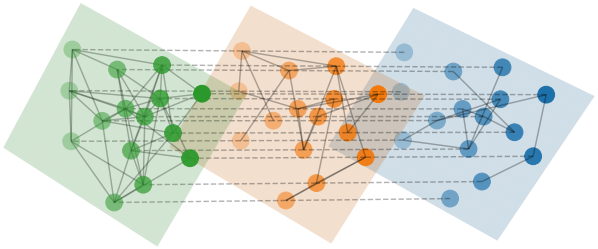
# Clustering with Multi-Layer Graphs: A Spectral Perspective

Capucine Garçon
capucine.garcon@ens-paris-saclay.fr
MVA - ENS Paris-Saclay

Sonia Mazelet
sonia.mazelet@ens-paris-saclay.fr
MVA - ENS Paris-Saclay



## ABSTRACT

The goal of the article [2] is to put forward clustering with Multi-layer Graphs that are more informative than single graphs. The two methods are leveraging on the "joint spectrum", one by using the eigen-decomposition of the Laplacian matrix, the other with graph regularization. In this work, we study these two algorithms and present their theoretical aspects as well as the numerical results we obtained by implementing them on real-world datasets. These algorithms outperform naive methods such as Kernel K-means clustering. Our experiments are available at https://github.com/CapucineGARCON/MVA2022-Geometric-Data-Analysis.git.

## KEYWORDS

multi-layer graphs, clustering, joint spectrum

## 1 INTRODUCTION

### 1.1 Context

A multi-layer graph is a group of graphs that share the same vertices, but have different edges. Multi-layer graphs are a natural way of representing multimodal data, such as social network data. For example, we constructed a multi-layer graph from the MIT Reality Mining dataset. It is a dataset constructed by analysing the phone signal of 90 people during a certain time. From this dataset, we made 3 layers that have the same nodes (each node is one person) but represent a particular aspect of the information collected. The adjacency matrices of these 3 layers are displayed in Figure 1. The first layer corresponds to the proximity between individuals: two individuals are linked if they were close to each other at some point during the time of the study. The second layer corresponds to friendships: two individuals are linked if they are friends. The third layer displays phone calls, it links individuals who have called each other during the time of the survey.

When doing clustering on multi-layer graphs, we want to capture the information every graph provides and combine it to improve the clustering we would have obtained if there was only one layer. We studied two different ways of doing multi-layer graph clustering that were introduced in [2]. The first one, presented in Section 2.2, is generalized eigen-decomposition which is a generalization of a well-known one-layer graph clustering algorithm. The second one, presented in Section 2.3, is a spectral regularization method that relies on the construction of a 'smooth' function on the different layers of the graphs. In Section 3, we present the experimentations we made on real-world datasets, and in Section 4 we investigate the limitations and extensions of the methods.

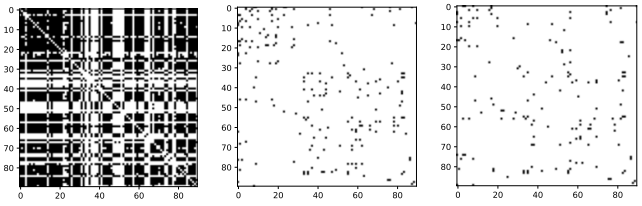


**Figure 1: Adjacency matrices of the three layers of the multi-layer graph obtained with the MIT Reality Mining dataset. Each black dot corresponds to a connection between individuals, following the criteria explained in Section 1.1. (Proximity, Calls, Friendship from left to right).**

### 1.2 Related work

The main contribution of [2] relies on the different roles given to each layer when combining them. While state-of-the-art algorithms, such as Co-Regularization and Kernel K-means algorithms, treat the layers equally, they manage to combine layers based on their importance. We will see in Section 3.1 how we can quantitatively measure the importance of a layer. There are other methods for combining the information given by several graphs, as in [4] where the authors model the clustering as a mixture of random walks on the different layers. In [1], they use an average of the Laplacian matrices of all the graphs. Both these methods also treat the layers equally, instead of considering their relative importance.

