Rapport de stage dans l'UMR MIA Paris-Saclay

Louis Lacoste

June 27, 2023

Contents

Chapter 1

Présentation de l'UMR

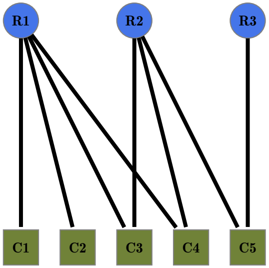
Chapter 2

Context

2.1 Usage and importance of bipartite graphs

Bipartite graphs, denoted as $G = (U, V, E)$ with U and V two disjoint and independent sets of vertices and E the set of edges connecting U vertices to V vertices.

Bipartite network



Incidence matrix

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This representation can be used to represent various forms of interactions were two kinds of “actors“ interact. Those interactions can be binary or valued and a numeric representation is the incidence matrix, in the above example B.

Among the use case of bipartite graphs one can find the Netflix Problem, which was a prize organized by Netflix to improve its Recommender system. The row nodes are the movies and the columns are the user, at the intersection the value is the review of the user j for the movie i.

Another use is the representation of ecological interactions like plant-pollinator (Ramos-Jiliberto et al. 2010), birds-seed dispersion, prey-predator or host-parasite (Kaszewska-Gilas et al. 2021). In those cases, the rows are pollinator species and the columns are plant species, and the intersection is a value, binary if it is a presence/absence or a value if it is an abundance count.

Bipartite graphs are widely used in biology, in various fields, among which the previously cited ecological networks, but also in medicine with biomedical networks, biomolecular networks or epidemiological networks. (Pavlopoulos et al. 2018)

Some interesting results can arise when applying a tool widely used on a particular kind of interactions is used on another kind of interactions. Companies like Netflix use recommender system, to recommend another product to consumers based on their previous interactions. In Desjardins-Proulx et al. 2017 the authors use the *K-nearest neighbour* (KNN) algorithm as a Recommender to predict missing preys for predators in a predator-prey network.

2.2 Latent Block Model

The Latent Block Model (LBM) introduced by Gérard Govaert and Mohamed Nadif 2010 adapts the Stochastic Block Model (SBM) (Holland et al. 1983; Snijders and Nowicki 1997) to bipartite graphs.

Please note that we prefer the term "BiSBM" and will use both LBM and BiSBM to designate the Stochastic Block model applied on bipartite networks.

This model supposes that:

- Row nodes are members of row blocks and column nodes are members of column blocks.
- The connectivity of two individuals is determined by their block memberships.
- An interaction can only occur between a row and a column node.

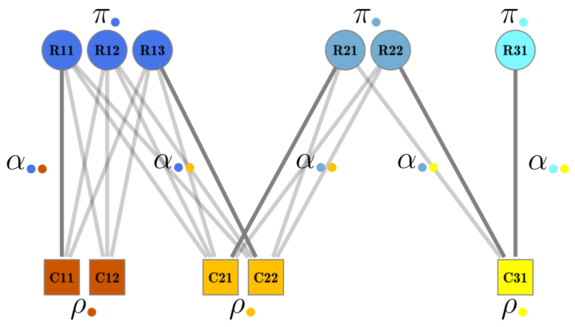

$\pi_\bullet$
R11
R12
R13
$\pi_\bullet$
R21
R22
$\pi_\bullet$
R31
$\alpha_{\bullet\bullet}$
$\alpha_{\bullet\bullet}$
$\alpha_{\bullet\bullet}$
$\alpha_{\bullet\bullet}$
$\alpha_{\bullet\bullet}$
C11
C12
C21
C22
C31
$\rho_\bullet$
$\rho_\bullet$
$\rho_\bullet$

Figure 2.1: An LBM model visualization

Parameters

- $Q_1 = \{\bullet, \bullet, \bullet\}$ blocks in rows
- $Q_2 = \{\bullet, \bullet, \bullet\}$ blocks in columns
- $\pi_\bullet = \mathbb{P}(i \in \bullet)$ in row and $\rho_\bullet = \mathbb{P}(j \in \bullet)$ in column

- $\alpha_{\bullet\bullet} = \mathbb{P}(i \leftrightarrow j | i \in \bullet, j \in \bullet)$ connectivity probability between two nodes, given their clustering

On 2.1, π are the probabilities for a row node to belong to the row block of corresponding color, ρ are the probabilities for a column node to belong to the column block of corresponding color and α are the connectivity parameters between the row and column blocks.

This model can be used to easily generate bipartite graphs with complex and very varied structures. But when trying to determine the structure of a given network we need to find those parameters.

