Robust spectral clustering using LASSO regularization

Champion Camille[1], Blazre Mlanie[1], Burcelin Rmy[2], Loubes Jean-Michel[1], Risser Laurent[3]
[1] Toulouse Mathematics Institute (UMR 5219)
University of Toulouse F-31062 Toulouse, France
[2] Metabolic and Cardiovascular Diseases Institute (UMR 1048)
University of Toulouse F-31432 Toulouse, France
[3] Toulouse Mathematics Institute (UMR 5219)
CNRS F-31062 Toulouse, France

Abstract

Cluster structure detection is a fundamental task for the analysis of graphs, in order to understand and to visualize their functional characteristics. Among the different cluster structure detection methods, spectral clustering is currently one of the most widely used due to its speed and simplicity. Yet, there are few theoretical guarantee to recover the underlying partitions of the graph for general models. This paper therefore presents a variant of spectral clustering, called ℓ_1-spectral clustering, performed on a new random model closely related to stochastic block model. Its goal is to promote a sparse eigenbasis solution of a ℓ_1 minimization problem revealing the natural structure of the graph. The effectiveness and the robustness to small noise perturbations of our technique is confirmed through a collection of simulated and real data examples.

Keywords: Spectral clustering, community detection, eigenvectors basis, ℓ_1-penalty.

1 Introduction

Graphs play a central role in complex systems as they can conveniently model interactions between the variables of a system. Finding variable sets with similar attributes can then help understanding the mechanisms underlying a complex system. Graphs are commonly used in a wide range of applications, ranging from Mathematics (graph theory) to Physics [12], Social Networks [10], Informatics [27] or Biology [14, 23]. For instance, in genetics, groups of genes with high interactions are likely to be involved in a same function that drives a specific biological process.

One of the most relevant features when analyzing graphs is cluster structures. Clusters are generally defined as connected subsets of nodes that are more densely connected to each other than to the rest of the graph. Different strategies make it possible to define more specifically variable clusters depending on whether this property of vertices is considered locally (on a connected subset of vertices) or globally (on the whole network). First, cliques (subset of vertices such that every two distinct vertices in the clique are adjacent)[37], n-clique (maximal subgraph such that the distance of each pair of its vertices is not larger than n) [37] and k-core (maximal connected subgraph of G in which all vertices have degree at least k) [33] characterize local cluster structure. Secondly, one of the global cluster structure definition is based on the notion of modularity [24, 25] that quantifies the extent to which the fraction of the edges that fall within the given groups differs from the expected fraction if edges were distributed at random. The most popular random model is proposed by [24], where edges are reconnected randomly, under the constraint that the expected degree of each vertex corresponds to the degree of the vertex in the original graph. The last definition of cluster structure, and the most natural is related to similarity between each pair of vertices, that includes local or global definitions of a cluster structure. It is really natural to assume that cluster structures are groups of vertices that are close to each other. Similarity measures are the foundations of traditional methods as detailed below. These include traditional distance measures such as Manhattan or Euclidean distances or computing correlations between rows of the adjacency matrix or random walk based similarities [29].

Once the definition of cluster structure is fixed, it is crucial to build efficient procedures and algorithms for the identification of such structures in the network. The ability to find and to analyze such groups can provide an invaluable help in understanding and visualizing the functional components of the whole graph [9, 24]. Classical techniques for data clustering, like hierarchical clustering, partitioning clustering and spectral clustering, detailed below, are sometimes adopted for graph clustering too. Hierarchical clustering [11] builds a hierarchy of nested clusters organized as a tree. partitioning clustering [30] decomposes the graph into a set of disjoint clusters. Given N variables/nodes, it builds k partitions of the data by satisfying: (i) each group contains at least one point (ii) each point belongs to exactly one group. In recent years, spectral clustering has become one of the most widely used methods due to its speed and simplicity [22, 4, 26, 7]. This method extracts the geometry and local information of the dataset by computing the top or bottom eigenvectors of specially constructed matrices. The observations are projected into this eigenspace to reduce the dimensionality of the problem and k-means procedure is then applied in an easier subspace to detect clusters.

k-means, that belongs to partitional clustering methods, aims to find a set of k cluster centers of a dataset such that the sum squared of distances of each point to its closest cluster center is minimized. Lloyds 1957 procedure [21] remains one of the widely used because of its speed and simplicity. It has been studied for several decades [21, 38] and many versions of this technique has recently been developed. [39] proposed alternatives to Lloyds algorithm that preserves its simplicity, makes it more robust to initialization and relieves its tendency to get trapped by local minima. [18] developed a new variant of k-means++ seeding algorithm [1] to achieve a constant approximation guarantee.

