Overlapping clustering methods for networks

Pierre Latouche, Etienne E. Birmelé, Christophe Ambroise

1

Overlapping clustering methods for Networks

Pierre Latouche

Laboratoire SAMM, Université Paris 1 Panthéon-Sorbonne, France

Etienne Birmelé

Laboratoire Statistique et Génome, Université d'Évry-val-d'Essonne, France
Laboratoire Biométrie et Biologie Evolutive, INRIA Rhône-Alpes, Lyon, France

Christophe Ambroise

Laboratoire Statistique et Génome, Université d'Évry-val-d'Essonne, France

CONTENTS

1.1 Introduction

Because networks are a straightforward formalism for representing interactions between objects of interest, they are used in many scientific fields. For instance, in Biology, regulatory networks allow to describe the regulation of gene expression through transcriptional factors [40], while metabolic networks focus on

representing pathways of biochemical reactions [36]. Besides, the binding procedures of proteins are often described as protein-protein interaction networks [3, 10]. In social sciences, networks are widely used to represent relational ties between actors [55, 47, 48]. Other examples of networks are powergrids [57] and the world wide web [59].

As a network describes the presence or absence of links between objects, the notion of groups of nodes having a similar behavior naturally arises. In some cases, this notion of similarity is even the process from which the network originates. Common affinities will for instance lead to edges in social networks, whereas gene duplication is the main growth process of protein-protein interaction networks.

The most widely assumed group structure is the partition, where each node does belong to only one group. When dealing with real world applications, this assumption of empty intersections between groups is often too rigid. For instance, so-called *moonlighting proteins* are known to have several functions in the cells [34]. Considering social networks, it seems also obvious that actors might belong to several groups of interests [49]. Thus, exploring structures which allow more complex membership for each node is thus of great practical interest. One possibility consists in considering fuzzy clustering, where each node is allowed to belong to all groups with a fuzzy membership coefficient. Fuzzy clustering assumes that each individual (node) to classify, has its membership coefficients summing to one. This approach is for example the one developed in the Latent Dirichlet Allocation [12] or the Mixed Membership Stochastic Block model [2]. A less stringent alternative consist in considering that each individual belongs to each group entirely or not at all.

In this chapter, we propose to give an overview of the methods using the latter approach, that is retrieving group memberships of nodes based on their connectivity pattern, the memberships of each node being summarized in a $\{0,1\}$-vector. The first section introduces the notion of network and the characteristics of real networks one should have in mind when building models. The second section deals with the partitioning of nodes, that is methods assigning each vertex to exactly one group. The last section presents generalizations of those methods which allow overlapping groups of nodes.

1.2 Networks and their characteristics

1.2.1 Network representations

A network is commonly represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of N vertices and $\mathcal{E}$ is a set of edges between pairs of vertices. The graph is said to be directed (Figure 1.1) if the pairs (u, v) in $\mathcal{E}$ are ordered. Conversely, unordered pairs form an undirected graph (Figures 1.2 and 1.3). Note that

the edges can be weighted by a function $w : \mathcal{E} \to \mathbb{F}$ for any set $\mathbb{F}$. However, we will concentrate only on binary graphs, that is $\mathbb{F} = \{0, 1\}$. The size of $\mathcal{G}$ is then given through the edge count $m = |\mathcal{E}|$. The graph is said to be dense if m is close to the maximal number M of edges whereas a low value of m leads to a sparse graph. To characterize the density of $\mathcal{G}$, a criterion $\delta(\mathcal{G})$ is often used. It is defined as the ratio of the number m of existing edges over the number M of potential edges:

$$\delta(\mathcal{G}) = \frac{m}{M}.$$

For a directed graph, $M = N^2$ while $M = N(N+1)/2$ otherwise. If $\mathcal{G}$ does not contain any self loop, that is an edge from a vertex to itself, then $M = N(N-1)$ for a directed graph and $M = N(N-1)/2$ otherwise.

The neighbourhood $N_{\mathcal{G}}(u)$ of vertex u is defined as the set of all the vertices connected to u. Its degree $d(u)$ is equal to its number of incident edges. Finally, a path from a vertex u to a vertex v is a sequence of edges in $\mathcal{E}$ starting at vertex $v_0 = u$ and ending at vertex $v_{k+1} = v$:

$$\{u, v_1\}, \{v_1, v_2\}, \ldots, \{v_k, v\}.$$

If there exists at least one path between every pair of vertices then the graph is said to be connected. For instance, the graph in Figure 1.1 is connected contrary to the graphs in Figures 1.2 and 1.3 which have some isolated vertices.

A network can equivalently be represented by a so-called adjacency matrix $\mathbf{X}$, which describes the presence or absence of an edge in a graph. As mentioned already, we focus on binary graphs and therefore $\mathbf{X}$ is in $\{0, 1\}^{N \times N}$. Thus, if there exists an edge from vertex i to vertex j then X_{ij} equals 1 and 0 otherwise. If the network is undirected, the matrix is symmetric, i.e. X_{ij} and X_{ji} are equals. Non-zero entries of the diagonal correspond to self-loops. Every property of a graph can be interpreted in terms of its adjacency matrix. The degree of a vertex is for instance the sum of the row or the column corresponding to it, or the fact that two vertices (i, j) are in different connected compounds is equivalent to $(X^k)_{ij} = 0$ for all power $1 \leq k \leq N$.

1.2.2 Properties of real networks

Very interestingly, most real networks have been shown to share some properties [4, 15, 21, 8, 56] that we briefly recall in the following.

- **Sparsity:** The number of edges is linear in the number of vertices. In other terms, the mean degree remains bounded when N grows, implying that the density tends to 0.
- **Existence of a giant component:** Real networks are often disconnected. However, a majority of the vertices are contained in a same component, the other components being significantly smaller.

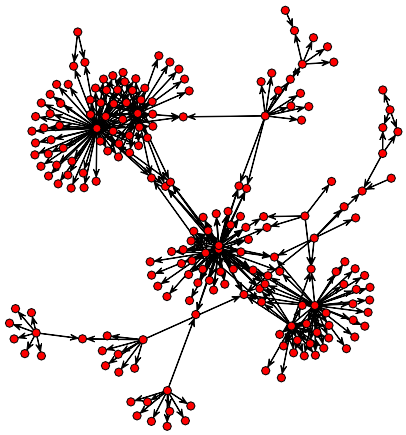

FIGURE 1.1
Subset of the yeast transcriptional regulatory network [40]. Nodes of the directed network correspond to genes, and two genes are linked if one gene encodes a transcriptional factor that directly regulates the other gene.

