Stat Comput (2014) 24:675–692
DOI 10.1007/s11222-013-9395-3

Detection of structurally homogeneous subsets in graphs

Jean-Benoist Leger · Corinne Vacher · Jean-Jacques Daudin

Received: 12 January 2012 / Accepted: 21 March 2013 / Published online: 16 May 2013

Abstract The analysis of complex networks is a rapidly growing topic with many applications in different domains. The analysis of large graphs is often made via unsupervised classification of vertices of the graph. Community detection is the main way to divide a large graph into smaller ones that can be studied separately. However another definition of a cluster is possible, which is based on the structural distance between vertices. This definition includes the case of community clusters but is more general in the sense that two vertices may be in the same group even if they are not connected. Methods for detecting communities in undirected graphs have been recently reviewed by Fortunato. In this paper we expand Fortunato's work and make a review of methods and algorithms for detecting essentially structurally homogeneous subsets of vertices in binary or weighted and directed and undirected graphs.

Keywords Graphs · Clusters · Random walk · Spectral Clustering · Stochastic Block Model · Bipartite graphs

J.-B. Leger · J.-J. Daudin
INRA, UMR 518 MIA, 16 rue Claude Bernard, Paris, France

J.-B. Leger (✉) · J.-J. Daudin
AgroParisTech, UMR 518 MIA, 16 rue Claude Bernard, Paris, France
e-mail: jleger@agroparistech.fr

J.-J. Daudin
e-mail: daudin@agroparistech.fr

C. Vacher
INRA, UMR 1202 BioGeCo, 69 route d'Arcachon, Cestas, France
e-mail: corinne.vacher@pierroton.inra.fr

C. Vacher
Université de Bordeaux, UMR 1202 BioGeCo, 69 route d'Arcachon, Cestas, France

1 Introduction

The analysis of complex networks is a rapidly growing topic with many applications in different fields such as social sciences, physics, computer science, molecular biology and ecology. The size of the social and biological datasets and the size of the networks created by human-kind are growing with time. This is an issue because networks with thousands of vertices are difficult to analyze as a whole object.

An obvious strategy consists in dividing the big network into smaller independent ones and analyzing each small network separately. Therefore at this time, one of the most important challenges is to build unsupervised classification of the vertices. Most of the current research is focused on the search for a community structure with high connectivity between vertices of the same cluster and low connectivity between vertices of different clusters. This strategy has been used in the field of molecular biology to obtain "independent modules" in metabolic or Protein-Protein interaction networks in the domain of molecular biology. It has also been used to extract the scientific communities from bibliometrics networks or social groups in social networks. Recently a very large and impressive review of community detection methods and algorithms in graphs has been made by Fortunato (2010). This paper describes many methods but also gives some elements for comparing them on benchmarks.

However this strategy has its own limits because in some cases connected vertices may be very different. A typical example is bipartite graphs such as host-parasite networks, where there is a connection between a host species and a parasite species if the species parasites the host species. Therefore the host species and the parasite species may be in the same community, putting in the same bag two very different species. Therefore there is a need for a more general definition of what constitutes a cluster of vertices in networks.

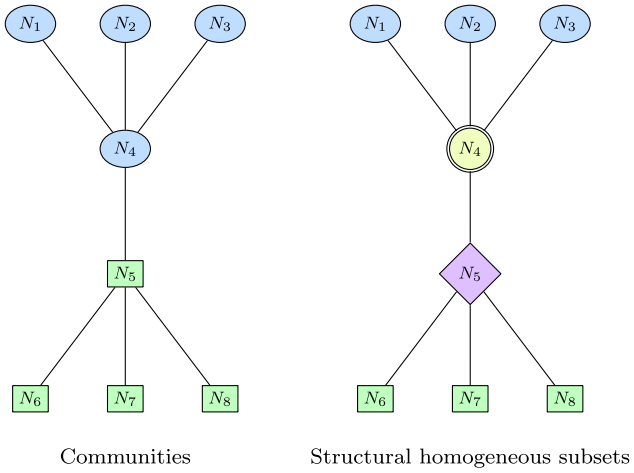


Fig. 1 Difference between communities and structural homogeneous subsets in a hub structure network

Another definition of a cluster is possible, which is based on the structural distance between vertices. Two vertices are in the same group if they have a similar profile of connection to the other vertices. This definition includes the case of community clusters but is more general in the sense that two vertices may be in the same group even if they are not connected. Moreover it is possible to obtain groups of vertices which are not connected within groups but are highly connected to another group of vertices. This notion of structural distance is related to the definition of the Structural Equivalence of Actors in a social network defined by Lorrain and White (1971): actors are structurally equivalent if they have identical relational ties to and from all the actors in a network. These two different approaches are introduced by Burt (1978): *"There are several questions that can be posed for a specific project that might lead an individual to analyze subgroups in terms of cohesion versus structural equivalence. Here, considering a series of such questions, I conclude that subgroups based on structural equivalence are to be preferred to those based on cohesion."*

Two classes of methods for clustering the vertices of graphs can thus be defined with two different goals:

1. to obtain communities i.e. subsets of vertices strongly connected within subsets and loosely connected between subsets,
2. to obtain structurally homogeneous subsets, i.e. subsets of vertices having the same or similar interaction profiles.

The concept of structurally homogeneous subsets generalizes the concept of community in the sense that a community is also a structurally homogeneous subset when the structure of the graph is represented by communities, because vertices in the same community share the same structural connectivity behavior. In Fig. 1, in the same non-bipartite network, there is an example of the difference between communities and structurally homogeneous subsets with a hub structure. Even if hubs are within communities, they have a different behavior in the network structure, and they are classified in different structurally homogeneous subsets.

Fortunato's review is focused on community detection for binary undirected graphs. In this paper we expand Fortunato's work and give a review of methods and algorithms for detecting essentially structurally homogeneous subsets of vertices in binary or weighted and directed or undirected graphs. Moreover we do not try to give an exhaustive list of methods. We prefer to limit the scope to what we have presumed to be the main methods, and to make a self-contained presentation of each of them. Note that we do not present the methods for very large graphs with more than 10^6 vertices, such as the world-wide web or telephone network.

The methods for detecting Structurally Homogeneous Subsets come from three different scientific fields: computer science, physics and statistics. Each scientific community has its own journals and there are few links between them. Statisticians prefer to use probabilistic or statistical models whereas the other two communities use algorithmic or optimization methods. Optimization methods optimize a criterion which represents the quality of the partition of the graph. Algorithmic methods use a sequence of operations to build a partition of the graph. Probabilistic models are models of the process which are supposed to have generated the data and statistical methods are used to estimate the parameters of the probabilistic model. In this review a significant part is given to statistical models which had little space in Fortunato's review.

Section 2 gives the basic notations, some transformations of the base data and a toy graph that will be analyzed throughout the paper. Section 3 presents the methods of clustering based on an algorithm, Sect. 4 presents the methods based on an optimization criterion, Sect. 5 is devoted to statistical models for clustering graphs. Section 6 illustrates methods on the Zachary's Karate Club Network. The last section gives a summary of the methods and some links between them.

2 Notation and an example

Let us consider a graph (or network) $G = (V, E)$ with V the set of n vertices (or vertices) and $E \subset V \times V$ the set of edges. The value of the edge from i to j is $w_{ij} \geq 0$. The self loops ($w_{ii} \neq 0$) may be accepted or not. The matrix W is the matrix of weights. Let $(d^o_W)_i = \sum_j w_{ij}$ be the out-degree of vertex i and $(d^i_W)_j = \sum_i w_{ij}$ be the in-degree of vertex j. The matrix of outgoing degrees D^o_W is the diagonal matrix composed of $(d^o_W)_i, i = 1, \ldots, n$ (with a similar definition for D^i_W).

The graph is non-weighted (binary) when $w_{ij} \in \{0, 1\}$. In that case, W is called the adjacency matrix. The graph is undirected when W is symmetric. In that case, $(d^o_W)_i = (d^i_W)_i = (d_W)_i$ and D_W is called the degree matrix. A bipartite graph is a graph whose vertices can be divided into two disjoint sets A and B such that every edge connects a vertex in A to one in B.

Each method will be illustrated on the toy example presented in the following section. Searching for clusters is only relevant if the graph is void-free, so we suppose that no vertex of the toy graph is isolated.

2.1 A toy example

The toy example (Fig. 2) is a binary, undirected graph, with 10 vertices. Vertices 1 to 5 are of one type, and vertices 6 to 10 of another. The adjacency matrix W and the degree matrix D are (only non-zero values are printed):

$$W = \begin{pmatrix} & & & & & 1 & 1 & 1 & & \\ & & & & & & 1 & 1 & & \\ & & & & & & & 1 & 1 & 1 \\ & & & & & & & & 1 & 1 \\ & & & & & & & & & 1 \\ 1 & & & & & & & & & \\ 1 & 1 & & & & & & & & \\ 1 & 1 & 1 & & & & & & & \\ & & & 1 & 1 & & & & & \\ & & & 1 & 1 & 1 & & & & \end{pmatrix}$$

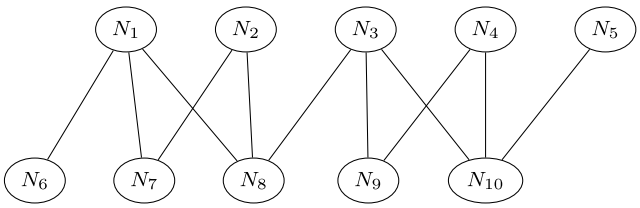


Fig. 2 Bipartite toy-example

$$D = \begin{pmatrix} 3 & & & & & & & & & \\ & 2 & & & & & & & & \\ & & 3 & & & & & & & \\ & & & 2 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 2 & & & \\ & & & & & & & 3 & & \\ & & & & & & & & 2 & \\ & & & & & & & & & 3 \end{pmatrix}$$

The difference between the community and structural homogeneity appears clearly in Fig. 3: if one searches for communities, that is groups of vertices more connected within groups than between groups, one will find the clusters of the column titled "communities" including some vertices which have different connectivity behavior, with respect to bipartite network, in the same group. On the other hand the structurally homogeneous subsets are exclusively composed of one type of vertex. The bottom right case, with 4 clusters, decomposes each of the two clusters of the top left case into two subgroups, separating the two types of vertices. Note that this toy-example is proposed here for illustration purposes, showing that structurally homogeneous clusters may be composed of vertices that are not connected within-group. With real data things are not so clear-cut: in the same graph some clusters may be communities and other ones may be poorly connected within-group and characterized by a high connectivity to a given cluster. Some examples of this configuration of clusters are given in Daudin et al. (2008) for a metabolic network and in Picard et al. (2009) for different biological networks.