For this a common approach is to use a VEM algorithm (proposed for SBM in Daudin et al. 2008 and for LBM in G. Govaert and M. Nadif 2005) those groups and the required parameters can be inferred by maximizing a lower bound of the likelihood minus a penalty.

2.3 colSBM model, a joint model for a collection of networks

The *colSBM* model introduced by Chabert-Liddell et al. 2023 propose an extension of the SBM model to collections of SBMs. A collection is a set of networks which nodes are not common or linked between different networks, the interactions have the same valuations and are of the same type.

The model can retrieve the shared structure in a collection, indicate if networks should be grouped in a collection and in a large pool of networks, collections with common structures.

The next step after designing this collection model for unipartite was to adapt it to the bipartite case.

Chapter 3

Structure detection in a collection of bipartite networks : Adjustment of colSBM to the bipartite case

3.1 Separate BiSBM (sepBiSBM)

A first approach to deal with a collection of networks is to adjust separate BiSBM for each network of the collection.

For network m, let n_1^m (resp. n_2^m) be the number of nodes in row (resp. column) divided into $Q_{1,m}$ row clusters (resp. $Q_{2,m}$ column clusters).
Let $Z^m = (Z_i^m, \dots, Z_{n_1^m}^m)$ and $W^m = (W_j^m, \dots, W_{n_2^m}^m)$ be independent latent variables such that $Z_i^m = q$ if row node i of network m belongs to cluster q

3.2 Definition of the model

Here are some common notations and conventions that we will use in the following sections.

3.2.1 A collection of i.i.d bipartite SBM

As for *colSBM* this first model is the most constrained. It assumes that all the networks are the independent realizations of the same Q_1-Q_2-BiSBM with identical parameters. The *iid-colBiSBM* is defined as follows:

$$X^m \sim \mathcal{F} - BiSBM_{n_1,n_2}(Q_1, Q_2, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha}), \forall m = 1, \dots M, \qquad (\textit{iid-colBiSBM})$$

3.2.2 A collection of bipartite SBM with varying block size on either rows or columns

3.3 Variational estimation of the parameters

In practice, the estimation of the likelihood is not tractable. Following the classical approach defined in Daudin et al. 2008 we use a variatonal version of the Expectation Maximization (VEM) algorithm.

We maximize a variational lower bound of the log-likelihood of the observed data by approximating $p(\boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{X}; \boldsymbol{\theta})$ with a distribution on $\boldsymbol{Z}$ and $\boldsymbol{W}$ named $\mathcal{R}$ issued from a family of factorizable distribution (Daudin et al. 2008):

$$\mathcal{J}(\mathcal{R}; \boldsymbol{\theta}) \coloneqq \mathbb{E}_{\mathcal{R}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] + \mathcal{H}(\boldsymbol{Z}, \boldsymbol{W}) \leq \ell(\boldsymbol{X}; \boldsymbol{\theta})$$

$\mathcal{H}$ is the entropy of the distribution. We define $\tau_{iq}^{1,m} = \mathbb{P}_{\mathcal{R}}(Z_{iq}^m = 1)$ and $\tau_{jr}^{2,m} = \mathbb{P}_{\mathcal{R}}(W_{jr}^m = 1)$.

The VEM algorithm alternates between two steps, the variational E step and the M step. The E steps consists in optimizing $\mathcal{J}(\mathcal{R}; \boldsymbol{\theta})$ for a current value of $\boldsymbol{\theta}$ with respect to $\mathcal{R}$. And the M step consists of maximizing $\mathcal{J}(\mathcal{R}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ and for a given variational distribution $\mathcal{R}$.

3.3.1 Variational E step

At this step we maximize with respect to $\boldsymbol{\tau}$:

$$\widehat{\boldsymbol{\tau}}^{(t+1)} = \arg\max_{\boldsymbol{\tau}} \mathcal{J}(\boldsymbol{\tau}, \widehat{\boldsymbol{\theta}}^{(t)})$$

And we obtain the following formulae for the $\boldsymbol{\tau^m}$:

$$\widehat{\tau}_{iq}^{1,m} \propto \widehat{\pi}_q^{m(t)} \prod_{j=1}^{n_2^m} \prod_{r \in \mathcal{Q}_{2,m}} f(X_{ij}^m; \widehat{\alpha}_{qr}^{(t)})^{\widehat{\tau}_{jr}^{2,m(t+1)}} \forall i = 1, \ldots, n_1^m, q \in \mathcal{Q}_{1,m}$$

$$\widehat{\tau}_{jr}^{2,m} \propto \widehat{\rho}_r^{m(t)} \prod_{i=1}^{n_1^m} \prod_{q \in \mathcal{Q}_{1,m}} f(X_{ij}^m; \widehat{\alpha}_{qr}^{(t)})^{\widehat{\tau}_{iq}^{1,m(t+1)}} \forall j = 1, \ldots, n_2^m, r \in \mathcal{Q}_{2,m}$$

