Graph mining - lesson 2
Graph Clustering

Nathalie Vialaneix

nathalie.vialaneix@inrae.fr
http://www.nathalievialaneix.eu

M2 Statistics & Econometrics
January 15th, 2020

Sketch of this lesson

Issue at stake:

- short overview of different types of methods for vertex clustering
- only simple clustering (although some methods for overlapping clustering, clustering according to vertex/edge attributes, clustering of bipartite graphs... also exist)

Notations for this class

Notations

In the following, a graph $\mathcal{G} = (V, E, W)$ with:

- V: set of vertices $\{x_1, \ldots, x_n\}$;
- E: set of (undirected) edges. $m = |E|$;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$ (also called, *adjacency matrix*).

Notations for this class

Notations

In the following, a graph $\mathcal{G} = (V, E, W)$ with:

- V: set of vertices $\{x_1, \ldots, x_n\}$;
- E: set of (undirected) edges. $m = |E|$;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$ (also called, *adjacency matrix*).

If needed, attributes for the nodes will be denoted by $f_j(x_i)$ (jth attribute for node i) and attributes for the edges (other than the weights) by $g_j(x_i, x_{i'})$ (jth attribute for the edge $(x_i, x_{i'})$).

A short overview of vertex clustering

Purpose: Find communities or modules (*i.e.*, groups of vertices) st vertices inside the community are strongly connected whereas vertices between two communities are slightly connected.

A short overview of vertex clustering

Purpose: Find communities or modules (*i.e.*, groups of vertices) st vertices inside the community are strongly connected whereas vertices between two communities are slightly connected.

Some approaches to perform such task:

- optimizing a given criterion (*e.g.*, modularity maximization)
- spectral clustering
- model based clustering
- ... (see **[Fortunato and Barthélémy, 2007, Schaeffer, 2007, Brohée and van Helden, 2006]**)

Outline

Modularity optimization

Spectral clustering

Model based clustering

Clustering based on criterion optimization

- "Cut" criteria: Given a number of clusters, K, find the partition of V, $\mathcal{C}_1, \ldots, \mathcal{C}_K$ such that it solves the mincut problem, *i.e.*, it minimizes

$$\mathrm{cut}(\mathcal{C}_1, \ldots, \mathcal{C}_K) = \frac{1}{2} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{C}_k,\, x_j \notin \mathcal{C}_k} W_{ij}$$

Clustering based on criterion optimization

- "Cut" criteria: Given a number of clusters, K, find the partition of V, $\mathcal{C}_1, \ldots, \mathcal{C}_K$ such that it solves the mincut problem, *i.e.*, it minimizes

$$\text{cut}(\mathcal{C}_1, \ldots, \mathcal{C}_K) = \frac{1}{2} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{C}_k,\, x_j \notin \mathcal{C}_k} W_{ij}$$

Problem: The mincut problem often only separates individual vertices from the rest of the graph.

Clustering based on criterion optimization

- "Cut" criteria: Given a number of clusters, K, find the partition of V, $\mathcal{C}_1, \ldots, \mathcal{C}_K$ such that it solves the "RatioCut" problem, *i.e.*, it minimizes

$$\text{RatioCut}(\mathcal{C}_1, \ldots, \mathcal{C}_K) = \frac{1}{2} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{C}_k,\, x_j \notin \mathcal{C}_k} \frac{W_{ij}}{|\mathcal{C}_k|}$$

(forces larger communities than the mincut problem).

Clustering based on criterion optimization

- "Cut" criteria: Given a number of clusters, K, find the partition of V, $\mathcal{C}_1, \ldots, \mathcal{C}_K$ such that   it solves the "NCut" problem, *i.e.*, it minimizes

$$\mathrm{NCut}(\mathcal{C}_1, \ldots, \mathcal{C}_K) = \frac{1}{2} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{C}_k,\, x_j \notin \mathcal{C}_k} \frac{W_{ij}}{\mathrm{Vol}(\mathcal{C}_k)}$$

in which $\mathrm{Vol}(\mathcal{C}_k) = \sum_{x_i,\, x_j \in \mathcal{C}_k} W_{ij}$ (also forces larger communities than the mincut problem).

