Statistical Modelling 2015; **15(6)**: 564–589

Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood

Etienne Côme[1] **and Pierre Latouche**[2]
[1]Université Paris-Est, IFSTTAR, GRETTIA, Noisy-Le-Grand, France
[2]Laboratoire SAMM, Université Paris 1 Panthéon-Sorbonne, France

Abstract: The stochastic block model (SBM) is a mixture model for the clustering of nodes in networks. The SBM has now been employed for more than a decade to analyze very different types of networks in many scientific fields, including biology and the social sciences. Recently, an analytical expression based on the collapsing of the SBM parameters has been proposed, in combination with a sampling procedure that allows the clustering of the vertices and the estimation of the number of clusters to be performed simultaneously. Although the corresponding algorithm can technically accommodate up to 10 000 nodes and millions of edges, the Markov chain, however, tends to exhibit poor mixing properties, that is, low acceptance rates, for large networks. Therefore, the number of clusters tends to be highly overestimated, even for a very large number of samples. In this article, we rely on a similar expression, which we call the integrated complete data log likelihood, and propose a greedy inference algorithm that focuses on maximizing this exact quantity. This algorithm incurs a smaller computational cost than existing inference techniques for the SBM and can be employed to analyze large networks (several tens of thousands of nodes and millions of edges) with no convergence problems. Using toy datasets, the algorithm exhibits improvements over existing strategies, both in terms of clustering and model selection. An application to a network of blogs related to illustrations and comics is also provided.

Key words: greedy inference; integrated classification likelihood; networks; random graphs; stochastic block models

Received May 2014; **revised** November 2014; **accepted** February 2015

1 Introduction

1.1 Context

Research on networks has a long history dating back to the earlier work of Moreno (1934). Because networks are simple data structures that can represent complex systems, they are used in many scientific fields (Barabási and Oltvai, 2004; Palla *et al.*, 2007). Networks were first applied in the social sciences (Fienberg and

Address for correspondence: Pierre Latouche, Laboratoire SAMM, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, F-75634 Paris Cedex 13, France.
E-mail: pierre.latouche@univ-paris1.fr

 10.1177/1471082X15577017

Wasserman, 1981) to characterize relationships among actors (Holland *et al*., 1993; Boulet *et al*., 2008) but they are now also used to describe neural networks (White *et al*., 1986), power grids (Watts and Strogatz, 1998) and the Internet (Adamic and Glance, 2005; Zanghi *et al*., 2008). Other examples of real networks can be found in biology, in which regulatory networks are used to describe the regulation of genes by transcriptional factors (Milo *et al*., 2002) and metabolic networks are used to represent biochemical reaction pathways (Lacroix *et al*., 2006). As the number of networks used in practice increases, substantial effort focuses on the development of graph-clustering algorithms to extract knowledge from network topology. Existing methods are typically focused on uncovering very specific patterns in the data, namely, communities or disassortative mixing. For an exhaustive review, we refer to Goldenberg *et al*. (2010).

Most graph-clustering algorithms attempt to identify communities. According to the definition of community, two nodes of the same community are more likely to be connected than nodes of different communities. These techniques (Newman, 2004, 2006) often maximize the modularity score proposed by Girvan and Newman (2002) for clustering, using, for example, greedy heuristics (Blondel *et al*., 2008). However, the recent work of Bickel and Chen (2009) has demonstrated that this approach is asymptotically biased and tends to lead to the identification of an incorrect community structure, even for large graphs. Alternative strategies (see, for instance, Krivitsky *et al*. (2009)) are generally related to the probabilistic model of Handcock *et al*. (2007), which generalizes the work of Hoff *et al*. (2002). In this approach, nodes are first mapped into a latent space and then clustered depending on their latent positions. Community structure algorithms are commonly used for affiliation network analysis. As mentioned in Newman and Leicht (2007), other graph-clustering algorithms are focused on uncovering disassortative mixing in networks; in this type of network pattern, nodes are mostly connected to nodes of different clusters, in contrast to a community structure. These algorithms are particularly suitable for the analysis of bipartite or quasi-bipartite networks (Estrada and Rodriguez-Velazquez, 2005).

In contrast to these methods, graph-clustering algorithms based on the stochastic block model (SBM) that can retrieve heterogeneous structures have been developed. The SBM was first proposed by Nowicki and Snijders (2001) and is a probabilistic generalization (Fienberg and Wasserman, 1981; Holland *et al*., 1993) of the work of White *et al*. (1976). The SBM assumes that the nodes are arranged in K clusters and uses a $K \times K$ matrix π to describe the probabilities of connection between pairs of nodes. No assumption is imposed on π, and thus a variety of very different structures can be considered. In particular, as shown in Latouche *et al*. (2009), the SBM can be used to retrieve both communities and disassortative mixing in networks.

Although recent research has focused on the proposal of new types of SBMs to address, for instance, valued edges (Mariadassou *et al*., 2010), overlapping clusters (Airoldi *et al*., 2006, 2007, 2008; Latouche *et al*., 2011), or degree (number of edges of each node) heterogeneity (Karrer and Newman, 2011), Mc Daid *et al*. (2013) have chosen to consider the standard SBM and to focus on the inference task. These authors have proposed a new inference procedure, which shall be discussed in Section 1.3, that has yielded very encouraging results. Following their

work, in this article, we consider the standard SBM, which has been widely used in practice for network analysis for more than a decade. Our objective is to develop a new optimization procedure to improve upon existing inference strategies. This framework can be extended to other types of SBMs.

1.2 Inference in stochastic block models

In an SBM, the posterior distribution over the latent variables, given the parameters and the observed data, cannot be factorized because of conditional dependency. Therefore, optimization techniques such as the expectation maximization (EM) algorithm cannot be used directly for clustering. In response, Daudin *et al.* (2008) have proposed an approximation method based on a variational EM algorithm available in the **mixer** R package written in R and C. Note that an online version of this algorithm is also available (Zanghi *et al.*, 2010). A Bayesian framework was also considered by Nowicki and Snijders (2001); in this artilce, conjugate priors for the model parameters were introduced. Again, because the posterior distribution over the model parameters, given the data, is not tractable, approximation techniques must be employed for inference. Thus, Nowicki and Snijders (2001) used a Gibbs sampling procedure, whereas Latouche *et al.* (2012) relied on a variational Bayes EM algorithm. Note that a similar approach has been considered by Hofman and Wiggins (2008) for a constrained SBM, in which all terms on the diagonal of the connectivity matrix $\boldsymbol{\pi}$ are set to a unique parameter λ and all off-diagonal terms are set to another parameter ϵ. A MATLAB and C implementation of this method are available in the software **vbmod**.

Two model selection criteria, the integrated classification likelihood (ICL) and the integrated likelihood variational Bayes (ILvb), have been developed for the SBM for the purpose of estimating the number of clusters, K, in a network. Standard criteria, such as the Akaike information criterion or the Bayesian information criterion, cannot be used because they rely on the SBM observed data log likelihood, which is not tractable in practice (see, for instance, Latouche *et al.* (2009)). However, as demonstrated in Biernacki *et al.* (2010), the ICL tends to miss some important structures in the data for small data samples because the ICL is based on asymptotic approximations. To address this shortcoming, Latouche *et al.* (2012) proposed the ILvb criterion, which relies on a variational Bayes approximation of the integrated observed data log likelihood.

1.3 Contributions

An alternative inference strategy has recently been proposed for the SBM in Mc Daid *et al.* (2013). The authors first derived an analytical expression based on the collapsing of the SBM parameters. Then, they relied on an allocation sampler algorithm, as in Nobile and Fearnside (2007), which allows the clustering of the vertices and the estimation of the number of clusters to be performed simultaneously. This sampling procedure, implemented in C in a software utility that we refer to as **colsbm**,

represents an improvement over existing inference strategies for the SBM in terms of both clustering and model selection. Although the algorithm can technically accommodate up to 10 000 nodes and millions of edges, the corresponding Markov chain tends to exhibit poor mixing properties, that is, low acceptance rates, for such large networks. In practice, certain procedures intended to reduce the model complexity are rarely accepted, and the convergence of the chain is slow. Therefore, the number of clusters tends to be highly overestimated, even for a very large number of samples. In this article, we attempt to avoid this by relying on a similar analytical expression, which we call the integrated complete data log likelihood. The corresponding criterion is denoted by $\mathrm{ICL}_{\mathrm{ex}}$, where ex stands for 'exact'. In contrast to the ICL criterion presented in Daudin *et al.* (2008), $\mathrm{ICL}_{\mathrm{ex}}$ does not rely on any asymptotic approximations. We then propose a greedy inference algorithm that maximizes this exact quantity.

In contrast to the clustering algorithms of Daudin *et al.* (2008) and Latouche *et al.* (2012), our proposed algorithm maximizes an analytical criterion and does not rely on any lower bounds for approximation. The lower bound of the variational EM algorithm proposed by Daudin *et al.* (2008) approximates the observed data log likelihood, whereas Latouche *et al.* (2012) introduced a lower bound to estimate the integrated observed data log likelihood. Advantageously, our greedy search approach can perform the clustering of the vertices and the estimation of the number of clusters simultaneously, as in Mc Daid *et al.* (2013), and no model selection criterion must be computed for various values of K. Starting from a complex model with $K = K_{\mathrm{up}}$ clusters (where K_{up} is an upper bound of K), the proposed algorithm swaps labels until $\mathrm{ICL}_{\mathrm{ex}}$ reaches a local maximum. During this process, clusters may disappear, that is, their cardinality may reach zero. Such an approach leads to a simple and time-conserving algorithm with a complexity of $\mathcal{O}(L + NK_{\mathrm{up}}^2)$, where L is the total number of edges in the network and N is the number of vertices. Thus, this approach incurs a lower computational cost than existing inference techniques for the SBM and can be employed to analyze large networks while avoiding the convergence sampling issues of Mc Daid *et al.* (2013).