From the above formulae we obtain for the Bernoulli distribution:

- *iid* :

$$\boldsymbol{\tau}^{m,1} = {}^t\pi + \exp((\text{Mask}^m \odot A^m)\boldsymbol{\tau}^{m,2}\ {}^t(\text{logit}(\alpha)) + \text{Mask}^m \boldsymbol{\tau}^{m,2}\ {}^t\log(\mathbf{1} - \alpha))$$

$$\boldsymbol{\tau}^{m,2} = {}^t\rho + \exp(\ {}^t(\text{Mask}^m \odot A^m)\boldsymbol{\tau}^{m,1}\text{logit}(\alpha) + \ {}^t\text{Mask}^m \boldsymbol{\tau}^{m,1} \log(\mathbf{1} - \alpha))$$

- $\rho\pi$:

$$\begin{aligned}\boldsymbol{\tau}^{m,1} &= {}^t\pi^m + \exp((\text{Mask}^m \odot A^m)\boldsymbol{\tau}^{m,2}\,{}^t(\text{logit}(\alpha)) + \text{Mask}^m\boldsymbol{\tau}^{m,2}\,{}^t\log(\mathbf{1}-\alpha))\\ \boldsymbol{\tau}^{m,2} &= {}^t\rho^m + \exp(\,{}^t(\text{Mask}^m \odot A^m)\boldsymbol{\tau}^{m,1}\text{logit}(\alpha) + {}^t\text{Mask}^m\boldsymbol{\tau}^{m,1}\log(\mathbf{1}-\alpha))\end{aligned}$$

with Mask^m the matrix containing 0 if the value is a NA and a 1 otherwise.

3.3.2 M step of the algorithm

At iteration (t) the M-step maximizes the variational bound with respect to the model parameters $\boldsymbol{\theta}$:

$$\widehat{\boldsymbol{\theta}}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \mathcal{J}(\widehat{\boldsymbol{\tau}}^{(t+1)}, \boldsymbol{\theta})$$

The following quantities are involved in the obtained formulae:

$$e_{qr}^m = \sum_{i=1}^{n_1^m}\sum_{j=1}^{n_2^m} \tau_{iq}^{1,m}\tau_{jr}^{2,m}X_{ij}^m, \quad n_{qr}^m = \sum_{i=1}^{n_1^m}\sum_{j=1}^{n_2^m} \tau_{iq}^{1,m}\tau_{jr}^{2,m}, \quad n_q^{1,m} = \sum_{i=1}^{n_1^m}\tau_{iq}^{1,m}, \quad n_r^{2,m} = \sum_{j=1}^{n_2^m}\tau_{jr}^{2,m}$$

The block proportions, in free mixture models, $(\pi_q^m)_{q\in\mathcal{Q}_{1,m}}, (\rho_r^m)_{r\in\mathcal{Q}_{2,m}}$ are estimated as

$$\widehat{\pi}_q^m = \frac{n_q^{1,m}}{n_1^m} \qquad \text{for } \pi\text{-}colBiSBM \text{ and } \pi\rho\text{-}colBiSBM$$

$$\widehat{\rho}_r^m = \frac{n_r^{2,m}}{n_2^m} \qquad \text{for } \rho\text{-}colBiSBM \text{ and } \pi\rho\text{-}colBiSBM$$

while on the other hand,

$$\widehat{\pi}_q = \frac{\sum_{m=1}^M n_q^{1,m}}{\sum_{m=1}^M n_1^m} \qquad \text{for } iid\text{-}colBiSBM \text{ and } \rho\text{-}colBiSBM$$

$$\widehat{\rho}_r = \frac{\sum_{m=1}^M n_r^{2,m}}{\sum_{m=1}^M n_2^m} \qquad \text{for } iid\text{-}colBiSBM \text{ and } \pi\text{-}colBiSBM$$

the parameters takes into account all the networks at the same time. The connectivity parameters α_{qr} for all models are estimated as the ratio of the number of interactions between row block q and column block r among all networks over the number of number of possible interactions:

$$\widehat{\alpha}_{qr} = \frac{\sum_{m=1}^M e_{qr}^m}{\sum_{m=1}^M n_{qr}^m}$$

3.4 Model selection

As discussed in Chabert-Liddell et al. 2023, the algorithmic aspect becomes complex when dealing with the bipartite case. Due to the size of the latent space being $\mathbb{N}^2$, conducting a complete exploration of the latent space is practically infeasible. Therefore, in addition to adapting the existing formulas, our contribution to addressing this challenge involved making significant choices, which are outlined below.