- **Degree heterogeneity:** A few vertices have a lot of connections while most of the vertices have very few links. The degrees of the vertices are sometimes characterized using a scale free distribution [for instance see 9].
- **Small world:** The shortest path from one vertex to another is generally rather small, typically of size $O(\log N)$.

All the properties listed above can be verified through easy computable statistics which are the degrees and the paths of length at most N. As they are key properties in the interpretation of real network behaviors with respect to information diffusion [51] or attack tolerance [5], they have to be taken into account when proposing random graph models to describe networks.

Most of the real networks exhibit another property, which is the one of interest in this chapter, namely an **underlying group structure**. This means that nodes can be spread into classes having similar connectivity patterns. In order to retrieve such structures, statistical and algorithmic tools have been developed.

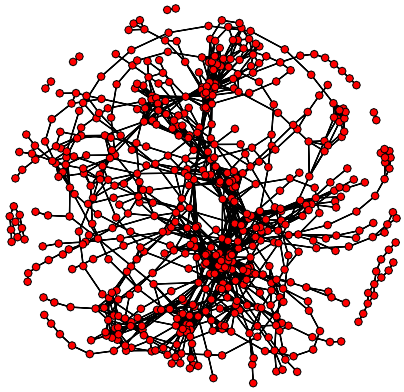

FIGURE 1.2
The metabolic network of bacteria *Escherichia coli* [36]. Nodes of the undirected network correspond to biochemical reactions, and two reactions are connected if a compound produced by the first one is a part of the second one (or vice-versa).

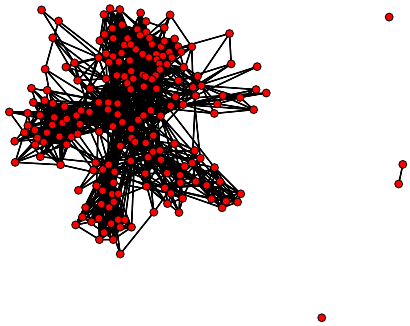

FIGURE 1.3
Subset of the french political blogosphere network. The data consists of a single day snapshot of political blogs automatically extracted on 14th october 2006 and manually classified by the "Observatoire Présidentielle project" [59]. Nodes correspond to hostnames and there is an edge between two nodes if there is a known hyperlink from one hostname to another (or vice-versa).

1.3 Graph clustering

We concentrate in this section on the classification of vertices depending on their connection profiles. There has been a wealth of literature on the topic which goes back to the earlier work of [41]. As shown in [42], it appears that available methods can be grouped into three significant categories. First, some models look for community structure, also called assortative mixing [44, 19], where vertices are partitioned into classes such that vertices of a class are mostly connected to vertices of the same class. Other models look for disassortative mixing in which vertices mostly connect to vertices of different classes. They are commonly used to analyze bipartite networks [22]. Finally, a few procedures look for heterogeneous structure where vertices can have different types of connection profiles. In particular, they can be used to uncover both community structure and disassortative mixing.

In this section, we describe some of the most widely used graph clustering methods. Note that many model free approaches exist [26]. However, except for the algorithmic approach presented in Section 1.3.1, we concentrate in the following on methods which rely on statistical models only.

1.3.1 Community structure

Most graph clustering methods aim at detecting community structure, also called assortative mixing, meaning the appearance of densely connected groups of vertices, with only sparser connections between groups (Figure 1.4). Most of them rely on the modularity score of [46]. However, we point out the recent work of [11] who showed that these algorithms are (asymptotically) biased and that using modularity scores could lead to the discovery of an incorrect community structure, even for large graphs.

1.3.1.1 Modularity score

Newman and Girvan [29, 46] proposed several intuitive community detection algorithms which involve iterative removal of edges from the network to split it into communities. Edges to be removed are identified using one of a number of possible betweenness measures. All of them are based on the same idea. If two communities are joined by only a few *inter* community edges, then all paths from vertices in one community to vertices in the other must pass along one of those few edges. Therefore, given a suitable set of paths, we expect the number of paths that go along an edge to be largest for *inter* community edges.

First, they introduced the edge betweenness which is a generalization to edges of the vertex betweenness measure of [28]. The edge betweenness of an edge is defined as the number of shortest paths between all pairs of vertices in the network that run along that edge. Second, they considered the random

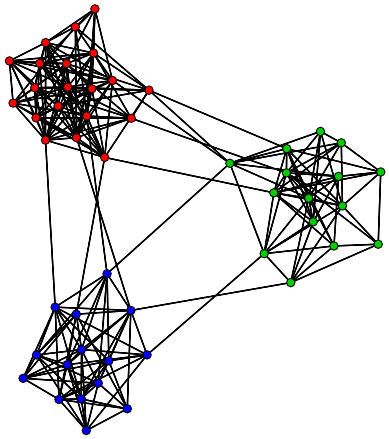

FIGURE 1.4
Example of an undirected affiliation network with 50 vertices. The network is made of three communities represented in red, blue, and green. Vertices connect mainly to vertices of the same community.

walk betweenness. The expected number of times a random walk between a particular pair of vertices will pass down a particular edge is calculated. This expected value is then summed over all pairs of vertices to obtain the random walk betweenness of the edge. As shown in [46], other scores can obviously be considered to obtain algorithms that may be more appropriate for some applications. However, it appears that the choice of measure does not highly influence the result of the algorithms. On the other hand, the recalculation step after each edge removal is crucial (see Algorithm 1).

All these algorithms produce a dendrogram (Figure 1.5) which represents an entirely nested hierarchy of possible community divisions for the network. In order to select one of these divisions, [46] proposed a modularity criterion. Consider a particular division with Q communities and let us denote e_{ql} the fraction of all edges in the network that link vertices in community q to vertices in community l. Moreover, consider the fraction $a_q = \sum_{l=1}^{Q} e_{ql}$ of edges that connect to vertices of community q. The modularity criterion is then given by:

$$\mathcal{Q}_{mod} = \sum_{q=1}^{Q} (e_{qq} - a_q^2). \tag{1.1}$$

The criterion is computed for all the divisions, and a division is chosen such that the modularity is maximized. Note that modularity can be generalized to both directed and valued graphs [26].