As will be made evident in a series of experiments, the greedy algorithm takes advantage of computing the exact ICL and represents an improvement over existing methods in terms of both clustering and model selection. The algorithm can also accommodate large networks with tens of thousands of vertices and millions of edges.

2 The stochastic block model

We consider a binary network with N nodes represented by an adjacency matrix $\mathbf{X}$, such that $X_{ij} = 1$ if there is an edge from node i to node j and $X_{ij} = 0$ otherwise. In this article, we focus on directed networks, that is, networks in which relations are oriented. Therefore, $\mathbf{X}$ is not symmetric. Moreover, we do not consider any self-loop, that is, an edge from a node to itself. We emphasize that all optimization equations derived in this work can be easily adapted for application to undirected networks or to account for self-loops.

2.1 Model and notations

The SBM assumes that the nodes are arranged in K clusters with prior probabilities $\{\alpha_1, \ldots, \alpha_K\}$, where the cluster of each node is given by its binary membership vector $\mathbf{Z}_i$, such that $Z_{ik} = 1$ if i belongs to cluster k and $Z_{ik} = 0$ otherwise:

$$\alpha_k = \mathbb{P}(Z_{ik} = 1) = \mathbb{P}(i \in k), \text{ with } \sum_{k=1}^{K} \alpha_k = 1.$$

In contrast to the work of Latouche *et al.* (2011), each node belongs to a single cluster, that is, $\sum_{k=1}^{K} Z_{ik} = 1, \forall i$. Then, the probability that there is an edge from a node of cluster k to a node of cluster l is denoted by π_{kl}. Finally, given the clusters of vertices i and j, all edges are assumed to be conditionally independent:

$$\begin{cases} X_{ij} | \{i \in k, j \in l\} & \sim B(\pi_{kl}) \quad \text{for } i \neq j \\ X_{ii} & = 0. \end{cases}$$

Generative model

This leads to a simple yet flexible generative model for networks. First, all vectors, $\mathbf{Z}_i$, are sampled independently. We denote by $\mathbf{Z}$ the binary $N \times K$ matrix that stores $\mathbf{Z}_i$s as raw vectors:

$$p(\mathbf{Z}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{k=1}^{K} \alpha_k^{Z_{ik}}. \tag{2.1}$$

Then, given the latent structure $\mathbf{Z}$, all edges in $\mathbf{X}$ are drawn independently:

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\pi}) &= \prod_{i \neq j}^{N} p(X_{ij}|\mathbf{Z}_i, \mathbf{Z}_j, \boldsymbol{\pi}) \\ &= \prod_{i \neq j}^{N} \prod_{k,l}^{K} \mathcal{B}(X_{ij}; \pi_{kl})^{Z_{ik} Z_{jl}} \\ &= \prod_{i \neq j}^{N} \prod_{k,l}^{K} \left(\pi_{kl}^{X_{ij}} (1 - \pi_{kl})^{1 - X_{ij}} \right)^{Z_{ik} Z_{jl}}. \end{aligned} \tag{2.2}$$

2.2 Integrated classification likelihood criteria

In this article, we will consider the integrated complete data log likelihood $\log p(\mathbf{X}, \mathbf{Z}|K)$, which will allow us to focus on the inference of $\mathbf{Z}$ and K from the observed data $\mathbf{X}$, because all SBM parameters $(\boldsymbol{\alpha}, \boldsymbol{\pi})$ are integrated out. A similar

quantity was described in Mc Daid *et al.* (2013), in a different context and for a different purpose, when the authors derived the posterior distributions of an allocation sampler algorithm.

We first provide a brief summary of the existing approximations and then, in Section 2.2.2, derive the integrated complete data log likelihood.

2.2.1 Asymptotic ICL criterion

When a factorized prior distribution $p(\boldsymbol{\alpha}, \boldsymbol{\pi}|K) = p(\boldsymbol{\alpha}|K)p(\boldsymbol{\pi}|K)$ over the model parameters is considered, as in Biernacki *et al.* (2000), the integrated complete data log likelihood straightforwardly decomposes into two terms:

$$\begin{aligned}\log p(\mathbf{X}, \mathbf{Z}|K) &= \log\left(\int_{\boldsymbol{\alpha},\boldsymbol{\pi}} p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\alpha}|K)d\boldsymbol{\alpha}d\boldsymbol{\pi}\right)\\ &= \log\left(\int_{\boldsymbol{\pi}} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\pi}, K)p(\boldsymbol{\pi}|K)d\boldsymbol{\pi}\int_{\boldsymbol{\alpha}} p(\mathbf{Z}|\boldsymbol{\alpha}, K)p(\boldsymbol{\alpha}|K)d\boldsymbol{\alpha}\right)\\ &= \log p(\mathbf{X}|\mathbf{Z}, K) + \log p(\mathbf{Z}|K).\end{aligned} \tag{2.3}$$

However, for an arbitrary choice of the priors $p(\boldsymbol{\alpha}|K)$ and $p(\boldsymbol{\pi}|K)$, the marginal distributions $p(\mathbf{X}|\mathbf{Z}, K)$ and $p(\mathbf{Z}|K)$ are usually not tractable and Equation(2.3) does not have any analytical form. To address this issue, Daudin *et al.* (2008) have relied on an asymptotic approximation of $\log p(\mathbf{X}, \mathbf{Z}|K)$, the so-called ICL. Note that the ICL was originally proposed by Biernacki *et al.* (2000) for Gaussian mixture models. The ICL was then adapted by Biernacki *et al.* (2010) for application to mixtures of multivariate multinomial distributions and for application to the SBM by Daudin *et al.* (2008). In the case of a directed graph without self-loops, such as the one considered here, the ICL is given by

$$\begin{aligned}\mathrm{ICL}(\mathbf{Z}, \mathrm{K}) &\approx \log p(\mathbf{X}, \mathbf{Z}|K)\\ &= \max_{\boldsymbol{\alpha},\boldsymbol{\pi}} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\alpha}, \boldsymbol{\pi}, K) - \frac{1}{2}K^2 \log\left(N(N-1)\right) - \frac{K-1}{2}\log(N).\end{aligned} \tag{2.4}$$

For an extensive description of the use of the Laplace and Stirling approximations to derive the ICL criterion, we refer to Biernacki *et al.* (2000). Because it approximates the integrated complete data log likelihood, the ICL is particularly suitable when the focus is on the clustering task and not on the estimation of the data density. However, as demonstrated in Biernacki *et al.* (2010); Mariadassou *et al.* (2010), analyses based on the ICL tend to miss certain important structures present in the data because of the use of (asymptotic) approximations.

We emphasize that the ICL is only used in the literature as a model selection criterion. In practice, a clustering method, such as an EM-like algorithm, is generally employed to obtain several estimates $\tilde{\mathbf{Z}}$ of $\mathbf{Z}$ for various values of the number of classes K. The ICL is then computed for every pair $(\tilde{\mathbf{Z}}, K)$, and the pair $(\tilde{\mathbf{Z}}^*, K^*)$ is chosen such that the criterion is maximized. Thus, the ICL is optimized only through the results

$(\tilde{\mathbf{Z}}, K)$ that are produced by the clustering algorithm. Conversely, after providing an analytical expression ICL_{ex} for the integrated complete data log likelihood in the next section, we will demonstrate in Section 3 how to directly optimize ICL_{ex} with respect to $\mathbf{Z}$ and K. As shown in Section 3.1, such an approach reduces the computational cost of the inference procedure.

2.2.2 Exact ICL criterion

We rely on the same Bayesian framework used in Latouche *et al.* (2009). Thus, we consider non-informative conjugate priors for the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$. Because $\boldsymbol{\alpha}$, which describes the cluster proportions, parameterizes a multinomial distribution (2.1), we rely on a Dirichlet prior distribution:

$$p(\boldsymbol{\alpha}) = \text{Dir}\left(\boldsymbol{\alpha};\ \mathbf{n}^0 = (n_1^0, \ldots, n_K^0)\right).$$

The hyperparameters are frequently fixed to 1/2, that is, $n_k^0 = 1/2, \forall k$. Such a distribution corresponds to a non-informative Jeffreys prior, which is known to be proper (Jeffreys, 1946). A uniform distribution can also be obtained by setting the hyperparameters to 1.

Moreover, because the presence or absence of an edge between nodes is sampled from a Bernoulli distribution, we consider independent beta prior distributions to model the connectivity matrix $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi}) = \prod_{k,l}^{K} \text{Beta}(\pi_{kl};\ \eta_{kl}^0, \zeta_{kl}^0).$$

Again, if no prior information is available, then all hyperparameters η_{kl}^0 and ζ_{kl}^0 can be set to 1/2 or 1 to obtain a Jeffreys or uniform distribution.

