HAL
open science

Cluster or co-cluster the nodes of oriented graphs?

Christine Keribin

▶ To cite this version:

Christine Keribin. Cluster or co-cluster the nodes of oriented graphs?. 2021. hal-03139333v1

HAL Id: hal-03139333
https://inria.hal.science/hal-03139333v1

Preprint submitted on 11 Feb 2021 (v1), last revised 27 Oct 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Journal de la Société Française de Statistique
arXiv: math.PR/00000000

Cluster or co-cluster the nodes of oriented graphs?

Titre: Classification non supervisée de graphes orientés : faut-il distinguer les nœuds origines des noeuds terminaux ?

Christine Keribin[1]

Abstract: When clustering the nodes of a graph, a unique partition of the nodes is usually built, either the graph is undirected or directed. While this choice is pertinent for undirected graphs, it should be discussed for directed graphs because it implies that no difference is made between the clusters of source and target nodes. We examine this question in the context of probabilistic models with latent variables and compare the use of the Stochastic Block Model (SBM) and of the Latent Block Model (LBM). We analyze and discuss this comparison through simulated and real data sets and suggest some recommendation.

Résumé : Lors de la classification non supervisée des nœuds d'un graphe, une partition unique des nœuds est généralement construite, que le graphe soit orienté ou non. Bien que ce choix soit pertinent pour les graphes non orientés, il devrait être discuté pour les graphes orientés car il implique qu'aucune différence n'est faite entre les clusters de nœuds source et cible. Nous examinons cette question dans le contexte des modèles de clustering probabilistes à variables latentes et comparons l'utilisation des modèles de blocs stochastiques (SBM) et de blocs latents (LBM). Nous analysons et discutons cette comparaison à travers des jeux de données simulées et réelles.

Keywords: Clustering, Directed graphs, Co-clustering, Genes networks, Latent block Model, Stochastic Block Model
Mots-clés :
Classification non supervisée, Graphes orientés, Co-clustering, Réseaux de gènes, Modèle à blocs latents, Modèle à blocs stochastiques
AMS 2000 subject classifications: 62-09, 62J05, 62P10

1. Introduction

Graphs are powerful tools to model complex phenomena arising with structured network data taking into account the interactions between individuals or entities. They are used in many fields ranging from biology to social science. The availability of such large, structured network data drives the development of models and algorithms to handle them (Kolaczyk, 2009; Newman, 2010). The behaviour of nodes in a network is usually heterogeneous, and clustering a graph, i.e. partitioning its nodes into classes sharing the same connection behaviours, is of great interest to describe the graph heterogeneity, as it sums up the network through groups with different behaviours. For example in community detection, one wants to find groups of people that are highly connected between them, and differently connected to people from other groups.

A graph is undirected when the direction of the interactions between its nodes does not matter or directed in the contrary case. The clustering of directed graphs is adapted from the clustering of undirected graphs to take into account the direction of the connection and it usually provides one partition of the nodes as for the clustering of undirected graphs. However, as pointed out by

[1] Université Paris-Saclay, CNRS, Inria, Laboratoire de mathématiques d'Orsay, 91405, Orsay, France.
E-mail: christine.keribin@universite-paris-saclay.fr

Soumis au Journal de la Société Française de Statistique
File: Article_JFDS_CK.tex, compiled with jsfds, version : 2018/06/13
date: February 11, 2021

Celeux and Vasseur (2018), nodes of a directed graph intrinsically own different roles (source or emitter nodes in one hand, target or receiver nodes in another hand) and an interesting question is whether source and target nodes should be similarly clustered.

Our main goal is to address this general question in the context of blocks models, a family of probabilistic models for simultaneous clustering of rows and columns of a data matrix, where the group memberships are considered as latent variables. Let us deeper enlighten the general problematic and introduce oriented graphs and their clustering.

Graphs A graph $\mathscr{G}$ is defined by a couple of two sets $\mathscr{G} = (V,E)$, where V is a set of vertices, also called nodes, representing the individuals or entities, and E is a set of edges, pairs of vertices representing the connections between these vertices. A graph is characterized by its binary adjacency matrix A where $A_{ij} = 1$ if there is an edge between the nodes i and j, and 0 otherwise.

The edges may be directed (or oriented) or not. The edges are said to be directed if edge (i,j) and edge (j,i) (with $i,j \in V$, $i \neq j$) are different, and the considered graph is called a directed graph. If the edges are not directed, (i,j) and (j,i) represent the same edge and the graph is called undirected. For example, in case of mail exchanges, considering an oriented graph between people allows not only to model the range of the relationship, but also to take into account which is the sender and which is the receiver. Figure 1 gives an example of an undirected graph of five nodes A,B,C,D and E while Figure 2 depicts an oriented graph on these nodes.

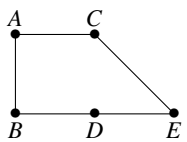


Figure 1: Undirected graph.

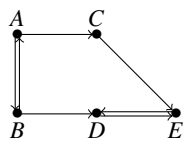


Figure 2: Directed graph.

The adjacency matrix is symmetric for an undirected graph and is in general non symmetric for a directed graph, see for example the adjacency matrix A_u (resp. A_d) for the undirected (resp. directed) graph of Figure 1 (resp. 2).

$$A_u = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} ; \quad A_d = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Graph clustering Clustering the nodes of a graph aims to find groups of nodes having a similar profile of connection, different from the connection profile of the nodes of the other groups. In case of the directed graph on Figure 2, nodes A, B and C have a similar sending behavior (similar rows), while C, D and E have a similar receiving behavior (similar columns).

Imposing a unique clustering as it is usually done can lead to Figure 3 (a) with the two groups $\{(A,B),(C,D,E)\}$ to take into the source behavior, to Figure 3 (b) with the two groups $\{(A,B),(C,D,E)\}$ to take into account the target behavior. None of them seems to be well suited. In fact, C is a special node: its behavior as a source node is close to the sending behavior of nodes D and E (similar rows C, D and E), but its behavior as a target node is close to the receiving behavior of nodes A and B (similar columns A, B and C). Considering the clustering with the three groups $\{(A,B),(C),(D,E)\}$ depicted in 3 (c) seems to better suit, by adding a specific cluster for this special node. But in this case, information is lost, namely the similar behavior of C with A and B as target node and with D and E as emitter node. Hence considering two different clusterings to qualify source and target node behaviors as in Figure 3 (d) seems to be the best way, but does not fulfill the one group constraint anymore.

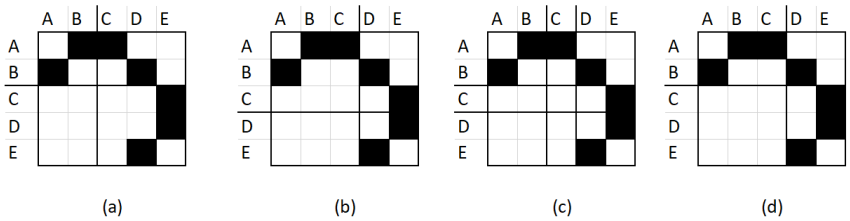
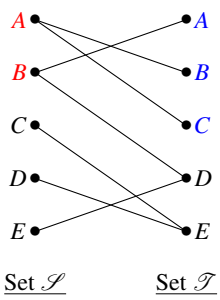


Figure 3: Different clusterings of the directed graph of Figure 2: with the same partition for source and target nodes (a) with the two groups $\{(A,B),(C,D,E)\}$, (b) with the two groups $\{(A,B,C),(D,E)\}$, (c) with the three groups $\{(A,B),(C),(D,E)\}$; (d) with different partitions $\{(A,B),(C,D,E)\}$ for source nodes and $\{(A,B,C),(D,E)\}$ for target nodes. The connection matrices are represented as 2D arrays where black squares indicate connections while white squares stand for no connection.