Our contribution. Observed real networks differ from random graphs from their edge distribution and from their underlying structures. Erds Renyi random graphs models [5], where all the pairs of nodes have equal probability of being connected by an edge, independently of all other pairs fail to model real observed graphs. Additionnally, stochastic block models are not always relevant to infer their structures. To remedy this problem, we developed a new random model, closely related to stochastic block model, but better suited to model graphs that have been inferred from the observations. In practice, graphs that are studied are not known beforehand but often estimated.To achieve a good clustering recovery, random graph models are often associated to their similarity matrix to maintain the clustering structure of the graph. [36] developed a model to learn a doubly stochastic matrix which encodes the probability of each pair of data points to be connected, used to normalize the affinity matrix such that the data graph is more suitable for clustering tasks. [28] has shown that for a wide class of graphs, spectral clustering gives a good approximation of the optimal cluster. In our model, we assume that a group does not emerge by chance but because there exists an underlying structure. This randomized version of the deterministic graph with exact cluster structure, is used to check whether it displays the original cluster structure. [35, 19, 32] proved consistency of spectral clustering applied to stochastic block models for some specific adjacency type matrices. Even if the consistency of spectral clustering has been proved for stochastic block models, there is no convergence guarantee for general models. Thus, k-means can fail to reach the true underlying partitions of the graph. Moreover, spectral clustering technique fails to recover the original clusters when it comes to a higher randomization coefficient. This is mostly due to the computed eigenbasis that is not equally informative.

In order to tackle this issue, we develop an alternative method to the spectral clustering that promote a sparse eigenvectors basis solution of an ℓ_0 optimization problem, corresponding to the indicator vectors of each cluster. Since the natural constrained ℓ_0 is a NP-hard problem, it was then replaced by its convex relaxation ℓ_1 [31]. Actually we can show that the solution of the ℓ_0 optimization problem is still the same when replacing the ℓ_0-norm by the ℓ_1-norm if we add a constraint on the maximum of the coefficients. Hence, the algorithm turns out to solve an ℓ_1-penalty optimization problem that is feasible and easy to implement, even for very large graphs. In a wider scope, research papers have explored differently regularized spectral clustering to robustly identify clusters in large networks. Although [40] and [15] show the effect of regularization on spectral clustering through graph conductance and respectively through stochastic block models. Equally, [17], shows on a simple block model that the spectral regularization separates the underlying blocks of the graph.

In this paper, we introduce in Section 2 a new random graph model, used to solve spectral clustering (Section 3) and its new variant (Section 4) objective function. We prove the efficiency and accuracy of the variant algorithm in Section 5 through experiments on simulated and real medical dataset (Section 6).

2 New random graph model

2.1 Notations

This work considers the framework of an unweighted undirected graph $G(V, E)$ with no self-loops consisting of vertices $V = \{1, \dots, n\}$ and p edges connecting each pair of vertices. An edge $e \in E$ that connects a node i and a node j is denoted by $e = (i, j)$. In this paper, we consider that the existence of a link between two nodes in an interaction network is already inferred from the estimation of a statistical dependance measure. The graph G is represented hereusing an adjacency matrix $A = (A_{ij})_{(i,j) \in V^2}$ defined by

$$A_{i,j} = \begin{cases} 1 & \text{if there is an edge between } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

Since the graph is undirected and with no self-loops, $A \in \mathbb{M}_n(\mathbb{R})$ is a symmetric matrix with coefficients zero on the diagonal.

For each node i, the degree d_i is defined as the number of edges incident to i and is equal to : $d_i = \sum_{j=1}^{n} A_{ij}$. We denote by D the diagonal degree matrix containing $(d_1, \dots, d_n)$ on the diagonal and zero elsewhere.

A subset $C \in V$ of a graph is said to be connected if any two vertices in C are connected by a path in C. Non empty sets $C_1, \dots, C_k$ form a partition of the graph $G(V, E)$ if $C_i \cap C_j = \emptyset$ and $C_1 \cup \cdots \cup C_k = V$. In addition, C_i are called connected components if there are no connections between vertices in C_i and $\overline{C_i}$ for all i in $\{1, \dots, k\}$.

We define the indicators of connected components $\mathbf{1}_{C_i}$ whose entries are defined by:

$$(\mathbf{1}_{C_i})_j = \begin{cases} 1 & \text{if vertex } j \text{ belong to } C_i, \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Graph models

As mentioned in Section 1, random graph models, in general, are not always relevant to represent the structure of a graph that has been inferred from observations. To tackle this issue, we create a new random model with an underlying structure that is a randomized version of a deterministic graph with exact cluster structure.

2.2.1 Ideal model

We consider that the graph $G_*(V, E)$ is the union of k complete graphs that are disconnected from each other. We denote by $C_1, \dots, C_k$ the k connected components of the graph, that match the k clusters. We allow the number of vertices in each subgraph to be different. We denote by $c_1, \cdots, c_k$ (≥ 2) their respective size ($\sum_{i=1}^{k} c_i = n$). To simplify, we assume that the nodes, labeled from 1 to n, are ordered with respect to their block membership and in increasing order with respect to the size of the blocks.

From a matricial point of view, the associated adjacency matrix A_* is a k-block diagonal matrix of size n of the form:

$$A_* = \begin{bmatrix} C_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & C_k \end{bmatrix}$$

where $C_1, \cdots, C_k$ are symmetric matrices of size $c_1 \times c_1, \cdots, c_k \times c_k$.

2.2.2 Perturbed model

The reality is that we consider the graph G_* but we observe a randomized version of this graph, denoted by $\tilde{G}$.

We introduce the ErdsRnyi model of a graph [5, 34], one of the oldest and best studied random graph model.

