Model-based graph clustering of a collection of networks using an agglomerative algorithm

Tabea Rebafka

Model-based graph clustering of a collection of networks using an agglomerative algorithm

Tabea Rebafka

Laboratoire de Probabilités, Statistique et Modélisation
Sorbonne Université, Université Paris Cité, CNRS
Paris, France
tabea.rebafka@sorbonne-universite.fr

Abstract

Graph clustering is the task of partitioning a collection of observed networks into groups of similar networks. Here similarity means networks have a similar structure or graph topology. To this end, a model-based approach is developed, where the networks are modelled by a finite mixture model of stochastic block models. Moreover, a computationally efficient clustering algorithm is developed. The procedure is an agglomerative hierarchical algorithm that maximizes the so-called integrated classification likelihood criterion. The bottom-up algorithm consists of successive merges of clusters of networks. Those merges require a means to match block labels of two stochastic block models to overcome the label-switching problem. This problem is addressed with a new distance measure for the comparison of stochastic block models based on their graphons. The algorithm provides a cluster hierarchy in form of a dendrogram and valuable estimates of all model parameters.

Keywords Graph clustering, multiple networks, stochastic block model, agglomerative algorithm, graphon, integrated classification likelihood, network comparison, node matching, clustering hierarchy

1 Introduction

Entire collections of networks are more and more available in various fields of application. In medical research, for instance, each patient may be described by her or his personal metabolic network. In ecology, the interactions of species in different ecosystems are represented by ecological networks like foodwebs. In social sciences or communication, social interactions of individuals are observed for many different communities. Though there is

a lack of statistical methods to analyze those data, as the literature mainly deals with the analysis of a single graph, which is already a challenging problem notably due to the complex dependencies inherent to relational data.

To analyze a huge collection of networks, we may wish to cluster them into groups of similar networks. To do so, it is necessary to define what it is means that networks are similar. This question is related to the problem of graph comparison. As networks have complex structure, do not necessarily share the same nodes and can be of different sizes, comparing networks is not a trivial task. A conventional approach is to use a graph embedding, that is, a vector representation of the network. A graph embedding may be a number of hand-designed characteristics as densities of edges or triangles, clustering coefficients or the network's diameter. With a graph embedding at hand, any standard machine learning clustering procedure as k-means, DBSCAN or GMM can be applied to obtain a partitioning of the observed networks. The main drawback of this approach is that the clustering result heavily depends on the chosen graph embedding. Indeed, many network indicators are highly correlated. Moreover, a graph embedding that is meaningful in one application may yield poor results in a different context (Botella et al., 2022).

In this work, we take a different view point by focusing on a comparison in terms of graph topology. We propose to directly compare networks by their structure or general organization. We emphasize that no node correspondance among the different networks is assumed. Our method is a model-based clustering approach, where a statistical model is introduced to describe the collection of networks. More precisely, we introduce a finite mixture model of random graph models. This means that we search clusters of networks that are all generated according to a common probabilistic model explaining the similar graph topology. In a mixture model the task of graph clustering becomes an inference problem.

For the random graph model describing the networks of a cluster, we choose the popular stochastic block model (SBM) (Nowicki and Snijders, 2001). The SBM is a highly flexible model offering a large variety of graph topologies. Its is especially suitable for heterogeneous networks as often encountered in applications. A further advantage of the SBM is the interpretability of its model parameters. Today numerous variants of the SBM exist (binary, valued, including covariates, degree-corrected, multipartite, dynamic versions, overlapping, mixed membership) emphasizing the relevance of the model. For a review on the SBM we refer to Matias and Robin (2014).

The SBM is a discrete latent variable model and parameter estimation is challenging. Several inference algorithms have been proposed as a variational EM-algorithm (Daudin et al., 2008), MCMC methods (Peixoto, 2014), a pseudo-likelihood approach (Amini et al., 2013), a Bayesian approach based on the integrated classification likelihood (ICL) (Côme and Latouche, 2015), or more recently a variational autoencoder using neural networks (Mehta et al., 2019). These algorithms can be time-consuming and may not be scalable to networks with a very large number of nodes. While some of them are fast for a single run, they do not provide stable solutions and many runs are required. The problem of model selection, that is, the choice of the best number of latent blocks of the SBM, increases the difficulty considerably.

Modelling a collection of networks by a mixture model of SBMs, as we propose, is obviously much more involved than using a simple SBM for a single network. Consequently, fitting such a mixture model to the data is challenging and raises a number of issues. In this paper we develop a hierarchical algorithm, which is in line with the method of Côme and Latouche (2015) for general discrete latent variable models. The idea is to maximize the so-called integrated classification likelihood (ICL) criterion by an agglomerative procedure. The algorithm automatically selects the best number of network clusters. Moreover, a complete cluster hierarchy in form of a dendrogramm is provided, which is very appealing for the interpretation of the results.

The proposed bottom-up algorithm consists in successive merges of clusters. This amounts to merge the corresponding SBMs. However, this raises a concern related to the label-switching problem in the SBM. To address this problem, we propose a new tool based on the graphon of a SBM. More precisely, we introduce a distance measure to compare two SBMs that can be used to match block labels in a computationally efficient way.

The algorithm does not only cluster the networks, but also provides estimates of all model parameters. And as the SBM is a highly interpretable model, the SBM parameters of a cluster give valuable insights on the topology and characteristics of the networks in that cluster.

The contributions of this paper are as follows.

- A finite mixture model of SBMs is introduced to model a collection of networks (Section 2.
- A hierarchical algorithm to cluster the networks and estimate model parameters is developed (Section 3 and 4).
- We propose a new tool to match block labels of two SBMs. This tool may have a more general interest beyond the graph clustering context (Section 5).
- Details for an efficient implementation of the algorithm are provided (Section 6 and 7).

2 Mixture of stochastic block models

In this section we first recall the definition of the classical SBM for a single network. Then we introduce the mixture of SBM for a collection of networks. Throughout the paper we consider directed binary networks without self-loops, but extensions to other types of networks are straightforward.

Chabert-Liddell et al. (2022) also consider the modelling of a collection of networks using the SBM. Their model variants all include a common SBM parameter and allow for individual variations of this parameter from one network to another. However, their models do not encompass the finite mixture model of SBMs proposed below (Subsection 2.2).

2.1 Stochastic block model for a single network

Consider a network with n vertices. Denote $(\boldsymbol{\pi}, \boldsymbol{\gamma})$ the parameters of a SBM with K blocks, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K) \in (0,1)^K$ are the block proportions with $\sum_{k \in [\![K]\!]} \pi_k = 1$ and $\boldsymbol{\gamma} = (\gamma_{k,l})_{k,l} \in (0,1)^{K \times K}$ the connectivity matrix. Let $\mathbf{Z} = (Z_1, \ldots, Z_n) \in [\![K]\!]^n$ be a vector of independent discrete latent variables for the nodes, with $\mathbb{P}(Z_i = k) = \pi_k$ for all $k \in [\![K]\!]$ and $i \in [\![n]\!]$. When convient, we use the one-hot encoding $Z_i = (Z_{i,1}, \ldots, Z_{i,K}) \in \{0,1\}^K$, where Z_i has multinomial distribution $\mathcal{M}(1, \boldsymbol{\pi})$. In the SBM, conditionally on the block memberships $\mathcal{Z} = (Z)_{m \in [\![M]\!]}$, the observed adjacency matrix $A = (A_{i,j})_{1 \leq i,j \leq n} \in \{0,1\}^{n \times n}$ verifies

$$A | Z = \bigotimes_{i \neq j} A_{i,j} | Z_i, Z_j = \bigotimes_{i \neq j} \mathcal{B}\left(\gamma_{Z_i, Z_j}\right),$$

where $\mathcal{B}(\gamma)$ denotes the Bernoulli distribution with parameter γ. Denote $\mathcal{SBM}_n\left(\boldsymbol{\pi}, \boldsymbol{\gamma}\right)$ the distribution of A.

2.2 Mixture of SBMs for a collection of networks

Now we consider a collection of networks partitioned into a finite number of clusters. We suppose that the networks belonging to the same cluster are independent realizations of a common SBM.

