Project Report: A mixture model for random graphs

Sacha Braun
sacha.braun@polytechnique.edu
École Polytechnique
Palaiseau, France

Alexandre François
alexandre.francois@polytechnique.edu
École Polytechnique
Palaiseau, France

Abstract

Traditional graph analysis methods predominantly focus on node degree to infer graph structure, typically grouping nodes into classes based on high interaction rates within each class. However, this approach often fail to capture the underlying structure of a graph. In this report, we delve into a mixture model that shifts the focus from nodes to edges, thereby offering a more nuanced depiction of graph structure. Our study emphasizes the importance of parameter initialization in the algorithm and demonstrates its application through the analysis of two real-world datasets. We explore how this edge-centric approach provides a detailed understanding of graph topology, capturing complex structures that node-based methods may miss. However, the effectiveness of this method is somewhat constrained by the graph's size. This limitation highlights the need for further refinement to extend its applicability to larger networks.

1 Introduction

Graph models, particularly mixture models, have predominantly focused on node analysis. The node distribution in models like the Erdös-Rényi model (Erdös and Rényi 1959) has been extensively explored but typically emphasize the degree distribution of nodes, overlooking the intricate topological structures of graphs. As the simplest form of the Erdös-Rényi model is denoted as $\{X_{ij}\}, X_{ij} \sim \mathcal{B}(p)$, where (X_{ij}) signifies the presence or absence of an edge between nodes i and j, those models often fail to incorporate edge-specific information, uniformly applying the probability of edge formation between any two nodes.

Other researches have attempted to introduce underlying structures, proposing a mixture model for degree distribution. Here, nodes belong to specific classes with a prior probability, leading to a more nuanced degree distribution, often hypothesized to follow a Poisson distribution (Handcock and Jones 2004). It has been claimed that many networks observed across technological, social, and biological domains are a scale-free, characterized by a few highly connected nodes and a majority with minimal connections, and that the degree distribution in these networks often fits a power-law pattern. However, (Stumpf et al. 2005) showed that random subnets sampled from scale-free networks are not themselves scale-free which contrast the Erdös-Rényi. These issues have thus prompted the development of novel models more adept at capturing the overall structure of a graph.

The focal study of this report is a deep analysis of the paper (Daudin and Robin. 2008) that introduces a novel approach to graph analysis. The Erdös-Rényi Mixture for Graphs (ERMG) shifts the focus from node distribution to edge distribution. This pivotal change enables the conditioning of an edge's distribution, belonging to class q, based on its connectivity across all graph classes. This model extends beyond traditional node distribution models, encompassing a more comprehensive framework. It facilitates the generation of new data with specific structural properties and supports mathematical analysis of the generated graph's attributes. Using the mixture model, a variational EM (Estimation-Maximization) algorithm is employed for parameter estimation. This algorithm works even in large networks (exceeding 1,000 nodes), a unique capability for inferential method at the time of the paper's publication, but not on larger graphs which limits its application.

In this report, we will delve into the model's theoretical underpinnings, linking them to concepts covered in the course "Introduction to Probabilistic Graphical Models and Deep Generative Models" (Latouche and Mattei 2023). We will implement the model from scratch and conduct various studies on its estimation aspect. One focus will be identifying scenarios where the model may not yield satisfactory results, particularly in terms of convergence. Like a standard EM algorithm, the initialization of parameters in this model significantly influences its performance. We will examine the impact of different initialization methods on the algorithm's convergence, implementing methods from several papers (Fortunato 2010, Schaeffer 2007). It's worth noting that while most clustering methods emphasize groups with high node connectivity, the community interpretation in our focal study may differ, as many nodes may be classified in the same cluster even if the do not share any edge, in a star graph for instance. Lastly, we plan to apply the model to real-world graphs, with a special emphasis on analyzing a graph representing the French play "Les Misérables".

2 Methodology

2.1 The model

Continuing from the introduction, let us delves into the fundamental assumptions and properties of the Erdös-Rényi Mixture for Graphs (ERMG). The model is built on two primary assumptions:

- The nodes in the graph are categorized into classes Q, with α_q representing the probability that a randomly selected node belongs to class q.
- The edge distribution between two nodes is contingent upon their class membership, expressed as $X_{ij}|\{i \in q\} \cap \{j \in l\} = \mathcal{B}(\pi_{ql})$. Here, $\{i \in q\}$ indicates that node i is a member of class q.

For analytical convenience, let's define a set of binary variables $\mathcal{Z}$, which represent the class distribution of nodes within the graph, such that $P(Z_{iq} = 1) = \alpha_q$. It is also important to note that the matrix π should be symmetric and contain values between 0 and 1, though it need not be stochastic. Additionally, the class probabilities must sum to one, i.e., $\sum_q \alpha_q = 1$.

These assumptions facilitate the generation of graphs with specific structures, as will be demonstrated later in Section 3. For instance, to model a star-shaped graph, we could define two classes: class 0 representing the 'center' and class 1 the 'edges' of the star.

MVA 2023, December 2023, Paris, FR
© 2023

By setting the diagonal coefficients of matrix π to zero, we ensure connections only between nodes of different classes, mimicking a star's structure.

Furthermore, we aim to derive certain properties of graphs characterized by the structure parameters π and α, which will be crucial for interpreting the results in Section 3 and in the notebook.

Conditional Distribution of the Degree: As detailed in Appendix 6, the degree distribution of a node i belonging to class q can be modeled as

$$deg(\text{node}_i)|\{i \in q\} \sim B(n-1, \pi_q) \approx P(\lambda)$$

, where this approximation holds true for small p and large n, such that $\lambda = (n-1)p$. The specific definitions of "small" and "large" in this context are further explained in Appendix 6.

Clustering Coefficient: The clustering coefficient (Watts and Strogatz 1998) for a graph generated by the ERMG can be calculated using the formula:

$$c = \frac{\sum_{q,l,m} \alpha_q \alpha_l \alpha_m \pi_{ql} \pi_{qm} \pi_{lm}}{\sum_{q,l,m} \alpha_q \alpha_l \alpha_m \pi_{ql} \pi_{qm}}$$

This coefficient provides a measure of the degree to which nodes in a graph tend to cluster together.

2.2 Estimation

Now that we have a model to generate graph given some parameters, it is natural to look for algorithms to estimate the parameters for a given graph, namely the latent clustering priors π and α (recall that α_q denotes the probability for a vertex to be in cluster q, and π_{ql} denotes the probability for a vertex in cluster q to be connected to a vertex in cluster l.) Let's first compute the log-likelihood of the problem:

2.2.1 Log-likelihood.

Denoting $\mathcal{X}$ the random variables describing the edges' distribution of the graph, and $\mathcal{Z}$ the vertices' distribution into clusters, let's write the log-likelihood:

$$\log \mathcal{L}(\mathcal{X}, \mathcal{Z}) = \log \mathcal{L}(\mathcal{Z}) + \log \mathcal{L}(\mathcal{X}|\mathcal{Z})$$

Let's break down this expression:

$$\begin{aligned}
\log \mathcal{L}(\mathcal{Z}) &= \log\Big(\prod_{i=1}^{n}\prod_{q=1}^{Q} \mathrm{p}(Z_{iq})\Big) \\
&= \sum_{i=1}^{n}\sum_{q=1}^{Q} Z_{iq} \log(\alpha_q) \\
\log \mathcal{L}(\mathcal{X}|\mathcal{Z}) &= \log \prod_{i=1}^{n}\prod_{\substack{j<i \\ j\neq i}}^{n}\prod_{q=1}^{Q}\prod_{\ell=1}^{Q} \left(\pi_{ql}^{X_{ij}} \cdot (1-\pi_{ql})^{(1-X_{ij})}\right)^{Z_{iq}Z_{jl}} \\
\log \mathcal{L}(\mathcal{X}|\mathcal{Z}) &= \frac{1}{2}\sum_{i\neq j}\sum_{q,\ell} Z_{iq}Z_{j\ell} \log b(X_{ij}; \pi_{q\ell})
\end{aligned}$$

and $b(x;\pi) = \pi^x(1-\pi)^{1-x}$.