2.2 Transformation of the raw graph

The methods for detecting clusters of vertices can either be applied to the raw matrix of weights (or adjacency matrix) or to a weighted matrix obtained by a similarity transformation

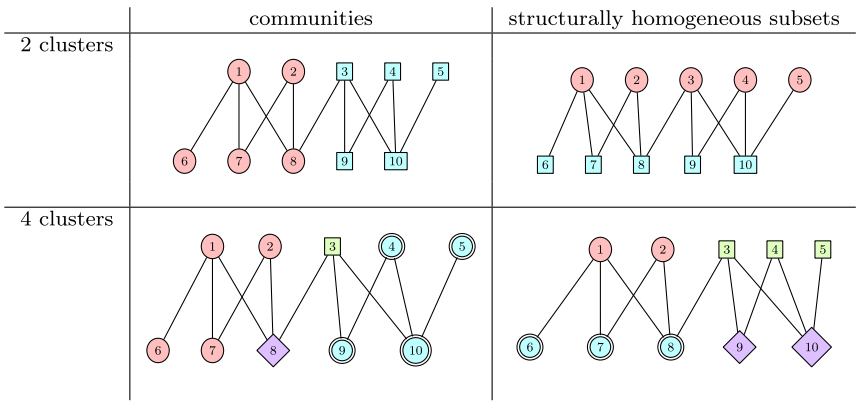


Fig. 3 Various clusters obtained for the toy-example

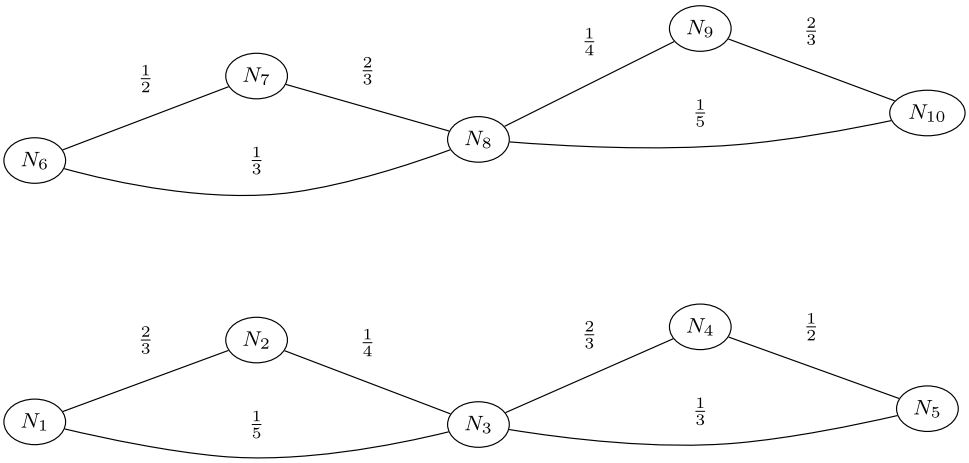


Fig. 4 The toy example graph transformed with Jaccard's measure of similarity

of the raw matrix. The same method used on the raw graph or the transformed one may give different results.

Therefore the process of clustering graphs may contain two successive steps:

1. The pre-treatment step, i.e. a transformation of the raw graph into a modified one, This step is *not* mandatory.
2. The application of a given clustering method to the modified graph.

In this paper we focus on the clustering methods, but the importance of the pre-treatment step, i.e. the transformation of the raw graph in a modified one, should not be underestimated in practice. Note that there are two types of transformation:

1. the transformation does not bring any new information concerning the vertices or the edges. In this case the transformation defines a specific similarity measure between vertices suited to answering a specific question. These transformations are generally not useful with generative statistical models that are supposed to model how the raw data have been generated. They are more often used in combination with algorithmic or optimization methods.
2. there is some more information available which is not included in the raw data W. This information may be included in the statistical model using co-variates on the edges or on the vertices. The algorithmic or optimization methods must include a pre-treatment using a transformation of W incorporating the new information.

The transformation of the raw data provides a weighted graph whose weights are a measure of similarity between each pair of vertices. Note that new edges may be produced by this procedure and old ones can be deleted. Note also that many similarity indices exist and that the similarity index should be chosen according to the scientific question. Let us consider the toy-example (Fig. 2). If the aim is to cluster the vertices with similar connectivity behavior, using the Jaccard similarity index may be a good choice. The Jaccard coefficient between two vertices i and j is the number of vertices connected to i and j divided by the number of vertices connected to i or j.

After this transformation, we obtain the following similarity matrix and the graph of Fig. 4 which has two connected components, one per type of vertex.

For the particular case of bipartite graph note that two connected components are obtained, separating the two types of vertices. This is not the general case.

$$S_J = \begin{pmatrix} - & & & & & & & & & \\ \frac{2}{3} & - & & & & & & & & \\ \frac{1}{5} & \frac{1}{4} & - & & & & & & & \\ & & \frac{2}{3} & - & & & & & & \\ & & \frac{1}{3} & \frac{1}{2} & - & & & & & \\ & & & & & - & & & & \\ & & & & & \frac{1}{2} & - & & & \\ & & & & & \frac{1}{3} & \frac{2}{3} & - & & \\ & & & & & & & \frac{1}{4} & - & \\ & & & & & & & \frac{1}{5} & \frac{2}{3} & - \end{pmatrix} \quad \text{(sym)}.$$

Note that similarity transformation can change the meaning of groups. With the Jaccard similarity index, communities in the transformed graph are structurally homogeneous subsets in the original graph.

3 Algorithmic methods

The clustering methods that do not use a statistical model may be divided into two classes: both are defined by an algorithm, but this algorithm may be designed or not to optimize a criterion. The following section presents the algorithmic methods that do not explicitly optimize any criteria.

3.1 Markov Cluster algorithm (MCL)

Synopsis	
Name	Markov Cluster algorithm (MCL)
Type of method	Algorithm
Type of graphs	Undirected,[1] weighted or not, inducing an associated ergodic Markov Chain
Type of clusters	Structurally Homogeneous Subsets
Summary	This method uses random walks on the graph and classifies in the same group the vertices whose associated random walk converge to the same state
Time complexity	The author claims a time complexity of $O(\|V\|^3)$, but this complexity is obtained by considering the number of iterations as constant

[1] The condition of ergodicity is more problematic to obtain for directed graph

The MCL algorithm (Van Dongen 2000) allows the search for structurally homogeneous subsets by considering a random walk on the graph.

A random walk on the graph is a sequence of moves at discrete time points, from one vertex to another, along graph edges. The probability of a move along an edge is proportional to its weight. Let E_{it} be the event of being in the set i in time t. For all t, $(E_{it})_{i=1,\ldots,n}$ only depends on $(E_{it-1})_{i=1,\ldots,n}$, therefore the random walk is a Markov chain. The behavior of a random walk from a starting vertex is determined by the set of probabilities of a move from this vertex to another vertex j in t steps for all (j,t). A vertex is characterized by the behavior of the random walk starting from this vertex. The main idea of this method is to consider that vertices with the same random walk behavior are in the same cluster.

A standard random walk (with no inflation factor) on a connected graph converges to the same asymptotic state of the Markov chain for any starting vertex. The objective of the MCL algorithm is to build k clusters with $k > 1$, so the usual random walk has to be modified to achieve this goal. The idea of the MCL algorithm is to constrain the random walk to converge to a different state depending on the starting vertex. This is achieved using the *inflation* operation. The more important the *inflation* operation is, the more numerous the obtained asymptotic states are. The aim of the MCL algorithm is to group vertices whose associated random walks converge to the same state.

Let the transition matrix of the Markov chain be $T = (W_{\mathrm{sl}})(D^o)^{-1}_{W_{\mathrm{sl}}}$. T_{ij} is the probability of going from vertex j to vertex i in one step. Therefore T is a column-wise stochastic matrix ($\forall j$, $\sum_i T_{ij} = 1$). Note that in MCL notation, the transition matrix is the transpose of the usual notation for the transition matrix of the Markov chain, which is a row-wise stochastic matrix. T is the matrix of probabilities of transition in one step and T^k is the matrix of probabilities of transition in k steps.

Let $T^{(1)} = T$. MCL alternates two operations indexed by k starting at $k = 1$:

1. $T^{(2k)} = (T^{(2k-1)})^{e_k}$, is the *transition* operation which allows the progress of the random walk. The importance of this operation is larger when e_k is large.
2. $T^{(2k+1)} = \Gamma_{r_k}(T^{(2k)})$ is the *inflation* operation which allows the random walk to converge toward several stable states. Γ_{r_k} is a term by term r_k power operator followed by a column sum normalization. This operation inflates the high values of the matrix $T^{(2k)}$ and reduces the small ones. For example, $\Gamma_2([.5, .3, .2]) = [.25, .09, .04]/.38 = [.66, .24, .11]$. The importance of the inflation operation is larger when r_k is large.

The algorithm ends when $T^{(k)}$ is idempotent ($T^{(k+1)} = T^{(k)}$). Denote $T^{(\infty)}$ this idempotent matrix. The columns of $T^{(\infty)}$ correspond to the vertices of the graph. Each row of $T^{(\infty)}$ defines a cluster. Non-zero values within each row indicate the composition of the cluster. In the general case, several clusters are empty. Therefore there are fewer clusters than vertices. When a vertex belongs to several clusters, different affectation rules may be applied.

The Markov chain of the random walk must be ergodic. In particular the Markov Chain must be aperiodic and irreducible. Some graphs must be modified to satisfy this aperiodicity, generally by adding self-loops. The irreducible condition is always satisfied for undirected graphs, but can be not satisfied for directed graphs. For example in a bipartite graph, when all edges connect one type of vertex to another, there is one set of absorbent states, and consequently the Markov Chain is reducible. A directed bipartite graph must be transformed (for example by symmetrization) before using the MCL algorithm.

In any case, applying the MCL method means clustering a graph which is not the true one (because of the addition of self-loops). However the true graph can be approached with a graph with very low weighted self-loops.