The below procedures are implemented in the *colSBM* package, available on https://github.com/Chabert-Liddell/colSBM.

3.4.1 The BIC-L criterion for model selection

The Integrated Classified Likelihood (ICL) is a well-established tool in the SBM and LBM domains for selecting the appropriate number of blocks. It was introduced by Biernacki et al. 2000; Daudin et al. 2008. The ICL is derived from an asymptotic approximation of the marginal complete likelihood. In this approach, the model parameters are integrated out using a prior distribution, resulting in a penalized likelihood criterion. By employing the ICL, one can effectively determine the optimal number of blocks for the given problem in a systematic manner. We obtain the following expression

$$\mathrm{ICL} = \max_{\boldsymbol{\theta}} \mathbb{E}_{\widehat{\mathcal{R}}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] - \frac{1}{2}\mathrm{pen}$$

with pen the penalties.
Using the formula $\mathbb{E}_{\widehat{\mathcal{R}}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] \approx \ell(\boldsymbol{X}; \boldsymbol{\theta}) - \mathcal{H}(\widehat{\mathcal{R}})$, it becomes evident, as highlighted in the existing literature, that the Integrated Classified Likelihood (ICL) gives preference to well-separated blocks by imposing a penalty on the entropy of node grouping. However, the objective of our study extends beyond grouping nodes into coherent blocks. We also aim to assess the similarity of connectivity patterns across different networks. Consequently, we aim to permit models that offer more flexible node grouping without penalizing entropy. This leads us to formulate a BIC-like criterion in the following manner:

$$\text{BIC-L} = \max_{\boldsymbol{\theta}} \mathbb{E}_{\widehat{\mathcal{R}}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] + \mathcal{H}(\widehat{\mathcal{R}}) - \frac{1}{2}\mathrm{pen} = \max_{\boldsymbol{\theta}} \mathcal{J}(\widehat{\mathcal{R}}, \boldsymbol{\theta}) - \frac{1}{2}\mathrm{pen}$$

We provide below the expression for the penalties for the 4 models that we propose.

iid-colBiSBM For the *iid-colBiSBM* the penalties were modified in the following way:

- For the πs and ρs:

$$\text{pen}_\pi(Q_1) = (Q_1 - 1)\log(\sum_{m=1}^{M} n_1^m)$$

$$\text{pen}_\rho(Q_2) = (Q_2 - 1)\log(\sum_{m=1}^{M} n_2^m)$$

- For the αs :

  $$\text{pen}_\alpha(Q_1, Q_2) = Q_1 \times Q_2 \log(N_M)$$

  with

  $$N_M = \sum_{m=1}^{M} n_1^m \times n_2^m$$

And thus the BIC-L formula is now:

$$\text{BIC-L}(\boldsymbol{X}, Q_1, Q_2) = \max_\theta \mathcal{J}(\hat{\mathcal{R}}, \boldsymbol{\theta}) - \frac{1}{2}[\text{pen}_\pi(Q_1) + \text{pen}_\rho(Q_2) + \text{pen}_\alpha(Q_1, Q_2)]$$

$\rho\pi$-colBiSBM For the *$\rho\pi$-colBiSBM* the penalties are the following:

- The support penalties are:

  $$\text{pen}_{S_1}(Q_1) = -2\log p_{Q_1}(S_1)$$

  $$\text{pen}_{S_2}(Q_2) = -2\log p_{Q_2}(S_2)$$

  with

  $$\log p_{Q_1}(S_1) = -M\log(Q_1) - \sum_{m=1}^{M} \log\binom{Q_1}{Q_1^{(m)}}$$

  $$\log p_{Q_2}(S_2) = -M\log(Q_2) - \sum_{m=1}^{M} \log\binom{Q_2}{Q_2^{(m)}}$$

- Penalties for the ρs and πs:

  $$\text{pen}_\pi(Q_1, S_1) = \sum_{m=1}^{M} (Q_1^{(m)} - 1)\log n_1^m$$

  $$\text{pen}_\rho(Q_2, S_2) = \sum_{m=1}^{M} (Q_2^{(m)} - 1)\log n_2^m$$

- Penalties for the αs:

$$\text{pen}_\alpha(Q_1, Q_2, S_1, S_2) = (\sum_{q=1}^{Q_1}\sum_{r=1}^{Q_2} \mathbb{1}_{(S_1)'S_2>0}) \log(N_M)$$

