Clustering multilayer graphs with missing nodes

Guillaume Braun
guillaume.braun@inria.fr
Inria, Université de Lille, CNRS

Hemant Tyagi
hemant.tyagi@inria.fr
Inria, Université de Lille, CNRS

Christophe Biernacki
christophe.biernacki@inria.fr
Inria, Université de Lille, CNRS

Abstract

Relationships between agents can be conveniently represented by graphs. When these relationships have different modalities, they are better modelled by multilayer graphs in which each layer is associated with one modality. Such graphs arise naturally in many contexts including biological and social networks. Clustering is a fundamental problem in network analysis where the goal is to regroup nodes with similar connectivity profiles. In the past decade, various clustering methods have been extended from the unilayer setting to multilayer graphs in order to incorporate the information provided by each layer. While most existing works assume – rather restrictively - that all layers share the same set of nodes, we propose a new framework that allows for layers to be defined on different sets of nodes. In particular, the nodes not recorded in a layer are treated as missing. Within this paradigm, we investigate several generalizations of well-known clustering methods in the complete setting to the incomplete one and prove some consistency results under the Multi-Layer Stochastic Block Model assumption. Our theoretical results are complemented by thorough numerical comparisons between our proposed algorithms on synthetic data, and also on real datasets, thus highlighting the promising behaviour of our methods in various settings.

1 Introduction

Graphs are a powerful tool to represent relationships between agents. Due to applications in a wide array of fields including biology, sociology, ecology and economics (see for e.g., Braun et al. (2015); Han et al. (2015); Kivelä et al. (2014); Kim and Lee (2015)), the analysis of networks has received significant interest over the last two decades. One fundamental problem of network analysis is *clustering* which involves detecting communities by regrouping nodes having similar connectivity properties. Numerous clustering algorithms have been developed over the years based on different approaches such as modularity maximization, maximum likelihood, random walks, semi-definite programming and spectral clustering (see for instance the survey articles by Fortunato (2009) and Abbe (2018)).

Often, relationships are better understood through different modalities. These multiple aspects of relationships can be represented by a multilayer graph where each layer is a graph representing the interactions between agents for one modality. For e.g., social interaction between a set of people can be recorded via email exchanges, phone calls, professional links, and so on. Each level of interaction can be encoded into a simple graph and the collection of these graphs leads to a multilayer representation. Another important example of a multilayer graph is given by a time-varying network where each view of the network at a given time corresponds to a different layer.

Over the last decade, many methods have been proposed for clustering multilayer graphs such as those based on matrix factorization, spectral methods, maximisation of a modularity function or probability model-based approaches; see Kim and Lee (2015) for a survey. Consistency results for the recovery of the partition under a stochastic generative model have also been shown for some algorithms, see for example Paul and Chen (2020), Pensky and Zhang (2019), Lei (2020) and Bhattacharyya and Chatterjee (2018).

Most existing approaches assume that all the layers share the same set of nodes. In practice, however, data are often incomplete; in particular, the set of observed nodes can clearly vary across layers. For example, in social networks evolving over time, the set of nodes can change due to people leaving/joining the network.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

This is the setting considered in the present paper.

1.1 Related work

Clustering on multi-layer graphs. As noted by Paul and Chen (2020), clustering strategies for multilayer graphs can be roughly categorized into three groups: **early fusion** methods where all views are aggregated and then clustering is performed, **intermediate fusion** methods where the algorithm finds a factor common to all the views, and **final aggregation** methods where each individual view is processed separately and a consensus partition is formed. In the complete setting, different algorithms have been proven to be consistent under a multilayer stochastic block model assumption (see Section 2.2). Among them are spectral clustering on the sum of adjacency matrices (e.g., Bhattacharyya and Chatterjee (2018); Paul and Chen (2020)) or on the sum of squared adjacency matrices with bias correction (e.g., Lei (2020); Bhattacharyya and Chatterjee (2020)), orthogonal linked matrix factorization (e.g., Paul and Chen (2020)), and co-regularized spectral clustering (e.g., Paul and Chen (2020)). Existing misclustering bounds for these methods are gathered in the supplementary material.

Incomplete Multi-View Clustering (IMVC). Recently a similar problem has been addressed in the context of IMVC, see for example Liu et al. (2020), Hu and Chen (2019) and references therein. To the best of our knowledge, no consistency results for the recovery of the ground truth clustering structure are shown in this line of work. Algorithms designed for the IMVC framework cannot be directly applied to our setting since they apply to a collection of feature vectors. However they could possibly be adapted, in a non trivial manner, to our framework. For example, in the complete setting, the OMVC method proposed by Hu and Chen (2019) can be considered as a variant of the OLMF estimator proposed by Paul and Chen (2020) where the optimization problem is modified in order to take into account the symmetry of the inputs. Similarly, if there were no missing views, the algorithm proposed by Liu et al. (2020) resembles a variant of the co-regularized spectral clustering method of Paul and Chen (2020) for clustering multilayer graphs. We leave the adaptation of the algorithm proposed by Liu et al. (2020) to our setting for future work.

1.2 Contributions

We consider the problem of clustering multilayer graphs with missing nodes under a Multi-Layer Stochastic Block Model (MLSBM) described in Section 2. Our contributions are as follows.

- In Section 3.1 we propose a final aggregation method based on a variant of k-means for incomplete data (Algorithm 1), and derive a bound for the misclustering rate.

- Section 4 extends a popular early fusion method – based on spectral clustering applied to the sum of adjacency matrices – to the missing nodes setting. Section 4.1 studies this by imputing the missing entries with zeros (Algorithm 2), and contains an upper bound for the misclutering rate. Section 4.2 proposes an alternative method (Algorithm 3) wherein the missing entries are imputed iteratively. This method is shown to perform well in our experiments.

