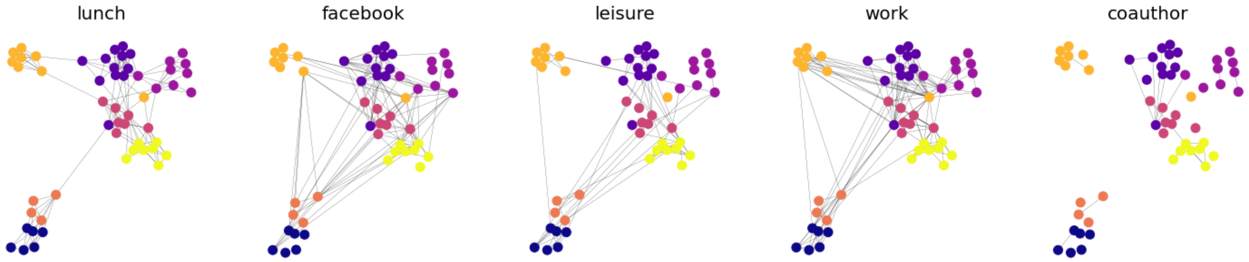
Clustering with Multi-Layer Graphs: A Spectral Perspective

Souheib Ben Mabrouk
souheib.benmabrouk@telecom-paris.fr
Institut Polytechnique de Paris

Imen Mahdi
imen.mahdi@telecom-paris.fr
Institut Polytechnique de Paris



A multi-layer graph representing AUCS social network interactions.

ABSTRACT

The main objective of [1] is to explore the effectiveness of multi-layer graphs for clustering, offering richer information than simple graphs. Two different methods are proposed to take advantage of the joint spectrum of the graph, one using Laplacian matrix eigen-decomposition and the other based on graph regularization. We review the theoretical aspects of both algorithms and our experimental results on real datasets. In our experiments, we show that in most cases, those methods outperform clustering baselines on multi-layer graphs. We later discuss the limitations of the proposed approaches and speculate about the reasons behind them. Our implementation is available at https://github.com/souheib1/Clustering-with-Multi-Layer-Graphs.

KEYWORDS

Multi-Layer Graphs, Spectral Clustering

1 CONTEXT

Graphs are widely used as a data structure for representing interdependence between objects, such as interactions in dynamic social networks. Clustering is a fundamental task for graph mining across various domains such as biology, sociology, and economics. It unveils intrinsic structures and patterns within the graph, enabling a deeper understanding of the underlying relationships[3]. A multi-layer graph is a collection of interconnected graphs that share a common set of vertices while featuring distinct sets of edges. Multi-layer graphs have received significant research attention thanks to their inherent representation of multi-model data. Indeed, the different modalities for a given population can be perceived as a graph whose layers share the same set of nodes (representing the population) but have different edges for each modality (different relations between the individuals). In the classical approach, each layer contains a feature that is significant for graph clustering, and we can regroup vertices according to this feature. However, we can intuitively expect that an appropriate combination of layers will enable a better clustering that takes into account the information embedded in each layer. The main contribution of [1] is proposing two novel methods that properly leverage the spectrum of multiple graph layers to create a “joint spectrum” that will be used to perform the clustering. In Section 2.2, we introduce the first method based on a generalized eigen-decomposition, a broader adaptation of the conventional single-layer graph spectral clustering algorithm. The second method, detailed in Section 2.3, involves a spectral regularization approach that seeks to find a smoothing function across the diverse layers of the graphs. Moving forward, Section 3 details the experiments conducted on real-world datasets, while Section 4 highlights the limitations of the methods and explores potential avenues for extension.

2 METHODS

2.1 Specral clustering

The spectral clustering algorithm [4] is a conventional graph clustering method, based on the analysis of the Laplacian matrix $L = D - W$, where D represents the degree matrix and W the adjacency matrix of the graph. The algorithm aims to identify k clusters by picking out the k smallest eigenvalues of L and their corresponding eigenvectors $u_1, ..., u_k$. Using the K-means algorithm on these eigenvectors simplifies the clustering process, efficiently projecting the vertices of the graph into a lower-dimensional spectral domain. Known for its efficiency and simplicity of implementation, this algorithm is very popular and motivates the study of [1] to extend this methodology to multi-layer graphs.