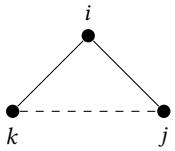
Because of the latent variables $\mathcal{Z}$, this expression does not seem to be simplified easily. Then, it seems logical to look for a conditional expectation of the log-likelihood such that:

$$\begin{aligned}
Q(\mathcal{X}) &= \mathbb{E}\left[\log \mathcal{L}(\mathcal{X}, \mathcal{Z})|\mathcal{X}\right] \\
&= \sum_i \sum_q \tau_{iq} \log(\alpha_q) + \frac{1}{2}\sum_{i\neq j}\sum_{q,l} \theta_{i,j,q,l} \log(b(X_{ij}, \pi_{ql}))
\end{aligned}$$

with $\theta_{ijql} = \mathbb{E}[Z_{iq}Z_{jl}|\mathcal{X}]$ and $\tau_{iq} = \mathbb{P}(Z_{iq} = 1|\mathcal{X})$ being the posterior distribution.

2.2.2 Variational approach.

Intuitively, we have here a hierarchical model, with latent variables describing the belonging to classes, every class defining a specific distribution. Naturally, that leads us to consider a expectation-maximisation (EM) algorithm to estimate the parameters. Nevertheless, the expectation step requires to compute the posterior distribution (to optimize the ELBO) which is not possible here, according the the following example.



$$\Pi = \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix}$$

On this graph, the posterior probability of i belonging to class 1 ($\mathbb{P}(Z_{i1} = 1)$ depends on the whole set of edges $\mathcal{X}$, and not only on the neighbourhood of i. Actually, as no connection is possible between two vertices that are not in the same class, i, j and k are in the same class. But in that case, whether i belongs to cluster 1 or cluster 2 depends on the value of the edge X_{jk}: if $X_{jk} = 0$, then $\mathbb{P}(Z_{i1} = 1|\mathcal{X}_i, \mathcal{X}_{jk}) = 0$, and if $X_{jk} = 1$, then $\mathbb{P}(Z_{i1} = 1|\mathcal{X}_i, \mathcal{X}_{jk}) > 0$. So, from a more general point of view, $\mathbb{P}(Z_{iq} = 1|\mathcal{X}_i) \neq \mathbb{P}(Z_{iq} = 1|\mathcal{X})$, and the posterior distribution cannot be computed easily. In comparison, in classical EM, the posterior distribution can be computed using a neighbourhood for every data, making it more tractable. Therefore, we need to use a variational approach to optimize the classical EM lower bound of $\log \mathcal{L}(\mathcal{X})$, called the ELBO.

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

where $\mathrm{KL}\left[R_{\mathcal{X}}(\cdot), \mathbb{P}(\cdot|\mathcal{X})\right]$ is the Kullback-Leibler divergence.
In a classical EM, the maximization step consists of setting the KL-divergence to 0, setting $R_{\mathcal{X}} = \mathbb{P}(\cdot|\mathcal{X})$. As mentionned above in our case, that is not possible because of the complete non-tractability of the posterior in that case. That is where the variational approach of EM comes in: we will look for the best $R_{\mathcal{X}}$ in terms of Kullback-Leibler divergence minimization, restricted to a certain class of functions. This variational strategy has been described in (Jordan et al. 1999) and (Jaakola 2000), and alternates the maximization of $\mathcal{J}(R_{\mathcal{X}})$ (i) with respect to $R_{\mathcal{X}}$ and (ii) with respect to parameters α and π. We will assume that the approximate conditional distribution $R_{\mathcal{X}}$ takes the following form:

$$R_{\mathcal{X}}(\mathcal{Z}) = \prod_i h(\mathcal{Z}_i; \tau_i)$$

where τ_i=($\tau_{i1}, \ldots, \tau_{iQ}$) and $h(\cdot;\tau)$ denotes the multinomial distribution with parameter τ. τ_{iq} can be interpreted as an approximation of the posterior distribution $\mathbb{P}(Z_{iq} = 1)$. Here are a few insights of why we make the assumption that the approximal conditional distribution has the following form:

- **Independence Assumption:** The variational distribution $R_{\mathcal{X}}(\mathcal{Z})$ is assumed to factorize into a product of independent distributions for each latent variable. That is, if $Z = \{Z_1, Z_2, \ldots, Z_n\}$, then $R_{\mathcal{X}}(\mathcal{Z}) = \prod_{i=1}^{n} h(\mathcal{Z}_i; \tau_i)$, where each term is independent.
- **Multinomial Distribution in Mixture Models:** Each factor in the product form of the variational distribution often takes the form of a multinomial distribution. This reflects

the scenario where each node in the graph is assumed to belong to one of several possible categories or clusters, with the multinomial distribution capturing the probabilities of these cluster assignments for each node.
- **Computational-accuracy Trade-off:** The product form represents a balance between computational tractability and the accuracy of the inference. It is chosen in our scenarios because exact inference is infeasible, accepting some loss of accuracy for practical computation.

Now, let's look for the actual variational EM algorithm and how updates at each step are made. As in a classical EM, the algorithm breaks down into two steps, an estimation step and a maximization step:

$$(\alpha^{(h+1)}, \pi^{(h+1)}) = \arg\max_{(\alpha,\pi)} \mathcal{J}\left(R_{\mathcal{X}}; \{\tau_i^{(h)}\}, \alpha, \pi\right),$$
$$\{\tau_i^{(h+1)}\} = \arg\max_{\{\tau_i\}} \mathcal{J}\left(R_{\mathcal{X}}; \{\tau_i\}, \alpha^{(h+1)}, \pi^{(h+1)}\right).$$

Let's now see and understand how these updates are made:

Estimation step: Given the variational parameters $\{\tau_i\}$, the values of parameters α and π that maximize $\mathcal{J}(R_{\mathcal{X}})$ are

$$\hat{\alpha}_q = \frac{1}{n}\sum_i \hat{\tau}_{iq}, \quad \hat{\pi}_{ql} = \frac{\sum_{i\neq j}\hat{\tau}_{iq}\hat{\tau}_{jl}X_{ij}}{\sum_{i\neq j}\hat{\tau}_{iq}\hat{\tau}_{jl}}.$$

Proof: Being a probability law, we must have $\sum_q \hat{\alpha_q} = 1$. So we have to optimize a constrained convex optimization problem. The Lagrangian writes

$$L(R_{\mathcal{X}}, \alpha, \lambda) = \mathcal{J}(R_{\mathcal{X}}) + \lambda\left(\sum_q \alpha_q - 1\right)$$

Using the expression of $J(R_{\mathcal{X}}$, the derivative of $J(R_{\mathcal{X}}$ with respect to α_q is straightforward and leads to

$$\frac{\partial J(R_{\mathcal{X}})}{\partial \alpha_q} = \frac{1}{\alpha_q}\sum_i \tau_{iq}$$

Plugging it into the derivative of the Lagrangian, the rest follows immediately.
To derive the update of π, let's remark that

$$\frac{\partial \log b(x,\pi)}{\partial \pi} = \frac{x}{\pi} + \frac{(1-x)}{\pi}$$

and plug it into the derivative $\frac{\partial J(R_{\mathcal{X}})}{\partial \pi_{ql}}$ and then in the derivative of the Lagrangian.

