

A Compressive Sensing Approach to Community Detection with Applications

Ming-Jun Lai*

Daniel McKenzie †

September 1, 2017

Abstract

The community detection problem for graphs asks one to partition the n vertices V of a graph G into k communities, or clusters, such that there are many intracenter edges and few intercluster edges. Of course this is equivalent to finding a permutation matrix \mathbf{P} such that, if \mathbf{A} denotes the adjacency matrix of G , then \mathbf{PAP}^\top is approximately block diagonal. As there are k^n possible partitions of n vertices into k subsets, directly determining the optimal clustering is clearly infeasible. Instead one seeks to solve a more tractable approximation to the clustering problem. In this paper we reformulate the community detection problem via sparse solution of a linear system associated with the Laplacian of a graph G and then develop a two-stage approach based on a thresholding technique and a compressive sensing algorithm to find a sparse solution which corresponds to the community containing a vertex of interest in G . Crucially, our approach results in an algorithm which is able to find a single cluster of size n_0 in $\mathcal{O}(n \ln(n)n_0)$ operations and all k clusters in fewer than $\mathcal{O}(n^2 \ln(n))$ operations. This is a marked improvement over the classic spectral clustering algorithm, which is unable to find a single cluster at a time and takes approximately $\mathcal{O}(n^3)$ operations to find all k clusters. Moreover, we are able to provide robust guarantees of success for the case where G is drawn at random from the Stochastic Block Model, a popular model for graphs with clusters. Extensive numerical results are also provided, showing the efficacy of our algorithm on both synthetic and real-world data sets.

1 Introduction

The clustering problem for a graph $G = (V, E)$ is to divide the vertex set V into subsets $V = C_1 \cup \dots \cup C_k$ such that there are many intracenter edges (edges between vertices in the same cluster) and few intercluster edges (edges between vertices in different clusters) in E . This is a widely studied problem in exploratory data analysis, as one can reasonably assume that vertices in the same cluster a ‘similar’, in some sense. We refer the reader to the survey article [20] for further details and a thorough overview of existing algorithmic approaches. We note that [20] refers to the clustering problem as the community detection problem, and we shall use these two phrases interchangeably. As is well-known, detecting clusters in G is equivalent to finding a permutation matrix \mathbf{P} such that if \mathbf{A} is the adjacency matrix of G , then \mathbf{PAP}^\top is almost block diagonal. Thus, we can think of the clustering problem as a special case of the matrix reduction problem where the matrix in question has binary entries.

*mjlai@uga.edu. Department of Mathematics, University of Georgia, Athens, GA 30602. This research is partially supported by the National Science Foundation under the grant #DMS 1521537.

†mckenzie@math.uga.edu. Department of Mathematics, University of Georgia, Athens, GA 30602. The second author gratefully acknowledges the financial support of the NRF of South Africa while this research was conducted

One class of robust and accurate algorithms used to solve the clustering problem are the spectral algorithms. Loosely, they work as follows. Suppose that $|V| = n$ and let \mathbf{L} denote the graph Laplacian, while $\mathbf{1}_{C_a}$ denotes the indicator vector of the a -th cluster (both to be defined in §3). Suppose further that it is known *a priori* that G has k clusters. Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be orthogonal eigenvectors associated to the k smallest eigenvalues of L , and consider the subspace $\text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \subset \mathbb{R}^n$. One can show that, under certain conditions, this subspace is ‘close’ to $\text{span}\{\mathbf{1}_{C_1}, \dots, \mathbf{1}_{C_k}\}$ and hence one can use the basis $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ to infer the supports of the basis $\{\mathbf{1}_{C_1}, \dots, \mathbf{1}_{C_k}\}$, thus determining the clusters C_1, \dots, C_k (of course $\text{supp}(\mathbf{1}_{C_a}) = C_a$). We refer the reader to [41], [35] or [37] for details.

Despite its theoretical and experimental success, the spectral approach has three main drawbacks:

1. The number of clusters k , needs to be known *a priori*. Clearly this is not always the case for real data sets.
2. The algorithm cannot be used to find only a few clusters. As in many applications one is only interested in finding one or two clusters (thinking of the problem of identifying friends or associates of a given user from a social network data set). Moreover in other cases where the data set is extremely large, or only partially known, it might be computationally infeasible to identify all clusters.
3. Computing an eigen decomposition of \mathbf{L} typically requires $\mathcal{O}(n^3)$ operations, making spectral methods prohibitively slow for truly large data sets, such as those arising from electronic social networks like Facebook or LinkedIn, or those arising from problems in Machine Learning.

However, in many situations (the social network example being one such case), the expected size of the clusters, n_0 , is small compared to n , and hence the indicator vectors $\mathbf{1}_{C_a}$ will be *sparse*. Our approach is to adapt sparse recovery algorithms from the compressive sensing study to solve the following problem:

$$\arg \min \|\mathbf{L}\mathbf{x}\|_2 : \quad \text{subject to } x_i = 1 \text{ and } \|\mathbf{x}\|_0 \leq n_0 \quad (1.1)$$

eq: CompClus

to determine, directly, an approximation to the indicator vector of the cluster containing the vertex of interest \mathbf{v}_i . One can then recover the cluster by considering the support of this vector. If desired, one can iterate the algorithm to find all the remaining clusters of G .

Adapting compressive sensing algorithms to solve (1.1) proves challenging, as \mathbf{L} is a poorly conditioned sensing matrix. In general, a greedy type algorithm such as orthogonal matching pursuit or iterative hard thresholding (cf. [21]) work very well when there are no intercluster edges (in this case finding clusters reduces to finding connected components). Unfortunately, in the presence of even a small number of intercluster edges, the first few iterations of a greedy algorithm are likely to pick some indices outside the desired cluster. To overcome this difficulty, we propose a novel two stage algorithm (see Algorithm 4 in §8) in which the first stage identifies a subset $\Omega \subset V$ which contains the cluster of interest with high probability. The second stage then extracts the cluster of interest from Ω using a greedy algorithm (we use Subspace Pursuit cf. [14]). In addition to the aforementioned algorithm, the main contributions of this paper are the following:

1. An analysis of the Restricted Isometry and Coherence properties of the graph Laplacian. In particular, we provide a series of probabilistic bounds on the restricted isometry constants and coherence of Laplacians of graphs drawn from a well-studied model of random graphs (the Stochastic Block Model (SBM)). See §5 and §6.
2. A proof that the Optimal Matching Pursuit (OMP) algorithm can be successfully used to detect connected components of any graph, by solving (1.1) (see §7.)

3. A proof that, given a vertex \mathbf{v}_i , our Single Cluster Pursuit (SCP) Algorithm 4 successfully finds the cluster containing \mathbf{v}_i when G is drawn from the Stochastic Block Model (SBM), for a certain range of parameters and with probability tending to 1 as $n \rightarrow \infty$. We achieve this by combining the bounds of contribution with the theory of totally perturbed compressive sensing, e.g. in [28].
4. An analysis of the computational complexity of Algorithm 4, showing that it finds a single cluster in $\mathcal{O}(n^2 + \ln(n)nn_0)$ time and all clusters in $\mathcal{O}(kn^2 + \ln(n)n^2)$ time.

The structure of this paper is as follows. After briefly reviewing related work in §2, in §3 and §4 we acquaint the reader with the necessary concepts from spectral graph theory and compressive sensing, respectively. In §5 and §6 we study the restricted isometry property and coherence property of Laplacians of random graphs. In §7 and §8, we describe two algorithms to handle graphs from the Stochastic Block Model $\mathcal{G}(n, k, p, q)$ for $q = 0$ and $q > 0$, with respectively. We shall show that, under certain mild assumptions on the graph G , the algorithms will indeed find the correct clustering with high probability. §9 contains the computational complexity analysis, some possible extensions and several future research directions. Finally, in §10 we present the results of several numerical experiments to demonstrate the accuracy and speed of our algorithm and its powerful performance.

2 Related Work

The notion of community detection in graphs arises independently in multiple fields of applied science, such as Sociology ([47],[40]), Computer Engineering ([26]), Machine Learning ([43]) and Bioinformatics ([13]). In addition, many data sets can be represented as graphs by considering data points as vertices and attaching edges between vertices that are ‘close’ with respect to an appropriate metric. Thus, community detection algorithms can also be used to detect clusters in general data sets, and indeed they have been shown to be superior to other clustering algorithms (for example k -means) at detecting non-convex clusters ([30]).

The canonical probabilistic model of a graph containing communities is the Stochastic Block Model (SBM), first explicitly introduced in the literature in the early 1980’s by Holland, Laskey, and Leinhardt in [29]. Since then, there has been an explosion of interest in the SBM, driven in large part by its many applications. In [1], Abbe identifies three forms of the community detection problem, based on the kind of asymptotic accuracy we require (here, as in the rest of the paper, when we speak of asymptotics we are considering the situation where the number of vertices, n , goes to ∞). In this paper we shall be concerned with the *Exact Recovery Problem*, where we require that $\mathbb{P}(\# \text{misclassified vertices} > 0) = o(1)$. A fundamental information theoretic barrier to exact recovery is given by the following result (cf. [3]):

Theorem 2.1. *The exact recovery problem, i.e. $\mathbb{P}(\# \text{misclassified vertices} > 0) = o(1)$ with respect to n for the symmetric SBM $\mathcal{G}(n, k, p, q)$ is solvable in polynomial time if, writing $p = P \ln(n)/n$ and $q = Q \ln(n)/n$:*

$$\frac{1}{k} \left(\sqrt{P} - \sqrt{Q} \right) > 1 \tag{2.1}$$

and not solvable if:

$$\frac{1}{k} \left(\sqrt{P} - \sqrt{Q} \right) < 1 \tag{2.2}$$

Remark 2.2. This result was proved in [3] (and see also [1]). We note that in the cases our algorithm is guaranteed to solve the exact recovery problem (see Theorem 8.2), we assume that $\frac{1}{k} \left(\sqrt{P} - \sqrt{Q} \right) \rightarrow \infty$, well above this theoretical bound.

Given the bound (2.1), the challenge then is to construct efficient algorithms to solve the Exact Recovery Problem. There are myriad algorithmic approaches, such as the spectral approach (originally proposed by Fiedler in [19] for the two cluster case), hierarchical approaches like the DIANA algorithm popular in bioinformatics ([31]) and message passing algorithms ([24]), to name a few. Recently, two new classes of algorithms, namely degree-profiling ([3]) and semidefinite programming approaches ([2], [36], [27] and [33] among others) have been shown to solve the exact recovery problem with high probability right down to the theoretical bound (2.1). Degree-profiling even runs in quasi-linear time, although it requires the parameters p , q and k as inputs, making it less than ideal for analysing real world data sets. As the new algorithm we propose is most closely related to the spectral approach, let us recall this algorithm here (as formulated by Ng, Jordan and Weiss in [41]).

Algorithm 1 The Spectral Clustering Algorithm (SC)

Input: the adjacency matrix \mathbf{A} .

- (1) Form the degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$ and Laplacian $\mathbf{L} := \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$
- (2) Find the k (orthogonal) eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ corresponding to the k largest eigenvalues of \mathbf{L} .
- (3) Form the matrix $\mathbf{U} = [\mathbf{v}_1, \dots, \mathbf{v}_k]$ and normalize the *rows* to get \mathbf{V}
- (4) Sort the rows of \mathbf{V} into k clusters B_1, \dots, B_k using k -means.
- (5) Assign vertex i to community C_j if and only if row i is in B_j .

Output: Communities C_1, \dots, C_k .

We mention that notions from Compressive Sensing have been applied to community detection before, notably in the semidefinite programming approaches mentioned above, and in [45] where signal processing techniques for functions defined on a graph G are used to speed up the computation of the eigenvectors of \mathbf{L} . Our approach is distinct from these. To the best of the authors' knowledge, the study in this paper is the first attempt to find the indicator vectors $\mathbf{1}_{C_i}$ directly using sparse recovery.

3 Preliminary on Graph Theory

3.1 Some Elementary Notions and Definitions

Formally, by a graph G we mean a set of vertices V together with a subset $E \subset \{\{u, v\} : u, v \in V\}$ of edges¹. As we are only concerned with finite graphs, we shall always identify the vertex set V with a finite set of consecutive natural numbers: $V = [n] := \{1, 2, \dots, n\}$. The *degree* of any vertex $i \in G$ is the total number of edges incident to i , that is $d_i = |\{\{i, j\} \in E\}|$.

A *subgraph* G' of G is a subset of vertices $V' \subset V$ together with a subset of edges $E' \subset E \cap V' \times V'$. Given any subset $S \subset V$, we denote by G_S the subgraph with vertex set S and edge set *all* edges $\{i, j\}$ with $i, j \in S$. A *path* in G is a set of 'linked' edges $\{i_1, i_2\}, \{i_2, i_3\}, \dots, \{i_{k-1}, i_k\}$, and we say that G is *connected* if there is a path linking any two vertices $i, j \in V$, and *disconnected otherwise*. If G is disconnected, any subgraph $G_S \subset G$ which is connected and maximal with respect to the property of being connected is called a *connected component*.

If G is connected, we define the *diameter* of G to be the length of (i.e. the number of edges in) the longest path. Given any $i \in V$ and any non-negative integer j , we define the ball $B_j(i) \subset V$ to be the set of all vertices connected to i by a path of length j or shorter. A good reference on elementary graph theory is [5].

¹We only consider undirected graphs

3.2 The Graph Laplacian

To any graph G with $|V| = n$ we associate a symmetric, $n \times n$, non-negative matrix called the *adjacency matrix* \mathbf{A} , defined as $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 1$ if $\{i, j\} \in E$ and $\mathbf{A}_{ij} = 0$ otherwise. The graph Laplacians of G are defined as follows.

Definition 3.1. Let \mathbf{A} denote the adjacency matrix of a graph G and let \mathbf{D} denote the matrix $\text{diag}(d_1, \dots, d_n)$ where d_i is the degree of the i -th vertex. We define the *normalized, symmetric graph Laplacian* of G as $\mathbf{L}_s := \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ and the *normalized, random walk graph Laplacian* as $\mathbf{L}_{rw} := \mathbf{I} - \mathbf{D}^{-1} \mathbf{A}$.

We first have a few basic properties of graph Laplacians.

Theorem 3.2. Suppose that $\mathbf{L} = \mathbf{L}_s$ or \mathbf{L}_{rw} . We have the following properties:

1. The eigenvalues of \mathbf{L} are real and non-negative.
2. $\lambda_{n-1} \leq 2$
3. Let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ denote the eigenvalues of L in ascending order. Let k denote the number of connected components of G . Then $\lambda_i = 0$ for $i \leq k$ and $\lambda_i > 0$ for $i > k$.

PROOF. For $\mathbf{L} = \mathbf{L}_s$, Items 1 to 3 follow from Lemma 1.7 in [12]. If $\mathbf{L} = \mathbf{L}_{rw}$, observe that $\mathbf{L}_{rw} = \mathbf{D}^{-1/2} \mathbf{L}_s \mathbf{D}^{1/2}$, and so the eigenvalues of \mathbf{L}_{rw} and \mathbf{L}_s coincide. Hence the above hold for \mathbf{L}_{rw} as well. \square

For any subset $S \subset [n]$ we define its *indicator vector*, denoted $\mathbf{1}_S \in \mathbb{R}^n$, by $(\mathbf{1}_S)_i = 1$ if $i \in S$ and zero otherwise. Let C_1, \dots, C_k be the clusters of G . One can check (and see also [35] proposition 2) that $\mathbf{L}_{rw} \mathbf{1}_{C_i} = 0$ for $i = 1, \dots, k$. Thus we have the following:

Theorem 3.3. The indicator vectors of the connected components of G , $\mathbf{1}_{C_1}, \dots, \mathbf{1}_{C_k}$, form a basis for the zero eigenspace (i.e. the kernel) of \mathbf{L}_{rw} .

For the rest of this paper, by \mathbf{L} we shall mean \mathbf{L}_{rw} . We shall refer to the i -th column of \mathbf{L} as ℓ_i . One can easily check that:

$$(\ell_i)_k = \delta_{ik} - \frac{\mathbf{A}_{ik}}{d_k}.$$

Finally, we shall denote by \mathbf{L}_{-i} the submatrix of \mathbf{L} obtained by dropping the column ℓ_i .

3.3 Random Graph Theory

As outlined in §2, the Stochastic Block Model (SBM) is a widely used mathematical model of a random graph with clusters.

Definition 3.4. Given $n = kn_0$, fix a partition of $V := [n]$ into k subsets C_1, \dots, C_k of equal size n_0 . We say G is drawn from the SBM $\mathcal{G}(n, k, p, q)$ if, for all $i, j \in [n]$ with $i \neq j$, the edge $\{i, j\}$ is inserted independently and with probability p if $i, j \in C_a$ for some $a = 1, \dots, k$ and q otherwise.

As we are interested in clustering we assume that $q \ll p$. We emphasize that the partition $V = C_1 \cup C_2 \cup \dots \cup C_k$ is fixed before any edges are assigned. We note that the subgraphs G_{C_a} are i.i.d instances of a simpler random graph model, the Erdős-Rényi (ER) model $\mathcal{G}(n_0, p)$, first introduced in [18]

Definition 3.5. We say H is drawn from the ER model $\mathcal{G}(n_0, p)$ if H has n_0 vertices and for all $i, j \in [n_0]$ the edge $\{i, j\}$ is inserted independently and with probability p .

Returning to the SBM, for any vertex i in community C_a , we define its *in-community degree* as $d_i^0 = \#\{\{i, j\} \in E : j \in C_a\}$ and its *out-of-community degree* as $d_i^\epsilon := \#\{\{i, j\} \in E : j \notin C_a\}$. One can easily see that

$$\mathbb{E}[d_i^0] = p(n_0 - 1) \text{ and } \mathbb{E}[d_i^\epsilon] = q(n - n_0)$$

and by definition $d_i = d_i^0 + d_i^\epsilon$. In fact d_i^0 is the degree of i considered as a vertex in the ER subgraph G_{C_a} . An important fact about degrees in ER random graphs is that they concentrate around their mean:

Theorem 3.6. *Suppose G is drawn from $\mathcal{G}(n, k, p, q)$.*

1. *For any $\alpha > 0$, if $p \geq \frac{4 \ln(n)}{\alpha^2 n_0}$ then with probability at least $1 - 1/n$:*

$$(1 - \alpha)n_0 p \leq d_i^0 \leq (1 + \alpha)n_0 p \quad \text{for all } i \in \{1, 2, \dots, n\} \quad (3.1)$$

eq:AlphaDegBo

2. *In particular, if $p \geq \frac{4k(\ln(n))^2}{n}$ then (3.1) holds, with probability at least $1 - 1/n$, for $\alpha = 1/\sqrt{\ln(n)} = o(1)$.*

DegBound

PROOF. This theorem is a variation on a well known result for ER graphs (e.g. theorem 3.6 in [25]). Each d_i^0 follows the binomial distribution with parameters $n_0 - 1$ and p , so by the Chernoff bound: $\mathbb{P}[|d_i^0 - n_0 p| \geq \alpha n_0 p] \leq e^{-\alpha^2 n_0 p/2}$. Hence:

$$\begin{aligned} \mathbb{P}\left[\max_{i \in [n]} |d_i^0 - n_0 p| \geq \alpha n_0 p\right] &= \mathbb{P}[|d_1^0 - n_0 p| \geq \alpha n_0 p \text{ or } |d_2^0 - n_0 p| \geq \alpha n_0 p \text{ or } \dots \text{ or } |d_n^0 - n_0 p| \geq \alpha n_0 p] \\ &\leq \sum_{i=1}^n \mathbb{P}[|d_i^0 - n_0 p| \geq \alpha n_0 p] = n e^{-\alpha^2 n_0 p/2} \end{aligned}$$

Thus $\mathbb{P}\left[\max_{i \in [n_0]} |d_i^0 - n_0 p| \leq \alpha n_0 p\right] = 1 - n e^{-\alpha^2 n_0 p/2}$ If $p \geq \frac{4 \ln(n)}{\alpha^2 n_0} = \frac{2 \ln(n^2)}{\alpha^2 n_0}$ then:

$$1 - n e^{-\alpha^2 n_0 p/2} \leq 1 - n e^{-\ln(n^2)} = 1 - 1/n$$

This proves part 1. Part 2 follows by taking $\alpha = 1/\sqrt{\ln(n)}$, in which case the lower bound on p becomes:

$$\frac{4 \ln(n)}{\alpha^2 n_0} = \frac{4(\ln(n))^2}{n_0} = \frac{4(\ln(n))^2}{n/k} = \frac{4k(\ln(n))^2}{n}$$

□

The second remarkable property of the ER model is that the eigenvalues of \mathbf{L} also concentrate around their mean:

Theorem 3.7. *Let \mathbf{L} be the Laplacian of a random graph drawn from $\mathcal{G}(n_0, p)$ with $p \gg (\ln(n_0))^2/n_0$ and let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_0}$ denote its eigenvalues. Then almost surely²:*

$$\max_{i \neq 1} |1 - \lambda_i| \leq (1 + o(1)) \frac{4}{\sqrt{p n_0}} + \frac{g(n_0) \log^2(n_0)}{p n_0}$$

where $g(n_0)$ is a function tending to infinity arbitrarily slowly.