2.2 Clustering with Generalized Eigen-Decomposition

The core idea behind this method is to aggregate information from multiple layers by finding a set of joint eigenvectors to the Laplacian matrices $L_{\mathrm{rw}}^{(i)}$ for $i = 1, \ldots, M$. These eigenvectors are then used to construct an average spectral embedding matrix, which serves to cluster the nodes. Formally, we want to find a matrix P such that:

$$L_{\mathrm{rw}}^{(i)} \approx P\Lambda^{(i)}P^{-1} \text{ for } i = 1, \ldots, M \tag{1}$$

where P is a $n{\times}n$ matrix that contains the set of joint eigenvectors as columns, and $\Lambda^{(i)}$ is the eigenvalue matrix of $L_{\text{rw}}^{(i)}$ for $i = 1, \ldots, M$. One possible approach is to average the Laplacian matrices and then find the eigenvectors of the average matrix.

$$\begin{aligned}\bar{L}_{\text{rw}} &= \frac{1}{M}\sum_{i=1}^{M} L_{\text{rw}}^{(i)} \\ &= P\left(\frac{1}{M}\sum_{i=1}^{M}\Lambda^{(i)}\right)P^{-1} \\ &= P\bar{\Lambda}P^{-1}\end{aligned} \tag{2}$$

where $\bar{L}$ is the average Laplacian matrix, and $\bar{\Lambda}$ is the average eigenvalue matrix.

This assumes that all layers are equally informative, which is not always the case. Instead, we choose one layer that captures the most meaningful relationships between nodes and compute its eigenvalue decomposition. We approximate the joint eigenvectors by setting the eigenvectors of the chosen layer as the initialization and refine the solution by solving the following optimization problem:

$$\begin{aligned}\arg \min_{P,Q\in\mathbb{R}^{n\times n}} S = \frac{1}{2}\sum_{i=1}^{M}\left\|L_{\text{rw}}^{(i)} - P\Lambda^{(i)}Q\right\|_F^2 \\ + \frac{\alpha}{2}\left(\|P\|_F^2 + \|Q\|_F^2\right) + \frac{\beta}{2}\|PQ - I_n\|_F^2\end{aligned}$$

where α and β are regularization parameters. The first term is the data fidelity term, and the second and third terms are regularization terms. Q is an $n{\times}n$ matrix representing the inverse of P. We enforce the orthogonality of P and Q through the orthogonality penalty term $\frac{\beta}{2}\|PQ - I_n\|_F^2$. The second term $\frac{\alpha}{2}\left(\|P\|_F^2 + \|Q\|_F^2\right)$ is a ridge penalty on the columns of P and Q to ensure numerical stability. The first term ensures that the solution to the problem truly describes the Laplacian matrices of all layers as much as possible. This method emphasizes the weight of the chosen layer in the final solution while still considering information gained from the rest of the layers.

Intuitively, through the initialization with eigenvectors from the most relevant layer for clustering, we prioritize information originating from this specific layer. However, given the sparsity of these layers, some nodes may not be well-classified due to insufficient information. To mitigate this, we compensate by integrating information from the remaining layers, which is ensured through the optimization process.

2.3 Clustering with spectral regularization

In this section, we explain the second method presented in [1] where we take into account all of the layers of the graph according to their respective importance. Specifically, certain layers may provide information that is more relevant to the problem and should therefore have a stronger impact on the clustering. Intuitively, this method is based on mapping the graph into 1-dimensional space such that projections of connected vertices are preserved as close as possible. Formally, we seek a scalar function $f : V \longrightarrow \mathbb{R}$ that satisfies, for each node v_i and $v_j \in V, thatf\left(v_i\right) - f\left(v_j\right)$ is small if v_i and v_j are connected. This function corresponds to the solution to the optimization problem :

$$\arg\min_f \sum_{i=1}^{n} \omega_{i,j}\left(f\left(v_i\right) - f\left(v_j\right)\right)^2, \ \ \textit{such that} \ \ \|f\| = 1, f \perp 1 \tag{3}$$