Maximization step: Given parameters α and π, the optimal variational parameters $\{\hat{\tau}_i\} = \text{argmax}_{\tau_i}\mathcal{J}(R_{\mathcal{X}})$ satisfy the following fix point relation:

$$\hat{\tau}_{iq} \propto \alpha_q \prod_{j\neq i}\prod_\ell b(X_{ij}; \hat{\pi}_{ql})^{\hat{\tau}_{j\ell}}.$$

Proof: Based on the definition of the Kullback-Leibler divergence, we first rewrite $J(R_{\mathcal{X}})$ as

$$\begin{aligned} J(R_{\mathcal{X}}) &= \sum_z R_{\mathcal{X}}(z)\log\mathbb{P}\{\mathcal{Z}|\mathcal{X}\} - \sum_z R_{\mathcal{X}}(\mathcal{Z})\log R_{\mathcal{X}}(\mathcal{Z}) \\ &= \sum_i\sum_q \tau_{iq}\log\alpha_q \\ &\quad + \frac{1}{2}\sum_{i\neq j}\sum_{q,\ell}\tau_{iq}\tau_{j\ell}\log b(X_{ij};\tau_{q\ell}) \\ &\quad - \sum_i\sum_q \tau_{iq}\log\tau_{iq}. \end{aligned}$$

We now have to maximize $J(R_{\mathcal{X}})$ with respect to the τ_{iq}'s, subject to $\sum_q \tau_{iq} = 1$, for all i, i.e., to maximize $J(R_{\mathcal{X}}) + \sum_i \lambda_i(\sum_q \tau_{iq} - 1)$ where λ_i is the Lagrange multiplier. The derivative with respect to τ_{iq} is

$$\log\alpha_q + \sum_{j\neq i}\sum_\ell \tau_{j\ell}\log b(X_{ij};\tau_{q\ell}) - \log\tau_{iq} + 1 + \lambda_i.$$

This derivative is null iff $\hat{\tau}_{iq}$'s satisfy the relation given in the proposition, $\exp(1+\lambda_i)$ being the normalizing constant.
At the end of the day, these two updates rules enable us to compute a sequence of $\{\{\tau_i^{(h)}\}, \alpha^{(h)}, \pi^{(h)}\}$ such that, according to the estimation and the maximization steps:

$$\mathcal{J}\left(R_{\mathcal{X}}; \{\tau_i^{(h+1)}\}, \alpha^{(h+1)}, \pi^{(h+1)}\right) \geq \mathcal{J}\left(R_{\mathcal{X}}; \{\tau_i^{(h)}\}, \alpha^{(h)}, \pi^{(h)}\right).$$

We perform these updates over a fixed number of iterations, leading to the final convergence of the parameters. We also want to highlight that this inequality may not be true if the fixed update of the τ using the fixed point function did not converge.

2.2.3 Choice of the number of clusters.

In the approach previously described, we have always worked with a fixed number of classes Q. However, this number is of great semantic importance, and its value should not be chosen randomly. In this report, we use a Bayesian model selection criterion, the Integrated Classification Likelihood (ICL) developed by (Christophe Biernacki 1998). For a model m_Q with Q classes, the ICL criterion is

$$\begin{aligned} \text{ICL}(m_Q) &= \max_\theta \log\mathcal{L}(\mathcal{X}, \tilde{\mathcal{Z}}|\theta, m_Q) \\ &\quad -\frac{1}{2}\frac{Q(Q+1)}{2}\log\frac{n(n-1)}{2} - \frac{Q-1}{2}\log(n) \end{aligned}$$

where $\theta = (\alpha, \pi)$ is the entire set of the mixture parameters. Intuitively, this criterion penalize a model that overfits the data by a penalty growing in the number of classes.
However, this criterion requires to know the maximum of the log-likelihood with respect to the set of parameters θ, and the goal of the variational EM algorithm is actually to maximize this log-likelihood. Therefore, it is mandatory to perform a full EM computation for every number of clusters that we want to test, which can in some cases be unefficient, especially for very large graphs for which EM computation is very long.

2.3 Algorithm Performance and Limitations

This section addresses the scenarios where the proposed algorithm might encounter difficulties. Our focus will be on specific cases of the fixed-point function and the necessary minor adjustments for improved algorithmic stability. In the ensuing discussion, i, j denote nodes, while q, l represent clusters.

Vanishing Updates: The estimated values $\hat{\tau}_{iq}$ are updated using a multiplicative function, which can lead to complications under certain conditions. A key issue arises when one element in the update multiplication approaches zero, leading to the updated $\hat{\tau}_{iq}$ becoming a zero value. Specifically, this occurs when $b(X_{ij}\hat{\pi}_{ql})^{\hat{\tau}_{j\ell}} = 0$, which is true when $\hat{\pi}_{ql} = 1$ or $\hat{\pi}_{ql} = 0$.

1. **Case of $\pi_{ql} = 1$:** In scenarios where the true value of the π matrix is one, the algorithm may fail to converge. For the update not to vanish, it is necessary that either $X_{ij} = 1$ or $\hat{\tau}_{jl} = 0$. The first condition is likely for edges between classes q and l that are fully connected ($\pi_{ql} = 1$). However, when the nodes do not both belong to these classes, $b(X_{ij}\hat{\pi}_{ql})$ might be zero, necessitating $\hat{\tau}_{j\ell} = 0$. This requirement can be challenging since $\hat{\tau}_{j\ell}$ values are continuously updated and the converge may not be finished for those values.
2. **Case of $\pi_{ql} = 0$:** Similarly, when $\hat{\tau}_{j\ell}$ has not fully converged, $b(X_{ij}\hat{\pi}_{ql})^{\hat{\tau}_{j\ell}} = 0$ might occur.

To address these issues, a small ϵ value was introduced to prevent vanishing updates: $b(x;\pi) = (\pi+\epsilon)^x(1-\pi+\epsilon)^{1-x}$. This modification enables the algorithm to handle situations where $\pi_{ql} = 1$ or $\pi_{ql} = 0$, a desirable feature for robustness.

Estimation of $\pi_{ql} = 0$: Another challenge arises in a particular setup when $\pi_{ql} = 0$. In this case, the condition $\sum_{i\neq j}\hat{\tau}_{iq}\hat{\tau}_{jl}X_{ij} = 0$ must be met. However, as $\hat{\tau}_{iq}\hat{\tau}_{jl}$ are approximations and not exact zeros, even when $X_{ij} = 1$, the numerator becomes minuscule but non-zero. If we further add the assumption that $\hat{\tau}_{iq}\hat{\tau}_{jl} = 0$ when $X_{ij} = 0$, that is to say that neither that when there is no edge between two nodes, then $\{i \notin q\}$ or $\{j \notin l\}$ (this situation typically arises in star graphs); then the denominator, being equal to the numerator, also approaches zero, leading to an erroneous estimation of π_{ql} as 1 instead of 0. To mitigate this, we enforce $\pi_{ql} = 0$ when the numerator is smaller than a predefined ϵ threshold.

2.4 Different initialization methods

Intuitively in the case of the variational approach for mixture models, the initialization of the approximate posterior probabilities of belonging to a class τ_{iq} plays a crucial role. In fact, the structure and connectivity patterns play a significant role. If the initial approximate posterior probabilities do not capture the inherent modularity or community structure of the graph, the optimization steps might reinforce incorrect assignments, leading to a clustering that does not accurately reflect the true community structure. The approximate posteriors guide the early iterations of the algorithm, affecting both the update of the cluster parameters and the re-estimation of the probabilities themselves in a feedback loop. In this report, we will compare at two different initialization methods, in addition to the random one, and evaluate the different performances of these initializations.

2.4.1 Spectral clustering: Spectral clustering (White and Smyth 2005) is an algorithm that leverages the eigenvalues and eigenvectors of a graph's Laplacian matrix to identify clusters within the graph. The algorithm begins by constructing the adjacency matrix A of the graph $G = (V, E)$. It then computes the Laplacian matrix $L_{rw} = I - D^{-1}A$, where I is the identity matrix and D is the diagonal degree matrix of G, indicating the number of neighbors for each node. The next step involves eigenvalue decomposition of L_{rw}, selecting the eigenvectors corresponding to the d smallest eigenvalues. These eigenvectors are assembled into a matrix $U \in \mathbb{R}^{m\times d}$. Each row of U, denoted by y_i, represents a node in a new d-dimensional space where traditional clustering techniques, such as k-means, can be applied to partition the nodes into k clusters $C_1, C_2, \ldots, C_k$. This method effectively transforms the problem of graph clustering into a geometric one, where spatial closeness in the new eigenvector space implies membership in the same cluster.

