Mixture Models for Graph Clustering

Sofiane Ezzehi[1], **Bastien Le Chenadec**[1], **and Theïlo Terrisse**[1]

[1]École des Ponts ParisTech

Contribution Statement

TODO

1 Introduction

Graph clustering, also referred to as "community detection" [1], is the problem of revealing clusters within a graph, where a "cluster" is often defined as a set of nodes that are more densely connected between them than with the rest of the network. More precisely, a *homophilic* cluster C is defined as a group of nodes with an internal density $\delta_{int}(C)$ larger than its inter-community density $\delta_{ext}(C)$, where

$$\delta_{int} = \frac{\text{\# internal edges of } C}{n_C(n_C-1)/2} \tag{1}$$

and

$$\delta_{ext} = \frac{\text{\# inter-cluster edges of C}}{n_C(n-n_C)} \tag{2}$$

with n the number of nodes of the graph, n_C the number of nodes in the cluster C. Some applications may also be interested in *heterophilic* clusters, defined such that $\delta_{int}(C) < \delta_{ext}(C)$. Graph clustering has tremendous applications in the study of real networks such as social [1] or biological [4]. This problem is of importance to perform a mesoscopic study of graphs, where clusters can be interpreted as meta-nodes that reveal new interactions within a complex system.

Graph clustering is known to be an NP-hard problem, as testing all possible partitions of a graph has complexity $O(B_n)$ with B_n the Bell number of order n. This is particularly poblematic as real networks easily grow to thousands or even millions of nodes. As well documented in [1], several classes of algorithms can be identified in the literature. To name a few, modularity-based approaches solve an optimization problem that maximizes modularity (defined in Section 3.3); hierarchical algorithms rely on similarity measures to aggregate or divise groups of nodes; spectral algorithms also rely on a similarity matrix, but operate in the space spanned by the eigenvectors of its Laplacian matrix. Yet, defining a similiarty measure between nodes is not always easy.

Alternatively to algorithmic methods, model-based methods fit a mathematical model capable of explaining the observed connectivity, and from which descriptive properties of a graph may be extracted or computed. The Stochastic Block Model (SBM) introduced in [2] is a renowned example of model for graphs. In [3], J.-J. Daudin *et al.* study this model and propose a variational Expectation-Maximization (EM) algorithm to find parameters that best fit a given graph. In this report, their method will be referred to as "Daudin-EM".

In this report, we detail the Daudin-EM method as well as a variant from the literature in Section 2. In Section 3, we present our implementation of the method and the experiments performed to evaluate it, with a discussion of the results in Section 4. Lastly, we conclude in Section 5.

2 Proposed methods and variants

2.1 A mixture model for random graphs [3]

In "A mixture model for random graphs" [3], Daudin *et. al.* propose a bayesian model for graphs, and an EM approach to fit this model to a given graph.

We will follow the same notations as in [3]. We consider an undirected graph with n nodes and no self-loops. We denote X the adjacency matrix of this graph. As such, $X_{ij} \in \{0,1\}$ denotes the existence of an edge between nodes i and j, and $\forall i \in \{1,\ldots,n\}, X_{ii} = 0$.

As explained by the authors, the SBM draws inspiration both from mixture models for distriubtion of degrees, which have the disadvantage of not dealing with the probability of two given nodes of being connected, and from the foundational Erdös-Rényi model, which is known to fit real graphs poorly. To define the SBM, we consider a mixture model that spreads the vertices in Q classes, with an a priori repartition $\{\alpha_1,\ldots,\alpha_Q\}$ between classes. We introduce the random variables $Z_{iq} \in \{0,1\}$ for $i \in \{1,\ldots,n\}$ and $q \in \{1,\ldots,Q\}$, that represent the membership of node i to class q. We have the following prior distribution on Z:

$$\forall i \in \{1,\ldots,n\}, \quad \sum_{q=1}^{Q} Z_{iq} = 1 \tag{3}$$

and

$$\forall q \in \{1,\ldots,Q\}, \quad \mathbb{P}(Z_{iq}=1) = \alpha_q. \tag{4}$$

We introduce priors on the existence of edges between nodes of different classes. We denote π_{ql} the probability of an edge between a node of class q and a node of class l. Because the graph is undirected, we have $\pi_{ql} = \pi_{lq}$. Finally we suppose the following conditional distribution on the existence of edges between nodes :

$$\forall q,l \in \{1,\ldots,Q\}\,,\quad \forall i \neq j \in \{1,\ldots,n\}\,,$$
$$\mathbb{P}(X_{ij} = 1 \,|\, Z_{iq} = 1, Z_{jl} = 1) = \pi_{ql}. \tag{5}$$

Figure 1 shows the graphical model of this mixture model.

