Mixture Models for Graph Clustering

Sofiane Ezzehi [1] Bastien Le Chenadec [1] Theïlo Terrisse [1]

École des Ponts ParisTech

Graph Clustering

Graph clustering involves the identification of clusters within a graph, where nodes exhibit denser connections among themselves than with the rest of the network. This task is crucial for conducting comprehensive macro- and mesoscopic analyses of large complex systems, such as social networks. As exact graph clustering is recognized as an NP-hard problem, numerous clustering methods have been developed [2]. Notably, model-based approaches fit a mathematical model to a graph, providing an explanatory framework for observed connectivity patterns. In [1], J.-J. Daudin *et al.* investigate the renowned **Stochastic Block Model (SBM)**, proposing a **variational Expectation-Maximization (EM)** algorithm to optimize model fit for a given graph.

The Stochastic Block Model

Stochastic Block Model: A *mixture* model for graphs where each node is assigned to a *class* and the edge probability between nodes are conditioned on their class memberships.

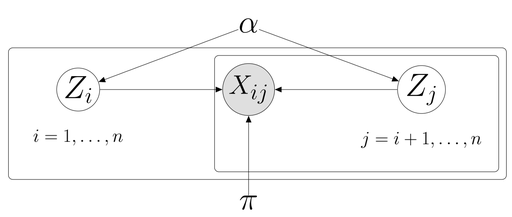


Figure 1. Graphical model of the SBM

Formally, let $\mathcal{G}$ be an undirected graph with n nodes and no self-loop, and X its adjacency matrix, *i.e.* $X_{ij} = 1$ if an edge exists between i and j, and $X_{ij} = 0$ otherwise.

The model is parametrized by Q the number of classes, $\alpha \in [0,1]^Q$ the prior distribution on the classes, and $\pi \in [0,1]^{Q \times Q}$ the probability of an edge between two nodes of different classes. We also introduce the random variables $Z_i \in \{0,1\}^Q$ for $i \in [\![1, n]\!]$, that represent the membership of node i to each class. The prior distributions on Z and X are given by :

$$\begin{cases} \forall i \in [\![1,n]\!], \quad \sum_{q=1}^{Q} Z_{iq} = 1 & \text{(unique class)} \\ \forall q \in [\![1,Q]\!], \quad \mathbb{P}(Z_{iq} = 1) = \alpha_q & \text{(class distribution)} \\ \forall q,l \in [\![1,Q]\!], \quad \forall i \neq j \in [\![1,n]\!], \quad \mathbb{P}(X_{ij} = 1 \,|\, Z_{iq} = 1, Z_{jl} = 1) = \pi_{ql} & \text{(edge probability)} \end{cases}$$

The variational Expectation-Maximization algorithm

In [1], Daudin *et al.* propose a variational EM algorithm to fit the SBM.

$$\log \mathcal{L}(X, Z) = \sum_i \sum_q Z_{iq} \log \alpha_q + \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{iq} Z_{jl} \log \left(\pi_{ql}^{X_{ij}} \left(1 - \pi_{ql}\right)^{1 - X_{ij}} \right)$$

$\implies$ No closed-form solution for the parameters of the model from the likelihood.
Problem : Using an EM algorithm, no closed-form solution for the E-step either.
Solution: Daudin *et al.* search for an approximated distribution R_X of Z given X, among the family of product of multinomial distributions, thus introducing the random variables $\tau_i \in [0,1]^Q$ for $i \in [\![1, n]\!]$. This approximation leads to maximizing the following lower bound of $\log \mathcal{L}(X)$:

$$\mathcal{J}(R_X) = \log \mathcal{L}(X) - \mathrm{KL}[R_X(\cdot), P(\cdot|X)]$$

The authors derive a closed form solution for the M-step, and a **fixed-point relation** for the E-step :

$$\begin{cases} \text{(E-step)} & \hat{\tau}_{iq} \propto \alpha_q \prod_{j \neq i} \prod_l \left[\pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{1 - X_{ij}} \right]^{\hat{\tau}_{jl}} \\ \text{(M-step)} & \hat{\alpha}_q = \frac{1}{n} \sum_i \tau_{iq} \quad \text{and} \quad \hat{\pi}_{ql} = \frac{\sum_{i \neq j} \tau_{iq} \tau_{jl} X_{ij}}{\sum_{i \neq j} \tau_{iq} \tau_{jl}} \end{cases}$$