- Section 5.2 proposes an extension of an intermediate fusion method – namely the Orthogonal Linked Matrix Factorization (OLMF) method studied by Paul and Chen (2020) – to the missing nodes setting.

- In Section 6 we empirically evaluate our algorithms on synthetic data, and also on real datasets.

1.3 Notations

The set of integers $\{1,\ldots,n\}$ will be denoted by $[n]$. For a matrix $M \in \mathbb{R}^{n\times n}$, its Frobenius (resp. operator) norm is denoted by $||M||_F$ (resp. $||M||$). The notation M_{i*} (resp. M_{*j}) denotes the i-th row (resp. j-th column) of M. For any subset J of $[n]$ and symmetric matrix $M \in \mathbb{R}^{n\times n}$, $M_J \in \mathbb{R}^{|J|\times|J|}$ denotes the square submatrix of M obtained by deleting rows and columns whose index doesn't belong to J. For a non symmetric matrix $Z \in \mathbb{R}^{n\times K}$, Z_J denotes the submatrix of Z obtained by deleting rows whose index doesn't belong to J. Sometimes, it will also be convenient to consider A_J (resp. Z_J) as a $n\times n$ (resp. $n\times K$) matrix where the rows and columns (resp. only the rows) whose index doesn't belong to J are filled with zeros; this will be clear from the context. I_n denotes the identity matrix of size n. Constants will be denoted by the letters c and C, eventually indexed by a number to avoid confusion. Within proofs the values of constants can change from line to line whereas they are denoted with the same letter for simplicity.

2 Problem setup

A multilayer graph is a sequence of graphs $\mathcal{G} = (\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(L)})$. If all the graphs are defined on the same set of nodes $\mathcal{N}$ indexed by $[n]$, then $\mathcal{G}$ is said to be pillar. Throughout, we will assume that for all

$l \leq L$ each graph $\mathcal{G}^{(l)}$ is undirected and has no self-loop. This implies that its associated adjacency matrix $A^{(l)} \in \{0,1\}^{n \times n}$ is symmetric with $A^{(l)}_{ii} = 0$ for all i.

Given $\mathcal{G}$ as input, our goal is to recover a partition of $\mathcal{N}$ into K disjoint sets (or communities), so that nodes belonging to the same community share a similar connectivity profile. To make the setup more precise, we will study this problem in the setting where $\mathcal{G}$ is generated via an underlying (unknown) stochastic model, with a latent community structure. This model is a common extension of the well-studied stochastic block model (SBM) for the unilayer case which we now describe.

2.1 Stochastic Block Model (SBM)

The stochastic block model (SBM) – first proposed in Holland et al. (1983) – is a simple yet popular stochastic generative model for unilayer graphs which captures the community structures of networks often observed in the real world. A SBM with the set of nodes $\mathcal{N}$ and K communities $\mathcal{C}_1, \dots, \mathcal{C}_K$ forming a partition of $\mathcal{N}$ is parameterized as follows.

- There is a membership matrix $Z \in \mathcal{M}_{n,K}$ where $\mathcal{M}_{n,K}$ denotes the class of membership matrices. Here, $Z_{ik} = 1$ if node i belongs to $\mathcal{C}_k$, 0 otherwise. Each membership matrix Z can be associated bijectively with a function $z : [n] \to [K]$ such that $z(i) = k$ where k is the unique column index satisfying $Z_{ik} = 1$.
- There is a full-rank, symmetric, connectivity matrix of probabilities
$$\Pi = (\pi_{kk'})_{k,k' \in [K]} \in [0,1]^{K \times K}.$$

Let us denote $P = (p_{ij})_{i,j \in [n]} := Z \Pi Z^T$. A graph $\mathcal{G}$ is distributed according to a stochastic block model $\text{SBM}(Z, \Pi)$ if the corresponding symmetric adjacency matrix A has zero diagonal entries and

$$A_{ij} \overset{\text{ind.}}{\sim} \mathcal{B}(p_{ij}), \quad 1 \leq i < j \leq n,$$

where $\mathcal{B}(p)$ denotes a Bernoulli distribution with parameter p. Hence the probability that two nodes are connected depends only on the community memberships of these two nodes.

Let us denote by n_k the size of the community $\mathcal{C}_k$, n_{min} (resp. n_{max}) to be the size of the smallest (resp. largest) community, and $\beta = \frac{n_{max}}{n_{min}}$. The communities are said to be balanced if they all have the same size (equivalently, $\beta = 1$). The communities are approximately balanced if $\beta = O(1)$. The maximum value of the connectivity parameter is denoted by $p_{max} := \max_{i,j} p_{ij}$ and can be interpreted as the sparsity level (depending on n).

The misclustering rate associated to an estimated membership matrix $\hat{Z}$ is measured by

$$r(\hat{Z}, Z) = r(\hat{z}, z) = \frac{1}{n} \min_{\sigma \in \mathfrak{S}} \sum_i \mathbf{1}_{\{\hat{z}(i) \neq \sigma(z(i))\}},$$

where $\mathfrak{S}$ denotes the set of permutations on $[K]$. A clustering algorithm is said to be strongly consistent – or achieving exact recovery – if $r(\hat{Z}, Z) = 0$ with probability $1 - o(1)$ as n tends to infinity. It is said to be weakly consistent – or achieving almost exact recovery – if $\mathbb{P}(r(\hat{Z}, Z) = o(1)) = 1 - o(1)$ as n tends to infinity. A more complete overview of the different types of consistency and the sparsity regimes where they occur can be found in Abbe (2018).