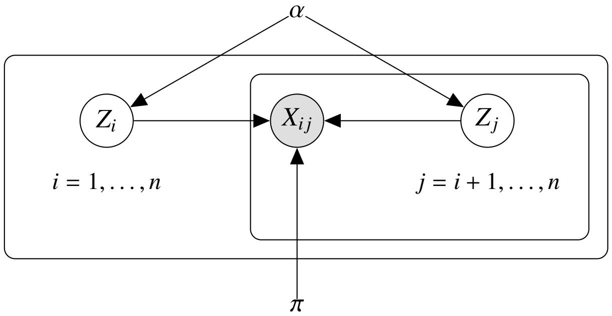


Figure 1: Graphical model of the mixture model

2.1.1 Variational Expectation-Maximization algorithm

The log-likelihood of the model is given by :

$$\begin{aligned}\log \mathcal{L}(X,Z) &= \sum_i \sum_q Z_{iq} \log \alpha_q \\ &+ \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{iq} Z_{jl}\, \pi_{ql}^{X_{ij}} (1-\pi_{ql})^{1-X_{ij}}\end{aligned} \tag{6}$$

Because the likelihood $\mathcal{L}(X)$ is not tractable, the authors propose to use an EM algorithm to fit the model. However the E-step is not tractable either because of the posterior distribution of Z given X. Instead the authors propose to optimize (7) which is a lower bound of $\log \mathcal{L}(X)$ obtained using the Kullback-Leibler divergence between the posterior distribution of Z given X and an approximated distribution R_X.

$$\mathcal{J}(R_X) = \log \mathcal{L}(X) - \mathrm{KL}[R_X(\cdot), P(\cdot|X)] \tag{7}$$

By choosing the approximated distribution R_X to be a product of independant multinomial distributions (8), the authors obtain a fixed point relation (9) between the parameters of the model and the parameters of the approximated distribution maximizing the lower bound $\mathcal{J}(R_X)$. This fixed point relation is used in the E-step of the algorithm.

$$\begin{aligned}R_X(Z;\tau) &= \prod_{i=1}^{n} h(Z_i, \tau_i) \\ \forall i \in \{1,\ldots,n\}\,,\quad h(Z_i,\tau_i) &= \prod_{q=1}^{Q} \tau_{iq}^{Z_{iq}}.\end{aligned} \tag{8}$$

Then, $\forall i \in \{1,\ldots,n\}\,, \forall q \in \{1,\ldots,Q\}\,,$

$$\hat{\tau}_{iq} \propto \alpha_q \prod_{j \neq i} \prod_l \left[\pi_{ql}^{X_{ij}} (1-\pi_{ql})^{1-X_{ij}}\right]^{\hat{\tau}_{jl}} \tag{9}$$

This fixed point relation does not assure the theoretical convergence of the algorithm. We will see later that the convergence of the algorithm is not guaranteed in practice either. Finally we have the following updates for the M-step maximizing $\mathcal{J}(R_X)$:

$$\forall q,l \in \{1,\ldots,Q\}\,,\quad \hat{\alpha}_q = \frac{1}{n} \sum_i \hat{\tau}_{iq} \tag{10}$$

and

$$\hat{\pi}_{ql} = \frac{\sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{jl} X_{ij}}{\sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{jl}}. \tag{11}$$

2.1.2 Selection of Q

[ICL]

2.2 An alternative mixture-model by Newman et. al. [4]

A different mixture model has been proposed the same year as Daudin-EM, by Newman *et al.* [4]. The main difference lies in the way a cluster is defined. In the previous model, the probability of connection between a node in cluster A and a node in cluster B is the same, regardless of the considered nodes.

In the Newman model, we build directed edges between nodes. In the undirected case, we simply consider that two nodes are connected if there are directed edges between them in both directions. By construction, 2 nodes belong to the same cluster if the probability of a directed edge between either of them and any other node in the graph is the same. Therefore, for any given pair of nodes (i, j) belonging to the same cluster C and any other node k, we have,

$$\mathbb{P}(i \to k) = \mathbb{P}(j \to k) = \Pi_{C,k}. \tag{12}$$

Therefore, we see that matrix Π, which is different from the π of the previous model, describes the probability of a directed edge existing between a cluster and a node. Furthermore, contrary to the previous model, the probability of a directed edge existing between a node in cluster A and a node in cluster B is not necessarily invariant with respect to the considered nodes.

2.2.1 Model

As in the previous model, we denote X the adjacency matrix of the input graph which is assumed to be undirected and containing n nodes. We also denote $(\alpha_1,\ldots,\alpha_Q)$ the probabilities for a node of belonging to a cluster $q \in \{1,...,Q\}$. We introduce the random variables $g_{i \in \{1,\ldots,n\}} \in \{1,...,Q\}$ which represent the cluster of node i; these are analogous to the random variables Z_{iq} previously introduced. Therefore,

$$\forall i \in \{1,...,n\},\quad \mathbb{P}(g_i = q) = \alpha_q. \tag{13}$$

Finally, we introduce the matrix $\Pi \in [0,1]^{Q \times n}$ defined earlier. Since the input graph is undirected, the probability of an undirected edge existing between two nodes i and j is,

$$\mathbb{P}(X_{ij} = 1) = \mathbb{P}(i \to j)\mathbb{P}(j \to i) = \Pi_{g_i,j}\Pi_{g_j,i}. \tag{14}$$

Two normalization conditions are imposed,

$$\sum_{q=1}^{Q} \alpha_q = 1 \quad \text{and} \quad \forall q \in \{1,...,Q\}, \sum_{i=1}^{n} \Pi_{q,i} = 1 \tag{15}$$

2.2.2 Expectation-Maximization algorithm

The expected log-likelihood (with respect to the random variables g) is,

$$\mathcal{L}(X) = \sum_{i=1}^{n} \sum_{q=1}^{Q} \tau_{iq} \left[\log(\alpha_q) + \sum_{j=1}^{n} X_{ij} \log(\Pi_{q,j}) \right] \quad (16)$$

where $\tau_{iq} = \mathbb{P}(g_i = q | X, \alpha, \Pi)$.
Remark : The authors of the article made the surprising choice of not taking into account, in the likelihood of X given the parameters, the probability of the non-existing edges. Taking them into account would have added in the last sum of the log-likelihood a term of the form $(1 - X_{ij}) \log(1 - \Pi_{q,j})$.