A limiting factor of these community detection algorithms is their poor scaling with the number m of edges and the number N of vertices in the net-

work. For instance calculating the shortest paths between a particular pair of vertices can be done in $O(m)$ [1, 18]. Because they are $O(N^2)$ vertex pairs, the computational cost to compute all the edge betweenness scores is in $O(mN^2)$. This complexity was improved independently by [43] and [14] finding all betweennesses in $O(mN)$. Since this calculation has to be repeated for the removal of each edge, the entire algorithm runs in worst-case time $O(m^2N)$. In other words, for dense networks, where m is in $O(N^2)$, it runs in $O(N^5)$ while it scales in $O(N^3)$ for sparse networks, where m is linear in N.

Algorithm 1: Example of a community structure detection algorithm with a betweenness score.

repeat
  Calculate betweenness scores for all edges;
  Remove the edge with the highest score;
until *No edges remain*;

Rather than building the complete dendrogram (with edge removals) and then choosing the optimal division using the modularity criterion, [45] suggested to focus directly on the optimization of the modularity. Thus, he proposed an algorithm which falls in the general category of agglomerative hierarchical clustering methods [24, 53]. Starting with a configuration in which each vertex is the sole member of one of N communities, the communities are iteratively joined together in pairs, choosing at each step the join that results in the greatest increase (or smallest decrease) in *mod* (1.1). Again, this leads to a dendrogram for which the best cut is chosen by looking for the maximal value of the modularity. The computational cost of the entire algorithm is in $O\left((m+N)N\right)$, or $O(N^3)$ for dense networks and $O(N^2)$ for sparse networks. It was shown to be capable of handling a collaboration network with 50000 vertices in [45].

1.3.1.2 Latent position cluster model

An alternative approach for community detection in networks is the Latent Position Cluster Model (LPCM) of [30]. Consider a $N \times N$ binary adjacency matrix $\mathbf{X}$ such that X_{ij} equals 1 if there is an edge from vertex i to vertex j, and 0 otherwise. Moreover, let us define $\mathbf{Y}$ a covariate information where $\mathbf{Y}_{ij}$ denotes some observed characteristics about the pair (i, j) of vertices. This might represent for instance the traffic information of users from blog i to blog j in a blogosphere network (see Figure 1.3). Several characteristics can possibly be observed for each pair of vertices and therefore $\mathbf{Y}_{ij}$ can be vector valued. Note that a few other random graph models have been proposed in the literature to take covariates into account [see for instance 60, 39]. They will not be considered in this chapter where we consider vertices clustering by using the network topology only. Here, we describe LPCM in a general setting, as in

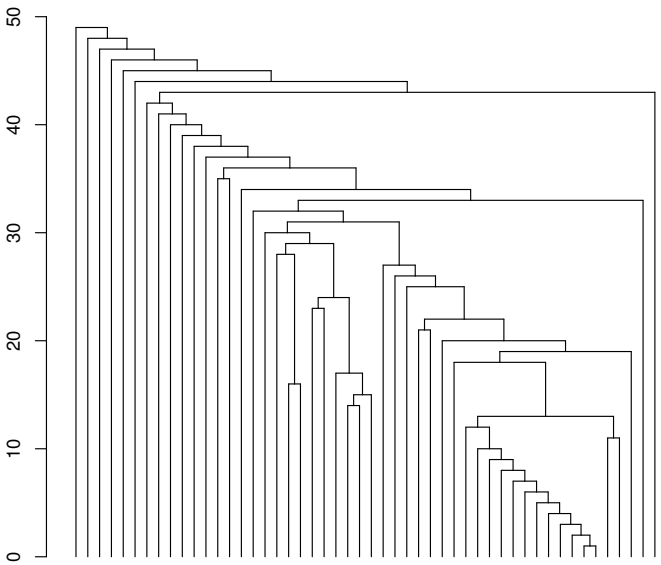

FIGURE 1.5
Dendrogram of a network with 50 vertices for the community detection algorithm with edge betweenness. It should be read from top to bottom. The algorithm starts with a single community which contains all the vertices. Edges with the highest edge betweenness are then removed iteratively splitting the network into several communities. After convergence, each vertex, represented by a leaf of the tree, is a sole member of one of the 50 communities.

[30], and emphasize that the algorithm can also be used if $\mathbf{Y}$ is not available, simply by removing the terms in $\mathbf{Y}_{ij}$ in the following expressions.

LPCM assumes that the network does not contain any self loop while both directed and undirected relations can be analyzed. It is assumed that each vertex, usually called actor in social sciences, has an unobserved position in a d dimensional Euclidean latent space as in [31]. Given the latent positions and the covariate information, the edges are assumed to be drawn from a Bernoulli distribution:

$$X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij} \sim \mathcal{B}\left(g(a_{\mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij}})\right).$$

The function $g(x) = (1 + e^{-x})^{-1}$ is the logistic sigmoid function. Moreover $a_{\mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij}}$ is given by:

$$a_{\mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij}} = \mathbf{Y}_{ij}^{\intercal} \boldsymbol{\beta}_0 - \beta_1 | \, \mathbf{Z}_i - \mathbf{Z}_j \, |, \tag{1.2}$$

where $\boldsymbol{\beta}_0$ as the same dimensionality as $\mathbf{Y}_{ij}$ and β_1 is a scalar. Both $\boldsymbol{\beta}_0$ and β_1 are unknown parameters to be estimated. To represent clustering, the positions are assumed to be drawn from a finite mixture of Q multivariate normal distributions, each one representing a different class of vertices. Each multivariate distribution has its own mean vector as well as spherical covariance matrix:

$$\mathbf{Z}_i \sim \sum_{q=1}^{Q} \alpha_q \mathcal{N}(\boldsymbol{\mu}_q, \sigma_q^2 \mathbf{I}),$$

and $\boldsymbol{\alpha}$ denotes a vector of class proportions which satisfies $\alpha_q > 0, \forall q$ and $\sum_{q=1}^{Q} \alpha_q = 1$. Finally, according to LPCM, the latent positions $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$ are iid and given this latent structure, all the edges are supposed to be independent. Consider now the second term on the right hand side of (1.2). By construction, if β_1 is positive, we expect the L_1 distance $| \, \mathbf{Z}_i - \mathbf{Z}_j \, |$ to be smaller if vertices i and j are in the same class. In other words, the probability $g(a_{\mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij}})$ of an edge between i and j is supposed to be higher for vertices sharing the same class. Note that this corresponds exactly to the definition of a community.

