Rapport de stage dans l'UMR MIA Paris-Saclay

Louis Lacoste

22 mai 2023

Table des matières

Chapitre 1

Présentation de l'UMR

Chapitre 2

Adjustment of colSBM to the bipartite case : colBiSBM

2.1 Variational Expectation step

Fixed point formula for the Bernoulli distribution :

— *iid* :

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

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

— $\rho\pi$:

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

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

avec Mask^m la matrice qui contient des 0 si la valeur est un NA et des 1 sinon.

2.2 M step of the algorithm

2.3 Computation of the variational bound

2.4 Penalties

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

— For the πs and ρs :

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

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

— For the αs :

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

avec

$$N_M = \sum_{m=1}^{M} n_r^{(m)} \times n_c^{(m)}$$

And thus the $BIC - L$ formula is now :

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

ρπ-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 :

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

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

— 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 :

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

2.5 Latent space exploration and model selection

In order to explorer the bi-dimensional latent space (Q_1, Q_2) we use the following strategies.

2.5.1 Model selection

In the following steps the model selection consists of using the $BIC - L$ criterion to select the model. We choose among the proposed models the one that maximizes the $BIC - L$

2.5.2 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 (ρ, π, α).

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.

2.5.3 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 $Q = (Q_1 + 1, Q_2)$ and $Q = (Q_1, Q_2 + 1)$, 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 3 times in a row.

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

2.5.4 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 cen-

ter 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 procedure 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.

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.

2.6 Networks clustering

As in [1] we use a recursive algorithm to determine the best clustering of the given networks. The procedure being the same, only the technical modifications for the bipartite case will be explained below.

2.6.1 Distance between two networks

The distance weights uses π and ρ.

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

Bibliographie

[1] Saint-Clair Chabert-Liddell, Pierre Barbillon, and Sophie Donnet. Learning common structures in a collection of networks. An application to food webs, March 2023.

Table des figures

Liste des tableaux