where $\omega_{i,j}$ is the (i, j) coordinate of the adjacency matrix of the graph, and the constraints ensure a non-trivial and unique solution [7]. This problem is equivalent to: Eq. [4]

$$\arg\min_f f^T L f \ \ \textit{such that} \ \|f\| = 1, f \perp 1. \tag{4}$$

The quadratic objective in Eq. [4] can be interpreted as a measure of smoothness for f defined on the vertex set of a graph G and since the u_i eigenvalues of L satisfy $u_i L u_i^T = \lambda_i$ where λ_i are the associated eigenvalues, small eigenvalues indicate that the corresponding eigenvectors are promising candidates for the f function. This insight inspires the approach to leverage information from multiple graph layers through a set of shared eigenvectors that exhibit smoothness across all layers. However, rather than treating all layers in the same way, a better approach is to weigh the effect of each layer according to its importance. We start by sorting the layers in order of importance, with $\mathcal{G}^0$ being the most informative. We consider $(u_i)_i$ its eigenvectors. We determine a jointly smooth spectrum of two graph layers $\mathcal{G}^0$ and $\mathcal{G}^1$ by solving the optimization problem

$$\arg\min_f \frac{1}{2}\|f_i - u_i\|^2 + \lambda f_i^T L_{rw}^{(1)} f_i \tag{5}$$

where $f_i^T L_{rw}^{(1)} f_i$ is a quadratic term quantifying the smoothness of f_i on $\mathcal{G}^1$, and λ is a parameter balancing the trade-off between the data fidelity term and the regularization term in the objective function. In particular, $\mathcal{G}^0$ is used to obtain the eigenvectors, and $\mathcal{G}^1$ is the graph structure for the smoothing process. The closed-form solution for the problem in Eq. [5] is given by

$$f_i^* = \frac{1}{\lambda}(L_{\text{sym}} + \frac{1}{\lambda}I)^{-1}u_i \tag{6}$$

where $\text{Ł}_{\text{sym}} = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$ the symmetric normalized version of Laplacian matrix.

3 EXPERIMENTS

In this section, we start by presenting the datasets used to evaluate our methods against baseline approaches. Then, We describe the evaluation criteria used and give a speculative overview of the numerical results achieved.

3.1 Datasets

To evaluate and compare the performance of the proposed algorithms with existing methods, we use three different datasets among which two are used in the original paper.

(1) **The Cora Dataset [5]** serves as an extensive bibliographic repository comprising diverse research papers, each accompanied by pertinent details such as titles, authors, and citations. Papers within this dataset are categorized according to their primary field of interest. Given the dataset's size, our focus is specifically on papers categorized under Natural Language Processing, Data Mining, or Robotics. In

contrast to the original work, which evaluated methods on a subset of 292 nodes, our approach retains all 901 nodes. Notably, our selection criteria differ from the original paper due to the absence of explicit information regarding the preprocessing strategy employed. To construct the multilayer graph, we extract three adjacency matrices. The initial layer captures cosine similarity between titles, expressed as vectors of non-trivial words. The second layer represents shared authors as edges, while the third adjacency matrix represents citation connections among the papers.

(2) **The MIT Reality Mining Dataset [2]** forms a social network comprising 94 mobile users, consisting of both students and faculty members, situated on the MIT campus. The dataset spans a period of 9 months, encompassing recorded information about call logs, device proximity, and application usage, alongside self-reported relational data regarding proximity and friendship among individuals. Following preprocessing steps, we narrowed down our nodes to 90 subjects for whom affiliations are available, organizing them into 7 distinct clusters. Our first graph is constructed by setting the total number of calls as the weight of an edge between two nodes. The second graph is derived from the number of occurrences where two mobile devices were in proximity within a 30-minute window. The final layer is formed by computing an average of the self-reported friendship values between pairs of nodes. Similar to the previous case, replicating the exact dataset as in the original work proved challenging for us.