[30] proposed a two-stage maximum likelihood approach and a Bayesian algorithm, as well as a BIC criterion to estimate the number of latent classes. The two-stage maximum likelihood approach first maps the vertices in the latent space and then uses a mixture model to cluster the resulting positions. In practice, this procedure converges more quickly but looses some information by not estimating the positions and the cluster model at the same time. Conversely, the Bayesian algorithm (see Figure 1.6), based on Markov Chain Monte Carlo, estimates both the latent positions and the mixture model parameters simultaneously. It gives better results but is time consuming. Both the maximum likelihood and the Bayesian approach are limited in the sense that they can handle networks with a few hundreds of vertices only.

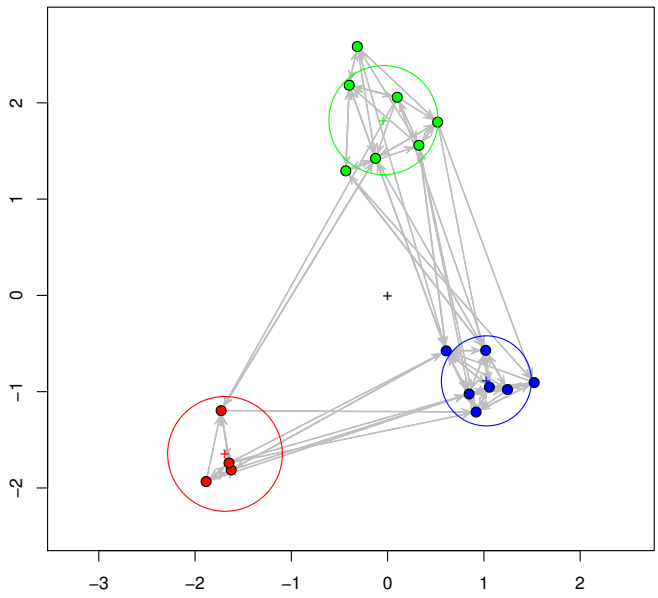

FIGURE 1.6
Directed network of social relations between 18 monks in an isolated American monastery [52, 58]. Sampson collected sociometric information using interviews, experiments, and observations. This network focus on the relation of "liking". A monk is said to have a social relation of "like" to another monk if he ranked that monk in the top three monks for positive affection in any of the three interviews given. The positions of the vertices in the two data dimensional latent space have been calculated using the Bayesian approach for LPCM. The position of the three class centers found are indicated as well as circles with radius equal to the square root of the class variances estimated.

1.3.2 Heterogeneous structure

So far, we have seen some algorithms to uncover communities. However, some vertices may be grouped while exhibiting connection patterns different from a dense group poorly linked to the rest of the network. In genetic regulatory networks, transcription factors co-regulating some biological process may for example not be linked one to each other but act jointly on the regulated genes. Some other approaches which can look for heterogeneous structure in networks, where vertices can have different types of connection profiles, have therefore been developed.

1.3.2.1 Hofman and Wiggins' model

Let us consider a binary adjacency matrix $\mathbf{X}$ representing a network $\mathcal{G}$. The model of Hofman and Wiggins [32] associates to each vertex of the network a latent variable $\mathbf{Z}_i$ drawn from a multinomial distribution:

$$\mathbf{Z}_i \sim \mathcal{M}\left(1, \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_Q)\right). \tag{1.3}$$

As in other standard mixture models, the vector $\mathbf{Z}_i$ has all its components set to zero except one such that Z_{iq} equals 1 if vertex i belong to class q. Thus, $\sum_{q=1}^{Q} Z_{iq} = 1, \forall i$ and the vector $\boldsymbol{\alpha}$ satisfies $\alpha_q > 0, \forall q$ as well as $\sum_{q=1}^{Q} \alpha_q = 1$. The edges are then assumed to be drawn from a Bernoulli distribution:

$$X_{ij} \sim \mathcal{B}(\lambda),$$

if vertices i and j are in the same class, that is $\mathbf{Z}_i = \mathbf{Z}_j$, and

$$X_{ij} \sim \mathcal{B}(\epsilon),$$

otherwise. Thus, the model is able to take into account both community structure ($\lambda > \epsilon$) (Figure 1.4) and disassortative mixing ($\lambda < \epsilon$) (Figure 1.7). As in the previous section, given the latent variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$, all the edges are supposed to be independent. In order to estimate the posterior distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \lambda, \epsilon | \mathbf{X})$ over the latent variables and model parameters, [32] used a variational Bayes Expectation Maximization (EM) algorithm with a factorized distribution:

$$q(\mathbf{Z}, \boldsymbol{\alpha}, \lambda, \epsilon) = q(\boldsymbol{\alpha}) q(\lambda) q(\epsilon) \prod_{i=1}^{N} q(\mathbf{Z}_i).$$

Moreover, they proposed a model selection criterion to estimate the number of latent classes in networks. It relies on a variational approximation of the marginal log-likelihood $\log p(\mathbf{X})$ and has shown promising results.

1.3.2.2 Stochastic block models

Originally developed in social sciences, the Stochastic Block Model (SBM) is a probabilistic generalization [25, 33] of the method described in [58]. Given

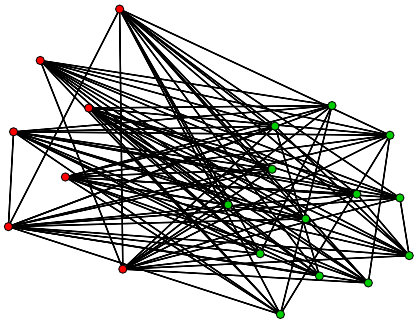

FIGURE 1.7
Example of an undirected network with 20 vertices. The connection probabilities between the two classes in red and green are higher than the *intra* class probabilities. Vertices connect mainly to vertices of a different class.

a network, it assumes that each vertex belongs to a hidden class among Q classes and uses a matrix $\mathbf{\Pi}$ to describe the *intra* and *inter* connection probabilities [27]. No assumption is made on the form of the connectivity matrix such that very different structures can be taken into account. In particular, SBM can characterize the presence of hubs which make networks locally dense [20]. Moreover and to some extent, it generalizes many of the existing graph clustering techniques, as shown in [42]. For instance, the model of Hofman and Wiggins can be seen as a constrained SBM where the diagonal of $\mathbf{\Pi}$ is set to λ and all the other elements to ϵ.