Figure 5 shows that the MCL algorithm applied to the toy example with unitary self-loops, retrieves communities instead of structurally homogeneous subsets. This is because the vertices with different structural connectivity behavior have the same structural behavior in the random walk when self-loops are added to the graph. To illustrate this idea, let us imagine the random walk on the graph *without* self-loops. In this case, the random walk would alternate between the two types of vertices. The algorithm would not converge and this is why the possibility of null self-loops is a priori excluded when using MCL. Nevertheless decreasing the weight (Δ) of self-loops is allowed. In the toy example

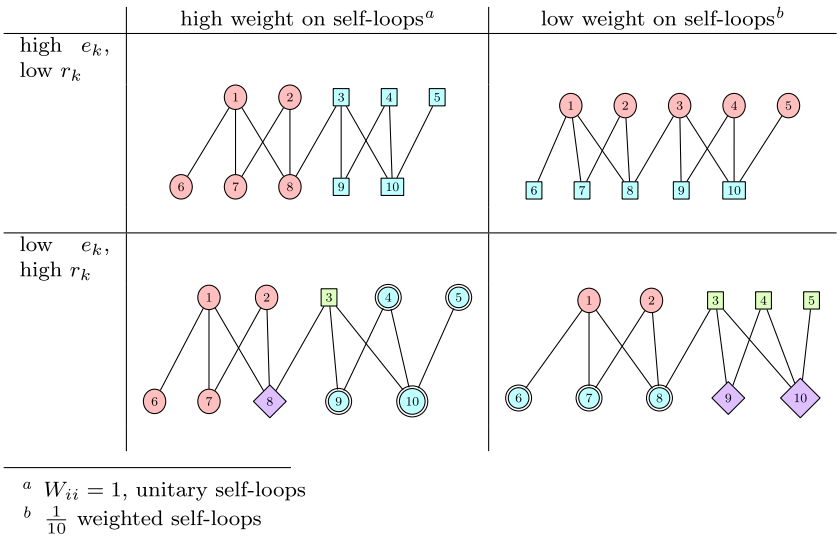


Fig. 5 MCL applied to the toy example with 4 combinations of tuning parameters

with a decreased value of the self-loops (10 times less than unitary edges), structural homogeneous subsets are obtained (see Fig. 5).

There are three tuning parameters, Δ, e_k and r_k. Their values have a great impact on the number of clusters of the final result, see Fig. 5. There are few groups when $\frac{e_k}{r_k}$ is high and many groups when $\frac{e_k}{r_k}$ is low. The author does not give any default option for the choice of e_k and r_k. After a (limited) number of empirical trials we have found that values around 2 for these two parameters could be a good choice. Δ should be small in order to capture structurally homogeneous clusters.

MCL gives satisfactory results for dense graphs, and is less efficient for sparse graphs. To our knowledge, this method has been applied mostly in the domain of molecular biology. Brohee and Van Helden (2006) found that MCL gives satisfactory results for the extraction of complexes from protein-protein interaction networks.

Pons-Latapy distance Pons and Latapy (2006) propose a distance based on a random walk on the graph. This distance is introduced for binary, undirected graphs but can be extended to the case of weighted, undirected graphs. Like MCL, this method requires the addition of self-loops if the Markov chain is not ergodic.

The method consists in stopping the random walk after a small number of steps, k. The transition matrix after k steps is T^k where T^k_{ij} is the probability of transition from vertex j to vertex i in k steps. As for MCL, a vertex is characterized by the behavior of the random walk starting at this vertex, but this method studies the behavior of a truncated random walk instead of the asymptotic behavior of a modified (by the inflation factor) random walk. Two vertices j_1, j_2 which have a similar structural behavior (in the graph with self-loops) spawn two random walks which have, for all i, a similar probability of going in k steps to the vertex i. Therefore the column vectors $T^k_{j_1}$ and $T^k_{j_2}$ are similar. To compare the structural behaviors of vertex j_1 and vertex j_2, the distance $\|T^k_{j_1} - T^k_{j_2}\|_2$ between $T^k_{j_1}$ and $T^k_{j_2}$ can be computed. The issue is that this distance is also influenced by the vertex degree because the probability of going from vertex j to vertex i is affected by the degree of vertex i (a random walk has a higher probability of going to a vertex of high degree). Therefore a re-normalization is applied to vectors $T^k_{j_1}$ and $T^k_{j_2}$ by dividing their rows by the degree of the corresponding vertices. The re-normalized vectors are $D^{-\frac{1}{2}}_{W_{\text{sl}}} T^k_{j_1}$ and $D^{-\frac{1}{2}}_{W_{\text{sl}}} T^k_{j_2}$. The Pons-Latapy distance between vertices j_1 and j_2 is defined as the distance between re-normalized vectors $D(j_1, j_2) = \|D^{-\frac{1}{2}}_{W_{\text{sl}}} T^k_{j_1} - D^{-\frac{1}{2}}_{W_{\text{sl}}} T^k_{j_2}\|_2$.

This distance is only an intermediate element in the algorithm and does not include the classification of the vertices. After computing the Pons-Latapy distances, a supplementary classification step is necessary. A hierarchical agglomerative algorithm is thus applied to the Pons-Latapy distance matrix in order to cluster the vertices of the graph.

There is a strong influence of the weight of self-loops. In the case of the toy example, the Fig. 6 shows that the results are completely changed when these weights are changed: with unitary self-loops, the distance matrix does not separate vertices which have a different structural linkage behavior. For example, N_1 and N_2 are closer to N_6, N_7, N_8, than to N_3, N_4, N_5. As with MCL, decreasing the weight of self-loops increases the ability of the method to separate vertices which have different structural behaviors.

There are three tuning parameters, k, Δ and the specific hierarchical algorithm used in the classification step, such

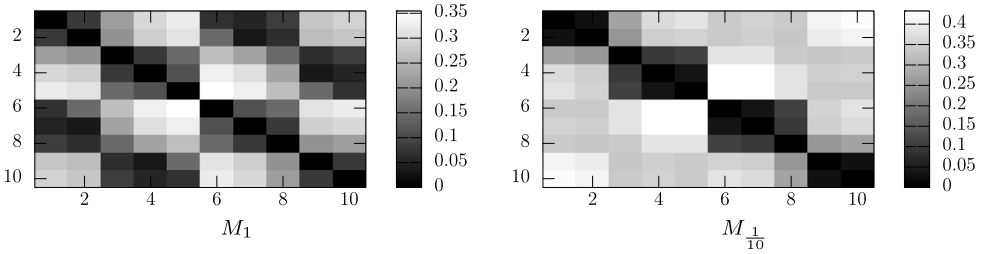


Fig. 6 Pons-Latapy distance matrices for $k = 4$ corresponding to the toy example (Fig. 2) with unitary (M_1) and $\frac{1}{10}$-weighted self-loops ($M_{\frac{1}{10}}$). Vertices are ordered as $N_1, \ldots, N_5$, $N_6, \ldots, N_{10}$

as UPGMA, Ward or maximum linkage algorithms. The authors do not give any advice about their choice. Our empirical trials suggest that $k = 3$ could be a good choice. As for MCL, Δ should be small in order to capture structurally homogeneous clusters.

3.2 Hierarchical agglomerative clustering algorithm

Synopsis	
Name	Hierarchical Agglomerative clustering algorithm
Type of method	Algorithm
Type of graphs	Graph with a dissimilarity between vertices
Type of clusters	Depends on the dissimilarity
Summary	This method groups vertices into meta-vertices recursively
Time complexity	Basically $O(\|V\|^3)$, with an additive cost in memory, a time complexity of $O(\|V\|^2 \log(\|V\|))$ can be reached

This algorithm is useful for clustering graphs once a distance between vertices has been defined. Note that we have two graphs, the original one and the weighted graph of dissimilarities. This algorithm gives communities of the graph of dissimilarities, but the clusters obtained can be structurally homogenous subsets or communities for the original graph, depending on the distance used in the algorithm. The result depends on the local or global building of the dissimilarities between vertices: for instance, if the dissimilarity between vertices is the Jaccard or Pons-Latapy (see Sect. 3.1) dissimilarity measures, then one obtains structurally homogeneous subsets. Conversely if the dissimilarity between two vertices equals one when two vertices are not linked and 0 when there is an edge between them, then one obtains communities. The usual hierarchical agglomerative algorithms are well known (Hartigan 1975): Ward, single linkage, complete linkage or UPGMA (Unweighted Pair Group Method with Arithmetic Mean). A classification algorithm such as a hierarchical agglomerative algorithm or a k-means method are the necessary final step for some methods that only compute a distance between vertices or continuous latent positions for vertices. Therefore the results of these methods depend not only on their own tuning parameter but also on the peculiar classification algorithm used to cluster the vertices.

3.3 Spectral Clustering

Synopsis	
Name	Spectral Clustering
Type of method	Algorithm
Type of graphs	Undirected, weighted or not
Type of clusters	Communities or Structurally Homogeneous Subsets
Summary	This method computes continuous latent variables using eigenvectors of the Laplacian matrix of the graph and classifies the vertices using a k-means algorithm based on the most important latent variables
Time complexity	Time complexity depends mainly on the computation of the eigen decomposition, basically $O(\|V\|^3)$

This algorithm first proposed by Donath and Hoffman (1973) allows the search for communities by considering the Laplacian matrix of the graph, $L = D_W - W$. This algorithm spawned a family of algorithms which are well described by Von Luxburg (2007). It applies only to undirected graphs, but there is some work in progress to extend it to directed graphs.

It is known that if a graph has k connected components, the Laplacian matrix has a null eigenvalue with multiplicity k (Von Luxburg 2007). Each eigenvector associated with the null eigenvalue is composed of zero and non-zero values. A non-zero value for the jth eigenvector and the row i means that vertex i is in connected component j. If the graph has k communities, the Laplacian matrix has k eigenvalues close to zero. The idea of Spectral Clustering is to determine the composition of the k communities by considering the k eigenvectors associated with the k lowest eigenvalues.