With these choices of conjugate prior distributions over the model parameters, the marginal distributions $p(\mathbf{X}|\mathbf{Z}, K)$ and $p(\mathbf{Z}|K)$ in Equation(2.3) have analytical forms, as does the integrated complete data log likelihood, as proven in Appendix A. We refer to the corresponding criterion as ICL_{ex}, where ex indicates 'exact'. This criterion is given by

$$\begin{aligned}\text{ICL}_{\text{ex}}(\mathbf{Z}, \text{K}) &= \log p(\mathbf{X}, \mathbf{Z}|K) \\ &= \sum_{k,l}^{K} \log\left(\frac{\Gamma(\eta_{kl}^0 + \zeta_{kl}^0)\Gamma(\eta_{kl})\Gamma(\zeta_{kl})}{\Gamma(\eta_{kl} + \zeta_{kl})\Gamma(\eta_{kl}^0)\Gamma(\zeta_{kl}^0)}\right) + \log\left(\frac{\Gamma(\sum_{k=1}^{K} n_k^0)\prod_{k=1}^{K}\Gamma(n_k)}{\Gamma(\sum_{k=1}^{K} n_k)\prod_{k=1}^{K}\Gamma(n_k^0)}\right),\end{aligned} \tag{2.5}$$

where the components n_k are

$$n_k = n_k^0 + \sum_{i=1}^{N} Z_{ik}, \forall k \in \{1, \ldots, K\}$$

and can be regarded as pseudo counters of the number of nodes in each class. Moreover, the parameters (η_{kl}, ζ_{kl}) are given by

$$\eta_{kl} = \eta_{kl}^0 + \sum_{i \neq j}^{N} Z_{ik} Z_{jl} X_{ij}, \forall (k, l) \in \{1, \ldots, K\}^2,$$

and

$$\zeta_{kl} = \zeta_{kl}^0 + \sum_{i \neq j}^{N} Z_{ik} Z_{jl} (1 - X_{ij}), \forall (k, l) \in \{1, \ldots, K\}^2.$$

These parameters represent pseudo counters of the number of edges and the number of non-edges that connect nodes of class k to nodes of class l.

Because the calculation of $\mathrm{ICL}_{\mathrm{ex}}(\mathbf{Z}, \mathrm{K})$ involves marginalization over the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$, which have non-informative priors, the number of classes, K, is automatically penalized, and the model complexity is therefore controlled. Indeed, as highlighted by Biernacki *et al.* (2000), for standard Gaussian mixture models, the penalization terms are encompassed through the use of the gamma function. For example, replacing the gamma function $\Gamma(\cdot)$ with the Stirling approximation $\Gamma(t+1) \approx t^{t+1/2} \exp(-t)(2\pi)^{1/2}$ in the second term in (2.5) would reveal the penalization $(1/2)(K-1)\log N$ in (2.4). Similarly, replacing the first term in (2.5) with such an asymptotic approximation would reveal the penalization $(1/2)K^2 \log(N(N-1))$ in (2.4).

Note that maximizing $\mathrm{ICL}_{\mathrm{ex}}(\mathbf{Z}, \mathrm{K}) = \log \mathrm{p}(\mathbf{X}, \mathbf{Z}|\mathrm{K})$ with respect to $\mathbf{Z}$ is only equivalent to maximizing $\log p(\mathbf{Z}|\mathbf{X}, K)$, because $\log p(\mathbf{X}, \mathbf{Z}|K) = \log p(\mathbf{Z}|\mathbf{X}, K) + \log p(\mathbf{X}|K)$. Although $\log(\mathbf{X}, \mathbf{Z}|K)$ has an analytical form, $\log p(\mathbf{Z}|\mathbf{X}, K)$ does not. Therefore, existing algorithms for the SBM that rely on $p(\mathbf{Z}|\mathbf{X}, K)$ have been obliged to consider approximation techniques, such as Gibbs sampling or variational bounds, for inference purposes, whereas we consider an exact quantity here. Moreover, the $\mathrm{ICL}_{\mathrm{ex}}$ criterion is related to the variational Bayes approximation of the integrated observed data log likelihood $\log p(\mathbf{X}|K)$ proposed by Latouche *et al.* (2012). The key difference is that the parameters $(n_k, \eta_{kl}, \zeta_{kl})$ in $\mathrm{ICL}_{\mathrm{ex}}$ depend on the hard assignment $\mathbf{Z}$ of nodes to classes and not on the approximated posterior probabilities $\boldsymbol{\tau}$. Moreover, the calculation of $\mathrm{ICL}_{\mathrm{ex}}$ does not involve any entropy term.

3 Greedy optimization

Because the model parameters have been marginalized out, the $\mathrm{ICL}_{\mathrm{ex}}$ criterion involves only the cluster indicator matrix $\mathbf{Z}$, whose dimensionality depends on the number of clusters K. Thus, this integrated likelihood is a function only of a partition $\mathcal{P}$, that is, an assignment of the vertices to clusters. Directly searching for a global

maximum of ICL_{ex} is not feasible, because every possible partition of the vertices must be tested using various values of K. However heuristics are available to obtain local maxima for this combinatorial problem. These approaches have already been used for graph clustering using *ad hoc* criteria such as modularity (Newman, 2004; Blondel *et al.*, 2008) and are reminiscent of the well-known iterated conditional modes algorithm of Besag (1986) used for maximum *a posteriori* estimation in Markov random fields.

The algorithm (see Algorithm 1) begins with an SBM with $K = K_{\text{up}}$ clusters, where K_{up} is an upper bound on the number of clusters. K_{up} is assumed to be given as an input, along with an $N \times K_{\text{up}}$ matrix $\mathbf{Z}$. In practice, K_{up} is set to a large value based on user knowledge of the problem at hand, whereas $\mathbf{Z}$ can be initialized using the methods described in the next section. The algorithm then cycles randomly through all vertices of the network. At each step, a single node i is considered, while all membership vectors $\mathbf{Z}_j$ for $j \neq i$ are fixed. If i is currently in cluster g, the method searches for every possible label swap, that is, removes i from cluster g, assigns it to a cluster $h \neq g$ and then computes the corresponding change $\Delta_{g \to h}$ in the ICL_{ex} criterion. Note that $\Delta_{g \to h}$ takes two forms (see B) depending on whether cluster g is empty after the removal of i. If no label swap leads to an increase in the criterion, then the vector $\mathbf{Z}_i$ remains unchanged. Otherwise, the label swap that yields the maximal increase is applied and $\mathbf{Z}_i$ is modified accordingly. During this process, clusters may disappear, that is, their cardinality may reach zero. Each time one such modification is accepted, the model is updated and the corresponding column is removed from the cluster indicator matrix $\mathbf{Z}$. Finally, the algorithm terminates when a complete pass over the vertices does not lead to any increase in the ICL_{ex} criterion. Thus, the algorithm automatically infers the number of clusters while clustering the vertices of the network. Beginning from an over-segmented initial solution, our approach simplifies the model until a local maximum is reached.

3.1 Complexity

To construct such an algorithm, it is sufficient to know how to compute the changes in the ICL_{ex} criterion that are induced by the possible swaps (from cluster g to cluster h) for a given node i while the other nodes are fixed. Such changes can be efficiently computed (see B for details), and the complexity of identifying the best swap movement for a node is, on average, $\mathcal{O}(l + K^2)$, where l is the average number of edges per node. Such complexity can be achieved in practice, because good approximations of the logarithm of the gamma function are available with constant running time. The greedy algorithm therefore has a total complexity of $\mathcal{O}(N(l + K_{\text{up}}^2) + L)$, because the cost of a swap movement is $\mathcal{O}(l + K^2)$; the cost of the initialization of the edge counters (η_{kl}, ζ_{kl}) is L (the total number of edges in the graph), and several complete passes over the set of nodes will be performed (typically fewer than 10). Eventually, this can be simplified to $\mathcal{O}(NK_{\text{up}}^2 + L)$, because K_{up}^2 may certainly dominate l in contrast to the

Algorithm 1: Greedy ICL

```
Set K = K_up ; swap = 1 ;
Initialize the N × K_up matrix Z ; Compute η,ζ,n ;
while swap == 1 do
    V = {1, ..., N} ; swap = 0 ;
    while V not empty do
        Select a node i randomly in V ; Remove i from V ;
        If i is in cluster g, compute all terms Δ_{g→h}, ∀h ≠ g ;
        if at least one Δ_{g→h} is positive then
            swap = 1 ;
            Find h such that Δ_{g→h} is maximum ;
            Swap labels of i: Z_ig = 0 and Z_ih = 1 ;
            if g is empty then
                Remove column g in Z ; Set K = K − 1 ;
            end
            Update rows and columns (g, h) of the matrices η and ζ ;
            Update the components g and h of vector n;
        end
    end
end
Result: (Z, K)
```

complexity of $\mathcal{O}(LK_{\text{up}}^3)$ achieved using a variational algorithm and a model selection criterion as in Daudin *et al.* (2008); Latouche *et al.* (2012). Indeed, in contrast to our approach, which estimates the number of clusters within a single run while clustering the nodes, these approaches are run multiple times for various values of K, and K^* is then chosen, such that the corresponding model selection criterion is maximized. Because each run has a cost of $\mathcal{O}(LK^2)$, the overall complexity is $\mathcal{O}(LK_{\text{up}}^3)$.

3.2 Initialization and restarts

Several solutions are possible for the initialization of the algorithm; a simple choice is to sample random partitions, whereas a more relevant, but more expensive, starting point can be obtained using the k-means algorithm (using the adjacency matrix by rows as the input and a classical Euclidean distance). One possible compromise in terms of computational burden is to use only few iterations of k-means. We used the latter approach in all experiments that we conducted. Moreover, because our method is only guaranteed to reach a local optimum, a common strategy is to conduct the optimization algorithm with multiple initializations and to retain the best one based on the ICL_{ex} criterion. From a practical perspective, the sole tuning parameter that the user must provide is K_{up}, the initial number of clusters.

This choice may have an impact on the quality of the solution generated by the algorithm; a larger value may help prevent the identification of a bad local optima.

3.3 Hierarchical clustering

Thus far, we have considered simple label swaps, and the solution offered by the greedy algorithm is the local optimum with respect to the neighbouring set of candidate solutions, where the labels of each node can be changed only one at a time. Eventually, as a final step, the neighbouring set can be relaxed, thereby permitting multiple (simultaneous) label swaps, by applying merge movements between clusters. Such movements are applied if the increase in the value of the objective function can be accomplished using a greedy hierarchical algorithm at a cost of $\mathcal{O}(K^3)$ (see C for details). Because the label-swap algorithm usually considerably reduces the number of clusters ($K << K_{\text{up}}$), the computational cost of this final step is low.