Given a set of n vertices, we consider the variable X_{ij} that indicates the presence/absence of an edge between vertices i and j. Then, for all $\{X_{ij}\}$ *i.i.d.*, we have $X_{ij} \sim B(p)$. Some edges have been added between the clusters and others have been removed within the clusters independently with respect to the same probability p. The adjacency matrix B of the Erdos-Renyi graph of size n, whose upper entries are realizations of independent Bernoulli variables, can be written as

$$\begin{cases} B_{ij} & \sim & \mathcal{B}(p) \;\; i.i.d, \; i < j \\ B_{ii} & = & 0 \\ B_{ij} & = & Bji \end{cases}$$

The graph $\tilde{G}$ of the new model, is derived from a deterministic graph with an exact cluster structure, whose edges have been disturbed Erds Renyi random graph. For instance, this perturbation may arise because of a partial knowledge of the graph.

Let $\tilde{A}$ be the adjacency matrix associated to $\tilde{G}$. $\tilde{A}$ is defined as follows:

$$\tilde{A} = A_* \overset{2}{\bigoplus} B$$

where $\tilde{A}_{ij} = A_{ij*} + B_{ij}$ [2], namely,

$$\tilde{A}_{ij} = \begin{cases} 0 \quad \text{if } B_{ij} = 1 \text{ and } A_{ij} = 1 \\ \quad \text{or } B_{ij} = 0 \text{ and } A_{ij} = 0, \\ 1 \quad \text{if } B_{ij} = 1 \text{ and } A_{ij} = 0 \\ \quad \text{or } B_{ij} = 0 \text{ and } A_{ij} = 1. \end{cases}$$

3 Graph clustering through spectral clustering

3.1 Spectral clustering algorithm

Looking first at the ideal graph model, let A_* and D_* the adjacency and degree matrices associated to the graph G_*.

Spectral clustering algorithm is based on graph Laplacian matrices. Among them, three different variants are used:

- the Unormalized Laplacian:

$$L_* = D_* - A_*,$$

- the Symmetric Laplacian:

$$L_{sym*} = D_*^{-\frac{1}{2}} A_* D_*^{-\frac{1}{2}},$$

- the Random Walk Laplacian:

$$L_{rw*} = I - D_*^{-1} A_*.$$

The original spectral clustering method has been proposed by [22] to cluster the nodes of the graph into k connected components. The idea behind spectral clustering is to use the first k eigenvectors (corresponding the k smallest eigenvalues) a normalized or unormalized version of the Laplacian matrix (derived from the adjacency one) to recover the connected components of the graph. If these matrices are so appealing in graph clustering, it is because of the following proposition:

Proposition 1 *(Number of connected components): The multiplicity k of the eigenvalue 0 of L_* and L_{rw*} and the multiplicity of the generalized eigenvalue 0 of L_{sym*} are equal the number of connected components $C_1, \ldots, C_k$ in the graph. For L_* and L_{rw*}, the eigenspace associated to 0 is spanned by the indicators of connected components $\{1_{C_i}\}_{1 \le i \le n}$. For L_{sym*}, the eigenspace associated to 0 is spanned by $\left\{D_*^{1/2} 1_{C_i}\right\}_{1 \le i \le n}$.*

We deduce from **Proposition 1** that a particular basis of the associated eigenspace is spanned by the connected components indicators. In addition, the rows of the matrix resulting from the concatenation of the k first eigenvectors, are equal for indices corresponding to nodes in the same component. Therefore, it is natural to apply k-means to these rows to provide, by the same way the blocks. Moreover, as the graph is made of exactly k connected components, the computation of the eigenvectors of L_*, L_{sym*}, L_{rw*} enables to recover these components.

3.2 Limits

Secondly, we consider the perturbed version $\tilde{G}$ of the graph G_*. Thus, $\tilde{G}$ is no longer made of connected components, but of densely connected subgraphs that are sparsely connected to each other. These densely connected subgraphs represent somehow a perturbed version of the initial connected components that form the clusters. As our model is closely related to stochatic block models and if the perturbation is not too high, we can still hope that the rows of the k concatenated eigenvectors are still closed for indices corresponding to nodes in the same cluster. But there is no theoritical guarantee that it still contains enough information on the graph structure to detect these clusters using k-means procedure.

To overcome this issue, we developed an alternative to the standard spectral clustering, called ℓ_1-spectral clustering, that aims at finding the k underlying connected components of a graph G_* with an exact cluster structure from its perturbed version $\tilde{G}$.

4 ℓ_1-spectral clustering, a new method for connected component detection

To ensure a good recovery of the connected components, the way the eigenvectors basis is built is of the highest importance. The key is to replace the k-means procedure by the selection of relevant eigenvectors that provide useful information about the structure of the data. ℓ_1-spectral clustering focused on the graph adjacency matrix instead of the Laplacian matrix or its normalized version, and its good properties. The idea remains the same if we replace the adjacency matrix by the Laplacian or its normalized version as proved by [35, 32].

We consider the ideal adjacency matrix A_* associated to the graph $G_*(V, E)$, we assume in what follows that the eigenvalues of A_* are sorted increasing order. And the same goes for the associated eigenvectors.

The indicators $\{\mathbf{1}_{C_i}\}_{n-k+1 \leq i \leq n}$ of the connected components $C_{n-k+1}, \dots, C_n$ are the eigenvectors associated this time to the largest eigenvalues. Theses eigenvalues are equal to the degree coefficients of the connected components $d_{n-k+1}, \dots, d_n$. The k first eigenvectors of A_* (associated to the k largest eigenvalues) are thus denoted $v_{n-k+1}, \dots, v_n$. Let $V_{1,k}$ the matrix that contains $v_{n-k+1}, \cdots, v_n$ in columns and by V_2, n the one that contains $v_1, \dots, v_{n-k}$. We define $\mathcal{V}^0_{1,k} = \mathrm{Span}\{v_{n-k+1}, \dots, v_n\}$.