And the corresponding BIC-L formula:

$$\begin{aligned}\text{BIC-L}(\boldsymbol{X}, Q_1, Q_2) = \max_{S_1,S_2}[\max_{\theta_{S_1,S_2}\in\Theta_{S_1,S_2}} & \mathcal{J}(\hat{\mathcal{R}}, \theta_{S_1,S_2}) \\ & -\frac{1}{2}(\text{pen}_\pi(Q_1, S_1) + \text{pen}_\rho(Q_2, S_2) \\ & \quad + \text{pen}_\alpha(Q_1, Q_2, S_1, S_2) \\ & \quad + \text{pen}_{S_1}(Q_1) + \text{pen}_{S_2}(Q_2))]\end{aligned}$$

3.4.2 Initialization and pairing of the models

First to combine the information from the M networks we fit a collection model for each network at the two points $Q = (1, 2)$ and $Q = (2, 1)$. Using the previously described VEM algorithm we obtain for each network its parameters $(\boldsymbol{\rho}, \boldsymbol{\pi}, \boldsymbol{\alpha})$.

We then compute the marginal laws for each dimension, for each network. Then we order the network blocks by the probabilities obtained in decreasing order.

- For the memberships on the columns: *col order*$_m$ = *order* $(\pi_m \times \alpha_m)$
- For the memberships on the rows: *row order*$_m$ = *order* $(\rho_m \times\ {}^t(\alpha_m))$

Using this order we relabel the memberships for the M fitted collection of a single network. Then we use the M memberships to fit a collection containing the M networks.

3.4.3 Greedy exploration to find an estimation of the mode

Using the previously fitted models for $Q = (1, 2)$ and $Q = (2, 1)$ we choose to perform a greedy exploration to find a first mode.

Meaning that for a given $Q = (Q_1, Q_2)$ we will compute all the possible memberships for the points $Q \in \{(Q_1+1, Q_2), (Q_1, Q_2+1), (Q_1-1, Q_2), (Q_1, Q_2-1)\}$, fit the corresponding models and choose the one that maximizes the BIC-L as the next point from which to repeat the procedure. We repeat the procedure until the BIC-L stops increasing 2 times in a row.

```
Input  : Fitted models for $Q = (1,2)$ and $Q = (2,1)$
Output: Estimation of the mode using greedy exploration
Initialize $Q = (1,2)$ as the starting point Initialize BIC-L$_{\max}$ as the
  maximum achieved BIC-L value Initialize consecutive_count as 0
while consecutive_count $< 2$ do
    Compute possible memberships for
      $Q \in \{(Q_1+1, Q_2), (Q_1, Q_2+1), (Q_1-1, Q_2), (Q_1, Q_2-1)\}$;
    Fit models with the computed memberships Choose the model with
      the maximum BIC-L as the next point
    if BIC-L $>$ BIC-L$_{max}$ then
        BIC-L$_{\max} \leftarrow$ BIC-L consecutive_count $\leftarrow 0$
    end
    else
        consecutive_count $\leftarrow$ consecutive_count $+ 1$
    end
    $Q \leftarrow$ Next selected point
end
Output: Estimation of the mode using greedy exploration
```

Algorithm 1: Greedy Exploration for Mode Estimation

When this first estimation of the BIC-L mode has been find we apply the moving window on it.