The main advantage of this model is that the expectation step is much simpler than in the previous model as it doesn't require to solve a fixed-point problem. Indeed, we have an explicit expression for τ_{iq},

$$\tau_{iq} = \frac{\alpha_q \prod_{j=1}^{N} \Pi_{qj}^{X_{ij}}}{\sum_{s=1}^{Q} \alpha_s \prod_{j=1}^{N} \Pi_{sj}^{X_{ij}}}.$$

The M-step updates are,

$$\alpha_q = \frac{1}{n} \sum_{i=1}^{n} \tau_{iq} \quad \text{and} \quad \Pi_{qj} = \frac{\sum_{i=1}^{n} X_{ij} \tau_{iq}}{\sum_{i=1}^{n} k_i \tau_{iq}}.$$

where $k_i = \sum_{j=1}^{n} X_{ij}$ is the degree of node i. We verify that the normalization conditions are satisfied.

2.3 Spectral clustering

A third method that we will use as a comparison baseline is spectral clustering, first introduced in [5]. The idea is to use the eigenvectors of the Laplacian matrix of the graph to cluster the nodes. The Laplacian matrix is defined as $L = D - X$ where X is the adjacency matrix of the graph and D is the diagonal matrix of degrees of the nodes.

The algorithm is simple: stacking the k first eigenvectors of L in a matrix $U \in \mathbb{R}^{n \times k}$, we perform a k-means clustering on the rows of U.

3 Experiments

The experiments we have carried out can be separated into three parts, all based on a re-implementation the methods covered in Section 2. A first experiment illustrates the lack of robustness of the fixed-point algorithm used in the E-step of the Daudin-EM algorithm. Second, Daudin-EM is tested isolately on synthetic graphs generated from the SBM introduced in Section 2.1. This set of experiments aims at studying the capacity of the proposed variational EM algorithm to recover parameters for given graphs, but also the efficiency of the model in terms of graph clustering. Lastly, the performance of the mixture model is evaluated on classical real datasets, namely Zachary's Karate club and the Cora dataset. This performance is compared to the baselines presented in Sections 2.2 and 2.3.

3.1 Implementation

All of the experiments were performed using the code attached with this report. The Daudin-EM method (simply referred to as "SBM" in the code) and its variant by Newman Sofiane: n'hésite pas à adapter are both implemented in 2 versions: using Numpy and using PyTorch. Two additional PyTorch versions were added for Daudin-EM, one to handle memory issues occurring for large graphs and the other that takes the log of products encountered in the computations to turn them into sums and speed up the process Bastien: hésite pas à adapter au besoin. As an indication, a speed comparison Est-elle à jour ? between the versions of the mixture model is available in appendix A. The spectral clustering algorithm was implemented in Numpy in a dedicated notebook. Lastly, the algorithm proposed by Lancichinetti, Fortunato, and Kertesz in [6] to deal with overlapping communities was implemented in Numpy, but was not exploited in experiments for conciseness of this report.

The implementation also comes with a battery of tests to validate the implementations, as well as scripts and notebooks to reproduce and visualize the experiments. All experiences used Pytorch implementations when available, especially the `PytorchLogImplementation` for Daudin-EM, except for the smallest graphs of the second set of experiments that used the Numpy version as it was faster in this case .

3.2 Robustness of the fixed-point algorithm

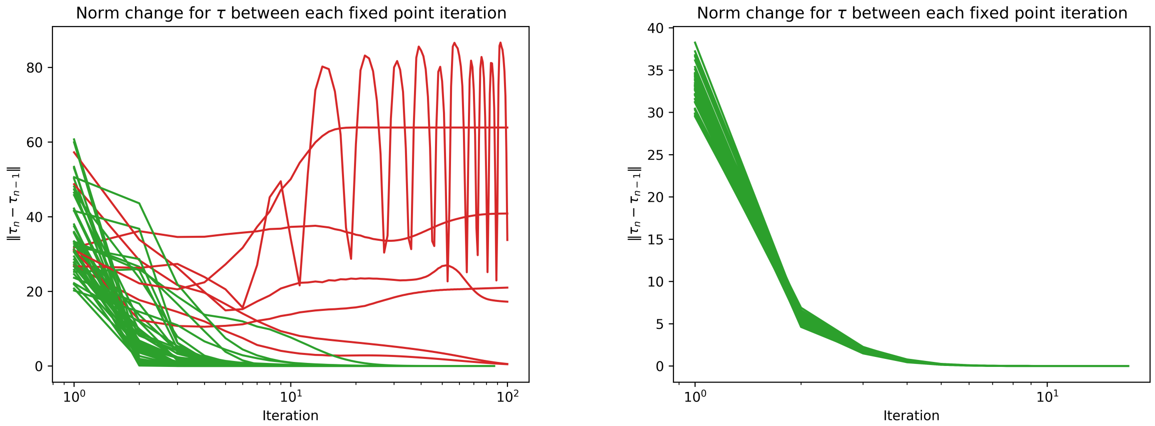
Early stages of the implementation of the method revealed a lack of robustness of the fixed-point algorithm used in the E-step of the method. In particular, the fixed-point algorithm was found to be sensitive to the initialization of the parameters α and π, and does not converge in all cases.