# 2 MAIN CONTENT OF THE PAPER

## 2.1 Spectral clustering on graphs

The spectral clustering algorithm, studied in [3], is a popular algorithm for clustering on graphs that relies on the analysis of the spectrum of the Laplacian matrix $L$ of the graph. The Laplacian $L$ is defined as $L = D - W$ where $D$ is the degree matrix and $W$ is the adjacency matrix of the graph. When searching for $k$ clusters, the method consists in finding the $k$ smallest eigenvalues of $L$ and their associated eigenvectors $u_1, ..., u_k$, and in constructing the matrix $U$ containing $u_1, ..., u_k$ in columns. Each row of $U$ is associated with the vertex of the same number, and the clustering can be done using the K-means algorithm. This method amounts to projecting the vertices of the graph into the low-dimensional spectral domain, where clustering is easier to perform. This algorithm is very popular and widely used because it is efficient and easy to implement. The main objective of [2] is to generalize this idea to multi-layer graphs.

## 2.2 Clustering with generalized eigen-decomposition

The spectral clustering algorithm cannot be used on multi-layer graphs, because instead of one Laplacian matrix, multi-layer graphs have several Laplacian matrices (one for each layer). The idea of clustering with generalized eigen-decomposition is to construct a joint spectrum for the multi-layer graph, consisting of eigenvectors that are as close as possible to the eigenvectors of all the different Laplacians. Formally, we denote by $\mathcal{G}$ a multi-layer graph, and by $M$ its number of layers. We consider the normalized Laplacian $L_{rw}$ of each graph:

$$L_{rw} = D^{-\frac{1}{2}}(D - W)D^{\frac{-1}{2}}.$$

For $i \in [1, ..., M]$, denote $L_{rw}^{(i)}$ the normalized Laplacian of the $i^{th}$ graph, and $\Lambda^{(i)}$ the diagonal matrix containing its eigeinvalues. The goal is to find $P$ such that

$$L_{rw}^{(i)} \approx P\Lambda^{(i)}P^{-1}.$$

To do so, we solve the following optimization problem.

$$\arg\min_{P,Q} S = \frac{1}{2}\sum_{i=1}^{M} ||L_{rw}^{i} - P\Lambda^{i}Q||_F^2 + \frac{\alpha}{2}(||P||_F^2 + ||Q||_F^2) + \frac{\beta}{2}||PQ - I_n||_F^2.$$

The first term on the right-hand side of the equation ensures that $P$ and $Q$ are close to the matrix of eigenvectors of $L_{rw}$ for each layer, the second term is a regularisation term for $P$ and $Q$ which ensures that they are not too large, and the third term ensures that $Q$ is close to the inverse of $P$. The columns of the resulting matrix $P$ play the role of the set of joint eigenvectors of the multi-graph. As for the classical spectral clustering algorithm presented in the section 2.1, performing K-means on its rows (each row representing a vertex), allows obtaining a clustering of the vertices.

## 2.3 Clustering with spectral regularization

The second method takes into account the relative importance of the different layers of the multi-graph. Indeed, some layers may provide more relevant information for the problem to be solved and should therefore have a greater impact on the clustering. Clustering with spectral regularisation is based on the idea that we can construct a function that maps nodes to a one-dimensional space, without losing their proximity.

We first consider the one-layer case. Denote $V$ the set of nodes. We want to construct $f : V \to \mathbb{R}$ that verifies, for every nodes $v_i$ and $v_j \in V$, $f(v_i) - f(v_j)$ is small if $v_i$ and $v_j$ are connected. This kind of function is called a 'smooth' function on the graph. A smooth function corresponds to a solution to the optimization problem

$$\arg\min_{f} \sum_{i=1}^{n} \omega_{i,j}(f(v_i) - f(v_j))^2,$$

where $\omega_{i,j}$ is 1 if $v_i$ and $v_j$ are connected and 0 if not.
This problem is equivalent to