Bipartite graph It is interesting to note that Figure 3 (d) can be seen as a sub-graph of an augmented graph called bipartite. Bipartite graphs are undirected graphs containing two sets of nodes, say sets $\mathscr{S}$ and $\mathscr{T}$. Nodes in $\mathscr{S}$ may be linked to nodes in $\mathscr{T}$, but links between two nodes of the same set are not considered (cf. for instance Wyse et al. (2018)). There are many real life networks that can be naturally viewed in this way, often used with different sets (set of customers and products for example). A directed graph can be represented as a bipartite network where the sets $\mathscr{S}$ and $\mathscr{T}$ are the same but the role of a node is different according it is regarded as a node in $\mathscr{S}$ or a node in $\mathscr{T}$. For example, node C in $\mathscr{T}$ is linked to node A in $\mathscr{T}$ but the node C in $\mathscr{S}$ is not linked to node A in $\mathscr{T}$. Such an example of directed graph modeled as a bipartite graph is represented in Figure 4, with its (symmetric) adjacency matrix A_b.

This representation is suited to define two sets of clusters, one for source nodes, for example $\mathscr{S} = \{(A,B),(C,D,E)\}$, and one for target nodes, for example $\mathscr{T} = \{(A,B,C),(D,E)\}$, see Figure 5 (a) and amounts to *co-cluster* the two sets.

Methods There are many methods to cluster the nodes of a graph, from heuristics or insights on communities networks (such as modularity measure (Newman, 2006)) to representation learning (such as spectral analysis (Von Luxburg, 2007)) and probabilistic or generative models. We



$$A_b = \begin{array}{c} \begin{array}{cc} \mathscr{S} & \mathscr{T} \end{array} \\ \begin{pmatrix} 0 & A_d \\ {}^tA_d & 0 \end{pmatrix} \end{array} \begin{array}{l} \\ \}\mathscr{S} \\ \}\mathscr{T} \end{array}$$

Figure 4: Directed graph of Figure 2 represented as a bipartite network between nodes with source and target roles on the left, with the corresponding adjacency matrix on the right.

focus here on the latter, see Matias and Robin (Matias and Robin, 2014) for a review. A reference model is the Stochastic Block Model (SBM). This model introduced by Frank and Harary (1982) and Holland et al. (1983) has been conceived for the clustering of undirected graphs and works on the adjacency matrix. SBM has been extended to the clustering of weighted graphs and directed graphs (Mariadassou et al., 2010). This extension of SBM to directed graphs means that this model can be applied to non symmetric adjacency matrices, but it only provides a *single* clustering of the nodes and, for this very reason, could not take fully into account the different roles of the nodes in a directed graph. However, it can also be used on the corresponding bipartite graph to define two partitions, but at the cost of a huge augmentation of the data size as the number of nodes is doubled.

In the perspective to particularize the behavior of source and target nodes, Celeux and Vasseur (2018) advocates to use the Latent Block Model (LBM) for clustering a directed graph. LBM (Govaert and Nadif, 2007) is devoted to the simultaneous clustering (a.k.a. co-clustering) of the rows and columns of a rectangular matrix where rows and columns can represent different entities. In fact, it can be applied on the square adjacency matrix of a graph as a special case, hence considering source nodes (in rows) and target nodes (in columns) to be distinct. So LBM not only takes into account the non symmetry of the relationships, but also naturally provides a distinct clustering for the source and target nodes.

In other words, LBM can be regarded as a *co-clustering* model for graphs (Figure 3 (d)) while SBM offers a simple *clustering*. In fact, LBM co-clusters the top right quarter sub-graph (Figure 5 (b)) of the augmented bipartite graph (Figure 5 (a)) associated with the directed graph.

Outline This introduction has developed and deeper enlightened the general problematic after presenting the context of directed graph clustering. In particular, it shows that defining two different clusters to capture different source and target behaviors remains to cluster an augmented bipartite graph. In the context of block models, the question is: should we use a simple clustering (SBM) or a bipartite or co-clustering (LBM) for oriented graphs?

The paper aims to answer this question and is structured as follows. Section 2 defines the framework of probabilistic block models LBM and SBM, presents their estimation and discusses model selection criteria. Section 3 is devoted to the comparison of both models on simulated data and section 4 on two real data sets. A short discussion ends the article.

Remark Worth notice that the diagonal terms of the adjacency matrices of graphs are deterministic and always assumed to be zero as no self loop is admitted, and this is neglected.

2. Block models

Block models are probabilistic generative models to cluster data arrays where the membership of the entities (nodes for SBM, rows and columns for LBM) are defined as latent variables. Once the parameters have been estimated, a Maximum A Posteriory rule is used to assign each entity to its most probable cluster. The estimation of both LBM and SBM follows the same principles and encounter similar difficulties and possible drawbacks, these being more challenging for LBM with its double latent structure. We sketch here the inference of these models and then discuss model selection criteria.

2.1. The latent block model (LBM)

LBM introduced by Govaert and Nadif (2007) is a co-clustering model of a matrix A, where the rows and columns can represent two different sets of individuals or entities. For example, rows can be considered as n individuals and columns as d variables. In text mining, rows are documents and columns are words, and an observation A_{ij} is a count. In recommendation systems, rows are consumers and columns are products and an observation A_{ij} is binary or a note. In general, the two sets are different and of different sizes.

As expressed in the book Govaert and Nadif (2013), the main idea of co-clustering is to summarize the matrix A with a matrix of smaller dimension (H,L) having the same structure than A. This smaller matrix is associated to a couple of partitions (v, w), with H (resp. L) clusters on the rows (resp. columns) of A.

LBM assumes that blocks form a Cartesian product of a row-partition v in H row-clusters by a column-partition w in L column-clusters. The latent categorical variables $v = (v_1,\dots,v_n)$ and $w = (w_1,\dots w_d)$ are supposed to be independent, and each component independently assigned with mutinomial distributions

$$v_i = (v_{i1},\dots,v_{iH}) \sim \mathscr{M}(1,\rho = (\rho_h)_{h=1,\dots,H}),\ i = 1,\dots,n$$
$$w_j = (w_{j1},\dots,w_{jd}) \sim \mathscr{M}(1,\tau = (\tau_\ell)_{\ell=1,\dots,L}),\ j = 1,\dots,d$$

For example, $v_{ih} = 1$ if the row i is in the cluster h, 0 otherwise. Then, knowing the latent variables v and w, the model assumes that random variables A_{ij} are independent with conditional density ϕ whose the parameter matrix α only depends on the *block* (h,ℓ). Thus the marginal density of A is a (generalized) mixture density

$$\begin{aligned} p(A;\theta) &= \sum_{(v,w)\in\mathscr{V}\times\mathscr{W}} p(v;\theta)p(w;\theta)p(A|v,w;\theta) \\ &= \sum_{(v,w)\in\mathscr{V}\times\mathscr{W}} \prod_{i,h}\rho_h^{v_{ih}} \prod_{j,\ell}\tau_\ell^{w_{j\ell}} \prod_{h,i,j,\ell}\phi(A_{ij};\alpha_{h\ell})^{v_{ih}w_{j\ell}}, \end{aligned}$$

where $\mathscr{V}$ and $\mathscr{W}$ represent the set of possible partitions respectively for the n rows and the p columns, ρ and τ the mixing weights and $\theta = (\rho,\tau,\alpha)$ gathers the parameters.