2.2 Multilayer Stochastic Block Model (MLSBM)

We now describe the multilayer stochastic block model (MLSBM), which is a common extension of the SBM to the setting of multilayer graphs (see for e.g., Paul and Chen (2020); Bhattacharyya and Chatterjee (2018); Lei et al. (2019)). The MLSBM is parametrized by the number of layers L, a common block membership matrix $Z \in \mathcal{M}_{n,K}$, and connectivity matrices $\Pi^{(1)}, \dots, \Pi^{(L)} \in [0,1]^{K \times K}$.

Similar to the unilayer case, let us denote $P^{(l)} = Z \Pi^{(l)} Z^T$ for $l = 1, \dots, L$. A multilayer graph $\mathcal{G}$ is distributed according to the model $\text{MLSBM}(Z, \Pi^{(1)}, \dots, \Pi^{(L)})$ if the adjacency matrix $A^{(l)}$ of each layer is distributed according to a $\text{SBM}(Z, \Pi^{(l)})$ for $l = 1, \dots, L$. Hence, while the probability that two nodes are connected can vary across layers, the block membership of each node remains unchanged. As in the unilayer case we can define the quantities $p^{(l)}_{max} = \max_{i,j} p^{(l)}_{ij}, \quad p_{max} = \max_l p^{(l)}_{max}$.

2.3 Missing nodes

The assumption that all the layers share the same set of nodes is quite restrictive since real world multilayer networks are often 'non-pillar'. We propose to deal with such networks by considering nodes present in some layers but not in others as missing. Let $w^{(l)}_i$ be a binary variable that records the presence of node i in the layer l where $w^{(l)}_i = 1$ if node i is observed in layer l and 0 otherwise. Denoting $w^{(l)} = (w^{(l)}_1, \dots, w^{(l)}_n)^T$, let $\Omega^{(l)} = w^{(l)} (w^{(l)})^T$ be the mask matrices and $\tilde{A}^{(l)} = A^{(l)} \odot \Omega^{(l)}$ for $l \leq L$ where $\odot$ is the usual Hadamard product. Let J_l denote the set of non-missing nodes in layer l with $n_{J_l} = |J_l|$. By a slight abuse of notation we will denote by A_{J_l} the matrix $A^{(l)}_{J_l}$. The number of observed nodes in $\mathcal{C}_k$ will also be denoted by $n_{J_l,k}$. Throughout, we assume that the missing nodes are generated as $w_i^{(l)} \overset{\text{ind.}}{\sim} \mathcal{B}(\rho)$ for $i = 1, \ldots, n$.

3 Final aggregation methods

A natural way to extend unilayer graph clustering to the multilayer setting is to analyze each layer separately and then find a consensus partition – such approaches are referred to as final aggregation methods. For example, one can apply any clustering method on each individual layer, take one layer's labels as a reference, find for each remaining layer the permutation of its labels that maximizes the agreement with the reference layer, and then define a consensus community by majority voting as discussed in Han et al. (2015). There exist alternative ways to avoid the cumbersome issue of label switching ambiguity such as the 'aggregate spectral kernel' considered in Paul and Chen (2020). Such methods rely on the quality of each individual layer and are often empirically outperformed by other methods as shown in Paul and Chen (2020); Han et al. (2015).

Final aggregation methods are still relevant in the missing nodes context. Indeed, if we have exact recovery for each layer, and if for all k there is at least one common node between two layers belonging to $\mathcal{C}_k$, then we can easily reconstruct the whole partition even when the set of common nodes is very small. Hence such methods can be considered as baseline methods.

3.1 A method based on a variant of k-means for incomplete data

We now propose a final aggregation method for clustering multilayer graphs in the incomplete setting; it avoids the aforementioned label switching problem.

For each layer l, we can compute the matrix $\hat{U}_{J_l}$ of size $|J_l| \times K$ corresponding to the eigenvectors associated with the top K eigenvalues (in absolute value) of $A_{J_l} \in \mathbb{R}^{|J_l|\times|J_l|}$. The matrix $\hat{U}_{J_l}$ can be transformed to a matrix $\hat{U}^{(l)}$ of size $n \times K$ by completing with 0 the rows of the nodes that haven't been observed[1]. Let $\hat{U}$ be the $n \times KL$ matrix obtained by stacking $\hat{U}^{(l)}$.

Analogously, let U_{J_l} be the matrix formed by the K eigenvectors corresponding to non-zero eigenvalues of $Z_{J_l}\Pi^{(l)}Z_{J_l}^T$, $U^{(l)}$ be the $n\times K$ matrix obtained from U_{J_l} by filling the rows corresponding to unobserved nodes with the row corresponding to an observed node (belonging to the same community), and U be the matrix obtained by stacking all the matrices $U^{(l)}$. For each l, let O_l be a $K \times K$ orthogonal matrix such that

$$O_l \in \underset{O^T O = I_k}{\operatorname{argmin}} ||\hat{U}_{J_l} - U_{J_l} O||_F.$$

As in the unilayer setting, k-means could be applied on the rows of $\hat{U}^{(l)}$ in order to recover the community structure for each l. But in order to avoid the label switching problem we propose to apply on the rows of $\hat{U}$ a variant of k-means described in Chi et al. (2015) that can handle missing values, see Algorithm 1.