4 Experiments using synthetic data

To assess the greedy optimization method, a simulation study was performed, and the solution proposed by our method was compared with those generated by available implementations of algorithms for SBM inference: **vbmod**, **mixer** and **colsbm**. For a description of these methods, refer to Section 1. In these experiments, we used the latest version of the **colsbm** code, which includes an additional movement type compared with the algorithm described in the associated publication. This movement was found to greatly enhance the results.

Our objective was to evaluate the ability of the different solutions to recover a simulated clustering *without* knowing the number of clusters. Only a reasonable upper bound K_{up} on K was provided to the algorithms when needed. Variational methods optimize a lower bound for various values of K and select K^*, such that the model selection criterion is maximized: ICL for **mixer** and ILvb for **vbmod**. Conversely, the collapsed Gibbs sampler automatically provides an estimate of K, because the posterior of K is made available.

As a baseline, we also compared our approach with a standard spectral clustering approach (Shi and Malik, 2000). In all simulations, we supplied this spectral approach using the true number of clusters.

Two indexes were used to assess the algorithm performance: The normalized mutual information (see Vinh and Epps, 2010 for details and a justification of this measure for partition comparison) and the adjusted Rand index (Hubert and Arabie, 1985). These two measures were used to compare the estimated cluster membership matrix and the simulated ground truth.

The performances were evaluated on simulated clustering problems of varying complexity and with different settings to gain insights on the influence of the number of clusters K, the number of vertices N and the type of connectivity matrix $\boldsymbol{\pi}$.

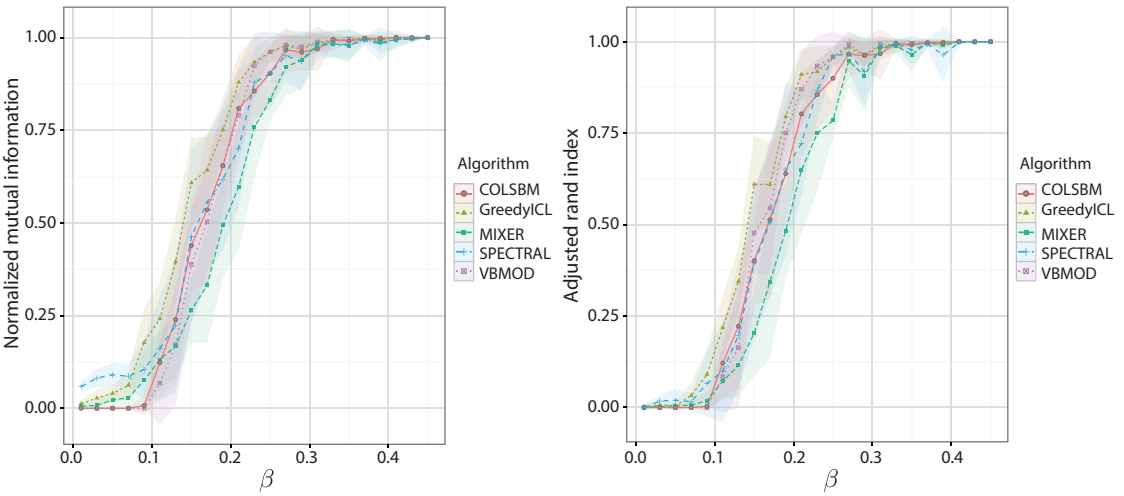

Figure 1 Means of mutual information (left) and adjusted Rand index (right) between the estimated and true cluster membership matrices using 20 simulated graphs for each value of β in $\{0.45, 0.43, \ldots, 0.03, 0.01\}$ and $N = 100$, $K = 5$, $\epsilon = 0.01$ for the algorithms **greedy ICL**, **vbmod**, **colsbm** and **mixer**. The spectral clustering approach was run with the true number of clusters as a baseline.
Source: Authors' own.

4.1 Setting 1: Small-Scale community structures

The first setting to be evaluated is a classical community simulation with $N = 100$ vertices and $K = 5$ clusters. The cluster proportions were set to $\boldsymbol{\alpha} = (1/5, 1/5, 1/5, 1/5, 1/5)$, and the connectivity matrix took a diagonal form with off-diagonal elements equal to 0.01 ($\pi_{kl} = 0.01, \forall k \neq l$) and diagonal elements given by $\pi_{kk} = \beta, \forall k$. β is a complexity-tuning parameter that ranges from 0.45 to 0.01. When β reaches 0.01, the model is not identifiable (the connectivity matrix is constant), and the true cluster memberships cannot be recovered. This model can therefore be used to simulate problems of varying complexity, from problems with a clear structure ($\beta = 0.45$) to problems without any structure ($\beta = 0.01$). The experiments were performed 20 times for each value of β, and the average values of the normalized mutual information and the adjusted Rand index over these 20 simulated graphs are depicted in Figure 1 for all algorithms, together with the standard deviation using ribbons. To ensure that the results produced were as comparable as possible, the parameters of the different algorithms were set as follows: **vbmod**, **mixer** and **greedy ICL** were all initialized 10 times, and for each method, the best run was selected based on the corresponding model selection criterion. The variational methods were run with K values between 2 and 20, and the best clustering was adopted as the final result. For **greedy ICL**, the parameters of the priors η^0, ζ^0 and n_k^0 were set to 1, and K_{up} was fixed at 20. Finally, the collapsed Gibbs sampler was run for 250,000 iterations (more than twice the default value).

The results illustrated in Figure 1 demonstrate that **greedy ICL** outperformed the other methods for complex problems, that is, low values of β. The simulated clustering was recovered until β reached 0.25. For larger values of β, the different algorithms performed identically, but beyond this limit, the results of **greedy ICL** were somewhat superior. In the transitional regime, **greedy ICL** yielded slightly better results than the other algorithms, followed by **colsbm**, **vbmod** and the spectral baseline, which yielded comparable results. **mixer** deviated from the planted clustering slightly sooner. In this simple setting, all algorithms were quite similar in terms of both normalized mutual information and adjusted Rand index.

4.2 Setting 2: Small-Scale community structures with a hub cluster

The purpose of the second setting was to explore the performances of the methods when the latent structure exhibits patterns other than a community structure. To this end, graphs were generated using the SBM with an affiliation probability matrix $\boldsymbol{\pi}$ with the following form:

$$\pi = \begin{pmatrix} \beta & \beta & \ldots & \ldots & \beta \\ \beta & \beta & \epsilon & \ldots & \epsilon \\ \beta & \epsilon & \beta & \ldots & \epsilon \\ \beta & \epsilon & \ldots & \beta & \epsilon \\ \beta & \epsilon & \ldots & \ldots & \beta \end{pmatrix}.$$

The clusters, therefore, corresponded to communities, with the exception of one cluster of hubs that was connected with probability β to all other clusters. Graphs with $N = 100$ vertices, $K = 5$ clusters and $\boldsymbol{\alpha} = (1/5, 1/5, 1/5, 1/5, 1/5)$ were generated using this connection pattern. The parameter ϵ was set to 0.01, and β ranged from 0.45 to 0.01, as before. The other simulation parameters did not change. The results are presented in Figure 2.

As expected, the **vbmod** algorithm, which searches only for communities, was strongly affected by this change of setting and systematically missed the hub cluster; the same was true of the spectral clustering. For the remaining methods, the best results were obtained using **greedy ICL**, which still recovered the planted clustering when $\beta > 0.25$, whereas the performance of **mixer** began to suffer at $\beta = 0.4$. The collapsed Gibbs sampler also yielded good results in this setting, very similar to those of **greedy ICL** and outperforming **mixer**. Eventually, the spectral clustering approach began to suffer from the same shortcoming as **vbmod** and tended to miss the hub class. Note that for difficult problems, the spectral clustering approach tends to perform slightly better. The variances of the results were comparable for all algorithms.

4.3 Setting 3: Small-Scale community structures with a hub cluster and unbalanced partitions

The third setting (see Figure 3) was identical to Setting 2 (community clusters plus a hub cluster) but with unbalanced clusters. This setting was constructed to confirm

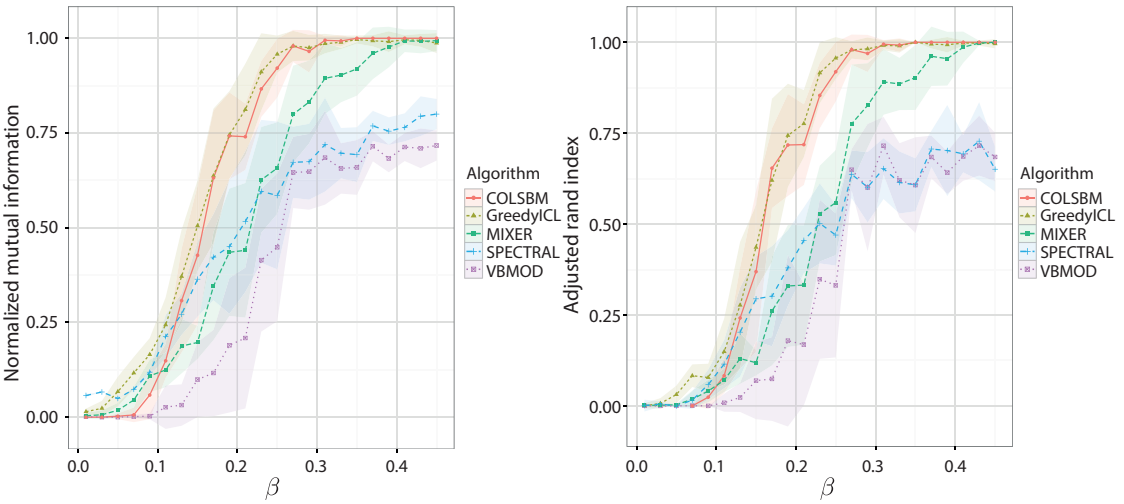

Figure 2 Mean of mutual information (left) and adjusted Rand index (right) between the estimated and true cluster membership matrices using 20 simulated graphs for each value of β in $\{0.45, 0.43, \ldots, 0.03, 0.01\}$ and $N = 100$, $K = 5$, $\epsilon = 0.01$ for the algorithms **greedy ICL**, **vbmod**, **colsbm** and **mixer**. The spectral clustering approach was run with the true number of clusters as a baseline.
Source: Authors' own.

that the proposed greedy approach would not fail under such conditions. The cluster proportions were fixed using an exponentially decreasing scheme, such that $\alpha_k \propto 0.7^k$. Thus, 36% of the nodes belonged to the largest cluster, whereas only 8.5% of the nodes, on average, belonged to the smallest one. The smallest proportion was assigned to the hub cluster to obtain a more realistic setting.