(3) **The AUCS Dataset [6]** is proposed as an extension to the original work. In this graph, the multiple layers represent relationships between 61 employees of a University department. Specifically, the layers encompass various aspects of social interaction during and outside of work such as having lunch together, working in the same office, friendship on Facebook, leisure activities outside of work, and co-authorship. Each subject is affiliated with one of 8 research groups and has a job title. For our implementation, we designate the individual research group as the ground truth cluster for that subject.

3.2 Clustering Algorithms

We implemented the normalized spectral algorithm for single-layer graphs and the two algorithms explained in Section 2. For the SC-GED implementation, we retained the choice of $\alpha = 0.5$ proposed in the paper for all datasets. However, in our case, $\beta = 10$ appeared to yield better performances. Following the paper's recommendation, we conducted the optimization process through repeated steps of the quasi-Newton method (Limited-Memory BFGS). For both SC-GED and SC-SR, it is essential to infer the order of the layers based on their relevance to clustering. Specifically, for SC-GED, we initialize P and Q with the eigenvectors of the Laplacian matrix of the most informative layer. The second algorithm computes the joint spectrum recursively by combining the spectrum based on their mutual information after constructing the first low-dimensional embedding using the most informative layer. Therefore, ordering the layers from the most informative to the least informative is necessary, a task that is not trivial in most cases, as discussed in Section 4. Finally, we also implemented some reference methods, namely the Kernel K-means (K-KMEANS) algorithm, the Spectral Clustering applied to the averaged Laplacian matrix algorithm (SC-AL), and Spectral Clustering applied to the summation of adjacency matrices (SC-SUM) described in [1].

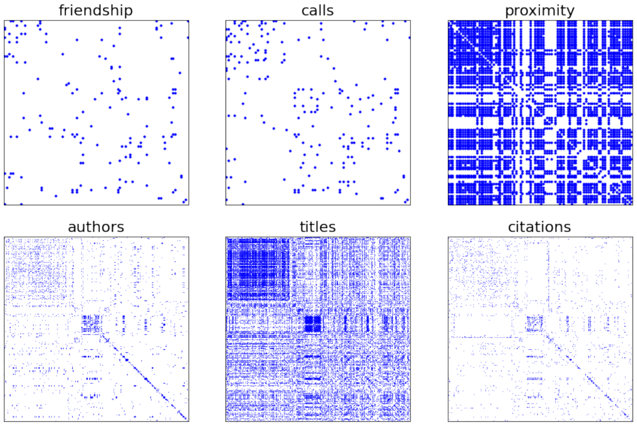


Figure 1: Spy plots illustrating the adjacency matrices for the three layers of the multilayer graphs, following the specified criteria in the titles. Top: MIT Reality Mining dataset. Bottom: Cora dataset.

3.3 Evaluation Metrics

To evaluate the clustering performance, we compare the predicted clusters $\Omega = \{\omega_1, \ldots, \omega_k\}$ with the ground truth classes $C = \{c_1, \ldots, c_k\}$ using Purity, Normalized Mutual Information (NMI), Rand Index (RI), and Modularity (Q).

- Purity :

$$\text{Purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j| \tag{7}$$

where N is the number of nodes, and $|\omega_k \cap c_j|$ is the number of common nodes of ω_k and c_j.

- Normalized Mutual Information :

$$NMI(\Omega, C) = \frac{I(\Omega; C)}{[H(\Omega) + H(C)]/2} \tag{8}$$

where I is the mutual information between the classes C and clusters Ω, and $(H(\Omega)\ , H(C))$ are the respective entropy.

- Rand Index (RI)

$$RI(\Omega, C) = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

where FP, FN, TP, and TN are false positive and false negative, true positive, and true negative decisions, respectively.

- Modularity :

$$Q = \frac{1}{2m} \sum_{ij} \left(W_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{10}$$

where k_i and k_j are the degrees of nodes i and j, W is the adjacency matrix of the graph, m is the total number of edges in the network, and δ is the Kronecker delta function.

3.4 Results

This table 2 shows a quantitative comparison of the performance of different clustering algorithms across three datasets. One key observation is the significant variability in performance across datasets, which we attribute to potential differences in the quality of extracted adjacency matrices, particularly evident in the Cora and MIT cases.