Algorithm 1 Variational Expectation-Maximization Algorithm

Input: Adjacency matrix X, number of communities Q
Initialize: Initialize τ_{iq} for $i \in [\![1, n]\!]$, $q \in [\![1, Q]\!]$
while not converged **do**
  E-step:
  for $i \in [\![1, n]\!]$ **do**
    for $q \in [\![1, Q]\!]$ **do**
      Fixed-point algorithm:
      $\hat{\tau}_{iq} \propto \alpha_q \prod_{j \neq i} \prod_l \left[\pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{1 - X_{ij}} \right]^{\hat{\tau}_{jl}}$ ▷ Update class memberships
  M-step:
  for $q \in [\![1, Q]\!]$ **do**
    $\hat{\alpha}_q = \frac{1}{n} \sum_i \tau_{iq}$ ▷ Update class proportions
    $\hat{\pi}_{ql} = \frac{\sum_{i \neq j} \tau_{iq} \tau_{jl} X_{ij}}{\sum_{i \neq j} \tau_{iq} \tau_{jl}}$ ▷ Update edge probabilities
Return: $\hat{\alpha}, \hat{\pi}, \hat{\tau}$

Some alternative algorithms

- **Newman *et al.* [3]** use a finer definition for clusters. While Daudin *et al.* define a cluster-cluster affinity (nodes belonging to the same cluster behave the same way), the model of Newman introduces a cluster-node affinity. Therefore, edges between different nodes of the same two clusters are not necessarily equivalent. $\implies$ $\Pi_{q,i}$ denotes the probability of an edge between a node in cluster q and node i.
  The expected log-likelihood (with respect to Z) is

$$\mathcal{L}(X) = \sum_{i=1}^{n} \sum_{q=1}^{Q} \tau_{iq} \left[\log(\alpha_q) + \sum_{j=1}^{n} X_{ij} \log(\Pi_{q,j}) \right] \quad \text{where} \quad \tau_{iq} = \mathbb{P}(Z_{iq} = 1 | X, \alpha, \Pi).$$

  Denoting k_i he degree of node $i \in [\![1, n]\!]$, the EM updates are

$$\text{(E-step)} \ \hat{\tau}_{iq} = \frac{\alpha_q \prod_{j=1}^{N} \Pi_{qj}^{X_{ij}}}{\sum_{s=1}^{Q} \alpha_s \prod_{j=1}^{N} \Pi_{sj}^{X_{ij}}}; \qquad \text{(M-step)} \ \hat{\alpha}_q = \frac{1}{n} \sum_{i=1}^{n} \tau_{iq} \ \text{and} \ \hat{\Pi}_{qj} = \frac{\sum_{i=1}^{n} X_{ij} \tau_{iq}}{\sum_{i=1}^{n} k_i \tau_{iq}}$$

  The update of τ (E-step) is now **deterministic**, with no resort to fixed-point methods.
- **Spectral clustering [4]** uses the eigenvectors of the Laplacian matrix of the graph to cluster the nodes. The k first eigenvectors of $L = D - X$ are stacked in a matrix $U \in \mathbb{R}^{n \times k}$ and a k-means clustering is performed on the rows of U.

Experiments: SBM dataset

Algorithm 1 is implemented in PyTorch. To test it, a first dataset is created by sampling the SBM for parameters given in **Table 1**. For each experiment, 10 graphs are generated and the algorithm is run for 100 iterations on each graph.

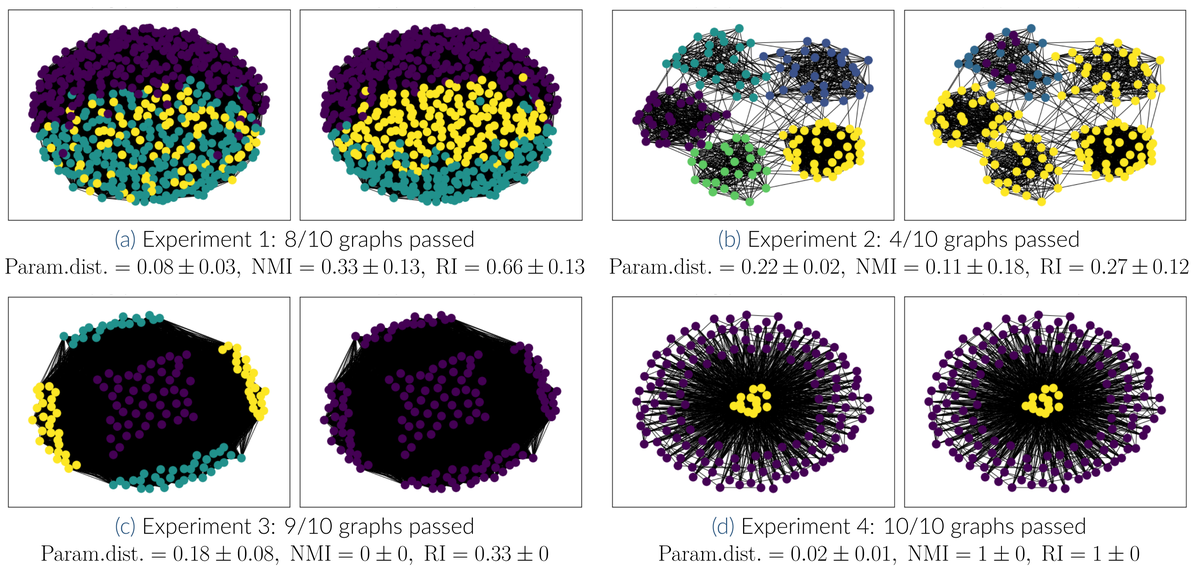
Figure 2 shows the results.