As a special case, LBM can be used to co-cluster square binary matrices of adjacency graphs such as A_d, where the rows represent the source nodes and the columns the target nodes, i.e. $n = d$, but in this case row nodes and column nodes are considered as disconnected. Hence, two partitions are provided. Then defining $\alpha_{h\ell}$ as the probability link in the block (h,ℓ), the conditional density $\phi(A_{ij};\alpha_{h\ell})$ of random variable A_{ij} knowing $v_{ih} = 1$ and $w_{j\ell} = 1$ is assumed to be a Bernoulli distribution $\mathscr{B}(\alpha_{h\ell})$:

$$\phi(A_{ij};\alpha_{h\ell}) = \alpha_{h\ell}^{A_{ij}} \times (1-\alpha_{h\ell})^{1-A_{ij}}.$$

α can be seen as the $H \times L$ summary matrix of probability connections, see Figure 5 (b) where $H = L$. The number of parameters to estimate in this case is $(H-1)+(L-1)+HL$.

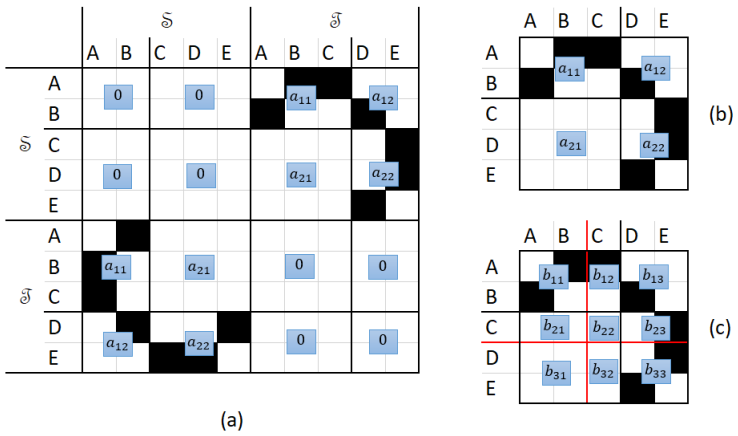


Figure 5: Different models for graph of Figure 2 with their parameter matrix in blue: (a) as a bipartite SBM graph with partition $\{\mathscr{S} = (A,B),(C,D,E)\}$ on rows (seen as source nodes) and partition $\mathscr{T} = \{(A,B,C),(D,E)\}$ on columns (seen as target nodes); (b) as a co-clustering LBM with rows as source nodes and column as target nodes; (c) with a non symmetric simple node clustering SBM with the same partition $\{(A,B),(C),(D,E)\}$ in rows and columns. Models (a) and (b) have the same likelihood.

2.2. The Stochastic Block Model (SBM)

SBM is a generative model for graphs that can be viewed as a LBM where rows ans columns are constraint to be the same entities in the same order, namely the nodes of a graph. Hence, there is only one set of latent categorical variables z defining the memberships of the nodes inside a partition of K clusters, which are independently drawn with a multinomial distribution of probability vector $\pi = (\pi_k = \mathbb{P}(z_{ik} = 1))_{k=1,\dots,K}$. Conditionally to the latent variables z, the random variables A_{ij} are independent binary variables with conditional density ϕ whose parameter matrix α only depends on the *block* (k,k'): $\alpha_{kk'} = \mathbb{P}(A_{ij} = 1 | z_{ik} = 1, z_{jk'} = 1)$. The marginal density of A is the mixture density

$$p(A;\theta) = \sum_{z\in\mathscr{Z}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{k,i,j,k'} \phi(A_{ij};\alpha_{kk'})^{z_{ik}z_{jk'}},$$

where $\mathscr{Z}$ represents the set of all possible partitions for the nodes, same for the rows and the columns, and $\theta = (\pi,\alpha)$ is the vector of parameters. Thus, the number of parameter to be estimated for SBM is $(K-1)+K^2$ for a directed graph and $(K-1)+K(K+1)/2$ for an undirected graph as the matrix α is symmetrical in this case.

2.3. SBM, bipartite SBM, LBM

Is is easy to check mathematically the equivalence between LBM defined on the adjacency matrix of a graph and a bipartite graph represented as a specific symmetric SBM, see Figures 5 (a) and (b) for an illustration, but using the latter increases the data size. Hence, a graph which is known to be bipartite must be clustered with LBM rather than with the bipartite version of SBM. We rather focus here on graphs that are not a priori bipartite, but whose source and target nodes are allowed to belong to different clusters.

We will see in section 3 that asymptotically, the best way SBM fits data generated with a model such as 5 (b) is to add a new class to take into account the specific behavior of node C, as sketched in Figure 5 (c). In fact an equivalent SBM would impose in this example the set of constraints

$$\mathscr{C} = \{b_{11} = b_{12};\ b_{21} = b_{22} = b_{31} = b_{32};\ b_{23} = b_{33}\}.$$

Figure 6 displays these constraints directly on the SBM graph and how they matches the corresponding LBM model. Unfortunately, these constraints cannot be used for the SBM estimation as the clusters likely to be grouped are unknown and this could penalize SBM in case of small size graphs.

In fact, it is difficult to compare the complexity of LBM Figure 5 (b) and SBM Figure 5 (c): (b) is more parsimonious with $L-1+H-1+HL = 6$ parameters while (c) has $L'-1+L'L' = 11$ parameters, but more complex as it needs to define two sets of latent membership variables ($2n = 10$) instead of one ($n = 10$) for SBM.

To precise a little more the way SBM follows to add new clusters when dealing with two different source and target clusterings, we introduce the source (resp. target) *profile* of a node as the row (resp.column) vector of parameter matrix α it refers to. For example, source profile of node C is $(a_{21}, a_{2,2})$ for co-clustering (b) and $(b_{21}, b_{2,2}, b_{1,3})$ for simple mode clustering (c).

As mentioned above, the way SBM deals with nodes sharing the same source (resp. target) profile with another group of nodes, but with a different target (resp. source) profile is to create a new node cluster, adding (a number of 2 times the current number k of clusters) new connection coefficients. In fact, SBM always defines a clustering where diagonal blocks of the adjacency matrix are squares, see red segments on Figure 5 (c) depicting the split. Using LBM can help to retrieve a higher level of clustering by gathering together nodes with the same source profile, independently of their target profile in one hand, and nodes with the same target profile, independently of their source profile in one hand on the other: it aims at finding which rows (resp. columns) of the parameter matrix α of SBM are identical. Several cases can be considered:

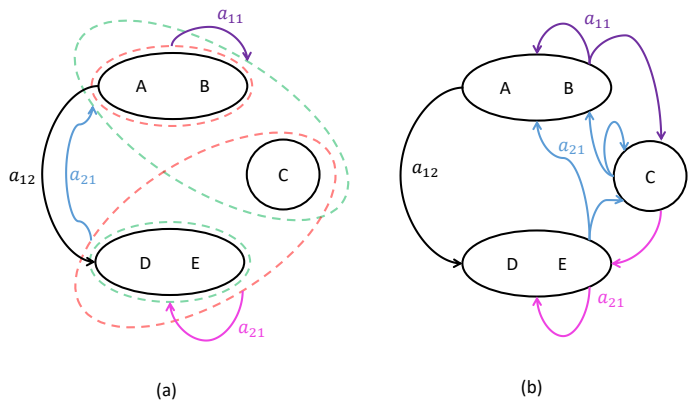


Figure 6: (a) A graph with a target node clustering (green dotted line) different from source node clustering (red dotted lines), only four connection coefficients are needed; (b) the corresponding SBM graph with a unique clustering and related contraints

— the generating model is a SBM where parameter matrix α has no identical rows neither identical columns. In this case, LBM has no interest and should lead to the same row and column clustering as SBM, $K = H = L$;
— the generating model is a SBM with a square matrix α and with constraints similar to $\mathscr{C}$: using LBM can give an insight on similar source (resp. target) profiles. The estimated LBM can be such that $H = L$ or $H \neq L$. In any cases, $K \geq max(H,L)$;
— the generating model is a LBM. In this case, as the row and column memberships u and v are not constrained to be the same, all combinations of row and column profiles are likely to appear, leading to $K = LH$ groups, number of groups of the corresponding SBM with H identical rows, and L identical columns. This induces a very special configuration of the graph. If this case is theoretically interesting, it could be however doubtful do encounter it in real data sets.