²Given a family of random graph models \mathcal{G}_n we say that some graph property P holds almost surely if $\mathbb{P}[G \in \mathcal{G}_n, G \text{ does not have } P] = o(1)$ with respect to n

ChungLuVu

PROOF. Given that the expected degree of each vertex in G is pn_0 , this is just Theorem 3.6 in [11]
 \square

Remark 3.8. For our purposes, it will be enough to note that this gives:

$$\lambda_2 \geq 1 - \frac{4/\sqrt{p}}{\sqrt{n_0}} - o\left(\frac{1}{\sqrt{n_0}}\right) \text{ and } \lambda_{n_0} \leq 1 + \frac{4/\sqrt{p}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)$$

almost surely.

4 Preliminaries on Compressive Sensing

Let $\Phi \in \mathbb{R}^{m \times N}$ and $\mathbf{y} \in \mathbb{R}^m$ with $m < N$. We say that a vector \mathbf{x} is *sparse* if it has few non-zero entries relative to its length. We follow the convention of defining the ‘0 quasi-norm’ as:

$$\|\mathbf{x}\|_0 = \#\{x_i : x_i \neq 0\}$$

and we say \mathbf{x} is s -sparse if $\|\mathbf{x}\|_0 \leq s$. Compressive sensing is concerned with solving

$$\arg \min \{\|\mathbf{x}\|_0 : \mathbf{x} \in \mathbb{R}^N, \Phi \mathbf{x} = \mathbf{y}\} \quad (4.1)$$

in the case where $m < N$ (that is, when the linear system is underdetermined). We call (4.1) the *Sparse Recovery Problem*. One also considers the *Perturbed Sparse Recovery Problem* where $\mathbf{y} = \Phi \mathbf{x}^* + \mathbf{e}$ with $\|\mathbf{e}\|_2 \leq \eta < \|\mathbf{y}\|_2$ and we wish to solve:

$$\mathbf{x}^\# := \arg \min \{\|\mathbf{x}\|_0 : \mathbf{x} \in \mathbb{R}^N, \|\Phi \mathbf{x} - \mathbf{y}\|_2 \leq \eta\} \quad (4.2)$$

while guaranteeing that $\mathbf{x}^\#$ is a good approximation to \mathbf{x}^* by bounding $\|\mathbf{x}^\# - \mathbf{x}^*\|_2$? We remark that problem (4.2) is equivalent to the dual problem:

$$\mathbf{x}^\# := \arg \min \{\|\Phi \mathbf{x} - \mathbf{y}\|_2 : \mathbf{x} \in \mathbb{R}^N, \|\mathbf{x}\|_0 \leq n_0\} \quad (4.3)$$

We refer the reader to [21] for an excellent introduction to the area.

4.1 Computational Algorithms

Many numerical algorithms have been invented for solving (4.1) and (4.3), for example, ℓ_1 convex minimization and its variations, hard thresholding iteration and its variations, greedy approaches such as orthogonal matching pursuit (OMP) as well as more exotic approaches like ℓ_q nonconvex minimization. See, for example, [10], [8], [6], [46], [7], [22], [23]. Due to its efficiency, we shall focus on the greedy approach in this paper, specifically the Orthogonal Matching Pursuit (OMP) and Subspace Pursuit (SP) algorithms, see Algorithms 2 and 3, respectively. For notational convenience, we shall follow [21] and denote by Φ_S the column submatrix of Φ consisting of the columns indexed by the subset $S \subset [N]$. For a vector $\mathbf{x} \in \mathbb{R}^N$ we denote by \mathbf{x}_S either the subvector in $\mathbb{R}^{|S|}$ consisting of the entries x_i for i indexed by S , or the vector

$$(\mathbf{x}_S)_i = \begin{cases} x_i & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

It should always be clear from the context which definition we are referring to. We also remind the reader of the following operations on vectors $\mathbf{v} \in \mathbb{R}^n$, defined in [21]:

$$\mathcal{L}_s(\mathbf{v}) = \text{index set of } s \text{ largest absolute entries of } \mathbf{v} \text{ and } \mathcal{H}_s(\mathbf{v}) = \mathbf{v}_{\mathcal{L}_s(\mathbf{v})} = \begin{cases} v_i & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

where \mathcal{H}_s is sometimes referred to as the *Hard Thresholding Operator*.

Algorithm 2 The OMP Algorithm

Inputs: \mathbf{y} and Φ

Initialize: $\mathbf{r}^{(0)} = \mathbf{y}$, $S^{(0)} = \emptyset$, $\mathbf{x}^{(0)} = \mathbf{0}$ and $\alpha = 0$.

while True **do**

$\alpha \leftarrow \alpha + 1$

$i^{(\alpha)} := \mathcal{L}_1(\Phi^\top \mathbf{r}^{\alpha-1})$

$S^{(\alpha)} = S^{(\alpha-1)} \cup \{i^{(\alpha)}\}$

$\mathbf{x}^{(\alpha)} := \arg \min \{ \|\Phi_{S^{(\alpha)}} \mathbf{z}_{S^{(\alpha)}} - \mathbf{y}\|_2 : \mathbf{z} \in \mathbb{R}^N \text{ and } \text{supp}(\mathbf{z}) \subset S^{(\alpha)} \}$

$\mathbf{r}^{(\alpha)} := \mathbf{y} - \Phi \mathbf{x}^{(\alpha)}$

if Stopping Criterion met **then**

 Output $\mathbf{x}^\# := \mathbf{x}^\alpha$

 Break

end if

end while

Algorithm 3 The SP algorithm ([14])

Inputs: \mathbf{y} , Φ and an integer $s \geq 1$

Initialization:

(1) $T^0 = \mathcal{L}_s(\Phi^\top \mathbf{y})$.

(2) $\mathbf{x}^0 = \arg \min \{ \|\mathbf{y} - \Phi_{T^0} \mathbf{x}\|_2 : \text{supp}(\mathbf{x}) \subset T^0 \}$

(3) $\mathbf{r}^0 = \mathbf{y} - \Phi_{T^0} \mathbf{x}^0$

Iteration:

for $k = 1 : k_{\max}$ **do**

(1) $\hat{T}^k = T^{k-1} \cup \mathcal{L}_s(\Phi^\top \mathbf{r}^{k-1})$

(2) $\mathbf{u} = \arg \min \{ \|\mathbf{y} - \Phi_{\hat{T}^k} \mathbf{x}\|_2 : \mathbf{x} \in \mathbb{R}^N \text{ and } \text{supp}(\mathbf{x}) \subset \hat{T}^k \}$

(3) $T^k = \mathcal{L}_s(\mathbf{u})$ and $\mathbf{x}^k = \mathcal{H}_s(\mathbf{u})$

(4) $\mathbf{r}^k = \mathbf{y} - \Phi_{T^k} \mathbf{x}^k$

end for

The most common stopping criteria for Algorithm 2 are $\alpha = k$, or $\|\mathbf{r}^{(\alpha)}\|_2 < \epsilon$ for a given k or ϵ . A sufficient condition to guarantee the convergence of Algorithm 2 is the following

Theorem 4.1. *For any $S \subset [N]$, if Φ_S is injective and satisfies*

$$\|\Phi_S^\dagger \Phi_{S^c}\|_{1 \rightarrow 1} < 1, \quad (4.4)$$

Tropp

where Φ_S^\dagger is the pseudo-inverse of Φ_S , then any vector \mathbf{x} with support S is recovered in at most $s = |S|$ steps of OMP.

RecovCond

We refer to [46] for a proof. See also Remark 3.6 in [21].

Remark 4.2. In [14] where the SP algorithm is introduced, they suggest solving the least squares problems (that is (2) in Initialization and (2) in Iteration) exactly. In our implementation, we use MATLAB's `lsqr` algorithm to solve them approximately, to a high precision. As pointed out by [38] in their analysis of CoSaMP, a very similar algorithm, this does not affect the convergence analysis of the algorithm.

4.2 Fundamental Concepts

A very important concept, the *restricted isometry property* (RIP), introduced by Candés and Tao ([10]) plays a critical role in the study of the existence and uniqueness of a sparse solution from a sensing matrix and whether the solution of an ℓ_1 minimization is the sparse solution. Another concept, *mutual coherence*, introduced by Donoho and his collaborators in ([17]) is also used in this study.

RIPdef

Definition 4.3. Letting $0 < s < m < N$ be an integer and Φ_S be a submatrix of Φ which consists of columns of Φ whose column indices are in $S \subset [N]$. The *restricted isometry constant* (RIC) δ_s of Φ is the smallest quantity such that

$$(1 - \delta_s) \|\mathbf{x}\|_2^2 \leq \|\Phi_S \mathbf{x}\|_2^2 \leq (1 + \delta_s) \|\mathbf{x}\|_2^2 \quad (4.5) \quad \text{RIP}$$

for all subsets S with cardinality $|S| \leq s$. If a matrix Φ has such a constant $\delta_s < 1$ for some s , Φ is said to possess RIP of order s . It is known that

$$\delta_s = \max_{\substack{S \subset \{1, \dots, n\} \\ |S| \leq s}} \|\Phi_S^\top \Phi_S - I_s\|_{2 \rightarrow 2}, \quad (4.6) \quad \text{Foucart}$$

where I_s is the identity matrix of size $s \times s$.

Definition 4.4. The *coherence* of a matrix Φ , denoted μ , is the largest normalized inner product between its columns:

$$\mu := \max_{i \neq j} \frac{|\langle \phi_i, \phi_j \rangle|}{\|\phi_i\|_2 \|\phi_j\|_2}.$$

4.3 Totally Perturbed Compressive Sensing

turbCS

Frequently it is useful to modify problem (4.3) further to allow for small perturbations in the observed measurement matrix. That is, suppose that $\mathbf{y} = \Phi \mathbf{x}^* + \mathbf{e}$ and let $\hat{\Phi} = \Phi + \mathbf{E}$, where \mathbf{E} is a small perturbation matrix. Denoting again:

$$\mathbf{x}^\# := \arg \min \{ \|\hat{\Phi} \mathbf{x} - \mathbf{y}\|_2 : \mathbf{x} \in \mathbb{R}^N, \|\mathbf{x}\|_0 \leq s \}, \quad (4.7) \quad \text{eq: SparseReco}$$

can we still guarantee that $\mathbf{x}^\#$ is a good approximation to \mathbf{x}^* by bounding $\|\mathbf{x}^\# - \mathbf{x}^*\|_2$? Problem (4.7) is called the *Totally Perturbed Sparse Recovery Problem*. Analyzing (4.7) is often more important for applications than analyzing (4.3), as frequently we only know the measurement matrix Φ to within a certain error tolerance.

turbRIC

Theorem 4.5 (Herman and Strohmer, [28]). Suppose that $\hat{\Phi} = \Phi + \mathbf{E}$. Let δ_s and $\hat{\delta}_s$ denote the s restricted isometry constants of Φ and $\hat{\Phi}$ respectively. Define³ $\epsilon_\Phi^s := \|\mathbf{E}\|_{2,s} / \|\Phi\|_{2,s}$. Then:

$$\hat{\delta}_s \leq (1 + \delta_s) (1 + \epsilon_\Phi^s)^2 - 1$$

To bound the error $\|\mathbf{x}^\# - \mathbf{x}^*\|_2$, we need the following result. Define $\mathbf{y} = \Phi \mathbf{x}$ and suppose that $\hat{\mathbf{y}} = \hat{\Phi} \mathbf{x} + \mathbf{e}$ is the observed (perturbed) measurement vector. Suppose further that we only have access to $\hat{\Phi} = \Phi + \mathbf{E}$, a small perturbation of the measurement matrix. Define $\epsilon_{\mathbf{y}} := \|\mathbf{e}\|_2 / \|\mathbf{y}\|_2$, $\epsilon_\Phi = \|\mathbf{E}\|_2 / \|\Phi\|_2$ and let ϵ_Φ^s be as above.

³For a matrix Φ we denote by $\|\Phi\|_p$ the induced operator norm $\|\Phi\|_p = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\Phi \mathbf{x}\|_p}{\|\mathbf{x}\|_p}$. By $\|\Phi\|_{p,s}$ we mean the semi-norm $\max_{\substack{S \subset [n] \\ |S|=s}} \|\Phi_S\|_p$.

Theorem 4.6. Suppose that \mathbf{x} is s -sparse. Define the following constants:

$$\hat{\rho} = \frac{\sqrt{2\hat{\delta}_{3s}^2(1 + \hat{\delta}_{3s}^2)}}{1 - \hat{\delta}_{3s}^2} \quad \text{and} \quad \hat{\tau} = \frac{(\sqrt{2} + 2)\hat{\delta}_{3s}}{\sqrt{1 - \hat{\delta}_{3s}^2}}(1 - \hat{\delta}_{3s})(1 - \hat{\rho}) + \frac{2\sqrt{2} + 1}{(1 - \hat{\delta}_{3s})(1 - \hat{\rho})}.$$

Suppose that $\hat{\delta}_{3s} \leq 0.4859$. Then after $m = \ln(\epsilon_\Phi + \epsilon_y)/\ln(\hat{\rho})$ iterations of Subspace Pursuit (Algorithm 3) applied to problem (4.7), we have

$$\frac{\|\mathbf{x} - \mathbf{x}^m\|_2}{\|\mathbf{x}\|_2} \leq \left(\hat{\tau} \frac{\sqrt{1 + \hat{\delta}_s}}{1 - \epsilon_\Phi^s} + 1 \right) (\epsilon_\Phi^s + \epsilon_y).$$

Remark 4.7. This is Theorem 2 in [34], adapted to the case the \mathbf{x} is s -sparse (the result *in. loc. cit.* is for the more general case where \mathbf{x} is compressible).

5 The RIP of Laplacian of Random Graphs

We first study the RIP for the Laplacian of a connected ER graph drawn from $\mathcal{G}(n_0, p)$. We then extend this to a result on the RIP for graphs drawn from the SBM $\mathcal{G}(n, k, p, 0)$, as these can be thought of as a disjoint union of k ER graphs. Finally, we extend to graphs drawn from $\mathcal{G}(n, k, p, q)$ for $0 < q \ll p$ using a perturbation argument.

5.1 RIP for Laplacian of $\mathcal{G}(n_0, p)$

Lemma 5.1. Let \mathbf{L} be the Laplacian of a connected graph G with n_0 vertices. Let $S \subset [n_0]$ with $|S| = s < n_0$. Then $\sigma_{\min}(\mathbf{L}_S) \geq (1 - s/n_0)\lambda_2$ and $\delta_s = \max\{1 - (1 - s/n_0)\lambda_2^2, \lambda_{n_0}^2 - 1\}$, where λ_i denotes the i -th eigenvalue of \mathbf{L} , ordered from smallest to largest.

PROOF. Let $\mathbf{w}_1, \dots, \mathbf{w}_{n_0}$ be an orthonormal basis of eigenvectors of \mathbf{L} with eigenvalues $\lambda_1 \leq \dots \leq \lambda_{n_0}$, where $\lambda_1 = 0$ and $\lambda_2 > 0$. For any \mathbf{v} supported on S with $\|\mathbf{v}\|_2 = 1$, write $\mathbf{v} = \sum_{i=1}^{n_0} \alpha_i \mathbf{w}_i$. Note that $\mathbf{w}_1 = \frac{1}{\sqrt{n_0}}\mathbf{1}$ and $\ker(L) = \text{span}(\mathbf{w}_1)$. Then:

$$\|\mathbf{L}\mathbf{v}\|_2^2 = \left\| \mathbf{L} \left(\sum_{i=1}^n \alpha_i \mathbf{w}_i \right) \right\|_2^2 = \left\| \sum_{i=2}^n \alpha_i \lambda_i \mathbf{w}_i \right\|_2^2 = \sum_{i=2}^n \alpha_i^2 \lambda_i^2 \geq \left(\sum_{i=2}^n \alpha_i^2 \right) \lambda_2^2$$

Because $\sum_{i=1}^{n_0} \alpha_i^2 = \|\mathbf{v}\|_2^2 = 1$ we have that $\sum_{i=2}^{n_0} \alpha_i^2 = 1 - \alpha_1^2$. Thus $\|\mathbf{L}\mathbf{v}\|_2^2 \geq \lambda_2^2(1 - \alpha_1^2)$ and so clearly this quantity is minimized by making α_1 as large as possible. Observe that:

$$\alpha_1 = \left(\frac{1}{\sqrt{n_0}}\mathbf{1} \right) \cdot \mathbf{v} \leq \frac{1}{\sqrt{n_0}} \|\mathbf{v}\|_1 \leq \frac{1}{\sqrt{n_0}} (\sqrt{s} \|\mathbf{v}\|_2) = \frac{\sqrt{s}}{\sqrt{n_0}}.$$

We remark that this bound on α_1 is sharp, and is achieved by taking $\mathbf{v} = \frac{1}{\sqrt{s}}\mathbf{1}_S$. Hence:

$$\sigma_{\min}(\mathbf{L}_S) = \min_{\substack{\text{supp}(\mathbf{v}) \subset S \\ \|\mathbf{v}\|_2=1}} \|\mathbf{L}\mathbf{v}\|_2^2 \geq \left(1 - \left(\sqrt{\frac{s}{n_0}} \right)^2 \right) \lambda_2^2 = \left(1 - \frac{s}{n_0} \right) \lambda_2^2.$$

On the other hand:

$$\max_{\substack{\text{supp}(\mathbf{v}) \subset S \\ \|\mathbf{v}\|_2=1}} \|\mathbf{L}\mathbf{v}\|_2^2 \leq \max_{\|\mathbf{v}\|_2=1} \|\mathbf{L}\mathbf{v}\|_2^2 = \lambda_{n_0}^2.$$

Hence for any $S \subset [n]$ with $|S| = s$, and any \mathbf{v} with $\text{supp}(\mathbf{v}) \subset S$ and $\|\mathbf{v}\|_2 = 1$, we have:

$$\left(1 - \frac{s}{n_0}\right) \lambda_2^2 \leq \|L\mathbf{v}\|_2^2 \leq \lambda_{n_0}^2$$

and the claim about δ_s follows. \square

thm:RIPER

Theorem 5.2 (RIP for Laplacian of ER graphs). *Suppose that $G \in \mathcal{G}(n_0, p)$ with Laplacian \mathbf{L} and suppose that $p \gg (\ln(n_0))^2/n_0$. If $s = \gamma n_0$ with $\gamma \in (0, 1)$ then*

$$\delta_s \leq \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)$$

almost surely ⁴

PROOF. By Theorem 3.7, and Remark 3.8,

$$\lambda_2 \geq 1 - \frac{4p^{-1/2}}{\sqrt{n_0}} - o\left(\frac{1}{\sqrt{n_0}}\right) \text{ and } \lambda_{n_0} \leq 1 + \frac{4p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)$$

almost surely. Combining this with Lemma 5.1, we get that:

$$\begin{aligned} \delta_s &\leq 1 - (1 - s/n_0) \left(1 - \frac{4p^{-1/2}}{\sqrt{n_0}} - o\left(\frac{1}{\sqrt{n_0}}\right)\right)^2 \\ &= 1 - (1 - \gamma) \left(1 - \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)\right) \\ &= 1 - \left(1 - \gamma - (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)\right) = \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right). \end{aligned}$$

as claimed. \square

We conclude this subsection with a lower bound for $\|\mathbf{L}\|_{2,s}$ (see §4.3 for the definition of $\|\cdot\|_{2,s}$), where \mathbf{L} is the Laplacian of a connected graph.