3.4.4 Moving window to update the block memberships and the BIC-L

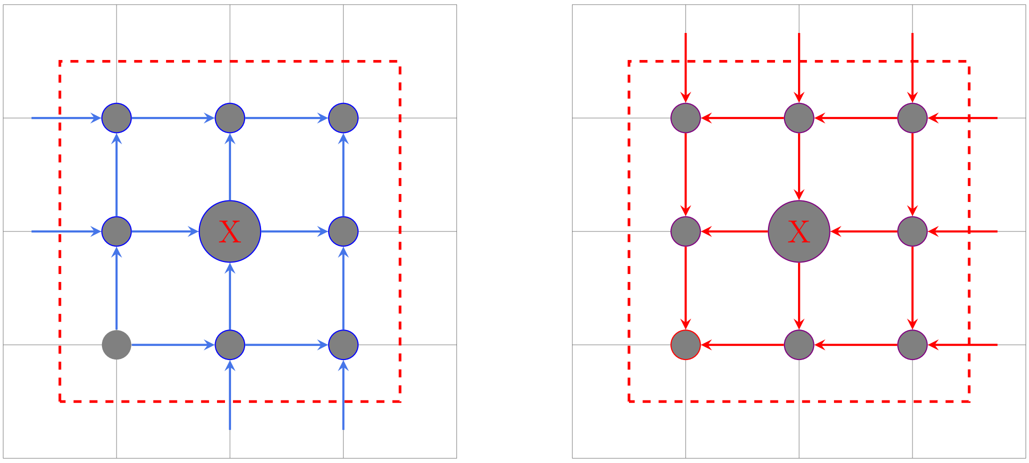
The *moving window* is used to update the block memberships on rows and columns and fit new models with those changes. To define the window, we use a center point and a *depth*, giving us the bottom left corner $(Q_{1,center} - depth, Q_{2,center} - depth)$ and the top right corner of the window $(Q_{1,center} + depth, Q_{2,center} + depth)$. All the points in this square will be updated and contribute to the update of the others. This procedure is repeated until convergence of the BIC-L.

The figure 3.1 illustrates the procedure. It consists of two alternating steps:

- the *forward pass*: repeatedly computing the possible splits to fit the current model.
- the *backward pass*: computing the possible merges to fit the current model.

```
Input  : Center point $(Q_{1,\text{center}}, Q_{2,\text{center}})$, depth
Output: Best model with maximum BIC-L in the window

Define bottom left corner $(Q_{1,\text{center}} - \text{depth}, Q_{2,\text{center}} - \text{depth})$
Define top right corner $(Q_{1,\text{center}} + \text{depth}, Q_{2,\text{center}} + \text{depth})$

while not converged do
    Forward pass:
    for $Q_1 \in [Q_{1,center} - depth; Q_{1,center} + depth]$ do
        for $Q_2 \in [Q_{2,center} - depth; Q_{2,center} + depth]$ do
            Compute possible splits from predecessors $(Q_1 - 1, Q_2)$ and
             $(Q_1, Q_2 - 1)$ Fit models with the block membership changes
             Compare and keep the best model based on BIC-L
        end
    end

    Backward pass:
    for $Q_1 \in [Q_{1,center} + depth; Q_{1,center} - depth]$ do
        for $Q_2 \in [Q_{2,center} + depth; Q_{2,center} - depth]$ do
            Compute possible merges from predecessors $(Q_1 + 1, Q_2)$ and
             $(Q_1, Q_2 + 1)$ Fit models with the block membership changes
             Compare and keep the best model based on BIC-L
        end
    end

    Update the best model based on the maximum BIC-L
end

Output: Best model with maximum BIC-L in the window
```

Algorithm 2: Moving Window Procedure



(a) Visualisation of a forward pass of moving window

(b) Visualisation of a backward pass of moving window

Figure 3.1: Moving window procedure, the center node marked with an X is the mode of BIC-L

Forward pass The forward pass consists for a model at (Q_1, Q_2) to compute the possible splits from the block memberships of its “predecessors“. The predecessors are the point at the left $(Q_1 - 1, Q_2)$ and below $(Q_1, Q_2 - 1)$ the current model (if they exist). To update the current model, we take its predecessors block memberships and try to split one of the blocks in two. Then the current model is fitted using this clustering as a starting clustering. Once all the possible splits are fitted, they are compared, keeping the best, in the sense of the BIC-L. If a model was already present it is also compared and the best is chosen as the model for this round at (Q_1, Q_2).
The procedure then repeats for the point at $(Q_1 + 1, Q_2)$ until it reaches $(Q_{1,center} + depth, Q_2)$ from which it repeats from $(Q_{1,center} - depth, Q_2 + 1)$. This repeats until computing the best model for $(Q_{1,center} + depth, Q_{2,center} + depth)$. *Note on the initialization:* The forward pass starts from the point $(Q_{1,center} + depth, Q_{2,center} + depth)$, so this points needs to have at least a model fitted. In the best case, the greedy exploration will have visited this point. But if the point has not been visited, a model will be fitted from a spectral initialization (i.e the block memberships is computed by using a spectral clustering). From this point, the next model will have at least one predecessor and the procedure can iterate.