Formally, consider a collection of M networks given by the observed adjacency matrices $\mathcal{A} = \{A^{(m)}), m \in [\![M]\!]\}$, where $A^{(m)} = (A^{(m)}_{i,j})_{1 \leq i,j \leq n^{(m)}} \in \{0,1\}^{n^{(m)} \times n^{(m)}}$ denotes the adjacency matrix of the m-th network. The number of vertices $n^{(m)}$ may vary from one network to another. There is no node correspondance among the nodes of the different networks.

Let $C \geq 1$ be the number of clusters. We introduce independent discrete latent variables $\mathcal{U} = (U^{(1)}, \ldots, U^{(M)}) \in [\![C]\!]^M$ defining a partitioning of the M networks into C clusters. More precisely, we have $\mathbb{P}(U^{(m)} = c) = p_c$ where $\mathbf{p} = (p_1, \ldots, p_C) \in (0,1)^C$ with $\sum_{c \in [\![C]\!]} p_c = 1$.

Consider C different stochastic block models with model parameters denoted by $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ for $c \in [\![C]\!]$. Suppose that all networks in cluster c are independent realizations of the SBM with parameter $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$. That is, conditionally on the clustering $\mathcal{U}$,

$$\mathcal{A} | \mathcal{U} = \bigotimes_{m=1}^{M} A^{(m)} | U^{(m)} = \bigotimes_{m=1}^{M} \mathcal{SBM}_{n^{(m)}}\left(\boldsymbol{\pi}^{(U^{(m)})}, \boldsymbol{\gamma}^{(U^{(m)})}\right).$$

We adapt the notation of the block memberships of the nodes introduced in the single network SBM to the multiple network setting by adding superscript $^{(m)}$, that is, $\mathbf{Z}^{(m)} = (Z_1^{(m)}, \ldots, Z_{n^{(m)}}^{(m)})$, and we also note $\mathcal{Z} = \{\mathbf{Z}^{(m)}), m \in [\![M]\!]\}$ the collection of the block memberships of all vertices in the collection. Furthermore, let K_c be the number of blocks of the SBM with parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$. The number of blocks K_c are not constrained to

be equal. Finally, denote θ all model parameters of a mixture of SBMs encompassing the cluster proportions $\mathbf{p}$ and the SBM parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ for $c \in [\![C]\!]$, that is,

$$\theta = \left(\mathbf{p}, \{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]\}\right).$$

As in any mixture model the parameter θ is identifiable only up to label switching. That is, switching cluster labels always results in the same probability distribution of $\mathcal{A}$. In addition, also the nodes' block membership labels in each SBM are identifiable only up to label switching.

When there is only a single cluster, that is, $C = 1$ and $U^{(m)} = 1$ for $m \in [\![M]\!]$, then the observed networks $\mathcal{A}$ are independent and identically distributed with a common SBM parameter $(\boldsymbol{\pi}, \boldsymbol{\gamma})$.

3 Clustering and estimation using the ICL criterion

In a finite mixture model of SBMs, the graph clustering problem becomes the recovery of the latent clustering $\mathcal{U}$ from the data $\mathcal{A}$. Most estimation procedures that fit a SBM to a single network may directly be adapted to mixtures of SBMs. In particular, this is the case of the variational EM algorithm (Daudin et al., 2008) and MCMC methods (Peixoto, 2014). Here we consider an estimation approach that aims at estimating not only the latent clustering, but also the nodes' block memberships $\mathcal{Z}$. This is achieved by the maximization of the so-called integrated classification likelihood criterion (ICL). Roughly, the ICL is the log-likelihood function of the complete data, that is, the observations and the latent variables. Traditionally, this criterion has been used for model selection in various latent variable models, often in connection with the EM algorithm (Biernacki et al., 2000). More recently, Côme and Latouche (2015) showed that the ICL can also be used for estimating directly the latent variables of a model. Here we adapt the ICL maximization approach to mixtures of SBMs. We will see that model selection is automatically performed. More precisely, the method selects both the best number of clusters and the best number of blocks of each SBM in a data-driven way. This is an unequivocal advantage compared to the variational EM algorithm, that only works for fixed user-defined numbers of clusters and numbers of SBM blocks.

In this section, we first define the ICL criterion for a single cluster with only one SBM, then we introduce the ICL in the mixture model.

3.1 ICL criterion for a single cluster

In this subsection $\mathcal{A}$ is assumed to be a collection of i.i.d. networks of a SBM with K blocks and parameters $(\boldsymbol{\pi}, \boldsymbol{\gamma})$. Considering a Bayesian setting, we introduce a prior distribution $p(\boldsymbol{\pi}, \boldsymbol{\gamma})$ on the SBM parameters. Then, the ICL criterion is defined as

$$\mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}) = \log(p(\mathcal{A}, \mathcal{Z})) = \log\left(\int p(\mathcal{A}, \mathcal{Z}|\boldsymbol{\pi}, \boldsymbol{\gamma}) p(\boldsymbol{\pi}, \boldsymbol{\gamma}) \mathrm{d}(\boldsymbol{\pi}, \boldsymbol{\gamma})\right). \tag{1}$$

Interestingly, by integrating out the model parameters, the criterion only depends on the observations $\mathcal{A}$ and the unknown latent variables $\mathcal{Z}$. The value of $\mathcal{Z}$ optimizing the ICL, that is,

$$\hat{\mathcal{Z}} = \arg\max_{\mathcal{Z}} \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}), \tag{2}$$

corresponds to the block memberships maximizing the posterior distribution of $\mathcal{Z}$ and hence is a natural estimate of the latent variables.

For an appropriate choice of the prior, the $\mathrm{ICL}^{\mathrm{sbm}}$ has closed-form expression. We choose classical independent conjugate priors

$$\begin{aligned} p(\boldsymbol{\pi}, \boldsymbol{\gamma}) &= p(\boldsymbol{\pi}) \times \prod_{k,l \in [\![K]\!]^2} p(\gamma_{k,l}) \\ &= \mathrm{Dirichlet}(\boldsymbol{\pi}; \alpha_1, \ldots, \alpha_K) \times \prod_{k,l \in [\![K]\!]^2} \mathrm{Beta}(\gamma_{k,l}; \eta_{k,l}, \zeta_{k,l}). \end{aligned}$$

where $\alpha_1, \ldots, \alpha_K, \eta_{k,l}, \zeta_{k,l}$ are hyperparameters. In the sequel we choose identical hyperparameters for all priors, that is, $\alpha = \alpha_k$, $\eta = \eta_{k,l}$ and $\zeta = \zeta_{k,l}$ for $(k,l) \in [\![K]\!]^2$.

It is useful to introduce the following count statistics

$$s_k^{(m)} = \sum_{i \in [\![n]\!]} Z_{i,k}^{(m)}, \qquad a_{k,l}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,l}^{(m)} A_{i,j}^{(m)}, \qquad b_{k,l}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,l}^{(m)} (1 - A_{i,j}^{(m)}),$$

where for the m-th network $s_k^{(m)}$ is the number of vertices assigned to block k, $a_{k,l}^{(m)}$ the number of edges that link a vertex of block k with a vertex in block l and $b_{k,l}^{(m)}$ is the number of pairs with a vertex of block k and a vertex in block l that are not connected. Moreover, denote

$$\mathbf{s}_k = \sum_{m \in [\![M]\!]} s_k^{(m)}, \qquad \mathbf{a}_{k,l} = \sum_{m \in [\![M]\!]} a_{k,l}^{(m)}, \qquad \mathbf{b}_{k,l} = \sum_{m \in [\![M]\!]} b_{k,l}^{(m)}.$$

With these notations at hand, one can show that the ICL is given by

$$\begin{aligned} \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}) = &\sum_{(k,l) \in \in [\![K]\!]^2} \log\left(\frac{\Gamma(\eta + \mathbf{a}_{k,l})\Gamma(\zeta + \mathbf{b}_{k,l})}{\Gamma(\eta + \zeta + \mathbf{a}_{k,l} + \mathbf{b}_{k,l})}\right) + \sum_{k \in [\![K]\!]} \log\left(\Gamma(\alpha + \mathbf{s}_k)\right) \\ &+ K^2 \log\left(\frac{\Gamma(\eta + \zeta)}{\Gamma(\eta)\Gamma(\zeta)}\right) + \log\left(\frac{\Gamma(K\alpha)}{\Gamma\left(K\alpha + \sum_m n^{(m)}\right)}\right) - K \log\left(\Gamma(\alpha)\right). \end{aligned} \tag{3}$$

3.2 ICL criterion for a mixture of SBMs

In a mixture of SBMs, there are two types of latent variables, the clustering $\mathcal{U}$ of the networks and the nodes' block labels $\mathcal{Z}$. The ICL is then defined as

$$\mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) = \log(p(\mathcal{A}, \mathcal{U}, \mathcal{Z})) = \log\left(\int p(\mathcal{A}, \mathcal{U}, \mathcal{Z}|\theta) p(\theta) \mathrm{d}\theta\right), \tag{4}$$

where $p(\theta)$ is a prior on the model parameters. The values of $(\mathcal{U}, \mathcal{Z})$ optimizing the ICL, that is,

$$(\hat{\mathcal{U}}, \hat{\mathcal{Z}}) = \arg\max_{\mathcal{U},\mathcal{Z}} \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}), \tag{5}$$

correspond to the graph clustering and block memberships that maximize the posterior distribution of $(\mathcal{U}, \mathcal{Z})$.