To illustrate this, we perform 2 experiments:

1. In the first experiment, we initialize 50 random graphs and fix a random $\hat{\tau}$ to get 50 initial states.
2. In the second experiment, we fix a random graph and initialize 50 random $\hat{\tau}$ to get 50 initial states.

In each experiment, the graphs are generated using the SBM described in Section 2.1, with $Q = 3$ and 100 nodes. More precisely, α and π are initialized randomly, then the latent variable Z is sampled from (4), and finally the adjacency matrix X is sampled from (5). Random $\hat{\tau}$s are sampled as a normalized uniform variable. In each experiment, 100 fixed-point algorithm iterations are run. These fixed-point iterations are run in parallel for the 50 initial states, resulting in figures 2a and 2b which plot the norm of the difference of $\hat{\tau}$ between two successive iterations as a function of the iteration, for all graphs, and for the two experiences.

Code for this experiment is available in file `fixed_point_convergence.py`.



(a) Fixed τ, variable α, π and τ.

(b) Variable τ, fixed α, π and τ.

Figure 2: Convergence of the fixed point algorithm. Green curves converged within 100 iterations; red curves did not.

3.3 Metrics

The literature offers various metrics to assess the quality of a clustering. For our experiments, we re-implemented some supervised and unsupervised metrics.

Supvervised metrics By "supervised metrics", we designate metrics that compare the labelling of nodes C^P as predicted by a clustering method, with a ground truth labelling C^T that is supposed to be given. We used two classical ones, the Rand index (RI) and the normalized mutal information (NMI).

RI measures the similarity of two clusterings by measuring how consistently the two clusterings identify edges as connecting nodes of a same cluster, or of different clusters:

$$\mathrm{RI} = \frac{a+b}{m} \tag{17}$$

with a the number of pair of nodes that are in a same clsuter in both clusterings, b the number of pairs of nodes that are in different clusters in both clusterings and m the number of pairs of nodes in the graph.

MI is a measure that evaluates the mutual dependence between two variables. For two clusters $C = (C_i)_{1\leq i\leq |C|}$ and $C' = (C'_j)_{1\leq i\leq |C'|}$, it is defined as

$$\mathrm{MI}(C, C') = \sum_{i=1}^{|C|}\sum_{j=1}^{|C'|} P(i,j)\log\left(\frac{P(i,j)}{P(i)P'(j)}\right) \tag{18}$$

with $P(i) = \frac{|C_i|}{n}$, $P'(j) = \frac{|C'_j|}{n}$ and $P(i,j) = \frac{|C_i \cap C'_j|}{n}$.

Both measures are insensitive to permutation of the clusters.

Unsupervised metrics Unsupervised metrics are defined without resorting to ground truth labelling.

The modularity M of a clustering is a common metric that compares the observed connectivity within clusters to its expected value for a random graph with the same degree distribution as the considered graph. It is defined as

$$\mathrm{M}(C) = \sum_{i=1}^{|C|}\left(\frac{e_i}{m} - \left(\frac{d_i}{2m}\right)^2\right) \tag{19}$$

with m the number of edges in the graph, and e_i and d_i respectively the number of edges and total degree of nodes in cluster C_i.

Additionally, we use the clustering coefficient defined as the ratio of all triangles within a set of nodes C, over the total number of triplets of nodes within this set; using the adjacency matrix, this writes:

$$\mathrm{CC}(C) = \frac{\displaystyle\sum_{i,j,k\in C} X_{ij}X_{jk}X_{ki}}{\displaystyle\sum_{i\in C} k_i(k_i-1)} \tag{20}$$

with k_i the degree of node i. We will use this metric both at the scale of clusters (to determine their individual qualities) and of whole graphs to measure their clustering tendency. To close this section on metrics, note that the SBM model defined in Section 2.1 allows a computation of the expected clustering coefficient associated to graphs generated by a given set of parameters (α, π):

$$\mathrm{CC}_{\mathrm{SBM}} = \frac{\displaystyle\sum_{q,l,m\in\{1,\ldots,Q\}} \alpha_q\,\alpha_l\,\alpha_m\,\pi_{ql}\,\pi_{qm}\,\pi_{lm}}{\displaystyle\sum_{q,l,m\in\{1,\ldots,Q\}} \alpha_q\,\alpha_l\,\alpha_m\,\pi_{ql}\,\pi_{qm}}. \tag{21}$$

3.4 Testing on SBM datasets

This second set of experiments aimed at evaluating the Daudin-EM algorithm is 3-fold:

1. assessing the capacity of the algorithm to recover the parameters of the model that generated a given graph;

	Experiment	Hyper-parameters		Parameters	
#	Name	n	Q	α	π
1	Random-small	30	3	$\alpha \sim \mathrm{Dir}(1.5)$	$\pi_{ij} \sim \mathcal{U}([0,1))$
2	Random-large	500	3	$\alpha \sim \mathrm{Dir}(1.5)$	$\pi_{ij} \sim \mathcal{U}([0,1))$
3	Homophilic	150	3	$\alpha_i = (\frac{1}{Q})_i$	$\pi_{ii} = 1 - \varepsilon_1,\ \pi_{ij} = \varepsilon_2$
4	Homophilic-hard	150	5	$\alpha_i = (\frac{1}{Q})_i$	$\pi_{ii} \sim \mathcal{U}([0.5,1)),\ \pi_{ij} = \varepsilon_2$
5	High-degree-minority	150	2	$\alpha^T = (\frac{9}{10}, \frac{1}{10})$	$\pi = \begin{pmatrix} \varepsilon_1 & a \\ \varepsilon_2 & b \end{pmatrix}$
6	Heterophilic	150	3	$\alpha_i = (\frac{1}{Q})_i$	$\pi_{ii} = \varepsilon_2,\ \pi_{ij} = 1 - \varepsilon_1$

Table 1: Parameters used for the generation of the SBM dataset.