2.4. Estimation and selection in both models

The parameters to be estimated for the LBM are the proportions $\rho_h, h = 1,\ldots,H$ and $\tau_\ell, \ell = 1,\ldots,L$ and the Bernoulli block parameters $\alpha_{h\ell}, h = 1,\ldots,H; \ell = 1,\ldots,L$. The parameters to be estimated for the SBM are the proportions $\pi_k, k = 1,\ldots,K$ and the Bernoulli block parameters $\alpha_{kk'}, k = 1,\ldots,K; k' = 1,\ldots,K$.

Estimation Estimating the parameters of the LBM or the SBM is not an easy task since neither the likelihood, nor the expectation of the logarithm of the complete likelihood (observation A, latent labels Z,V,W) conditionally to the observations are available due to the complex dependency

of the observations induced by the block structure. Hence strategies must be defined to perform the E step of the Expectation-Maximization algorithm of Dempster et al. (1977). Usually, a variational approximation of the conditional distribution is proposed with the so called Variational EM (VEM) algorithm. But this algorithm is often highly dependent of its initial position.

Otherwise, maximizing the likelihood with a stochastic version of the EM algorithm is tractable, but could lead to degenerate solutions with empty clusters (see for instance Keribin et al. (2015) for a discussion about these issues). In order to circumvent these issues, the estimation problem can be considered in a non informative Bayesian framework, as advocated in Keribin et al. (2015). Once the number of classes fixed and using non informative priors, the model parameters are estimated through Gibbs sampling followed by a variational approximation of the posterior modes. Several runs based on different initializations are performed and the best result is kept.

An alternative way is to initialize with absolute eigenvalues spectral clustering. This variant of spectral clustering seems to give a very good first approximation of a clustering for the SBM case. Furthermore, it can be shown to be consistent for binary SBM without covariate (Rohe et al., 2011). It can be adapted for LBM (Leger, 2016; Frisch et al., 2020), using a double spectral clustering (using absolute eigenvalues of the Laplacian (Rohe et al., 2011)) on rows and columns on similarity matrices.

At last, it is important to note that variational and maximum likelihood estimators are consistent for SBM (Bickel et al., 2013) and LBM (Brault et al., 2020).

Selection As so far, the estimation was performed with a given number of clusters. Selection of an adequate number of clusters is of major importance and is usually done with information criteria. The use of BIC (Schwarz et al., 1978) is here questionable as an exact value of the maximum likelihood is not tractable, and conditions to perform the Laplace approximation not fulfilled. In another hand, ICL (Integrated Completed Likelihood) (Biernacki et al., 2000) is a good candidate as it involves the complete likelihood which expression is tractable in SBM or LBM models. Moreover, the binary case allows to have access to an exact expression of ICL when non informative conjugate priors are used (Keribin et al., 2015); hence there is no need of an asymptotic approximation as usual, but an asymptotic development can be done to get rid of the influence of the priors, see Keribin et al. (2015) for more details. Hence, the cluster (for SBM) or the couple of clusters (for LBM) providing the best ICL value is selected.

It remains to study the strategy to run through the models. As it is easy for SBM as the number of clusters in row and column is the same, it is trickier for LBM as two sets of clusters must be selected. To reduce the number of pairs of clusters to be considered, Robert et al. (2020) designed the bi-km1 algorithm, an upward method which allows to reduce dramatically the number of considered pairs of clusters. Initialization for a model of an upper (resp. lower) dimension can be soundly performed by splitting (resp. merging) the estimated partition of a model of lower (resp. upper) dimension.

3. Comparison on simulated data

We now compare with simulations the behaviours of the two strategies, simple node clustering with SBM and co-clustering with LBM, in case of directed graphs with the same or with different

partitions for source and target nodes. We define three connection parameter matrices $\alpha_{4\times4}$, $\tilde{\alpha}_{4\times4}$, $\alpha_{5\times4}$, where $\varepsilon = 0.1$ stands for a mixture separation index:

$$\alpha_{4\times4} = \begin{pmatrix} 1-\varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1-\varepsilon & 1-\varepsilon & \varepsilon & \varepsilon \\ 1-\varepsilon & 1-\varepsilon & 1-\varepsilon & \varepsilon \\ 1-\varepsilon & 1-\varepsilon & 1-\varepsilon & 1-\varepsilon \end{pmatrix} ; \alpha_{4\times4} = \begin{pmatrix} 1-2\varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1-2\varepsilon & 1-2\varepsilon & \varepsilon & 2\varepsilon \\ 1-3\varepsilon & 1-\varepsilon & 1-4\varepsilon & 2\varepsilon \\ 1-4\varepsilon & 1-7\varepsilon & 1-2\varepsilon & 1-\varepsilon \end{pmatrix}$$

$$\alpha_{5\times4} = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1-\varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1-\varepsilon & 1-\varepsilon & \varepsilon & \varepsilon \\ 1-\varepsilon & 1-\varepsilon & 1-\varepsilon & \varepsilon \\ 1-\varepsilon & 1-\varepsilon & 1-\varepsilon & 1-\varepsilon \end{pmatrix}.$$

and consider the following cases to illustrate the section 2.3:

Case 1: Directed graph with same partition for source and target nodes Graphs are sampled according to a non symmetric generative SBM with $K = 4$ clusters, groups proportions $\pi = (0.1, 0.2, 0.3, 0.4)$ and connection matrix $\alpha_{4\times4}$.

Figure 7 (right) gives an incisive representation of the model parameters with segment lengths proportional to the cluster weights and fill color set according to the connection probability. Here, the diagonal is formed by square blocks.

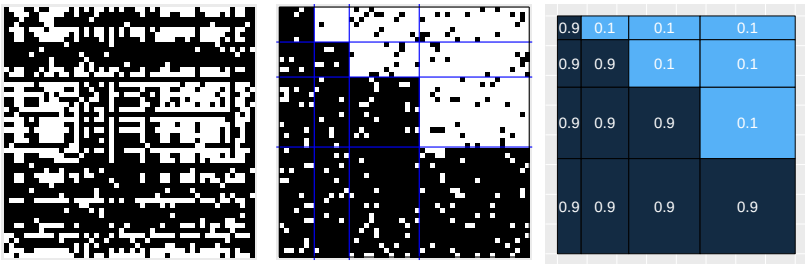


Figure 7: Case 1 example: a dataset (on the left) generated according to parameters on the right (summary representation with segment lengths proportional to cluster weights and colors to the connection probability), reordered according to the clusters (on the middle).