2.4.2 Modularity clustering: The idea of the modularity clustering (Jierui Xie 2013) is to iteratively try different clustering possibilities, and to provide a measurement capable of compute how good the clustering is. Modularity is one of the most popular and widely used metrics to evaluate the quality of a network's partition into communities. Considering a specific partition of the network into clusters, modularity measures the number of edges that lie within a cluster compared to the expected number of edges of a null graph (or configuration model), i.e., a random graph with the same degree distribution. In other words, the measure of modularity is built upon the idea that random graphs are not expected to present inherent community structure; thus, comparing the observed density of a subgraph with the expected density of the same subgraph in case where edges are placed randomly, leads to a community evaluation metric. Modularity is given by the following formula:

$$Q = \sum^{n_c}[\frac{l_c}{m} - (\frac{d_c}{2m})^2]$$

where n_c is the number of partitions in the specific case that we try , m is the number of edges of the graph, l_c is the number of edges that lie in the community c and d_c is the sum of the degrees of the nodes in the community c. Modularity takes value in $[-1, 1]$, higher value indicating a higher community structure.
The idea of the modularity clustering is to start with a graph where each node defines a community, and to progressively associate communities that are the most likely to be in the same cluster, according to the maximization of the modularity.
In community detection within graphs, the algorithm focuses on merging only those communities that are linked by edges, as unconnected community pairs do not contribute to an increase in the modularity measure Q. The modularity change ΔQ due to a merge is expressed as $\Delta Q = \frac{1}{m}l_{c_1\cup c_2} - \frac{1}{m}(l_{c_1}+l_{c_2}) + \frac{(d_{c_1}+d_{c_2})^2}{2m} - (\frac{d_{c_1}}{2m})^2 - (\frac{d_{c_2}}{2m})^2$, which is computationally efficient. Post-merge, the edge matrix updates in $O(n)$ time. Given that the algorithm conducts at most $n-1$ merge operations, the total computational complexity amounts to $O((m+n)n)$, reducing to $O(n^2)$ for sparse graphs. The algorithm also computes the modularity Q incrementally, thus streamlining the search for the optimal community structure.

3 Results

3.1 Dataset

For this project, we choose to show our results on two main dataset. The first one is the coappearance network of characters in the novel *Les Miserables* of Victor Hugo. It comprises 77 nodes representing *Les Misérables* characters and 254 edges, each edge connecting two characters that share a common scene in the book (see Appendix 6).

The second one is an X (formerly Twitter) interaction network for the 117$^{\text{th}}$ United States Congress House of Representatives. It

was constructed by first obtaining members' official Twitter handles. The Twitter API was then used to obtain all Tweets by members of Congress between February 9, 2022, and June 9, 2022. Each nodes represent the account of one member of the Congress, and it is connected to others when one of the two member replied, retweeted ou commented one tweet of the other member (Fink et al. 2023). It has has 475 nodes, and 10,222 edges.

The code we used is available on the github repository[1] of the project. We re-coded all the clustering methods as-well as the EM algorithm from scratch in order to show a deep understanding of the algorithms. Since we wanted to work with large graphs, we also propose a PyTorch version of the EM algorithm to leverage GPU and fasten computation time. Please note that all figures can be viewed in a larger format in Appendix 6.

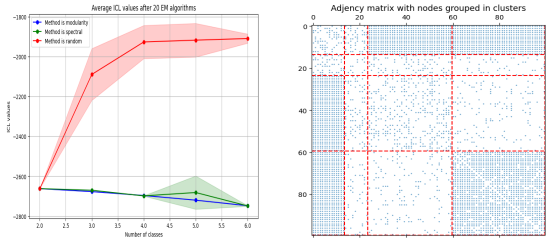
3.2 Impact of initialization

This section presents our findings regarding the influence of initialization methods on the convergence of the algorithm. We conducted our study in two parts: first, we generated various graphs using the ERMG model with specific structures to observe how different initializations affect the algorithm's convergence based on the true π matrix. Subsequently, we applied the same experimental setup to real-world data, where the true values of π are unknown.

3.3 Generated Data Analysis

For graphs generated using the ERMG model, our findings indicate that spectral clustering often yields better performance on certain graph structures compared to modularity-based methods. However, random initialization demonstrated significantly superior performance in specific scenarios. One other particularly noticeable remark is that modularity and spectral clustering methods, known for their low variance (tending to provide consistent clustering for repeated runs on the same data), also offer more stable results than random for a given graph.

Specifically, in the first scenario, when the diagonal coefficients of π are substantially less than 1, indicating a low likelihood of connections within the same class, random initialization outperformed both spectral clustering and modularity Fig. 1. This could be attributed to the inherent design of clustering methods, which typically define communities based on the prevalence of connections within them. In scenarios where π's structure is contrary to this principle, clustering methods may initialize far from the true class distributions of the model.

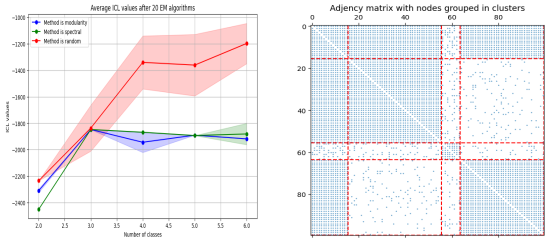


(a) ICL (b) Post-clustering reordered adjacency matrix

Figure 1. Results for the first scenario

[1] https://github.com/ElSacho/Mixture4graph

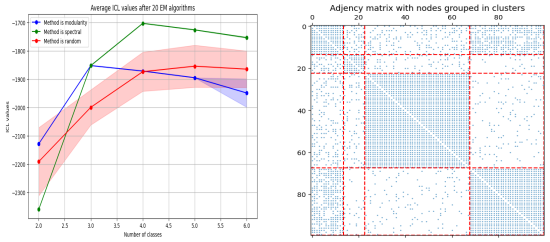
In a second scenario where the diagonal coefficients of π are close to 1, but other coefficients are also high, clustering methods struggle to accurately identify the true classes. Here too, random initialization proved more effective on average Fig. 2. This phenomenon might be linked to an early convergence to a local optimum in the likelihood function, especially in graphs with a higher edge density. These graphs are inherently more complex to analyze, and starting from a likely solution may lead to converge to local optima. In this particular example, it is also worth noting that initializing with the random method converges to the ground truth. However, the ICL values suggest that 4 may not be the optimal number of cluster even if it is indeed the number that has been used to generate the graph. This opens the question on the choice of the ICL value to determine the number of clusters.



(a) ICL (b) Post-clustering reordered adjacency matrix

Figure 2. Results for the second scenario

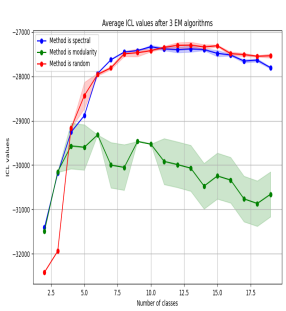
Conversely, in the third scenario, when the diagonal coefficients of π are near 1, and other coefficients are significantly lower, spectral clustering surpassed random initialization in performance Fig. 3. This outcome aligns with expectations, as such initializations help the algorithm in rapidly converging to the optimal solution.



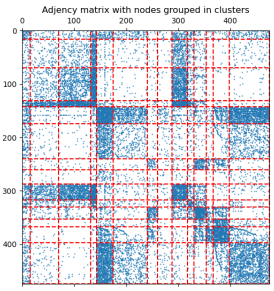
(a) ICL (b) Post-clustering reordered adjacency matrix

Figure 3. Results for the third scenario

Let us now study the results of the algorithm on real-graph. The first one we study is the Twitter interactions one. On this particular dataset, the performance of the random initialization are similar to the one from the spectral clustering model, where the modularity seems to perform poorly Fig 14. We also plotted the reordered adjacency matrix for 13 clusters. We can clearly observe that the model highlights some really interesting results, and succeed in regrouping into classes twitter accounts that share the same structure's connections with other accounts. Unfortunately, twitter does not allow to study non-anonymous accounts and we could therefore not qualitatively interpret those classes segmentations.