#	Experiment Name	Hyper-parameters n	Q	Parameters α	π
1	Random	500	3	$\alpha \sim \mathrm{Dir}(1.5)$	$\pi_{ij} \sim \mathcal{U}([0,1))$
2	Homophilic	150	5	$\alpha_i = (\frac{1}{Q})_i$	$\pi_{ii} \sim \mathcal{U}([0.5,1)),\ \pi_{ij} = 0.01$
3	Heterophilic	150	3	$\alpha_i = (\frac{1}{Q})_i$	$\pi_{ii} = 0.01,\ \pi_{ij} = 0.9$
4	Dense minority	150	2	$\alpha^T = (\frac{9}{10}, \frac{1}{10})$	$\pi = \begin{pmatrix} 0.01 & 0.7 \\ 0.7 & 0.8 \end{pmatrix}$

Table 1. Parameters for generating the SBM dataset.
Dir is the Dirichlet law. In experiment 1, π is scaled and made symmetric.

- **Metric for model-fitting capacity**: Sum of (cluster-permutation-invariant) Euclidean distances between predicted (α, π) and the ground truth of Table 1. The result is normalized by the number of classes.
- **Metric for clustering capacity**: Normalized Mutual information (NMI) and Rand Index (RI).

NB: The fixed-point algorithm does not always converge within 1000 iterations for all graphs.



(a) Experiment 1: 8/10 graphs passed
$\text{Param.dist.} = 0.08 \pm 0.03,\ \text{NMI} = 0.33 \pm 0.13,\ \text{RI} = 0.66 \pm 0.13$

(b) Experiment 2: 4/10 graphs passed
$\text{Param.dist.} = 0.22 \pm 0.02,\ \text{NMI} = 0.11 \pm 0.18,\ \text{RI} = 0.27 \pm 0.12$

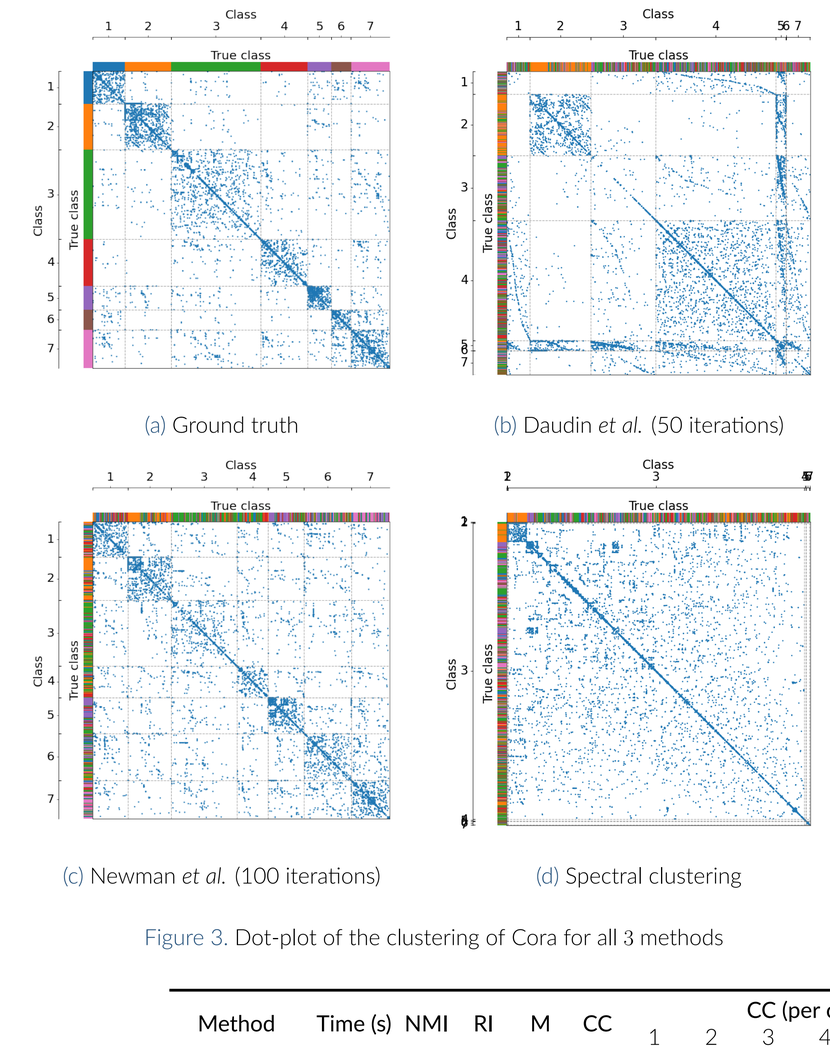
(c) Experiment 3: 9/10 graphs passed
$\text{Param.dist.} = 0.18 \pm 0.08,\ \text{NMI} = 0 \pm 0,\ \text{RI} = 0.33 \pm 0$