Formally, SBM considers a latent variable $\mathbf{Z}_i$, drawn from a multinomial distribution (1.3), for each vertex in the network, as in section 1.3.2.1. Thus, each vertex belongs to a single class, and that class is q if Z_{iq} equals 1. The edges are then assumed to be drawn from a Bernoulli distribution:

$$X_{ij}|Z_{iq}Z_{jl} = 1 \sim \mathcal{B}(\pi_{ql}),$$

where $\mathbf{\Pi}$ is a $Q \times Q$ matrix of connection probabilities. Again, given all the latent variables, the edges are supposed to be independent. Note that SBM was originally described in a more general setting [47], allowing any discrete relational data. However, as explained in Section 1.2.1, we concentrate in the following on binary edges only.

The identifiability of the parameters in SBM was studied by [6, 7], who showed that the model is generically identifiable up to a permutation of the classes. In other words, except in a set of parameters which has a null Lebesgue's measure, two parameters imply the same random graph model if and only if they differ only by the ordering of the classes.

Many methods have been proposed in the literature to jointly estimate SBM model parameters and cluster the vertices of the network. They all face

the same difficulty. Indeed, contrary to many mixture models, the conditional distribution of all the latent variables $\mathbf{Z}$ and model parameters, given the observed data $\mathbf{X}$, can not be factorized due to conditional dependency. Therefore, optimization techniques such as the Expectation Maximization (EM) algorithm can not be used directly. In the case of SBM, [47] proposed a Bayesian probabilistic approach. They introduced some prior Dirichlet distributions for the model parameters and used Gibbs sampling to approximate the posterior distribution over the model parameters and posterior predictive distribution. Their algorithm is implemented in the software BLOCKS, which is part of the package StoCNET [13]. It gives accurate a posteriori estimates but can not handle networks with more than 200 vertices. [20] proposed a frequentist variational EM approach for SBM which can handle much larger networks and developed an Integrated Classification Likelihood (ICL) criterion for the model selection. [38] adapted it in a Bayesian framework, yielding an algorithm which retrieves better small classes and does the model selection with a non-asymptotic criterion. Online strategies have also been developed [59] as well as extensions to deal with discrete or continuous edges [39].

1.4 Overlapping clustering

As mentioned previously, most graph clustering methods suffer from the restriction they impose by requiring that each vertex belongs to exactly one class. We present in this section some algorithmic and statistical adaptations of the existing clustering methods which tackle this issue. We focus here on the methods assigning to each vertex a vector of $\{0,1\}^Q$, where Q denotes the number of classes. In other terms, each individual belongs completely to all groups it participates in. Methods using vectors of coefficients summing to 1 and giving the relative importance of each class in the individual behavior have also been developed [12, 2] and are more detailed in Chapter ???.

1.4.1 Algorithmic approaches

The issue of overlapping clustering has received growing attention in the last few years, starting with an algorithmic approach based on Clique Percolation developed by [49] and implemented in the software CFinder [50]. In this approach, a k-clique community is defined as the union of all k-cliques (complete sub-graphs of size k) that can be reached from each other through a series of adjacent[1] k-cliques. Given a network, the algorithm first locates all cliques and then identifies the communities using a clique-clique overlap matrix [23]. By construction, the resulting communities can overlap. In order to select the

[1] Two k-cliques are adjacent if they share $k-1$ vertices

optimal value of k, the authors suggested a global criterion which looks for a community structure as highly connected as possible. Small values of k leads to a giant community which smears the details of a network by merging small communities. Conversely, when k increases, the communities tend to become smaller, more disintegrated, but also more cohesive. Therefore, they proposed a heuristic which consists in running their algorithm for various values of k and then to select the lowest value such that no giant community appears.

Shen et al.[54] adapted the classification method of Girvan and Newman to overlapping clusters in a method called EAGLE. To do so, they first build a bottom-up dendrogram by starting from some well-chosen and possibly overlapping maximal cliques. At each step, a distance is computed for every pair of communities, based on the proportion of edges linking those communities. The two nearest ones are then merged. The cut level of the dendrogram is chosen according to a generalization of the modularity to overlapping communities, namely:

$$\mathcal{Q}_{ov} = \frac{1}{2m} \sum_q \sum_{ij} \frac{1}{O_i O_j} \big(X_{ij} - \frac{k_i k_j}{2m}\big) \delta(C_i, C_j),$$

where O_i is equal to the number of communities i belongs to. It can be shown that if all O_i's are equal to 1, this expression is equal to the modularity defined in Equation 1.1. The contribution of each edge then decreases when its incident vertices belong to several communities.

However, those algorithmic procedures are limited to the detection of communities. Statistical tools are then needed to find overlapping heterogeneous structures.

1.4.2 Overlapping stochastic block model

Let us now investigate the adaptation of the Stochastic Block Model to overlapping classes. The hidden structure can no longer be a mixture model, so the constraints $\sum_q Z_{iq} = 1$ and $\sum_q \alpha_q = 1$ present in SBM are relaxed. Thus a new latent vector $\mathbf{Z}_i$ is introduced for each vertex i of the network. This vector is composed from Q independent Boolean variables $Z_{iq} \in \{0, 1\}$, drawn from a multivariate Bernoulli distribution:

$$\mathbf{Z}_i \sim \prod_{q=1}^{Q} \mathcal{B}(Z_{iq};\, \alpha_q) = \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} (1 - \alpha_q)^{1 - Z_{iq}}. \tag{1.4}$$