Clustering based on criterion optimization

- "Cut" criteria

- "Modularity" criterion **[Newman and Girvan, 2004]**: Given a number of clusters, K, find the partition of V, $\mathcal{C}_1, \ldots, \mathcal{C}_K$ which maximizes

$$Q(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \frac{1}{2m} \sum_{k=1}^{K} \sum_{x_i,\, x_j \in \mathcal{C}_k} (W_{ij} - P_{ij})$$

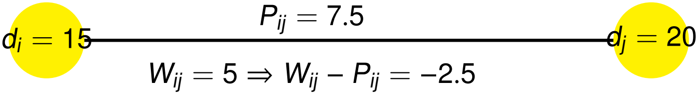
with P_{ij}: weight of a "null model" (graph with the same degree distribution but no preferential attachment): $P_{ij} = \frac{d_i d_j}{2m}$ with $d_i = \frac{1}{2} \sum_{j \neq i} W_{ij}$.

Interpretation of the modularity

A good clustering should maximize the modularity:

- $Q \nearrow$ when (x_i, x_j) are in the same cluster and $W_{ij} \gg P_{ij}$
- $Q \searrow$ when (x_i, x_j) are in two different clusters and $W_{ij} \gg P_{ij}$

$(m = 20)$



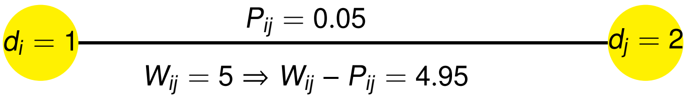
i and j in the same cluster decreases the modularity

Interpretation of the modularity

A good clustering should maximize the modularity:

- $Q \nearrow$ when (x_i, x_j) are in the same cluster and $W_{ij} \gg P_{ij}$
- $Q \searrow$ when (x_i, x_j) are in two different clusters and $W_{ij} \gg P_{ij}$

$(m = 20)$



i and j in the same cluster increases the modularity

Interpretation of the modularity

A good clustering should maximize the modularity:

- $Q \nearrow$ when (x_i, x_j) are in the same cluster and $W_{ij} \gg P_{ij}$
- $Q \searrow$ when (x_i, x_j) are in two different clusters and $W_{ij} \gg P_{ij}$
- Modularity
  - helps separate hubs ($\neq$ spectral clustering or min cut criterion);
  - is not an increasing function of the number of clusters: useful to choose the relevant number of clusters (with a grid search: several values are tested, the clustering with the highest modularity is kept)

Advantages and drawbacks

- mincut is not adapted to vertex clustering in practice (clusters with isolated vertices)
- the other three methods are NP hard to solve...

Advantages and drawbacks

- mincut is not adapted to vertex clustering in practice (clusters with isolated vertices)
- the other three methods are NP hard to solve...
- the modularity takes into account skewness in degree distribution by correcting the importance of a vertex by its degree: it is often more adapted to real life graphs
- [**Fortunato and Barthélémy, 2007**] showed that modularity has a resolution issue. [**Bickel and Chen, 2009**] gave conditions for consistency of the clusters obtained by modularity optimization in Stochastic Block Models (SBM).

Advantages and drawbacks

- mincut is not adapted to vertex clustering in practice (clusters with isolated vertices)
- the other three methods are NP hard to solve...
- the modularity takes into account skewness in degree distribution by correcting the importance of a vertex by its degree: it is often more adapted to real life graphs
- [Fortunato and Barthélémy, 2007] showed that modularity has a resolution issue. [Bickel and Chen, 2009] gave conditions for consistency of the clusters obtained by modularity optimization in Stochastic Block Models (SBM).

Remark: Relaxation of RatioCut problem and NCut problem gives spectral clustering. Modularity optimization is often solved by approximation methods.