Again we consider classical independent conjugate priors given by

$$\begin{aligned} p(\theta) &= p(\mathbf{p}) \prod_{c \in [\![C]\!]} p(\boldsymbol{\pi}^{(c)}) p(\boldsymbol{\gamma}^{(c)}) \\ &= \mathrm{Dirichlet}(\mathbf{p}; \lambda_1, \ldots, \lambda_C) \prod_{c \in [\![C]\!]} \mathrm{Dirichlet}(\boldsymbol{\pi}^{(c)}; \alpha_1, \ldots, \alpha_{K_c}) \prod_{(k,l) \in [\![K_c]\!]^2} \mathrm{Beta}(\gamma_{k,l}^{(c)}; \eta_{k,l}, \zeta_{k,l}), \end{aligned}$$

where $\lambda_c, \alpha_k, \eta_{k,l}, \zeta_{k,l}$ are appropriate hyperparameters. Again, for simplicity, we set all hyperparameters to identical values, that is, $\lambda = \lambda_c$, $\alpha = \alpha_k$, $\eta = \eta_{k,l}$ and $\zeta = \zeta_{k,l}$ for all $c \in [\![C]\!]$ and $(k,l) \in [\![K_c]\!]^2$. Moreover, let I_c be the set of indices of networks belonging to cluster c, that is, $I_c = \{m \in [\![M]\!] : U^{(m)} = c\}$ for $c \in [\![C]\!]$, and denote $\mathcal{A}^{(c)} = \{A^{(m)}, m \in I_c\}$ and $\mathcal{Z}^{(c)} = \{\mathbf{Z}^{(m)}, m \in I_c\}$.

Then the ICL can be rewritten as

$$\begin{aligned} \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) &= \log\left(\int \int p(\mathcal{A}, \mathcal{Z} | \mathcal{U}, \boldsymbol{\pi}, \boldsymbol{\gamma}) p(\boldsymbol{\pi}) p(\boldsymbol{\gamma}) \mathrm{d}(\boldsymbol{\pi}, \boldsymbol{\gamma}) \; p(\mathcal{U}|\mathbf{p}) p(\mathbf{p}) \mathrm{d}\mathbf{p} \right) \\ &= \log\left(\int \prod_{c \in [\![C]\!]} \int \prod_{m \in I_c} p(A^{(m)}, \mathbf{Z}^{(m)} | \boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}) p(\boldsymbol{\pi}^{(c)}) p(\boldsymbol{\gamma}^{(c)}) \mathrm{d}(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}) \; p(\mathcal{U}|\mathbf{p}) p(\mathbf{p}) \mathrm{d}\mathbf{p} \right) \\ &= \sum_{c \in [\![C]\!]} \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}^{(c)}, \mathcal{Z}^{(c)}) + \log\left(\int p(\mathcal{U}|\mathbf{p}) p(\mathbf{p}) \mathrm{d}\mathbf{p} \right), \end{aligned} \tag{6}$$

where $\mathrm{ICL}^{\mathrm{sbm}}$ is the ICL criterion of a single SBM for a collection of i.i.d. networks as defined in (3). The last term in (6) has closed form given by

$$\log\left(\int p(\mathcal{U}|\mathbf{p}) p(\mathbf{p}) \mathrm{d}\mathbf{p} \right) = \log\left(\frac{\Gamma(C\lambda)}{(\Gamma(\lambda))^C \Gamma(C\lambda + M)} \right) + \sum_{c \in [\![C]\!]} \log\left(\Gamma(\lambda + |I_c|) \right).$$

We remark that the ICL criterion is not exactly a similarity measure that compares clusters of networks. It is rather a model-based likelihood criterion that defines which is the best clustering. Moreover, we note that the criterion depends on the hyperparameters of the priors. Namely the hyperparameter λ influences the granularity of the best clustering $\hat{\mathcal{U}}$. Small values of λ favor a small number of clusters.

4 Hierarchical clustering algorithm

Now we develop a numerical solution to approximate the optimal clustering $\hat{\mathcal{U}}$ defined by (5). Note that this is a discrete optimization problem and thus inherently difficult to solve. We propose a greedy hill-climbing algorithm that increases the ICL criterion by convient merges of current clusters. More precisely, it is a bottom-up hierarchical algorithm that starts with numerous small clusters which are merged successively. In this section we present the global structure of the ICL maximization algorithm. The following sections 5, 6 and 7 give more details on some specific aspects of the procedure and its implementation.

The general principle of our procedure is as follows: The algorithm is initialized by a mixture of M SBMs, where every network forms a cluster on its own. So we set $U^{(m)} = m$ for $m \in [\![M]\!]$. Then, at every iteration, the potential gain in terms of the ICL obtained by merging any two clusters to a single one is computed. That is, for any pair of clusters $(c, c') \in [\![C]\!]^2$, we compute the ICL variation $\Delta_{c,c'}$ given by

$$\Delta_{c,c'} = \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}_{c\cup c'}, \mathcal{Z}_{c\cup c'}) - \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}),$$

where $\mathcal{U}$ and $\mathcal{Z}$ are the current latent variables and $\mathcal{U}_{c\cup c'}$ and $\mathcal{Z}_{c\cup c'}$ the ones obtained by merging the clusters c and c'. Finally, the cluster aggregation that yields the largest ICL increase is actually performed.

These steps of evaluating potential ICL gains for all possible merges and performing the most favorable cluster aggregation is repeated until any further cluster fusion would result in a decrease of the ICL. As such, the algorithm has a natural stopping criterion and the maximum number of iterations is $M - 1$, when all clusters are merged to a single one. However, in general the procedure stops earlier and the final number of clusters depends on the data and the hyperparameters, namely on λ that governs the prior distribution of the cluster proportions $\mathbf{p} = (p_1, \ldots, p_C)$. Thus, model selection, that is the choice of the best number of clusters, is done automatically.

Note that one could also imagine a top-down algorithm starting from a single cluster, which is then split into ever smaller clusters of networks. However, when several clusters are actually present in the data, then fitting a common SBM to the collection of networks may not make a lot of sense. The fitted SBM may not represent any of the networks adequately. In turn, when we fit a SBM to a very small cluster in a bottom-up approach, the estimation accuracy may be poor, but the model assumption should be correct.

The ICL criterion $\mathrm{ICL}^{\mathrm{mix}}$ depends on the clustering $\mathcal{U}$, but also on the latent block labels $\mathcal{Z}$ of the nodes. We choose to initialize the nodes' block memberships by adjusting a simple SBM to each network $A^{(m)}$ yielding an estimate $(\boldsymbol{\pi}^{(m)}, \boldsymbol{\gamma}^{(m)})$ of the SBM parameters as well as a block labels $\mathbf{Z}^{(m)}$. Any standard estimation algorithm for binary SBMs can be applied here. In our implementation we use the variational EM algorithm by Leger (2016).

In view of the computing time, it is important that the evaluation of ICL variations $\Delta_{c,c'}$ is fast. Section 6 provides explicit and fast formulae for the computation of $\Delta_{c,c'}$. Furthermore, a speed up can be obtained by performing not only the best but several cluster merges per iteration. This makes sense especially at the beginning of the algorithm.