Case 2: Directed graph with some overlap or split between source and target partitions This can happen whether both partitions have the same number of clusters or not and we consider both:

(2a) $H = L = 4$: connection matrix $\tilde{\alpha}_{4\times4}$, row groups proportions $\pi = (0.3, 0.1, 0.3, 0.2)$, column groups proportions $\rho = (0.3, 0.2, 0.3, 0.3)$

(2b) $H = 5$, $L = 4$, connection matrix $\alpha_{5\times4}$, row groups proportions $\pi = (0.1, 0.1, 0.3, 0.3, 0.2)$, column groups proportions $\rho = (0.1, 0.2, 0.3, 0.4)$

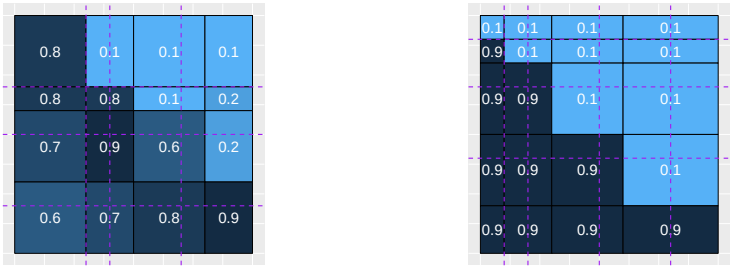


Figure 8: Model parameters: Case (2a) with $H = 4$ source clusters and $L = 4$ target clusters on the left, Case (2b) with $H = 5$ source clusters and $L = 4$ target clusters on the right. Dashed pink lines represent splits to define identical proportion vectors on rows and columns.

In these cases, graphs are generated according to the corresponding refined and constraint generative SBM: proportions π and ρ are fused into a single $\tilde{\pi}$, resulting of the refinement of both source and target node clusters (see Figure 8). Each new parameter of a block resulting of the split keeps the value of the original block parameter. It leads to $K = 6$ refined clusters for Case (2a) and $K = 7$ for Case (2b). The resulting SBM model is then used to generate the samples.

Case 3: Directed graph with complete overlap between source and target partitions They are generated with the LBM generative model and the model parameters of Case (2a) and Case (2b). Hence, all combination of rows and columns profiles are possible because groups are sampled independently on rows and columns. This leads to a corresponding SBM with $K = HL$ clusters.

Case 4: Bipartite graph The objective is to compare both methods on a bipartite graph with no a priori knowledge of this characteristic. A $n \times d$ binary matrix A_d is generated according to a LBM generative model with parameter matrix α, $\pi = \rho = (0.1, 0.2, 0.3, 0.4)$ and $d = 1.2n$. The corresponding augmented bipartite matrix A_b is then created, and rows and columns are reordered according to the same shuffling.

Results We first comment the choice of the software: among R packages, `blockcluster` (Singh Bhatia et al., 2017) is dedicated to co-clustering with a wide range of algorithms and data types, it offers ICL values but does not directly implement model selection; `bikm1` performs co-clustering for binary and Poisson latent block models and provides the bi-km1 procedure to investigate more efficiently the grid of numbers of clusters with exact ICL or approximate BIC. In another hand, `blockmodels` (Leger, 2016) provides VEM estimation with absolute eigenvalues spectral clustering initialization for binary LBM and SBM. A forward method for model selection is implemented using the asymptotic ICL criteria. All these packages are available on available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org`. We decide to use this latter as it allows to compare both models with the same software tool, hence setting them in the same experimental conditions, despite the fact that it would have been better to consider an exact ICL criterion.

TABLE 1. *The frequency of the models selected by the ICL criterion in the three cases, for different number of nodes n, and according to estimation with SBM (K) or with LBM (H\L). The generative model selection is framed.*

Case 1 (SBM with K=4)

n	K	3	4	H\L	3	4
50		10	$\boxed{40}$	3	10	
				4	19	$\boxed{21}$
100			$\boxed{50}$	4	3	$\boxed{47}$
≥ 500			$\boxed{50}$	4		$\boxed{50}$

Case 2 (overlapping source and target classes)

H=5, L=4

n	K	4	5	6	7	H\L	3	4
50		1	40	9		4	49	
						5	1	
100				1	$\boxed{49}$	5		$\boxed{50}$
≥ 500					$\boxed{50}$	5		$\boxed{50}$

H=L=4

n	K	3	4	5	6	H\ L	3	4
50		17	33			3	33	3
						4	3	$\boxed{11}$
100				19	$\boxed{31}$		3	$\boxed{47}$
≥ 500					$\boxed{50}$	4		$\boxed{50}$

Case 3 (LBM with $H = 5, L = 4$)

n	K	4	5	6	7	8	9	10	12	15	16	18	19	20	H \ L	3	4
50		1	27	22											4	27	8
															5	1	$\boxed{14}$
100					4	20	19	6	1						4		2
															5		$\boxed{48}$
500												2	41	7	5		$\boxed{50}$
1000													1	$\boxed{49}$	5		$\boxed{50}$

Case 3 (LBM with $H = L = 4$)

n	K	4	5	6	7	8	9	10	12	15	16	18	19	20	H \ L	3	4
50		12	25	13											3	26	23
															4		$\boxed{7}$
100				1	1	16	32								3	4	
															4		$\boxed{46}$
500										19	$\boxed{31}$				4		$\boxed{50}$
1000										1	$\boxed{49}$				4		$\boxed{50}$

For different sizes (number of nodes $n = 50, 100, 500, 1000$) and for each case, $B = 50$ graphs are generated and estimation and selection are performed for both models. The results are gathered in Table 1. Case 1 (directed graph with a single partition for rows and columns) is without surprise favorable to SBM as the generative model. However, when the number of nodes n increases, LBM also recovers the right block structure and it can be checked that the estimated partitions are similar to the SBM estimated single partition. Case 4 (results not shown) gives exactly the same behaviour. As soon as $n = 100$, SBM and LBM perfectly retrieves the attended number (8×8) of blocks, but the execution time is two to three times longer with LBM than with SBM.

When dealing with different partitions with some overlap for target and source nodes (Case 2), LBM performs very well as soon as $n \geq 100$, for both cases $H = L$ or $H \neq L$, while number of clusters K chosen by SBM increases with the number of nodes, even for a model generated with the same number of clusters for source and target nodes ($H = L$).

This behavior is exacerbated in Case 3, where the graph is generated according to a binary LBM: the number of clusters K chosen by SBM dramatically increases with the number of nodes, even for a model generated with the same number of clusters for source and target nodes ($H = L$). LBM on his side rapidly retrieves to generating model.

In both Case 2 and 3, we can moreover observe for sufficiently large n that SBM chooses $K = HL$, H being equal to L or not. Hence as expected, SBM clusters asymptotically, or at least with a sufficiently large number observations, the nodes according to their cross behavior between source and target profiles. Remember that constraints like $\mathscr{C}$ cannot be taken into account into the SBM estimation, as there is no way to know on which clusters to apply them. Hence, adding a new class (in row and then column) for SBM amounts to add $2k$ parameters if k is the current number of groups instead of only one if the constraints were taken into account, corresponding to the proportion of this new class. The increase of likelihood is then not sufficient with smaller n, there is not enough information and SBM loses its capacity to subdivide the clusters. Hence, it is stuck between the fact of subdividing the columns (resp. the rows) to improve the partitions while not subdividing the rows (resp. the columns) too much so as not to break homogeneous groups.

We illustrate this phenomenon on Case (2b) with $n = 500$. Model selection with ICL fully retrieves the two partitions $(H = 5, L = 4)$ for LBM co-clustering while leads to a SBM clustering with $K = 18$ clusters, see Figure 9 where the estimated groups have be reordered to let the structure appear. SBM replicates the structure of rows and column profile to manage the constraint of a single partition. Vertical pink dashed lines delimit five blocks of columns: the first two are similar with three different column profiles, the last three are also similar but with four different column profiles; horizontal pink dashed lines delimit five blocks of similar rows: the two from above gathering three clusters, the three from below gathering four clusters. Clusters 1 and 4 are not well separated (Figure 10 center and right): split of one them would also split too similar rows. Moreover, this would have required 76 additional coefficients for little improvement in the likelihood. Notice that in this example, SBM with $K = 20$ clusters splits cluster 1 in two, and another cluster rather than cluster 4 (figure not shown).