The normalized mutual information and the adjusted Rand index yielded comparable results in this setting. Because the hub cluster corresponded to a smaller proportion of the network than in the previous experiment, **vbmod** and the spectral approach were less strongly penalized, although they still exhibited lower performances than the other approaches. Consistent with the previous results, the **greedy ICL** and **colsbm** algorithms yielded better results than **mixer**. In the transitional region between simple and complex problems (near a β value of 0.2), the performance of **greedy ICL** was superior to that of **colsbm**.

4.4 Setting 4: Medium-Scale community structures with a hub cluster

The fourth setting was similar to Setting 2 (community clusters plus a hub cluster) but featured additional nodes and clusters to enable the study of the effects of these two parameters. Thus, the number of vertices was set to $N = 500$, and the number of clusters was set to $K = 10$. The cluster proportions were defined as $\boldsymbol{\alpha} = (1/10, \ldots, 1/10)$, and the values of all other parameters were kept the same as before. For this fourth experiment, the results presented in Figure 4 were very similar for **greedy ICL** and **colsbm**, which outperformed the other approaches. **Mixer** also yielded very good

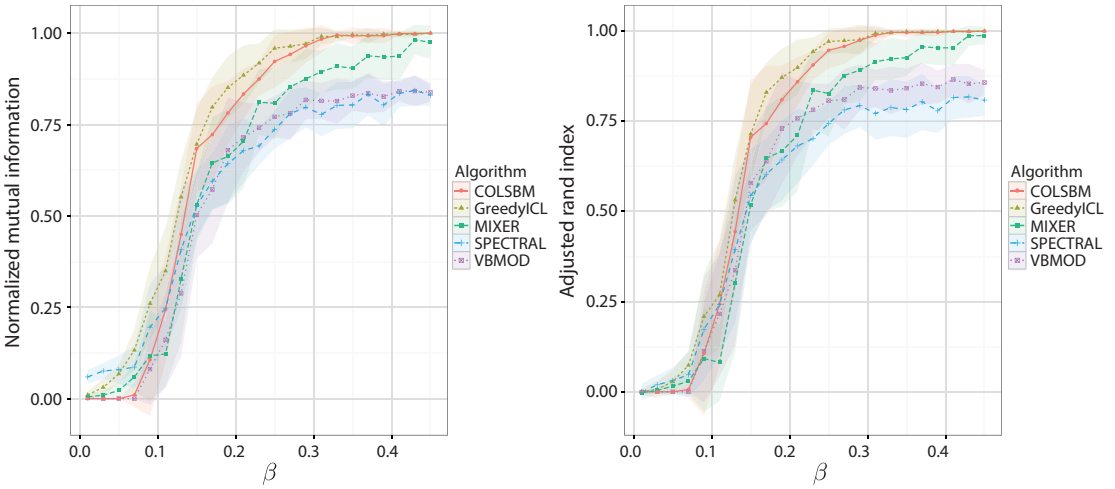

Figure 3 Means of mutual information (left) and adjusted Rand index (right) between the estimated and true cluster membership matrices using 20 simulated graphs for each value of β in $\{0.45, 0.43, \ldots, 0.03, 0.01\}$ and $N = 100$, $K = 5$, $\epsilon = 0.01$ and $\boldsymbol{\alpha} = (0.36, 0.25, 0.17, 0.12, 0.08)$ for the algorithms **greedy ICL**, **vbmod**, **colsbm** and **mixer**. The spectral clustering approach was run with the true number of clusters as a baseline.
Source: Authors' own.

results until β reached 0.3, at which point the quality of the results began to decrease quite rapidly. Although the spectral baseline approach and **vbmod** did not recover the exact planted partitions even when the problem was simple (β values near 0.4), they outperformed **mixer** when the planted structure was not particularly strong.

The results obtained using the various algorithms in this scenario were better than those obtained previously (in terms of both normalized mutual information and adjusted Rand index). The variances in the results were lower than for Setting 2. This result is readily explained by the increase in the number of nodes per cluster. The transitions between high and low values of the normalized mutual information and the adjusted Rand index were also sharper than in the previous experiments, for the same reason.

4.5 Setting 5: Large-Scale problem with complex structure

The final tested setting involved larger graphs with $N = 10\,000$ vertices. The planted structure was not a pure community pattern. Some interactions between clusters were activated randomly using a Bernoulli distribution, as described by the following generative model:

$$\pi_{kl} = \begin{cases} ZU + (1 - Z)\epsilon, \text{ if } k \neq l \\ U, \text{ if } k = l, \end{cases} \tag{4.1}$$

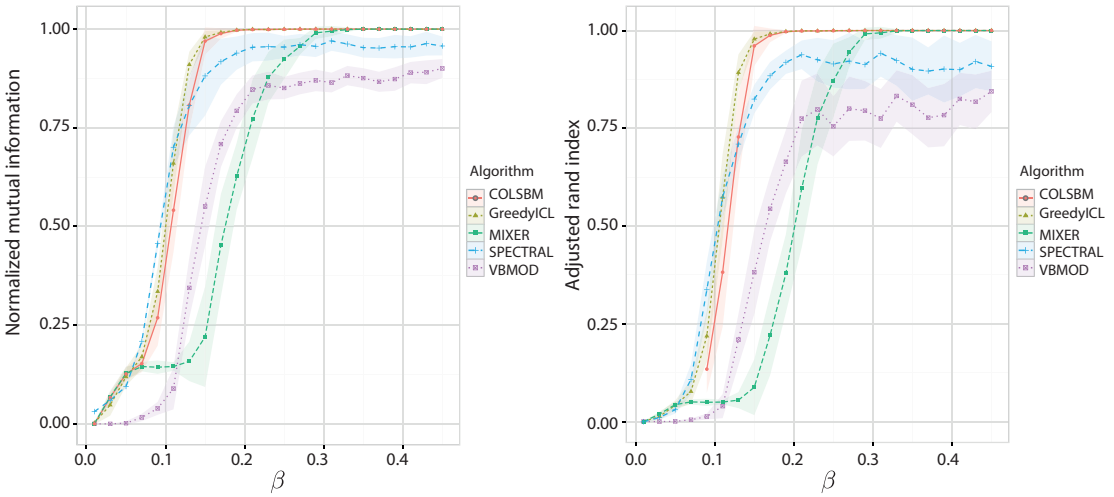

Figure 4 Means of mutual information (left) and adjusted Rand index (right) between the estimated and true cluster membership matrices using 20 simulated graphs for each value of β in $\{0.45, 0.43, \ldots, 0.03, 0.01\}$ and $N = 500$, $K = 10$, $\epsilon = 0.01$ for the algorithms **greedy ICL**, **vbmod**, **colsbm** and **mixer**. The spectral clustering approach was run with the true number of clusters as a baseline.
Source: Authors' own.

with $Z \sim \mathcal{B}(0.1)$, $U \sim \mathcal{U}(0.45)$ and $\epsilon = 0.01$. Because of the size of the problem and the complex nature of the underlying structure, only four algorithms were appropriate for these graphs, namely, **greedy ICL**, **colsbm**, **vbmod** and spectral clustering. **mixer** was not tested because it is not compatible with such large graphs. All approaches were used to cluster 20 simulated graphs generated using this scheme. The greedy algorithm was initialized using $K_{\text{up}} = 100$ and the same parameters used previously for the prior distributions. The results, which are presented as boxplots in Figure 5, reveal that **greedy ICL** exhibited a clear advantage over all other methods. Specifically, **greedy ICL** achieved an average normalized mutual information value of 0.88, whereas **colsbm** achieved only a value of 0.67. In fact, the greedy solution yielded approximately 80 clusters for all simulations, whereas the Gibbs sampler yielded more than 240 clusters on average, and therefore produced highly over-segmented partitions of the graphs. Although they were supplied with the true number of clusters, the other two approaches, namely, **vbmod** and the spectral method, produced results that were clearly inferior to those of **greedy ICL**, with average normalized mutual information values of approximately 0.71 for the spectral method and 0.66 for **vbmod**.

In summary, the results of these experiments indicate that **greedy ICL** compares favourably with the other existing solutions for SBM inference in all settings. The results obtained in complex settings, that is, large graphs and a complex underlying structure (Setting 5), are particularly encouraging because **greedy ICL** clearly outperformed the collapsed Gibbs sampler and the other solutions, even when these methods were provided with the true number of clusters.