Unlike the traditional spectral clustering method, ℓ_1-spectral clustering does not directly use the subspace spanned by the eigenvectors associated to the largest eigenvalues to recover the connected components but computes another eigenbasis that promotes sparse solutions for the eigenvectors.

4.1 General ℓ_0 minimization problem

Proposition 2 and 3 below show that the connected components indicators are solution of some specific problem.

Proposition 2 *The minimization problem* $(\mathcal{P}_0)$

$$\underset{v \in \mathcal{V}^0_{1,k} \setminus \{0\}}{\arg\min} \|v\|_0$$

has a unique solution (up to a constant) given by $\mathbf{1}_{C_{n-k+1}}$.

In other words, $\mathbf{1}_{C_{n-k+1}}$ is the sparsest non-zero eigenvector in the space spanned by the eigenvectors associated to the k largest eigenvectors.

Proof *We recall that* $\|v\|_0 = |\{j : v_j \neq 0\}|$. *Let* $v \in \mathcal{V}^0_{1,k} \setminus \{0\}$. *Therefore, as* $\mathbf{1}_{C_{n-k+1}} \in \mathcal{V}^0_{1,k}$, v *can be decomposed as* $v = \sum_{j=n-k+1}^{n} \alpha_j \mathbf{1}_{C_j}$ *where* $\alpha = (\alpha_{n-k+1}, \dots, \alpha_n) \in \mathbb{R}^k$ *and* $\exists j,\ \alpha_j \neq 0$.

The connected components of sizes $c_{n-k+1}, \ldots, c_n$ are sorted in increasing order of size. Therefore, by Proposition 1, $\|v\|_0 = \mathbf{1}_{\alpha_{n-k+1}\neq 0}c_{n-k+1} + \cdots + \mathbf{1}_{\alpha_n\neq 0}c_n$. The solution of ($\mathcal{P}_0$) is given by the vector in $\mathcal{V}_{1,k}\backslash\{0\}$ with the smallest ℓ_0-norm such that $\alpha = (\alpha_{n-k+1}, 0, \ldots, 0)$ where $\alpha_{n-k+1} \neq 0$.

We can generalize Proposition 2 to find, iteratively and with sparsity constraint, the other following indicators of connected components.

For $i = n-k+2, \ldots, n$, let $\mathcal{V}^i_{1,k} = \{v \in \mathcal{V}_{1,k} : v \perp \mathbf{1}_{C_l}, l = n-k+1, \ldots, i-1\}$.

Proposition 3 *The minimization problem ($\mathcal{P}_i$)*

$$\underset{v\in\mathcal{V}^i_{1,k}\backslash\{0\}}{\arg\min}\ \|v\|_0$$

has a unique solution (up to a constant) given by $\mathbf{1}_{C_i}$.

Solving ($\mathcal{P}_0$) (Proposition 2) is a NP-hard problem. In order to tackle this issue, we replace the ℓ_0-norm by its convex relaxation ℓ_1-norm. We can show that the solution of the ℓ_0 optimization problem is still the same by replacing the ℓ_0-norm by the ℓ_1-norm, if we add the constraint on the maximum of the coefficients.

4.2 General ℓ_1 minimization problem to promote sparsity

In addition to the number of connected components, we assume that we know one representative of each component i.e. a node belonging to this component. This assumption is not so restrictive compared to traditional spectral clustering where the number of clusters is assumed to be known. If we do not exactly know a representative for each component, we can estimate them by first applying a rough partitioning algorithm or just an algorithm that aims to find hubs of very densely connected parts of the graph.

Let $I_{n-k+1}, \ldots, I_n$ be the row indices of the representative element of each component and let $\tilde{\mathcal{V}}^1_{1,k} = \{v \in \mathcal{V}^0_{1,k} : v_{I_j} = 1\}$ for all $j \in \{n-k+1, \ldots, n\}$. This is straightfoward to see that the indicator vector of the smallest component is solution of the following optimization problem.

Proposition 4 *The minimization problem ($\mathcal{P}_1$)*

$$\underset{v\in\tilde{\mathcal{V}}^1_{1,k}}{\arg\min}\|v\|_1$$

has a unique solution given by $\mathbf{1}_{C_{n-k+1}}$.

Proof *We recall that $\|v\|_1 = \sum\limits_{i=1}^{n}|v_i|$. Let $v \in \tilde{\mathcal{V}}^1_k$. Therefore, as $\mathbf{1}_{C_{n-k+1}} \in \mathcal{V}^0_{1,k}$, v can be decomposed as $v = \sum\limits_{j=n-k+1}^{n} \alpha_j \mathbf{1}_{C_j}$ where $\alpha = (\alpha_{n-k+1}, \ldots, \alpha_n) \in \mathbb{R}^k$ and there exists $j \in \{n-k+1, \ldots, n\}$, $\alpha_j \neq 0$.*

The connected components of sizes $c_{n-k+1}, \ldots, c_n$ are sorted in increasing order of size. Therefore, $\|v\|_1 = \alpha_{n-k+1}c_{n-k+1} + \cdots + \alpha_n c_n$. The solution of ($\mathcal{P}_1$) is given by the vector in $\tilde{\mathcal{V}}^1_{1,k}$ that satisfies $\|v\|_{+\infty}$ and with the smallest ℓ_1-norm such that $\alpha = (\alpha_{n-k+1}, 0, \ldots, 0)$ where $\alpha_{n-k+1} = 1$.