2. assessing the performance of the algorithm in terms of graph clustering;
3. experimenting with ICL to select the number of clusters. A voir

Bulding the dataset Using the SBM, we generate 6 batches of 10 graphs to test the method on. Each batch is generated using a specific set of parameters with the intent of modelling a specific type of graphs, as indicated by the name of the associated experience. The parameters of the SBM for each batch are given in Table 1, Dir the Dirichlet law, $\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.01$, $a = 0.7$ and $b = 0.8$. Note that π was made symmetric, normalized and scaled in experiments 1 and 2. Sample graphs from each batch are visualized on Figure 4 in appendix.

Assessing parameter recovering performance Because the graphs are precisely generated using the same model as the one we are trying to fit, these experiments are primarily designed to estimate the performance of the proposed EM algorithm. In particular, a ground truth $(\alpha\mathrm{T}, \pi_{\mathrm{T}})$ is trivially given for the expected best-fitting model. Consequently, we could simply measure the distance between the laws induced by the predicted (α_P, π_P) and the ground truth $(\alpha\mathrm{T}, \pi_{\mathrm{T}})$. However, as explained in [3], the likelihood of an adjacency matrix X given a choice of parameters is intractable, making it hard to apply criteria like the Kullback-Leibler divergence or the χ^2 statistic. We have considered using the relative distance between $\mathcal{J}(\mathcal{R}_{\mathcal{X}}(\cdot;\tau_{\mathcal{P}})$ computed for both sets of parameters, but ditched this metric due to its lack of interpretability across batches. Ultimately, even though this solution is not ideal, [ça vous va ?] we decided to simply sum the Euclidean distances between α_T and α_P and between π_T and π_P, normalized by the number of classes.

Assessing clustering performance To assess the clustering capacity of the algorithm, we use the metrics introduced in Section 3.3. In particular, the true and predicted clusterings are defined as $\forall i \in \{1, ..., Q\}$,

$$C_i^T = \arg\max_q z_{iq} \quad \text{and} \quad C_i^P = \arg\max_q \hat{\tau}_{iq}. \tag{22}$$

with $\hat{\tau}$ the predicted value of τ output by the algorithm. For more intepratbility of modularity, instead of averaging over graphs, we choose the graph that produced largest modularity with the predicted parameters.

Running the experiments each experiment is run for 100 iterations, and each time using 5 random initializations of α and π to limit the effects of the lack of robustness underlined in Section 3.2.

The results of the experiments are presented in Table 2, along with sample predictions on Figure 5 in appendix. Note that, due to the lack of robustness underlined in Section 3.2, Daudin-EM did not converge within the budget of 1000 iterations for all graphs. This is why the number of passed graphs is indicated in the results.

3.5 *Comparison to variants*

In the last set of experiments, the mixture model-based method [3] described in Section 2.1 is tested on real graphs and compared with 2 other methods:

1. The alternative mixture model from [4] described in section 2.2;
2. The spectral clustering method from [5] described in section 2.3.

The results are then visualized and compared in terms of running time and clustering power, using the metrics defined in Section 3.3.

Datasets These methods are all tested on 2 famous datasets, commonly used as benchmarks for community detection algorithms:

1. Zachary's Karate-club dataset: Published in an Anthropology journal in 1977 [7], it describes the friendship between 34 members of a karate club at a US university after an internal dispute that split the social relationships in two factions. The graph is undirected and contains 34 nodes and 78 edges. Figure 6a shows the ground truth of the dataset.
2. The Cora dataset: A citation network of scientific publications [8]. The graph is undirected and contains 2708 nodes(=papers) and 5429 edges(=citations). The nodes are labeled with 7 different classes. Figure

7a shows the dot-plot representation of the adjacency matrix of the ground truth.

While the Karate-club dataset is a small graph with a clear community structure, the Cora dataset is a much larger graph with a more complex structure. Both datasets are constructed from real-world data and observations which makes them interpretable. They also have a ground truth that we can use to evaluate the performance of the algorithms.

Running the experiments For Zachary's karate club, the SBM variational EM algorithm as well as the Newman variant are run for 100 iterations, using 20 parallel initializations of (α, π, τ) to increase probability of converging.

For Cora, the SBM variational EM algorithm as well as the Newman variant is run for 50 iterations, using 10 initializations of (α, π, τ).

In both cases, the spectral clustering algorithm is run once.

Figures 6 and 7 show for both datasets the true clusterings and those proposed by the 3 methods, and tables 3 and 4 show the resulting metrics.