Let us describe the principle behind this algorithm. The classical k-means problem seeks a partition Z and centroid values (encoded in the matrix C) that solves

$$\min_{\substack{Z\in\mathcal{M}_{n,K}\\ C\in\mathbb{R}^{K\times KL}}} ||\hat{U} - ZM||_F^2.$$

When there are missing values one can instead solve

$$\min_{\substack{Z\in\mathcal{M}_{n,K}\\ C\in\mathbb{R}^{K\times KL}}} ||(\hat{U} - ZM) \odot \Omega_{\mathrm{U}}\,||_F^2 \tag{3.1}$$

where $\Omega_{\mathrm{U}} = (w^{(1)} \otimes \mathbf{1}_K \ \cdots \ w^{(L)} \otimes \mathbf{1}_K)$ is the $n \times KL$ mask matrix with $\mathbf{1}_K \in \mathbb{R}^{1\times K}$ denoting the all ones vector. It is a matrix composed of L blocks where the rows of each block are 1 if the corresponding node is observed and 0 otherwise.

Algorithm 1 k-pod clustering

```
Input: The number of communities $K$, the sets $J_l$ and the adjacency matrices $A_{J_l}$.
Form $\hat{U}^{(l)}$ from $A_{J_l}$ as explained at the beginning of Section 3.1.
Form the matrix $\hat{U}$ by stacking the matrices $\hat{U}^{(l)}$.
Initialize the partition $\hat{Z}$ and the centroid matrix $\hat{M}$.
repeat
    Replace $\hat{U}$ by $\hat{U} \odot \Omega_{\mathrm{U}} + (\hat{Z}\hat{M}) \odot (\mathbf{1}\mathbf{1}^T - \Omega_{\mathrm{U}})$.
    Apply $K$-means on the complete matrix $\hat{U}$ and update $\hat{M}$ and $\hat{Z}$.
until convergence.
Output: A partition of the nodes $\mathcal{N} = \cup_{i=1}^K \mathcal{C}_i$ based on $\hat{Z}$.
```

In the worst case, the complexity of the algorithm is $O((L + K)n^2)$. But in practice the layers are often sparse and so the complexity will be much less[2].

Theorem 1. *Consider the missing nodes MLSBM in Section 2.3, and suppose that $\rho L \geq 1$, $KL \leq C_0 n$, $\rho n_{min} \geq C_1 K^2 \max(\log^2 n, \sqrt{np_{max}})$ and $np^{(l)}_{max} \geq$*

[1] It is easy to verify that $\hat{U}^{(l)}$ is also the eigenvector matrix corresponding to the top K eigenvalues (in absolute value) of $A^{(l)} \odot \Omega^{(l)}$.

[2] This remark regarding the complexity applies to our other methods as well.

$C_2\rho^{-1}\log n$. Let $\lambda_K^{(l)}$ be the K-th largest singular value of $\Pi^{(l)}$ and recall that $\beta = n_{max}/n_{min}$. If

$$\frac{1}{\rho L n}\sum_l \frac{p_{max}^{(l)}}{(\lambda_K^{(l)})^2} < (30C_3\beta^4K^3)^{-1}$$

then with probability at least $1-O(n^{-1})$, it holds that the solution $\hat{Z}\in\mathcal{M}_{n,K}$ of (3.1) satisfies

$$r(\hat{Z},Z) \leq C_4\exp(-c'\rho L) + \frac{C_5\beta^3K^2}{\rho L n}\sum_l \frac{p_{max}^{(l)}}{(\lambda_K^{(l)})^2}.$$

The proof of all our theoretical results are deferred to the supplementary material.

Remark 1. *The assumption $\rho L \geq 1$ is natural since ρL corresponds to the expected total number of times a node is observed, and a node needs to be observed at least once in order to be classified. The condition $\rho n_{min} \geq C_1K^2\log^2 n$ ensures that ρ and n_{min} are not too small. If the communities are well-balanced and the parameters ρ and K are fixed independently of n, then the previous condition is satisfied for n large enough.*

Remark 2. *Our analysis assumes that each layer is sufficiently informative, and doesn't use the fact that there is more information contained in the whole set of layers than in individual layers. This is why the bound does not improve when L increases. The obtained upper-bound is unlikely to be optimal since as shown in the experiments, the clustering performance does seem to improve a bit when L increases.*

4 Early fusion methods: spectral clustering on sum of adjacency matrices

Late fusion methods rely heavily on the quality of each layer. However, by simultaneously using all the information contained in all layers, the clustering performance can be improved in some settings (see the numerical experiments in Paul and Chen (2020) or Han et al. (2015)). One way to do this is to aggregate the information across layers and then apply a suitable clustering method. This approach will be referred to as an early fusion method. One simple but popular way to do this is to take the mean of the adjacency matrices (see for e.g., Bhattacharyya and Chatterjee (2018); Paul and Chen (2020)). Then, the k-means algorithm can be applied to the rows of the $n\times K$ eigenvector matrix associated with the top K eigenvalues (in absolute value) of $A = L^{-1}\sum_l A^{(l)}$.

4.1 Imputing missing entries with zeros

A natural way to extend the aforementioned approach to the setting of missing nodes is to fill the missing entries with zeros, thus leading to Algorithm 2. The worst-case complexity of the algorithm is $O((L+K)n^2)$.