We point out that $\mathbf{Z}_i$ can also have all its components set to zero which is a useful feature in practice as we shall see in section 1.4.2.2. The edge probabilities are then given by:

$$X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j \sim \mathcal{B}\big(X_{ij};\, g(a_{\mathbf{Z}_i, \mathbf{Z}_j})\big) = e^{X_{ij} a_{\mathbf{Z}_i, \mathbf{Z}_j}} g(-a_{\mathbf{Z}_i, \mathbf{Z}_j}),$$

where

$$a_{\mathbf{Z}_i, \mathbf{Z}_j} = \mathbf{Z}_i^\intercal \, \mathbf{W} \, \mathbf{Z}_j + \mathbf{Z}_i^\intercal \, \mathbf{U} + \mathbf{V}^\intercal \, \mathbf{Z}_j + W^*, \tag{1.5}$$

and $g(x) = (1 + e^{-x})^{-1}$ is the logistic sigmoid function. $\mathbf{W}$ is a $Q \times Q$ real matrix whereas $\mathbf{U}$ and $\mathbf{V}$ are Q-dimensional real vectors. The first term in the right-hand side of (1.5) describes the interactions between the vertices i and j. If i belongs only to class q and j only to class l, then only one interaction term remains ($\mathbf{Z}_i^\intercal \, \mathbf{W} \, \mathbf{Z}_j = W_{ql}$). However, the model can take more complex interactions into account if one or both of these two vertices belong to multiple classes (Figure 1.8). Note that the second term in (1.5) does not depend on $\mathbf{Z}_j$. It models the overall capacity of vertex i to connect to other vertices. By symmetry, the third term represents the global tendency of vertex j to receive an edge. These two parameters $\mathbf{U}$ and $\mathbf{V}$ are related to the sender/receiver effects δ_i and γ_j in the Latent Cluster Random Effects Model (LCREM) of [35]. However, contrary to LCREM, $\delta_i = \mathbf{Z}_i^\intercal \, \mathbf{U}$ and $\gamma_j = \mathbf{V}^\intercal \, \mathbf{Z}_j$ depend on the classes. In other words, two different vertices sharing the same classes, will have exactly the same sender/receiver effects, which is not the case in LCREM. Finally, we use the scalar W^* as a bias, to model sparsity.

If we associate to each latent variable $\mathbf{Z}_i$ a vector $\tilde{\mathbf{Z}}_i = \big(\mathbf{Z}_i, 1\big)^\intercal$, then (1.5) can be written:

$$a_{\mathbf{Z}_i, \mathbf{Z}_j} = \tilde{\mathbf{Z}}_i{}^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j, \tag{1.6}$$

where

$$\tilde{\mathbf{W}} = \begin{pmatrix} \mathbf{W} & \mathbf{U} \\ \mathbf{V}^\intercal & W^* \end{pmatrix}.$$

The $\tilde{Z}_{i(Q+1)}$s can be seen as random variables drawn from a Bernoulli distribution with probability $\alpha_{Q+1} = 1$. Thus, one way to think about the model is to consider that all the vertices in the graph belong to a $(Q+1)$-th cluster which is overlapped by all the other clusters. In the following, we will use (1.6) to simplify the notations.

Finally, given the latent structure $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$, all the edges are supposed to be independent. Thus, when considering directed graphs without self-loop, the Overlapping Stochastic Block Model (OSBM) is defined through the following distributions:

$$p(\mathbf{Z} \,|\, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} (1 - \alpha_q)^{1 - Z_{iq}}, \tag{1.7}$$

and

$$p(\mathbf{X} \,|\, \mathbf{Z}, \tilde{\mathbf{W}}) = \prod_{i \neq j}^{N} e^{X_{ij} a_{\mathbf{Z}_i, \mathbf{Z}_j}} g(-a_{\mathbf{Z}_i, \mathbf{Z}_j}).$$

The graphical model of OSBM is given in Figure 1.9.

1.4.2.1 Modeling sparsity

As mentioned in 1.2, real networks are often sparse and it is crucial to distinguish the two sources of non-interaction. Sparsity might be the result of the rarity of interactions in general but it might also indicate that some class

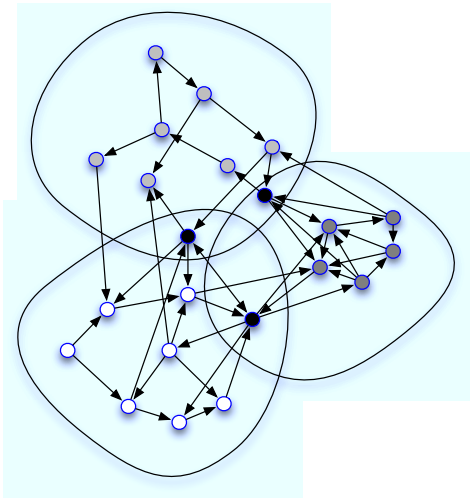

FIGURE 1.8
Example of a directed graph with three overlapping clusters.

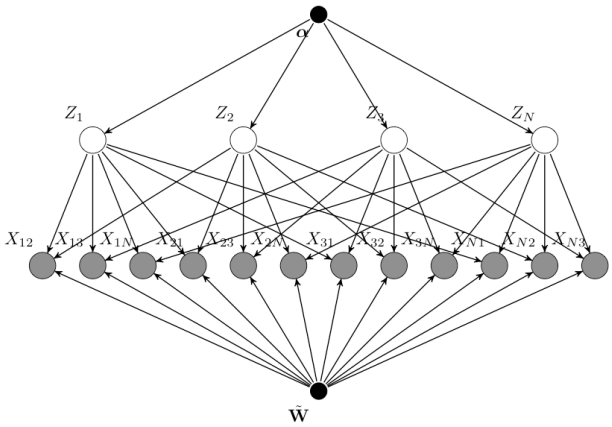

FIGURE 1.9
Directed acyclic graph representing the frequentist view of the overlapping stochastic block model. Nodes represent random variables, which are shaded when they are observed and edges represent conditional dependencies.

(*intra* or *inter*) connection probabilities are close to zero. For instance, social networks are often made of communities where vertices are mostly connected to vertices of the same community. This corresponds to classes with high *intra* connection probabilities and low *inter* connection probabilities. In (1.5), we can notice that W^* appears in $a_{\mathbf{Z}_i,\mathbf{Z}_j}$ for every pair of vertices. Therefore, W^* is a convenient parameter to model the two sources of sparsity. Indeed, low values of W^* result from the rarity of interactions in general, whereas high values signify that sparsity comes from the classes (parameters in $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{V}$).

1.4.2.2 Modeling outliers

When applied on real networks, graph clustering methods often lead to giant classes of vertices having low output and input degrees [20, 37]. These classes are usually discarded and the analysis of networks focus on more highly structured classes to extract useful information. The product of Bernoulli distributions (1.7) provides a natural way to encode these "outliers". Indeed, rather than using giant classes, OSBM uses the null component such that $\mathbf{Z}_i = \mathbf{0}$ if vertex i is an outlier and should not be classified in any class.