Assessing the first method, SC-GED demonstrates marginal improvement over single-layer eigen decomposition, with slight advantages over other averaging methods like SC-SUM and SC-AL. However, the effectiveness depends heavily on the choice of the initialization layer. This parameter is not always readily inferable and often requires expert knowledge. This dependency is evident in Figure 3, where we illustrate in the first row how SC-GED brings limited improvement to the clustering results and can, at times, even degrade them when the initialization layer is selected to be the most informative. We speculate that the reason behind this is that the optimization process attempts to approximate eigenvectors that are common to all layers, posing challenges in finding a solution that accurately represents the data across all graphs. In fact, we argue that this is counter-intuitive as the graphs are very different and can not be represented by the same eigenvectors. The dataset's size exacerbates this issue, as illustrated in the Cora dataset with 901 nodes compared to around 100 for the other two datasets, resulting in the most significant performance gap. In the second row, choosing a less informative initializing layer and comparing its spectral clustering to that of SC-GED reveals that, in this scenario, improvement is still achievable by incorporating information from other layers. We generalize this by stating that when we lack a clear idea of the most informative layers, using SC-GED could be a viable strategy to enhance results. If our choice of the informative layer aligns with the optimal one, the results will be similar to the optimal ones; if the choice of the layer is suboptimal, SC-GED can improve performance compared to using that same graph for single-layer spectral clustering.

Additionally, SC-SR shows slight improvements on the well constructed AUCS dataset, presenting competitive results when compared to K-Kmeans. Nevertheless, the performance of SC-SR depends on the chosen ranking order and associated hyperparameters. To illustrate this further, we conducted an exhaustive search, testing the performance on all possible permutations of the informative layer order. Figure 2 demonstrates that the results can vary significantly.

Finally, K-Kmeans consistently outperforms all methods, exhibiting significant enhancements, particularly in the Cora and AUCS datasets.

As we have illustrated the significance of the layer ranking choice in both methods, we propose an automated approach to find a good order. To achieve this, we evaluate the modularity of the spectral clustering results on each single layer. Essentially, we evaluate the compactness of the formed clusters, providing a means to assess the informativeness of each layer even in the absence of true labels. Higher modularity values indicate more cohesive clusters with fewer connections to other clusters. If a layer generates more compact clusters, it is more meaningful to the clustering task.

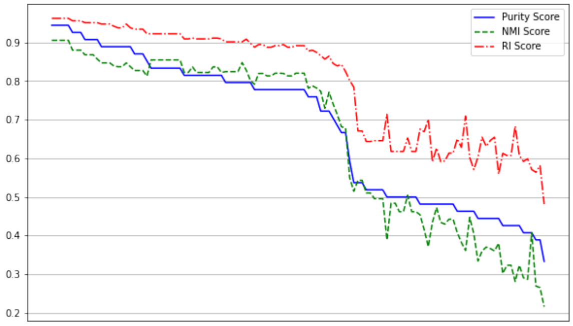


Figure 2: Analyzing the Influence of Informative Layer Order on SC_SR Method Performance with metrics (Purity, NMI, RI) evaluated on AUCS Dataset, Sorted for Non-Increasing Purity, with Informative Layer Order on the X-axis

To validate this approach, we examine the modularity results on the layers of the AUCS dataset. Table 1 reveals that layers such as Co-author and Lunch yield the highest modularity. This aligns with expectations, considering that individuals working in the same research group are likely to have more co-authorship and share lunch at work. Following this, we find Leisure and Work, where subjects are more likely to work in the same office and engage in activities outside the office, although they may not belong to the same research group. Finally, the least informative layer is the least sparse matrix, representing Facebook connections, as individuals are likely to connect with anyone within the same institute.

It is essential to emphasize that while this approach does not guarantee optimal results, we argue that it serves as a good approximation, particularly in the absence of domain expertise.