Backward pass The backward pass consists for a model at (Q_1, Q_2) to compute the possible merges from the block memberships of its “predecessors“. The predecessors are the point at the right $(Q_1 + 1, Q_2)$ and on top $(Q_1, Q_2 + 1)$ of the current model (if the predecessors exist). To update the current model, we take its predecessors block memberships and try to merge two blocks in one. Then

the current model is fitted using this clustering as a starting clustering. Once all the possible merges are fitted, they are compared, keeping the best, in the sense of the BIC-L. If a model was already present it is also compared and the best is chosen as the model for this round at (Q_1, Q_2).
The procedure then repeats for the point at $(Q_1 - 1, Q_2)$ until it reaches $(Q_{1,center} - depth, Q_2)$ from which it repeats from $(Q_{1,center} - depth, Q_2 - 1)$. This repeats until computing the best model for $(Q_{1,center} - depth, Q_{2,center} - depth)$. *Note on the initialization:* The backward pass starts from $(Q_{1,center} + depth, Q_{2,center} + depth)$, we know it was initialized at least by the forward pass, no special case here.

At the end of the moving window pass, the model of max BIC-L is the new best fit and the procedure can repeat until convergence.

3.5 Networks clustering

As in Chabert-Liddell et al. 2023 we use a recursive algorithm to determine the best clustering of the given networks. The procedure being the same, we will present it briefly and focus on adjustments.

When networks in a collection do not share the same mesoscale connectivity structure we want to be able to partition them correctly. For this we perform a clustering of networks.

The process of clustering a collection of networks involves discovering a partition $\mathcal{G} = (\mathcal{M}_g)_{g=1,\dots,G}$ of $\{1, \dots, M\}$. Given $\mathcal{G}$ we set the following model on $\boldsymbol{X}$:

$$\forall g \in \{1, \dots, G\}, \forall m \in \mathcal{M}_g, X^m \sim \mathcal{F}\text{-}BiSBM(Q_1^g, Q_2^g, \boldsymbol{\pi^m}, \boldsymbol{\rho^m}, \boldsymbol{\alpha}^g)$$

And we defined the score of a given partition $\mathcal{G}$:

$$Sc(\mathcal{G}) = \sum_{g=1}^{G} \max_{Q^g = 1, \dots, Q_{\max}} \text{BIC-L}((X^m)_{m \in \mathcal{M}_g}, Q_1^g, Q_2^g)$$

Thus the score consists of the sum of the BIC-L of the sub-collections for the partition $\mathcal{G}$.

3.5.1 Dissimilarity between two networks

The parameters for the dissimilarity are defined as follow:

$$\widetilde{n}_{qr}^{m} = \sum_{i=1}^{n_1^m}\sum_{j=1}^{n_2^m} \widehat{\tau}_{iq}^{1,m}\widehat{\tau}_{jr}^{2,m}, \qquad \widetilde{\alpha}_{qr}^{m} = \frac{\sum_{i=1}^{n_1^m}\sum_{j=1}^{n_2^m} \widehat{\tau}_{iq}^{1,m}\widehat{\tau}_{jr}^{2,m} X_{ij}^{m}}{\widetilde{n}_{qr}^{m}},$$

$$\widetilde{\pi}_{q}^{m} = \frac{\sum_{i=1}^{n_1^m} \widehat{\tau}_{iq}^{1,m}}{n_1^m}, \qquad \widetilde{\rho}_{r}^{m} = \frac{\sum_{j=1}^{n_2^m} \widehat{\tau_{jr}}^{2,m}}{n_2^m}$$

And the dissimilarity between any pair of networks $(m, m') \in \mathcal{M}^2$ is then:

$$D_{\mathcal{M}}(m, m') = \sum_{q=1}^{Q_1}\sum_{r=1}^{Q_2} \max(\widetilde{\pi}_q^m, \widetilde{\pi}_q^{m'}) \left(\widetilde{\alpha}_{qr}^{m} - \widetilde{\alpha}_{qr}^{m'}\right)^2 \max(\widetilde{\rho}_r^m, \widetilde{\rho}_r^{m'})$$