To simplify and without loss of generality, we assume that I_{n-k+1} corresponds to the first index (up to a permutation). We can rewrite ($\mathcal{P}_1$) (Proposition 4) as:

$$\underset{\substack{v\in\mathbb{R}^{n-1}\\(1,v)\in\mathcal{V}_{1,k}}}{\arg\min}\|v\|_1$$

Constraints in ($\mathcal{P}_1$) can be moved to the following equality contraints:

Proposition 5 *Let w be the first column of $V_{2,n}^T$. We define W as the matrix $V_{2,n}^T$ whose first column w has been deleted.*

The minimization problem $(\tilde{\mathcal{P}}_1)$

$$\underset{\substack{v\in\mathbb{R}^{n-1}\\ Wv=-w}}{\arg\min}\|v\|_1$$

with $w = V_{2,n}^{(1)}$ and $W = V_{2,n}^{(n-1)}$ has a unique solution $\tilde{v}$ equals to $\mathbf{1}_{C_{n-k+1}}$ such that $\tilde{v} = (1, v)$.

Proof *We recall that $V_{2,n}$ is the restriction of the eigenvectors matrix to the $n-k$ first columns. Because the columns of this matrix form an orthogonal basis, $v \in \mathcal{V}_{1,k}$ is equivalent to $V_{2,n}^T v = 0$. Thus, $\tilde{v} = (1, v)$ satisfies the equation: $V_{2,n}^{(1)} + V_{2,n}^{(n-k-1)} v$, where $V_{2,n}^{(1)}$ is the first row of $V_{2,n}$ and $V_{2,n}^{(n-k-1)}$ the matrix $V_{2,n}$ whose first row has been deleted.*

For all $\tilde{v} = (1, v)$,

$$\begin{aligned} V_{2,n}^T\tilde{v} &= V_{2,n}^{(1)} + V_{2,n}^{(n-k-1)}v \\ &= 0 \\ \Leftrightarrow \qquad V_{2,n}^{(n-k-1)}v &= -V_{2,n}^{(1)} \end{aligned}$$

Note that in Proposition 5, $V_{2,n}^{(1)}$ and $V_{2,n}^{(n-k-1)}$ are denoted w and W.

Remark: Constraint problem $(\tilde{\mathcal{P}}_1)$ (Proposition 5) can be equivalently written as the following penalized problem:

$$\underset{v\in\mathbb{R}^{n-1}}{\arg\min}\ \|Wv + w\|_2^2 + \lambda\|v\|_1.$$

where $\lambda > 0$ is the regularization parameter that controls the balance between the constraint and the sparsity norm, $W \in \mathbb{M}_{n-k,n-1}$ is the matrix $V_{2,n}^T$ whose first column w has been deleted.

In the following, we will provide an algorithm based on the contraint problem $(\tilde{\mathcal{P}}_1)$ introduced in Proposition 5.

5 ℓ_1-spectral clustering algorithm

Now, we only consider graphs with an exact cluster structure whose edges have been perturbed by a coefficient $p \in [0, 1]$.

ℓ_1-spectral clustering algorithm is developed in a Matlab software. Starting with the number of blocks k of an adjacency matrix A and the column index of one representative element of each block $I_{n-k+1}, \ldots, I_n$, the pseudo-code for the algorithm is presented in Algorithm 1.

Steps from 3 to 14 are dedicated to the recovery of the indicators of connected components. The minimization problem introduced in Section 4.2 is solved using the ℓ_1-eq function of the Matlab optimization package ℓ_1-magic [2]. Vector $\tilde{v}_j$ contains the solution of the minimization problem (step 11).

To find the other connected components indicators, we add the constraint of being orthogonal to the previous computed vectors by deflating the matrix A (step 13) and we do the same to estimate the other connected component indicators.

Let F be the concatenation of the vectors $\tilde{v}_j$. As the algorithm is applied on a perturbed adjacency matrix, the elements in F are not exactly equal to one or zero but are very close to one for the indices associated to edges belonging to a same cluster and to zero for the remaining ones. Therefore, we shrink the solution (steps 16 to 20):

For all $j = 1, \ldots, n$, for all $i = 1, \ldots k$,

$$F_{ij} = \begin{cases} 1 & \text{if } F_{ij} > \frac{1}{2}, \\ 0 & \text{if } F_{ij} \le \frac{1}{2}. \end{cases}$$

The indicators of the clusters are given by the k column vectors of F.

6 ℓ_1-spectral clustering applications

6.1 Spectral clustering and ℓ_1-spectral clustering on simulated dataset

6.1.1 Performances

In Section 5, we introduced a new algorithm (called ℓ_1-spectral clustering) that aims to detect cluster structures in complex graphs. To illustrate graphically the performances of this method, we simulated a perturbed version of a graph with an exact group structure. The associated adjacency matrix is composed of $k \in [5, 10]$ blocks of size $c_{n-k+1}, \dots, c_n \in [10, 20]$. Let p be the level of Bernoulli noise applied on the adjacency matrix.