Table 1: Modularity of Layers Based on Clustering Results

Lunch	Facebook	Leisure	Work	Coauthor
0.599	0.272	0.474	0.400	0.662

4 LIMITATIONS AND EXTENSIONS

While the methods outlined in this paper offer an intuitive interpretation and are straightforward to implement, they are not without

Table 2: Performance Comparison of Single-layer and Multi-layer Graph Clustering Algorithms

Cora Dataset

	Single layer graph		Multi-layer graph				
	Titles	Citations	SC-SUM	SC-AL	K-Kmeans	SC-GED	SC-SR
Purity	0.609	0.521	0.523	0.526	**0.705**	0.560	0.584
NMI	0.237	0.005	0.009	0.018	**0.427**	0.122	0.238
RI	0.524	0.401	0.396	0.396	**0.640**	0.567	0.600

MIT Dataset

	Single layer graph		Multi-layer graph				
	Friendship	Calls	SC-SUM	SC-AL	K-Kmeans	SC-GED	SC-SR
Purity	0.578	0.500	0.478	0.478	**0.589**	0.578	0.433
NMI	**0.361**	0.181	0.222	0.220	0.302	0.313	0.202
RI	**0.682**	0.414	0.530	0.529	0.571	0.653	0.625

AUCS Dataset

	Single layer graph		Multi-layer graph				
	Lunch	Facebook	SC-SUM	SC-AL	K-Kmeans	SC-GED	SC-SR
Purity	0.870	0.500	0.833	0.833	**0.944**	0.889	**0.944**
NMI	0.849	0.413	0.857	0.857	**0.908**	0.880	0.905
RI	0.937	0.725	0.925	0.925	**0.966**	0.950	0.962

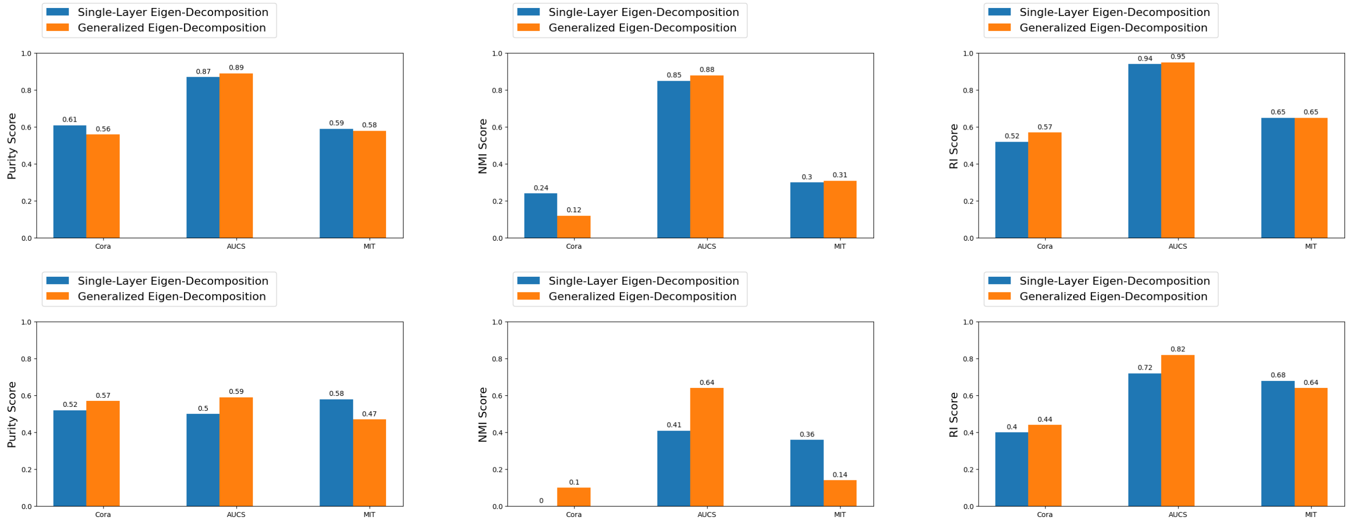


Figure 3: Comparison of metrics (Purity, NMI, RI) between Single Layer and Generalized Eigen Decomposition with varying initialization layer choices. The first row corresponds to selecting the most informative layer, while the second row corresponds to choosing a less informative layer.

limitations. In this section, we enumerate several challenges encountered during the reproduction of these methods and propose extensions to address them.