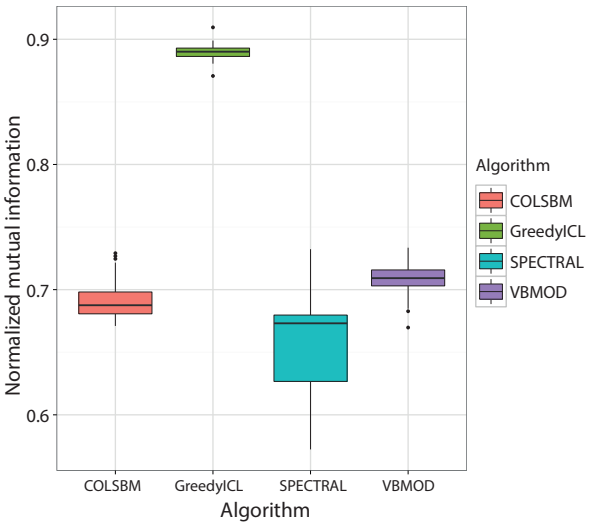

Figure 5 Means of the mutual information between the estimated and true cluster membership matrices based on 20 simulated graphs with $N = 10\,000$ and $K = 50$. The spectral clustering approach and **vbmod** were run using the true number of clusters as a baseline.
Source: Authors' own.

5 Real dataset: Communities of blogs

Finally, the proposed algorithm was tested on a real network in which the vertices corresponded to blogs and the edges corresponded to known hyperlinks between blogs. All blogs included in the network were related to a common topic, that is, illustrations and comics.

The network was constructed using a community extraction procedure (Côme and Diemert, 2010) that begins with known seeds and expands from them to identify a dense core of nodes surrounding the seeds. The network consisted of 1360 blogs linked by 33,805 edges. The dataset was expected to exhibit specific patterns, namely, communities, such that two blogs of the same community were more likely to be connected than nodes of different communities. To test this hypothesis, we performed a qualitative comparison of the results of the greedy ICL algorithm and the community discovery method of Blondel *et al.* (2008).

Beginning with $K_{\text{up}} = 100$ clusters, the greedy ICL identified $K = 37$ clusters. The corresponding clusters are illustrated in Figure 6, which presents an image of the adjacency matrix with rows/columns sorted by cluster number. Thus, it appears that the vast majority of the identified clusters correspond to small sub-communities. These sub-communities, all correspond to known groups. For instance, a group of blogs of illustrators for Disney was identified. Other examples include clusters of blogs of students who attended the same illustration school, such as the ECMA in Angoulême or 'Gobelins L'École de L'Image.' However, some clusters had more complex connectivity

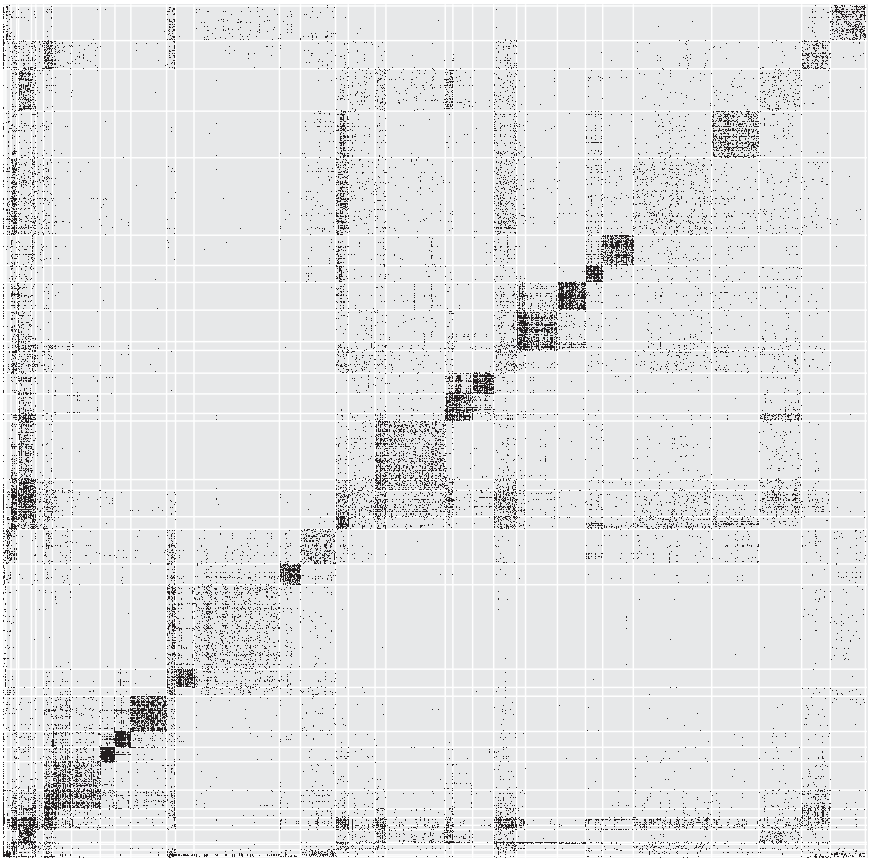

Figure 6 Adjacency matrix of the network of blogs; the rows/columns are sorted by cluster number based on the clusters identified by the greedy ICL algorithm. The cluster boundaries are depicted as white lines.
Source: Authors' own.

structures and consisted of hubs with high connectivity to blogs of different clusters. These clusters corresponded to the blogs of famous writers, such as Boulet.

To provide a qualitative understanding of the level of interest of the identified clustering, we also report the results obtained using the community discovery algorithm of Blondel *et al.* (2008) in Figure 7. Using this approach, only eight clusters were identified, all of which corresponded to sub-communities. Clusters of hubs could not be recovered. The substantial difference between the numbers of clusters estimated by the two methods may be explained by two factors. First, modularity is prone to a resolution-limit problem (Fortunato and Barthélemy, 2007), which prevents such a solution from extracting small-scale structures. This problem explains why the small sub-communities extracted by **greedy ICL** were not recovered using the modularity. The behaviour of the $\mathrm{ICL}_{\mathrm{ex}}$ criterion with respect to the resolution-limit problem is not clear and requires further investigation. However, we observed that when the proposed criterion was applied to this dataset, finer structures than those obtained

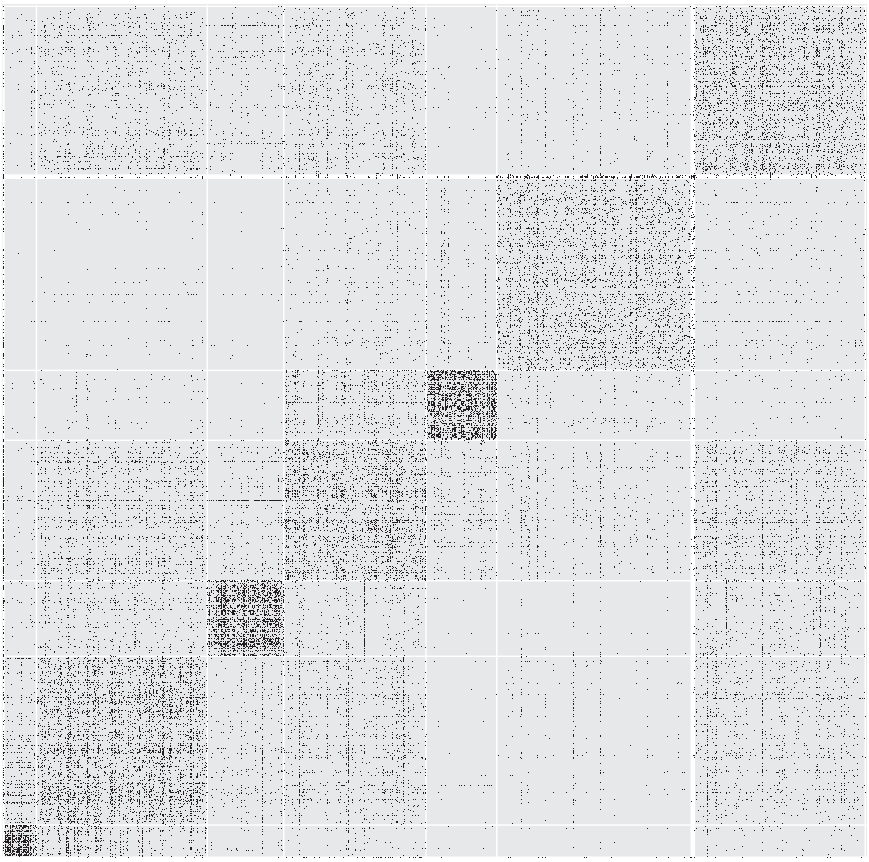

Figure 7 Adjacency matrix of the network of blogs; the rows/columns are sorted by cluster number based on the clusters identified via modularity optimization. The cluster boundaries are depicted as white lines.
Source: Authors' own.

using modularity were recovered. Second, the difference in the manner in which the two criteria use or do not use degree correction (Karrer and Newman, 2011) may also explain the disparity in the number of identified clusters. Whereas modularity is a degree-corrected criterion that downscales the weights of the edges between highly connected vertices, the ICL_{ex} criterion for the basic SBM used here is not. Whether a degree correction is applied is a modelling choice that merits investigation and validation; however, it seems that even without degree correction, the results obtained by **greedy ICL** are meaningful, particularly the identification of hub clusters.

6 Conclusion

In this article, we considered an analytical expression of the integrated complete data log likelihood. We then proposed a greedy optimization algorithm to maximize this

exact quantity. Starting from an over-segmented partition, this approach simultaneously simplifies the model and clusters the vertices until a local maximum is reached. This greedy algorithm has a competitive complexity and is capable of handling networks with tens of thousands of vertices and millions of edges. We demonstrated using simulated data that the method is an improvement over existing graph-clustering algorithms in terms of both model selection and clustering of vertices. A qualitative comparison between methods was also performed using an original network constructed from blogs related to illustration, comics and animation.