Algorithm 2 Sum of adjacency matrices with missing entries filled with zeros

```
Input: The number of communities $K$, the matrices $A^{(l)}$ and $\Omega^{(l)}$.
Compute $A = L^{-1}\sum_l A^{(l)} \odot \Omega^{(l)}$.
Compute the eigenvectors $u_1,\dots,u_K$ associated with the $K$ largest eigenvalues of $A$ (ordered in absolute values) and form $U_K = [u_1\ u_2\ \cdots\ u_K]$.
Apply $K$-means on the rows of $U_K$ to obtain a partition of $\mathcal{N}$ into $K$ communities.
Output: A partition of the nodes $\mathcal{N} = \cup_{i=1}^K \mathcal{C}_i$.
```

Let us denote $\tilde{A} = \rho^{-2}L^{-1}\sum_l A^{(l)}\odot\Omega^{(l)}$ (clustering on A or $\tilde{A}$ is equivalent since the two matrices are proportional, but for the analysis it is more convenient to work with $\tilde{A}$). Clearly $\mathbb{E}(\tilde{A}) = L^{-1}\sum_l \mathbb{E}(A^{(l)})$ (since the diagonal entries of $A^{(l)}$ are zero). Denote by $\mathbb{E}(X|\Omega)$ to be the expectation of X conditionally on $\Omega = (\Omega^{(1)},\dots,\Omega^{(L)})$ and let λ_K denote the Kth largest singular value of $\mathbb{E}(\tilde{A})$. We have $\mathbb{E}(\tilde{A}|\Omega) = \rho^{-2}L^{-1}\sum_l \mathbb{E}(A^{(l)})\odot\Omega^{(l)}$. Using the same kind of perturbation arguments and concentration inequalities as in Lei and Rinaldo (2015), we can relate $\tilde{A}$ to $\mathbb{E}(\tilde{A}|\Omega)$ and then use Bernstein inequality to relate $\mathbb{E}(\tilde{A}|\Omega)$ with $\mathbb{E}(\tilde{A})$. This leads to the following bound on the misclustering rate.

Theorem 2. *Under the missing nodes MLSBM in Section 2.3, there exist constants $C_0, C_1 > 0$ such that with probability at least $1-O(n^{-1})$, the solution $\hat{Z}\in\mathcal{M}_{n,K}$ obtained from Algorithm 2 satisfies*

$$r(\hat{Z},Z) \leq \underbrace{\frac{C_0K}{\rho^4\lambda_K^2}\left(\frac{np_{max}}{L}+\frac{\log n}{L}\right)}_{\textit{noise error}} +$$

$$\underbrace{C_1K\frac{(\rho^{-2}-1)^2}{\lambda_K^2}\left((np_{max})^2\frac{\log(n)}{L}+\left(\frac{np_{max}\log n}{L}\right)^2\right)}_{\textit{missing data error}}.$$

If L is small then the missing data error could be larger than one making the upper bound trivial. In the best case scenario, we expect that λ_K scales as np_{max}. So we need at least $C\log n$ layers to get a non trivial upper bound. In order to obtain asymptotic consistency, it is necessary that $L \gg \log n$. However, experiments show that even when L is small, Algorithm 2 gives good results as long as the layers are dense enough and the number of missing nodes is not too large.

When $\rho = 1$ and $np_{max} \geq \log n$ the upper bound becomes $O((Lnp_{max})^{-1/2})$ thus matching the bound obtained by Bhattacharyya and Chatterjee (2018) in a more general regime. See the supplementary material for other comparisons.

4.2 Iteratively imputing the missing entries

When the number of missing nodes is important, filling missing entries with zero can lead to a huge bias and hence poor clustering performances. In order to reduce the bias we propose an alternative way of imputing the missing values (outlined as Algorithm 3) based on the fact that each adjacency matrix is a noisy realization of a structured matrix.

At iteration t, given an initial estimate $\hat{U}_K^t \in \mathbb{R}^{n\times K}$ of the common subspace we can estimate the membership matrix $\hat{Z}^t$ by applying k-means on $\hat{U}_K^t$. Then, we can estimate the connectivity matrix $\hat{\Pi}^{(l),t}$ for each l as

$$\hat{\Pi}^{(l),t} = ((\hat{Z}^t)^T\hat{Z}^t)^{-1}(\hat{Z}^t)^T A^{(l),t}\hat{Z}^t((\hat{Z}^t)^T\hat{Z}^t)^{-1}. \tag{4.1}$$

Given $\hat{Z}^t$ and $\hat{\Pi}^{(l),t}$ we estimate the rows and columns corresponding to missing nodes. Indeed, the connectivity profile of a node i in layer l is given by the ith row of $\hat{Z}^t\hat{\Pi}^{(l),t}(\hat{Z}^t)^T$. By replacing the rows and columns of missing nodes by their estimated profiles, and leaving the value of observed nodes unchanged, we obtain the updated imputed matrix $A^{(l),t+1}$. Applying spectral clustering on $L^{-1}\sum_l A^{(l),t+1}$ then leads to an updated estimate $\hat{U}_K^{t+1}$ of the common subspace. The procedure can be repeated using $\hat{U}_K^{t+1}$ and $A^{(l),t+1}$, thus iteratively imputing the missing values in order to obtain "completed" adjacency matrices that share the same K rank structure across layers. In the worst case, the complexity of the algorithm run with T iterations is $O((K+L)n^2T + LKnT)$.

Similar iterative imputation methods have been studied in the context of principal component analysis, see for e.g., Zhang et al. (2018); Zhu et al. (2019). In our experiments, Algorithm 3 is seen to perform significantly better than other methods when ρ decreases. While we do not currently have any statistical performance guarantee for Algorithm 3, establishing this is an interesting direction for future work.