It is interesting to notice that the ICL curve for SBM against the number K of clusters (Figure 10 left) grows rapidly until $K = 12$ to reach a relative plateau in $K = 18$ before decreasing more rapidly after $K = 20$. $K = 12$ corresponds to a LBM model with $H = 4$, $L = 3$ as gathering

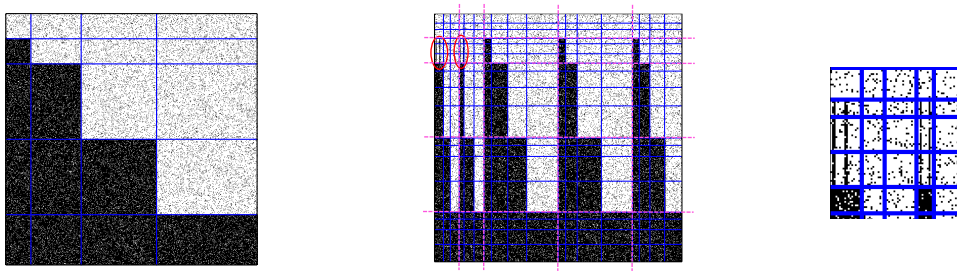


Figure 9: A graph Case (2b), $n = 500$, co-clustered by LBM with 4×5 blocks (left), clustered by SBM with 18 classes (hence 18×18 blocks) (center). Zoom to the place where SBM struggles to cluster the columns (right).

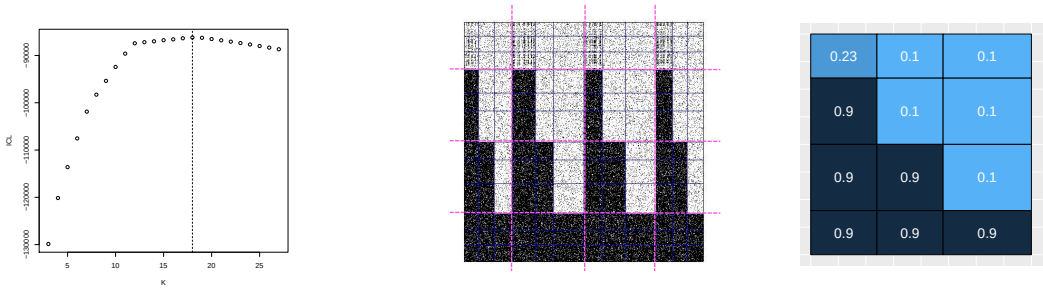


Figure 10: Clustering with SBM for Case (2b), $n = 500$. ICL against the number K of clusters form model selection (left); Clustering with $K = 12$ node clusters (middle) and corresponding LBM summary parameter matrix (right)

the two first row classes in rows and the two column classes, connection parameters being the weighted mean of the coefficients concerned by the merge. We can see here the trade-off between the refinement of the columns without splitting homogeneous rows.

4. Comparison on real datasets

We compare now the two models on two real datasets, the Arabidopsis and Enron email datasets.

4.1. Enron email dataset

We first study a version of the Enron email dataset [1], corpus of emails sent by employees of the Enron corporation. The version analyzed here is described in Priebe et al. (2019) and provided

[1] Enron email dataset available at the following URL:https://www.cs.cmu.edu/~./enron/

by Passino and Heard (2020) [2]. It consists of $n = 184$ nodes and 3 010 directed edges. A directed edge i to j is drawn if the employee i (in row) sent an email to the employee j (in column), see Figure 11.

Model selection leads to SBM clustering (ICL=-7836) with $K = 10$ node clusters and LBM co-clustering (ICL=-8118) with $H = 10$ emitter clusters and $L = 7$ emitter clusters. It is here relatively easy to map the single SBM partition and the two LBM partitions in row and columns. The resulting clustering and co-clustering are represented Figure 11, according to this correspondence. We can see that LBM column cluster 4 (resp.5) are approximately split in SBM clusters 4 and 5 (resp. 6 and 7). In that sense, LBM reveals a situation like Case 2, that can be interpreted as employees from cluster 4 and 5 receiving equivalently mails from other employees (equivalent column coefficients), but with different sending profiles: group 4 sends more emails to the other groups than group 5 (different row coefficients). Figure 12 allows to compare the estimated SBM parameters after this merge to the corresponding estimated LBM parameters.

We can check that LBM column cluster 6 also results from the merge of SBM clusters 8 and 9 although these two latter do not have the column profile. A further co-clustering with 10×10 blocks does not really change the situation. Moreover, we can also check that SBM node groups 8 and 9 and LBM row clusters 8 and 9 refer approximately to the same employees, but the sub-clustering is different depending on the method: co-clustering defines a row cluster 9 with employees never sending mails regardless what they receive while SBM prefers to distinguish two sub communities differing mainly by mail exchange inside the group. This difference is inherent to the intrinsic structure of each model.

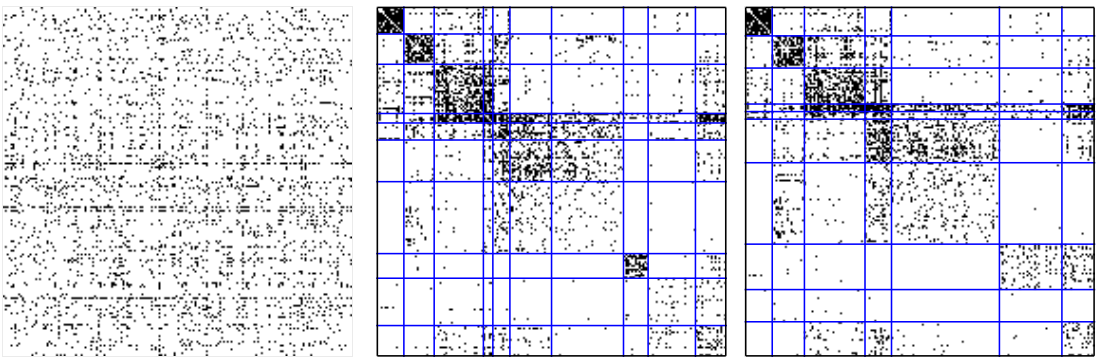


Figure 11: Enron dataset: original data (left), reorganized according to SBM with $K = 10$ node clusters (middle); co-clustering with $H = 10$ classes of emitters (rows) and 7 classes of receivers (columns).

