Stochastic block model and Spectral Clustering

julien.chiquet@gmail.com

M2 ISG - summer 2016 -
network modelling - practical 1

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 : undirected random network

1.1 Notations

We let $\mathcal{P} = \{1, \ldots, p\}$ be a set of nodes. Presence or absence of an edge between two nodes i and j is described by the random variable

$$\Theta_{ij} = \mathbb{1}_{\{i \leftrightarrow j\}}.$$

We assume by convention that the nodes are not connected to themselves, that is, $\Theta_{ii} = 0$ for all $i \in \mathcal{P}$. we denote by K_i the degree of node i, i.e.,

$$K_i = \sum_{j \in \mathcal{P}} \Theta_{ij}. \tag{1}$$

1.2 Erdös-Rényi model

In this model, all variables $(\Theta_{ij})_{i,j \in \mathcal{P}}$ are independent from each other and follow the same Bernoulli distribution

$$\Theta_{ij} \sim \mathcal{B}(\pi), \quad \pi \in [0, 1]. \tag{2}$$

1.3 Stochastic Block Model

This model has several representation. We adopt the one given by Daudin, Picard and Robin (2007), known as "mixture model for random graphs". This model spreads the nodes 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. Thus

$$\alpha_q = \mathbb{P}(Z_{iq} = 1) = \mathbb{P}(i \in q), \quad \text{such that} \sum_q \alpha_q = 1. \tag{3}$$

It is straightforward to see that $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iQ})$ has a multinomial distribution

$$\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\alpha}). \tag{4}$$

Finally, let $\pi_{q\ell}$ be the probability that a node in class q connects to a node in class ℓ [1]. The probability for having edge between nodes i and j is defined *conditionally on* the classes they belong to :

$$\Theta_{ij} | \{i \in q, j \in \ell\} \sim \mathcal{B}(\pi_{q\ell}), \quad i \neq j. \tag{5}$$

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 Preliminary questions

1. *degree distribution.* For the Erdös-Rényi model, show that the degree distribution of a node may be approximated by a Poisson distribution with a well chosen parameter. Adapt the result to the the stochastic block model in order to determine an approximation of the distribution of $K_i | \{i \in q\}$, *i.e.* the degree of i conditional on its class belonging. Derive a approximated distribution for K_i with given parameters.
2. *Connectivity.* For two classes q and ℓ, let us define "connectivity" by

$$C_{q\ell} = \sum_{i<j} Z_{iq} Z_{j\ell} \Theta_{ij},$$

which represents the number of edges between two classes. Derive $\mathbb{E}[C_{q\ell}]$.

3 Simulation

1. *random graph generation.* Write a function which takes the parameters $p, \boldsymbol{\alpha}, \boldsymbol{\pi}$ for arguments and draw a random graph encoded with an adjacency matrix $\boldsymbol{\Theta}$. This function also sends back a vector of classes $\mathbf{C} \in \mathcal{Q}^p$ such that $C_i = q$ if $i \in q$.
2. *Examples.* Draw some graphs with the following topologies, by choosing :
   — Erdös-Rényi random networks,
   — Affiliation networks, *i.e.* with clusters of highly connected nodes,
   — star networks, *i.e.* networks where a few nodes are highly connected to the others, the later being unconnected with each other.
   These networks can be represented via an image of the corresponding adjacency matrix, or thanks to the `R`-package **igraph**.
3. **Optional** *Degree distribution.* For a graph with a topolgy chosen from above and a fixed number of nodes, represent in the same Figure the theoretical distribution and the empirical distribution of the K_i. The same by computing the empirical distribution on n graphs drawn with the same set of parameters.

1. Since the network is undirected, the matrix $\boldsymbol{\Theta}$ is symmetric and so $\pi_{q\ell} = \pi_{\ell q}$.

4. **Optional** *Connectivity.* For a graph with a topolgy chosen from above and a fixed number of nodes, compare the empirical connectivity and the theoretical connectivity of the different classes. Do the same by averaging the empirical connectivities on n graphs drawn with the same set of parameters.

4 Spectral clustering

Spectral clustering is an algorithm for finding efficiently clusters in affiliation networks by applying an (unsupervised) clustering method such as the K-means in the spectral domain of the adjacency matrix. To this end, we introduce the normalized graph Laplacian $\mathbf{L} = (L_{ij})$ associated with a graph $\mathcal{G}$:

$$\mathbf{L} = \mathbf{I}_p - \mathbf{D}^{-\frac{1}{2}}\mathbf{\Theta}\mathbf{D}^{-\frac{1}{2}},$$

where $\mathbf{I}_p$ is the identity matrix with size $p \times p$ and$\mathbf{D}$ a diagonal matrix with the respective degrees of the nodes, that is to say, $D_{ii} = K_i$.

The spectral clustering algorithm that you must implement is as follows :

Input : Adjacency matrix $\mathbf{\Theta}$, number of classes Q

Compute the normalized graph Laplcian $\mathbf{L}$ Compute the first Q eigen vectors of $\mathbf{L}$ Define $\mathbf{U}$, the $p \times Q$ matrix that encompasses these Q vectors Define $\tilde{\mathbf{U}}$, the row-wise normalized version of $\mathbf{U}$

$$\tilde{u}_{ij} = \frac{u_{ij}}{\|\mathbf{U}_i\|_2},$$

where $\mathbf{U}_i$ is the ith row of $\mathbf{U}$ Apply k-means to $(\tilde{\mathbf{U}}_i)_{i=1,\dots,p}$

Output : vector of classes $\mathbf{C} \in \mathcal{Q}^p$, such as $C_i = q$ if $i \in q$

Algorithme 1 : Spectral Clustering par Ng, Jordan ant Weiss (2002)

1. Apply this algorithm on affiliation networks drawn from your own implementation. Plot images of the adjacency matrix, of the graph Laplacian and of the input matrix of the k-means. Comment.
2. Install the package **sand** and load the Ecoli data set. From the object `regDB.adj`, create a symmetrix adjacency matrix of the E coli network. Remove any isolated nodes. Perform spectral clustering on the matrix with various number of classes. Represent the resulting adjacency matrix the rows and columns of which are reordered according to the class belonging.