Let $L = D_W - W$ be the unnormalized Laplacian and $L_N = I - D_W^{-1/2} W D_W^{-1/2}$ the normalized Laplacian. The Spectral Clustering algorithm has several variants. The first

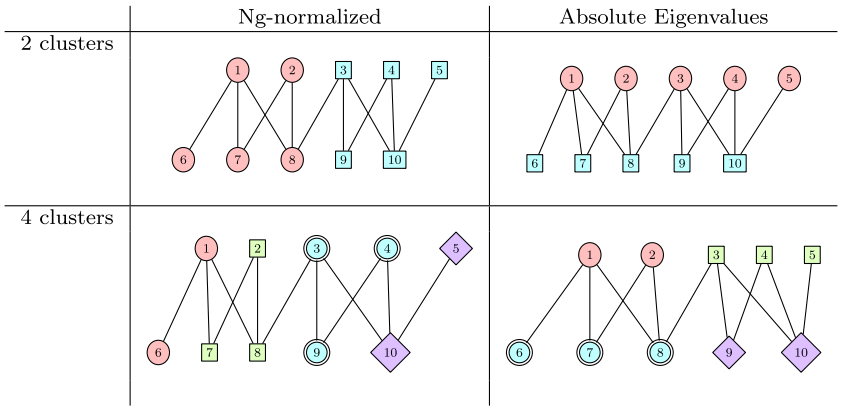


Fig. 7 Clusters obtained with Ng-normalized and Absolute Eigenvalue Spectral Clustering, with $k \in \{2, 4\}$

three are described in Von Luxburg (2007) and the last one is more recent:

1. the unnormalized Spectral Clustering computes the first k eigenvectors sorted by the eigenvalues in ascending order $U = [u_1, \ldots, u_k]$ of L,
2. the Shi-normalized Spectral Clustering computes the first k eigenvectors sorted by the eigenvalues in ascending order $U = [u_1, \ldots, u_k]$ of $D_W^{-1}L$,
3. the Ng-normalized Spectral Clustering computes the first k eigenvectors sorted by the eigenvalues in ascending order, $V = [v_1, \ldots, v_k]$ of L_N and U is the V-matrix row-norm normalized,
4. the Absolute Eigenvalue Spectral Clustering computes the first k eigenvectors sorted by the *absolute value* of eigenvalues in descending order $U = [u_1, \ldots, u_k]$ of $I - L_N$, see Rohe et al. (2011). In contrast to the other three variants, it allows the search for structurally homogeneous subsets.

Then the clusters are obtained by a k-means algorithm with k clusters on the n row vectors of matrix U. Each vertex is associated to a point in a k-Euclidean space. The coordinates of vertex i in this space are given by row i of the matrix U.

The toy example allows us to show the differences between the Absolute Eigenvalue Spectral Clustering and the Ng-normalized Spectral Clustering (Fig. 7). One can see that the first one detects the bipartite structure and the second one does not.

The tuning parameters of the Spectral Clustering algorithm are the choice of a specific method among the four (or more) possible ones, the number of latent variables and the number of clusters k.

Correspondence Analysis (CA) The CA developed by Hirschfeld (1935), Benzecri (1973) is a general method to analyze contingency tables. For undirected graphs, it can be described as a variant of Spectral Clustering considering the square of the normalized Laplacian, $L_{CA} = [D_W^{-1/2} W D_W^{-1/2}]^2$. Let k be the number of clusters. The CA clustering computes the first k eigenvectors sorted by the eigenvalues in descending order $V = [v_1, \ldots, v_k]$ of L_{CA} and $U = D_W^{1/2} W V$. Note that the eigenvalues of L_{CA} are the square of the eigenvalues of $I - L_N$ with the same associated eigenvectors. Therefore the k first eigenvectors of CA (sorted by the eigenvalues of L_{CA}) are the k first eigenvectors of $I - L_N$ sorted by the absolute values of the eigenvalues of $I - L_N$. As for Spectral Clustering, the clusters can be obtained by a k-means algorithm with k clusters on the n row vectors of matrix U. Therefore the Correspondence Analysis is equivalent to the Absolute Eigenvalue Spectral Clustering. This confirms the fact observed by Von Luxburg (2007) that many Spectral Clustering methods developed in different scientific communities are actually identical.

3.4 Edge-Betweenness

Synopsis	
Name	Edge Betweenness
Type of method	Algorithm
Type of graphs	Undirected Unweighted
Type of clusters	Communities
Summary	This method introduces a measure of the importance of a link to connect communities and it cuts edges with high values until the communities are disconnected from each other
Time complexity	No complexity is given by authors. `igraph` implementation (Csardi and Nepusz 2006) complexity is $O(\|V\|\|E\|)$

This algorithm, proposed by Girvan and Newman (2002), allows the search for communities. The main idea is to remove edges from the network until the communities are disconnected from each other. The edges to be removed are chosen as a function of a criterion called edge-betweenness.

For illustrating the concept of betweenness, let us imagine that one should go from one "side" the network to the other by following edges. An edge with a high betweenness is an edge that is included in most paths between the two "sides" of the network. For instance, on the toy example, the edge with the highest betweenness is the edge between N_3 and N_8 in Fig. 2.

More formally, the betweenness of one edge is equal to the number of shortest paths, using this edge, for all the pairs of vertices of the graph. The algorithm alternates the following steps:

1. the betweenness of all existing edges in the network is computed,
2. the edge with the highest betweenness is removed,
3. the betweenness of all edges affected by the removal is computed.

The method iterates as long as an edge remains. At the end of the algorithm, one obtains a classification tree showing the sequence of divisions of the network. Communities are obtained by truncating the classification tree.

This algorithm needs to choose the level for stopping the classification tree, which is equivalent to choosing the number of groups. Applied to the toy example, it gives the communities of Fig. 8. By definition, this algorithm cannot detect the structurally homogeneous clusters that are not communities.

4 Methods optimizing a criterion

The following section presents the algorithmic methods that explicitly optimize a criterion.

4.1 Modularity criterion

Synopsis	
Name	Modularity
Type of method	Optimization
Type of graphs	Directed or not, weighted or not
Type of clusters	Communities
Summary	This method optimizes the Modularity which represents the quality of partition as the difference between the expectation of edges inside and outside communities
Time complexity	The algorithm provided by Guimera is very time expensive and in many cases not usable in practice. An example of greedy implementation (Clauset et al. 2004) complexity is $O(\|E\|d\log(\|V\|))$, where d is the depth of the dendrogram describing the community structure

The modularity, proposed by Newman and Girvan (2004), is a global quality measure of a partition.

The modularity measures the difference between the actual and expected within-community edges relative to a null model assuming a connectivity between vertices that is proportional to their degrees. Let $C = (C_1, \ldots, C_k)$ be a partition of G. There are two equivalent definitions of the modularity of C in the undirected graph G:

1. $\mathcal{M}_C = \sum_q (e_{qq} - a_q^2)$ with $e_{ql} = \frac{1}{2m}\sum_{ij} W_{ij}\delta_q(i)\delta_l(j)$ where m is the total number of edges, $\delta_q(i) = \mathbb{1}_{i\in C_q}$ is equal to one if i is in the class q and zero if not and $a_q = \sum_l e_{ql}$,
2. $\mathcal{M} = \frac{1}{2m}\sum_{ij}(W_{ij} - \frac{(d_W)_i (d_W)_j}{2m})\delta(i,j)$ with $\delta(i,j) = \sum_q \delta_q(i)\delta_q(j)$ equal to one if i and j are in the same class.

For directed graphs the modularity is defined by: $\mathcal{M}_C = \frac{1}{2m}\sum_{ij}(W_{ij} - \frac{(d_W^o)_i (d_W^i)_j}{2m})\delta(i,j)$.

The partition with the best (maximum) modularity is obtained using an optimization algorithm such as greedy algorithms or simulated annealing algorithms. Obtaining the best partition is NP-hard.

The optimization can be done conditionally to a fixed number of groups, or not.

Guimera et al. (2010) proposed the following algorithm. Optimization is done by a Simulated Annealing (SA), with levels of temperature decreasing exponentially. Three movements are possible:

1. individual movement of a vertex from one module to another
2. merging of two modules
3. splitting of one module into two, choice of modules being made by another SA at the level of the module

This algorithm does not need any proper tuning parameter, but there are optimization parameters in the simulated annealing. This algorithm is highly computationally intensive; therefore one may have to modify some optimization parameters in order to obtain a result with a reasonable time. This algorithm, applied on the toy example, gives the communities clusters of the Fig. 9. By construction, this algorithm cannot detect the structurally homogeneous clusters that are not communities.

Since the Guimera algorithm is usable only for small graphs, greedy algorithms exist to optimize the modularity, see Clauset et al. (2004).

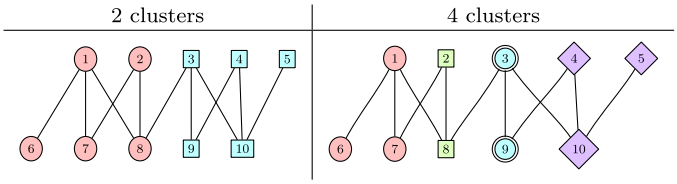


Fig. 8 Clusters obtained with Edge-Betweenness with $k \in \{2, 4\}$

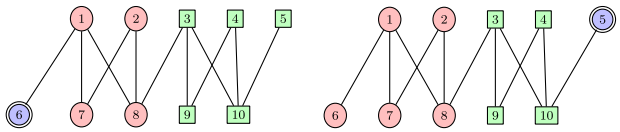


Fig. 9 Clusters obtained by maximizing the Modularity (each of them have the same modularity)

4.2 Cut

Synopsis	
Name	Cut criterion
Type of method	Optimization
Type of graphs	Undirected, unweighted
Type of clusters	Communities
Summary	This method minimizes the number of edges between communities by removing edges from the network until the communities are disconnected
Time complexity	Depends on greedy implementation

The idea (see Raj and Wiggins 2010 for a recent reference) is to suppress some edges from G to obtain an unconnected partition of vertices with a minimum modification cost. The cut between two subsets (V_1, V_2) of V from the (V, E) graph is $\text{Cut}(V_1, V_2) = \sum_{v_1 \in V_1, v_2 \in V_2} W_{v_1, v_2}$. There are three cut criteria on a partition C, the cut $\text{Cut}(C)$, the ratio cut, $\text{rCut}(C)$ and the normalized Cut, $\text{nCut}(C)$:

1. $\text{Cut}(C) = \sum_{q<l} \text{Cut}(C_q, C_l) = \frac{1}{2} \sum_{q=1}^{k} \text{Cut}(C_q, V \setminus C_q)$.
2. $\text{rCut}(C) = \sum_{q=1}^{k} \frac{\text{Cut}(C_q, V \setminus C_q)}{\sum_{i,j \in V_q} W_{ij}}$.
3. $\text{nCut}(C) = \sum_{q=1}^{k} \frac{\text{Cut}(C_q, V \setminus C_q)}{\text{Cut}(C_q, V)}$.