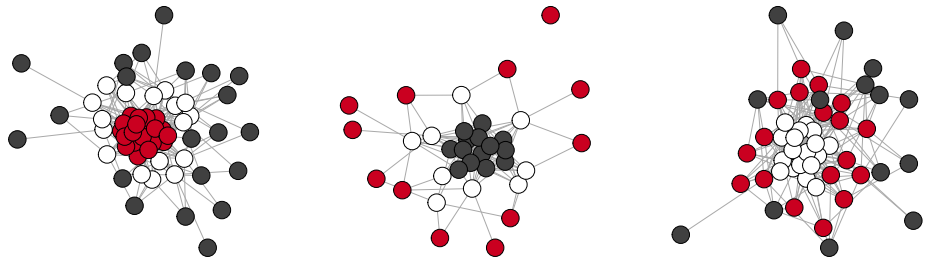


Figure 1: Illustration of the label-switching problem of block labels in the SBM. Three networks with similar topology, colors indicate the block labels $\mathbf{Z}^{(m)}$. A same block label does not correspond to the same block of a common SBM across all networks.

For large collections of networks, we recommend to perform the best 20% of all possible merges during the first 3 iterations.

When two clusters, say c and c', are merged, there are two things to do. First, the clustering is updated by setting $U^{(m)} = \min\{c, c'\}$ for all $m \in I_c \cup I_{c'}$. Second, the nodes' block labels $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$ associated with the original clusters c and c' must be updated. Here, a difficulty appears which is related to the label-switching problem in the SBM. Indeed, in a given SBM node labels can be permuted while still defining the same probability distribution. So it occurs that node labels in two SBMs do not refer to the same type of blocks. This is illustrated in Figure 1 with three networks having similar topology. More precisely, we see that they all contain a single community, some nodes in the periphery with very low degrees and some moderately connected nodes. However, for instance, the block label for the community is not the same from one network to another. Now, if the block labels of these networks should correspond to a common SBM, it is necessary to relabel the nodes such that a given block label refers to the same SBM block in all networks. In Section 5 we develop a new tool to match node labels of two SBMs using the so-called graphon function of the SBM. The tool provides permutations of the node labels of both SBMs, that are used to relabel the nodes accordingly. Finally, the relabeled block memberships of the new cluster may be improved by maximizing the ICL^{sbm} criterion of a single cluster of networks. This is done by a procedure similar to the algorithm to fit a SBM to a single network proposed by (Côme and Latouche, 2015). This algorithm is presented in Section 7.

Algorithm 1 is a summary of the entire hierarchical algorithm for finding the best clustering $\hat{\mathcal{U}}$ and the best block memberships $\hat{\mathcal{Z}}$ of the vertices. The algorithm also provides estimates of the model parameters. Moreover, we may be interested in the entire history of merges along the iterations in order to extract the cluster hierarchy that can be nicely visualized by a dendrogram.

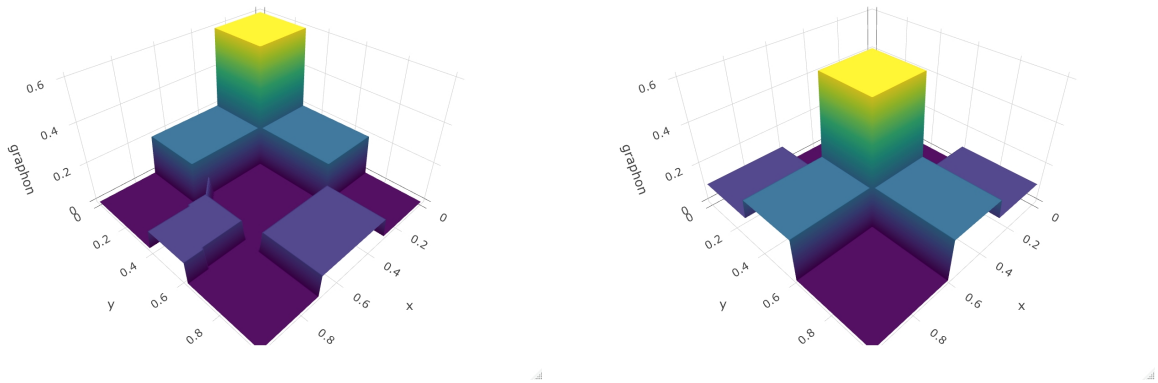
Algorithm 1: Agglomerative algorithm for graph clustering

Input: Collection of networks $\mathcal{A}$.
Set $U^{(m)} = m$ for $m \in [\![M]\!]$ and set $C = M$.
for $m \in [\![M]\!]$ **do**
  Fit a SBM to $A^{(m)}$ yielding parameters $(\boldsymbol{\pi}^{(m)}, \boldsymbol{\gamma}^{(m)})$ and block labels $\mathbf{Z}^{(m)}$.
end
Set $\mathcal{Z} = \{\mathbf{Z}^{(m)}, m \in [\![M]\!]\}$ and $\boldsymbol{\theta} = \{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]\}$.
while $C > 0$ **do**
  for $(c, c') \in [\![C]\!]^2$ **do**
    Compute $\Delta_{c,c'}$ according to Section 6.
  end
  Choose (c_1, c_2) such that $\Delta_{c_1,c_2} = \max_{c,c'} \Delta_{c,c'}$.
  if $\Delta_{c_1,c_2} > 0$ **then**
    Set $U^{(m)} = \min\{c_1, c_2\}$ for all $m \in I_c \cup I_{c'}$.
    Update $\mathcal{Z}$ and $\boldsymbol{\theta}$ according to Algorithm 3.
    Set $C = C - 1$.
  end
  else
    exit **while**
  end
end
Output: Clustering $\mathcal{U} = \{U^{(m)}, m \in [\![M]\!]\}$, block memberships $\mathcal{Z}$, SBM parameters $\{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]\}$.

5 Matching of SBM block labels

Given block memberships for two network clusters, the goal is to match the block labels of one cluster to the block labels of the other cluster. In a SBM the blocks determine the connecting behavior of the nodes, which may be very different from one block to the other. So, for a better understanding of the blocks, we may compute estimates of the associated SBM parameters and then match block labels by matching the associated parameters.

labelinof the block labels of two SBMs, we need a representation of the models that depends on the block labelings. Thus, the probability distribution of a graph for a given SBM parameter is inappropriate for this purpose, as all permutations of the block labels give rise to the same distribution. However, the SBM parameter $(\boldsymbol{\pi}, \boldsymbol{\gamma})$ is an object that depends on the block labeling. A naive way to match blocks consists in ordering both model parameters in the same way. As the SBM parameter has two parts, namely $(\boldsymbol{\pi}, \boldsymbol{\gamma})$, one may order one of its parts, for instance, either the block proportions $\pi_1, \ldots, \pi_K$ or the diagonal elements of the connectivity matrix $\boldsymbol{\gamma}$. However, this is far from optimal, since none of the parts of the parameter contains all relevant information and one can always find scenarios where such an ordering is meaningless. For example, when all diagonal elements $\gamma_{k,k}$ are equal, but not the off-diagonal connectivity parameters, ordering the diagonal elements of



(a) Original order of blocks

(b) Permuted blocks

Figure 2: Graphons of SBM parameters.

$\boldsymbol{\gamma}$ is unsatisfactory.

Indeed, to match block labels, both parts of the parameter $(\boldsymbol{\pi}, \boldsymbol{\gamma})$ must be considered jointly. A model representation relying on $\boldsymbol{\pi}$ and $\boldsymbol{\gamma}$ and on the specific block order is provided by the graphon of a SBM. We propose to define a distance measure to compare two SBM parameters based on a comparison of their graphons. Then this distance measure is used to search for the best permutations of the block labels to match the blocks.

5.1 Graphon of a SBM parameter

We propose a tool to compare two SBMs based on their graphons, a notion first introduced by Lovász and Szegedy (2006). A graphon is a function $g : [0,1]^2 \to [0,1]$ that can be used as a generative model for a huge family of exchangeable random graphs in the following way: First, for every vertex i draw a random variable $U_i \sim U[0,1]$ independently. Then, conditionally on U_i and U_j, draw an edge $A_{i,j} \sim \mathcal{B}(g(U_i, U_j))$.