4.2. Arabidopsis dataset

This dataset comes from a genomics network study where the biological setting is the Chlorophyllian plant *Arabidopsis thaliana (At)* (Swarbreck et al., 2007; Castrillo et al., 2011), composed

[2] https://www.github.com/fraspass/sbm

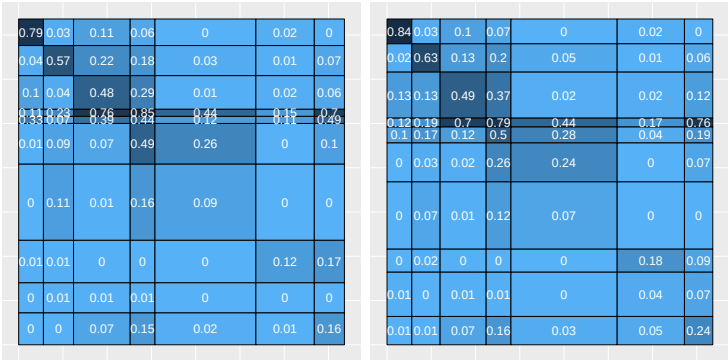


Figure 12: Enron dataset: estimated parameters for SBM after merging the following pairs of column clusters (4,5), (6,7), (8,9) on the left; for LBM on the right.

of about 25,000 genes. Using gene expression, the objective is to model the interactions between genes, namely which genes called transcription factors are to target other genes to activate or inhibit them. From DNA chips experiments, Vasseur (2017) sets up a directed graph with $n = 1937$ genes (nodes) modeling the gene network. The adjacency matrix of this graph is such that $A_{ij} = 1$ if gene i is regulated by gene j and 0 otherwise. This graph was produced using variable selection procedures based on penalized linear regressions and resampling methods.

In fact, transcription factors work in groups and the simultaneous action of several transcription factors is necessary to activate or inhibit groups of target genes (Wolpert, 1971):

— active genes having an effect on the same regulated gene are called *co-regulators*

— target genes being activate or deactivate by the same co-regulators are called *co-regulated* genes.

The objective is to cluster the regulatory network regarding these two behaviors. As a directed graph, it is interesting to study whether a co-clustering view to group simultaneously co-regulators (in rows) in one hand and co-regulated (in columns) in another hand could be helpful or better suited. Celeux and Vasseur (2018) already performed this comparison and found that the chosen number of regulator and regulated clusters are almost the same (resp. $H = 30$ co-regulator clusters and $L = 29$ co-regulated clusters) with LBM co-clustering while SBM clustering chooses $K = 22$ node clusters. This result is surprising because we saw in section 2.3 that K is in general greater than $\min(H, L)$. This could mean that the solution is not the optimal one for one or both results, regarding the known difficulties of local optima. Another explanation is that these authors used two different block model softwares, namely `bikm1` for LBM and `blockmodels` for SBM which do not have the same procedures, and especially for model selection: the former uses the exact ICL whereas the latter the asymptotic ICL. We run again the LBM co-clustering with `blockmodels` and find $H = 19$ and $L = 17$ which is now consistent with $K = 22$.

It remains to compare the two partitions. The Adjusted Rand Index (Hubert and Arabie, 1985; Rand, 1971; Youness and Saporta, 2004) is poor between the LBM row partition and the SBM node partitions (0.26), and between the LBM column partition and the SBM node partition (0.37). CARI (Robert et al., 2020), an adaptation of ARI to the block classification framework,

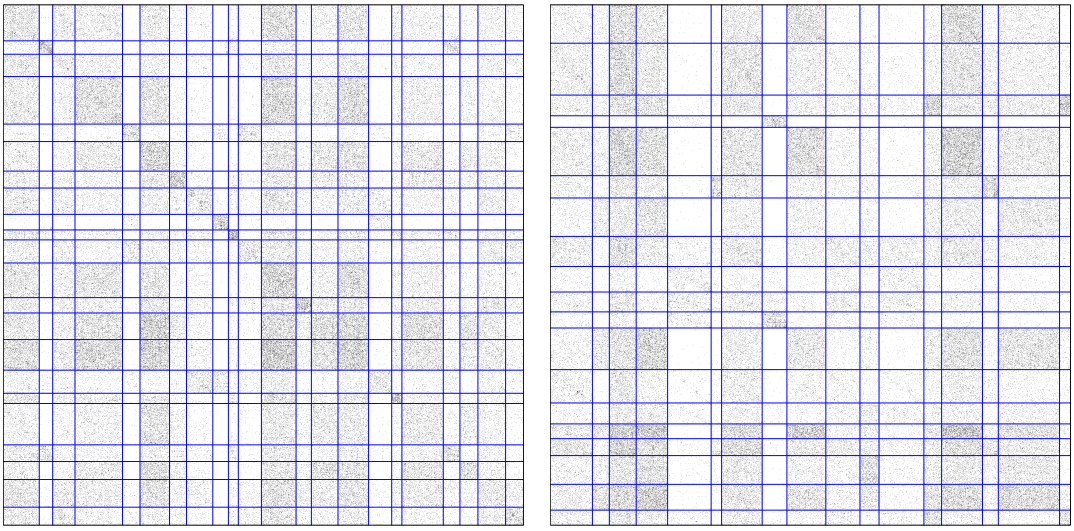


Figure 13: Reorganized adjacency matrices for Arabidopsis data: standard clustering with $K = 22$ node clusters (left); co-clustering with $H = 17$ classes of regulated genes (rows) and 19 classes of regulator genes (columns).

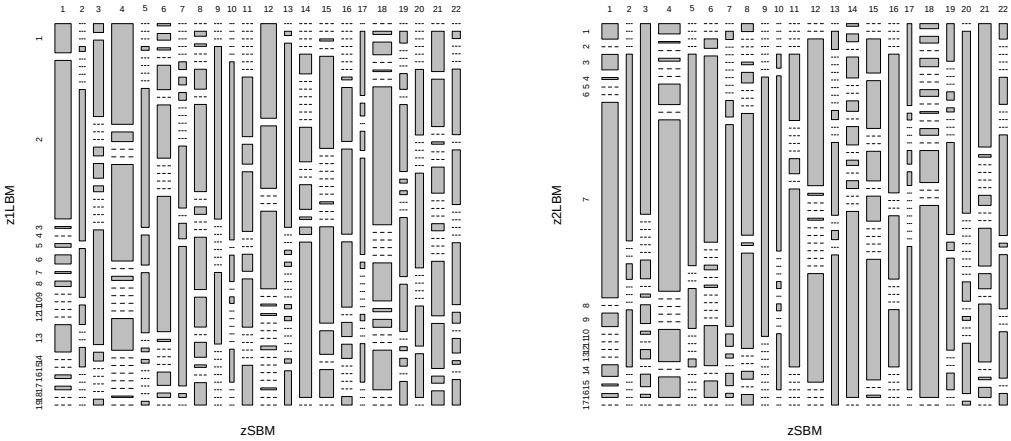


Figure 14: Proportion of row (left) and column (right) LBM clusters in each SBM class.

can be used to compare the couple of row and column partitions of LBM to the couple of same SBM partitions. Its value here is 0.12. These scores indicate poor fits between these clusterings. Hence, it is difficult to compare or map them as they are largely entangled, see Figure 14: no SBM class is mainly composed by a unique row or column LBM class as it could be attended.

In this case, it is difficult to say which representation must be preferred. First, LBM provides

a slightly smaller ICL value than SBM (SBM : -669 030, LBM : -671 234). Secondly, the LBM blocks seems less contrasted and small clusters (which are of special interest for the biologists) are rare. Moreover, the LBM representation on Figure 13 could be misleading for a user of SBM adjacency graph, as nodes are no longer represented in the same order in row and column. Hence, in this case, LBM does not seem to deeper enlighten the dataset, except to propose outsider clusters. In other words, SBM and LBM induce different links between regulated and regulator genes. In this cas, an expert view would be necessary to propose a definite choice if any between these two representations.

5. Discussion

The aim of this article is to question the use of simple clustering versus co-clustering for directed graphs, in order to fully take into account the non symmetry of such situation, namely not only the non symmetry of the connection parameter matrix, but also to the two specific roles (source, target) of a node: indeed, if node clustering such as SBM is largely used for directed graphs, could it be more relevant to perform a co-clustering LBM in some situations?

— Clustering a directed graph with a single partition could be restrictive since this model enforces source and target nodes to be in the same cluster.