A short description of approximation methods for modularity optimization

- simple greedy algorithms ([Newman, 2004] and [Clauset et al., 2004] for a fast version): hierarchical clustering which merges pairs of vertices with the highest contribution to modularity

A short description of approximation methods for modularity optimization

- simple greedy algorithms ([Newman, 2004] and [Clauset et al., 2004] for a fast version): hierarchical clustering which merges pairs of vertices with the highest contribution to modularity
- multi-level greedy algorithms ([Blondel et al., 2008], also known as "Louvain algorithm" and [Noack and Rotta, 2009] for an improved version): hierarchical approach in which vertices are sometimes re-assigned to a different community in a greedy way

A short description of approximation methods for modularity optimization

- simple greedy algorithms ([Newman, 2004] and [Clauset et al., 2004] for a fast version): hierarchical clustering which merges pairs of vertices with the highest contribution to modularity
- multi-level greedy algorithms ([Blondel et al., 2008], also known as "Louvain algorithm" and [Noack and Rotta, 2009] for an improved version): hierarchical approach in which vertices are sometimes re-assigned to a different community in a greedy way
- simulated annealing ([Reichardt and Bornholdt, 2006] uses a spin-glass model which, in some cases, is equivalent to modularity maximization)

A short description of approximation methods for modularity optimization

- simple greedy algorithms ([Newman, 2004] and [Clauset et al., 2004] for a fast version): hierarchical clustering which merges pairs of vertices with the highest contribution to modularity
- multi-level greedy algorithms ([Blondel et al., 2008], also known as "Louvain algorithm" and [Noack and Rotta, 2009] for an improved version): hierarchical approach in which vertices are sometimes re-assigned to a different community in a greedy way
- simulated annealing ([Reichardt and Bornholdt, 2006] uses a spin-glass model which, in some cases, is equivalent to modularity maximization)

...to be compared (when usable) with the exact optimization (only useable for small graphs).

Example

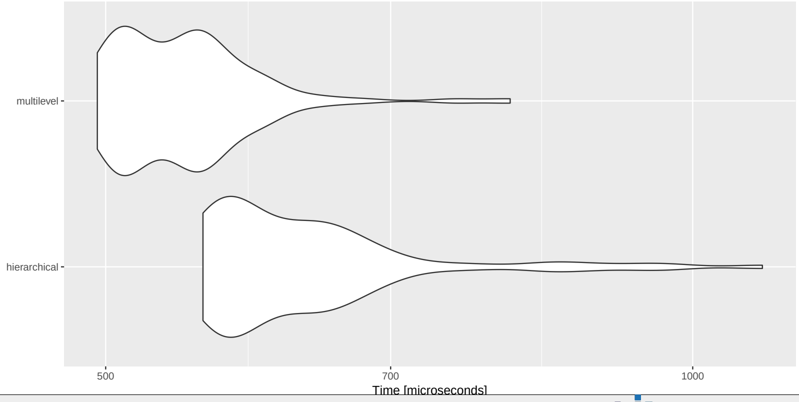
Computational time needed by the different solution to find a clustering for NVV network:

	time
hierarchical	0.003
multilevel	0.002
annealing	1.266

Computational time (greedy approaches)

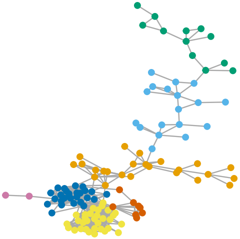
Difference (computational time) between the first two approaches (100 evaluations):

```
## Coordinate system already present.  Adding new
coordinate system, which will replace the existing
one.
```