$$\arg\min_{f} f^T L f \quad \text{st} \quad ||f|| = 1, f \perp 1.$$

Since the eigenvalues $u_i$ of $L$ verify $u_i L u_i^T = \lambda_i$ where $\lambda_i$ are the associated eigenvalues, if the eigenvalues are small, the eigenvectors are good candidates for $f$.
We use this idea to find a function $f$ that is smooth on all layers of the multi-graph, in order to find $k$ clusters. We order the graphs according to their importance, $\mathcal{G}_0$ being the graph carrying the most information. We consider $(u_i)$ its eigenvectors and solve the minimization problem

$$\arg\min_{f} \frac{1}{2}||f_i - u_i||^2 + \lambda f_i^T L_{rw}^{(2)} f_i,$$

where $L_{rw}^{(2)}$ is the normalized Laplacian of the second graph.
The first term guarantees that $f$ is close to the eigenvectors of $\mathcal{G}_0$, which means it will be smooth on $\mathcal{G}_0$. The second term ensures that $f$ is smooth on $\mathcal{G}_2$. The parameter $\lambda$ characterizes the trade-off between these two aspects. We iterate on all the multi-graph layers to find the final $f$.
The final $f$ plays the role of a joint spectrum on which we can perform K-means clustering, as in the generalized eigen-decomposition method (see Section 2.2).

# 3 OUR EXPERIMENTS

In this section, we present the datasets we used, the experiments we conducted and finally, we compare the performances of the different algorithms in terms of two clustering metrics.

## 3.1 Datasets and clustering algorithms

We used two real-world datasets. The first is the Cora dataset, which is composed of titles, abstracts, and citations of several research articles in different fields. We randomly selected 40 articles and constructed a two-layer graph (see its adjacency matrices in Figure 2): one layer consists of titles and abstracts, which we refer to as words in the following, and the other consists of citations. The aim is to classify them in their research area. We chose three topics: Neural networks, probabilistic methods, and genetic algorithms. The word graph was constructed as follows. We used a vocabulary and constructed a histogram for each article, which indicates whether

or not a word is present in the title and abstract of the article. We then performed a cosine similarity between each histogram, which gave a number between 0 and 1 for each pair of articles. We decided to transform this weighted graph into an unweighted graph in the following way. If the cosine similarity between two graphs was greater than 0.1, we considered them to be linked in the graph (we experimentally determined this threshold to obtain an informative graph). This led to better results. With this method, we assumed that articles that are in the same research area have similar words in their title or abstract. The citation graph was constructed by linking two articles if one of them cited the other.

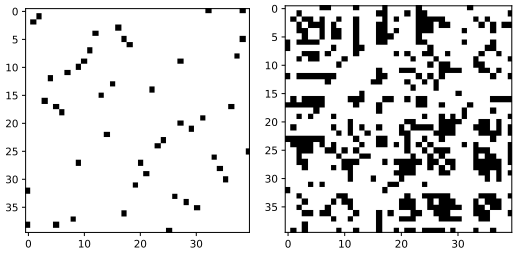


**Figure 2: Adjacency matrices of the two-layer graph built from the Cora dataset: the citation layer (left) and the words layer (right). Black dots correspond to links.**

The second is the MIT Reality Mining Dataset. It contains the results of a study conducted in 2004-2005 at the MIT Media Laboratory, on the phone activity of 94 different people. It is difficult to perform clustering on this dataset due to the lack of reference clusters. We decided to consider that clusters are made up of individuals who have the same job title, such as 'professor', 'staff', or 'sloan', which gave us 6 clusters. The dataset was difficult to analyse and a lot of information was missing, so we could only consider 90 people instead of 94. With this dataset, we constructed a three-layer graph (see Figure 1.1), which is detailed in Section 1.1.