ISemiNorm

Lemma 5.3. *If \mathbf{L} is the Laplacian of a connected graph G with n_0 vertices, then $\|\mathbf{L}\|_{2,s} \geq \lambda_{s-1}$.*

PROOF. Recall that:

$$\|L\|_{2,s} := \max_{\substack{S \subset [n] \\ |S|=s}} \|L_S\|_2 = \max_{\substack{S \subset [n] \\ |S|=s}} \sigma_{\max}(L_S) \quad (5.1) \quad \text{newnorm}$$

We shall apply Theorem 5.4 below. For a matrix such as \mathbf{L} which is conjugate to a symmetric non-negative definite matrix, the eigenvalues coincide with the singular values. Translating the notation of this Theorem 5.4 into the current situation, for $p = n$, $q = s$, $\beta_i = \sigma_i(\mathbf{L}_S)$ and $\alpha_i = \lambda_i(\mathbf{L})$. Clearly, $\min(m, n) = \min(n, n) = n$ and $\min(p, q) = \min(n, s) = s$. We use Theorem 5.4 again to get:

$$\sigma_{\max}(\mathbf{L}_S) \geq \lambda_{s-1}.$$

And so $\max_{\substack{S \subset [1, \dots, n] \\ |S|=s}} \sigma_{\max}(\mathbf{L}_S) \geq \lambda_{s-1}$. \square

In the proof above, we have used the following classic interpolation Theorem for singular values:

⁴Recall that here and throughout this paper we say that a property holds almost surely if the probability that it does not hold is $o(1)$ with respect to n_0 (or sometimes n)

Theorem 5.4. Let \mathbf{A} be an $m \times n$ matrix with singular values $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_{\min(m,n)}$. Suppose that \mathbf{B} is a $p \times q$ submatrix of \mathbf{A} , with singular values $\beta_1 \leq \beta_2 \leq \dots \leq \beta_{\min(p,q)}$. Then:

$$\alpha_{\min(m,n)-i} \geq \beta_{\min(p,q)-i} \text{ for } i = 0, 1, \dots, \min(p, q) - 1$$

and

$$\beta_i \geq \alpha_{\min(m,n)-\min(p,q)-(m-p)-(n-q)-1+i}$$

for integer i satisfying $\min(p, q) + (m - p) + (n - q) + 2 - \min(m, n) \leq i \leq \min(m, n)$.

PROOF. This is Theorem 1 in [44]. Note that they use the opposite notational convention: $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_{\min(m,n)}$ and $\beta_1 \geq \beta_2 \geq \dots \geq \beta_{\min(p,q)}$. \square

5.2 RIP for Laplacian of Graphs from $\mathcal{G}(n, k, p, 0)$

Lemma 5.5. Suppose that a graph G has k connected components C_1, \dots, C_k , all of size n_0 (for example, $G \in \mathcal{G}(n, k, p, 0)$). Let G_{C_1}, \dots, G_{C_k} denote the subgraphs on these components and let \mathbf{L}^i denote their Laplacians. Then for any $s < n_0$, $\delta_s(\mathbf{L}_G) = \max_i \delta_s(\mathbf{L}^i)$.

PROOF. Suppose $S \subset C_i$ for some i . For simplicity we assume $i = 1$, but the other cases are identical. In this case $\mathbf{L}_S = \begin{bmatrix} \mathbf{L}_S^1 \\ \mathbf{0} \end{bmatrix}$ where \mathbf{L}^1 denotes the Laplacian of G_{C_1} and $\mathbf{0}$ here is the zero matrix of the appropriate size. If $\text{supp}(\mathbf{v}) \subset S$, then $\|\mathbf{L}_S \mathbf{v}\|_2^2 = \|\mathbf{L}_S^1 \mathbf{v}_s\|_2^2$ and so:

$$(1 - \delta_s(\mathbf{L}^1)) \|\mathbf{v}\|_2^2 \leq \|\mathbf{L}_S \mathbf{v}\|_2^2 \leq (1 + \delta_s(\mathbf{L}^1)) \|\mathbf{v}\|_2^2 \quad (5.2)$$

eq:Contained

It follows that, for all index sets S contained in a single component (i.e. $S \subset C_i$ for some i), we have:

$$(1 - \max_i (\delta_s(\mathbf{L}^i))) \|\mathbf{v}\|_2^2 \leq \|\mathbf{L}_S \mathbf{v}\|_2^2 \leq (1 + \max_i (\delta_s(\mathbf{L}^i))) \|\mathbf{v}\|_2^2.$$

Now suppose that $S \not\subset C_i$. Write $S = \cup_i S_i$ where $S_i := S \cap C_i$. Given any \mathbf{v} with $\text{supp}(\mathbf{v}) = S$, we can write $\mathbf{v} = \sum_{i=1}^k \mathbf{v}_i$ with $\mathbf{v}_i = \mathcal{H}_{S_i}(\mathbf{v})$. Then:

$$\|\mathbf{L}_S \mathbf{v}\|_2^2 = \left\| \sum_{i=1}^k \mathbf{L}_{S_i} \mathbf{v}_i \right\|_2^2$$

Crucially, observe that all the terms $\mathbf{L}_{S_i} \mathbf{v}_i$ have disjoint support. Hence:

$$\begin{aligned} \|\mathbf{L}_S \mathbf{v}\|_2^2 &= \left\| \sum_i \mathbf{L}_{S_i} \mathbf{v}_i \right\|_2^2 = \sum_i \|\mathbf{L}_{S_i} \mathbf{v}_i\|_2^2 \geq \sum_i (1 - \delta_{s_i}(\mathbf{L}^i)) \|\mathbf{v}_i\|_2^2 \text{ by (5.2)} \\ &\geq \min_i (1 - \delta_{s_i}(\mathbf{L}^i)) \sum_i \|\mathbf{v}_i\|_2^2 = \min_i (1 - \delta_{s_i}(\mathbf{L}^i)) \|\mathbf{v}\|_2^2 \\ &= \left(1 - \max_i \delta_{s_i}(\mathbf{L}^i) \right) \|\mathbf{v}\|_2^2 \geq \left(1 - \max_i \delta_s(\mathbf{L}^i) \right) \|\mathbf{v}\|_2^2 \end{aligned}$$

with the final inequality holding as $s_i \leq s$ for all i and δ_t is non-decreasing in t . An identical argument yields that:

$$\|\mathbf{L}_S \mathbf{v}\|_2^2 \leq \left(1 + \max_i \delta_s(\mathbf{L}^i) \right) \|\mathbf{v}\|_2^2$$

and so we have

$$(1 - \max_i (\delta_s(\mathbf{L}^i))) \|\mathbf{v}\|_2^2 \leq \|\mathbf{L}_S \mathbf{v}\|_2^2 \leq (1 + \max_i (\delta_s(\mathbf{L}^i))) \|\mathbf{v}\|_2^2.$$

This completes the proof. \square

IPBoundq0

Theorem 5.6 (RIP for Laplacian of Graphs from Stochastic Block Model). *Suppose $G \in \mathcal{G}(n, k, p, 0)$, with $n_0 = n/k$ and Laplacian \mathbf{L} . Suppose further that $p \gg (\ln(n_0))^2/n_0$ and that k is $O(1)$ with respect to n . If $s = \gamma n_0$ with $\gamma \in (0, 1)$ then:*

$$\delta_s(\mathbf{L}) \leq \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right) \quad (5.3)$$

almost surely.

PROOF. Because $q = 0$, G will have k connected components with probability 1. Note that each subgraph G_i is an i.i.d ER graph, drawn from $\mathcal{G}(n_0, p)$. Let \mathbf{L}^i denote the Laplacian of G_{C_i} . By Theorem 5.2

$$\delta_s(\mathbf{L}^i) \leq \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right) \quad (5.4)$$

eq:OneBlockR

almost surely. That is, there exists a function $f(n_0)$ going to 0 as $n_0 \rightarrow \infty$ such that (5.4) holds with probability $1 - f(n_0)$. As the G_{C_i} are i.i.d:

$$\max_{1 \leq i \leq k} \delta_s(\mathbf{L}^i) \leq \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right) \quad (5.5)$$

eq:ManyBlockR

with probability $(1 - f(n_0))^k$. As long as k is eventually bounded with respect to n_0 (i.e. k is $O(1)$ with respect to n), $(1 - f(n_0))^k \rightarrow 1$ as $n_0 \rightarrow \infty$. Hence (5.5) holds almost surely. \square

Again, we conclude this subsection with a lower bound for $\|\mathbf{L}\|_{2,s}$.

Lemma 5.7. *Suppose that a graph G has k connected components C_1, \dots, C_k , all of size n_0 . Let G_{C_1}, \dots, G_{C_k} denote the subgraphs on these components and let \mathbf{L}^i denote their Laplacians. Then $\|\mathbf{L}\|_{2,s} \geq \max_{a \in [k]} \lambda_{s-1}(\mathbf{L}^a)$.*

SemiNorm2

PROOF. We leave the proof to the interested reader. \square

5.3 RIP for Laplacian of Graphs from $\mathcal{G}(n, k, p, q)$ with $q > 0$

q_nonzero

Finally, we study the RIP for Laplacian of graphs from $\mathcal{G}(n, k, p, q)$ with $q > 0$. Observe that for $G \in \mathcal{G}(n, k, p, q)$ we may write $\mathbf{A} = \mathbf{A}^0 + \mathbf{A}^\epsilon$ where \mathbf{A}^0 contains the intracluster edges and \mathbf{A}^ϵ contains the intercluster edges:

$$\mathbf{A}_{ij}^0 = \begin{cases} 1 & \text{if } \{i, j\} \in E \text{ and } i, j \in C_a \text{ for an } a \in \{1, \dots, k\} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{A}_{ij}^\epsilon = \begin{cases} 1 & \text{if } \{i, j\} \in E \text{ and } i \in C_a \text{ and } j \in C_b \text{ for some } a \neq b \\ 0 & \text{otherwise.} \end{cases}$$

Assuming that $q \ll p$, there are much fewer intercluster than intracluster edges, and thus \mathbf{A}^ϵ is much sparser than \mathbf{A}^0 . Observe further that we can regard \mathbf{A}^0 as the adjacency matrix of a subgraph $G^0 \subset G$, obtained by removing all the intercluster edges. In fact, $G^0 \in \mathcal{G}(n, k, p, 0)$. Let \mathbf{L}^0 denote the Laplacian of the underlying graph G^0 . It is tempting to assume that $\mathbf{L} = \mathbf{L}^0 + \mathbf{L}^\epsilon$, where \mathbf{L}^ϵ is a

Laplacian associated to A^ϵ , but unfortunately this is not the case. However, one can still regard \mathbf{L} as a small perturbation of \mathbf{L}^0 , i.e. $\mathbf{L} = \mathbf{L}^0 + \mathbf{E}$. In order to establish bounds on the restricted isometry constants of \mathbf{L} , we need to bound the size of the perturbation \mathbf{E} , which we do in Theorem 5.8. First, recall that, as in §3.3, for any $i \in C_a$, d_i^0 denotes the in-community degree (equivalently the degree of i in G_{C_a} or G^0) and d_i^ϵ denotes the out-of-community degree. In terms of \mathbf{A}^0 and \mathbf{A}^ϵ , $d_i^0 = \sum_j \mathbf{A}_{ij}^0$ and $d_i^\epsilon = \sum_j \mathbf{A}_{ij}^\epsilon$. Define $r_i = d_i^\epsilon/d_i^0$. We can assume that $d_i^\epsilon \ll d_i^0$, equivalently $r_i \ll 1$, for all i .

Theorem 5.8. *Let \mathbf{L} be the Laplacian of a graph drawn at random from $\mathcal{G}(n, k, p, q)$ with $p \geq 4k(\ln(n))^2/n$ and let α be as in part 2 of Theorem 3.6. Suppose further that $\max_{1 \leq i \leq n} r_i \leq r \ll 1$. Then $\mathbf{L} = \mathbf{L}^0 + \mathbf{E}$ where \mathbf{L}^0 is the Laplacian of G^0 and $\mathbf{E} = \mathbf{E}^1 + \mathbf{E}^2$ such that $\text{supp}(\mathbf{E}^1) = \text{supp}(\mathbf{A}^0)$ and $\text{supp}(\mathbf{E}^2) = \text{supp}(\mathbf{A}^\epsilon)$ and the following bounds hold:*

1. $\|\mathbf{E}^1\|_\infty, \|\mathbf{E}^2\|_\infty \leq r + \mathcal{O}(r^2)$.
2. $\|\mathbf{E}^1\|_1, \|\mathbf{E}^2\|_1 \leq \beta^2 r + \mathcal{O}(r^2)$
3. $\|\mathbf{E}^1\|_2, \|\mathbf{E}^2\|_2 \leq \beta r + \mathcal{O}(r^2)$.
4. for any $s < n$, $\|\mathbf{E}^1\|_{2,s}, \|\mathbf{E}^2\|_{2,s} \leq \beta r + \mathcal{O}(r^2)$ and $\|\mathbf{E}\|_{2,s} \leq 2\beta r + \mathcal{O}(r^2)$

where $\beta^2 = (1 + \alpha)/(1 - \alpha)$.

The proof of Theorem 5.8 uses the following lemma:

Lemma 5.9. *Suppose that \mathbf{L} is the Laplacian of a graph G drawn from $\mathcal{G}(n, k, p, q)$, and let \mathbf{L}^0 denote the Laplacian of the underlying (disconnected) graph G^0 , thought of as drawn from $\mathcal{G}(n, k, p, 0)$. Suppose further that $\max_{1 \leq i \leq n} r_i \leq r \ll 1$. Then $\mathbf{L} = \mathbf{L}^0 + \mathbf{E}$, where:*

$$\mathbf{E}_{ij} = \begin{cases} \frac{1}{d_i^0} (r_i + \mathcal{O}(r^2)) \mathbf{A}_{ij}^0 & \text{if } i, j \in C_a \\ \frac{1}{d_i^0} ((r_i - 1) + \mathcal{O}(r^2)) \mathbf{A}_{ij}^\epsilon & \text{if } i \in C_a \text{ and } j \in C_b \text{ for } a \neq b \end{cases}$$

PROOF. By definition $\mathbf{L}_{ij} = \delta_{ij} - \frac{\mathbf{A}_{ij}}{d_i}$ where $\delta_{ij} = 1$ if $i = j$ and zero otherwise. Observe that $\frac{1}{d_i} = \frac{1}{d_i^0 + d_i^\epsilon} = \frac{1}{d_i^0} \frac{1}{1 + \frac{d_i^\epsilon}{d_i^0}} = \frac{1}{d_i^0} \frac{1}{1 + r_i}$. Because $r_i \ll 1$ for all i , by Taylor's Theorem $\frac{1}{1 + r_i} = \frac{1}{d_i^0} (1 - r_i + \mathcal{O}(r_i^2))$. Direct computation reveals that, for $i, j \in C_a$:

$$\begin{aligned} \mathbf{L}_{ij} &= \delta_{ij} - \frac{1}{d_i^0} (1 - r_i + \mathcal{O}(r_i^2)) \mathbf{A}_{ij}^0 = \left(\delta_{ij} - \frac{\mathbf{A}_{ij}^0}{d_i^0} \right) + \frac{1}{d_i^0} (r_i + \mathcal{O}(r^2)) \mathbf{A}_{ij}^0 \\ &= \mathbf{L}_{ij}^0 + \frac{1}{d_i^0} (r_i + \mathcal{O}(r^2)) \mathbf{A}_{ij}^0 \end{aligned}$$

as $1/d_i^0 \leq r_i = \mathcal{O}(r)$. The calculation for i, j in different clusters is similar. \square

Let us write $\mathbf{E} = \mathbf{E}^1 + \mathbf{E}^2$ where \mathbf{E}^1 will contain the ‘on-diagonal’ perturbation, and \mathbf{E}^2 will contain the ‘off diagonal’ perturbation:

$$\mathbf{E}_{ij}^1 = \begin{cases} \mathbf{E}_{ij} & \text{if } i, j \in C_a, a \in [k] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mathbf{E}_{ij}^2 = \begin{cases} \mathbf{E}_{ij} & \text{if } i \in C_a \text{ and } j \in C_b \text{ for } a \neq b, a, b \in [k] \\ 0 & \text{otherwise} \end{cases}$$

We are now ready to prove Theorem 5.8.

PROOF. [Proof of theorem 5.8] By definition, $\|\cdot\|_\infty$ is the maximum absolute row sum of the matrix in question. For any i , let C_a be the cluster to which it belongs. Then:

$$\sum_j |\mathbf{E}_{ij}^1| = \sum_{j \in C_a} \frac{r_i + \mathcal{O}(r^2)}{d_i^0} \mathbf{A}_{ij}^0 = \frac{r_i + \mathcal{O}(r^2)}{d_i^0} \sum_{j \in C_a} \mathbf{A}_{ij}^0 = \frac{r_i + \mathcal{O}(r^2)}{d_i^0} (d_i^0) = r_i + \mathcal{O}(r^2).$$

Hence, we have

$$\|\mathbf{E}^1\|_\infty = \max_i \left(\sum_j |\mathbf{E}_{ij}^1| \right) = \max_i (r_i) + \mathcal{O}(r^2) \leq r + \mathcal{O}(r^2).$$

Similarly:

$$\sum_j |\mathbf{E}_{ij}^2| = \sum_{j \notin C_a} \frac{|(r_i - 1) + \mathcal{O}(r^2)|}{d_i^0} \mathbf{A}_{ij}^\epsilon \leq \frac{1 + \mathcal{O}(r^2)}{d_i^0} \sum_{j \notin C_a} \mathbf{A}_{ij}^\epsilon = \frac{1 + \mathcal{O}(r^2)}{d_i^0} (d_i^\epsilon) \leq r_i + \mathcal{O}(r^2)$$

as $r_i := d_i^\epsilon / d_i^0$, and so again:

$$\|\mathbf{E}^2\|_\infty = \max_i \left(\sum_j |\mathbf{E}_{ij}^2| \right) = \max_i (r_i) + \mathcal{O}(r^2) \leq r + \mathcal{O}(r^2)$$

Before continuing, recall that as $p \geq 4k(\ln(n))^2/n$ we may choose an $\alpha = o(1)$ with respect to n such that $(1 - \alpha)n_0p \leq d_i^0 \leq (1 + \alpha)n_0p$ almost surely for all $i \in [n]$, by part 2 of Theorem 3.6. Now, because $\|\cdot\|_1$ is the maximum absolute column sum, for any j in cluster C_a , we have

$$\begin{aligned} \sum_{1 \leq i \leq n} |\mathbf{E}_{ij}^1| &= \sum_{i \in C_a} \frac{r_i + \mathcal{O}(r^2)}{d_i^0} \mathbf{A}_{ij}^0 \leq \frac{r + \mathcal{O}(r^2)}{d_{\min}^0} \sum_{i \in C_a} \mathbf{A}_{ij}^0 \leq \frac{r + \mathcal{O}(r^2)}{d_{\min}^0} (d_{\max}^0) \\ &\leq (r + \mathcal{O}(r^2)) \frac{(1 + \alpha)n_0p}{(1 - \alpha)n_0p} = \frac{r(1 + \alpha)}{(1 - \alpha)} + \mathcal{O}(r^2) = \beta^2 r + \mathcal{O}(r^2) \end{aligned}$$

and so:

$$\|\mathbf{E}^1\|_1 = \max_j \left(\sum_i |\mathbf{E}_{ij}^1| \right) \leq \beta^2 r + \mathcal{O}(r^2).$$

Similarly;

$$\begin{aligned} \sum_{1 \leq i \leq n} |\mathbf{E}_{ij}^2| &= \sum_{i \notin C_a} \frac{|(r_i - 1) + \mathcal{O}(r^2)|}{d_i^0} \mathbf{A}_{ij}^\epsilon \leq \frac{1 + \mathcal{O}(r^2)}{d_{\min}^0} \sum_{i \notin C_a} \mathbf{A}_{ij}^\epsilon = (1 + \mathcal{O}(r^2)) \frac{d_j^\epsilon}{d_{\min}^0} \\ &\leq (1 + \mathcal{O}(r^2)) \frac{1 + \alpha}{1 - \alpha} \frac{d_j^\epsilon}{d_{\max}^0} \leq (1 + \mathcal{O}(r^2)) \frac{1 + \alpha}{1 - \alpha} \frac{d_j^\epsilon}{d_j^0} \leq \beta^2 r_j + \mathcal{O}(r^2) \end{aligned}$$

and so again we have

$$\|\mathbf{E}^2\|_1 = \max_j \left(\sum_i |\mathbf{E}_{ij}^2| \right) \leq \beta^2 \max_j (r_j) + \mathcal{O}(r^2) \leq \beta^2 r + \mathcal{O}(r^2).$$

Now, by the Riesz-Thorin interpolation theorem:

$$\|\mathbf{E}^1\|_2 \leq \|\mathbf{E}^1\|_1^{1/2} \|\mathbf{E}^1\|_\infty^{1/2} \leq (\beta^2 r + \mathcal{O}(r^2))^{1/2} (r + \mathcal{O}(r^2))^{1/2} = \beta r + \mathcal{O}(r^2)$$

and by an identical argument $\|\mathbf{E}^2\|_2 \leq \beta r + \mathcal{O}(r^2)$. Part 4 follows from the simple fact that for any matrix \mathbf{B} , $\|\mathbf{B}\|_{2,s} \leq \|\mathbf{B}\|_2$, and the bound for $\|\mathbf{E}\|_{2,s}$ follows from the fact that $\mathbf{E} = \mathbf{E}^1 + \mathbf{E}^2$ and the triangle inequality. \square

We are now ready to state and prove the main result of this section.