The graphon of a SBM with parameters $(\boldsymbol{\pi}, \boldsymbol{\gamma})$ can be defined as

$$g(u,v) = \gamma_{k,l} \qquad \text{for every } (u,v) \in R_{k,l} = (q_{k-1}, q_k] \times (q_{l-1}, q_l]\,, \tag{7}$$

where $q_k = \sum_{s \in [\![k]\!]} \pi_s$ for $k \in [\![K]\!]$ and $q_0 = 0$. Then the latent block labels Z_i and the uniform random variables U_i are related by

$$Z_i = k \quad \Longleftrightarrow \quad U_i \in (q_{k-1}, q_k]. \tag{8}$$

The graphon is a piecewise constant function depending on both parts of the SBM parameter: the values of g are the connectivity parameters $\{\gamma_{k,l}\}$ and the rectangles $R_{k,l}$, on which the graphon is constant, are defined by the block proportions $\{\pi_k\}$. It is also clear that the graphon of a SBM depends on the order of the block labels. Changing the block

labels implies the permutation of the piecewise constant parts of the graphon. Figure 2 (a) illustrates the graphon of the SBM parameters $\boldsymbol{\pi} = (0.3, 0.4, 0.3)$ and

$$\boldsymbol{\gamma} = \begin{pmatrix} 0.6 & 0.2 & 0 \\ 0.2 & 0 & 0.1 \\ 0 & 0.1 & 0 \end{pmatrix},$$

whereas Figure 2 (b) shows the graphon of the same SBM parameter, but with permuted blocks.

We mention that (7) is not the unique way to define the graphon of a given SBM, but it is the definition used in this paper.

5.2 Label-dependent distance measure for two SBM parameters

Now for a SBM parameter $(\boldsymbol{\pi}, \boldsymbol{\gamma})$ denote $g_{(\boldsymbol{\pi},\boldsymbol{\gamma})}$ the associated graphon defined by (7). To compare two SBMs with parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$, respectively, we can consider the L^2-distance of their graphons, that is,

$$\|g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})} - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}\|_2 = \left(\int_{[0,1]^2} (g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})}(u,v) - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}(u,v))^2 \mathrm{d}(u,v) \right)^{\frac{1}{2}}. \quad (9)$$

This graphon distance has a simple explicit expression due to the piecewise constant character of the graphons of SBMs. Denoting

$$R_{k,l,k',l'} = \left\{ \left(\pi^{(c)}_{k-1}, \pi^{(c)}_{k} \right] \cap \left(\pi^{(c')}_{k'-1}, \pi^{(c')}_{k'} \right] \right\} \times \left\{ \left(\pi^{(c)}_{l-1}, \pi^{(c)}_{l} \right] \cap \left(\pi^{(c')}_{l'-1}, \pi^{(c')}_{l'} \right] \right\},$$

we see that the difference $g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})} - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}$ is piecewise constant on the non empty rectangles $R_{k,l,k',l'}$ and we have

$$g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})}(u,v) - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}(u,v) = \sum_{k,l,k',l'} \left(\gamma^{(c)}_{k,l} - \gamma^{(c')}_{k',l'} \right) \mathbb{1}\{(u,v) \in R_{k,l,k',l'}\}.$$

Hence, the squared graphon distance is a finite weighted sum of squared differences of connectivity parameters with weights depending on the block proportions, that is,

$$\|g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})} - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}\|_2^2 = \sum_{k,l,k',l'} \left(\gamma^{(c)}_{k,l} - \gamma^{(c')}_{k',l'} \right)^2 |R_{k,l,k',l'}|, \quad (10)$$

where $|R_{k,l,k',l'}|$ denotes the area of $R_{k,l,k',l'}$, which depends on the block proportions $\boldsymbol{\pi}^{(c)}$ and $\boldsymbol{\pi}^{(c')}$.

It is clear that the graphon distance $\|g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})} - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}\|_2^2$ is zero if and only if $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}) = (\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$, that is, parameter values must be identical as well as the order of the blocks. Thus, this is a label-dependent distance measure for the SBM, in opposition to measures that depend on the probability distribution of a network as the Kullback-Leibler divergence. Furthermore, the graphon distance does not require the specification

of the number of vertices of one or several networks. Hence, the computational complexity does not depend on whether we compare the clusterings of only two small networks or of two huge collections of large networks. Furthermore, this distance measure is well-defined even if the number of blocks of the two models differ.

5.3 Label-invariant distance measure for SBM

The graphon distance given by (9) can be used to define a label-invariant distance measure for SBMs. Obviously, two parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$ are identical up to label switching if and only if there exist permutations of the blocks such that the corresponding graphon distance is zero. When the parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$ define slightly different distributions, the graphon distance does not vanish for any permutation, but it still makes sense to search the permutations of the labels minimizing the graphon distance. This may be useful when parameters are estimated on two independent datasets, so that their values may be close, but not identical.

Let K_c and $K_{c'}$ be the number of blocks in $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$, resp. We define a label-invariant distance measure to compare two SBMs by the minimal graphon distance obtained with the best permutation of the blocks, that is,

$$\mathrm{d}^{\mathrm{SBM}}((\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), (\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})) = \min_{\sigma_1 \in \mathfrak{S}_{K^{(c)}}, \sigma_2 \in \mathfrak{S}_{K^{(c')}}} \| g_{\sigma_1(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})} - g_{\sigma_2(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})} \|_2, \tag{11}$$

where $\mathfrak{S}_K$ denotes the set of all permutations of $[\![K]\!]$ and

$$\sigma(\boldsymbol{\pi}, \boldsymbol{\gamma}) = \left((\pi_{\sigma(1)}, \ldots, \pi_{\sigma(K)}), (\gamma_{\sigma(k),\sigma(l)})_{k,l} \right).$$

This distance measure may be useful whenever a dissimilarity measure between two SBMs is required. Compared to divergences that depend on a probability distribution as Kullback-Leibler, $\mathrm{d}^{\mathrm{SBM}}((\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), (\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')}))$ does not depend on a specified network with a given number of vertices.

5.4 Matching blocks

Now our tool to match block labels of two SBMs consists in determining the permutations of the block labels that achieve the minimum in (11). That is, we search the permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ such that

$$(\hat{\sigma}_c, \hat{\sigma}_{c'}) \in \arg \min_{\sigma_1 \in \mathfrak{S}_{K^{(c)}}, \sigma_2 \in \mathfrak{S}_{K^{(c')}}} \| g_{\sigma_1(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})} - g_{\sigma_2(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})} \|_2. \tag{12}$$

There is no uniqueness of the solution, as for any permutation $\tau \in \mathfrak{S}_{K^{(c)}}$ the minimum is also attained with the permutations $\tau \circ \hat{\sigma}_c$ and $\tau \circ \hat{\sigma}_{c'}$, that is,

$$\| g_{\hat{\sigma}_c(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})} - g_{\hat{\sigma}_{c'}(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})} \|_2 = \| g_{\tau \circ \hat{\sigma}_c(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})} - g_{\tau \circ \hat{\sigma}_{c'}(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})} \|_2.$$

For our purpose, uniqueness of the permutations is not important, but only the identification of blocks that play the same role in the SBM.

In principle, it is possible to replace the L^2-norm in (12) by any other norm to compare the graphons. For instance, one could use the supremum norm, but as we are interested in a good overall fit of the graphons and not in the worst-case scenario, the L^2-norm is more suitable for our purpose.

For the practical computation of $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$, we first recall that the graphon distance $\|g_{\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)}} - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}\|_2$ is a finite sum and easily evaluated using equation (10). Concerning the sets of all possible permutations $\mathfrak{S}_{K^{(c)}}$ and $\mathfrak{S}_{K^{(c')}}$ that have to be explored, a exhaustive search remains feasible in short time when the number of blocks K_c and $K_{c'}$ are not too large. When the numbers of blocks are large, we may use some heuristics to reduce the set of possible permutations.

We have implemented the following procedure: first the elements of one of parameters, say $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$, are ordered such that the graphon $g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})}$ is relatively smooth. For instance, we can order the diagonal elements of the first row or column of $\boldsymbol{\gamma}^{(c)}$ in a monotone order. Then we consider permutations only of the second parameter $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$. In this way the number of permutations that are considered is reduced.