We implemented the Normalized Spectral algorithm for one-layer graphs, and the two algorithms developed in [2], Clustering with Generalized Eigen Decomposition the Clustering with Spectral Regularization. For these two multi-layer clustering algorithms, we need to know the most informative layer. Indeed, for the algorithm based on eigen-decomposition, it is suggested to initialize P and Q with the eigenvectors of the most informative layer Laplacian matrix. The second algorithm is using a recursive approach by selecting and then combining layers based on their mutual information after considering the most informative layer to build the first low-dimensional embedding $U$. Therefore, it is necessary to order layers from the most informative to the least ones. For this purpose, we have implemented a function that is able to rank the layers by using the Normalized Mutual Information (NMI) criteria. The most informative layer is the one with the highest mutual information with each of the other layers. Then, we rank the layers based on their NMI with the first one.

Finally, we have also implemented the Kernel K-means algorithm as a baseline for the two novel methods.

## 3.2 Evaluation criteria and results

To evaluate our algorithms, we used the Purity criterion and the Normalized Mutual Information criterion (NMI).

These criteria aim at comparing the clusters $\Omega = \{\omega_1, ..., \omega_k\}$ we computed with the algorithms, with the ground truth clusters $C = \{c_1, ..., c_k\}$. Purity is defined as:

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

where $N$ is the number of nodes, and $\omega_k \cup c_j$ is the number of common nodes in the clusters $\omega_k$ and $c_j$.
Purity can be considered as the measure of the extent to which the computed clusters contain only one class.
NMI is defined as:

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{(H(\Omega) + H(C))/2},$$

where $I$ is the mutual information and $H$ the entropy.
The NMI measures the amount of common information between the calculated clusters and the real clusters.

**Table 1: Performance evaluation of our experiments on CORA dataset**

| | Single layer graph | | Multi-layer graph | | |
|---|---|---|---|---|---|
| | Words | Cites | SC-GED | SC-SR | K-Kmeans |
| Purity | 0.70 | **0.75** | 0.70 | 0.70 | 0.60 |
| NMI | 0.23 | **0.30** | 0.27 | 0.23 | 0.05 |

**Table 2: Performance evaluation of our experiments on MIT dataset**

| | Single layer graph | | | Multi-layer graph | | |
|---|---|---|---|---|---|---|
| | Friendship | Proximity | Calls | SC-GED | SC-SR | K-Kmeans |
| Purity | 0.42 | 0.41 | 0.37 | 0.43 | 0.51 | **0.61** |
| NMI | 0.20 | 0.08 | 0.11 | 0.13 | 0.22 | **0.37** |

Table 1 shows the performances of different clustering algorithms on the CORA dataset. By looking at the adjacency matrices of the two layers, we can say that the layer Citations gives us a better indication of the relation between articles because there are fewer relations between the articles. Visually, the layer Words is more informative, therefore, we expect to get better results when performing Normalized Spectral Clustering on this layer. However, Table 1 shows that we obtain 0.75 for the layer Cites and 0.70 for the layer Words. We can say that in a certain sense, the layer is more discriminating.

For the multi-layer clustering, the results displayed in Table 1 were obtained with the layer Words as the most informative to initialize P and Q in the optimization loop for the **SC-GED** algorithm and to initialize the low-dimensional embedding in the **SC-SR** algorithm. This layer contains more information but is not the one that performs better one-layer clustering. On the contrary, when

considering the layer Citations as the first one, we obtain 0.65, 0.60 purity, respectively for the algorithms **SC-GED** and **SC-SR**.

As explained by the authors [2] the adjacency matrices with fewer connections are better indicators of the relationship between the people i.e. the nodes in our graphs. This is why the layer friendship in Figure 1 is a better indicator of the relationship between people. But, due to the sparsity of the matrix, this layer is not sufficient to obtain good performances in clustering. However, the proximity matrix is very informative because it consists of more connected components. The function we build to rank the layer is giving us the following ranking (from the most informative layer to the less one): Calls, Friendship, and Proximity. This ranking is not the optimal one because it gives poorer results than the following combination: Proximity, Calls, Friendship. The performance of this combination supports the authors' explanations about the informative layers. Indeed, this combination orders the layers from the one containing the most information (i.e. more connections) to the one containing the least. The results for this combination are displayed in Table 2. Multi-layer clustering methods are performing slightly better than single-layer clustering. As expected **SC-SR** is showing better results than **SC-GED** but Kernel K-means outperforms the two novel methods.

