Extensions of the EM algorithm for fitting the Stochastic Block Models

julien.chiquet@gmail.com

M2 ISG - summer 2016 -
network modelling - practical 3

Instructions. Each student must send a small report generated with `R` markdown answering the questions and including comments associated with the `R` code.

1 Context

Recall the notation for the Stochastic Block Model. The p nodes are spread among a set of Q classes $\mathcal{Q} = \{1, \ldots, Q\}$ with *a priori* distribution $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_Q)$. The hidden random indicator variables $(Z_{iq})_{i \in \mathcal{P}, q \in \mathcal{Q}}$ define the classes each node belongs to such that

$$\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\alpha}). \tag{1}$$

The probability for having an edge between nodes i and j is defined *conditionally on* the classes they belong to :

$$\Theta_{ij} | \{i \in q, j \in \ell\} \sim \mathcal{B}(\pi_{q\ell}), \quad i \neq j. \tag{2}$$

To sum up, the parameters are

— $\boldsymbol{\Theta} = (\Theta_{ij})$, the $p \times p$ adjacency matrix of the graph,
— $\boldsymbol{\pi} = (\pi_{q\ell})$ the $Q \times Q$ connectivity matrix,
— $\boldsymbol{\alpha} = (\alpha_q)$, the size-$Q$ vector of class proportions.

2 Questions

1. *Complete Likelihood.* Derive the model complete-data loglikelihood. Write the corresponding conditional expectation as a function of $\tau_{iq} = \mathbb{P}(Z_{iq} = 1)$ and $\nu_{ijq\ell} = \mathbb{P}(Z_{iq} Z_{j\ell} = 1)$.
2. *M-step.* For fixed values of $\hat{\tau}_{iq}$ and , $\hat{\nu}_{ijq\ell}$, derive estimators for α_q and $\pi_{q\ell}$ by maximizing the conditional expectation of the EM algorithm.
3. *Approximated E-step.* The E-step cannot be implemented straightforwardly since the distribution of the $Z_{iq}|X$ doesn't factorize. We rely on the variational approach, which consists in approximating the distribution of Z_{iq} by an appropriate distribution. This strategy is implemented in the **mixer** `R` package that you can use to fit SBMs.

4. *Simple tests.* Use the `rNetwork` function developed during the first practical to draw SBM with various structures. Show that you can recover the original parameter thanks to the `mixer R` function. Also train yourself on the corresponding `getModel` and `plot` functions.
5. *Simulations 1 : assessing parameter estimation.* Consider for instance an affiliation network with 3 classes. Do some simulations showing that the mean square error of $\hat{\pi}_{q\ell}$ decrease when p increases. To do so, assume that you know the correct number of classes in your graph when calling the `mixer` function.
6. *Simulations 2 : assessing clustering efficiency.* Again, consider an affiliation network with 3 classes. Do some simulations showing that the adjusted rand index between the classification recovered by `mixer`, when the number of classes is chosen so as to maximize the ICL, increases when p increases. You may use the function `adjustedRandIndex` from the **mclust** package to compare two classifications
7. *Escherichia coli regulatory network.* Consider the network found in the **Ecoli.data** dataset from the **sand** package. Symmetrize the network and remove the isolated nodes. Then, apply the variational EM and the variational Bayes EM to this network. Chose the number of classes with the ICL and ILvB, respectively. Conclusion ?