Once the matrix is disturbed by a strictly positive coefficient, we no longer have exact block structures. To recover it, we applied the traditional spectral clustering algorithm and the new ℓ_1-spectral clustering algorithm. Figure 1 gives the performances of both algorithm with a perturbation coefficient of $p = 2$.

We can notice that our model performs well in this task as both methods effectively recovers the clustering structure, which indicates the robustness of our model.

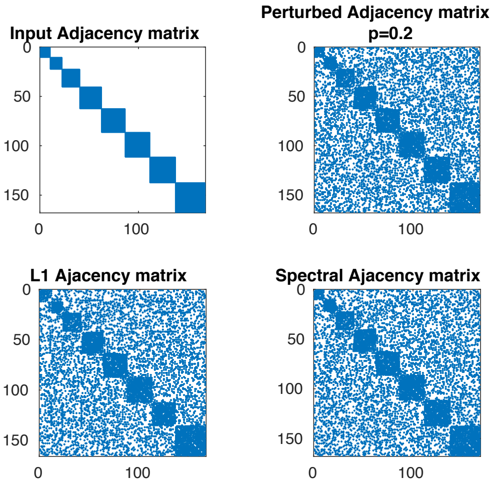


Figure 1: Input Adjacency matrix: Adjacency matrix with exact community structure. Perturbed Adjacency matrix $p = 0.2, 0.3$: Adjacency matrix after perturbation. L1 Adjacency matrix: Adjacency matrix recovery after ℓ_1-spectral clustering application. Spectral Adjacency matrix: Adjacency matrix recovery after spectral clustering application.

6.1.2 Robustness to perturbations

Then, we tested the robustness under perturbations of the spectral clustering ang l_1-spectral clustering algorithms. Let p be the level of Bernoulli noise, discretized in this section between 0 and 0.4. In this experiment, we simulate 100 graphs with $k \in [5, 10]$ clusters of size $c_{n-k+1}, \dots, c_n \in [10, 20]$.

We introduce the block membership function: for all node $i \in \{1, \dots, n\}$ of a graph $G(V, E)$ made of block structures of size $c_{n-k+1}, \dots, c_n$,

$$\begin{aligned} \tau \colon V &\to \{n-k+1, \dots, n\} \\ i &\mapsto c. \end{aligned}$$

For each value of p, we test the perfomances of both algorithms to recover the clusters of the graphs. The performances of the algorithms were evaluated by computing the percentage of missassigned nodes in average defined as $\frac{1}{100} \sum_{j=1}^{100} |\{i \in V : \tau(i) \neq \hat{\tau}_j(i)\}|$, where τ_j is the block membership function and $\hat{\tau}_j$ is the estimated membership function for the j-th model.

The results are plotted in Figure 3.

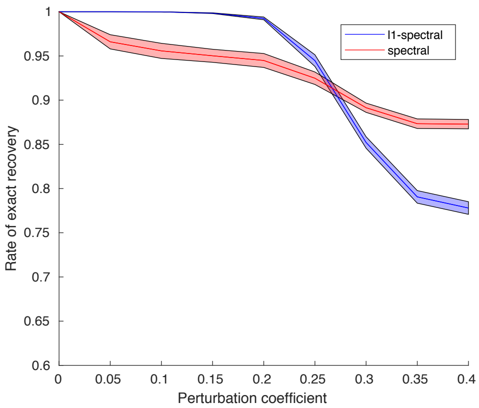


Figure 2: Fraction of nodes correctly classified using spectral (red line) and ℓ_1-spectral clustering (blue line) under increasing perturbation coefficient.

Figure 3 captures the fraction of nodes correctly classified and the associated region of confidence when ℓ_1-spectral clustering (blue) and spectral clustering (red) are applied under increasing perturbation coefficient.

6.1.3 Results

Both simulations show that the perturbation coefficient has an impact on the performance of spectral clustering and ℓ_1-spectral clustering. Moreover, we observe that ℓ_1-spectral clustering works better on simulated data for small perturbations (up to 30% Bernoulli noise) than spectral clustering. Thus, the new technique provides powerful results on small perturbations (rate of exact assignment is equal or very close to one).

6.2 ℓ_1-spectral clustering on real dataset

6.2.1 FLORINASH dataset

This section is dedicated to experimental studies to assess the performances of our method through real dataset. Experiments have been performed on R using the packages igraph, PLNmodels [3]. The dataset we used belongs to the project FLORINASH that proposes an innovative research concept to address the role of intestinal microfloral activity in Non-Alcoholic Fatty Liver Disease (NAFLD).

Hepatic steatosis is often observed in obese patients and is a preliminary stage to non-alcoholic fatty liver disease. The studied cohort [13] is made of obese patients featured with hepatic steatosis. It has been deeply studied and numerous clinical and biochemical data sets are available. We ran an ancillary study on 51 control and 6 diabetic patients with a median age of $42, 50$ years, and characterized by a median body weight of $124, 125$ kg and a glycemia of $5.8, 6.5$ mM.

The underlying dataset includes the output of sequencing 16S rDNA gene from liver biopsies to study microbial composition and diversity of obese patients. The standard approach to analyzing 16S rDNA sequence data relies on clustering reads by sequence similarity into Operational Taxonomic Units (OTUs). All OTUs are assigned to a taxonomic rank (phylum, class, order, family, genus and species). The standard way of representing the community structure inferred from microbial data is by means of an abundance table, where the rows correspond to samples (57 patients) and columns to features (831 microbial taxa). The goal of this analysis is to detect clusters of OTUs at their family taxonomy level according to their abundance by patients (53 OTUs at this taxonomy). Our aim is to identify the associations between the different microbial families by reconstructing the ecological network and make a direct comparisons between the two groups of patients.

