* controlled by *mixing parameter* μ

$$n = 1000, n_c \in [10, 50]$$

$$\gamma_k \in [2, 3], \gamma_s \in [1, 2]$$

$$\mu = \frac{\langle k^{\text{ext}} \rangle}{\langle k \rangle}$$



clustering *references*

-  Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann.
Link communities reveal multiscale complexity in networks.
Nature, 466(7307):761–764, 2010.
-  A.-L. Barabási.
Network Science.
Cambridge University Press, Cambridge, 2016.
-  V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre.
Fast unfolding of communities in large networks.
J. Stat. Mech., P10008, 2008.
-  Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj.
Generalized Blockmodeling.
Cambridge University Press, Cambridge, 2005.
-  Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas.
Comparing community structure identification.
J. Stat. Mech., page P09008, 2005.
-  Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj.
Exploratory Social Network Analysis with Pajek: Expanded and Revised Second Edition.
Cambridge University Press, Cambridge, 2011.
-  David Easley and Jon Kleinberg.
Networks, Crowds, and Markets: Reasoning About a Highly Connected World.
Cambridge University Press, Cambridge, 2010.
-  Ernesto Estrada and Philip A. Knight.
A First Course in Network Theory.
Oxford University Press, 2015.

clustering *references*

-  T. S. Evans and R. Lambiotte.
Line graphs, link partitions and overlapping communities.
Phys. Rev. E, 80(1):016105, 2009.
-  M. Fiedler.
Algebraic connectivity of graphs.
Czech. Math. J., 23:298–305, 1973.
-  Gary William Flake, Steve Lawrence, and C. Lee Giles.
Efficient identification of web communities.
In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, USA, 2000.
-  M. Girvan and M. E. J Newman.
Community structure in social and biological networks.
P. Natl. Acad. Sci. USA, 99(12):7821–7826, 2002.
-  Jussi M. Kumpula, Mikko Kivelä, Kimmo Kaski, and Jari Saramäki.
Sequential algorithm for fast clique percolation.
Phys. Rev. E, 78(2):026109, 2008.
-  Brian W. Kernighan and S. Lin.
An efficient heuristic procedure for partitioning graphs.
Bell Sys. Tech. J., 49(2):291–308, 1970.
-  Brian Karrer, Elizaveta Levina, and M. E. J. Newman.
Robustness of community structure in networks.
Phys. Rev. E, 77(4):046119, 2008.

clustering *references*

-  Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi.
Benchmark graphs for testing community detection algorithms.
Phys. Rev. E, 78(4):046110, 2008.
-  F. Lorrain and H. C. White.
Structural equivalence of individuals in social networks.
J. Math. Sociol., 1(1):49–80, 1971.
-  Marina Meila.
Comparing clusterings: An information based distance.
J. Multivariate Anal., 98(5):873–895, 2007.
-  Mark E. J. Newman.
Networks.
Oxford University Press, Oxford, 2nd edition, 2018.
-  Gergely Palla, Imre Derényi, Illes Farkas, and Tamas Vicsek.
Uncovering the overlapping community structure of complex networks in nature and society.
Nature, 435(7043):814–818, 2005.
-  Tiago P. Peixoto.
Model selection and hypothesis testing for large-scale network models with overlapping groups.
Phys. Rev. X, 5(1):011033, 2015.
-  Usha Nandini Raghavan, Reka Albert, and Soundar Kumara.
Near linear time algorithm to detect community structures in large-scale networks.
Phys. Rev. E, 76(3):036106, 2007.
-  M. Rosvall and C. T. Bergstrom.
Maps of random walks on complex networks reveal community structure.
P. Natl. Acad. Sci. USA, 105(4):1118–1123, 2008.

clustering *references*

-  Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi.
Defining and identifying communities in networks.
P. Natl. Acad. Sci. USA, 101(9):2658–2663, 2004.
-  E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and Albert László Barabási.
Hierarchical organization of modularity in metabolic networks.
Science, 297(5586):1551–1555, 2002.
-  Lovro Šubelj and Marko Bajec.
Robust network community detection using balanced propagation.
Eur. Phys. J. B, 81(3):353–362, 2011.
-  V. A. Traag, Ludo Waltman, and Nees Jan Van Eck.
From Louvain to Leiden: Guaranteeing well-connected communities.
Sci. Rep., 9:5233, 2019.
-  D. R. White and K. P. Reitz.
Graph and semigroup homomorphisms on networks of relations.
Soc. Networks, 5(2):193–234, 1983.