Appendix

A Integrated complete data log likelihood

Using factorized and conjugate prior distributions over the model parameters, the integrated complete data log likelihood is given as follows:

$$\log p(\mathbf{X}, \mathbf{Z}|K) = \sum_{k,l}^{K} \log\left(\frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)}\right) + \log\left(\frac{C(\mathbf{n})}{C(\mathbf{n}^0)}\right),$$

where

- $\eta_{kl} = \eta_{kl}^0 + \sum_{i \neq j}^{N} Z_{ik} Z_{jl} X_{ij}$ for all (k, l) in $\{1, \ldots, K\}^2$
- $\zeta_{kl} = \zeta_{kl}^0 + \sum_{i \neq j}^{N} Z_{ik} Z_{jl} (1 - X_{ij})$ for all (k, l) in $\{1, \ldots, K\}^2$
- the components of the vector $\mathbf{n}$ are $n_k = n_k^0 + \sum_{i=1}^{N} Z_{ik}$, for all k in $\{1, \ldots, K\}$
- the function $B(a, b)$ is such that $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ for all (a, b) in $\mathbb{R}^2$
- the function $C(\cdot)$ is such that $C(\mathbf{x}) = \frac{\prod_{k=1}^{K} \Gamma(x_k)}{\Gamma(\sum_{k=1}^{K} x_k)}$ for all $\mathbf{x}$ in $\mathbb{R}^K$.

Proof: Considering factorized prior distributions, the integrated complete data log likelihood decomposes into two terms:

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{Z}|K) &= \log\left(\int_{\boldsymbol{\alpha}, \boldsymbol{\pi}} p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\alpha}|K) d\boldsymbol{\alpha} d\boldsymbol{\pi}\right) \\ &= \log\left(\int_{\boldsymbol{\pi}} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\pi}, K) p(\boldsymbol{\pi}|K) d\boldsymbol{\pi} \int_{\boldsymbol{\alpha}} p(\mathbf{Z}|\boldsymbol{\alpha}, K) p(\boldsymbol{\alpha}|K) d\boldsymbol{\alpha}\right) \\ &= \log p(\mathbf{X}|\mathbf{Z}, K) + \log p(\mathbf{Z}|K). \end{aligned} \tag{A.1}$$

The first term in (A.1) can be obtained as follows:

$$\begin{aligned}
p(\mathbf{X}|\mathbf{Z}, K) &= \int_{\boldsymbol{\pi}} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\pi}, K) p(\boldsymbol{\pi}|K) d\boldsymbol{\pi} \\
&= \int_{\boldsymbol{\pi}} \Big(\prod_{k,l}^{K} \pi_{kl}^{\sum_{i \neq j}^{N} Z_{ik} Z_{jl} X_{ij}} (1 - \pi_{kl})^{\sum_{i \neq j}^{N} Z_{ik} Z_{jl} (1 - X_{ij})} \Big) \\
&\times \prod_{k,l}^{K} \frac{1}{B(\eta_{kl}^0, \zeta_{kl}^0)} \pi_{kl}^{\eta_{kl}^0 - 1} (1 - \pi_{kl})^{\zeta_{kl}^0 - 1} d\boldsymbol{\pi} \\
&= \prod_{k,l}^{K} \left(\frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)} \int_{\pi_{kl}} \text{Beta}(\pi_{kl}; \eta_{kl}, \zeta_{kl}) d\pi_{kl} \right) \\
&= \prod_{k,l}^{K} \frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)}.
\end{aligned}$$

The second term in (A.1) can be obtained as follows:

$$\begin{aligned}
p(\mathbf{Z}|K) &= \int_{\boldsymbol{\alpha}} p(\mathbf{Z}|\boldsymbol{\alpha}, K) p(\boldsymbol{\alpha}|K) d\boldsymbol{\alpha} \\
&= \int_{\boldsymbol{\alpha}} \Big(\prod_{k=1}^{K} \alpha_k^{\sum_{i=1}^{N} Z_{ik}} \Big) \frac{1}{C(\mathbf{n}^0)} \prod_{k=1}^{K} \alpha_k^{n_k^0 - 1} d\boldsymbol{\alpha} \\
&= \frac{C(\mathbf{n})}{C(\mathbf{n}^0)} \int_{\boldsymbol{\alpha}} \text{Dir}(\boldsymbol{\alpha}; \mathbf{n}) d\boldsymbol{\alpha} \\
&= \frac{C(\mathbf{n})}{C(\mathbf{n}^0)}.
\end{aligned}$$

Finally,

$$\log p(\mathbf{X}, \mathbf{Z}|K) = \sum_{k,l}^{K} \log \left(\frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)} \right) + \log \left(\frac{C(\mathbf{n})}{C(\mathbf{n}^0)} \right).$$

B Change in ICL induced by a swap movement $i : g \to h$

In each step of the greedy ICL algorithm, a single node i is considered. If i is currently in cluster g, the method tests every possible label swap $g \to h$, that is, it removes i from cluster g and assigns it to a cluster $h \neq g$. The corresponding change in the ICL_{ex} criterion is denoted by $\Delta_{g \to h}$. To calculate each term $\Delta_{g \to h}$ for all $h \neq g$, we

consider two cluster indicator matrices, $\mathbf{Z}$ and $\mathbf{Z}^{\text{test}}$. $\mathbf{Z}$ describes the current partition of the vertices in the network, whereas $\mathbf{Z}^{\text{test}}$ represents the partition after the swap $g \to h$ is applied:

$$\begin{cases} \mathbf{Z}_j^{\text{test}} = \mathbf{Z}_j, \forall j \neq i \\ \\ Z_{ik}^{\text{test}} = Z_{ik} = 0, \forall k \neq g, h, \end{cases}$$

while

$$\begin{cases} Z_{ig}^{\text{test}} = 0, Z_{ig} = 1 \\ \\ Z_{ih}^{\text{test}} = 1, Z_{ih} = 0. \end{cases}$$

Thus,

$$\Delta_{g \to h} = \text{ICL}_{\text{ex}}(\mathbf{Z}^{\text{test}}, K^{\text{test}}) - \text{ICL}_{\text{ex}}(\mathbf{Z}, \text{K}).$$

Note that $\Delta_{g \to h}$ takes two forms depending on whether cluster g is empty after the removal of i. If g is empty, then the model dimensionality changes ($K^{\text{test}} = K - 1$), and this change in dimensionality must be taken into account when evaluating the potential increase induced by the swap movement.

B.1 Case 1: $\sum_i Z_{ig}^{\text{test}} > 0$, cluster g is not empty after the removal of i

$$\begin{aligned} \Delta_{g \to h} &= \log\left(\frac{C(\mathbf{n}^{\text{test}})}{C(\mathbf{n})}\right) + \sum_{k,l}^{K} \log\left(\frac{B(\eta_{kl}^{\text{test}}, \zeta_{kl}^{\text{test}})}{B(\eta_{kl}, \zeta_{kl})}\right) \\ &= \log\left(\frac{\Gamma(n_g^{\text{test}})\Gamma(n_h^{\text{test}})}{\Gamma(n_g)\Gamma(n_h)}\right) + \sum_{l=1}^{K} \sum_{k \in \{g,h\}} \log\left(\frac{B(\eta_{kl}^{\text{test}}, \zeta_{kl}^{\text{test}})}{B(\eta_{kl}, \zeta_{kl})}\right) \\ &\quad + \sum_{k \notin \{g,h\}} \sum_{l \in \{g,h\}} \log\left(\frac{B(\eta_{kl}^{\text{test}}, \zeta_{kl}^{\text{test}})}{B(\eta_{kl}, \zeta_{kl})}\right) \\ &= \log\left(\frac{\Gamma(n_g - 1)\Gamma(n_h + 1)}{\Gamma(n_g)\Gamma(n_h)}\right) + \sum_{l=1}^{K} \sum_{k \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) \end{aligned}$$

$$
\begin{aligned}
&+ \sum_{k \notin \{g,h\}} \sum_{l \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&= \log\left(\frac{n_h}{n_g - 1}\right) + \sum_{l=1}^{K} \sum_{k \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&+ \sum_{k \notin \{g,h\}} \sum_{l \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right),
\end{aligned}
$$

where $\delta_{kl}^{(i)}$ the change in the edge counter η_{kl} induced by the label swap:

$$
\begin{aligned}
\delta_{kl}^{(i)} = {} & \mathbb{1}_{\{k=h\}} \sum_{j \neq i}^{N} Z_{jl} X_{ij} + \mathbb{1}_{\{l=h\}} \sum_{j \neq i}^{N} Z_{jk} X_{ji} - \mathbb{1}_{\{k=g\}} \sum_{j \neq i}^{N} Z_{jl} X_{ij} \\
& - \mathbb{1}_{\{l=g\}} \sum_{j \neq i}^{N} Z_{jk} X_{ji}.
\end{aligned}
$$

Moreover, $\rho_{kl}^{(i)}$ is defined as follows:

$$
\rho_{kl}^{(i)} = \left(\mathbb{1}_{\{k=h\}} - \mathbb{1}_{\{k=g\}}\right)(n_l - n_l^0 - Z_{il}) + \left(\mathbb{1}_{\{l=h\}} - \mathbb{1}_{\{l=g\}}\right)(n_k - n_k^0 - Z_{ik}) - \delta_{kl}^{(i)}.
$$

These updated quantities can be computed in $O(l_i)$, where l_i is the degree of i (total number of edges from and to i). Therefore, the average complexity of identifying the best swap movement for a node is $\mathcal{O}(l + K^2)$, where l is the average degree of the network for computing $\delta_{kl}^{(i)}$ and K^2 is the complexity of computing Δ_{swap} with all possible h labels and identifying the best one.