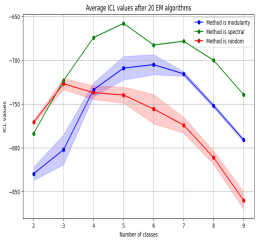
(a) ICL



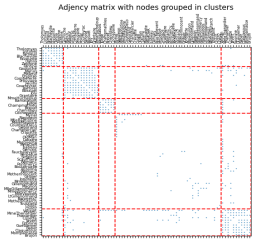
(b) Post-clustering reordered adjacency matrix

Figure 4. Results on the Twitter interactions graphs

The last real graph we will study is the one of "Les Misérables". In the method, spectral clustering outperforms all other methods, and modularity again fails to be the most effective method Fig 5. In this particular example, where the final structure resembles that of scenario 3, the random method works less well than the other two methods, whose initialization is more relevant, as it's better adapted to the final result. We propose a deepened analysis of those results in the next section.



(a) ICL



(b) Post-clustering reordered adjacency matrix

Figure 5. Results on the Les Miserables coappearance networks

In summary, the selection of the initialization method for the ERMG model is of paramount importance and largely depends on the intrinsic configuration of the π matrix. This realization is important for the practical implementation of the model, particularly in handling a variety of graph structures. Random initialization has proven effective across numerous scenarios, yet spectral clustering might surpass its effectiveness when the graph exhibits community clustering. However, the modularity method seems to be less efficient compared to the aforementioned approaches.

3.4 Interpretation of the results from Les Miserables

Intuitively, the mixture model clustering for clusterizing characters in a novel is relevant, as every character can be defined through his relationships and his non-relationships with others: while a simple intra-connectivity algorithm, like spectral clustering, would only look at connexions between connected characters, the global connectivity patterns can somehow unveil some information.

The algorithm seems to split the characters into 5 clusters. Let's break those clusters down:

- **Strong intra-connectivity clusters:** we seem to have 3 out of 5 clusters with very high intra-connectivity, and very low connectivity with other clusters. For example, the first cluster gathers Gavroche, Enjolras or Courfeyrac, who are all members of the ABC community, a group of French revolutionners who share the stage during the very specific part of the Paris' barricades (to see better the names of the character, please refer to Appendix 6). The second cluster gathers Favourite, Dahlia or Zephine, who are old Fantine's friends that only appear together to understand Fantine's past. So this type of clusters gather characters that appear at a very specific time of the novel and only interact with each other, with no other appearance thoughout the timeline.
- **Strong intra and extra connectivity clusters:** this cluster gathers the main characters, or the characters of the present time, that crosses through the different periods of the novel, and interact with many others. A pretty striking case is Marius, who is not clustered with the ABC community as he also shares the stage with Cosette's world. Other clustering algorithms would fail to capture this kind of patterns.
- **Isolated characters in a cluster:** Some clusters can feature high intra-connectivity with a pretty low average extra-connectivity, except for some characters. This is the case for the cluster with Mgr Myriel, who gathers characters that share a common connexion with Mgr Myriel but with no other character. Mgr Myriel in turn, has many connexions with extra-cluster characters. This highlights a very specific kind of character, who bridges the gap between two parts of the story. All the characters with this connexions pattern in this kind of cluster play the same role.

4 Contribution Statement

Task	Sacha	Alexandre
Report	50%	50%
Model	70%	30%
Application to real graphs	50%	50%
Impact of the initialization	30%	70%

Table 1. Contribution of each member to the project.

5 Conclusion

This report has demonstrated the efficacy of the Erdös-Rényi Mixture for Graphs (ERMG) model in elucidating the underlying structures of graphs. The transition from a node-based mixture model to an edge-centric approach has proven pivotal in achieving this deeper insight. Nonetheless, the model's scalability to larger graphs remains a significant challenge. In our experiments, even with substantial GPU resources, memory constraints were encountered when analyzing graphs with more than 500 nodes for 20 classes. Additionally, the computational time can be extensive, especially when multiple iterations are conducted on the same dataset to minimize output variance.

It is crucial to recognize that a high likelihood value for the model does not necessarily equate to a definitive validation of the

theoretical structure it represents. The applicability and success of the model are contingent on the assumption that the actual graph structure can be reasonably approximated by the model's framework. This was evident in our attempts to apply the model to certain graphs, such as the Paris Metro network (6), where the algorithm failed to yield satisfactory results.

Further refinement is needed in the initialization of the τ parameters within the algorithm. Our findings indicate that traditional clustering methods do not universally apply to all graph structures. Surprisingly, a random initialization approach, seemingly suboptimal, emerged as the most effective strategy in our studies. This counterintuitive result underscores the complexity and diversity of graph structures and the need for flexible initialization strategies.

Finally, as highlighted in our analysis of initialization methods, future research could consider comparing the Integrated Completed Likelihood (ICL) criterion with other criteria, such as the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC). Such comparative studies could provide valuable insights into the most effective approaches for determining the optimal number of clusters in complex graph models.

References

Gérard Govaert Christophe Biernacki, Gilles Celeux. 1998. Assessing a Mixture Model for Clustering with the Integrated Classification Likelihood. *INRIA intern* (1998).

Franck Picard Daudin, J-J. and Stéphane Robin. 2008. A mixture model for random graphs. In *Statistics and Computing*. 173–183.

P Erdös and A Rényi. 1959. On random graphs I. *Publ. math. debrecen* 6, 290-297 (1959), 18.

Christian G Fink, Kelly Fullin, Guillermo Gutierrez, Nathan Omodt, Sydney Zinnecker, Gina Sprint, and Sean McCulloch. 2023. A centrality measure for quantifying spread on weighted, directed networks. *arXiv preprint arXiv:2303.09684* (2023).

Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

Mark S Handcock and James Holland Jones. 2004. Likelihood-based inference for stochastic models of sexual network formation. *Theoretical population biology* 65, 4 (2004), 413–422.

Jaakola. 2000. Advanced mean field methods: theory and practice. *Neurocomputing* (2000).

Boleslaw K. Szymanski Jierui Xie, Stephen Kelley. 2013. Overlapping Community Detection in Networks, the State of the Art and Comparative Study. *ACM Computing Surveys, vol. 45, no. 4* (2013).

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. 1999. An introduction to variational methods for graphical models. *Machine learning* 37 (1999), 183–233.

P. Latouche and P.A. Mattei. 2023. Introduction to Probabilistic Graphical Models and Deep Generative Models. *Master MVA course* (2023).

Satu Elisa Schaeffer. 2007. Graph clustering. *Computer science review* 1, 1 (2007), 27–64.

Michael PH Stumpf, Carsten Wiuf, and Robert M May. 2005. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences* 102, 12 (2005), 4221–4224.

Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440–442.

S. White and P Smyth. 2005. A spectral clustering approach to finding communities in graphs. *Proc. SIAM International Conference on Data Mining* (2005).

6 Appendix

6.1 Theoretical proofs

Proposition : Given the label of a vertex, the conditional distribution of the degree of this vertex is Binomial (approximately Poisson):

$$deg(\text{node}_i)|\{i \in q\} \sim B(n-1, \pi_q) \approx P(\lambda_q),$$

where

$$\pi_q = \sum_\ell \alpha_\ell \pi_{q\ell} \quad \text{and} \quad \lambda_q = (n-1)\pi_q.$$

Proof Conditionally to the belonging of vertices to classes, edges connecting vertex i belonging to class q are independent. The conditional connection probability is:

$$\begin{aligned} Pr\{X_{ij} = 1 | i \in q\} &= \sum_\ell Pr\{X_{ij} = 1 | i \in q, j \in \ell\} Pr\{j \in \ell\} \\ &= \sum_\ell \alpha_\ell \pi_{q\ell} = \pi_q. \end{aligned}$$

The result second part of the result is based on the following lemma.