5.5 Relabeling of clusters

Now we describe in detail how to use the graphon distance to match SBM block labels of two clusters.

First, estimates of the SBM parameters for both clusters are computed. In the Bayesian context of the ICL approach, it is natural to consider the maximum a posterior estimator. The MAP estimate of the SBM parameter $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ associated with cluster c is given by

$$\begin{aligned}(\hat{\boldsymbol{\pi}}^{(c)}, \hat{\boldsymbol{\gamma}}^{(c)}) &= \arg\max_{(\boldsymbol{\pi},\boldsymbol{\gamma})} p((\boldsymbol{\pi},\boldsymbol{\gamma})|\mathcal{A}^{(c)}, \mathcal{Z}^{(c)}) \\ &= \arg\max_{(\boldsymbol{\pi},\boldsymbol{\gamma})} p(\mathcal{A}^{(c)}, \mathcal{Z}^{(c)}|(\boldsymbol{\pi},\boldsymbol{\gamma}))p(\boldsymbol{\pi},\boldsymbol{\gamma}).\end{aligned}$$

In our model, the estimator has simple closed-form expressions and is given by

$$\hat{\pi}_k^{(c)} = \frac{\sum_{m\in I_c} s_k^{(m)} + \alpha - 1}{\sum_{m\in I_c} n_k^{(m)} + K(\alpha-1)}, \qquad \hat{\gamma}_{k,\ell}^{(c)} = \frac{\sum_{m\in I_c} a_{k,\ell}^{(m)} + \eta - 1}{\sum_{m\in I_c}(a_{k,\ell}^{(m)} + b_{k,\ell}^{(m)}) + \eta + \zeta - 2}, \qquad k,\ell \in [\![K_c]\!]. \tag{13}$$

Next, the best permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ to match block labels as defined by (12) are determined. When both SBMs have the same number of blocks, that is $K_c = K_{c'}$, then the block labels $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$ are updated accordingly by reordering the labels as

$$\mathcal{Z}^{(\ell)}_{\text{update}} = (\hat{\sigma}_\ell(\mathbf{Z}^{(j)}), j \in I_\ell), \quad \text{with} \quad \hat{\sigma}_\ell(\mathbf{Z}^{(j)}) = (\mathbf{Z}^{(j)}_{\hat{\sigma}_\ell(1)}, \ldots, \mathbf{Z}^{(j)}_{\hat{\sigma}^{(\ell)}(n)}), \quad \ell \in \{c, c'\}.$$

In the case where the number of blocks K_c and $K_{c'}$ are different, we apply the following strategy. We search blocks, say $k_1, \ldots, k_L$, in the larger model that correspond to a single block, say m, in the smaller one. Then we split block m into several small blocks that match the blocks $k_1, \ldots, k_L$ in the other model.

6 Computation of $\Delta_{c,c'}$

At the beginning of every iteration, the potential ICL gain $\Delta_{c,c'}$ resulting from the fusion of clusters c and c' has to be evaluated for any pair of clusters c and c'. As the algorithm is initialized with $C = M$ clusters, it is important that the computation of $\Delta_{c,c'}$ is fast. However, the direct evaluation of $\mathrm{ICL}^{\mathrm{mix}}$ is relatively slow. In this section it is shown how to evaluate $\Delta_{c,c'}$ efficiently. In particular, for most cluster pairs $(c, c') \in [\![C]\!]^2$, $\Delta_{c,c'}$ is simply updated by adding a constant to the previous value of $\Delta_{c,c'}$.

Denote $\mathcal{U}_{c\cup c'}$ the cluster labels after fusion, that is, $U^{(m)}_{c\cup c'} = \min\{c, c'\}$ if $m \in I_c \cup I_{c'}$ and $U^{(m)}_{c\cup c'} = U^{(m)}$ otherwise. Likewise, denote $\mathcal{Z}_{c\cup c'}$ the nodes' block labels after fusion with $\mathcal{Z}^{(\ell)}_{c\cup c'} = \{\hat{\sigma}_\ell(\mathbf{Z}^{(j)}), j \in I_\ell\}$ for $\ell \in \{c, c'\}$, where $\hat{\sigma}_\ell$ are the best permutations that match the block labels defined by (12). For convenience, denote by $\mathrm{lB}(x, y) = \log\left(\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}\right)$ the logarithm of the Beta function of x and y. Moreover, for any $c \in [\![C]\!], (k, l) \in [\![K_c]\!]$, denote

$$\mathbf{s}^{(c)}_k = \sum_{m\in I_c} s^{(m)}_k, \qquad \mathbf{a}^{(c)}_{k,l} = \sum_{m\in I_c} a^{(m)}_{k,l}, \qquad \mathbf{b}^{(c)}_{k,l} = \sum_{m\in I_c} b^{(m)}_{k,l}.$$

Then we can show that $\Delta_{c,c'}$ is given by

$$\begin{aligned}
\Delta_{c,c'} &= \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}_{c\cup c'}, \mathcal{Z}_{c\cup c'}) - \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) \\
&= \sum_{(k,\ell)} \mathrm{lB}\left(\eta + \mathbf{a}^{(c)}_{\hat{\sigma}_c^{-1}(k),\hat{\sigma}_c^{-1}(l)} + \mathbf{a}^{(c')}_{\hat{\sigma}_{c'}^{-1}(k),\hat{\sigma}_{c'}^{-1}(l)} + \mathbf{b}^{(c)}_{\hat{\sigma}_c^{-1}(k),\hat{\sigma}_c^{-1}(l)} + \mathbf{b}^{(c')}_{\hat{\sigma}_{c'}^{-1}(k),\hat{\sigma}_{c'}^{-1}(l)}\right) \\
&\quad - \sum_{(k,\ell)} \mathrm{lB}\left(\eta + \mathbf{a}^{(c)}_{k,l}, \zeta + \mathbf{b}^{(c)}_{k,l}\right) - \sum_{(k,\ell)} \mathrm{lB}\left(\eta + \mathbf{a}^{(c')}_{k,l}, \zeta + \mathbf{b}^{(c')}_{k,l}\right) \qquad (14) \\
&\quad + \sum_k \log\left(\Gamma(\alpha + \mathbf{s}^{(c)}_{\hat{\sigma}_c^{-1}(k)} + \mathbf{s}^{(c')}_{\hat{\sigma}_{c'}^{-1}(k)})\right) - \log\left(\Gamma(\alpha + \mathbf{s}^{(c)}_k)\right) - \log\left(\Gamma(\alpha + \mathbf{s}^{(c')}_k)\right) \\
&\quad + \log\left(\frac{\Gamma(K_c\alpha + \sum_{m\in I_c} n^{(m)})\Gamma(K_{c'}\alpha + \sum_{m\in I_{c'}} n^{(m)})}{\Gamma\left(K_{\max}\alpha + \sum_{m\in I_c\cup I_{c'}} n^{(m)}\right)}\right) + \log\left(\frac{\Gamma(\lambda + |I_c| + |I_{c'}|)}{\Gamma(\lambda + |I_c|)\Gamma(\lambda + |I_{c'}|)}\right) \\
&\quad + K^2_{\min}\mathrm{lB}(\eta, \zeta) + K_{\min}\log\left(\Gamma(\alpha)\right) \\
&\quad + \mathrm{lB}\left((C-1)\lambda, \lambda\right) + \log\left(\frac{\Gamma(C\lambda + M)}{\Gamma((C-1)\lambda + M)}\right),
\end{aligned}$$

where $K_{\max} = \max\{K_c, K_{c'}\}$ and $K_{\min} = \min\{K_c, K_{c'}\}$ are the maximal and minimal number of blocks in the clusters c and c'.