1.4.2.3 Identifiability

As in the case of the SBM, reordering the Q classes of the OSBM and doing the corresponding modification in $\boldsymbol{\alpha}$ and $\tilde{\mathbf{W}}$ does not change the generative random graph model.

There is another family of operations which does not change the generative random graph model, which we call inversions. They correspond to fix a subset $S \subset \{1, \dots, Q\}$ and to exchange the labels 0 to 1 and vice-versa on the coordinates of the Z_i's included in S. To give an intuition, let us consider the inversion with $S = \{1\}$. If we denote by "cluster 1" the vertices whose Z_i's have a 1 as first coordinate, the initial graph sampling procedure consists in sampling the set "cluster 1" and then drawing the edges conditionally on that information. After the inversion, it samples the vertices which are not in "cluster 1" and draws the edges conditionally on that information, which is an equivalent procedure.

As shown in [38], the OSBM is generically identifiable up to permutations of the classes and inversions. In other words, except in a set of parameters which has a null Lebesgue's measure, two parameters imply the same random graph model if and only if the second can be obtained from the first by a permutation and an inversion.

1.4.2.4 Parameter estimation

The log-likelihood of the observed data set is defined through the marginalization: $p(\mathbf{X} \,|\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \,|\alpha, \tilde{\mathbf{W}})$. This summation involves 2^{NQ} terms and quickly becomes intractable. To tackle this issue, the EM algorithm has

been applied on many mixture models. However, the E-step requires the calculation of the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ which cannot be factorized in the case of networks [20]. In order to obtain a tractable procedure, some approximations based on global and local variational techniques have to be done.

The global variational technique consists in considering, for any distribution $q(\mathbf{Z})$, the decomposition

$$\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) + \mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})\right), \tag{1.8}$$

where

$$\mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})}{q(\mathbf{Z})} \right\}, \tag{1.9}$$

and $\mathrm{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence. The maximum $\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ of the lower bound $\mathcal{L}_{ML}$ (1.9) is reached when $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$. Thus, if the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ was tractable, the optimizations of $\mathcal{L}_{ML}$ and $\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})$, with respect to $\boldsymbol{\alpha}$ and $\tilde{\mathbf{W}}$, would be equivalent. However, in the case of networks, $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ cannot be calculated and $\mathcal{L}_{ML}$ cannot be optimized over the entire space of $q(\mathbf{Z})$ distributions. Thus, the optimisation is restricted to the class of distributions which satisfy:

$$q(\mathbf{Z}) = \prod_{i=1}^{N} q(\mathbf{Z}_i), \tag{1.10}$$

with

$$\begin{aligned} q(\mathbf{Z}_i) &= \prod_{q=1}^{Q} \mathcal{B}(Z_{iq};\, \tau_{iq}) \\ &= \prod_{q=1}^{Q} \tau_{iq}^{Z_{iq}} (1 - \tau_{iq})^{1 - Z_{iq}}. \end{aligned}$$

Each τ_{iq} is a variational parameter which corresponds to the posterior probability of node i to belong to class q.

This global variational approximation is sufficient to obtain a tractable problem in the case of SBM. Unfortunately, in the case of OSBM, a term $\mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j}[\log g(-a_{\mathbf{Z}_i, \mathbf{Z}_j})]$ appears when writing down the complete formula of $\mathcal{L}_{ML}(q)$. Since the logistic sigmoid function is non linear, it cannot be computed analytically. Thus, we need a second level of approximation to optimize the lower bound of the observed data set. It consists in considering again a lower bound and new parameters such that the bound is tight for the optimal values of the parameters.

More precisely, given a variational parameter ξ_{ij}, $\mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j}[\log g(-a_{\mathbf{Z}_i, \mathbf{Z}_j})]$ satisfies:

$$\mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j}[\log g(-a_{ij})] \geq \log g(\xi_{ij}) - \frac{(\tilde{\boldsymbol{\tau}}_i{}^{\intercal} \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j + \xi_{ij})}{2} - \lambda(\xi_{ij}) \Big(\mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j}[(\tilde{\mathbf{Z}}_i{}^{\intercal} \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j)^2] - \xi_{ij}^2 \Big). \tag{1.11}$$

Eventually, it leads to the two steps approximation:

$$\log p(\mathbf{X} \,|\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \geq \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \geq \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}), \tag{1.12}$$

The developed expression of $\mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi})$ is then tractable. It can be found in [38]. The resulting variational EM algorithm (see Algorithm 2) alternatively computes the parameters ξ_{ij}, the posterior probabilities $\boldsymbol{\tau}_i$ and the parameters $\boldsymbol{\alpha}$ and $\tilde{\mathbf{W}}$ maximizing

$$\max_{\boldsymbol{\xi}} \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}).$$

Algorithm 2: Overlapping stochastic block model for directed graphs without self loop.

`// INITIALIZATION;`

Initialize $\boldsymbol{\tau}$ with an Ascendant Hierarchical Classification algorithm;
Sample $\tilde{\mathbf{W}}$ from a zero mean σ^2 spherical Gaussian distribution;

`// OPTIMIZATION;`

repeat

`// ξ-transformation;`

$\xi_{ij} \leftarrow \sqrt{\mathrm{Tr}\left(\tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}}\, \boldsymbol{\Sigma}_j\right) + \tilde{\boldsymbol{\tau}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j}, \forall i \neq j;$

`// M-step;`

$\alpha_q \leftarrow \frac{\sum_{i=1}^N \tau_{iq}}{N}, \forall q;$

Optimize $\mathcal{L}_{ML}\left(q; \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}\right)$ with respect to $\tilde{\mathbf{W}}$, with a gradient based optimization algorithm [e.g. quasi-Newton method of 16];

`// E-step;`

repeat

for *i=1:N* **do**

Optimize $\mathcal{L}_{ML}\left(q; \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}\right)$ with respect to $\boldsymbol{\tau}_i$, with a box constrained ($\tau_{iq} \in [0, 1]$) gradient based optimization algorithm [e.g. Byrd method 17];

until $\boldsymbol{\tau}$ *converges*;

until $\mathcal{L}_{ML}\left(q; \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}\right)$ *converges*;

The computational cost of the algorithm is equal to $O(N^2Q^4)$. For comparison the computational cost of the methods proposed by [20] and [37] for (non-overlapping) SBM is equal to $O(N^2Q^2)$. Analyzing a sparse network with 100 nodes takes about ten seconds on a dual core, and about a minute for dense networks.