Lemma : $\mathcal{B}(n,p) \approx \text{Poisson}(\lambda)$ for n large and p small, that is when n is large and $np = \lambda$

Proof : We want to show that as n approaches infinity and np approaches a finite constant λ, the binomial distribution $\mathcal{B}(n,p)$ approaches the Poisson distribution Poisson(λ).

Let $X \sim \mathcal{B}(n,p)$. The probability mass function (pmf) of X is given by:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, 2, \ldots, n$$

Now, let's compute the limit as n approaches infinity and np approaches λ. Define $\lambda = \lim_{n\to\infty} np_n$. For simplicity we will write p instead of p_n

$$\begin{aligned} \lim_{n\to\infty} P(X = k) &= \lim_{n\to\infty} \binom{n}{k} p^k (1-p)^{n-k} \\ &= \lim_{n\to\infty} \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \\ &= \lim_{n\to\infty} \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} p^k (1-p)^{n-k} \\ &= \lim_{n\to\infty} \frac{n^k}{k!} \left(\frac{(n-1)(n-2)\cdots(n-k+1)}{n^k} \right) p^k (1-p)^{n-k} \end{aligned}$$

Now, let's consider the terms inside the limit:

$$\begin{aligned} \lim_{n\to\infty} \frac{(n-1)(n-2)\cdots(n-k+1)}{n^k} &= \lim_{n\to\infty} \frac{n}{n} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n} \cdots \frac{n-k+1}{n} \\ &= 1 \cdot 1 \cdot 1 \cdots 1 \\ &= 1 \end{aligned}$$

So, the terms involving $(n-1)(n-2)\cdots(n-k+1)/n^k$ all converge to 1.

Now, let's evaluate the limit of the remaining terms:

$$\begin{aligned} \lim_{n\to\infty} \frac{n^k}{k!} p^k (1-p)^{n-k} &= \frac{1}{k!} \lim_{n\to\infty} (n^k) \cdot (p^k) \cdot (1-p)^{n-k} \\ &= \frac{1}{k!} \cdot \lambda^k \cdot e^{-\lambda} \quad (\text{because } \lim_{n\to\infty} np = \lambda) \end{aligned}$$

So, we have shown that:

$$\lim_{n\to\infty} P(X = k) = \frac{1}{k!} \cdot \lambda^k \cdot e^{-\lambda}$$

This is the probability mass function of a Poisson distribution with parameter λ. Therefore, as n approaches infinity and np approaches λ, the binomial distribution $\mathcal{B}(n, p)$ approaches the Poisson distribution Poisson(λ).

Lemma : For a graph with correspond ERMG prior parameters π and α, its clustering coefficient is denoted by :

$$c = \frac{\sum_{q,l,m} \alpha_q \alpha_l \alpha_m \pi_{ql} \pi_{qm} \pi_{lm}}{\sum_{q,l,m} \alpha_q \alpha_l \alpha_m \pi_{ql} \pi_{qm}}$$

Proof : For any triplet (i, j, k), we have

$$\begin{aligned} \Pr\{\triangledown\} &= \sum_{q,l,m} \alpha_q \alpha_l \alpha_m \Pr\{X_{ij} X_{jk} X_{ki} = 1 | i \in q, j \in l, k \in m\}, \\ &= \sum_{q,l,m} \alpha_q \alpha_l \alpha_m \pi_q \pi_l \pi_m \end{aligned}$$

6.2 Results seen in section 3

We propose to show the results that were previously presented in Results 3 larger for any deeper analysis.

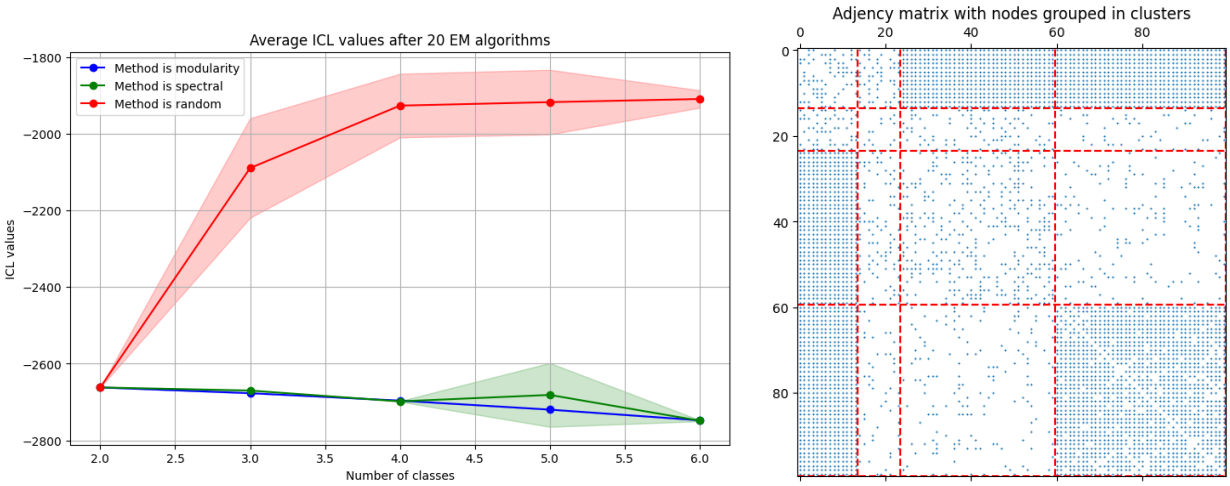


Figure 6. Different methods comparison when two nodes from the same classes are not laikely to be connected

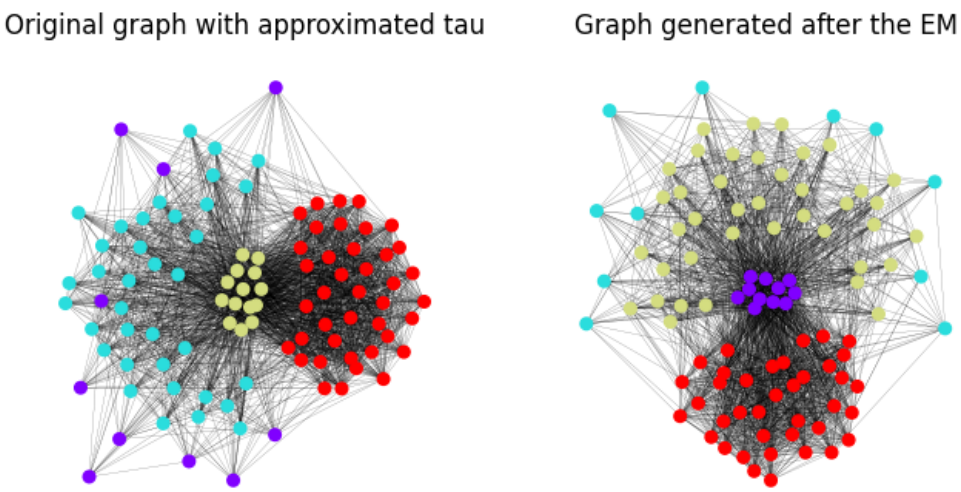


Figure 7. The generated graph in scenario 1 with the clustering found by the algorithm, and the one created from the results obtained