Twostory **Theorem 5.10.** *Suppose that $G \in \mathcal{G}(n, k, p, q)$, with block sizes $n_0 = n/k$ and Laplacian \mathbf{L} . Suppose further that:*

1. $p \gg (\ln(n_0))^2/n_0$
2. k is $\mathcal{O}(1)$ with respect to n .
3. $\max_{1 \leq i \leq n} r_i \leq r < 1$.

If $t = \gamma n_0$ with $\gamma \in (0, 1)$ then:

$$\delta_t(\mathbf{L}) \leq \gamma + D_1 r + \frac{D_2}{\sqrt{n_0}} + D_3 \frac{r}{\sqrt{n_0}} + \mathcal{O}(r^2) + o\left(\frac{1}{\sqrt{n_0}}\right) \quad (5.6)$$

eq:Twostory

almost surely, where D_1 , D_2 and D_3 are $\mathcal{O}(1)$ with respect to n_0

PROOF. As above, we write $\mathbf{L} = \mathbf{L}^0 + \mathbf{E}$, where \mathbf{L}^0 is the Laplacian of the subgraph G^0 , thought of as drawn from $\mathcal{G}(n, k, p, 0)$. We define $\delta_t := \delta_t(\mathbf{L}^0)$ and $\hat{\delta}_t := \delta_t(\mathbf{L})$. By Theorem 4.5:

$$\hat{\delta}_t \leq (1 + \delta_t)(1 + \epsilon_{\mathbf{L}}^t)^2 - 1. \quad (5.7)$$

eq:deltaTilde

As defined in a previous section, $\epsilon_{\mathbf{L}}^t := \|\mathbf{E}\|_{2,t}/\|\mathbf{L}^0\|_{2,t}$. Theorem 5.8 gives us that $\|\mathbf{E}\|_{2,t} \leq 2\beta r + \mathcal{O}(r^2)$, while Lemma 5.7 gives $\|\mathbf{L}^0\|_{2,t} \geq \max_{1 \leq a \leq k} \lambda_{t-1}(\mathbf{L}^a)$ (recall that \mathbf{L}^a denotes the Laplacian of the subgraph G_{C_a}). By a similar argument to that of Theorem 5.6, we can apply Lemma 3.7 (and see also remark 3.8) to get:

$$\max_{1 \leq a \leq k} \lambda_{t-1}(\mathbf{L}^a) \geq 1 - \frac{4p^{-1/2}}{\sqrt{n_0}} - o\left(\frac{1}{\sqrt{n_0}}\right)$$

almost surely, assuming that k is eventually bounded with respect to n_0 . For convenience, choose n_0 large enough such that $4p^{-1/2}/\sqrt{n_0} + o(\frac{1}{\sqrt{n_0}}) \leq 1/4$. Under this assumption:

$$\epsilon_{\mathbf{L}^0}^t = \frac{\|\mathbf{E}\|_{2,t}}{\|\mathbf{L}^0\|_{2,t}} \leq \frac{2\beta r + \mathcal{O}(r^2)}{1 - 1/4} = \frac{8\beta r}{3} + \mathcal{O}(r^2)$$

Combining this with (5.7) and Theorem 5.6:

$$\begin{aligned} \hat{\delta}_s &\leq \left(1 + \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)\right) \left(1 + \frac{8\beta r}{3} + \mathcal{O}(r^2)\right)^2 - 1 \\ &= \left(1 + \gamma + (1 - \gamma) \frac{8p^{-1/2}}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)\right) \left(1 + \frac{16\beta r}{3} + \mathcal{O}(r^2)\right) - 1 \\ &= \gamma + r \left(\frac{16\beta(1 + \gamma)}{3}\right) + \frac{1}{\sqrt{n_0}} \left(8(1 - \gamma)p^{-1/2}\right) + \frac{r}{\sqrt{n_0}} \left(\frac{128(1 - \gamma)p^{-1/2}\beta}{3}\right) + \mathcal{O}(r^2) + o\left(\frac{1}{\sqrt{n_0}}\right) \end{aligned}$$

and letting $D_1 = 16\beta(1 + \gamma)/3$, $D_2 = 8(1 - \gamma)p^{-1/2}$, $D_3 = 128(1 - \gamma)p^{-1/2}\beta^2/3$ gives equation (5.6). Finally, because $\beta \rightarrow 1$ as $n \rightarrow \infty$, we indeed have that D_1 , D_2 and D_3 are $\mathcal{O}(1)$ with respect to n_0 . \square

6 The Coherence Properties for Laplacians of Random Graphs

properties

In this section we study the coherence properties of Laplacians of random graphs. Mainly, we compute a lower bound for $|\langle \ell_i, \ell_j \rangle|$ when i and j are in the same cluster and hence a lower bound for $\mu(\mathbf{L}_{-1})$. We begin with a basic result for graphs from $\mathcal{G}(n, k, p, 0)$.

Theorem 6.1. *Let \mathbf{A} be the adjacency matrix of a graph G^0 drawn at random from $\mathcal{G}(n, k, p, 0)$ and define $\chi_{i,j} = \sum_k \mathbf{A}_{ik} \mathbf{A}_{kj}$. Suppose further that there exists an $\alpha \in (0, 1)$ such that $(1 - \alpha)pn_0 \leq d_i^0 \leq (1 + \alpha)pn_0$. Then for any $i, j \in C_1$ with $i \neq j$:*

$$(1 - \alpha)n_0p^2 - \sqrt{\frac{(1 + \alpha)n_0p \ln(1/\delta)}{2}} \leq \chi_{i,j} \leq (1 + \alpha)n_0p^2 + \sqrt{\frac{(1 + \alpha)n_0p \ln(1/\delta)}{2}} \quad (6.1)$$

chi1bound

with probability $1 - \delta$.

PROOF. Let $B_1(i)$ denote the ball of radius 1 centered at i . That is, $B_1(i) := \{k \in [n] : \{i, k\} \in E\}$. By definition $|B_1(i)| = d_i$. Observe that:

$$\chi_{i,j} = \sum_k \mathbf{A}_{ik} \mathbf{A}_{kj} = \sum_{k \in B_1(i)} \mathbf{A}_{kj}$$

Each \mathbf{A}_{kj} is an i.i.d. Bernoulli random variable with parameter p . Applying the Chernoff bound to this sum:

$$\mathbb{P} \left[\sum_{k \in B_1(i)} \mathbf{A}_{kj} \leq pd_i - \sqrt{\frac{d_i \ln(1/\delta)}{2}} \right] \leq \exp \left(- \frac{2 \left(\sqrt{\frac{d_i \ln(1/\delta)}{2}} \right)^2}{d_i} \right) = \delta$$

and $(1 - \alpha)pn_0 \leq d_i \leq (1 + \alpha)pn_0$ so:

$$\mathbb{P} \left[\chi_{i,j} \leq (1 - \alpha)n_0p^2 - \sqrt{\frac{(1 + \alpha)n_0p \ln(1/\delta)}{2}} \right] \leq \mathbb{P} \left[\chi_{i,j} \leq pd_i - \sqrt{\frac{d_i \ln(1/\delta)}{2}} \right] \leq \delta$$

The upper bound on $\chi_{i,j}$ is proved analogously. □

We now provide a bound on the inner product $|\langle \ell_i^0, \ell_j^0 \rangle|$ when i and j are in the same cluster:

leL1bound

Theorem 6.2. *Let \mathbf{L}^0 be the Laplacian of a graph G^0 drawn at random from $\mathcal{G}(n, k, p, 0)$. As in Theorem 6.1, suppose that there exists an $\alpha \in (0, 1)$ such that $(1 - \alpha)pn_0 \leq d_i^0 \leq (1 + \alpha)pn_0$ but in addition assume that α is $o(1)$ with respect to n . Define $\beta^2 = (1 + \alpha)/(1 - \alpha)$. Then for any $\delta \in (0, 1)$ and for any $i, j \in C_a$, $a \in [k]$, with $i \neq j$:*

$$|\langle \ell_i^0, \ell_j^0 \rangle| \geq \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{\ln(1/\delta)}{2(1 + \alpha)^3 p^3}}$$

with probability $1 - \delta$ for n large enough.

PROOF. Recall that

$$|\langle \ell_i^0, \ell_j^0 \rangle| = \left| - \left(\frac{1}{d_i} + \frac{1}{d_j} \right) \mathbf{A}_{ij} + \sum_{k=1}^n \frac{\mathbf{A}_{ik} \mathbf{A}_{jk}}{d_k^2} \right|$$

There are two possibilities; either i and j are connected ($\{i, j\} \in E$) or they are not ($\{i, j\} \notin E$). We treat these cases separately. Suppose first that $\{i, j\} \notin E$. In this case, $\mathbf{A}_{ij} = 0$ and so:

$$|\langle \ell_i^0, \ell_j^0 \rangle| = \sum_{k=1}^n \frac{\mathbf{A}_{ik} \mathbf{A}_{jk}}{d_k^2} \geq \frac{1}{d_{\max}^2} \chi_{ij}$$

by assumption, $d_{\max} \leq (1 + \alpha)pn_0$ and by so by Theorem 6.1:

$$\begin{aligned} |\langle \ell_i^0, \ell_j^0 \rangle| &\geq \frac{1}{(1 + \alpha)^2 p^2 n_0^2} \left((1 - \alpha)n_0 p^2 - \sqrt{-\frac{(1 + \alpha)n_0 p \ln(\delta)}{2}} \right) \\ &= \frac{1}{n_0} \left(\frac{1}{(1 + \alpha)\beta^2} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{-\ln(\delta)}{2(1 + \alpha)^3 p^3}} \end{aligned} \quad (6.2) \quad \text{eq:LeadingTer}$$

Alternatively, suppose that $\{i, j\} \in E$. Then $\mathbf{A}_{ij} = 1$ and by Theorem 6.1:

$$\begin{aligned} |\langle \ell_i^0, \ell_j^0 \rangle| &\geq \frac{2}{d_{\max}} - \frac{1}{d_{\min}^2} \chi_{ij} \geq \frac{2}{(1 + \alpha)pn_0} - \frac{1}{(1 - \alpha)^2 p^2 n_0^2} \chi_{ij} \\ &\geq \frac{2}{(1 + \alpha)pn_0} - \frac{1}{(1 - \alpha)^2 p^2 n_0^2} \left((1 + \alpha)n_0 p^2 + \sqrt{-\frac{(1 + \alpha)n_0 p \ln(\delta)}{2}} \right) \\ &= \frac{1}{n_0} \left(\frac{2}{(1 - \alpha)p} - \frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{-\ln(\delta)\beta^2}{2(1 - \alpha)^3 p^3}} \end{aligned} \quad (6.3) \quad \text{eq:LeadingTer}$$

Let us compare the leading terms in (6.2) and (6.3). Because $\alpha \rightarrow 0$ as $n \rightarrow \infty$, $\beta \rightarrow 1$ and thus $1/((1 + \alpha)\beta) \rightarrow 1$. On the other hand, $2/((1 - \alpha)p) - \beta^2/(1 - \alpha) \rightarrow 2/p - 1 > 1$, assuming $p < 1$. Thus, for n large enough, the right hand side of (6.3) is larger than the right hand side of (6.2), and so we can summarize both bounds as:

$$|\langle \ell_i^0, \ell_j^0 \rangle| \geq \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{-\ln(\delta)}{2(1 + \alpha)^3 p^3}}$$

for large enough n . □

Next we consider the case where intercluster edges are present, i.e. where $q > 0$.

Theorem 6.3. *Let \mathbf{L} be the Laplacian of G drawn from $\mathcal{G}(n, k, p, q)$, $q > 0$, and suppose that $i, j \in C_1$. Assume further that:*

1. $p \geq 4k(\ln(n))^2/n$
2. $\max_i r_i \leq r$ with $r = r_0/\sqrt{n_0}$ where $r_0 = \mathcal{O}(1)$ with respect to n_0 .

Then almost surely:

$$|\langle \ell_i, \ell_j \rangle| \geq \frac{1}{n_0} \left(\frac{\beta^2}{1 + \alpha} \right) - o\left(\frac{1}{n_0}\right).$$

where α is $o(1)$ with respect to n and $\beta^2 = (1 + \alpha)/(1 - \alpha)$.

PROOF. As before, let \mathbf{e}_i^1 (resp. \mathbf{e}_j^1) denote the i -th (resp. j -th) column of \mathbf{E}^1 , while \mathbf{e}_i^2 (resp. \mathbf{e}_j^2) denote the i -th (resp. j -th) column of \mathbf{E}^2 . Then:

$$\begin{aligned} \langle \ell_i, \ell_j \rangle &= \langle \ell_i^0 + \mathbf{e}_i^1 + \mathbf{e}_i^2, \ell_j^0 + \mathbf{e}_j^1 + \mathbf{e}_j^2 \rangle \\ &= \langle \ell_i^0, \ell_j^0 \rangle + \langle \ell_i^0, \mathbf{e}_j^1 \rangle + \langle \ell_i^0, \mathbf{e}_j^2 \rangle + \langle \mathbf{e}_i^1, \ell_j^0 \rangle + \langle \mathbf{e}_i^1, \mathbf{e}_j^1 \rangle + \langle \mathbf{e}_i^1, \mathbf{e}_j^2 \rangle + \langle \mathbf{e}_i^2, \ell_j^0 \rangle + \langle \mathbf{e}_i^2, \mathbf{e}_j^1 \rangle + \langle \mathbf{e}_i^2, \mathbf{e}_j^2 \rangle \end{aligned}$$

By construction, \mathbf{e}_i^1 and \mathbf{e}_j^2 have disjoint support (and similarly for \mathbf{e}_j^1 and \mathbf{e}_i^2), as do ℓ_i^0 and \mathbf{e}_j^2 (and similarly ℓ_j^0 and \mathbf{e}_i^2). Hence:

$$\langle \ell_i^0, \mathbf{e}_j^2 \rangle = \langle \mathbf{e}_i^1, \mathbf{e}_j^2 \rangle = \langle \mathbf{e}_i^2, \ell_j^0 \rangle = \langle \mathbf{e}_i^2, \mathbf{e}_j^1 \rangle = 0$$

and so:

$$|\langle \ell_i, \ell_j \rangle| \geq |\langle \ell_i^0, \ell_j^0 \rangle| - |\langle \ell_i^0, \mathbf{e}_j^1 \rangle| - |\langle \mathbf{e}_i^1, \ell_j^0 \rangle| - |\langle \mathbf{e}_i^1, \mathbf{e}_j^2 \rangle| - |\langle \mathbf{e}_i^2, \mathbf{e}_j^2 \rangle| \quad (6.4)$$

Now ℓ_i^0 and ℓ_j^0 can be thought of as columns in the Laplacian of a graph drawn at random from $\mathcal{G}(n, k, p, 0)$. By part 2 of Theorem 3.6 we may choose an $\alpha = o(1)$ with respect to n such that $(1 - \alpha)pn_0 \leq d_i^0 \leq (1 + \alpha)pn_0$ for all i , with probability $1 - 1/n$. Now we apply Theorem 6.2 to get:

$$|\langle \ell_i^0, \ell_j^0 \rangle| \geq \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{-\ln(\delta)}{2(1 + \alpha)^3 p^3}}$$

with probability $1 - \delta - 1/n$. Taking $\delta = e^{-p^3 \sqrt{n_0}}$ (and noting $e^{-p^3 \sqrt{n_0}} < 1/n_0$ for large enough n_0)

$$\begin{aligned} \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{-\ln(\delta)}{2(1 + \alpha)^3 p^3}} &= \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{3/2}} \sqrt{\frac{n_0^{1/2}}{2(1 + \alpha)^3}} \\ &= \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{1}{n_0^{5/4}} \frac{1}{\sqrt{2}(1 + \alpha)^{3/2}} = \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - o(1/n_0) \end{aligned}$$

with probability at least $1 - 1/n_0 - 1/n = 1 - (k + 1)/n$. Next, we consider the term $\langle \ell_i^0, \mathbf{e}_j^1 \rangle$.

$$\begin{aligned} |\langle \ell_i^0, \mathbf{e}_j^1 \rangle| &= \left| \sum_m \left(\delta_{im} - \frac{\mathbf{A}_{im}^0}{d_m^0} \right) \left(\frac{r_m \mathbf{A}_{mj}^0}{d_m^0} + \mathcal{O}(r^3) \right) \right| = \left| \frac{r_i \mathbf{A}_{ij}^0}{d_i^0} - \sum_m \frac{r_m \mathbf{A}_{im}^0 \mathbf{A}_{mj}^0}{(d_m^0)^2} + \mathcal{O}(r^3) \right| \\ &\leq \frac{r}{d_{\min}^0} + \frac{r}{(d_{\min}^0)^2} \sum_m \mathbf{A}_{im}^0 \mathbf{A}_{mj}^0 + \mathcal{O}(r^3) \leq \frac{r}{d_{\min}^0} + \frac{r}{(d_{\min}^0)^2} \min\{d_i^0, d_j^0\} + \mathcal{O}(r^3) \\ &\leq \frac{r}{d_{\min}^0} \left(1 + \frac{d_{\max}^0}{d_{\min}^0} \right) + \mathcal{O}(r^3) \end{aligned}$$

As stated earlier, $d_{\min}^0 \geq (1 - \alpha)pn_0$ and $d_{\max}^0 \leq (1 + \alpha)pn_0$ with probability $1 - 1/n$, and so:

$$|\langle \ell_i^0, \mathbf{e}_j^1 \rangle| \leq \frac{r}{(1 - \alpha)pn_0} \left(1 + \frac{(1 + \alpha)pn_0}{(1 - \alpha)pn_0} \right) + \mathcal{O}(r^3) = \frac{r}{(1 - \alpha)pn_0} (1 + \beta^2) + \mathcal{O}(r^3)$$

Clearly the same bound holds for $|\langle \mathbf{e}_i^1, \ell_j^0 \rangle|$. Finally, consider $|\langle \mathbf{e}_i^1, \mathbf{e}_j^1 \rangle|$. A similar calculation to the one above reveals that:

$$|\langle \mathbf{e}_i^1, \mathbf{e}_j^1 \rangle| \leq \frac{r^2}{n_0} \left(\frac{\beta^4}{(1 - \alpha)p} \right) + \mathcal{O}(r^4)$$

as well as:

$$|\langle \mathbf{e}_i^2, \mathbf{e}_j^2 \rangle| \leq \frac{r}{(1 - \alpha)pn_0} + \mathcal{O}(r^3)$$

Putting all these bounds back in to equation (6.4), we get:

$$\begin{aligned} |\langle \ell_i, \ell_j \rangle| &\geq \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - o\left(\frac{1}{n_0}\right) - \left(\frac{2r(1 + \beta^2)}{(1 - \alpha)pn_0} + \mathcal{O}(r^3) \right) + \frac{r^2}{n_0} \left(\frac{\beta^4}{(1 - \alpha)p} \right) + \mathcal{O}(r^4) + \frac{r}{(1 - \alpha)pn_0} + \mathcal{O}(r^3) \\ &= \frac{1}{n_0} \left(\frac{\beta^2}{(1 + \alpha)} \right) - \frac{r}{n_0} \left(\frac{2(1 + \beta^2)}{(1 - \alpha)p} + \frac{r\beta^4}{(1 - \alpha)p} + \frac{1}{(1 - \alpha)p} \right) + \mathcal{O}(r^3) \end{aligned}$$

Using the assumption that $r = r_0/\sqrt{n_0}$, we see that $r/n_0 = r_0/n_0^{3/2}$ which is of order $o(1/n_0)$. Similarly $r^3 = r_0^3/n_0^{3/2}$ which is also of order $o(1/n_0)$. Thus:

$$|\langle \ell_i, \ell_j \rangle| \geq \frac{1}{n_0} \left(\frac{\beta^2}{1+\alpha} \right) - o\left(\frac{1}{n_0}\right)$$

with probability $1 - \mathcal{O}(1/n_0)$, as claimed. \square

coherence

Remark 6.4. Since the coherence μ of \mathbf{L} is defined as $\mu = \max_{i \neq j} |\langle \ell_i, \ell_j \rangle|$. Theorem 6.3 also implies that $\mu \geq \frac{1}{n_0} \left(\frac{\beta^2}{1+\alpha} \right) - o\left(\frac{1}{n_0}\right)$.