4.1 Informative Layer Dependency

The main challenge that we had while implementing [1] is the ordering of the layers according to their importance. This ordering not only initializes the **SC_SR** algorithm but also determines the most informative layer for **SC_GED**, significantly impacting the quality of clustering (see Figure 2). The choice of layer ordering is non-trivial and ideally requires expert knowledge. In our case, we were able to carry out exhaustive tests for all possible orders since we are dealing with small datasets with 3 and 5 layers. However, as the data size increases, the complexity grows in factorial time for larger datasets (with $d!$ possible orders for a graph with d layers).

A practical solution is to define selection criteria for layer ordering. One effective method that we propose is to perform one-layer clustering and order the layers according to their modularity scores. This approach is intuitive and valuable especially when we are not provided with true labels which is the typical clustering framework. Using the modularity score as a measure provides an automated way to determine a proper layer ordering, making the algorithms more robust for datasets without ground truth labels.

4.2 Performance vs Computational Time

As demonstrated in 2, we observed a slight improvement in results with SC-GED compared to general averaging methods like SC-SUM and SC-AL. However, due to the eigenvalue decompositions of each layer and matrix inversions, SC-GED requires higher computation complexity compared to the other methods. The difference in computational times can span three orders of magnitudes, particularly noticeable on large datasets.

4.3 Reproducibility

One major challenge encountered during the reproduction of the results is the absence of information concerning the preprocessing steps. For instance, in the Cora[2] dataset, the selection criteria for the chosen papers are not specified. Similarly, the reason behind the selection of nodes and target clusters in the MIT[5] dataset is unclear, as the number of nodes and target clusters differs from the specifications in the original dataset. Extracting the adjacency matrices also demands extensive data engineering, a process that can yield significantly varied results.

To address this, we have introduced a new dataset AUCS[6] with carefully constructed adjacency matrices and precise ground truth labels for benchmarking purposes.

4.4 Scalability

The SC-GED method introduced in [1] exhibits limitations in scalability when applied to large multi-layer graphs with numerous layers. Firstly, the method's complexity is $O(mn^3)$, where n represents the number of nodes and m signifies the number of layers. This complexity arises from the eigen-decomposition and matrix inversions involved. Secondly, the method aims to approximate joint eigenvectors through the solution of a large optimization problem. As the number of layers and nodes increases, finding common eigenvectors becomes more challenging and affects the method's performance. To conclude, this method can only be applied to small networks comprising approximately 100 nodes and 3 to 5 layers. However, real-life datasets are larger and more complex, rendering these methods impractical for addressing the scalability demands of such datasets.

5 CONCLUSION

In our work, we studied the two proposed methods for multi-layer graph clustering based on a spectral perspective of the graph: Clustering with Generalized Eigen-Decomposition (**SC-GED**) and Clustering with Spectral Regularization (**SC-SR**). Although the proposed approaches showed a slight performance improvement compared to existing aggregation methods, they still do not outperform certain baseline methods. Despite their intuitive interpretation, the effectiveness of these methods heavily depends on the choice of hyperparameters and often requires expert domain knowledge for selecting the order of informative layers. Additionally, they can be computationally intensive and are not scalable to real-life applications. Furthermore, we also highlight the importance of the quality of the extracted adjacency matrices in the success of these methods.

REFERENCES

[1] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering with multilayer graphs: A spectral perspective. *IEEE Transactions on Signal Processing*, 60(11):5820–5831, 2012.

[2] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *Proceedings of the National Academy of Sciences (PNAS)*, 106(36):15274–15278, 2009.

[3] L. Liu, Z. Kang, J. Ruan, and X. He. Multilayer graph contrastive clustering network. *Information Sciences*, 613:256–267, 2022.

[4] B. Luo, R. C. Wilson, and E. R. Hancock. Spectral clustering of graphs. In N. Petkov and M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, pages 540–548, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[5] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000. www.research.whizbang.com/data.

[6] L. Rossi and M. Magnani. Towards effective visual analytics on multiplex and multilayer networks. *Chaos, Solitons and Fractals*, 72:68–76, 2015.

[7] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 12 2007.