— However we show, at least when the graph is sufficiently large, that if nodes have the same row (resp. column) profile, but different column (resp. row) profiles, SBM is able to handle by creating a new node cluster. But this is done at the cost of adding 2 times the current number of parameters for one merge even if many different coefficients are unnecessary. Moreover, if data is not sufficiently large with regards to the number of clusters, single SBM clustering is stuck between the fact of subdividing the columns (resp. the rows) to improve the partitions while not subdividing the rows (resp. the columns) too much so as not to break homogeneous groups. Co-clustering has here advantage as it allows naturally a more synthetic representation of this situation. It induces a specific set of connection constraints on the refined node clusters of the SBM clustering.

— In another hand, inference with LBM is more complex and numerical difficulties are more important. Hence, if no merge of row or column profiles can be defined, SBM has advantage.

— If the question is whether a graph is bipartite as a not known a priori property, SBM and LBM retrieves similarly the partitions and hence the bipartite property, but SBM outperforms LBM regarding the execution time.

— On real datasets, LBM co-clustering can help to give a higher level of representation for the SBM clustering, but the two methods can give also very different results with no clear cross interpretation, even if the number of blocks are of the same order. With its propensity to give a greater number of small clusters, SBM clustering could be preferred for some applications where small clusters are of special interest.

— Finally, to counterbalance its too much flexibility, it could be interesting to define a LBM with structural constraints between both partitions such as compatible order (overlap of adjacent classes) or split/merge between the classes.

As a general recommendation, (single) clustering still remains the first method to use with real datasets, whether the graph is oriented or not. However, performing an additional co-clustering for oriented graphs can give a higher insight on the dataset. If co-clustering gives much less blocks than the single node clustering, it brings a valuable information on the presence of (a lot of) identical row or column profiles, and reveals some specific constraints on the connection parameters between the SBM clusters. If number of blocks have around the same order, it could be more difficult to have a precise interpretation especially when the partitions poorly matches. It can be hence seen as another view of the data. If LBM provides more blocks than SBM, both results should be questioned.

A choice between the two models can be done with the ICL criterion or with a more pragmatic view based on the interpretation of the proposed blocks by the two models. A theoretical study to compare ICL behaviours in the two models is a very valuable axis of research. It is linked to the study of the asymptotic behaviour of itself, which is not as far as we now solved, see some steps on penalized likelihood criteria in Wang et al. (2017). Anyway, the comparison of the blocks exhibited by the two methods could lead to interesting remarks, even if doing such comparisons properly is not easy.

It should be noted that the co-clustering representation can be misleading for users who are used to SBM clustering, as nodes in rows do not have the same order as nodes in columns. To make easier the understanding, it could be interesting to transcript it in a SBM clustering representation with an identical node order in rows and columns, by creating as many new node clusters as needed. Moreover, one can take advantage to develop constraint versions of LBM, such as compatible order between the source and target partitions or with only split and merge operations between classes. As a more general comment, it is always interesting to compare methods and give methodological guidelines to their applications on real data. Hence, this study could be enlarged to more methods, and especially of the representation learning category. Finally, this study also shows the need to have a unified software framework on block models, providing the wide range of existing estimation methods, as well as the different model selection strategies and model selection criteria.

Acknowledgement The author wants to acknowledge Gilles Celeux for suggesting the seminal idea of this work on the comparison between LBM and SBM for directed graphs, Yann Vasseur for providing the Arabidopsis dataset and Louise Alamichel for performing some preliminary computations. The author also thanks an anonymous referee for helpful comments to enhanced the manuscript.

References

Bickel, P., Choi, D., Chang, X., Zhang, H., et al. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922–1943.

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

Brault, V., Keribin, C., and Mariadassou, M. (2020). Consistency and asymptotic normality of latent block model estimators. *Electronic journal of statistics*, 14(1):1234–1268.

Castrillo, G., Turck, F., Leveugle, M., Lecharny, A., Carbonero, P., Coupland, G., Paz-Ares, J., and Oñate-Sánchez, L. (2011). Speeding cis-trans regulation discovery by phylogenomic analyses coupled with screenings of an arrayed library of arabidopsis transcription factors. *PloS one*, 6(6):e21524.

Celeux, G. and Vasseur, Y. (2018). Classification non supervisée de graphes orientés: faut-il distinguer les nœuds origines des nœuds terminaux? In JFRB, editor, *Actes des Neuvièmes Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes 2018*, pages 68–75.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38.

Frank, O. and Harary, F. (1982). Cluster inference by using transitivity indices in empirical graphs. *Journal of the American Statistical Association*, 77(380):835–840.

Frisch, G., Léger, J.-B., and Grandvalet, Y. (2020). Learning from missing data with the latent block model. *arXiv preprint arXiv:2010.12222*.

Govaert, G. and Nadif, M. (2007). Clustering of contingency table and mixture model. *European Journal of Operational Research*, 183:1055–1066.

Govaert, G. and Nadif, M. (2013). *Co-Clustering*. ISTE Ltd and John Wiley & Sons, Inc.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic block models: First steps. *Social networks*, 5(2):109–137.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Keribin, C., Brault, V., Celeux, G., and Govaert, G. (2015). Estimation and selection for the latent block model on categorical data. *Statistics and Computing*, 25(6):1201–1216.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data Methods and Models*. Springer.

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. *arXiv preprint arXiv:1602.07587*.

Mariadassou, M., Robin, S., and Vacher, C. (2010). Uncovering latent structure in valued graphs: a variational approach. *The Annals of Applied Statistics*, 4(2):715–742.

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

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Passino, F. S. and Heard, N. A. (2020). Bayesian estimation of the latent dimension and communities in stochastic blockmodels. *Statistics and Computing*, pages 1–17.

Priebe, C. E., Park, Y., Vogelstein, J. T., Conroy, J. M., Lyzinski, V., Tang, M., Athreya, A., Cape, J., and Bridgeford, E. (2019). On a two-truths phenomenon in spectral graph clustering. *Proceedings of the National Academy of Sciences*, 116(13):5995–6000.

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

Robert, V., Vasseur, Y., and Brault, V. (2020). Comparing high-dimensional partitions with the co-clustering adjusted rand index. *Journal of Classification*, pages 1–29.

Rohe, K., Chatterjee, S., Yu, B., et al. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Singh Bhatia, P., Iovleff, S., and Govaert, G. (2017). blockcluster: An R package for model-based co-clustering. *Journal of Statistical Software*, 76(9):1–24.

Swarbreck, D., Wilks, C., Lamesch, P., Berardini, T. Z., Garcia-Hernandez, M., Foerster, H., Li, D., Meyer, T., Muller, R., Ploetz, L., et al. (2007). The arabidopsis information resource (tair): gene structure and function annotation. *Nucleic acids research*, 36(suppl_1):D1009–D1014.

Vasseur, Y. (2017). *Inférence de réseaux de régulation orientés pour les facteurs de transcription d'Arabidopsis thaliana et création de groupes de co-régulation*. PhD thesis, Paris Saclay.

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

Wang, Y. R., Bickel, P. J., et al. (2017). Likelihood-based model selection for stochastic block models. *The Annals of Statistics*, 45(2):500–528.

Wolpert, L. (1971). Positional information and pattern formation. *Current topics in developmental biology*, 6:183–224.

Wyse, J., Latouche, P., and Friel, N. (2018). Inferring structure in bipartite networks using the latent block model and exact ICL . *Network Science*, 5:45–69.

Youness, G. and Saporta, G. (2004). Une méthodologie pour la comparaison de partitions. *Revue de statistique*

appliquée, 52(1):97–120.