Finally, **SC-GED** is performing better than **SC-SR** for Cora dataset, but the authors show the contrary. Because it better captures the specificity of each layer, the **SC-SR** algorithm is supposed to perform better. We must keep in mind that we have not used the same layers as the authors, therefore not the same data, and especially we have considered less data for computational efficiency. Indeed, the **SC-GED** algorithm is performing eigen-decomposition for each layer and then an optimization loop is necessary. Whereas, **SC-SR** is very rapid and computationally efficient because only one eigen-decomposition is used.

## 4 LIMITATIONS AND EXTENSIONS

The main difficulty we have encountered is determining the importance between layers. While having the mutual information between two layers is quite obvious and easy to calculate, determining which is the most informative one is not that easy. As explained above, we have considered that the most informative layer is the one with the highest mutual information with each of the other layers. However, for graphs with only two layers, the rule we have chosen to determine the most informative layer is not applicable. Our method to rank the layers is not performing as desired, the ranking it proposes for the MIT dataset does not give the best results. One can explain it with the Normalized Mutual Information which gives strange results either to evaluate the clustering algorithm or to evaluate the result between the one-layer clustering of two different layers. The authors could have provided a more detailed approach to identify the most informative layer.

Also, for the Clustering with Spectral Regularization algorithm, the authors [2] specifies that it is necessary to recursively combine the layers with the most important mutual information. We find this criterion can be counter-intuitive. Indeed, if the most informative layer is first considered, it is better to take, as second layer for the combination, one layer that catches different relations between the nodes. As the first layer contains a lot of information about the relationships, the second one allows for obtaining additional information.

We could continue our work in many directions. First of all, many parameters play a role in the algorithm, such as $\lambda$ in the spectral regularization algorithm, and $\alpha, \beta$ in the clustering algorithm with generalized eigen-decomposition. We could study their impact on the results, and find the optimal ones for each dataset. We also tested the algorithm on multi-layer graphs containing only a few layers. We could try the algorithm on larger datasets, and see how the algorithm performs. There are two possible outcomes. Increasing the number of layers could produce an averaging phenomenon and smooth out the specificities of each layer, which would decrease the performance. It could also give better results since the algorithm has more information. Furthermore, due to computational time, we were only able to select 40 articles from the Cora dataset out of the 2708 scientific publications it contains. Improving our algorithm to make it faster would allow us to test it on larger datasets.

## 5 CONCLUSION

We have studied two spectral-based algorithms to perform multi-layer graphs clustering. Both are inspired by the spectral clustering algorithm on simple (one-layer) graphs. The first one is the clustering with generalized eigen-decomposition algorithm, which uses a joint spectrum close to all the spectra of the Laplacians of the different layers. The second is the clustering with spectral regularisation algorithm, which is based on the calculation of a smooth function on the graphs. The novelty of these two algorithms, and of the second one in particular, is the fact that they process the layers in the order of their importance. This makes it possible to process the information given by each layer according to its relevance, which gives better results than previous methods. We presented the theoretical aspects of both algorithms and tested them on two real-world datasets. We find that the results of multi-layer clustering were not significantly better than those obtained using a single layer, and that **SC-GED** and **SC-CR** give similar results, even though **SC-CR** is way faster. But, we have to keep in mind that due to the computational time and the characteristics of the datasets, it was difficult to test under the same conditions as in [2]. Nevertheless, we have shown that multi-layer clustering works well on real-world datasets.

## REFERENCES

[1] Andreas Argyriou, Mark Herbster, and Massimiliano Pontil. 2005. Combining graph Laplacians for semi–supervised learning. *Advances in Neural Information Processing Systems* 18 (2005).
[2] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. 2012. Clustering with multi-layer graphs: A spectral perspective. *IEEE Transactions on Signal Processing* 60, 11, 5820–5831.
[3] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22, 8 (2000), 888–905.
[4] Dengyong Zhou and Christopher JC Burges. 2007. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine learning*. 1159–1166.