B.2 Case 2: $\sum_i Z_{ig}^{\text{test}} = 0$, cluster g disappears

In this case, the dimensionality of $\mathbf{n}^0$ changes, and we will denote the corresponding vector of size $K - 1$ by $\mathbf{n}^{0*} = (n^0, \ldots, n^0)$:

$$
\begin{aligned}
\Delta_{g \to h} = {} & \log\left(\frac{C(\mathbf{n}^0)}{C(\mathbf{n})} \frac{C(\mathbf{n}^{\text{test}})}{C(\mathbf{n}^{0*})}\right) \\
& + \sum_{\substack{(k,l) \neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0, \zeta_{kl}^0)}{B(\eta_{kl}, \zeta_{kl})}\right)
\end{aligned}
$$

$$= \log\left(\frac{n_h}{n^0}\frac{\Gamma\left((K-1)n^0\right)\Gamma(Kn^0+N)}{\Gamma(K\,n^0)\Gamma((K-1)n^0+N)}\right) + \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)},\,\zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\,\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\,\zeta_{kl}^0)}{B(\eta_{kl},\,\zeta_{kl})}\right).$$

The complexity in this case is the same as previously, that is, $\mathcal{O}(l+K^2)$.

C Change in ICL induced by a merge movement

$$\begin{aligned}\Delta_{g\cup h} &= \log\left(\frac{C(\mathbf{n}^0)}{C(\mathbf{n})}\frac{C(\mathbf{n}^{\text{test}})}{C(\mathbf{n}^{0*})}\right) + \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)},\,\zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\,\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\,\zeta_{kl}^0)}{B(\eta_{kl},\,\zeta_{kl})}\right) \\ &= \log\left(\Gamma(n^0)\frac{\Gamma\left((K-1)n^0\right)\Gamma(Kn^0+N)}{\Gamma(K\,n^0)\Gamma((K-1)n^0+N)}\frac{\Gamma(n_h+n_g-n^0)}{\Gamma(n_g)\Gamma(n_h)}\right) \\ &\quad + \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)},\,\zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\,\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\,\zeta_{kl}^0)}{B(\eta_{kl},\,\zeta_{kl})}\right)\end{aligned}$$

where $\delta_{kl}^{(i)}$ is the change in the edge counter η_{kl} induced by the merge:

$$\delta_{kl}^{(i)} = \mathbb{1}_{\{k=h\}}(\eta_{gl}-\eta_{gl}^0) + \mathbb{1}_{\{l=h\}}(\eta_{kg}-\eta_{kg}^0) + \mathbb{1}_{\{k=h \text{ and } l=h\}}(\eta_{gg}-\eta_{gg}^0). \tag{C.1}$$

Moreover, $\rho_{kl}^{(i)}$ is defined as follows:

$$\rho_{kl}^{(i)} = \mathbb{1}_{\{k=h\}}(\zeta_{gl}-\zeta_{gl}^0) + \mathbb{1}_{\{l=h\}}(\zeta_{kg}-\zeta_{kg}^0) + \mathbb{1}_{\{k=h \text{ and } l=h\}}(\zeta_{gg}-\zeta_{gg}^0). \tag{C.2}$$

References

Adamic L and Glance N (2005) The political blogosphere and the 2004 US election. In Proceedings of the *WWW Workshop on the Weblogging Ecosystem*. Chicago, IL, USA.

Airoldi E, Blei D, Xing E and Fienberg S (2006) Mixed membership stochastic block models for relational data with application to protein–protein interactions. In *Proceedings of the International Biometrics Society Annual Meeting*. Monteral, Quebec, Canada.

Airoldi E, Blei D, Fienberg SE and Xing EP (2007) Mixed membership analysis of high-throughput interaction studies: Relational data. *2007, ArXiv e-prints*.

Airoldi EM, Blei DM, Fienberg SE and Xing EP (2008) Mixed-membership stochastic blockmodels. *Journal of Machine Learning Research*, **9**, 1981–2014.

Barabási AL and Oltvai ZN (2004) Network biology: understanding the cell's functional organization. *Nature Reviews. Genetics*, **5**, 101–13.

Besag J (1986) On the statistical analysis of dirty pictures (with discussions). *Journal of the Royal Statistical Society, Series B*, **48**, 259–302.

Bickel PJ and Chen A (2009) A non parametric view of network models and Newman–Girvan and other modularities. *Proceedings of the National Academy of Sciences*, **106**, 21068–73.

Biernacki C, Celeux G and Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **7**, 719–25.

Biernacki C, Celeux G and Govaert G (2010) Exact and Monte Carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, **140**, 2991–3002.

Blondel VD, Guillaume JL, Lambiotte R and Lefebvre E (2008) Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **10**, 10008–20.

Boulet R, Jouve B, Rossi F and Villa N (2008) Batch kernel SOM and related Laplacian methods for social network analysis. *Neurocomputing*, **71**, 1257–73.

Côme E and Diemert E (2010) The noise cluster model, a greedy solution to the network community extraction problem. *Information Intelligence Interaction*, **11**, 40–59.

Daudin J, Picard F and Robin S (2008) A mixture model for random graph. *Statistics and computing*, **18**, 1–36.

Estrada E and Rodriguez-Velazquez JA (2005) Spectral measures of bipartivity in complex networks. *Physical Review E*, **72**, 046105.

Fienberg S and Wasserman S (1981) Categorical data analysis of single sociometric relations. *Sociological Methodology*, **12**, 156–92.

Fortunato S and Barthélemy M (2007) Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, **104**, 36–41.

Girvan M and Newman MEJ (2002) Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*, **99**, 7821–26.

Goldenberg A, Zheng A, Fienberg S and Airoldi E (2010) A survey of statistical network models. *Foundations and Trends in Machine Learning*, **2**, 129–233.

Handcock MS, Raftery AE and Tantrum JM (2007) Model-based clustering for social networks. *Journal of the Royal Statistical Society*, **170**, 1–22.

Hoff P, Raftery A and Handcock M (2002) Latent space approaches to social network analysis. *Journal of the Royal Statistical Society*, **97**, 1090–98.

Hofman J and Wiggins C (2008) A Bayesian approach to network modularity. *Physical Review Letters*, **100**, 258701–9900.

Holland J, Laskey K and Leinhard S (1993) Stochastic block models: First steps. *Social Networks*, **5**, 109–37.

Hubert L and Arabie P (1985) Comparing partitions. *Journal of classification*, **2**, 193–218.

Jeffreys H (1946) An invariant form for the prior probability in estimations problems. In *Proceedings of the Royal Society of London. Series A*, **186**, 453–61.

Karrer B and Newman MEJ (2011) Stochastic blockmodels and community structure in networks. *Physical Review E*, **83**, 016107.

Krivitsky PN, Handcock MS, Raftery AE and Hoff PD (2009) Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, **31**, 204–13.

Lacroix V, Fernandes C and Sagot MF (2006) Motif search in graphs: Application to metabolic networks. *Transactions in Computational Biology and Bioinformatics*, **3**, 360–68.

Latouche P, Birmelé E and Ambroise C (2009) Bayesian methods for graph clustering. *Advances in Data Analysis Data Handling*

and Business Intelligence, pages 229–39. Springer.

Latouche P, Birmelé E and Ambroise C (2011) Overlapping stochastic block models with application to the french political blogosphere. *Annals of Applied Statistics*, **5**, 309–36.

Latouche, Birmelé E and Ambroise C (2012) Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, **12**, 93–115.

Mariadassou M, Robin S and Vacher C (2010) Uncovering latent structure in valued graphs: A variational approach. *Annals of Applied Statistics*, **4**, 715–42.

Mc Daid A, Murphy TNF and Hurley N (2013) Improved Bayesian inference for the stochastic block model with application to large networks. *Computational Statistics and Data Analysis*, **60**, 12–31.

Milo R, Shen-Orr S, Itzkovitz S, Kashtan D, Chklovskii D and Alon U (2002) Network motifs: Simple building blocks of complex networks. *Science*, **298**, 824–27.

Moreno, JL (1934) *Who shall survive? A new approach to the problem of human interrelations*. Washington DC: Nervous and Mental Disease Publishing.

Newman M and Leicht E (2007) Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, **104**, 9564–69.

Newman, MEJ (2004) Fast algorithm for detecting community structure in networks. *Physical Review Letter E*, **69**, 0066133.

Newman MEJ (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, **103**, 8577–82.

Nobile A and Fearnside A (2007) Bayesian finite mixtures with an unknown number of components: the allocation sampler. *Statistics and Computing*, **17**, 147–62.

Nowicki K and Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, **96**, 1077–87.

Palla G, Barabási A and Vicsek T (2007) Quantifying social group evolution. *Nature*, **446**, 664–67.

Shi J and Malik J (2000) Normalized cuts and image segmentation. *IEEE Transactions on PAMI*, **22**, 888–905.

Vinh NX and Epps J (2010) Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, **11**, 2837–54.

Watts DJ and Strogatz SH (1998) Collective dynamics of small-world networks. *Nature*, **393**, 440–42.

White H, Boorman S and Breiger R (1976) Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology*, **81**, 730–80.

White JG, Southgate E, Thompson JN, and Benner S (1986) The structure of the nervous system of the nematode caenorhabditis elegans. *Philosophical Transactions. Royal Society London B*, **314**, 1–340.

Zanghi H, Ambroise C and Miele V (2008) Fast online graph clustering via Erdos–Renyi mixture. *Pattern Recognition*, **41**, 3592–99.

Zanghi H, Picard F, Miele, V and Ambroise C (2010) Strategies for online inference of network mixture. *Annals of Applied Statistics*, **4**, 687–714.