The partition with the best (minimum) cut is obtained using an optimization algorithm such as heuristics, greedy algorithms or simulated annealing algorithms. Obtaining the best Cut partition is NP-hard for the three criteria. In practice only approximated methods can be used.

Applied to the toy example, these methods give the community clusters of the first column of Fig. 10. By definition, these algorithms cannot detect the structurally homogeneous clusters that are not communities.

Tuning parameters are the choice of the cut criteria and the number of groups.

5 Model-based methods

Statisticians propose probabilistic models that are supposed to take into account the random variability in the data. These models are generative models in the sense that they mimic the real data generation. This section presents a synthetic summary of the more detailed review made in Daudin (2011).

All of these models use latent variables. These latent variables may be discrete and give directly the classification of the vertices such as in the Stochastic Block Model (SBM, Sect. 5.3). Alternative models such as the Model-Based Clustering for Social Networks (MBCSN, Sect. 5.1) and Random Dot Product Graph (RDPG, Sect. 5.2) use continuous latent variables; therefore a supplementary step of classification is necessary.

The above models assume that a vertex pertains to only one class, but there are alternative models such as the Continuous Stochastic Block Model (CSBM, Sect. 5.4), which allows each vertex to pertain to several classes, these models are also known under the name of Grade of Membership (see Manton et al. 1994 and Erosheva 2005).

5.1 Model-based clustering for social networks (MBCSN)

Synopsis	
Name	Model-based clustering for social networks (MBCSN)
Type of method	Model-based method
Type of graphs	Undirected and unweighted
Type of clusters	Communities
Summary	This method assumes that the graph is a realization of a generative model and infers its parameters. The model assumes that each vertex has a position in a continuous latent space, and linking behavior of each pair of vertex is determined by the distance between the vertices in the latent space
Time complexity	Not known

The model (Handcock et al. 2007) assumes that, conditionally to d-multidimensional latent variables z_i attached to the vertices and observed variables $x_{i,j}$ attached to the

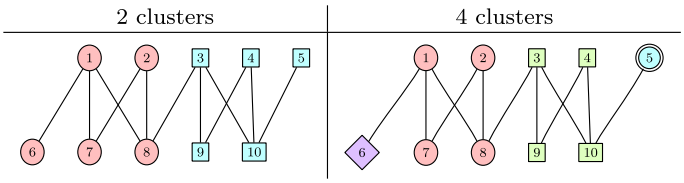


Fig. 10 Clusters obtained with Cut cost with $k \in \{2, 4\}$

edges, the $W_{i,j}$ are independent and

$$P_{ij} = P(W_{ij} = 1) = \frac{e^{\beta_0 x_{i,j} - \beta_1 |z_i - z_j|}}{1 + e^{\beta_0 x_{i,j} - \beta_1 |z_i - z_j|}}.$$

The probability of connection between vertices i and j is greater for vertices whose latent variable values are similar. The distribution of the latent variables is a mixture of k multivariate Gaussian distribution. The parameters of the mixture model and β_0 and β_1 are estimated with Bayesian or Maximum-Likelihood methods using MCMC and the values of the latent variables are predicted for each vertex. An R Package *latentnet* is available. Applied to the toy example with no covariate $X_{i,j}$ on the edges, $d = 2$ and $k = 2$, the package *latentnet* gives the community clusters of the top left corner of Fig. 2. With $d = 4$ and $k = 4$, one obtains only two clusters that are the same ones as with $k = 2$ and $d = 2$.

Tuning parameters are d the dimension of the latent space, and k the number of distributions which is the number of wanted groups k.

5.2 Random Dot Product Graphs (RDPG)

Synopsis	
Name	Random Dot Product Graphs (RDPG)
Type of method	Model-based method
Type of graphs	Undirected and unweighted
Type of clusters	Communities
Summary	This method assumes that the graph is the realization of a generative model and infers its parameters. The model assumes that the vertices lie in a continuous latent space and the linking behavior of each pair of vertices is determined by the position of the vertices in the latent space. Then a classification method such as the k-means algorithm classifies the vertices
Time complexity	Not known

The multidimensional scaling (MS) method, applied to the similarity matrix P, consists in positioning each vertex in a metric space of latent variables so that the similarity between vertices is approximately kept. The underlying model is $P = TT'$, where the (n, d)-matrix T contains the coordinates of the vertices in a d-dimensional metric space. The naive MS method is not well suited for modeling P, with two major drawbacks: TT' does not lie in $[0, 1]^{n^2}$ if $T \in \mathbb{R}^d$ and TT' is symmetric so it is not suited for the modeling of directed graphs.

The Random Dot Product Graph defined in Marchette and Priebe (2008) is

$$P_{ij} = f\left(t_i' t_j\right) \quad \text{with } t_i \in \mathbb{R}^d \text{ and } f(x) \in [0, 1].$$

f is a simple threshold in Marchette and Priebe (2008): $f(x) = 0$ if $x < 0$, $f(x) = x$ if $0 \leq x \leq 1$ and $f(x) = 1$ if $x > 1$.

To get around the second drawback, the RDPG model is extended with two vectors for each vertex, an in-vector V and an out-vector U, such that the model becomes $P_{ij} = f(u_i'.v_j)$. Another way to get around the symmetry of P, called DEDICOM, was proposed by Harshman (1978) and well described in Trendafilov (2002). This model uses only one vector for each vertex but inserts a non-symmetric (d, d)-matrix A in the dot product. The model is

$$X = TAT' + E$$

the matrix T is constrained by $T'T = I$ and T and A are obtained by minimizing $\|X - TAT'\|^2$. Several algorithms have been proposed to achieve this task (see Kiers et al. 1990).

Tuning parameters are d the dimension of the latent space, and the number of groups.

5.3 Stochastic Block Model (SBM)

Synopsis	
Name	Stochastic Block Model
Type of method	Model-based method (SBM)
Type of graphs	Directed or not, weighted or not
Type of clusters	Structurally Homogeneous Subsets
Summary	SBM is a mixture model where each vertex is supposed to pertain to only one structurally homogeneous subset. The assignment of vertices to subsets is done by inferring the model parameters
Time complexity	Each iteration of VEM is $O(\|V\|^2)$. The number of iterations depends on the number of nodes. For sparse graphs the inference can be made in $O(\|E\|)$

The first probabilistic model which explicitly integrates heterogeneity in the network topology, the Stochastic block model, has been proposed by mathematicians and statisticians working in the domain of social science such as White et al. (1976), Holland et al. (1983) and Snijders and Nowicki (1997). These authors have developed this model in concordance with the notion of Structural Equivalence in a graph. Therefore the SBM is built to detect structurally homogeneous subsets.

The intuitive idea developed by White et al. (1976) (see also Arabie et al. 1978 and Winship and Mandel 1983) is that the vertices of a graph may be classified into groups. Two vertices of the same group are connected in the same way to the other vertices. Therefore the adjacency matrix, sorted by the number of the group in row and column, appears to be partitioned in homogeneous blocks composed of 0 or alternatively of 1.

BLOCKER and CONCOR (White et al. 1976) were historically the first algorithms for clustering vertices. More recently Snijders and Nowicki (1997) used the Markov Chain Monte Carlo method for estimating the parameters.

The modern version of the Stochastic Block Model is a mixture model, using discrete latent variables giving the assignment of each vertex to a group, where each vertex is supposed to pertain to only one group. The model for a binary directed network is the following:

$i = 1, n$ vertices pertains to $q = 1, k$ classes. The class of each vertex is defined by a hidden discrete latent variable $Z_i = q$, if vertex i pertains to class q, with Probability Distribution Function (pdf) given by $Z_i \sim \mathcal{M}(1, \alpha_1, \alpha_2, \ldots, \alpha_k)$ and $\mathcal{M}$ is a multinomial pdf.

$W_{ij} = 1$ if there is an edge from vertex i to vertex j and 0 if there is no edge, and conditionally to Z, W_{ij} are independent Bernoulli random variables with

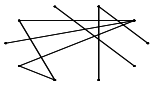
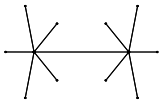
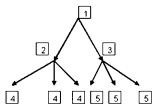
$$P(W_{ij} = 1/Z_i = q, Z_j = l) = \pi_{ql}.$$

Table 1 shows that the model is very flexible for it is able to modelize hubs, communities or hierarchical structures.

Sinkkonen et al. (2008b) propose an alternative mixture model which allows the analysis of large graphs. This model uses latent variables which operate not on the vertex level but on the edge level.

Daudin et al. (2008) used a variational method of estimation allowing the analysis of network up to 3000 vertices. Hofman and Wiggins (2008) use a Bayesian variational approach for a particular case of SBM with two parameters, $\pi_{q,q} = a$ and $\pi_{q,l} = b$ for $q \neq l$. Mariadassou et al. (2010) have extended the variational estimation method to weighted networks with probability distribution of the weights pertaining to the exponential family. Identifiability and consistency results have been obtained by Celisse et al. (2012), Rohe et al. (2011), Ambroise and Matias (2011), Bickel and Chen (2010) and Choi et al. (2012). A frequent criticism against SBM is that the number of clusters is not a fixed number for real-life networks and large networks have more clusters than small ones. However this point has been taken into account and some consistency results are obtained in an asymptotic framework that allows the number of clusters to increase with the number of vertices.

Table 1 Examples of SBM

Description	Graph	k	$\boldsymbol{\pi}$
Erdos		1	p
Hubs		4	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Communities		2	$\begin{pmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{pmatrix}$
Hierarchical		5	$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

There is a C-Package *Mixnet* that uses the variational method. Applying this package to the toy example allows the retrieval of the subsets of the column "Structurally homogeneous subsets" in Fig. 3.

For sparse graphs, Decelle et al. (2011) use belief propagation to infer parameters of the stochastic block model, and achieve a complexity linear to the number of edges.

There is only one tuning parameter, the number of clusters, k. It is possible to infer it from the data, see Daudin et al. (2008) and Hofman and Wiggins (2008), using a penalized or a Bayesian criterion. Sinkkonen et al. (2008a) presents Bayesian non-parametric methods which address the criticism of a fixed number of clusters.