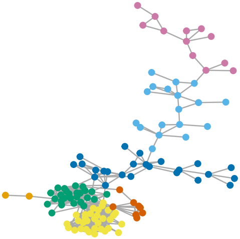


Accuracy of the clustering

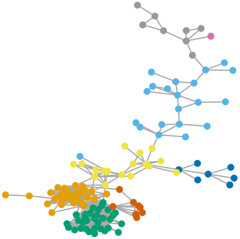
hierarchical − 0.567 − 7



multilevel − 0.567 − 7



simulated annealing − 0.5628 − 10



Outline

Modularity optimization

Spectral clustering

Model based clustering

Relation between RatioCut and Laplacian

[von Luxburg, 2007] shows that minimizing

$$\text{RatioCut}(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{2} \sum_{k=1}^{2} \sum_{x_i \in \mathcal{C}_k,\, x_j \notin \mathcal{C}_k} \frac{W_{ij}}{|\mathcal{C}_k|}$$

is equivalent to the following constrained problem:

$$\min_{\mathcal{C}_1,,\mathcal{C}_2} v^\top L v \text{ st } v \perp \mathbf{1}_n \text{ and } \|v\| = \sqrt{n}$$

for v the vector of $\mathbb{R}^n$ obtained from the partition by:

$$v_i = \begin{cases} \sqrt{(|\mathcal{C}_2|)/|\mathcal{C}_1|} & \text{if } v_i \in \mathcal{C}_1 \\ -\sqrt{|\mathcal{C}_1|/(|\mathcal{C}_2|)} & \text{otherwise.} \end{cases}$$

and L is the Laplacian of the graph, $n \times n$-matrix with entries:

$$L_{ij} = \begin{cases} -W_{ij} & \text{if } i \neq j \\ d_i = \sum_{j \neq i} W_{ij} & \text{otherwise} \end{cases}.$$

... and more remarks

- this is a discrete (since v can only have two values) and NP-hard problem;

... and more remarks

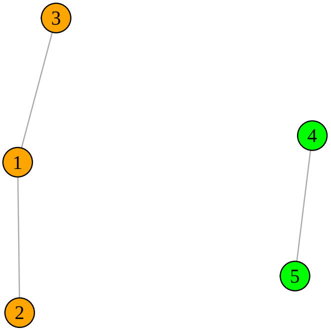
- this is a discrete (since v can only have two values) and NP-hard problem;
- the same relation holds between NCut problem and normalized Laplacian $D^{-1/2}LD^{-1/2}$ is which $D = \mathrm{Diag}(d_1, \ldots, d_n)$;

... and more remarks

- this is a discrete (since v can only have two values) and NP-hard problem;
- the same relation holds between NCut problem and normalized Laplacian $D^{-1/2}LD^{-1/2}$ is which $D = \mathrm{Diag}(d_1, \ldots, d_n)$;
- a generalization of these results exist for $K > 2$.

Some properties of the Laplacian

Relations with the graph structure:



has a null space spanned by the vectors $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$.

Some properties of the Laplacian

Relations with the graph structure: the vector $\mathbf{1}_n$ spans the null space for connected graphs.

Some properties of the Laplacian

Relations with the graph structure:

Random walk point of view: If we consider a random walk on the graph with probability to jump from one node to the other equal to $\frac{W_{ij}}{d_i}$ then $\text{NCut}(A_1, A_2)$ is interpreted as the probability to go from C_1 to C_2 or from C_2 to C_1.

Some properties of the Laplacian

Relations with the graph structure:

Random walk point of view: If we consider a random walk on the graph with probability to jump from one node to the other equal to $\frac{W_{ij}}{d_i}$ then the average time to go from one node to another (commute time) is given by L^+ **[Fouss et al., 2007]**.

Spectral clustering: relaxing the constrains

K has to be given. Solving $\min_{\mathcal{C}_1,,\mathcal{C}_2} \text{Tr}(\mathbf{U}^\top L \mathbf{U})$ for a $K \times n$ matrix $\mathbf{U}$ st $\mathbf{U}^\top \mathbf{U} = 1$:

1. Compute the first K eigenvectors of L, $\mathbf{u}^1$, ..., $\mathbf{u}^K$ and write $\mathbf{U} = (\mathbf{u}^1, \ldots, \mathbf{u}^K)$ (a $n \times K$ matrix).