An inspection of the above expression reveals that only the last two terms depend on the current number of clusters C. In addition, the other terms do not change from one iteration to another if both c and c' have not changed in the previous iteration, that is, if none of them is the result of the latest cluster aggregation performed by the algorithm. This makes the computation of $\Delta_{c,c'}$ easy. In this case the new value of $\Delta_{c,c'}$ is the previous

value plus some constant κ_C given by

$$\kappa_C = -\mathrm{lB}\left(C\lambda, \lambda\right) - \log\left(\frac{\Gamma((C+1)\lambda + M)}{\Gamma(C\lambda + M)}\right) + \mathrm{lB}\left((C-1)\lambda, \lambda\right) + \log\left(\frac{\Gamma(C\lambda + M)}{\Gamma((C-1)\lambda + M)}\right),$$

where we use that the number C of clusters has diminished by 1 compared to the previous iteration. Precisely, for all pairs of clusters (c, c') where both clusters have remained unchanged in the previous iteration, the update is simply

$$\Delta^{\text{new}}_{c,c'} = \Delta^{\text{old}}_{c,c'} + \kappa_C.$$

For all pairs (c, c') where one of the clusters has been obtained by the last cluster fusion, $\Delta_{c,c'}$ is computed by (14). We can avoid the computation of the statistics $s^{(m)}_k, a^{(m)}_{k,l}, b^{(m)}_{k,l}$ for all m at every iteration by stocking them during the entire algorithm and only perform local updates when necessary.

7 Update of SBM block labels in a cluster

The aggregation of two clusters involves updating the clustering $\mathcal{U}$ and relabeling the nodes' block memberships $\mathcal{Z}$ by matching them with the graphon tool. After relabeling, we propose to further improve the block labels by maximizing the ICL criterion $\mathrm{ICL}^{\mathrm{sbm}}$ defined in (2) to fit a single SBM to the new cluster. This update is the subject of this section.

Like the maximization of $\mathrm{ICL}^{\mathrm{mix}}$, the problem in (2) is a discrete optimization. For the case of a single network, that is $M = 1$, Côme and Latouche (2015) propose a greedy hill-climbing algorithm. Here we adapt their approach to multiple networks, that is $M > 1$. The idea of the procedure is to randomly choose a vertex and search the best block assignment of this vertex in terms of the ICL criterion. So one by one the block labels of the vertices are changed until no more swap further improves the ICL. In our case the algorithm may converge very rapidly, since we start from the current block labels $\mathcal{Z}$, which should be a rather could initial point. It can be expected that only very few nodes change blocks, if any.

For notational convenience, in this section we drop the superscript $^{(c)}$ of $\mathcal{A}^{(c)}$ and $\mathcal{Z}^{(c)}$ and simply write $\mathcal{A}$ and $\mathcal{Z}$. All computations only involve quantities related to the cluster under consideration. Indeed, this is an algorithm to adjust one SBM to a collection of i.i.d. networks.

Now, an iteration of the greedy algorithm that aims at maximizing $\mathrm{ICL}^{\mathrm{SBM}}$ is as follows. First, select a network indice, say $m^* \in [\![M]\!]$, and one of its vertices, say $i^* \in [\![n^{(m)}]\!]$. Denote $g = Z^{(m^*)}_{i^*}$ the current block assignment of i^*. For any block $h \in [\![K]\!]$ we compute the impact on the ICL of moving node i^* to block h. That is, we compute the difference

$$\Delta^{\to h}_{m^*,i^*} = \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}^{\to h}_{m^*,i^*}) - \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}),$$

Algorithm 2: ICL maximization algorithm for fitting one SBM to multiple networks

Input: Set of networks $\mathcal{A}$, initial block labels $\mathcal{Z}$.
while not converged **do**
  Select a network $m^* \in [\![M]\!]$ and one of its vertices $i^* \in [\![n^{(m^*)}]\!]$.
  for $h \in [\![K]\!]$ **do**
    Compute the impact $\Delta^{\to h}_{m^*,i^*}$ on the ICL of moving node i^* to block h.
  end
  Determine the best block assignment $h^* = \arg\max_{h \in [\![K]\!]} \Delta^{\to h}_{m^*,i^*}$.
  Set $Z^{(m^*)}_{i^*} = h^*$.
end
Output: Updated block labels $\mathcal{Z}$.

where $\mathcal{Z}$ denotes the current block labels of the nodes with $Z^{(m^*)}_{i^*} = g$, and $\mathcal{Z}^{\to h}_{m^*,i^*}$ the labels after moving node i^* to block h, that is, $Z^{(m^*)}_{i^*} = h$. Obviously, $\Delta^{\to g}_{m^*,i^*} = 0$, as all block labels remains unchanged. Finally, we choose the best block assignment as

$$h^* = \arg\max_{h \in [\![K]\!]} \Delta^{\to h}_{m^*,i^*},$$

and vertex i^* is definitely moved to block h^*, that is, we set $Z^{(m^*)}_{i^*} = h^*$. The algorithm is summarized in Algorithm 2.

Note that the procedure may empty a block by successively moving all nodes of a block to other blocks. Thus the selection of the best number K of SBM blocks is done automatically.

For this algorithm to be useful in practice, it is crucial that the changes $\Delta^{\to h}_{m^*,i^*}$ of the ICL can be computed very efficiently. Two cases have to be distinguished: moving node i^* to block h (i) does not empty block g; (ii) does empty block g and so the number of blocks K diminishes. But first, let us have a look on the evolution of the count statistics $s^{(m^*)}_k$, $a^{(m^*)}_{k,l}$ and $b^{(m^*)}_{k,l}$ induced by the swap.

Changes in the statistics. To study the effect of a swap on the statistics $s^{(m^*)}_k$, $a^{(m^*)}_{k,l}$ and $b^{(m^*)}_{k,l}$, denote the corresponding quantities after the swap by $\vec{s}^{(m^*)}_k$, $\vec{a}^{(m^*)}_{k,l}$ and $\vec{b}^{(m^*)}_{k,l}$. Clearly,

$$\vec{s}^{(m^*)}_g = s^{(m^*)}_g - 1, \qquad \vec{s}^{(m^*)}_h = s^{(m^*)}_h + 1,$$

while the other terms remain unchanged, that is, $\vec{s}^{(m^*)}_k = s^{(m^*)}_k$ for any $k \notin \{g, h\}$. Define

$$\delta_{k,\cdot i^*} = \sum_{i \neq i^*} Z^{(m^*)}_{i,k} A^{(m^*)}_{i,i^*}, \qquad \delta_{\ell,i^*\cdot} = \sum_{j \neq i^*} Z^{(m^*)}_{j,\ell} A^{(m^*)}_{i^*,j}.$$

Then, for any $k, \ell \in [\![K]\!]$,

$$\vec{a}_{k,\ell}^{(m^*)} = a_{k,\ell}^{(m^*)} - \mathbb{1}_{k=g}\delta_{\ell,i^*\cdot} + \mathbb{1}_{k=h}\delta_{\ell,i^*\cdot} - \mathbb{1}_{\ell=g}\delta_{k,\cdot i^*} + \mathbb{1}_{\ell=h}\delta_{k,\cdot i^*}.$$

When considering the matrix $(a_{k,\ell}^{(m^*)})_{k,\ell}$, only the g-th and h-th row and the g-th and h-th column change when moving i^* from g to h and these changes are easy to compute.

To simplify computations, we introduce the number of possible dyads from nodes in block k to nodes in block ℓ in graph m defined as

$$r_{k,\ell}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,\ell}^{(m)} = \begin{cases} s_k^{(m)} s_\ell^{(m)} & \text{if } k \neq \ell \\ s_k^{(m)}(s_k^{(m)} - 1) & \text{if } k = \ell \end{cases}$$

Then $b_{k,\ell}^{(m)} = r_{k,\ell}^{(m)} - a_{k,\ell}^{(m)}$ and

$$\vec{r}_{k,\ell}^{(m^*)} = r_{k,\ell}^{(m^*)} - s_\ell^{(m^*)}\mathbb{1}_{k=g} + s_\ell^{(m^*)}\mathbb{1}_{k=h} - s_k^{(m^*)}\mathbb{1}_{\ell=g} + s_k^{(m^*)}\mathbb{1}_{\ell=h} + 2\mathbb{1}_{k=g,\ell=g} - \mathbb{1}_{k=g,\ell=h} - \mathbb{1}_{k=h,\ell=g}.$$

and $\vec{b}_{k,l}^{(m^*)} = \vec{r}_{k,l}^{(m^*)} - \vec{a}_{k,l}^{(m^*)}$. For any $m \neq m^*$, the statistics remain unchanged, that is,$\vec{a}_{k,l}^{(m)} = a_{k,l}^{(m)}$, $\vec{b}_{k,l}^{(m)} = b_{k,l}^{(m)}$ and $\vec{r}_{k,l}^{(m)} = r_{k,l}^{(m)}$.