Algorithm 3 Sum of adjacency matrices with missing entries filled iteratively

```
Input: Number of communities $K$; $J_l$ and $A_{J_l} \in \mathbb{R}^{n\times n}$ for each $l$; initial estimate of the common subspace $\hat{U}_K^0 \in \mathbb{R}^{n\times K}$ (with orthonormal columns) obtained from Algorithm 2; number of iterations $T$.
Initialize $t = 0$ and $A^{(l),0} = A_{J_l}$ for all $l$.
repeat
    Given $\hat{U}_K^t$, estimate the membership matrix $\hat{Z}^t$ and the connectivity parameters $\hat{\Pi}^{(l),t}$ for all $l$ by using (4.1).
    For each $l$, replace rows (and corresponding columns) of $A^{(l)}$ corresponding to a missing node $i$ by the $i$th row of $\hat{Z}^t\hat{\Pi}^{(l),t}\hat{Z}^{t^T}$ to form $A^{(l),t+1}$.
    Compute the eigenvector matrix $\hat{U}_K^{t+1} = [u_1^{t+1}\ u_2^{t+1}\ \cdots\ u_K^{t+1}]$ associated with the $K$ largest (in absolute order) eigenvalues of $L^{-1}\sum_l A^{(l),t+1}$. Update $t \leftarrow t+1$.
until $t \le T$
Apply $K$-means on $\hat{U}_K^T$ to get a partition of $\mathcal{N}$.
Output: A partition of the nodes $\mathcal{N} = \cup_{i=1}^K \mathcal{C}_i$.
```

5 Intermediate fusion methods: OLMF estimator

Orthogonal linked matrix factorization (OLMF) is a clustering method for multilayer graphs that originated in the work of Tang et al. (2009) in the complete data setup, and was later analysed in Paul and Chen (2020). It shows good performance in various settings and outperforms spectral clustering when the multilayer network contains homophilic and heterophilic communities (see the numerical experiments in Paul and Chen (2020)).

5.1 The complete data setting

In the complete data setting, the OLMF estimator is a solution of the following optimization problem

$$(\hat{Q}, \hat{B}^{(1)}, \dots, \hat{B}^{(L)}) \in \underset{\substack{Q^TQ=I_k \\ B^{(1)},\dots,B^{(L)}}}{\operatorname{argmin}} \sum_l ||A^{(l)} - QB^{(l)}Q^T||_F^2, \tag{5.1}$$

where $Q \in \mathbb{R}^{n\times K}$, $B^{(l)} \in \mathbb{R}^{K\times K}$. Note that there is no constraint on the values taken by the entries of $B^{(l)}$.

A little algebra (see Paul and Chen (2020)) shows that the optimization problem (5.1) is equivalent to

$$\hat{Q} \in \underset{Q^TQ=I_k}{\operatorname{argmax}} \sum_l ||Q^TA^{(l)}Q||_F^2, \quad \hat{B}^{(l)} = \hat{Q}^TA^{(l)}\hat{Q} \tag{5.2}$$

for $l = 1, \dots, L$. The OLMF estimator can be computed with a gradient descent on the Stiefel manifold (see Paul and Chen (2020) and supplementary material therein). The community estimation is then obtained by applying K-means on the rows of $\hat{Q}$.

5.2 Extension to the missing nodes setting

We now present an extension of the OLMF estimator to the setting of missing nodes. By replacing the matrices $A^{(l)}$, Q in the objective function in (5.1) with $A_{J_l} \in \mathbb{R}^{n\times n}$, $Q_{J_l} \in \mathbb{R}^{n\times K}$, we end up with the following modification for the incomplete setting

$$(\hat{Q}, \hat{B}^{(1)}, \dots, \hat{B}^{(L)}) \in \underset{\substack{Q^TQ=I_k \\ B^{(1)},\dots,B^{(l)}}}{\operatorname{argmin}} \sum_l ||A_{J_l} - Q_{J_l} B^{(l)} Q_{J_l}^T||_F^2. \tag{5.3}$$

In our experiments, we employ a BFGS algorithm for solving (5.3). The worst-case complexity of the algorithm is $O(LK(n^2 + Kn))$. Denoting the objective function in (5.2) by F, its gradients are given by

$$\frac{\partial F}{\partial Q} = -2 \sum_l (A_{J_l} - Q_{J_l} B^{(l)} Q_{J_l}^T) Q_{J_l} B^{(l)},$$
$$\frac{\partial F}{\partial B^{(l)}} = -Q_{J_l}^T (A_{J_l} - Q_{J_l} B^{(l)} Q_{J_l}^T) Q_{J_l}.$$

We relax the constraint that the gradient remains on the Stiefel manifold of $n \times k$ matrices, and initialize the parameters using Algorithm 2.

The optimization problem in (5.2) can be motivated via the missing nodes MLSBM as follows. If we replace the noisy realization A_{J_l} with $(Z\Pi^{(l)}Z^T) \odot \Omega^{(l)}$ then one can show (under some conditions) that the solution $\hat{Q}$ of (5.3) has the same column span as the ground truth $Z \in \mathcal{M}_{n,K}$. This is shown formally in the following proposition.

Proposition 1. *Assume that $\Pi^{(l)}$ is full rank for each l, and that for each l, l' the sets $J_l \cap J_{l'}$ intersect all communities. Then if $A_{J_l} = (Z\Pi^{(l)}Z^T) \odot \Omega^{(l)}$, it holds that the solution of (5.3) is given by $\hat{Q} = Z(Z^TZ)^{-1/2}$ and $\hat{B}^{(l)} = (Z^TZ)^{1/2}\Pi^{(l)}(Z^TZ)^{1/2}$ and is unique up to an orthogonal transformation. Moreover if i, j belong to the same community, then $\hat{Q}_{i*} = \hat{Q}_{j*}$.*

The matrix $\mathbb{E}(A^{(l)})$ can be considered as a slight perturbation of $Z\Pi^{(l)}Z^T$ since the former has zeros on the diagonal. Thus the proposition shows that when there is no noise, the column-span of $\hat{Q}$ (the solution of (5.3)) is the same as the ground truth partition Z.