5.4 Continuous Stochastic Block Model (CSBM)

Synopsis	
Name	Continuous Stochastic Block Model (CSBM)
Type of method	Model based method
Type of graphs	Directed or not, weighted or not
Type of clusters	Structurally Homogeneous Subsets
Summary	This method assumes that the graph is a realization of a generative model and infers its parameters. The model assumes that each vertex is a mixture of virtual vertices whose connectivity properties are known
Time complexity	Not known

The Stochastic Block Model can be written under the form

$$P_{ij} = P(W_{ij} = 1) = \sum_{q,l=1,k} z_{iq} a_{ql} z_{jl}$$

where $z_{iq} = 1$ if the vertex i is in class q, and 0 if not, which gives the matrix relation $P = ZAZ'$, with Z the (n, k)-matrix containing the z_{iq}. If we allow z_{iq} to be in $[0, 1]$ (and not in $(0, 1)$) then each vertex does not pertain to only one group, which bears more flexibility to the model. This leads to the CSBM (Continuous-SBM) developed in Daudin et al. (2010).

This model displays the vertices in a continuous space. Therefore a supplementary step of clustering must by applied for obtaining groups. There is a MATLAB package *C-Mixnet* for this model. Applying this package, followed by a k-means clustering, to the toy example allows to retrieve the subsets of the column "Structurally homogeneous subsets" in Fig. 3.

There is only one tuning parameter, the number of clusters, k. However it is possible to infer it from the data, see Daudin et al. (2008).

Other models allow each vertex to pertain to several classes such as the Mixed Membership Stochastic Block Model (Airoldi et al. 2008) and the Overlapping Stochastic Block Model (Latouche et al. 2011).

6 Application to the Zachary's Karate Club

The Karate Club network, introduced by Zachary (1977) is one of the most famous data set from the social science literature. The members of a karate club at a US University in the 1970s are the vertices of the network. Edges represent friendship relations between the members. This example is highly interesting because shortly after the observation, the club was split into two components. The resulting two groups are represented in Fig. 11. Therefore one may easily compare the groups obtained by any clustering method to the exact split. A close examination of the graph shows that some members are highly connected to other members. These members, such as numbers 1 for the group on the left side of Fig. 11 and 33, 34 for the other group have probably played a leading role in the split. Therefore one may consider that there are four groups: the leadings members and the satellite members of each group. Therefore we have decided to show the results of each method with two and four groups.

The results of eight methods (EB, Pons-Latapy, Modularity, MCL, unnormalized SC, Absolute SV, SBM and CSBM) are given in Figs. 12–18 presenting the graph with vertices distinguishable by colors and shapes. Each color corresponds to a cluster obtained by the method. For all the methods excepted MCL we give the result with two groups

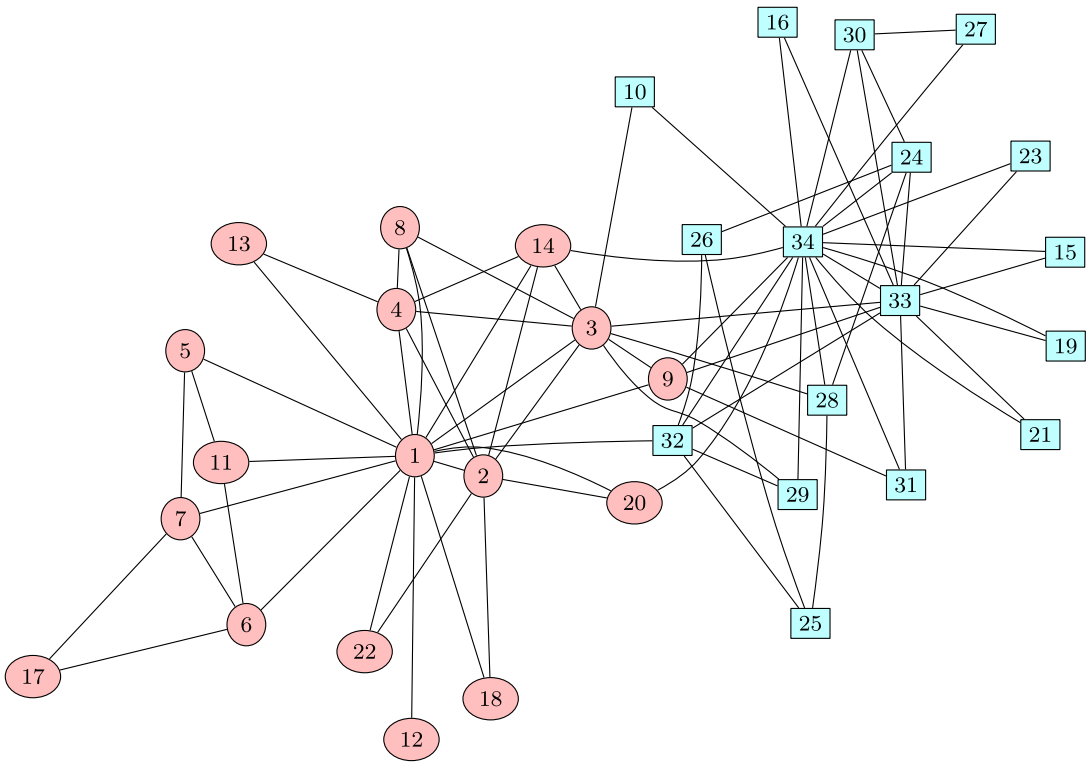


Fig. 11 Zachary's Karate Club network. Colors and shapes show real fission of the club

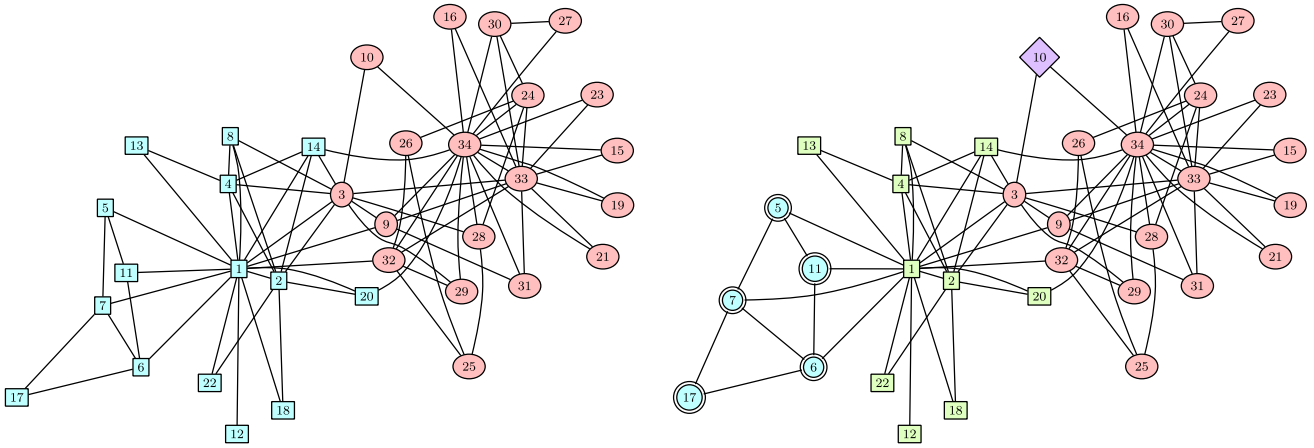


Fig. 12 Edge-Betweenness for two and four groups method applied to the Zachary's Karate Club network

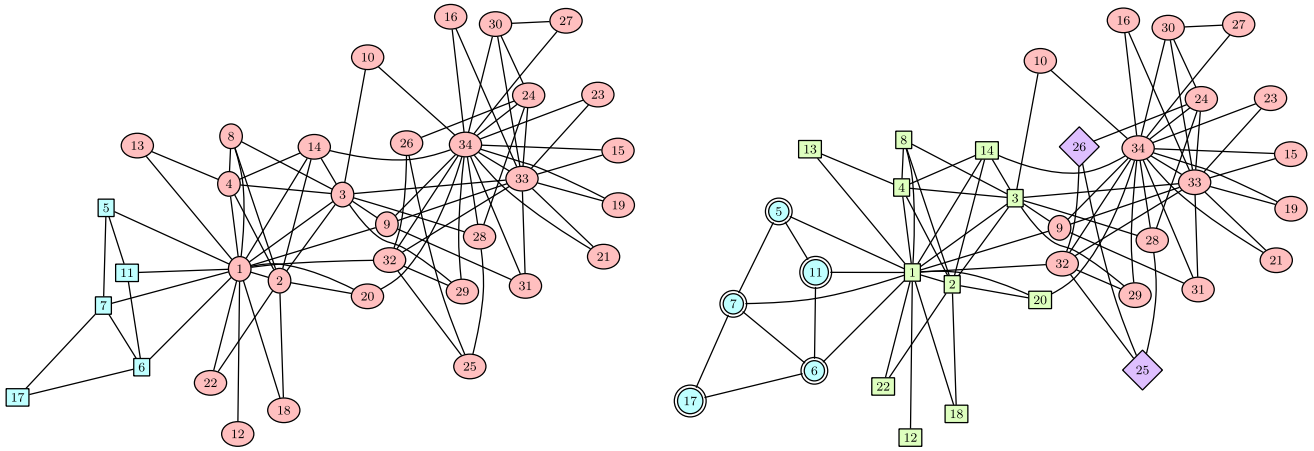


Fig. 13 Hierarchical Clustering method to Pons-Latapy distance for two and four groups applied to the Zachary's Karate Club network

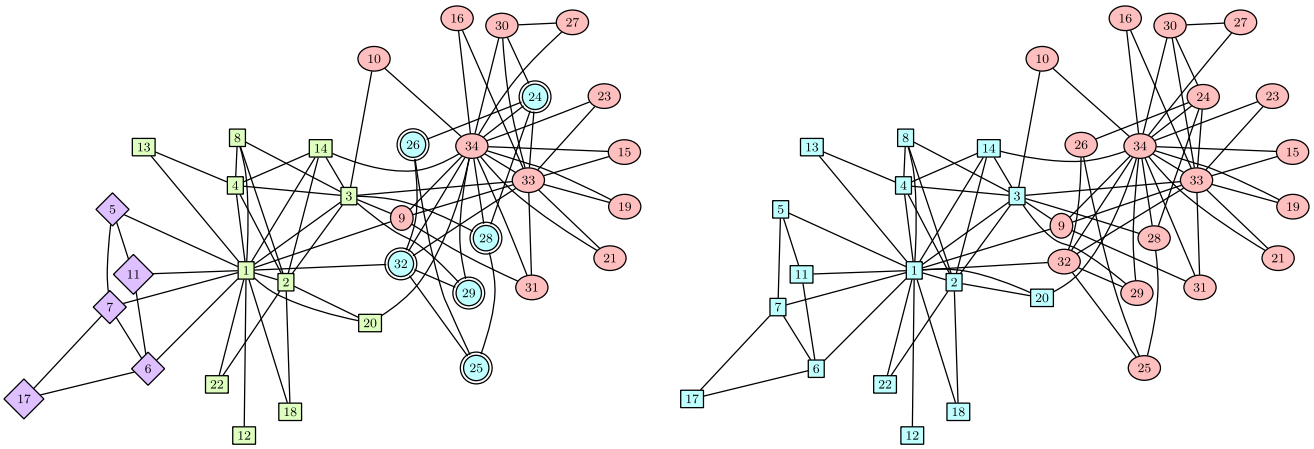


Fig. 14 Modularity method without choice of number of groups (*left*) and MCL method without choice of number of groups (*right*) applied to the Zachary's Karate Club network