4 Analysis and discussion

TODO

5 Conclusion

TODO

References

[1] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, February 2010. ISSN 03701573. doi: 10.1016/j.physrep.2009.11.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0370157309002841.

[2] Tom A.B. Snijders and Krzysztof Nowicki. Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification*, 14(1):75–100, January 1997. ISSN 1432-1343. doi: 10.1007/s003579900004. URL https://doi.org/10.1007/s003579900004.

[3] J.-J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, June 2008. ISSN 1573-1375. doi: 10.1007/s11222-007-9046-7. URL https://doi.org/10.1007/s11222-007-9046-7.

[4] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23): 9564–9569, June 2007. ISSN 1091-6490. doi: 10.1073/pnas.0610537104. URL http://dx.doi.org/10.1073/pnas.0610537104.

[5] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.

[6] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, March 2009. ISSN 1367-2630. doi: 10.1088/1367-2630/11/3/033015. URL https://iopscience.iop.org/article/10.1088/1367-2630/11/3/033015.

[7] Wayne Zachary. An information flow model for conflict and fission in small groups1. *Journal of anthropological research*, 33, 11 1976. doi: 10.1086/jar.33.4.3629752.

[8] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008. doi: 10.1609/aimag.v29i3.2157. URL https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2157.

Appendix

A Implementations speed comparison

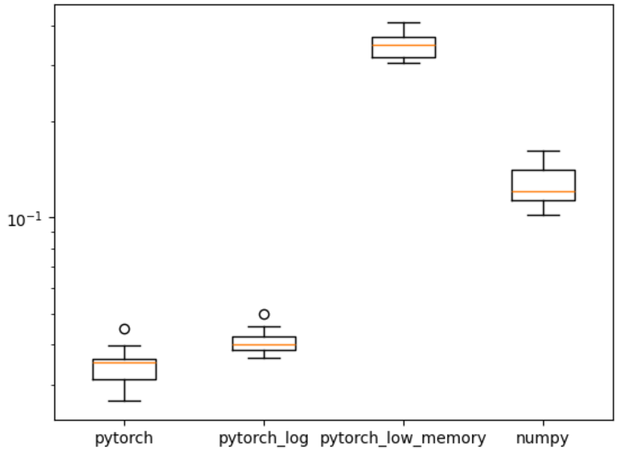


Figure 3: Speed comparison (in seconds) between implementations of the SBM-VEM for the mixture model. Each implementation was run 10 times, for 10 runs on a graph randomly generated using SBM for 100 nodes and 3 classes.

B Second set of experiments

B.1 Samples

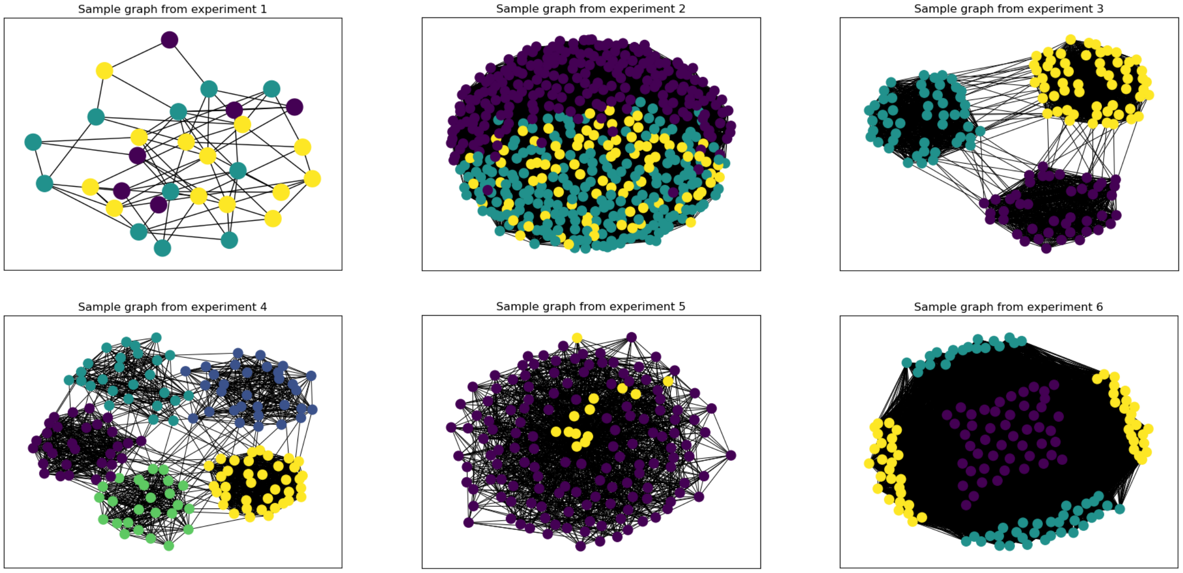

Sample graph from experiment 1
Sample graph from experiment 2
Sample graph from experiment 3
Sample graph from experiment 4
Sample graph from experiment 5
Sample graph from experiment 6