7 Compressive Sensing Clustering for Graphs from $\mathcal{G}(n, k, p, 0)$

Forq_zero

In this section, we consider graphs from $\mathcal{G}(n, k, p, 0)$. This is the case that there are no inter-cluster edges. We may safely assume that each G_{C_i} is connected, as long as $p > \ln(n_0)/n_0$.

Lemma 7.1. *Let \mathbf{L} be the Laplacian of a graph with k connected components C_1, \dots, C_k . Assume, without loss of generality, that vertex $1 \in C_1$. Define $\mathbf{x}^\#$ as:*

$$\mathbf{x}^\# := \arg \min \{ \|\mathbf{x}\|_0 : \mathbf{x} \in \mathbb{R}^n, \mathbf{L}\mathbf{x} = \mathbf{0} \text{ and } x_1 = 1 \} \quad (7.1)$$

eq:ConnCompor

Then $\mathbf{x}^\# = \mathbf{1}_{C_1}$

PROOF. From Theorem 3.3, $\ker(\mathbf{L}) = \text{span}\{\mathbf{1}_{C_1}, \dots, \mathbf{1}_{C_k}\}$ and hence any \mathbf{x} with $\mathbf{L}\mathbf{x} = \mathbf{0}$ can be written as: $\mathbf{x} = \sum_{i=1}^k c_i \mathbf{1}_{C_i}$. If $x_1 = 1$ (recall we are assuming that $1 \in C_1$) then $c_1 = 1$ and so $\mathbf{x} = \mathbf{1}_{C_1} + \sum_{i=2}^k c_i \mathbf{1}_{C_i}$. Recall that all the $\mathbf{1}_{C_i}$ have disjoint support, and so clearly the sparsest solution \mathbf{x} such that $\mathbf{L}\mathbf{x} = \mathbf{0}$ is that \mathbf{x} has $c_2 = c_3 = \dots = c_k = 0$. Hence indeed $\mathbf{x}^\# = \mathbf{1}_{C_1}$ \square

One can rephrase this slightly as a standard compressed sensing problem. Recall ℓ_i is the i -th column of \mathbf{L} and $\mathbf{L}_{-i} := [\ell_1, \dots, \ell_{i-1}, \ell_{i+1}, \dots, \ell_n]$ in which case:

$$x_i = 1 \quad \Leftrightarrow \quad \mathbf{L}_{-i}\mathbf{x}_{-i} = -\ell_i,$$

where if $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top \in \mathbb{R}^n$, $\mathbf{x}_{-i} = [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]^\top \in \mathbb{R}^{n-1}$. Thus problem (7.1) becomes:

$$\mathbf{x}^\# := \arg \min \{ \|\mathbf{x}\|_0 : \mathbf{x} \in \mathbb{R}^{n-1}, \mathbf{L}\mathbf{x} = -\ell_1 \}. \quad (7.2)$$

eq:ConnCompor

We now show that problem (7.2) can be efficiently solved using OMP (Algorithm 2 in §4)

Successq0

Theorem 7.2. *Suppose that $G \in \mathcal{G}(n, k, p, 0)$ has k connected components $C_1, \dots, C_k \subset V$ each of size n_0 . OMP (Algorithm 2) applied to the sparse recovery problem (7.2) will return $\mathbf{x}^\# = \mathbf{1}_{C_1 \setminus \{1\}}$ after $n_0 - 1$ iterations. C_1 can then be recovered as $\{1\} \cup \text{supp}(\mathbf{x}^\#)$.*

Remark 7.3. Note that the assumption that $G \in \mathcal{G}(n, k, p, 0)$ is not necessary; this theorem holds for all graphs with k connected components. Also, observe that this theorem is not probabilistic. It holds with certainty for all graphs G .

We shall appeal to the exact recovery condition, Theorem 4.1 to establish Theorem 7.2. Let us begin with

Injective

Lemma 7.4. *If $C_1 \subset V$ is a connected component then $\mathbf{L}_{C_1 \setminus \{1\}}$ is injective, where $\mathbf{L}_{C_1 \setminus \{1\}}$ is the submatrix from \mathbf{L} with column indices in $C_1 \setminus \{1\}$.*

PROOF. First, observe that $\mathbf{L}_{C_1 \setminus \{1\}} = \begin{bmatrix} \mathbf{L}_{-1}^1 \\ \mathbf{0} \end{bmatrix}$, where \mathbf{L}^1 is the Laplacian of G_{C_1} . It suffices to show that \mathbf{L}_{-1}^1 is injective. As G_{C_1} is connected, $\ker(\mathbf{L}^1) = \text{span}\{\mathbf{1}\}$, by Theorem 3.2. Suppose there exists a $\mathbf{u} \in \mathbb{R}^{n_0-1}$, $\mathbf{u} \neq \mathbf{0}$ such that $\mathbf{L}_{-1}^1 \mathbf{u} = \mathbf{0}$. Then $[0, \mathbf{u}^\top]^\top$ is in $\ker(\mathbf{L}^1) = \text{span}\{\mathbf{1}\}$, a contradiction. \square

Component

Theorem 7.5. *With notation as above (i.e. C_1 is the connected component of G containing the first vertex) we have that:*

$$\|\mathbf{L}_{C_1 \setminus \{1\}}^\dagger \mathbf{L}_{C_1^c}\|_1 = 0 \quad (7.3)$$

eq:Zero

PROOF. By Lemma 7.4 $\mathbf{L}_{C_1 \setminus \{1\}}$ is injective so its pseudo-inverse is given by:

$$\mathbf{L}_{C_1 \setminus \{1\}}^\dagger = \left(\mathbf{L}_{C_1 \setminus \{1\}}^\top \mathbf{L}_{C_1 \setminus \{1\}} \right)^{-1} \mathbf{L}_{C_1 \setminus \{1\}}^\top$$

As observed in the proof of Lemma 7.4, $\mathbf{L}_{C_1 \setminus \{1\}} = \begin{bmatrix} \mathbf{L}_{-1}^1 \\ \mathbf{0} \end{bmatrix}$. Similarly, $\mathbf{L}_{C_1^c} = \begin{bmatrix} \mathbf{0} \\ \tilde{\mathbf{L}} \end{bmatrix}$, where $\tilde{\mathbf{L}}$ denotes the Laplacian of $G_{C_1^c}$. In both cases $\mathbf{0}$ denotes the zero matrix of appropriate size. To show (7.3) it will suffice to show that $\mathbf{L}_{C_1 \setminus \{1\}}^\top \mathbf{L}_{C_1^c} = \mathbf{0}$, but this follows easily as:

$$\begin{bmatrix} (\mathbf{L}_{-1}^1)^\top & \mathbf{0}^\top \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \tilde{\mathbf{L}} \end{bmatrix} = (\mathbf{L}_{-1}^1)^\top \mathbf{0} + \mathbf{0}^\top \tilde{\mathbf{L}} = \mathbf{0}$$

This completes the proof. \square

Theorem 7.2 now follows easily:

PROOF. [Proof of Theorem 7.2] By Theorem 7.5 the exact recover condition (Theorem 4.1) holds, and thus the result follows. \square

8 Compressive Sensing Clustering for Graphs from $\mathcal{G}(n, k, p, q)$

1_nonzero

We now turn our attention to the same problem for $\mathcal{G}(n, k, p, q)$ with $0 < q \ll p$. We follow the notation established in §5.3, and decompose \mathbf{A} as $\mathbf{A} = \mathbf{A}^0 + \mathbf{A}^\epsilon$ where \mathbf{A}^ϵ contains the intercluster edges and \mathbf{A}^0 is the adjacency matrix of the subgraph G^0 with k connected components. Denoting the i -th column of \mathbf{L}^0 as ℓ_i^0 and the i -th column of \mathbf{E} as \mathbf{e}_i , observe that

$$\ell_1 = \ell_1^0 + \mathbf{e}_1 = -\mathbf{L}^0 \mathbf{1}_{C_1 \setminus \{1\}} + \mathbf{e}_1 \quad \text{or} \quad -\ell_1 = \mathbf{L} \mathbf{1}_{C_1 \setminus \{1\}} - \mathbf{e}_1.$$

Defining $\mathbf{x}^\#$ as:

$$\mathbf{x}^\# := \arg \min \{ \|\mathbf{L}_{-1} \mathbf{x} - (-\ell_1)\|_2 : \mathbf{x} \in \mathbb{R}^{n-1}, \|\mathbf{x}\|_0 \leq n_0 \}. \quad (8.1)$$

eq:Compressed

Then $\mathbf{x}^\# \approx \mathbf{1}_{C_1 \setminus \{1\}}$. We recognize this as a totally perturbed sparse recovery problem (4.7), with $\Phi = \mathbf{L}^0$, $\hat{\Phi} = \mathbf{L}^0 + \mathbf{E} = \mathbf{L}$, $\mathbf{y} = -\ell_1 = -\ell_1^0 - \mathbf{e}_1$. We shall show that, provided \mathbf{E} is small enough, $\text{supp}(\mathbf{x}^\#) = \text{supp}(\mathbf{1}_{C_1 \setminus \{1\}}) = C_1 \setminus \{1\}$. Hence, solving problem (8.1) is equivalent to finding the community C_1 .

Unfortunately, a straightforward application of Algorithm 2 to (8.1) does not work. Empirically, it is observed that the first several greedy steps in Algorithm 2 invariably pick up wrong indices due to the presence of noise \mathbf{E} . We need a new approach. Let us start with the following

Mckenzie

Theorem 8.1. *Let \mathbf{L} be the Laplacian of a graph $G \in \mathcal{G}(n, k, p, q)$ with $p \geq 4k(\ln(n))^2/n$. Suppose that*

$$\max_{i \in \{1, \dots, n\}} r_i := \max_{i \in \{1, \dots, n\}} d_i^\epsilon / d_i^0 \leq r, \quad (8.2)$$

as

where $r = r_0 / \sqrt{n_0}$ with r_0 a constant independent of n_0 . Define $\Omega := \mathcal{L}_{\lceil 10(n_0-1)/9 \rceil}(\mathbf{L}_{-1}^\top \ell_1)$. For n_0 large enough, we have that $C_1 \setminus \{1\} \subset \Omega$ almost surely.

PROOF. Suppose otherwise, then there exists an $i^* \in C_1 \setminus \{1\}$ not in Ω . Let $\Lambda = \Omega \cap C_1^c$. As we are assuming that $C_1 \setminus \{1\} \not\subset \Omega$, we have that $|\Lambda| \geq n_0/9$. Moreover, by definition of i^* , we have that $|\langle \ell_1, \ell_{i^*} \rangle| \leq |\langle \ell_1, \ell_j \rangle|$ for all $j \in \Omega$, and in particular:

$$|\langle \ell_1, \ell_{i^*} \rangle| \leq |\langle \ell_1, \ell_j \rangle| \text{ for all } j \in \Lambda$$

Summing over Λ we get:

$$|\Lambda| |\langle \ell_1, \ell_{i^*} \rangle| \leq \sum_{j \in \Lambda} |\langle \ell_1, \ell_j \rangle| = \|\mathbf{L}_\Lambda^\top \ell_1\|_1 \quad (8.3)$$

eq:ForceContr

We shall show that equation (8.3) cannot hold for n_0 large enough. Because $p \geq 4k(\ln(n))^2/n$ and $r \ll 1$, we may use Theorem 5.8, which we shall do repeatedly. By this theorem, we have that $\mathbf{L} = \mathbf{L}^0 + \mathbf{E} = \mathbf{L}^0 + \mathbf{E}^1 + \mathbf{E}^2$ with $\|\mathbf{E}^j\|_\infty \leq r = \mathcal{O}(r^2)$ for $j = 1, 2$. Moreover by construction $\mathbf{L}_\Lambda^0 = \begin{bmatrix} \mathbf{0} \\ \tilde{\mathbf{L}}_\Lambda \end{bmatrix}$ where $\tilde{\mathbf{L}}$ denotes the Laplacian of the subgraph $G_{C_1^c}$ and $\mathbf{0}$ is the zero matrix of size $n_0 \times |\Lambda|$. Similarly, $\ell_1 = \ell_1^0 + \mathbf{e}_1$ and we may write $\ell_1^0 = \begin{bmatrix} \ell_1^1 \\ \mathbf{0} \end{bmatrix}$ where $\ell_1^1 \in \mathbb{R}^{n_0}$ is the first column of \mathbf{L}^1 , the Laplacian of the subgraph G_{C_1} and $\mathbf{0}$ is the zero vector of length $n - n_0$. Thus:

$$\begin{aligned} \|\mathbf{L}_\Lambda^\top \ell_1\|_1 &\leq \|(\mathbf{L}_\Lambda^0)^\top \ell_1^0\|_1 + \|(\mathbf{L}_\Lambda^0)^\top \mathbf{e}_1\|_1 + \|\mathbf{E}^\top \ell_1\|_1 \\ &= \left\| \begin{bmatrix} \mathbf{0}^\top & \tilde{\mathbf{L}}_\Lambda^\top \end{bmatrix} \begin{bmatrix} \ell_1^1 \\ \mathbf{0} \end{bmatrix} \right\|_1 + \|(\mathbf{L}_\Lambda^0)^\top \mathbf{e}_1\|_1 + \|\mathbf{E}^\top \ell_1\|_1 \\ &\leq 0 + \|(\mathbf{L}_\Lambda^0)^\top\|_1 \|\mathbf{e}_1\|_1 + \|\mathbf{E}^\top\|_1 \|\ell_1\|_1 \end{aligned} \quad (8.4)$$

eq:ReturnToTh

Now $\|(\mathbf{L}_\Lambda^0)^\top\|_1 = \|\mathbf{L}_\Lambda^0\|_\infty$, as these are dual norms. As $\|\cdot\|_\infty$ is equal to the maximum absolute row sum, it is clear that $\|\mathbf{L}_\Lambda^0\|_\infty \leq \|\mathbf{L}^0\|_\infty$. Now:

$$\|\mathbf{L}^0\|_\infty = \max_i \sum_j \left| \delta_{ij} - \frac{\mathbf{A}_{ij}^0}{d_i^0} \right| = \max_i \left(1 + \frac{1}{d_i^0} \sum_{j \neq i} \mathbf{A}_{ij}^0 \right) = \max_i \left(1 + \frac{1}{d_i^0} (d_i^0) \right) = 2$$

Moreover, $\|\mathbf{e}_1\|_1 \leq \max_i \|\mathbf{e}_i\|_1 = \|\mathbf{E}\|_1$, hence $\|\mathbf{e}_1\|_1 \leq 2\beta^2 r + \mathcal{O}(r^2)$, while $\|\mathbf{E}^\top\|_1 = \|\mathbf{E}\|_\infty \leq \|\mathbf{E}^1\|_\infty + \|\mathbf{E}^2\|_\infty = 2r + \mathcal{O}(r^2)$ both by Theorem 5.8. Finally, $\ell_1 = \ell_1^0 + \mathbf{e}_1$ as above, so $\|\ell_1\|_1 \leq \|\ell_1^0\|_1 + \beta^2 r + \mathcal{O}(r^2)$. We compute $\|\ell_1^0\|_1$:

$$\|\ell_1^0\|_1 = \sum_{i=1}^n \left| \delta_{i1} - \frac{\mathbf{A}_{i1}^0}{d_i^0} \right| = 1 + \sum_{i=2}^n \frac{\mathbf{A}_{i1}^0}{d_i^0} \leq 1 + \frac{1}{d_{\min}} (d_1^0) \leq 1 + \frac{(1+\alpha)pn_0}{(1-\alpha)pn_0} = 1 + \beta^2$$

Returning to equation (8.4):

$$\|\mathbf{L}_\Lambda^\top \ell_1\|_1 \leq (2)(2\beta^2 r) + (2r)(1 + \beta^2) + \mathcal{O}(r^2) = (2 + 6\beta^2)r + \mathcal{O}(r^2) = \frac{(2 + 6\beta^2)r_0}{\sqrt{n_0}} + \mathcal{O}(r^2)$$

On the other hand, by Theorem 6.3 we have that:

$$|\langle \ell_1, \ell_{i^*} \rangle| \geq \frac{1}{n_0} \left(\frac{\beta^2}{1 + \alpha} \right) - o\left(\frac{1}{n_0}\right)$$

almost surely, and so we can bound the left hand side of (8.3) as:

$$|\Lambda| |\langle \ell_1, \ell_{i^*} \rangle| \geq \frac{n_0}{9} \left(\frac{1}{n_0} \left(\frac{\beta^2}{1+\alpha} \right) - o\left(\frac{1}{n_0}\right) \right) = \frac{\beta^2}{9(1+\alpha)} - o(1)$$

Thus, if inequality (8.3) were true, it would imply that:

$$\frac{\beta^2}{9(1+\alpha)} - o(1) \leq \frac{(2+6\beta^2)r_0}{\sqrt{n_0}} + \mathcal{O}(r^2)$$

which cannot be true for n_0 large enough, as α and r are $o(1)$ with respect to n_0 and β^2 is $O(1)$, thus $\beta^2/(9(1+\alpha)) - o(1) \rightarrow 1/9$ while $(2+6\beta^2)r_0/\sqrt{n_0} \rightarrow 0$. Hence we may always take n_0 large enough so that, with probability at least $1 - 1/n_0$, inequality (8.3) cannot hold. Thus no such i^* which is in $C_1 \setminus \{1\}$ but not in Ω can be found, and so the Theorem 8.1 is proved. \square

With the result of Theorem 8.1 in mind, we are able to derive a new algorithm to find the communities from $G \in \mathcal{G}(n, k, p, q)$ for $p \gg q > 0$:

Algorithm 4 Single Cluster Pursuit (SCP)

Input: The adjacency matrix A of a graph G , and estimate of the size of clusters n_0

(1) **Initialization** Compute $L = I - D^{-1}A$.

(2) **Trimming**

Let $\Omega = \mathcal{L}_{\lceil 10(n_0-1)/9 \rceil}(L_{-1}^\top \ell_1)$

(3) **Perturbed Sparse Recovery**

(a) $\mathbf{y} = \sum_{i \in \Omega} \ell_i + \ell_1$

(b) Solve $\mathbf{z}^\# = \arg \min \{\|L_\Omega \mathbf{z} - \mathbf{y}\|_2 \text{ s.t. } \mathbf{z} \in \mathbb{R}^{|\Omega|} \text{ and } \|\mathbf{z}\|_0 \leq n_0/9\}$ using the SP algorithm

(c) $\Lambda^\# = \text{supp}(\mathbf{z}^\#)$.

Output: $C_1^\# = \{1\} \cup (\Omega \setminus \Lambda^\#)$.

Next we show that Algorithm 4 succeeds almost surely.