Spectral clustering: relaxing the constrains

K has to be given. Solving $\min_{\mathcal{C}_1,,\mathcal{C}_2} \text{Tr}(\mathbf{U}^\top L \mathbf{U})$ for a $K \times n$ matrix $\mathbf{U}$ st $\mathbf{U}^\top\mathbf{U} = 1$:

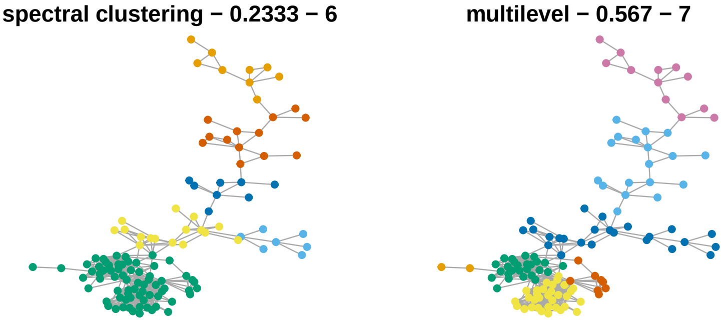
1. Compute the first K eigenvectors of L, $\mathbf{u}^1, \ldots, \mathbf{u}^K$ and write $\mathbf{U} = (\mathbf{u}^1, \ldots, \mathbf{u}^K)$ (a $n \times K$ matrix).

2. For $i = 1, \ldots, n$, denote $\mathbf{u}_i \in \mathbb{R}^K$ the i-th row of $\mathbf{U}$. Cluster the points $(\mathbf{u}_i)_{i=1,\ldots,n}$ using a clustering algorithm (*e.g.*, k-means).

Spectral clustering in practice

For NVV network, computation time is equal to 0.039 (between the greedy approaches for modularity optimization and simulated annealing for modularity optimization).

Accuracy of the clustering



Modularity is smaller (as expected) and clusters tend to be more unbalanced. An empirical comparison between the performance of spectral clustering and modularity optimization is provided in **[Bickel and Chen, 2009]**. **[Lei and Rinaldo, 2015]** gives conditions for the consistency of spectral clustering in stochastic block models.

Outline

A mixture model for networks

[Snijders and Nowicki, 1997]: The observed network $\mathcal{G}$ is supposed to be the realization of some random graph model in which vertices are organized in groups.

description of the model

- vertices x_i belong to an unknow class in $\{\mathcal{C}_1, ..., \mathcal{C}_K\}$ (K is given) $\Rightarrow$ latent (unobserved) variables

$$Z_i \sim \mathcal{M}(1, \alpha = (\alpha_1, \ldots, \alpha_K))$$

in which α_k is the probability that x_i belongs to $\mathcal{C}_k$

A mixture model for networks

[Snijders and Nowicki, 1997]: The observed network $\mathcal{G}$ is supposed to be the realization of some random graph model in which vertices are organized in groups.

description of the model

- vertices x_i belong to an unknow class in $\{\mathcal{C}_1, ..., \mathcal{C}_K\}$ (K is given) ⇒ latent (unobserved) variables

$$Z_i \sim \mathcal{M}(1, \alpha = (\alpha_1, \ldots, \alpha_K))$$

in which α_k is the probability that x_i belongs to $\mathcal{C}_k$

- given the class membership, the probabilities to have an edge between x_i and x_j are all independant and obtained by:

$$W_{ij} = 1 | Z_{ik} Z_{jk'} = 1 \sim \mathcal{L}(., \pi_{kk'})$$

for a given distribution $\mathcal{L}$

A mixture model for networks

[Snijders and Nowicki, 1997]: The observed network $\mathcal{G}$ is supposed to be the realization of some random graph model in which vertices are organized in groups.