6 Numerical experiments

6.1 Synthetic data

We now describe simulation results when the multilayer graph is generated from the missing nodes MLSBM. The normalized mutual information (NMI) criterion is used to compare the estimated community to the ground truth partition. It is an information theoretic measure of similarity taking values in $[0, 1]$, with 1 denoting a perfect match, and 0 denoting completely independent partitions. Nodes that are not observed at least once are removed. The diagonal (resp. off-diagonal) entries of the connectivity matrices are generated uniformly at random over $[0.18, 0.19]$ (resp. $0.7 * [0.18, 0.19]$). The ground truth partition is generated from a multinomial law with parameters $1/K$. While $K = 3$ is fixed throughout, the parameters n, ρ and L are varied suitably. The average NMI is reported over 20 Monte Carlo trials. As shorthand, we denote Alg. 1 by `k-pod`, Alg. 2 by `sumAdj0`, Alg. 3 by `sumAdjIter`, and (5.3) by `OLMFm`.

Figure 1 shows that `sumAdj0` gives good results unless ρ is too small. Then, the performance of this method decreases quickly. This suggests that there is a threshold involving ρ and the difference between intra and inter connectivity parameters. Figure 3 supports this claim. When ρ is small, the performance of `sumAdj0` doesn't improve when n increases. So even if the separation between communities improves, the intra and inter connectivity parameters remain the same suggesting a link between these parameters and ρ.

When L increases (see Figs. 1 and 2), the performance of all methods improves. However, performance of `k-pod` improves less quickly than other methods. This is expected since contrary to other methods, `k-pod` relies more on the quality of each individual layer. `OLMFm` and `sumAdjIter` exhibit better performance than others in the challenging situation when ρ is small, and perform as well as the others when $\rho \approx 1$. They perform significantly better than `k-pod`, especially when L is large.

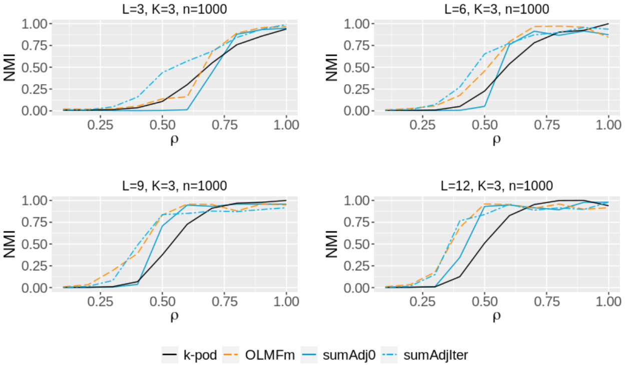

Figure 1: NMI vs ρ for different values of L

6.2 MIT Reality Mining dataset

This dataset records interactions (measured by cell phones activities) between 96 students and staff at MIT in the 2004-05 school year (see Eagle and Pentland (2006)). We used the dataset as provided by the R package 'GreedySTBM'. As in Han et al. (2015) we removed the first and last layers, then discretized the time into one week intervals. The number of times two persons had an interaction during the week is not conserved in order to have a simple undirected graph

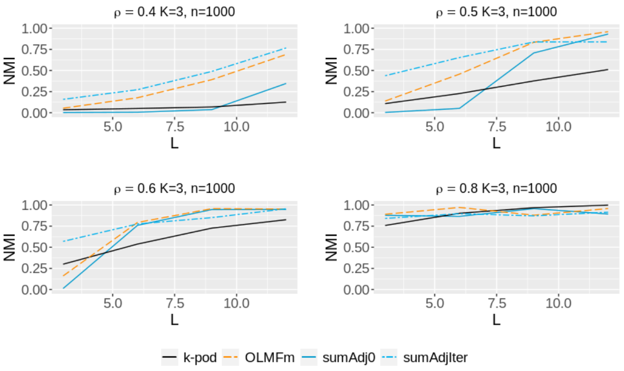

Figure 2: NMI vs L for different values of ρ

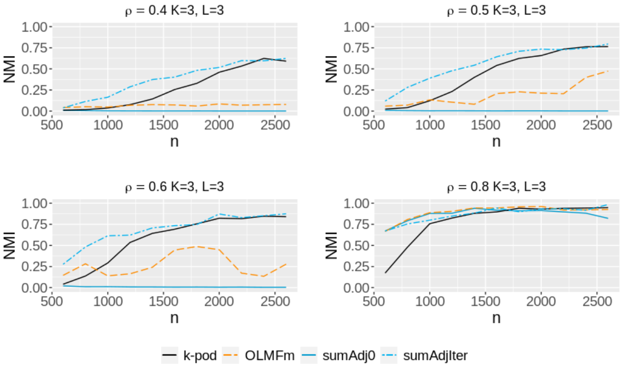

Figure 3: NMI vs n for different values of ρ

corresponding to each layer. In total we obtained 32 layers. For different values of ρ, we randomly removed nodes in each layer of the multilayer network. The average NMI over 50 Monte Carlo trials is reported in Table 1 for our methods. The ground truth partition here is taken to be that obtained from `sumAdj0` when $\rho = 1$. We disregarded `k-pod` because even when $\rho = 1$, its performance was disappointing and very sensitive to the initialization. This is not very surprising since this method works only if each layer is informative enough while we have a multilayer network where individual layers can be very sparse.

ρ	`sumAdj0`	`OLMFm`	`sumAdjIter`
1	1.00	1.00	1.00
0.9	0.99	0.96	0.99
0.8	0.97	0.86	0.97
0.7	0.96	0.93	0.96
0.6	0.94	0.79	0.94
0.5	0.89	0.91	0.90
0.4	0.76	0.73	0.78
0.3	0.56	0.57	0.62
0.2	0.26	0.41	0.36
0.1	0.09	0.10	0.11

Table 1: NMI vs ρ for MIT Reality Mining dataset

The performance of the other three methods studied are quite similar when ρ is not too small ($\rho \geq 0.4$). However, the performance of `OLMFm` seems to be quite sensitive to initialization since for $\rho \in \{0.6, 0.8\}$ its performance is worse than `sumAdj0` and `sumAdjIter`. Even if we remove half of the nodes in each layer we can still approximately recover the partition.