Theorem 8.2. *Let G be a graph drawn at random from the $\mathcal{G}(n, k, p, q)$ model. Suppose that $k = O(1)$ with respect to n and either:*

1. $q = \frac{Q}{n}$ and $p = \frac{P \ln(n)}{\sqrt{n} \ln(\ln(n))}$ with Q and P being constants and $P > 979$, or
2. $q = \frac{Q \ln(n)}{n}$ with $Q \rightarrow \infty$ as $n \rightarrow \infty$ and $p = \frac{P \ln(n)}{\sqrt{n}}$ with $P \rightarrow \infty$ as $n \rightarrow \infty$ and $\frac{Q}{P} < \frac{1}{979}$ for large enough n .

Then Algorithm 4 will recover C_1 (that is, $C_1^\# = C_1$) almost surely.

PROOF. For notational convenience let $s := (n_0 - 1)/9$. In both cases, $p \geq 4k(\ln(n))^2/n$ and so by part 2 of Theorem 3.6, $d_{\min}^0 \geq (1 - \alpha)n_0 p$ almost surely for $\alpha = o(1)$ with respect to n . In case 1, by (i) of Theorem 3.4 in [25], $d_{\max}^\epsilon \leq \ln(n)/(\ln(\ln(n)) - \ln(\ln(\ln(n))))$ almost surely. Ignoring the triple-logarithmic term:

$$\max_i \frac{d_i^\epsilon}{d_i^0} \leq \frac{d_{\max}^\epsilon}{d_{\min}^0} \leq \frac{1}{P(1-\alpha)\sqrt{n_0}} =: \frac{r_0}{\sqrt{n_0}}$$

with $r_0 < 1/979$ for large enough n as $\alpha = o(1)$. In case 2, by (ii) of Theorem 3.4 in [25], $d_{\max}^\epsilon \leq Q \ln(n)$ almost surely, and so:

$$\max_i \frac{d_i^\epsilon}{d_i^0} \leq \frac{d_{\max}^\epsilon}{d_{\min}^0} \leq \frac{Q/P}{(1-\alpha)\sqrt{n_0}} =: \frac{r_0}{\sqrt{n_0}}$$

and again $r_0 < 1/979$ for large enough n . We may thus apply Theorem 8.1 to get that $C_1 \setminus \{1\} \subset \Omega$ almost surely. As before we define $\Lambda := \Omega \cap C_1^c$; it is easy to see $|\Lambda| \leq n_0/9$. The challenge in step (3) of Algorithm 4 is to separate $C_1 \setminus \{1\}$ from Λ . We do this by solving for $\mathbf{1}_\Lambda = \mathbf{1}_\Omega - \mathbf{1}_{C_1 \setminus \{1\}}$ instead of $\mathbf{1}_{C_1 \setminus \{1\}}$, as:

$$\mathbf{L}_\Omega^0 \mathbf{1}_\Lambda = \mathbf{L}_\Omega^0 (\mathbf{1}_\Omega - \mathbf{1}_{C_1 \setminus \{1\}}) = \mathbf{L}_\Omega^0 \mathbf{1}_\Omega - \mathbf{L}_\Omega^0 \mathbf{1}_{C_1 \setminus \{1\}} = \sum_{i \in \Omega} \ell_i^0 - (-\ell_1^0) =: \mathbf{y}^0 \quad (8.5) \quad \text{eq:switch}$$

and noting that $\|\mathbf{1}_\Lambda\|_0 \leq n_0/9$, we see that $\mathbf{1}_\Lambda$ is the unique solution to

$$\arg \min \{ \|\mathbf{L}_\Omega^0 \mathbf{z} - \mathbf{y}^0\|_2 : \mathbf{z} \in \mathbb{R}^{\lceil 10n_0/9 \rceil}, \|\mathbf{z}\|_0 \leq n_0/9 \} \quad (8.6) \quad \text{eq:unperturbed}$$

Defining \mathbf{y} to be a perturbation of \mathbf{y}^0 and writing, as in §5.3, $\mathbf{L}_\Omega = \mathbf{L}_\Omega^0 + \mathbf{E}_\Omega$:

$$\mathbf{y} := \sum_{i \in \Omega} \ell_i + \ell_1 = \sum_{i \in \Omega} \ell_i^0 + \sum_{i \in \Omega} \mathbf{e}_i + \ell_1^0 + \mathbf{e}_1 = \mathbf{y}^0 + \left(\mathbf{e}_1 + \sum_{i \in \Omega} \mathbf{e}_i \right)$$

Defining $\mathbf{e} = \mathbf{e}_1 + \sum_{i \in \Omega} \mathbf{e}_i$ we recognize the problem:

$$\mathbf{z}^\# = \arg \min \{ \|\mathbf{L}_\Omega \mathbf{z} - \mathbf{y}\|_2 : \mathbf{z} \in \mathbb{R}^{\lceil 10n_0/9 \rceil}, \|\mathbf{z}\|_0 \leq n_0/9 \} \quad (8.7) \quad \text{eq:Perturbed2}$$

as a totally perturbed version of (8.6). Thus, we may apply the results of §4.3 to bound $\|\mathbf{z}^\# - \mathbf{1}_\Lambda\|_2$. As in §4.3, we define $\delta_t := \delta_t(\mathbf{L}^0)$ and $\hat{\delta}_t := \delta_t(\mathbf{L})$. (We refer the reader to §4.3 for the definitions of $\epsilon_{\mathbf{L}^0}^s$, $\epsilon_{\mathbf{y}}$, $\hat{\rho}$ and $\hat{\tau}$.) Let us bound on these quantities. Observe that in both cases, $p \gg (\ln(n_0))^2/n_0$ and by assumption k is $O(1)$ with respect to n , thus as in the proof of Theorem 5.10 we get $\epsilon_{\mathbf{L}^0}^s = 8\beta r/3 = 8\beta r_0/(3\sqrt{n_0})$, for n_0 large enough. Applying the result of Theorem 5.10 with $t = 3s = n_0/3$:

$$\begin{aligned} \hat{\delta}_{3s} &\leq \frac{1}{3} + D_1 r + \frac{D_2}{\sqrt{n_0}} + \frac{D_3 r}{\sqrt{n_0}} + \mathcal{O}(r^2) + o\left(\frac{1}{\sqrt{n_0}}\right) \\ &= \frac{1}{3} + \frac{(D_1 r_0 + D_2)}{\sqrt{n_0}} + \frac{D_3 r_0}{n_0} + \mathcal{O}\left(\frac{1}{n_0}\right) + o\left(\frac{1}{\sqrt{n_0}}\right) = \frac{1}{3} + \frac{(D_1 r_0 + D_2)}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right). \end{aligned}$$

Hence for large enough n_0 , we certainly have $\hat{\delta}_{3s} \leq 0.4859$ as required. In fact, let us take n_0 large enough such that $\hat{\delta}_{3s} \leq 1/3 + 1/18 = 7/18 \approx 0.39$. Choosing n_0 larger if necessary, we shall also assume that $\epsilon_{\mathbf{L}^0}^s < 1/3$. In this case, a straightforward, but tedious calculation will reveal that $\hat{\rho} \leq 0.7$ and similarly $\hat{\tau} < 166$ (see the statement of Theorem 4.6 for the definitions of $\hat{\rho}$ and $\hat{\tau}$). We now turn our attention to $\epsilon_{\mathbf{y}}$.

$$\|\mathbf{y}^0\|_2 = \|\mathbf{L}_{\Omega_1}^0 \mathbf{1}_\Lambda\|_2 \geq \sqrt{1 - \delta_s} \|\mathbf{1}_\Lambda\|_2 = \sqrt{1 - \delta_s} \sqrt{s}$$

while

$$\|\mathbf{e}\|_2 = \|\mathbf{E} \mathbf{1}_\Omega + \mathbf{e}_1\|_2 \leq \|\mathbf{E}\|_2 \|\mathbf{1}_{\Omega \cup \{1\}}\|_2 \leq (2\beta r + \mathcal{O}(r^2)) \sqrt{10s}.$$

So we have

$$\epsilon_{\mathbf{y}} := \frac{\|\mathbf{e}\|_2}{\|\mathbf{y}^0\|_2} = \frac{2\sqrt{10} (2\beta r + \mathcal{O}(r^2))}{\sqrt{1 - \delta_s}} = \frac{2\sqrt{10}\beta r_0}{\sqrt{n_0(1 - \delta_s)}} + o\left(\frac{1}{\sqrt{n_0}}\right)$$

Appealing to Theorem 5.10 with $s = n_0/9$:

$$\delta_s \leq 1/9 + \frac{64p^{-1/2}}{9\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right).$$

Again, we will assume that n_0 is large enough such that $\delta_s \leq 2/9 \approx 0.22$. Under this assumption:

$$\epsilon_{\mathbf{y}} \leq \frac{2\sqrt{10}}{\sqrt{7/9}} \frac{\beta r_0}{\sqrt{n_0}} \leq 8 \frac{\beta r_0}{\sqrt{n_0}}$$

Finally, we use Theorem 5.10 again, this time with $t = s = n_0/9$:

$$\hat{\delta}_s \leq 1/9 + \frac{D_1 r_0 + D_2}{\sqrt{n_0}} + o\left(\frac{1}{\sqrt{n_0}}\right)$$

and take n_0 large enough such that $\hat{\delta}_s \leq 2/9$ (this is possible as D_1 and D_2 are $\mathcal{O}(1)$ with respect to n_0). Now Theorem 4.6 guarantees that

$$\frac{\|\mathbf{x}^m - \mathbf{1}_\Lambda\|_2}{\|\mathbf{1}_\Lambda\|_2} \leq \left(\hat{\tau} \frac{\sqrt{1 + \hat{\delta}_s}}{1 - \epsilon_{L^0}^s} + 1 \right) (\epsilon_{L^0}^s + \epsilon_{\mathbf{y}})$$

Where $m := \ln(\epsilon_{L^0}^s + \epsilon_{\mathbf{y}}) / \ln(\hat{\rho}) = \mathcal{O}(\ln(n_0))$. Substituting in for the various constants:

$$\begin{aligned} \frac{\|\mathbf{x}^m - \mathbf{1}_\Lambda\|_2}{\|\mathbf{1}_\Lambda\|_2} &< \left((166) \frac{\sqrt{1 + 2/9}}{1 - 1/3} \right) \left(\frac{8\beta r_0}{3\sqrt{n_0}} + \frac{8\beta r_0}{\sqrt{n_0}} \right) = \frac{166(\sqrt{11})(32)}{6} \frac{\beta r_0}{\sqrt{n_0}} \\ \Rightarrow \|\mathbf{x}^m - \mathbf{1}_\Lambda\|_2 &< \left(\frac{\sqrt{n_0}}{3} \right) 2937 \frac{\beta r_0}{\sqrt{n_0}} = 979\beta r_0 \end{aligned}$$

As the Lemma 8.3 below, as long as $\|\mathbf{x}^m - \mathbf{1}_\Lambda\|_2 < 1$, we have that $\text{supp}(\mathbf{x}^m) = \text{supp}(\mathbf{1}_\Lambda)$. Because $\beta \rightarrow 1$ and $r_0 < 1/979$, this will hold for large enough n_0 . \square

cp **Lemma 8.3.** *Suppose that $\mathbf{x}^* \in \mathbb{R}^n$ is a binary vector with $\|\mathbf{x}^*\|_0 = t$, and $\mathbf{x} \in \mathbb{R}^n$ is any other vector that also has $\|\mathbf{x}\|_0 = t$. If $\|\mathbf{x}^* - \mathbf{x}\|_2 < 1$ then $\text{supp}(\mathbf{x}) = \text{supp}(\mathbf{x}^*)$.*

PROOF. Suppose otherwise. Then there exists an $i \in \text{supp}(\mathbf{x}^*) \setminus \text{supp}(\mathbf{x})$. Clearly:

$$\|\mathbf{x}^* - \mathbf{x}\|_2 \geq |x_i^* - x_i| = |1 - 0| = 1$$

which contradicts the hypotheses. \square

As mentioned in the introduction, we may iterate Algorithm 4 to find all the clusters of G . We call this algorithm Iterated Single Cluster Pursuit, or ISCP.

9 Computational Complexity and Extensions

In this section we bound the computational complexity and explain some extensions.

9.1 Computational Complexity

In this subsection we show that Algorithm 4 is faster than existing spectral methods. We do this by determining the approximate number of operations required in each step of Algorithm 4. Throughout, we shall assume that \mathbf{A} and \mathbf{L} are sparse matrices.

1. Computing $d_i = \sum_j A_{ij}$ requires d_i operations. We can safely assume that $d_i = d_i^0 + d_i^\epsilon \leq n_0$ so the cost of computing each d_i is at most $\mathcal{O}(n_0)$. This needs to be done a total of n times to compute $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$, giving a total cost of $\mathcal{O}(nn_0)$.

2. The cost of the thresholding step is dominated by the matrix-vector multiply $\mathbf{L}_{-1}^\top \ell_1$. Because ℓ_1 has $d_1 + 1 \leq n_0$ non-zero entries, the cost of this is bounded by $\mathcal{O}(nn_0)$.
3. The computational cost of solving the perturbed sparse recovery problem in step 3 using SP is equal to the number of iteration times the cost of each iteration. As shown in the proof of Theorem 8.2, it suffices to perform $\mathcal{O}(\ln(n_0)) = \mathcal{O}(\ln(n))$ iterations. The cost of each iteration is determined by calculating the cost of each step in the iterative part of SP (see Algorithm 3):
 - (a) Computing $\mathcal{L}_s(\mathbf{L}_\Omega^\top \mathbf{r}^{k-1})$ is dominated by the cost of the matrix-vector multiply $\mathbf{L}_\Omega^\top \mathbf{r}^{k-1}$, hence is $\mathcal{O}(nn_0)$.
 - (b) Solving the least square problem in step (2) using a good numerical solver (we used MATLAB's `lsqr` algorithm) is on the order of the cost of a matrix-vector multiply, i.e. $\mathcal{O}(nn_0)$, as explained in [38].
 - (c) The cost of sorting and thresholding (step (3)) is $\mathcal{O}(n \ln(n))$.
 - (d) Finally the cost of computing the new residual \mathbf{r}^k in step (4) is dominated by the matrix vector multiply $\mathbf{L}_{\Omega_1}^\top \mathbf{r}^k$, hence is $\mathcal{O}(nn_0)$.

We conclude that the cost of a single iteration of the SP Algorithm is $\mathcal{O}(nn_0)$, and so the cost of step 3 of Algorithm 4 is $\mathcal{O}(n \ln(n) n_0)$

Thus, the number of computations required to find a single cluster using Algorithm 4 is $\mathcal{O}(n \ln(n) n_0)$.

To find all the clusters C_1, \dots, C_k one can iterate Algorithm 4 $k - 1$ times. The cost of this is certainly less than $\mathcal{O}(kn \ln(n) n_0) = \mathcal{O}(n^2 \ln(n))$.

9.2 Extensions

1. Our Algorithm 4 can be extended to deal with the communities of non-equal size. Hence, our study can be extended too. A numerical example is included in the next section to demonstrate this extension. For simplicity, we leave the study for the interested reader.
2. Although Algorithm 4 works on binary adjacency matrices, it handles weighted adjacency matrices just as well (see §10.2.2).
3. As mentioned in the introduction, graph clustering can also be applied to more general data sets. Given any (finite) set of points $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ in some metric space (M, d) , there are several ways to associate a graph on n vertices to \mathbf{X} . For example, we could attach an edge between vertices i and j whenever $d(\mathbf{x}_i, \mathbf{x}_j) \leq C$ for some constant C . Alternatively, for each vertex i we could determine the K closest points to \mathbf{x}_i , say $\{\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_K}\}$, and insert the edges $\{i, j_1\}, \dots, \{i, j_K\}$ to obtain the K -nearest neighbours graph G^K . Empirically, we have found the latter approach to perform better, as $d_{\min}(G^K) = K$ giving better control over the quantities r_i crucial to the analysis of Algorithm 4. See §10.2.2 for an example of this approach.
4. Algorithm 4 can easily be extended to the co-clustering problem, as described in [15]. Briefly suppose that we are given two categorically different sets of variables $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_m\}$ as well as a $n \times m$ matrix \mathbf{B} such that the \mathbf{B}_{ij} -th entry represents the correlation between \mathbf{x}_i and \mathbf{y}_j . The goal of co-clustering is to simultaneously partition \mathbf{X} and \mathbf{Y} into subsets $\mathbf{X} = \mathbf{X}_1 \cup \dots \mathbf{X}_k$ and $\mathbf{Y} = \mathbf{Y}_1 \cup \dots \mathbf{Y}_k$ such that the correlations between $\mathbf{x} \in \mathbf{X}_i$ and $\mathbf{y} \in \mathbf{Y}_i$ are strong while the correlations between $\mathbf{x} \in \mathbf{X}_i$ and $\mathbf{y} \in \mathbf{Y}_j$ for $i \neq j$ are weak. As pointed out by Dhillon in [15], one can regard this as a graph clustering

problem for a weighted⁵ bipartite graph with vertex set $V = \{u_1, \dots, u_n, v_1, \dots, v_m\}$ and weights $w(u_i, v_j) = B_{ij}$, $w(u_i, u_j) = w(v_i, v_j) = 0$. The adjacency matrix and Laplacian of G are:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{0} \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} \mathbf{I}_{n \times n} & -\mathbf{D}_\mathbf{X}^{-1} \mathbf{B} \\ -\mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top & \mathbf{I}_{m \times m} \end{bmatrix}$$

where $\mathbf{D}_\mathbf{X}$ represents the row sums of \mathbf{B} while $\mathbf{D}_\mathbf{Y}$ represents the column sums. Assume temporarily that instead of clusters G has k connected components C_1, \dots, C_k . Each G_{C_i} will be a bipartite graph, and C_i splits as $C_i = C_i^\mathbf{X} \cup C_i^\mathbf{Y}$ where $C_i^\mathbf{X}$ corresponds to \mathbf{X}_i and similarly $C_i^\mathbf{Y}$ corresponds to \mathbf{Y}_i . The indicator vector will split similarly: $\mathbf{1}_{C_i} = [\mathbf{1}_{\mathbf{X}_i}^\top \mathbf{1}_{\mathbf{Y}_i}^\top]^\top$ and:

$$\mathbf{0} = \mathbf{L} \mathbf{1}_{C_i} = \begin{bmatrix} \mathbf{1}_{\mathbf{X}_i} - \mathbf{D}_\mathbf{X}^{-1} \mathbf{B} \mathbf{1}_{\mathbf{Y}_i} \\ \mathbf{1}_{\mathbf{Y}_i} - \mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top \mathbf{1}_{\mathbf{X}_i} \end{bmatrix}$$

We may convert these two ‘first-order’ equations into a single ‘second-order’ equation as $\mathbf{1}_{\mathbf{Y}_i} = \mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top \mathbf{1}_{\mathbf{X}_i}$ and so:

$$\mathbf{0} = \mathbf{1}_{\mathbf{X}_i} - \mathbf{D}_\mathbf{X}^{-1} \mathbf{B} \mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top \mathbf{1}_{\mathbf{X}_i} = \left(\mathbf{I} - \mathbf{D}_\mathbf{X}^{-1} \mathbf{B} \mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top \right) \mathbf{1}_{\mathbf{X}_i}$$

Defining the ‘bi-partite Laplacian’ as $\mathbf{L}^{BP} = \mathbf{I} - \mathbf{D}_\mathbf{X}^{-1} \mathbf{B} \mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top$ we see that if $\mathbf{x}_1 \in \mathbf{X}_1$ then the indicator vector $\mathbf{1}_{\mathbf{X}_1}$ is the unique solution to:

$$\arg \min \{ \|\mathbf{z}\|_0 : \mathbf{z} \in \mathbb{R}^n, \mathbf{L}^{BP} \mathbf{z} = \mathbf{0} \text{ and } z_1 = 1 \}$$

once we have found $\mathbf{1}_{\mathbf{X}_1}$ we can recover $\mathbf{1}_{\mathbf{Y}_1}$ as $\mathbf{1}_{\mathbf{Y}_1} = \mathbf{D}_\mathbf{Y}^{-1} \mathbf{B}^\top \mathbf{1}_{\mathbf{X}_1}$. Returning to the clustering problem, we see that we may approximate $\mathbf{1}_{\mathbf{X}_1}$ (and hence $\mathbf{1}_{\mathbf{Y}_1}$) by solving the problem

$$\arg \min \{ \|\mathbf{L}_{-1}^{BP} \mathbf{z} - (-\ell_{BP,1})\|_2 : \mathbf{z} \in \mathbb{R}^{n-1}, \|\mathbf{z}\|_0 \leq n_0^\mathbf{X} - 1 \}$$

using Algorithm 4, where $n_0^\mathbf{X}$ is the size of the clusters \mathbf{X}_i . We leave the technical analysis of this approach to the interested reader.