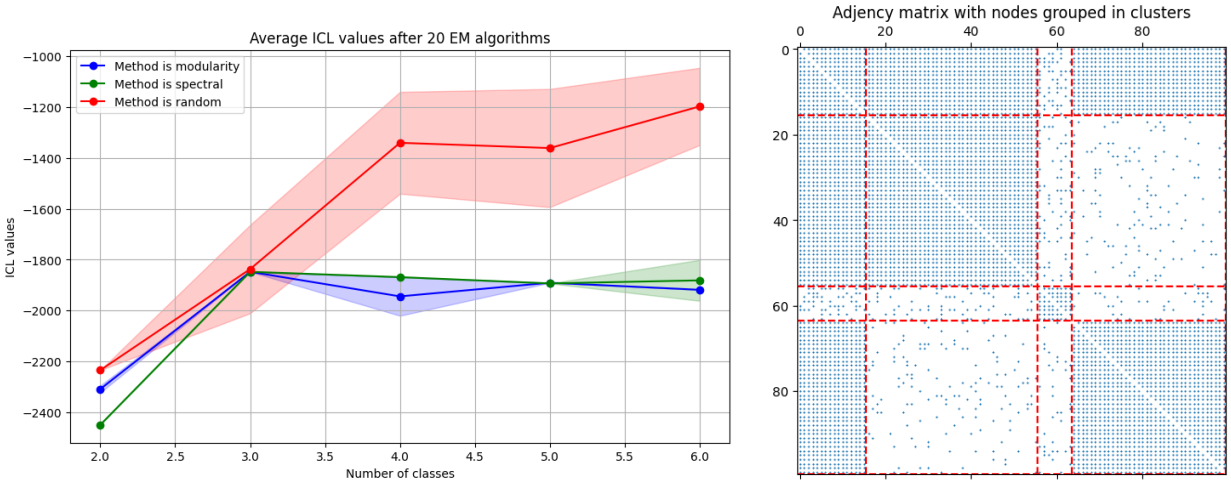


Figure 8. Different methods comparison when there is many edges

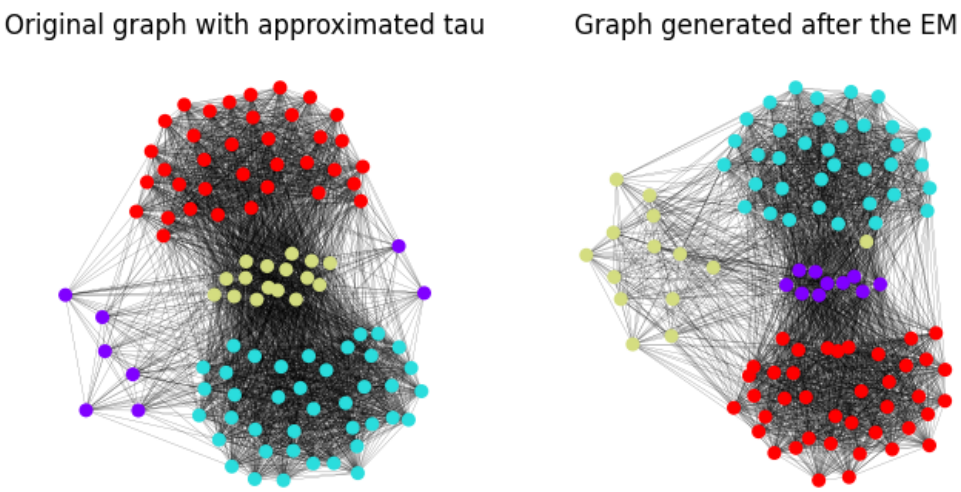


Figure 9. The generated graph in scenario 2 with the clustering found by the algorithm, and the one created from the results obtained

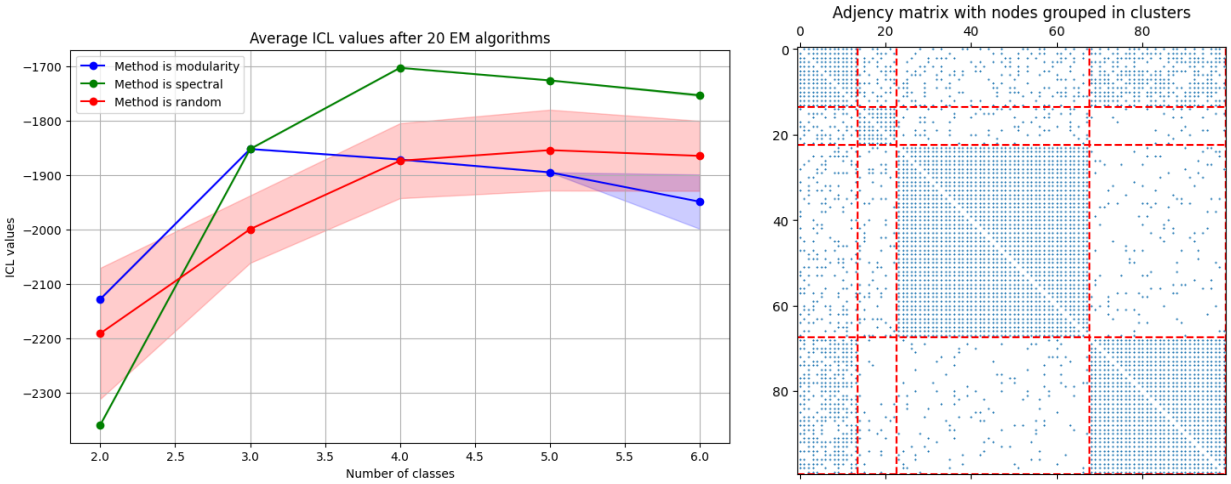


Figure 10. Different methods comparison when the structure is close to a classic community one

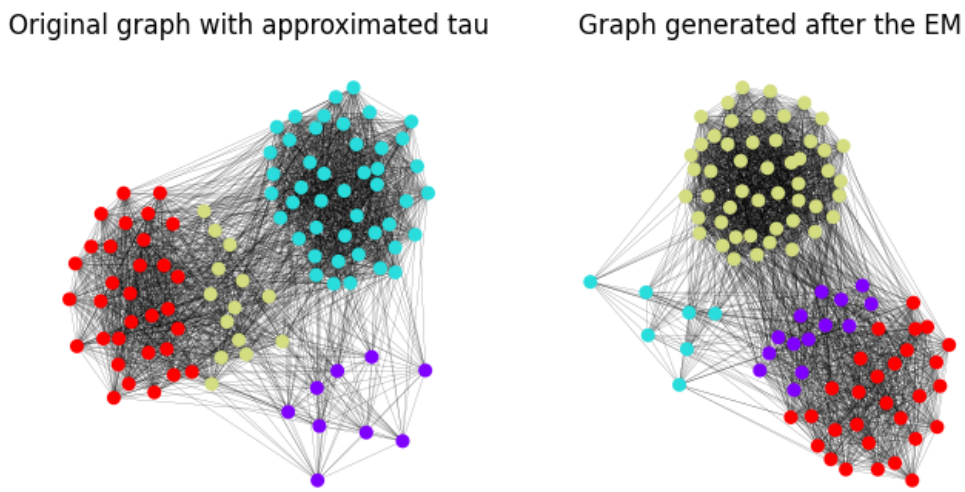


Figure 11. The generated graph in scenario 3 with the clustering found by the algorithm, and the one created from the results obtained