6.2.2 Results

Microbiome data is compositional because the information that abundance tables contain is relative, the total number of counts per sample being highly variable. Few universal multivariate models are available for compositional data and existing models often impose undesired constraints on the dependency structure. To tackle this issue, we use the Poisson Log Normal model [3]. We use the framework of graphical models to model the dependency structure of the dataset.

From the graph modeled, we deduce the adjacency matrix and the score of each underlying hubs [16]. A hub is a node with a number of links that greatly exceeds the average, also called a high degree node. A node is given a high hub score by linking to a large number of nodes.The number of hubs selected give us the total number of clusters do be detected. l_1-spectral clustering applied on the adjacency matrix outputs 4 (respectively 5) clusters in control and diabetic patients (Figure 3).

Figure 4 shows that the gut microbiota from control and diabetic patients is characterized by two dominating phyla Firmicutes and Bacteroidetes ([20, 8]). We also added to the knowledge that in diabetic patients there was a disappearance of the frequency of the variable Verrucomicrobiaceae. This is also in agreement with the data from literature which demonstrate that this bacteria could be considered as a probiotic controlling metabolic diseases [6]. Eventually, from ℓ_1-spectral clustering, we identified that a novel variable i.e. the Fusobacteriaceae are important discriminant signatures of the non diabetic group while that of the diabetic group is related to the Firmicutes variable. Altogether, our algorithm was validated by the previously published findings from the FLORINASH cohort and even add to the knowledge that some specific variables could be associated with the diabetic or non diabetic signatures.

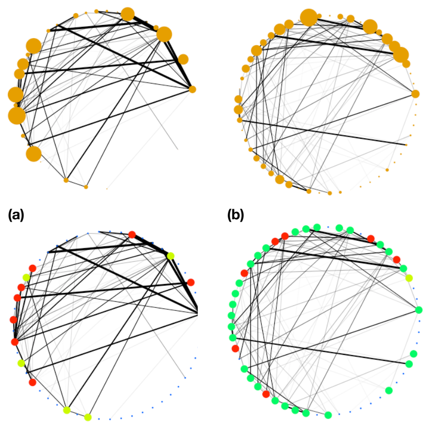
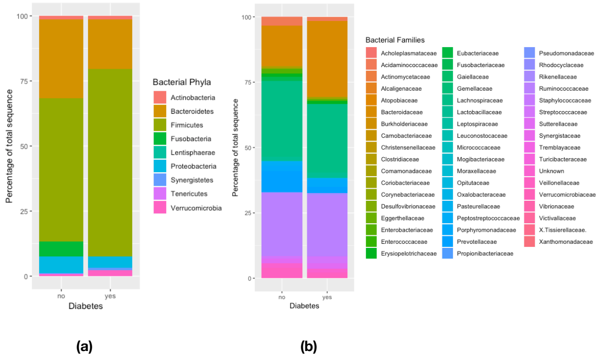


Figure 3: Application of ℓ_1-spectral clustering on the cohort composed of patients with and without diabetes. (a) Graph representing hubs and clusters related to diabetic patients (b) Graph representing hubs and clusters related to healthy patients.

7 Conclusion

We present ℓ_1-spectral clustering, a novel variation of spectral clustering algorithm based on promoting a sparse eigenvectors basis that pro-



(a)

(b)

Figure 4: (a) Proportion of Phyla in control and diabetic patients (b) Proportion of Families in control and diabetic patients.

vides information about the structure of the system observed. The associated graph is assumed to contain connected subnetworks. We characterized the indicators of these subnetworks as the ones that have the minimal ℓ_1-norm with respect to a specific restricted space. ℓ_1-spectral clustering benefits from this feasible objective function as a substitution of the k-means step of the traditional spectral clustering. Its effectiveness and robustness to small noise perturbations is confirmed by simulated and real examples.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the 18h Annual ACM-SIAM SYmposium on Discrete Algorithms*, pages 1027–1035, 2007.

[2] E. Candes and J. Romberg. L1-magic: Recovery of sparse signals via convex programming. Technical report, Caltech, 2005.

[3] J. Chiquet, M. Mariadassou, and S. Robin. Variational inference for sparse network reconstruction from count data. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[4] F.R.K. Chung. *Spectral graph theory*. American Mathematical Soc., 92 edition, 1997.

[5] P. Erds and A. Rnyi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[6] A. Everard. Cross-talk between akkermansia mucunuphila and intestinal epithelium controls diet-induced obesity. In *Proceedings of the national academy of sciences*, pages 9066–9071, 2013.

[7] S. Fortunato. Community detection in graphs. *Physics reports*, pages 75–174, 2010.

[8] S.R. Gill. Metagenomic analysis of the human distal gut microbiome. *Science*, 312:1355–1359, 2006.

[9] M. Girvan and E.J.M Newman. Community structure in social and biology networks. In *Proceedings of the national academy of sciences*, pages 7821–7826, 2002.

[10] M.S. Handcock and K.J. Gile. Modeling social networks form sampled data. *The Annals of Applied Statistics*, 4(1):5–25, 2010.