5. In future work, we intend to show how Algorithm 4 can be adapted into an ‘online’ algorithm which allows one to rapidly update the community assignments as more vertices are added to the graph.

10 Numerical Examples

10.1 Synthetic Data

We tested Single Cluster Pursuit (SCP, Algorithm 4) on graphs drawn from the SBM. We implemented this algorithm in MATLAB 2016b and run on a mid-2010 iMac computer with 8 GB of RAM and a 3.06 GHz Intel Core i3 processor. For comparison we used the Trembley, Puy, Gribonval and Vandergheynst implementation of Spectral Clustering (SC, Algorithm 1) included in their Compressive Spectral Clustering toolbox available at <http://cscbox.gforge.inria.fr/> ([45]).

⁵We assume that the correlations B_{ij} are non-negative

10.1.1 Example 1

Typical output of SCP applied to randomly drawn graphs from the SMB $\mathcal{G}(n, k, p, q)$ with $p = P \ln(n)/\sqrt{n}$ and $q = Q \ln(n)/n$ as in case 2 of Theorem 8.2 is shown in the third frame of figure 1. Both SCP and SC succeed in finding cluster 1 without error (indeed SC finds all clusters without error) but SCP is faster, taking 0.0262 seconds compared to the 0.1018 seconds required by SC. We remark that even for Q, P such that $Q/P \gg 1/979$ SCP is successful, suggesting that the bounds in Theorem 8.2 are too conservative.

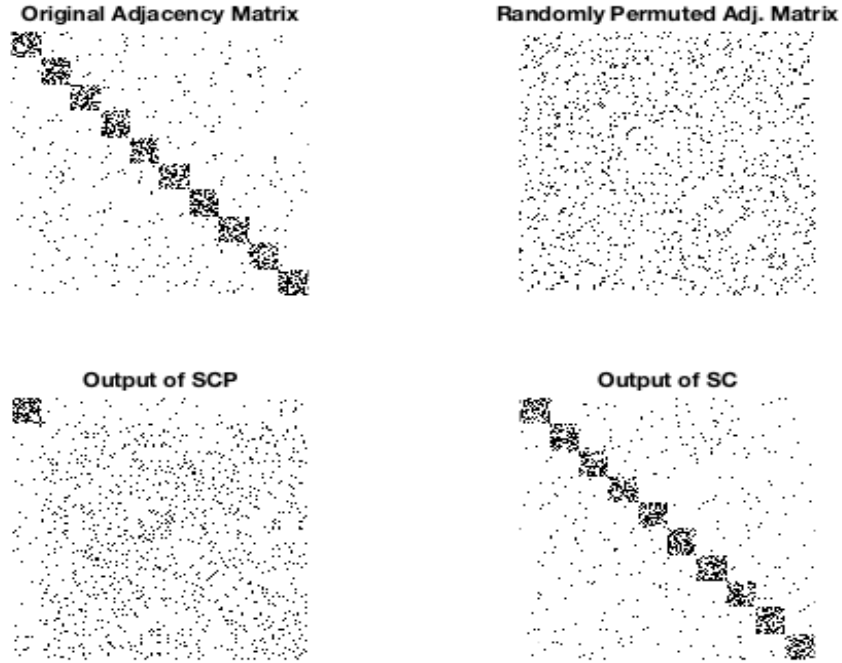


Figure 1: The adjacency matrix of a graph drawn from $\mathcal{G}(1000, 5, \frac{2 \ln(n)}{\sqrt{n}}, \frac{2 \ln(n)}{n})$ is shown in the top left. The same matrix, with its rows and columns randomly permuted (using the same permutation for both) is shown in the top right. The output of SCP applied to the permuted matrix is shown in the bottom left, while the output of SC on the same matrix is shown on the bottom right.

10.1.2 Example 2

We then tested Iterated SCP (ISCP) for graphs randomly drawn from the SBM $\mathcal{G}(n, k, p, q)$ with $p = P \ln(n)/\sqrt{n}$ and $q = Q \ln(n)/n$. Typical output is shown in Figure 2. Again, both algorithms are successful (that is, both find all 10 communities without error).

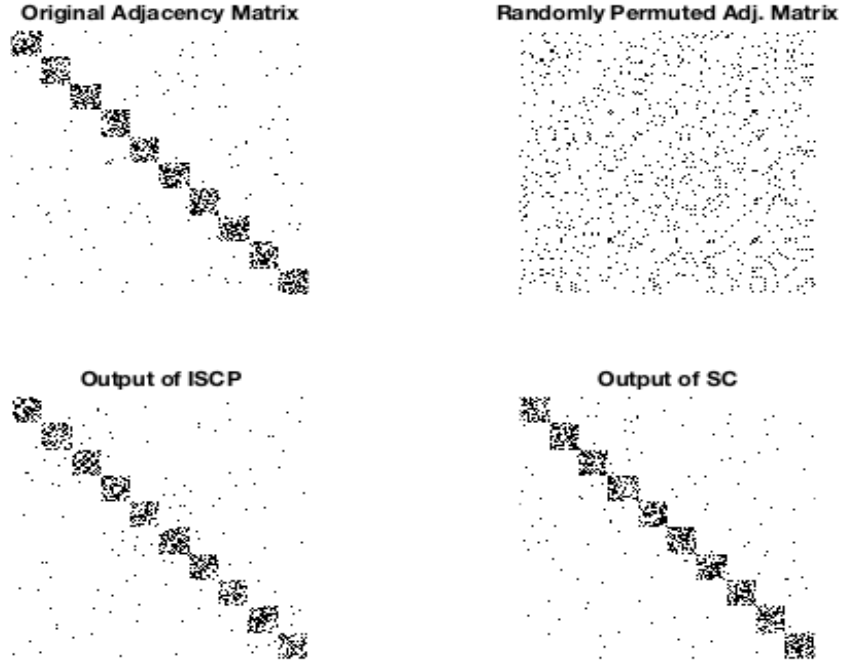


Figure 2: The adjacency matrix of a graph drawn at random from $\mathcal{G}(5000, 10, \frac{4\ln(n)}{\sqrt{n}}, \frac{4\ln(n)}{n})$ is shown in the top left. The same matrix, with its rows and columns randomly permuted (using the same permutation for both) is shown in the top right. The output of SCP applied to the permuted matrix is shown in the bottom left, while the output of SC on the same matrix is shown on the bottom right.

10.1.3 Example 3

Next, we tested the resilience of SCP to noise by running it on G drawn at random from $\mathcal{G}(2400, 6, 0.5, q)$ with q increasing from 0 to 150/2000. For each q , we ran SCP on 10 different G drawn independently and at random from $\mathcal{G}(2400, 6, 0.5, q)$ and computed the average fraction of indices that were misclassified. Figure 3 shows the result of this experiment, with $\mathbb{E}[d_i^c] = q(n - n_0)$ on the x -axis. As observed in Experiment 1, SCP performs much better than theoretically guaranteed by Theorem 8.2. Indeed, Figure 3 demonstrates that SCP detects community 1 without error up until $\mathbb{E}[d_i^c] = 40$, at which $r \approx 40/200 = 1/5$.

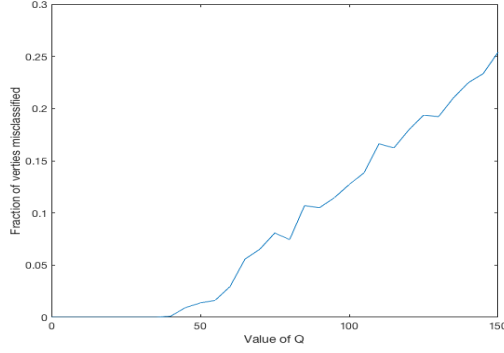


Figure 3: Fraction of indices misclassified by SCP, for $G \in \mathcal{G}(2400, 6, 0.5, Q/2000)$. The horizontal axis represents Q , that is, the level of ‘noise’ in the graph. The vertical axis represents the fraction of misclassified vertices $= |C_1^\# \setminus C_1|/|C_1^\#|$. As can be seen, SCP performs without error until $Q \approx 40$, well beyond the theoretical guarantees given by Theorem 8.2

g:ResTest

10.1.4 Example 5

In our final example, we use ISCP to solve the co-clustering problem, as discussed in §9.2. Typical output is shown in image 4

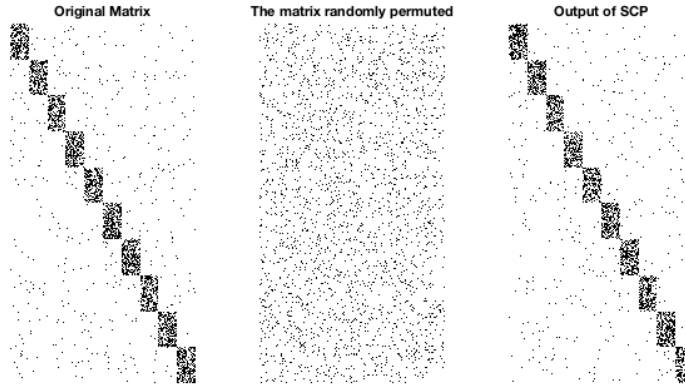


Figure 4: A 2000×1000 binary matrix with 10 equally sized block, randomly permuted and then co-clustered using ISCP, applied as discussed in §9.2.

leCluster

10.1.5 Example 4

Finally we compared the run times of SCP, SC and ISCP on $\mathcal{G}(n, k, p, q)$ for increasing n . We studied three regimes: constant k , constant n_0 (cluster size) and $k, n_0 = \mathcal{O}(\sqrt{n})$. In all three cases we fixed $p = 2 \log(n)/\sqrt{n}$ and $q = 2 \log(n)/n$. The results of these experiments are presented in figures 5c - 5b (the run times are in seconds, and are the average of ten independent trials for the same n). As is clear, SCP significantly outperforms SC in finding a single cluster. Moreover, when k is large enough compared to the number of vertices (n), ISCP finds all clusters faster than SC.

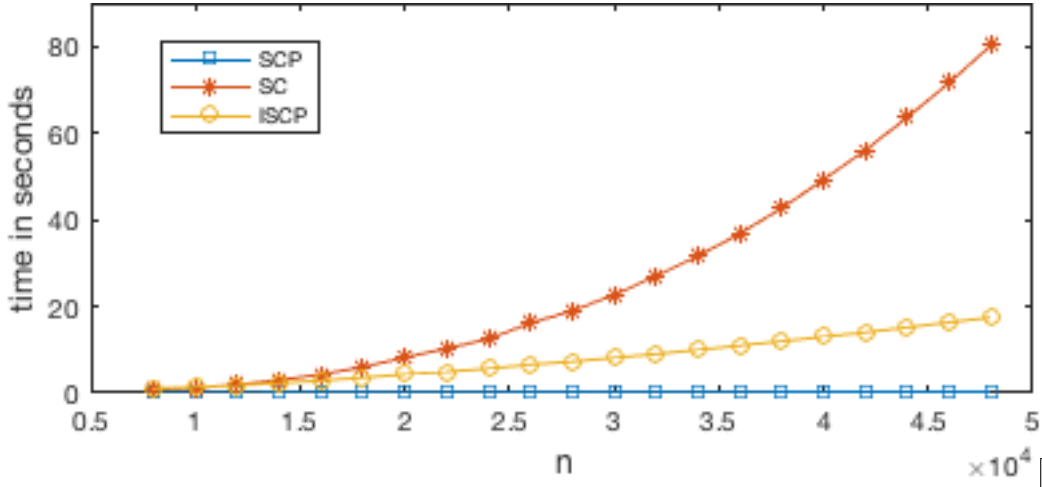


fig:CompareTimevaryK

(a) Varying k from 20 to 120 and n accordingly, while fixing n_0 . $G \in \mathcal{G}\left(400k, k, \frac{2\log(n)}{\sqrt{n}}, \frac{2\log(n)}{n}\right)$

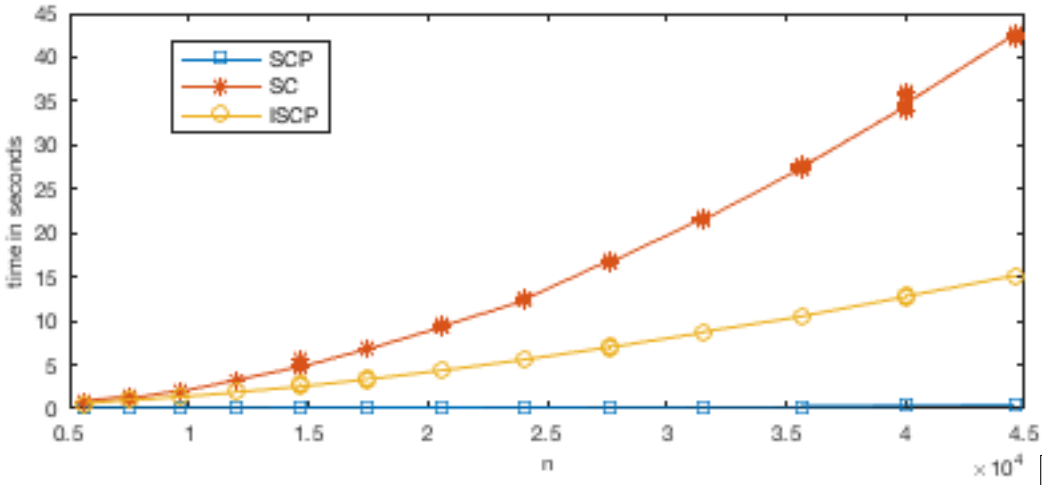


fig:CompareTimevaryKAnd

(b) Varying k , n_0 and n . $G \in \mathcal{G}\left(kn_0, k, \frac{2\log(n)}{\sqrt{n}}, \frac{2\log(n)}{n}\right)$

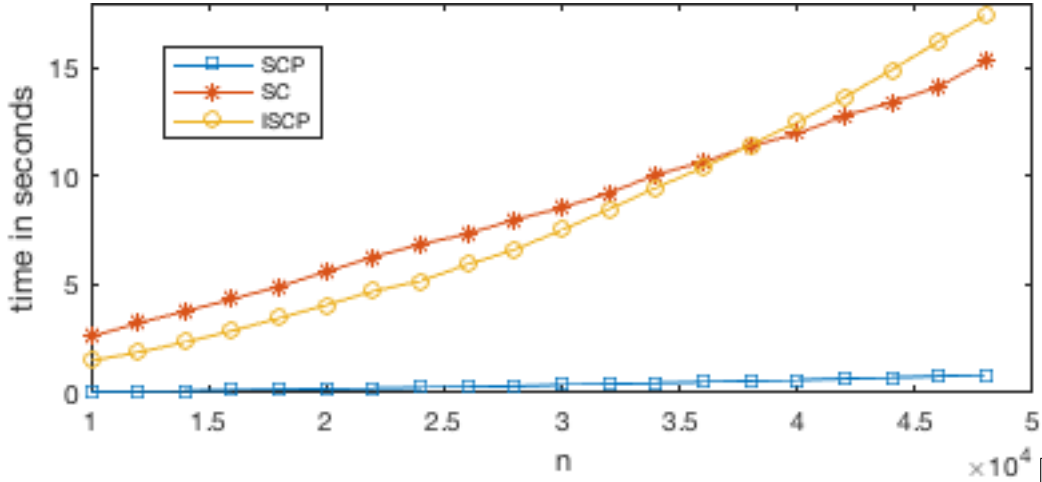


fig:CompareTime20Cluste

(c) Varying n and n_0 while fixing k . $G \in \mathcal{G}\left(n, 40, \frac{2\log(n)}{\sqrt{n}}, \frac{2\log(n)}{n}\right)$

Figure 5: Comparing SCP, ISCP and SC for graphs drawn from the SBM of increasing size. As can be seen, SCP is significantly faster than SC

10.2 Real Data Sets

We now present two examples of SCP and ISCP applied to real-world data sets.

10.2.1 Political Blogs Data Set

The `polblogs` data set is a collection of 1494 political weblogs, or blogs, collected by Adamic and Glance ([4]) prior to the 2004 American presidential election. The data is presented as an unweighted, undirected graph, with vertices corresponding to the blogs, and edges between blogs i and j if there is a hyperlink from i to j , or *vice versa*. This data set is well-studied (see, for example [32], [39] and [42], amongst others), and is a good test case for at least two reasons:

1. In addition to a natural division into two roughly equally sized clusters (liberal vs. conservative), the data set exhibits additional clustering at smaller scales. That is, within the set of all liberal blogs, one can identify subclusters of blogs, and similarly for the conservative blogs. This is explored further in [42].
2. The ground truth for the division into liberal *vs.* conservative is known, as Adamic and Glance ([4]) manually labelled the data set. This provides an opportunity to verify one's results that is rare for real-world data sets.

Our methodology was as follows. We follow Olhede and Wolfe ([42]) in using only the 1224 blogs with links to at least one other blog in the data set. As SCP can be thrown off by low degree vertices (these will have high values of r_i) we experimented with different thresholds. That is, we discarded vertices of degree lower than d_{thresh} for varying values of d_{thresh} . We then used SCP to detect a cluster containing an arbitrary liberal blog, of size approximately equal to the number of liberal blogs. We call this cluster Cluster 1 (C_1). We call the remaining vertices Cluster 2 C_2 . The results are tabulated in ???. We record the number of vertices left after thresholding, the percentage of the first cluster consisting of liberal blogs, the percentage of the second cluster consisting of conservative blogs and the time taken. The values shown are the averages of ten independent runs.

d_{thresh}	# of vertices	% of C_1 liberal	% of C_2 conservative	time (in seconds)
0	1224	63.71	66.93	0.0380
2	1087	71.57	75.68	0.0928
4	903	81.45	83.86	0.0467
6	813	89.12	91.06	0.0479
8	735	91.95	93.69	0.0428
10	693	93.14	94.57	0.0411
12	644	93.95	95.32	0.0256
14	596	93.05	94.66	0.0377
16	547	94.24	95.39	0.0379
18	505	93.97	94.87	0.0225

Table 1: Results of SCP applied to the `polblogs` data set for varying degree thresholds. When $d_{\text{thresh}} = 10$ the fraction of cluster 1 classified as liberal, and the fraction of cluster 2 identified as conservative, closely resemble the compositions reported by Newman in [39]

As is clear, increasing d_{thresh} above 10 makes little difference. Our results also agree with the clusters found by Newman in [39], where he finds one cluster which is 93% liberal, and a second

cluster which is 97% conservative. Of course, to achieve this accuracy we have had to discard over 40% of our data points. An interesting future line of research would be to improve the handling of low degree vertices by SCP.

To detect clustering at a finer scale, we ran SCP to find a cluster of size 80 containing a randomly selected liberal blog (smithersmpls.com). The output of this is shown in figure 7. As can be seen by the increased density of the top-left hand corner, our algorithm finds a subset of blogs containing smithersmpls.com that are more densely connected to each other than they are to the rest of the data set. We remark that this experiment took approximately 0.06 seconds, so with even modest computational resources one could investigate clustering at a large range of scales in such a data set.

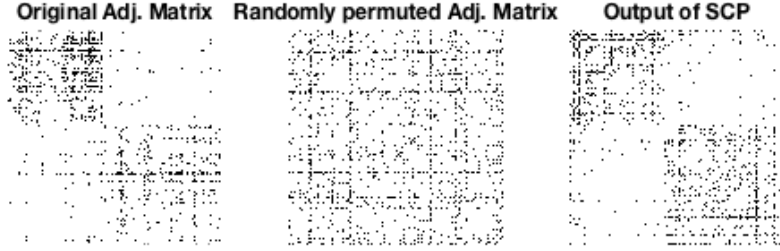


Figure 6: Typical Output of SCP applied to the `polblogs` data set with $d_{\text{thresh}} = 10$.