description of the model

- vertices x_i belong to an unknow class in $\{\mathcal{C}_1, ..., \mathcal{C}_K\}$ (K is given) $\Rightarrow$ latent (unobserved) variables

$$Z_i \sim \mathcal{M}(1, \alpha = (\alpha_1, \ldots, \alpha_K))$$

  in which α_k is the probability that x_i belongs to $\mathcal{C}_k$

- given the class membership, the probabilities to have an edge between x_i and x_j are all independant and obtained by: typically, the Bernouilli distribution with probability $\pi_{kk'}$ with $\pi_{kk'} = \begin{cases} p_1 & \text{if } k = k' \\ p_0 & \text{if } k \neq k' \end{cases}$ for $p_1 > p_0$.

Basic principle for using SBM

1. assignments of vertices to groups;
2. parameter estimation ($(\alpha_k)_k$ and $(\pi_{kk'})_{k,k'}$);
3. estimation of the number of groups.

Basic principle for using SBM

1. assignments of vertices to groups;
2. parameter estimation ($(\alpha_k)_k$ and $(\pi_{kk'})_{k,k'}$);
3. estimation of the number of groups.

Estimation is made by Bayesian or frequentist approaches and Variational EM (see *e.g.*, **[Daudin et al., 2008]** for the more computationally efficient frequentist approach). Number of nodes can be chosen using ICL **[Biernacki et al., 2000]**.

Basic principle for using SBM

1. assignments of vertices to groups;
2. parameter estimation ($(\alpha_k)_k$ and $(\pi_{kk'})_{k,k'}$);
3. estimation of the number of groups.

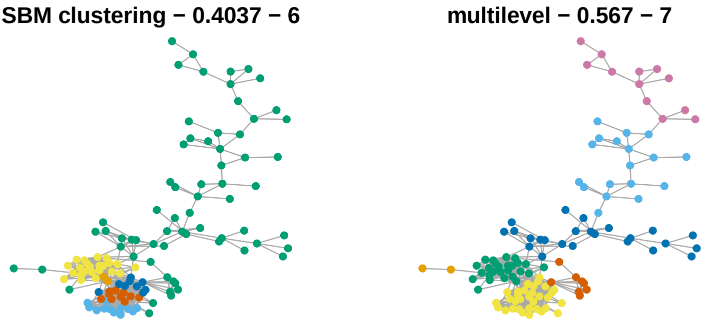
Estimation is made by Bayesian or frequentist approaches and Variational EM (see *e.g.*, **[Daudin et al., 2008]** for the more computationally efficient frequentist approach). Number of nodes can be chosen using ICL **[Biernacki et al., 2000]**.

All this is implemented in the package **blockmodels** **[Léger, 2016]**.

SBM in practice

For NVV network, the computational time of SBM clustering is 2.104. The number of clusters found by the method is 6.

Accuracy of the clustering



Modularity is smaller (as expected) but groups can be interpreted by being sets of vertices with similar connecting patterns.

Comparing clustering

Various metrics ((di)similarities) exist to compare clustering, among which:

- Rand Index **[Rand, 1971]**

$$\frac{\text{number of agreements between the two clusterings}}{n(n-1)/2}$$

- Normalized Mutual Information **[Danon et al., 2005]**

$$\sum_{k=1}^{K_1} \sum_{k'=1}^{K_2} \frac{n_{kk'}}{n} \log \left(\frac{n_{kk'} n}{n_k^1 n_{k'}^2} \right)$$

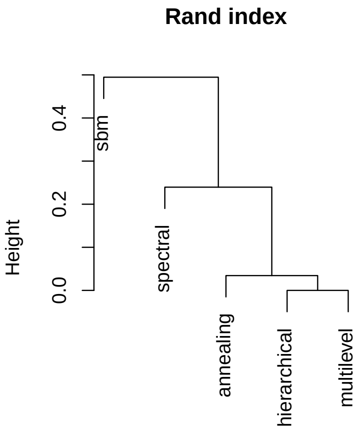
in which K_j is the number of clusters in clustering j, n_k^j is the number of vertices classified into cluster k for clustering j and $n_{kk'}$ is the number of vertices classified into cluster k for clustering 1 and cluster k' for clustering 2. The similarity is normalized so that it is between 0 and 1 (1 is for a perfect match).