(d) Experiment 4: 10/10 graphs passed
$\text{Param.dist.} = 0.02 \pm 0.01,\ \text{NMI} = 1 \pm 0,\ \text{RI} = 1 \pm 0$

Figure 2. Results for the SBM dataset experiments. Left: sample graph; right: associated prediction.

Experiments: Cora dataset

All three methods are tested on the **Cora citation network** [5]: an undirected graph of 2708 nodes (*i.e.*, scientific papers) connected by 5429 edges (*i.e.*, citations). Nodes are labeled into 7 different classes:
1: Cases – **2:** Genetic Alg. – **3:** Neural Net. – **4:** Prob. Methods – **5:** Reinforcement Learning – **6:** Rule Learning – **7:** Theory



(a) Ground truth

(b) Daudin *et al.* (50 iterations)

(c) Newman *et al.* (100 iterations)

(d) Spectral clustering

Figure 3. Dot-plot of the clustering of Cora for all 3 methods

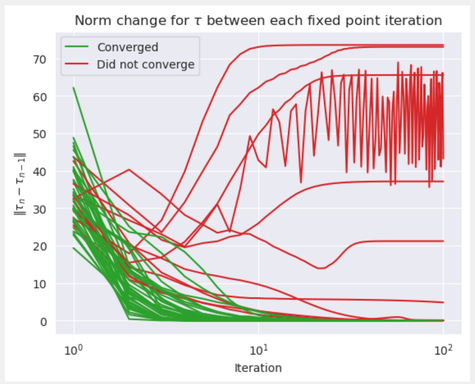
- **Daudin-EM** yields similar results as **Newman-EM** for the supervised metrics.
- **Daudin-EM** reveals some good-quality clusters (2, 4) despite some clusters having a clustering coefficient of 0.
- **Newman-EM** predicts the closest clustering coefficients to the ground truth.
- **Spectral clustering** performs comparatively poorly.

Method	Time (s)	NMI	RI	M	CC	CC (per cluster) 1	2	3	4	5	6	7
Ground truth	-	-	-	0.64	0.09	0.19	0.06	0.12	0.23	0.10	0.22	0.16
Daudin-EM	3802	0.15	0.70	0.22	-	0	0.16	0	0.29	0.25	0	0.86
Newman-EM	27	0.18	0.76	0.53	-	0.25	0.06	0.16	0.28	0.12	0.29	0.18
Spectral	1	0.01	0.21	0.02	-	0	0	0.09	0	0	0.66	0

Table 2. Results on the Cora dataset. **M**: Modularity; **CC**: Clustering Coefficient.

Discussion

- While Daudin-EM [1] can reveal both homophilic and heterophilic properties of a graph, it is not as powerful as Newman's variant (*cf.* Cora experiments).
- The algorithm is sensitive to the distribution of node degrees (*cf.* SBM experiments).
- Lack of guarantee of convergence of the fixed-point algorithm (see opposite figure: out of 50 runs of E-step with different initializations of $(\hat{\alpha}, \hat{\pi})$, only 40 runs converge within 100 iterations).



References

[1] J.-J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, June 2008.

[2] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, February 2010.

[3] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *PNAS*, 104, 2007.

[4] Andrew Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 2001.

[5] Prithviraj Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3), 2008.