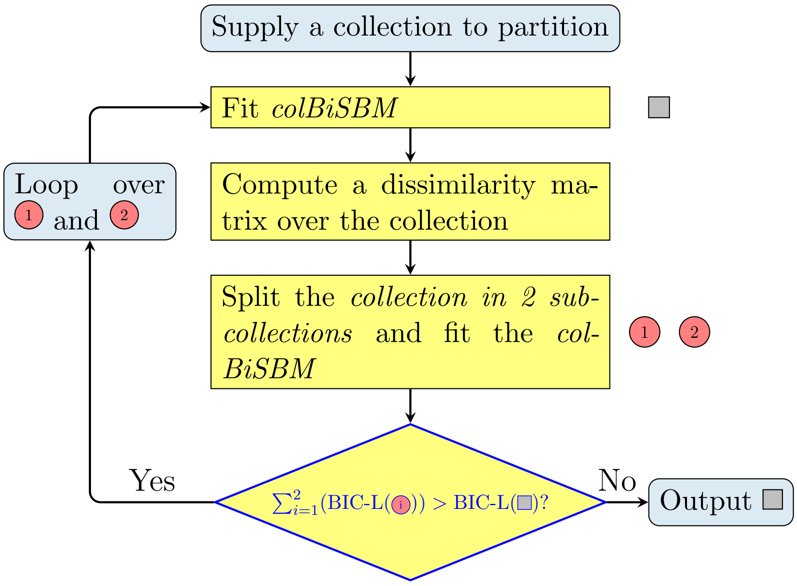

Supply a collection to partition
Fit *colBiSBM*
Compute a dissimilarity matrix over the collection
Split the *collection in 2 sub-collections* and fit the *colBiSBM*
1
2
Loop over 1 and 2
Yes
$\sum_{i=1}^{2}$(BIC-L(i)) > BIC-L(□)?
No
Output □

Figure 3.2: Network clustering procedure

The above figure (3.2) shows a condensed explanation of the network clustering algorithm.

The idea is to adjust the *colBiSBM* model over the full collection of M networks and then compute the dissimilarity matrix between all networks of the collection. We obtain the collection $\mathcal{G} = \{\mathcal{M}\}$ the trivial partition in a unique group.

Then using the *KNN* we split the collection in two sub-collections with the dissimilarity matrix. The two sub-collections are fitted and we compute the score of this new partition $\mathcal{G}^* = \{G_1, G_2\}$.

If $Sc(\mathcal{G}^*) > Sc(\mathcal{G})$ then we repeat the same procedure on G_1 and G_2. Else we return $\mathcal{G}$.

We illustrate our capacity to perform a partition of a collection for all colBiSBM models in 3.6.1.

3.6 Simulation studies

3.6.1 Network clustering of simulated networks

3.7 Application to Doré et al. 2021 data

Bibliography

Biernacki, C., Celeux, G., & 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. https://doi.org/10.1109/34.865189

Chabert-Liddell, S.-C., Barbillon, P., & Donnet, S. (2023). Learning common structures in a collection of networks. An application to food webs. https://doi.org/10.48550/arXiv.2206.00560

Daudin, J.-J., Picard, F., & Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, *18*(2), 173–183. https://doi.org/10.1007/s11222-007-9046-7

Desjardins-Proulx, P., Laigle, I., Poisot, T., & Gravel, D. (2017). Ecological interactions and the Netflix problem. *PeerJ*, *5*, e3644. https://doi.org/10.7717/peerj.3644

Doré, M., Fontaine, C., & Thébault, E. (2021). Relative effects of anthropogenic pressures, climate, and sampling design on the structure of pollination networks at the global scale. *Global Change Biology*, *27*(6), 1266–1280. https://doi.org/10.1111/gcb.15474

Govaert, G. [G.], & Nadif, M. [M.]. (2005). An EM algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 643–647. https://doi.org/10.1109/TPAMI.2005.69

Govaert, G. [Gérard], & Nadif, M. [Mohamed]. (2010). Latent Block Model for Contingency Table. *Communications in Statistics - Theory and Methods*, *39*(3), 416–425. https://doi.org/10.1080/03610920903140197

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, *5*(2), 109–137. https://doi.org/10.1016/0378-8733(83)90021-7

Kaszewska-Gilas, K., Kosicki, J. Z., Hromada, M., & Skoracki, M. (2021). Global Studies of the Host-Parasite Relationships between Ectoparasitic Mites of the Family Syringophilidae and Birds of the Order Columbiformes. *Animals*, *11*(12), 3392. https://doi.org/10.3390/ani11123392

Pavlopoulos, G. A., Kontou, P. I., Pavlopoulou, A., Bouyioukos, C., Markou, E., & Bagos, P. G. (2018). Bipartite graphs in systems biology and medicine:

A survey of methods and applications. *GigaScience*, *7*(4), giy014. https://doi.org/10.1093/gigascience/giy014

Ramos-Jiliberto, R., Domínguez, D., Espinoza, C., López, G., Valdovinos, F. S., Bustamante, R. O., & Medel, R. (2010). Topological change of Andean plant–pollinator networks along an altitudinal gradient. *Ecological Complexity*, *7*(1), 86–90. https://doi.org/10.1016/j.ecocom.2009.06.001

Snijders, T. A., & Nowicki, K. (1997). Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification*, *14*(1), 75–100. https://doi.org/10.1007/s003579900004

List of Figures

List of Tables