We introduce the following function $\Psi : \mathbb{R}_+ \times \mathbb{Z} \to \mathbb{R}$ defined by

$$\Psi(a, z) = \log\left(\frac{\Gamma(a+z)}{\Gamma(a)}\right) \qquad \text{if } a + z > 0,$$

and $\Psi(a, z) = 0$ otherwise.

First case: K does not change. If i^* is not the last vertex in block g, in other words if $\sum_m \sum_i Z_{i,g}^{(m)} > 1$, then moving node i^* to another block h does not empty block g and the number of blocks K remains unchanged. In this case, concerning the impact on the ICL, the swap only affects the first two terms on the right hand side of (3). More precisely, the change of the ICL by moving node i^* from network m^* from block g to block h is given by

$$\begin{aligned}
\Delta_{m^*,i^*}^{\to h} = & \sum_{(k,\ell)\in I_{g,h}} \left\{ \log\left(\frac{\Gamma(\eta + \sum_m \vec{a}_{k,l}^{(m)})\Gamma(\zeta + \sum_m \vec{b}_{k,l}^{(m)})}{\Gamma(\eta + \zeta + \sum_m \vec{r}_{k,l}^{(m)})}\right) \right. \\
& \qquad \left. - \log\left(\frac{\Gamma(\eta + \sum_m a_{k,l}^{(m)})\Gamma(\zeta + \sum_m b_{k,l}^{(m)})}{\Gamma(\eta + \zeta + \sum_m r_{k,l}^{(m)})}\right) \right\} \\
& + \sum_{k\in\{g,h\}} \left\{ \log\left(\Gamma(\alpha + \sum_m \vec{s}_k^{(m)})\right) - \log\left(\Gamma(\alpha + \sum_m s_k^{(m)})\right) \right\} \\
= & \sum_{(k,\ell)\in I_{g,h}} \left\{ \Psi\left(\eta + \sum_m a_{k,l}^{(m)}, \vec{a}_{k,l}^{(m^*)} - a_{k,l}^{(m^*)}\right) + \Psi\left(\zeta + \sum_m b_{k,l}^{(m)}, \vec{b}_{k,l}^{(m^*)} - b_{k,l}^{(m^*)}\right) \right. \\
& \qquad \left. - \Psi\left(\eta + \zeta + \sum_m r_{k,l}^{(m)}, \vec{r}_{k,l}^{(m^*)} - r_{k,l}^{(m^*)}\right) \right\} + \log\left(\frac{\alpha + \sum_m s_h^{(m)}}{\alpha + \sum_m s_g^{(m)} - 1}\right), \quad (15)
\end{aligned}$$

Algorithm 3: Graph cluster aggregation

Input: Two sets of networks $\mathcal{A}^{(c)}$ and $\mathcal{A}^{(c')}$ with associated clusterings $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$ and SBM parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$.

Step 1 Find the permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ defined by (12) giving the best match of blocks of $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$.

Step 2 Reorder block labels: $\mathcal{Z}^{(c)} \leftarrow \hat{\sigma}_c(\mathcal{Z}^{(c)})$ and $\mathcal{Z}^{(c')} \leftarrow \hat{\sigma}_{c'}(\mathcal{Z}^{(c')})$.

Step 3 Update the block labels $\mathcal{Z}_{c\cup c'}$ by the ICL maximization Algorithm 2.

Step 4 Compute the SBM parameter $(\boldsymbol{\pi}^{(c\cup c')}, \boldsymbol{\gamma}^{(c\cup c')})$ associated with $\mathcal{A}_{c\cup c'}$ and $\mathcal{Z}_{c\cup c'}$ according to (13).

Output: Block labels $\mathcal{Z}_{c\cup c'}$ and SBM parameter $(\boldsymbol{\pi}^{(c\cup c')}, \boldsymbol{\gamma}^{(c\cup c')})$ for the new merged cluster.

where

$$I_{g,h} = \left\{(k,\ell) \in [\![K]\!]^2, k \in \{g,h\} \text{ or } \ell \in \{g,h\}\right\}.$$

Second case: K changes. If the selected vertex i^* is the last node belonging to block g, then moving it to another block empties block g and thus diminishes the number K of blocks by one. This has an impact on all terms of the criterion $\mathrm{ICL}^{\mathrm{sbm}}$.

Before giving the formula of the impact of such a move on the ICL, we may have a closer look on the ICL criterion to better understand its dependency on the model size K. Let us compare the ICL evaluated on some data for a SBM with K blocks containing an empty block to the ICL of the same data and the same SBM but where the empty block is deleted, that is, a SBM with $K-1$ blocks. The relation is given by

$$\mathrm{ICL}^{\mathrm{sbm}}(K) = \mathrm{ICL}^{\mathrm{sbm}}(K-1) + \log\frac{\Gamma(K\alpha)}{\Gamma((K-1)\alpha)} + \log\frac{\Gamma((K-1)\alpha + \sum_m n^{(m)})}{\Gamma(K\alpha + \sum_m n^{(m)})}. \quad (16)$$

That is, the second and third term on the right-hand side are the penalty or the price to pay for using a larger model containing an empty block. Thus, by maximizing the ICL, parsimonious models are automatically favored.

In the case, where moving node i^* from network m^* from block g to block h empties a block, the change of the ICL $\Delta^{\to h}_{m^*,i^*}$ is exactly the same term as in (15) plus the penalty term given in (16). When a block is definitely emptied, then the effective dimension K of the model is diminished by 1.

7.1 Summary of merging two clusters

Now, as we have seen all details of merging two clusters in the graph clustering Algorithm 1, Algorithm 3 presents a summary of the steps to perform for an update of the nodes' block labels $\mathcal{Z}$ and the associated SBM parameters $(\boldsymbol{\pi}, \boldsymbol{\gamma})$.

8 Conclusion

In this paper an algorithm for the clustering of a collection of networks is proposed. The method aims at partitioning the data according to the general graph topology of the networks. This task was tackled by using a finite mixture of SBMs. The algorithm clusters the networks and also estimates all model parameters, which is useful for the interpretation of the results. Moreover, the hierarchical algorithm may provide the whole clustering history in form of a dendrogram, which gives further insights on the similarity among networks.

The clustering algorithm aims at maximizing the ICL, which is done via an agglomerative algorithm that automatically selects the best number of clusters. This is a clear advantage over EM-type algorithms. We provided details on the implementation of the clustering procedure. In particular, to address the label-switching problem in the SBM, we developed a tool to compare two SBMs and match their block labels.

In this paper only binary networks are considered. The adaptation of the approach to weighted graphs or to the integration of covariates is left to future work.

References

Amini, A. A., Chen, A., Bickel, P. J., and Levina, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. The Annals of Statistics, 41(4):2097–2122.

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

Botella, C., Dray, S., Matias, C., Miele, V., and Thuiller, W. (2022). An appraisal of graph embeddings for comparing trophic network architectures. Methods in Ecology and Evolution, 13(1):203–216.

Chabert-Liddell, S.-C., Barbillon, P., and Donnet, S. (2022). Learning common structures in a collection of networks. an application to food webs.

Côme, E. and Latouche, P. (2015). Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. Statistical Modelling, 15(6):564–589.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. Statistics and Computing, 18(2):173–183.

Leger, J.-B. (2016). Blockmodels: A R-package for estimating in latent block model and stochastic block model, with various probability functions, with or without covariates.

Lovász, L. and Szegedy, B. (2006). Limits of dense graph sequences. Journal of Combinatorial Theory, Series B, 96(6):933–957.

Matias, C. and Robin, S. (2014). Modeling heterogeneity in random graphs through latent space models: a selective review. Esaim Proc. & Surveys, 47:55–74.

Mehta, N., Duke, L. C., and Rai, P. (2019). Stochastic blockmodels meet graph neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 4466–4474. PMLR.

Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. Journal of the American Statistical Association, 96(455):1077–1087.

Peixoto, T. (2014). Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. Physical Review E, 89(1).