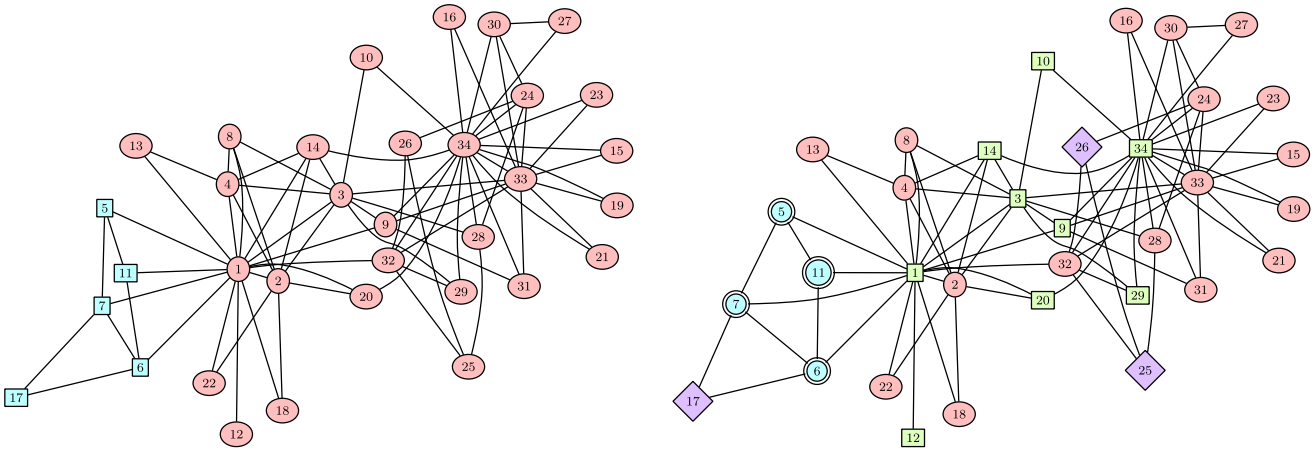


Fig. 15 Spectral Clustering (unweighted variant) method for two and four groups applied to the Zachary's Karate Club network

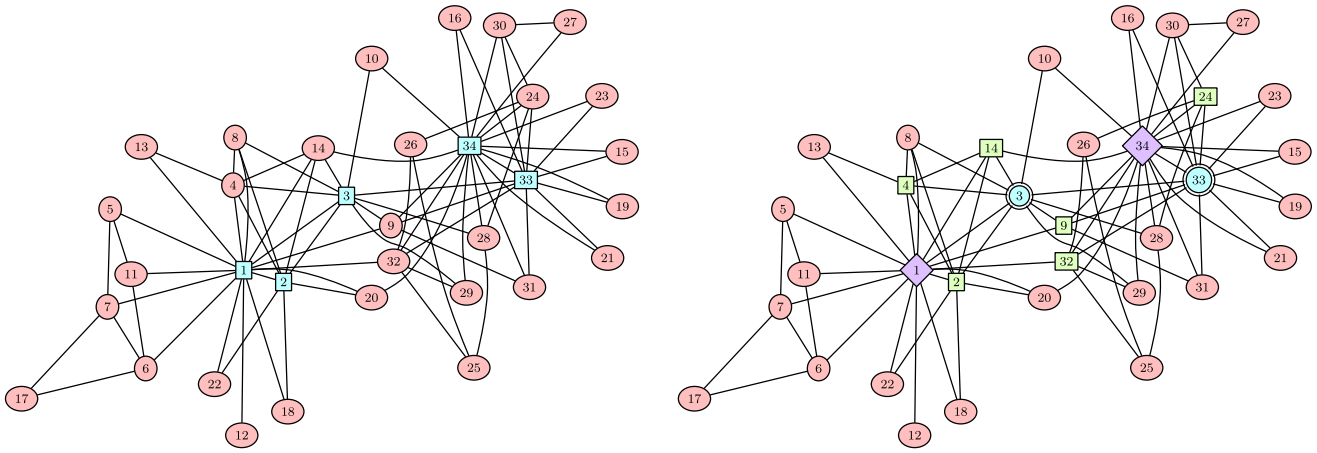


Fig. 16 Spectral Clustering (Absolute Eigenvalues variant) method for two and four groups applied to the Zachary's Karate Club network

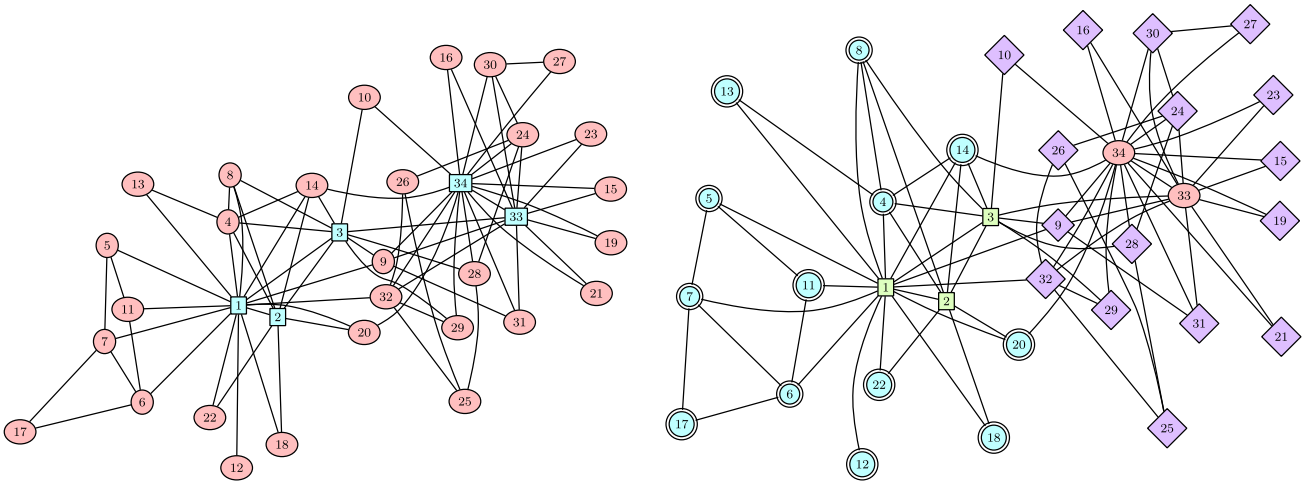


Fig. 17 Stochastic Block Model method applied for two and four groups applied to the Zachary's Karate Club network

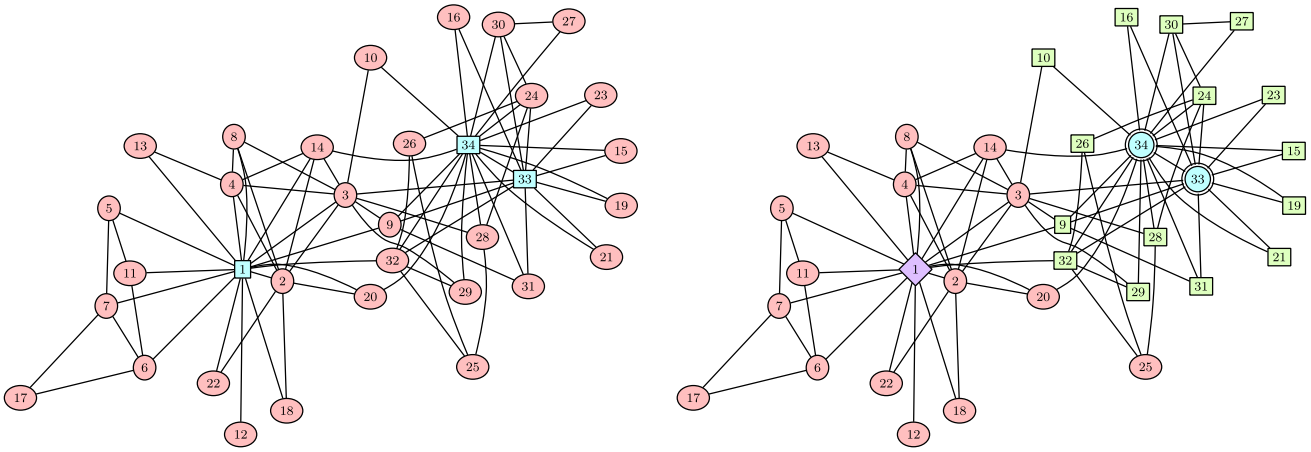


Fig. 18 Continuous Stochastic Block Model method for two and four groups applied to the Zachary's Karate Club network

(on the left side of the figure) and with four groups (on the right side).

In the case of two groups there are three classes of results:

1. The real split with one (MCL) or two (EB) errors of classification.
2. A somewhat isolated subgroup from the left group (members 5, 6, 11, 16 and 17), connected to no other member excepted member 1 on one side and all the other members on the other side (Pons-Latapy, SC unnormalized).
3. A group composed of high degree vertices (members 1, 2, 3, 33, 34) and a group composed of low degree vertices (SC absolute, SBM, CSBM).

In the case of four groups there are three classes of results:

1. EB and Modularity split the left group in one subgroup composed of the isolated subgroup (members 5, 6, 11, 16 and 17), and split the right group in two subgroups,
2. SBM and CSBM give the leadings members and the satellite members of each group, with one error of classification (member 9),
3. SC absolute gives groups separated by connectivity behavior, in leading behavior classes and satellite behavior classes.

This simple comparison from one small data set cannot be taken as a benchmark, but is only an illustration of two points:

1. the results may be very different from one method to another.
2. the result obtained with a given method are coherent with its objective. EB, Modularity and MCL (with $\frac{1}{10}$-weighted self-loops) find communities and SC (absolute), SBM and CSBM find structurally homogeneous subsets.