Bibliography

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, Upper Saddle River, New Jersey, 1993.

[2] E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[3] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Modern Physics*, 74:47–97, 2002.

[4] R. Albert, H. Jeong, and A.L. Barabasi. Diameter of the world-wide web. *Nature*, 401:130–131, 1999.

[5] R. Albert, H. Jeong, and A.L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[6] E.S. Allman, C. Matias, and J.A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *Annals of Statistics*, 37(6A):3099–3132, 2009.

[7] E.S. Allman, C. Matias, and J.A. Rhodes. Parameters identifiability in a class of random graph mixture models. *ArXiv e-prints*, 2010.

[8] L.A.N Amaral, A. Scala, M. Barthlmy, and H.E. Stanley. Classes of small-world networks. In *Proceedings of the National Academy of Sciences*, volume 97, pages 11149–11152, 2000.

[9] A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[10] A.L. Barabási and Z.N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Rev. Genet*, 5:101–113, 2004.

[11] P.J. Bickel and A. Chen. A non parametric view of network models and newman-girvan and other modularities. In *Proceedings of the National Academy of Sciences*, volume 106, pages 21068–21073, 2009.

[12] D. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[13] P. Boer, M. Huisman, T.A.B. Snijders, C.E.G. Steglich, L.H.Y Wichers, and E.P.H Zeggelink. *StOCNET : an open software system for the advanced statistical analysis of social networks.* Groningnen:ProGAMMA/ICS, 2006. Version 1.7.

[14] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[15] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33:309–320, 2000.

[16] C.G. Broyden, R. Fletcher, D. Goldfarb, and D.F. Shanno. Bfgs method. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.

[17] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *Journal on Scientific and Statistical Computing*, 16:1190–1208, 1995.

[18] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms.* MIT Press, Cambridge, 2001.

[19] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *J Stat Mech*, 2005.

[20] J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and Computing*, 18:1–36, 2008.

[21] S.N. Dorogovtsev, J.F.F. Mendes, and A.N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letter*, 85:4633–4636, 2000.

[22] E. Estrada and J.A. Rodriguez-Velazquez. Spectral measures of bipartivity in complex networks. *Physical Review E*, 72:046105, 2005.

[23] M.G. Everett and S.P. Borgatti. Analyzing clique overlap. *Connections*, 21:49–61, 1998.

[24] B. Everitt. *Cluster analysis.* Wiley, 1974.

[25] S.E. Fienberg and S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981.

[26] S. Fortunato. Community detection in graphs. *Physics Reports*, 3-5:75–174, 2010.

[27] O. Frank and F. Harary. Cluster inference by using transitivity indices in empirical graphs. *Journal of the American Statistical Association*, 77:835–840, 1982.

[28] L. Freeman. A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41, 1977.

[29] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*, volume 99, pages 7821–7826, 2002.

[30] M.S. Handcock, A.E. Raftery, and J.M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society*, 170:1–22, 2007.

[31] P.D. Hoff, A.E. Raftery, and M.S. Handcock. Latent space approaches to social network analysis. *Journal of the Royal Statistical Society*, 97:1090–1098, 2002.

[32] J.M. Hofman and C.H. Wiggins. A bayesian approach to network modularity. *Physical Review Letters*, 100:258701, 2008.

[33] P. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: some first steps. *Social Networks*, 5:109–137, 1983.

[34] C.J. Jeffery. Moonlighting proteins. *Trends in Biochemical Sciences*, 24:8–11, 1999.

[35] P.N. Krivitsky, M.S. Handcock, A.E. Raftery, and P.D. Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31:204–213, 2009.

[36] V. Lacroix, C.G. Fernandes, and M.-F. Sagot. Motif search in graphs: pplication to metabolic networks. *Transactions in Computational Biology and Bioinformatics*, 3:360–368, 2006.

[37] P. Latouche, E. Birmelé, and C. Ambroise. *Bayesian methods for graph clustering*, pages 229–239. Springer, 2009.

[38] P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block model with application to the french political blogosphere. *Annals of Applied Statistics*, 5(1):309–336, 2011.

[39] M. Mariadassou, S. Robin, and C. Vacher. Uncovering latent structure in valued graphs: a variational approach. *Annals of Applied Statistics*, 4(2), 2010.

[40] R. Milo, S. Shen-Orr, S. Itzkovitz, D. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[41] J.L. Moreno. *Who shall survive?: a new approach to the problem of Human interrelations*. Nervous and Mental Disease Publishing, Washington DC, 1934.

[42] M. Newman and E. Leicht. Mixture models and exploratory analysis in networks. In *Proceedings of the National Academy of Sciences*, volume 104, pages 9564–9569, 2007.

[43] M.E.J Newman. Scientific collaboration networks: Ii. shortest paths, weighted networks, and centrality. *Physical Review E*, 64:016132, 2001.

[44] M.E.J. Newman. Mixing patterns in networks. *Physical Review E*, 67:026126, 2003.

[45] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review Letter*, 69, 2004.

[46] M.E.J Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[47] K. Nowicki and T.A.B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.

[48] G. Palla, A.L Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, 2007.

[49] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[50] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. *CFinder, the community cluster finding program*, 2006. Version 2.0.1.

[51] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86:3200–3203, 2001.

[52] S.F. Sampson. *Crisis in a cloister*. PhD thesis, Cornell University, 1969.

[53] J. Scott. *Social network analysis: a handbook*. Sage publications, 2000.

[54] H. Shen, X. Cheng, K. Cai, and M.B. Hu. Detect overlapping and hierarchical structure in networks. *Physica A*, (388):1706–1712, 2009.

[55] T.A.B. Snijders and K. Nowicki. Estimation and prediction for stochastic block-structures for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

[56] S.H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[57] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[58] H.C. White, S.A. Boorman, and R.L. Breiger. Social structure from multiple networks. i. blockmodels of roles and positions. *American Journal of Sociology*, 81:730–780, 1976.

[59] H. Zanghi, C. Ambroise, and V. Miele. Fast online graph clustering via erdös renyi mixture. *Pattern Recognition*, 41(12):3592–3599, 2008.

[60] H. Zanghi, S. Volant, and C. Ambroise. Clustering based on random graph model embedding vertex features. *Pattern Recognition Letters*, 31(9):830–836, 2010.