Figure 4: Sample graphs from the second set of experiments

B.2 Results

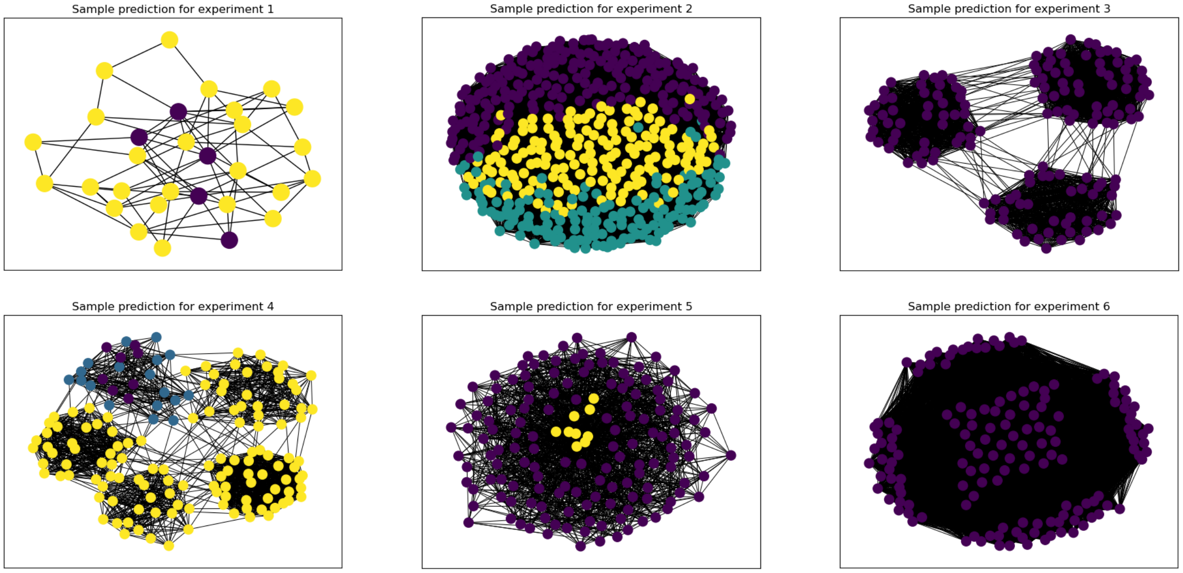

Sample prediction for experiment 1
Sample prediction for experiment 2
Sample prediction for experiment 3
Sample prediction for experiment 4
Sample prediction for experiment 5
Sample prediction for experiment 6

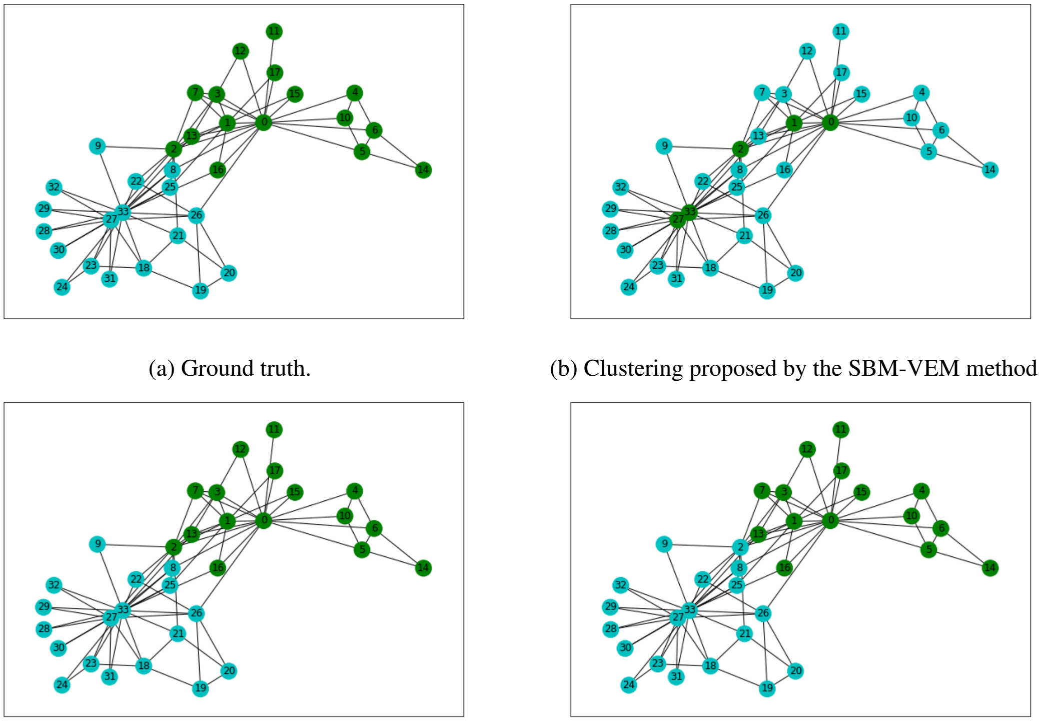
Figure 5: Some predicted graphs from the second set of experiments

Experiment #	Graphs passed	Param. dist.	NMI	RI	G.t. M	Pred. M
1	10	0.16 ± 0.06	0.07 ± 0.07	0.52 ± 0.10	0.08	0.02
2	8	0.16 ± 0.05	0.33 ± 0.13	0.66 ± 0.13	0.07	0.07
3	10	0.22 ± 0.12	0 ± 0	0.33 ± 0.00	0.63	0.00
4	4	0.22 ± 0.02	0.11 ± 0.18	0.27 ± 0.12	0.71	0.15
5	10	0.87 ± 0.01	0.61 ± 0.10	0.93 ± 0.02	0.01	-0.01
6	9	0.18 ± 0.08	0 ± 0	0.33 ± 0.00	-0.33	0.00

Table 2: Results of the second set of experiments.