7 Conclusion

There are mathematical relations between some of the methods:

1. The Markov Chain Clustering and the Spectral Clustering are two ways to study the behavior of the Markov Chain associated with a random walk along the graph. This behavior is controlled by the transition matrix and the asymptotic behavior of its power. The power of a matrix is related to its spectral decomposition which is studied in the Spectral Clustering method. Therefore the two methods are linked even if they do not necessarily give exactly the same results. Von Luxburg (2007) gives a detailed analysis of the connections between Spectral Clustering and Random Walks.
2. The Spectral Clustering and the cut-methods are also linked by the following relations: let x be a vector with $x_i = 1$ if $i \in V_1$ and $x_i = -1$ if $i \in V_2$. Then $\mathrm{Cut}(V_1, V_2) = \frac{1}{2}x'Lx$. Von Luxburg (2007) also gives a detailed analysis of the connections between spectral clustering and Cut criteria.
3. Rohe et al. (2011) proved that for undirected graphs, the Absolute Eigenvalue Spectral Clustering is asymptotically able to approximately retrieve the clusters if the data are generated by SBM. Some work is in progress from the same authors to extend these results to directed graphs.

We can summarize the methods in Table 2. They are sorted by ascending level of generality. The first three methods cannot retrieve structural homogeneous subsets of vertices that are not communities. These three methods are devoted to one objective, the detection of communities and it seems difficult to generalize them to a more general objective. On the other hand the SBM model, which has been built

Table 2 Summary of the clustering methods

Method	Type of method[a]	Directed[b]	Weighted[c]	Goal[d]	Tuning parameters
Edge-Betweenness	A	N	N	C	none
Cut	O	N	Y	C	Criteria
Modularity	O	Y	Y	C	none
Spectral Clustering	A	N	Y	C or SHS	method, k[e]
Hierarchical Clustering	A	N	Y	C or SHS	method
Markov Chain Clustering	A	Y	Y	SHS	r[f], e[f], Δ[g]
Pons-Latapy	A	N	Y	SHS	k[e], Δ[g]
SBM	M	Y	Y	SHS	k[e] or none
CSBM	M	Y	N	SHS	k[e] or none
MBCSN[h]	M	N	N	C	d[i] and k[e]
RDPG	M	N	N	C	d[i]

[a]A for algorithm, O for optimization, M for probabilistic model

[b]Y if the method can be applied to a directed graph, N otherwise

[c]Y if the method can be applied to a weighted graph, N otherwise

[d]C for Community research algorithm, SHS for Structural homogeneous subset research algorithm

[e]k is the number of groups

[f]e and r are the importance of transition and inflation step, $\frac{e}{r}$ control the number of groups

[g]Weight of self-loops added for ergodicity

[h]Model-based clustering for social network

[i]d is the dimension of the latent space

around the concept of structural equivalence, is more general for it can detect every type of structurally homogeneous subset. Spectral Clustering and Random Walk methods have been used for a long time to detect communities. However these methods may be customized to be able to detect structurally homogeneous subsets as well (Rohe et al. 2011). The trick for random walk methods consists in decreasing the value of self-loops. The modification of the usual Spectral Clustering consists in keeping not only the eigenvectors corresponding to the higher eigenvalues but also the ones corresponding to the negative eigenvalues that have a high absolute value, see Rohe et al. (2011). It is quite surprising that this new method is equivalent to a very old one, Correspondence Analysis.

Comparing methods to relevant datasets of networks is not a trivial task, many points must be considered. To have a reference partition, simulated networks can be a solution. However, the simulation method must be carefully chosen not to advantage a subset of methods, and in particular, must be different from generative models used by some methods. On the other hand, real world networks, in general case, do not provide any reference partition. A comparison of methods in a particular case of simulated bipartite networks is under work.

References

Airoldi, E., Blei, D., Fienberg, S., Xing, E.: Mixed membership stochastic blockmodels. J. Mach. Learn. Res. **9**, 1981–2014 (2008)

Ambroise, C., Matias, C.: New consistent and asymptotically normal parameter estimates for random-graph mixture models. J. R. Stat. Soc., Ser. B, Stat. Methodol. **74**, 3–35 (2011)

Arabie, P., Boorman, S., Levitt, P.: Constructing blockmodels: how and why. J. Math. Psychol. **17**, 21–63 (1978). doi:10.1073/pnas.0907096106

Benzecri, J.: L Analyse des Donnees. Volume II. L Analyse des Correspondances. Dunod, Paris (1973)

Bickel, P., Chen, A.: A nonparametric view of network models and Newman-Girvan and other modularities. Proc. Natl. Acad. Sci. USA, 1–6 (2010)

Brohee, S., Van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinform. **7**(1), 488 (2006)

Burt, R.: Cohesion versus structural equivalence as a basis for network subgroups. Sociol. Methods Res. **7**(2), 189–212 (1978)

Celisse, A., Daudin, J., Pierre, L.: Consistency of maximum likelihood and variational estimators in mixture models for random graphs. Electron. J. Stat. **6**, 1847–1899 (2012)

Choi, D., Wolfe, P., Airoldi, E.: Stochastic blockmodels with growing number of classes. Biometrika **99**(2), 273–284 (2012)

Clauset, A., Newman, M., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)

Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal, Complex Syst. **1695**, 38 (2006). http://igraph.sf.net

Daudin, J.: A review of statistical models for clustering networks with an application to a PPI network. J. Soc. Fr. Stat. **152**(2), 111–125 (2011)
Daudin, J., Picard, F., Robin, S.: A mixture model for random graphs. Stat. Comput. **18**(2), 173–183 (2008)
Daudin, J.J., Pierre, L., Vacher, C.: Model for heterogeneous random networks using continuous latent variables and an application to a tree-fungus network. Biometrics **66**(4), 1043–1051 (2010)
Decelle, A., Krzakala, F., Moore, C., Zdeborová, L.: Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Phys. Rev. E **84**(6), 066106 (2011)
Donath, W.E., Hoffman, A.J.: Lower bounds for the partitioning of graphs. IBM J. Res. Dev. **17**(5), 420–425 (1973)
Erosheva, E.: Comparing latent structures of the grade of membership, Rasch and latent class model. Psychometrika **70**(4), 619–628 (2005)
Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010). http://www.sciencedirect.com/science/article/pii/S0370157309002841. doi:10.1016/j.physrep.2009.11.002
Girvan, M., Newman, M.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA **99**(12), 7821 (2002)
Guimera, R., Stouffer, D., Sales-Pardo, M., Leicht, E., Newman, M., Nunes Amaral, L.: Origin of compartmentalization in food webs. Ecology (2010). http://www.esajournals.org/doi/abs/10.1890/09-1175.1. doi:10.1890/09-1175.1
Handcock, M.S., Raftery, A.E., Tantrum, J.: Model-based clustering for social networks. J. R. Stat. Soc. A **170**(2), 301–354 (2007)
Harshman, R.: Models for analysis of asymmetrical relationships among *N* objects or stimuli. In: First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology. McMaster University, Hamilton, Ontario, August (1978)
Hartigan, J.: Clustering Algorithms. Wiley, New York (1975)
Hirschfeld, H.: A connection between correlation and contingency. Proc. Camb. Philos. Soc. **31**, 520–524 (1935)
Hofman, J.M., Wiggins, C.H.: Bayesian approach to network modularity. Phys. Rev. Lett. **100**, 258701 (2008). http://link.aps.org/doi/10.1103/PhysRevLett.100.258701. doi:10.1103/PhysRevLett.100.258701
Holland, P., Laskey, K., Leinhardt, K.: Stochastic blockmodels: some first steps. Soc. Netw. **5**, 109–137 (1983)
Kiers, H., ten Berge, J., Takane, Y., de Leeuw, J.: A generalization of Takane's algorithm for DEDICOM. Psychometrika **55**(1), 151–158 (1990)
Latouche, P., Birmelé, E., Ambroise, C.: Overlapping stochastic block models with application to the French political blogosphere. Ann. Appl. Stat. **5**(1), 309–336 (2011)
Lorrain, F., White, H.: Structural equivalence of individuals in social networks. J. Math. Sociol. **1**, 49–80 (1971)
Manton, K., Woodbury, M., Tolley, H.: In: Statistical Applications Using Fuzzy Sets (1994)
Marchette, D., Priebe, C.: Predicting unobserved links in incompletely observed networks. Comput. Stat. Data Anal. **52**(3), 1373–1386 (2008)
Mariadassou, M., Robin, S., Vacher, C.: Uncovering latent structure in valued graphs: a variational approach. Ann. Appl. Stat. **4**, 715–742 (2010)
Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
Picard, F., Miele, V., Daudin, J.J., Cottret, L., Robin, S.: Deciphering the connectivity structure of biological networks using MixNet. BMC Bioinform. **10**, S7 (2009)
Pons, P., Latapy, M.: Computing communities in large networks using random walks. J. Graph Algorithms Appl. **10**(2), 191–218 (2006)
Raj, A., Wiggins, C.H.: An information-theoretic derivation of min-cut based clustering. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 988–995 (2010). doi:10.1109/TPAMI.2009.124
Rohe, K., Chatterjee, S., Yu, B.: Spectral clustering and the high-dimensional stochastic block model. Ann. Stat. **39**(4), 1878–1915 (2011)
Sinkkonen, J., Aukia, J., Kaski, S.: Component models for large networks (2008a). arXiv:0803.1628
Sinkkonen, J., Aukia, J., Kaski, S.: Inferring vertex properties from topology in large networks (2008b). arXiv:0803.1628v1 [stat.ML]
Snijders, T., Nowicki, K.: Estimation and prediction for stochastic blockmodels for graphs with latent block structure. J. Classif. **14**(1), 75–100 (1997)
Trendafilov, N.: GIPSCAL revisited. A projected gradient approach. Stat. Comput. **12**(2), 135–145 (2002)
Van Dongen, S.: Graph clustering by flow simulation. University of Utrecht 275 (2000)
Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
White, H.C., Boorman, S.A., Breiger, R.L.: Social structure from multiple networks. Am. J. Sociol. **81**, 730–780 (1976)
Winship, C., Mandel, M.: Roles and positions: a critique and extension of the blockmodeling approach. In: Sociological Methodology (1983)
Zachary, W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**(4), 452–473 (1977)