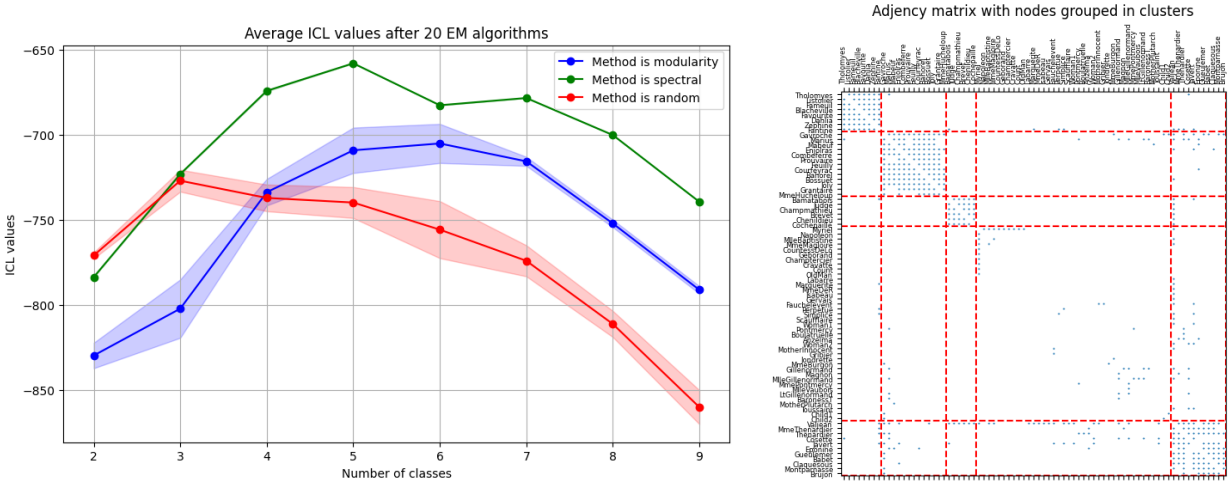
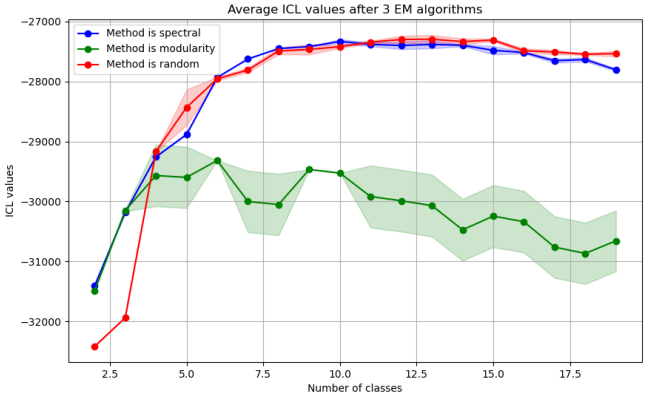
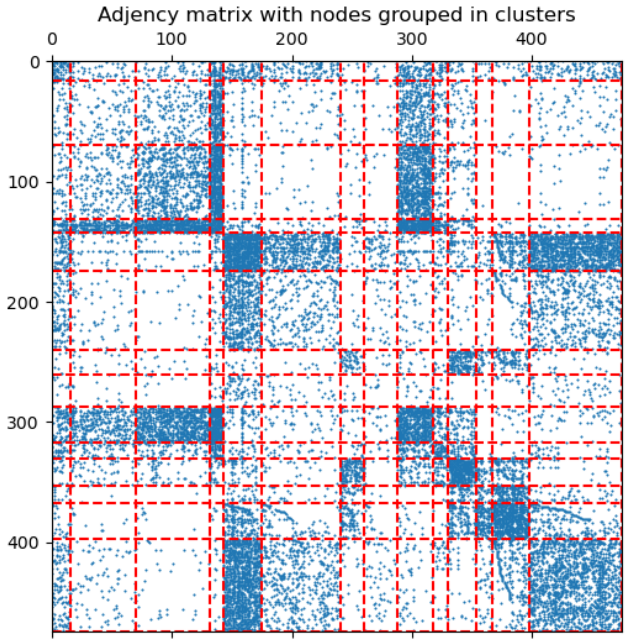


Figure 12. Different methods comparison for the graph of the Miserables

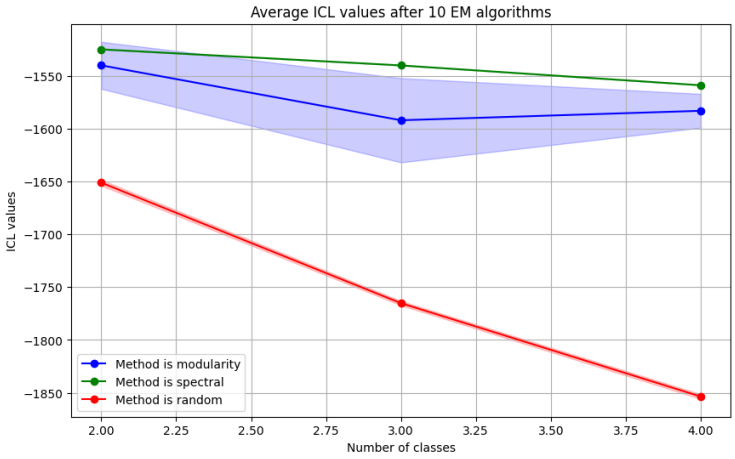


(a) ICL

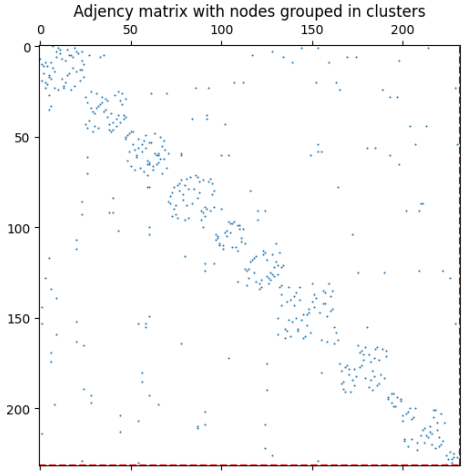


(b) Post-clustering reordered adjacency matrix

Figure 13. Different methods comparison for the graph of the twitter interactions



(a) ICL



(b) Post-clustering reordered adjacency matrix

Figure 14. Different methods comparison for the graph of the Paris Metro Stations. This graph is not likely to follow the ERMG model, and the model failed to propose any relevant structure

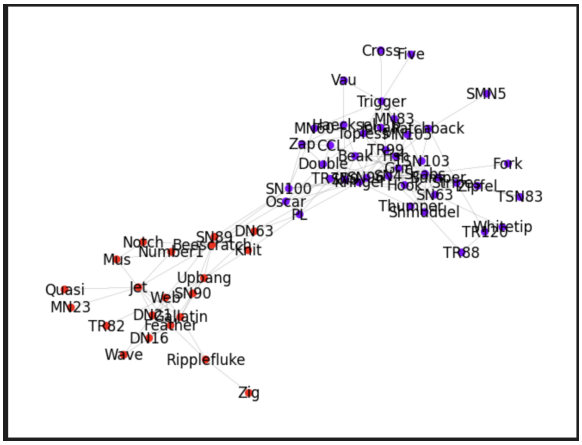


Figure 15. The variational EM finds two clusters for the dolphins' graphs

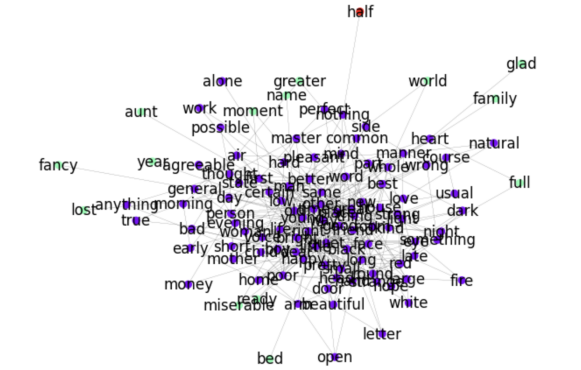


Figure 16. The variational EM finds three clusters for the David Copperfield's graphs

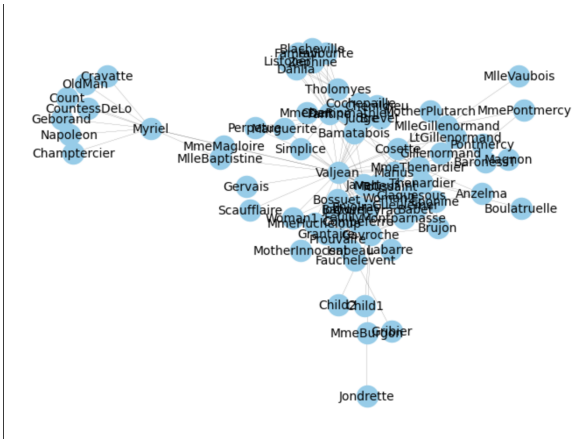


Figure 17. The variational EM finds three clusters for the David Copperfield's graphs