C Third set of experiments

C.1 *Zachary's Karate club*



(a) Ground truth.

(b) Clustering proposed by the SBM-VEM method.

(c) Clustering proposed by the Newman SBM method.

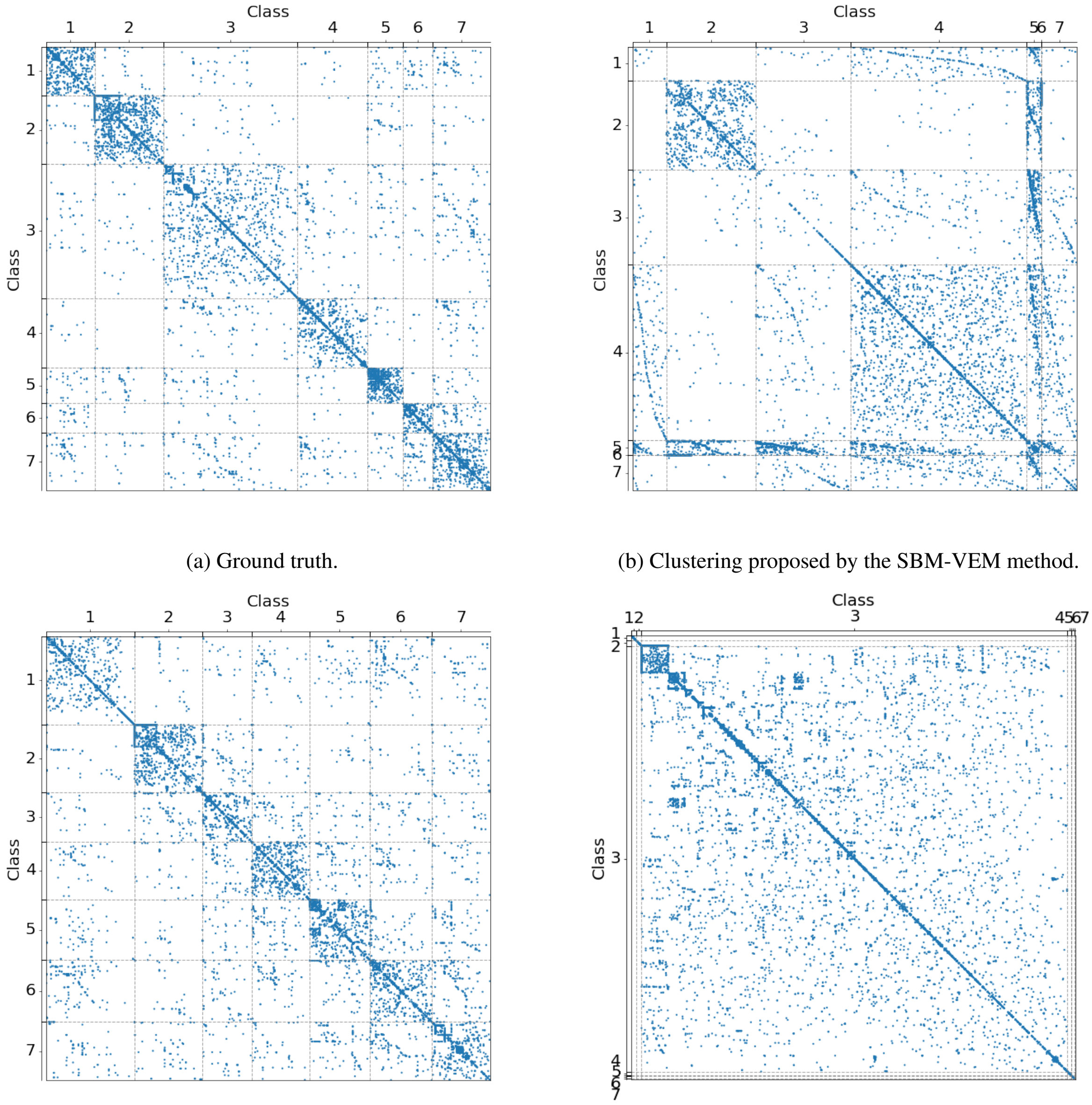
(d) Clustering proposed by the spectral method.

Figure 6: Clustering results for Karate Club.

Method	**Time (s)**	**NMI**	**RI**	**M**	**CC**	**CC (per cluster)**	
						1	2
Ground truth	-	-	-	0.37	0.26	0.42	0.26
Daudin-EM	137	0.01	0.49	-0.21	-	0.50	0.23
Newman-EM	2	1.00	1.00	0.37	-	0.41	0.26
Spectral	0	0.84	0.94	0.36	-	0.35	0.24

Table 3: Results on Zachary's Karate Club

C.2 Cora



(a) Ground truth.

(b) Clustering proposed by the SBM-VEM method.

(c) Clustering proposed by the Newman SBM method.

(d) Clustering proposed by the spectral method.

Figure 7: Clustering results for Cora.

Method	Time (s)	NMI	RI	M	CC	CC (per cluster)						
						0	1	2	3	4	5	6
Ground truth	-	-	-	0.64	0.09	0.19	0.06	0.12	0.23	0.10	0.22	0.16
Duadin-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	0.02	0.23	0.03	-	0.6	0.09	0	0	0	0	0

Table 4: Results on Cora.