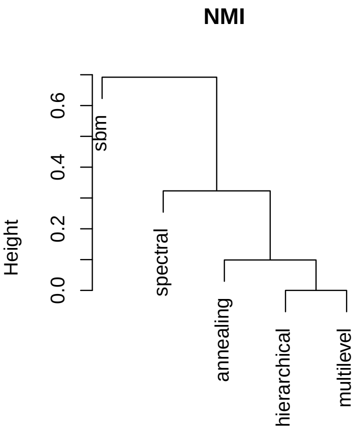
How do clusterings relate?

Method:

1. compute a dissimilarity based on Rand index or NMI (1 – value)
2. perform clustering (of the results of vertex clustering) using hierarchical clustering `hclust`

How do clusterings relate?





Bickel, P. and Chen, A. (2009).
A nonparametric view of network models and Newman-Girvan and other modularities.
Proceedings of the National Academy of Sciences, USA, 106(50):21068–21073.

Biernacki, C., Celeux, G., and Govaert, G. (2000).
Assessing a mixture model for clustering with the integrated completed likelihood.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(7):719–725.

Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008).
Fast unfolding of communites in large networks.
Journal of Statistical Mechanics: Theory and Experiment, P10008:1742–5468.

Brohée, S. and van Helden, J. (2006).
Evaluation of clustering algorithms for protein-protein interaction networks.
BMC Bioinformatics, 7(488).

Clauset, A., Newman, M. E. J., and Moore, C. (2004).
Finding community structure in very large networks.
Physical Review E, 70:066111.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005).
Comparing community structure identification.
Journal of Statistical Mechanics: Theory and Experiment, 2005:P09008.

Daudin, J., Picard, F., and Robin, S. (2008).
A mixture model for random graphs.
Statistics and Computing, 18:173–183.

Fortunato, S. and Barthélémy, M. (2007).
Resolution limit in community detection.
In *Proceedings of the National Academy of Sciences*, volume 104, pages 36–41.
doi:10.1073/pnas.0605965104; URL: http://www.pnas.org/content/104/1/36.abstract.

Fouss, F., Pirotte, A., Renders, J., and Saerens, M. (2007).

Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation.
IEEE Transactions on Knowledge and Data Engineering, 19(3):355–369.

Léger, J. (2016).
Blockmodels: a R-package for estimating in LBM and SBM, with many pdf, with or without covariates.
Preprint arXiv 1602.07587v1. Submitted for publication.

Lei, J. and Rinaldo, A. (2015).
Consistency of spectral clustering in stochastic block models.
The Annals of Statistics, 43(1):215–237.

Newman, M. (2004).
Fast algorithm for detecting community structure in networks.
Physical Review E, 69:066133.

Newman, M. and Girvan, M. (2004).
Finding and evaluating community structure in networks.
Physical Review, E, 69:026113.

Noack, A. and Rotta, R. (2009).
Multi-level algorithms for modularity clustering.
In *SEA 2009: Proceedings of the 8th International Symposium on Experimental Algorithms*, pages 257–268, Berlin, Heidelberg. Springer-Verlag.

Rand, W. (1971).
Objective criteria for the evaluation of clustering methods.
Journal of the American Statistical Association, 66(336):846–850.

Reichardt, J. and Bornholdt, S. (2006).
Statistical mechanics of community detection.
Physical Review, E, 74(016110).

Schaeffer, S. (2007).

Graph clustering.
Computer Science Review, 1(1):27–64.

Snijders, T. and Nowicki, K. (1997).
Estimation and prediction for stochastic block-structures for graphs with latent block structure.
Journal of Classification, 14:75–100.

von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and Computing, 17(4):395–416.