6.3 Malaria parasite genes network

The dataset was constituted by Larremore et al. (2013) to study the var genes parasite Plasmodium falciparum involved in Malaria. The nodes of the dataset correspond to 307 different amino acid sequences and each of the 9 layers corresponds to a highly variable region (HVR). Two nodes are linked in a given layer if there is a common block sequence between the corresponding amino acid sequences within the HVR associated to the layer. The analysis in Larremore et al. (2013) and Jing et al. (2020) shows that the first six layers share the same community structure with $K = 4$. Hence we restrict our study to the first six layers with $K = 4$. We use the same procedure as before to delete nodes and to select the ground truth partition. `k-pod` was disregarded for the same reason as the previous experiment. As ρ decreases, the clustering performance decreases rapidly due to a weak separation between the clusters as shown in Table 2.

ρ	`sumAdj0`	`OLMFm`	`sumAdjIter`
1	1.00	0.99	1.00
0.9	0.75	0.75	0.72
0.8	0.63	0.62	0.58
0.7	0.47	0.49	0.47
0.6	0.32	0.37	0.34
0.5	0.22	0.20	0.26
0.4	0.13	0.07	0.16

Table 2: NMI vs ρ for Malaria parasite genes network

7 Future work

Our theorems require different conditions for consistency (each layer has to be informative enough for Algorithm 1 and L has to be large for Algorithm 2). It would be interesting to gain a better understanding of the fundamental limit of clustering with missing nodes. In this regard the use of two-round algorithms (see for e.g., Abbe (2018)) that do local refinement after having found a global partition could improve the misclustering rate. It would also be interesting to consider model-based approaches by considering variational methods (Daudin et al. (2008)) or Stochastic-EM algorithms (Celeux et al. (1996)).

We assumed for simplicity that the nodes are missing under a Bernoulli sampling scheme, but other missing patterns could be considered. Another important direction would be to relax the strong condition imposed by MLSBM that all layers share the same common partition. For example, it would be more realistic to assume that the partition of networks evolving over time also evolves slowly.

Bibliography

E. Abbe. Community detection and stochastic block models. *Foundations and Trends® in Communications and Information Theory*, 14(1-2):1–162, 2018.

S. Bhattacharyya and S. Chatterjee. Spectral clustering for multiple sparse networks: I. *arXiv*, 1805.10594, 2018.

S. Bhattacharyya and S. Chatterjee. General community detection with optimal recovery conditions for multi-relational sparse networks with dependent layers. *arXiv*, 2004.03480, 2020.

U. Braun, A. Schäfer, H. Walter, S. Erk, N. Romanczuk-Seiferth, L. Haddad, J. Schweiger, O. Grimm, A. Heinz, H. Tost, A. Meyer-Lindenberg, and D. Bassett. Dynamic reconfiguration of frontal brain networks during executive cognition in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 112, 08 2015.

G. Celeux, D. Chauveau, and J. Diebolt. Stochastic versions of the em algorithm: an experimental study in the mixture case. *Journal of statistical computation and simulation*, 55(4):287–314, 1996.

J. Chi, E. Chi, and R. Baraniuk. k -pod a method for k -means clustering of missing data. *The American Statistician*, 70:1–29, 2015.

J.-J. Daudin, F. Picard, and S. Robin. A mixture model for random graph. *Statistics and Computing*, 18:173–183, 06 2008. doi: 10.1007/s11222-007-9046-7.

N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486, 2009.

Q. Han, K. Xu, and E. Airoldi. Consistent estimation of dynamic and multi-layer block models. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, page 1511–1520, 2015.

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983.

M. Hu and S. Chen. One-pass incomplete multi-view clustering. In *The Thirty-Third Conference on Artificial Intelligence*, pages 3838–3845, 2019.

B.-Y. Jing, T. Li, Z. Lyu, and D. Xia. Community detection on mixture multi-layer networks via regularized tensor decomposition. *arXiv*, 2002.04457, 2020.

J. Kim and J.-G. Lee. Community detection in multi-layer graphs: A survey. *SIGMOD Record*, 44:37–48, 2015.

M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

D. Larremore, A. Clauset, and C. Buckee. A network approach to analyzing highly recombinant malaria parasite genes. *PLoS computational biology*, 9: e1003268, 2013.

J. Lei. Tail bounds for matrix quadratic forms and bias adjusted spectral clustering in multi-layer stochastic block models. *arXiv*, 2003.08222, 2020.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 02 2015.

J. Lei, K. Chen, and B. Lynch. Consistent community detection in multi-layer network data. *Biometrika*, 107(1):61–73, 2019.

X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu. Efficient and effective regularized incomplete multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

S. Paul and Y. Chen. Spectral and matrix factorization methods for consistent community detection in multi-layer networks. *The Annals of Statistics*, 48 (1):230–250, 2020.

M. Pensky and T. Zhang. Spectral clustering in the dynamic stochastic block model. *Electron. J. Statist.*, 13(1):678–709, 2019.

W. Tang, Z. Lu, and I. Dhillon. Clustering with multiple graphs. In *IEEE International Conference on Data Mining*, pages 1016–1021, 2009.

A. Zhang, T. T. Cai, and Y. Wu. Heteroskedastic pca: Algorithm, optimality, and applications. *arXiv*, 1810.08316, 2018.

Z. Zhu, T. Wang, and R. J. Samworth. High-dimensional principal component analysis with heterogeneous missingness. *arXiv*, 1906.12125, 2019.