[11] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[12] J.J. Hopfield. Neural network and physical systems with emergent collective computational abilities. In *Proceedings of the national academy of sciences*, 1982.

[13] L. Hoyles. Molecular phenomics and metagenomics of hepatic steatosis in non-diabetic obese women. *Nature Medicine*, 24:1070–1080, 2018.

[14] H. Jeong, R. Albert B. Tombor, Z.N. Oltvai, and A.L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[15] A. Joseph and B. Yu. Impact of regularization on spectral clustering. *The Annals of Statistics*, 44(4):1765–1791, 2016.

[16] J. Kleinerg. Authoritative sources in a hyperlinked environment. In *Proceedings 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[17] N. De Lara and T. Bonald. Spectral embedding of regularized block models. (arXiv:1912.10903 [cs.LG]), 2020.

[18] S. Lattanzi and C. Sohler. A better k-means++ algorithm via local search. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[19] J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. *The annals of statistics*, 43(1):215–237, 2015.

[20] R.E. Ley. Obesity alters gut microbial ecology. In *Proceedings of the national academy of sciences*, pages 11070–11075, 2005.

[21] S. Lloyd. Least square quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–132, 1982.

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

[23] D. Meunier, R. Lambiotte, K.D. Ersche A. Fornito, and E.T. Bullmore. Hierarchical modularity in human brainfunctional networks. *Frontiers in neuroinformatics*, pages 3–37, 2009.

[24] E.J.M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026–113, 2004.

[25] M.E.J Newman. Modularity and community structure in networks. In *Proceedings of the National Academy of Sciences of the USA*, pages 8577–8696, 2006.

[26] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *In Advances in neural information processing systems*, pages 849–856, 2002.

[27] R. Pastor-Satorras and A. Vespignani. *Evolution and structure of the Internet: a statistical physics approach.* Cambridge University Press, 2007.

[28] R. Peng, H. Sun, and L. Zanetti. Partitioning well-clustered graphs: spectral clustering

works! In *Workshop and Conference Proceedings (JMLR 2015)*, pages 1–33, 2015.

[29] P. Pons and M. Latapie. Computing communities in large networks using random walks. *Computer and Information Sciences*, pages 284–293, 2005.

[30] A. Pothen. Graph partitioning algorithms with applications to scientific computing. *Parallel Numerical Algorithms*, 4:323–368, 1997.

[31] C. Ramirez. Why ℓ_1 is a good approximation to ℓ_0: A geometric explanation. *Journal of uncertain systems*, 7:203–207, 2013.

[32] K. Rohe and B. Yu. Spectral clustering and the high dimensional stochastic block model. *The annals of statistics*, 39(4):1878–1915, 2011.

[33] S.B. Seidman. Network structure and minimum degree. *Social Networks*, 5:269–287, 1983.

[34] W.G. Stewart, J. Sun, and B.H. Jovanovich. *Matrix pertubation theory.* Academic press New York, 175 edition, 1990.

[35] D.L. Sussman, M. Tang, D.E. Fishkind, and C.E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012.

[36] X. Wang, F. Nie, and H. Huang. Structured doubly stochastic matrix for graph based clustering: Structured doubly stochastic matrix. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016)*, pages 1245–1254, 2016.

[37] S. Wasserman and K. Faust. *Social network analysis: Methods and applications.* Cambridge university press, 8 edition, 1994.

[38] X. Wu. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[39] J. Xu and K. Lange. Power k-means clustering. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pages 6921–6931, 2019.

[40] Y. Zhang and K. Rohe. Understanding regularized spectral clustering via graph conductance. *In Advances in Neural Information Processing Systems*, pages 10631–10640, 2018.

Algorithm 1 ℓ_1-spectral clustering algorithm

```
Input: number of clusters $k$ , adjacency matrix $A$, indices of representative elements $Index = [I_{n-k+1}, \dots, I_n]$.
Initialize $F = []$.
{Recovery of the indicators of the connected components}
for $j = 1$ to $k$ do
    Eigen decomposition $[V, U]$ of $A$: $A = VU^tV$.
    Sort in ascending order the eigenvalues and the associated eigenvectors of $A$.
    Form the matrix $V_{2,n}$ by stacking the $n - k + j - 1$ eigenvectors associated to the smallest eigenvalues.
    Computation of constraints of the $l_1$ minimization problem
    Compute $T = V_{2,n}^t$
    Compute $W = T^{-Index[j]}$ and $w = T^{Index[j]}$(where $T^{Index[j]}$ is the column $Index[j]$ of $T$ and $V^{-I_j}$ the matrix $T$ wihtout the column $Index[j]$)
    Compute the solution $v^j$ of the following problem
        $$\underset{\substack{v\in\mathbb{R}^{n-1} \\ Wv=-w}}{\arg\min} \|v\|_1.$$
    Recovery of the $j_{th}$ cluster:
    $\tilde{v}_j = [v^j_1\ v^j_2\ \dots\ v^j_{Index[j]-1}\ 1\ v^j_{Index[j]+1}\ \dots\ v^j_n]$
    $F$ concatenation of the $j_{th}$ clusters and deflation of $A = A - \tilde{v}_j^t\tilde{v}_j$ to recover the other indicator vectors
end for
$F((F > 0.5)) = 1$
$F((F \leq 0.5)) = 0$
Output: $k$ column vectors $F$.
```