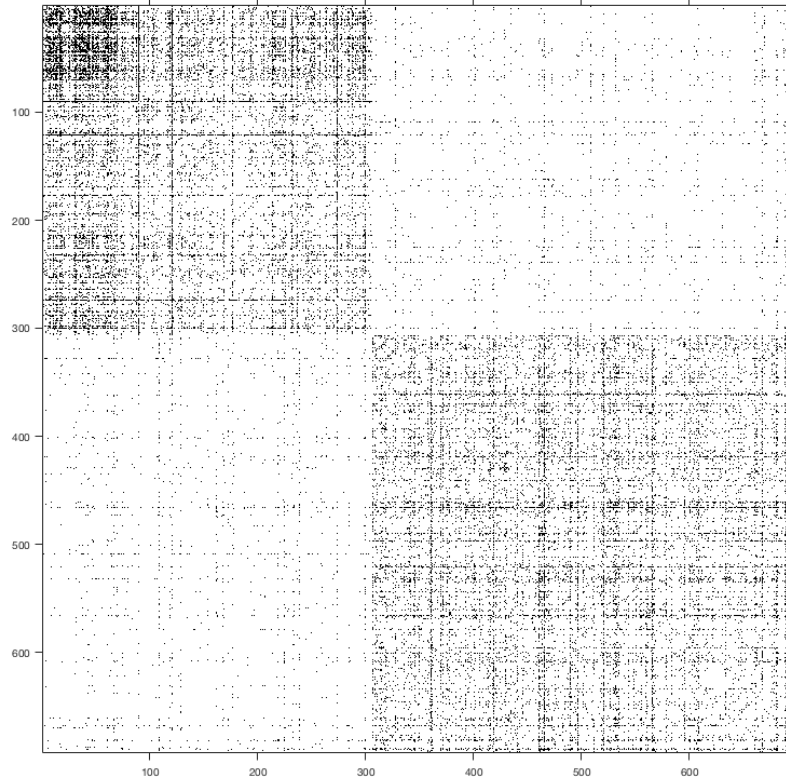


Figure 7: Detecting fine scale structure in the `polblogs` data set. The top left hand corner represents a sub-cluster of 80 liberal blogs containing a given randomly selected liberal blog.

10.2.2 Gene Expression Data Set

Our second data set consists of Gene Expression values collected via Microarray for a sample of *Neurospora crassa* at 13 different time points, originally studied in [16]. The data consists of 2436 time series, one for each Gene of interest, consisting of 13 scaled expression readings. Each scaled expression reading is a floating point number between 4 and 13. We treat this data as a set of 2436 data points $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_{2436}\}$ in \mathbb{R}^{13} . We then constructed an *Affinity Matrix* \mathbf{A}^1 as suggested in [41] defined as $\mathbf{A}_{ij}^1 = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / \sigma^2)$. Here σ was chosen to be $\sqrt{10}$ although we note that experimenting with other values of σ did not qualitatively change the results.

To promote sparsity, we added the additional step of constructing a K -nearest-neighbours adjacency matrix from \mathbf{A}^1 by retaining the (weighted) edge $\{i, j\}$ if and only if \mathbf{x}_j is among the K nearest neighbours of \mathbf{x}_i or *vice versa*. Call the resulting adjacency matrix $\mathbf{A}^{2,K}$. We remark that, much like the parameter σ used in constructing \mathbf{A}^1 , varying K did not qualitatively affect the results obtained. Thus, we shall fix $K = 100$ and refer to $\mathbf{A}^{2,100}$ simply as \mathbf{A}^2 . Figure 8 shows the results of this preprocessing in reverse-grayscale (that is, larger values are darker).

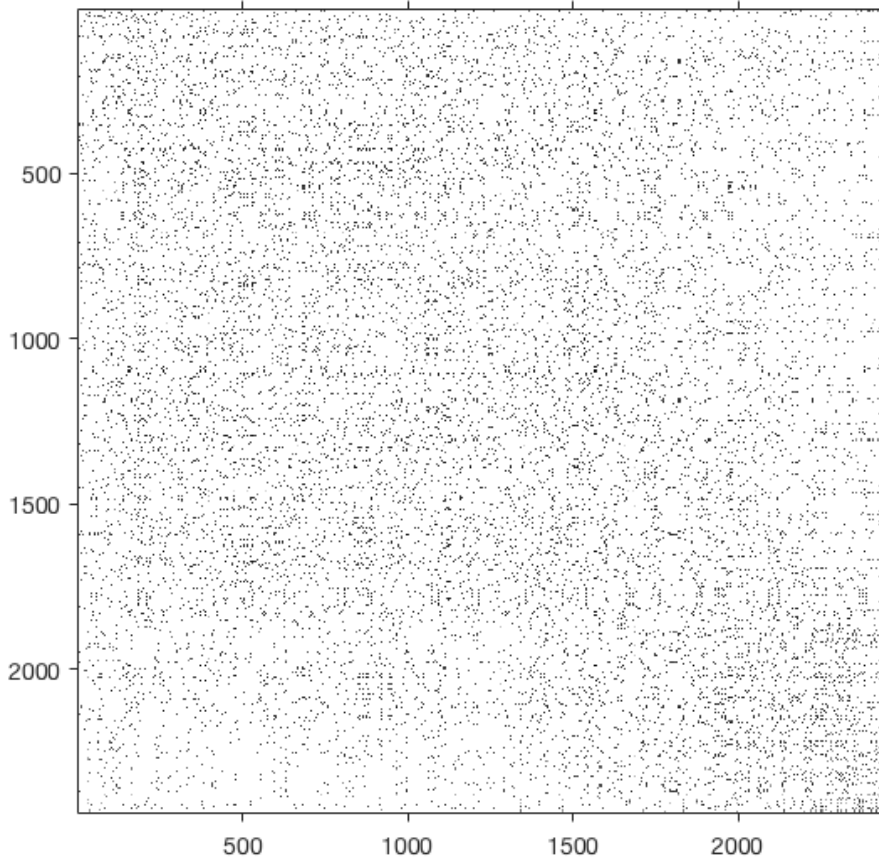


Figure 8: The adjacency matrix \mathbf{A}^2 , resulting from preprocessing the Gene Expression Data set originally studied in [16].

We used ISCP to look for clusters on two different scales, informed by the underlying biology of

the data set⁶. First, we looked for clusters of size 90 – 100, as this is around the number of genes targeted by a single regulator in *N. crassa*. The results of this experiment are shown in figure 9.

In a well studied organism such as *N. crassa*, the functions of many genes are known. In fact of the 2436 genes in this data set, more than half of them (1442) have been assigned a MIPS label, which is an hierarchical system for labelling genes by function. Moreover, in [16], the genes in this data set are grouped into twelve categories according to their functions, for example ‘signaling’ or ‘transcriptional control and regulation’. Biologically, an interesting question is which functional categories cluster together. Thus, for our second numerical experiment we used ISCP to detect clusters of size 450 – 550 (results pictured in Figure 10) and recorded the number of genes of each category⁷ occurring in each cluster. We can assess the biological significance of this clustering by performing a chi-squared test. The null hypothesis is that there is no relation between the clusters and the functional categories, in which the expected number of genes in Category j contained in Cluster i , in the notation of Table ??, is $E_{ij} = \mathbf{T}_j^C \times (\mathbf{T}_i^R/1442)$. For example, $E_{12} = 164 \times (321/1442) = 36.51$. Denoting the observed counts by O_{ij} (i.e. $O_{12} = 32$, $O_{21} = 19$ and so on) we compute the chi-squared test statistic as $\sum_i \sum_j (E_{ij} - O_{ij})^2 / E_{ij} = 73.87$. From a table of values for the χ^2 distribution⁸, we get that assuming the null hypothesis there is a 0.1% chance that this statistic is greater than 73.40. Thus, we may safely reject the null hypothesis, and assume that the Clustering found by ISCP is related to the functions of the genes in the data set.

Cluster \ Category	1	2	3	4	5	6	7	8	9	10	11	\mathbf{T}^R	Cluster size
1	13	32	15	22	42	46	36	19	8	20	68	321	527
2	19	36	18	28	24	37	11	10	12	25	70	290	513
3	13	23	21	33	34	57	10	13	11	32	71	318	452
4	15	32	14	17	22	27	17	2	5	22	47	220	554
5	20	41	19	31	35	26	18	15	9	13	66	293	390
\mathbf{T}^C	80	164	87	131	157	193	92	59	45	112	322	1442	2436

Table 2: The number of genes in each category present in each cluster. The row totals represent the number of genes with a MIPS classifier present in each cluster (the total number of genes with a MIPS classifier is 1442). Anomalously high or low counts are shown in bold.

⁶and the authors gratefully acknowledge many informative discussions with Professor Jonathan Arnold and his lab

⁷For a description of the categories the reader is referred to [16]. The counts are displayed in Table ???. The categories are numbered by the order in which they appear in figure 15 in [16], omitting the category ‘clock’ as there are no MIPS codes associated with this category.

⁸Note that there are $(\#rows - 1)(\#columns - 1) = 40$ degrees of freedom here

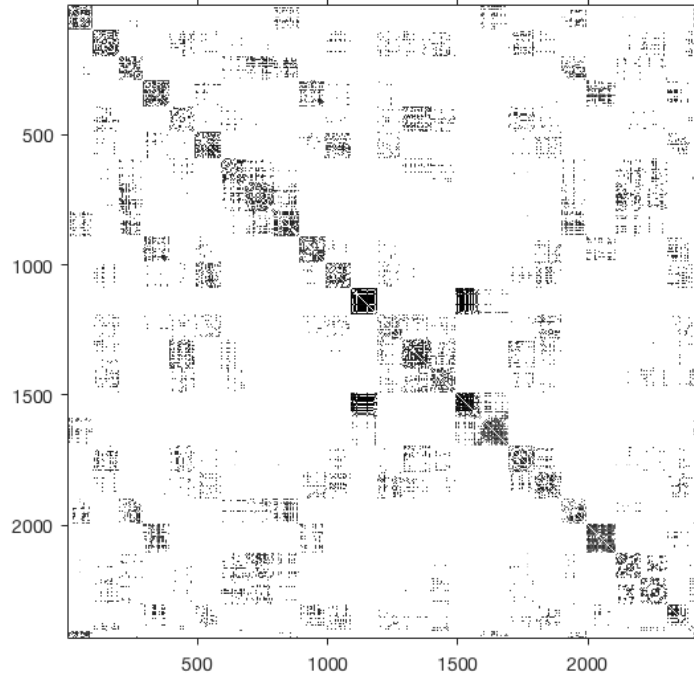


Figure 9: Output of ISCP applied to \mathbf{A}^2 , set to find clusters of size 90 – 100.

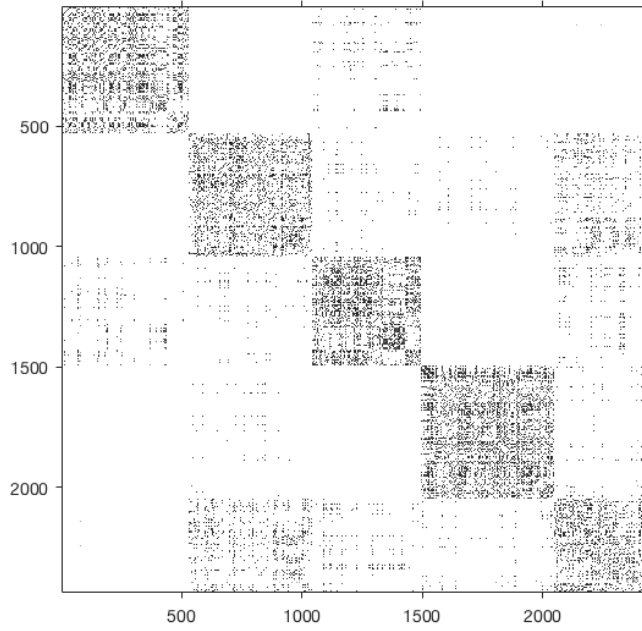


Figure 10: Output of ICSP applied to \mathbf{A}^2 , set to find clusters of size 450 – 550.

References

- Abbe2017 [1] E. Abbe, Community detection and stochastic block models: recent developments, *arXiv preprint arXiv:1703.10146* (2017).
- Abbe2016 [2] E. Abbe, A.S. Bandeira, and G. Hall, Exact recovery in the stochastic block model, *IEEE transactions on Information Theory* Vol. 62 no. 1, pp. 471-487 (2016).
- Abbe2015 [3] E. Abbe and C. Sandon, Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pp. 670-688 (2015).
- Adamic2005 [4] L. A. Adamic and N. Glance, The political blogosphere and the 2004 us election: divided they blog. *Proceedings of the 3rd international workshop on Link discovery*, pp. 36-43 (2005).
- Aldous2000a [5] J. M. Aldous and R. J. Wilson, *Graphs and Applications: An Introductory Approach*, Springer Science & Business Media (2000).
- BT09 [6] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Image Sciences*, Vol. 2, pp. 183–202 (2009).
- BD09 [7] T. Blumensath and M. E. Davies, Iterative hard thresholding for compressed sensing, *Appl. Comput. Harmon. Anal.*, Vol. 27, pp.265–274 (2009).
- CWB08 [8] E. J. Candès, M. B. Wakin, and S. Boyd, Enhancing sparsity by re-weighted ℓ_1 minimization, *Journal of Fourier Analysis and Applications*, Vol. 14, pp. 877-905 (2008).
- Candes2006 [9] E. J. Candès, J. K. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Communications on Pure and Applied Mathematics*, Vol. 59 issue 8, pp. 1207-1223 (2006).
- CT05 [10] E. J. Candès and T. Tao, Decoding by linear programming, *IEEE Transactions on Information Theory*, Vol 51, pp. 4203–4215 (2005).
- Chung2003 [11] F. R. K. Chung, L. Lu, and V. Vu. The spectra of random graphs with given expected degrees, *Proceedings of the National Academy of Sciences*, Vol. 100 issue 11, pp. 6313-6318 (2003).
- Chung1999 [12] F. R. K. Chung, *Spectral graph theory*, No. 92, American Mathematical Soc. (1997).
- Cline2007 [13] M. S. Cline, M. Smoot, E. Cerami, A. Kuchinsky, N. Landys, C. Workman, R. Christmas, I. Avila-Campilo, M. Creech, B. Gross *et al.*, Integration of biological networks and gene expression data using cytoscape, *Nature protocols*, Vol. 2 issue 10, pp. 2366-2382 (2007).
- Dai2009 [14] W. Dai and O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, *IEEE Transactions on Information Theory*, Vol. 55 issue 5, pp. 2230-2249 (2009).
- Dhillon2001 [15] I. S. Dhillon, Co-clustering documents and words using spectral graph partitioning, *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269-274 (2001).
- Dong2008 [16] W. Dong, X. Tang, Y. Yu, R. Nilsen, R. Kim, J. Griffith, J. Arnold and H-B. Schüttler, Systems biology of the clock in *Neurospora crassa*, *PloS one*, 3.8, e3105 (2008).

- Donoho2003 [17] D. L. Donoho and M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization, *Proceedings of the National Academy of Sciences*, Vol. 100 issue 5, pp. 2197-2202 (2003).
- Erdos59 [18] P. Erdős and A. Rényi, On Random Graphs 1, *Publicationes Mathematicae Debrecen*, Vol. 6, pp. 290-297 (1959).
- Fiedler1975 [19] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czechoslovak Mathematical Journal* Vol 25.4, pp. 619-633 (1975).
- Fortunato2010 [20] S. Fortunato, Community detection in graphs, *Physics reports*, Vol. 486 issues 3-5, pp. 75-174 (2010).
- Foucart2013 [21] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Birkäus Verlag (2013).
- F11 [22] S. Foucart, Hard thresholding pursuit: an algorithm for compressive sensing, *SIAM Journal on Numerical Analysis*, Vol. 49 issue 6, pp. 2543-2563 (2011).
- FL09 [23] S. Foucart and M-J. Lai, Sparsest solutions of underdetermined linear Systems via ℓ_q -minimization for $0 \leq q \leq 1$, *Applied and Computational Harmonic Analysis*, Vol. 26 issue 3, pp. 395-407 (2009).
- Frey2007 [24] B. J. Frey and D. Dueck, Clustering by passing messages between data points, *Science* Vol. 315 no. 5814, pp. 972-976 (2007).
- Frieze2016 [25] A. Frieze and M. Karoński, *Introduction to Random Graphs*, Cambridge University Press (2015).
- Hagen1992 [26] L. Hagen and A. B. Kahng, New spectral methods for ratio cut partitioning and clustering, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 11 issue 9, pp.1074-1085 (1992).
- Hajek2016 [27] B. Hajek, Y. Wu, and J. Xu, Achieving exact cluster recovery threshold via semidefinite programming, *IEEE Transactions on Information Theory* Vol. 62.5, pp. 2788-2797 (2016).
- Herman2010 [28] M. A. Herman and T. Strohmer, General deviants: An analysis of perturbations in compressed sensing, *IEEE Journal on Selected Topics in Signal Processing*, Vol. 4 issue 2, pp. 342-349 (2010).
- Holland1983 [29] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic block models: First steps, *Social Networks*, Vol. 5 issue 2, pp.109-137 (1983).
- Jain2010 [30] A. K. Jain, Data clustering: 50 years beyond K-means, *Pattern Recognition Letters*, Vol. 31 issue 8, pp. 651-666 (2010).
- Kaufman1990 [31] L. Kaufman and P. J. Rousseeuw, *Fitting Groups in Data. An Introduction to Cluster Analysis*, Wiley, New York (1990).
- Krzakala2013 [32] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová and P. Zhang, Spectral Redemption in clustering sparse networks *Proceedings of the National Academy of Sciences*, Vol. 110 no. 52, pp. 20935-20940 (2013).
- Le2015 [33] C. M. Le, E. Levina, and R. Vershynin, Sparse random graphs: regularization and concentration of the Laplacian, *arXiv preprint arXiv:1502.03049* (2015).

- [Li2016] [34] H. Li, Improved analysis of SP and CoSaMP under total perturbations, *EURASIP Journal on Advances in Signal Processing 2016*, no. 1, pp. 112 (2016).
- [Luxburg2007] [35] U. Von Luxburg, A tutorial on spectral clustering, *Statistics and Computing*, Vol. 17 issue 4, pp. 395-416 (2007).
- [Montanari2015] [36] A. Montanari and S. Sen, Semidefinite programs on sparse random graphs, *arXiv preprint arXiv:1504.05910* (2015).
- [Nascimento2011] [37] M. C. V. Nascimento and A. De Carvalho, Spectral methods for graph clustering - A survey, *European Journal of Operational Research*, Vol. 211 issue 2, pp. 221-231 (2011).
- [Needell2009] [38] D. Needell and J. A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, *Applied and Computational Harmonic Analysis*, Vol. 26 issue 3, pp. 301-321 (2009).
- [Newman2006] [39] M. E. J. Newman, Modularity and community structure in networks, *Proceedings of the National Academy of Sciences*, Vol. 103 no. 23, pp 8577-8582 (2006).
- [Newman2002] [40] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, Random graph models of social networks, *Proceedings of the National Academy of Sciences*, Vol. 99 suppl. 1, pp. 2566-2572 (2002).
- [Ng2001] [41] A. Y. Ng, M. I. Jordan, and Y. Weiss, On spectral clustering: analysis and an algorithm, *Advances in Neural Information Processing Systems*, pp. 849-856 (2002).
- [Olhede2014] [42] S. C. Olhede and P. J. Wolfe, Network histograms and universality of block model approximation, *arXiv preprint arXiv:312-5306v3* (2014).
- [Shi2000] [43] J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22 issue 8, pp. 888-905 (2000).
- [Thompson1972] [44] R. C. Thompson, Principal submatrices IX: Interlacing inequalities for singular values of submatrices, *Linear Algebra and Its Applications*, Vol. 5 issue 1, pp. 1-12 (1972).
- [Tremblay2016] [45] N. Tremblay, G. Puy, R. Gribonval and P. Vandergheynst, Compressive spectral clustering, *Machine Learning, Proceedings of the Thirty-third International Conference (ICML 2016)*, pp. 20-22 (2016).
- [Tropp2004] [46] J. A. Tropp, Greed is good: Algorithmic results for sparse approximation, *IEEE Transactions on Information Theory*, Vol. 50 issue 10, pp. 2231-2242 (2004).
- [Zachary1977] [47] W. W. Zachary, An Information flow model for conflict and fission in small groups, *Journal of Anthropological Research*, Vol. 33 issue 4, pp. 452-473 (1977).