Linear Algebra and its Applications 593 (2020) 45–73

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Exact recovery in the hypergraph stochastic block model: A spectral algorithm

Sam Cole [a,*], Yizhe Zhu [b,*]

[a] *Department of Mathematics, University of Manitoba, Winnipeg, MB R3T 2MB, Canada*
[b] *Department of Mathematics, University of California, San Diego, La Jolla, CA 92093, USA*

ARTICLE INFO

Article history:
Received 23 April 2019
Accepted 29 January 2020
Available online 1 February 2020
Submitted by S. Kirkland

MSC:
primary 54C40, 14E20
secondary 46E25, 20C20

Keywords:
Random hypergraph
Stochastic block model
Community detection
Spectral algorithm

ABSTRACT

We consider the exact recovery problem in the hypergraph stochastic block model (HSBM) with k blocks of equal size. More precisely, we consider a random d-uniform hypergraph H with n vertices partitioned into k clusters of size $s = n/k$. Hyperedges e are added independently with probability p if e is contained within a single cluster and q otherwise, where $0 \leq q < p \leq 1$. We present a spectral algorithm which recovers the clusters exactly with high probability, given mild conditions on n, k, p, q, and d. Our algorithm is based on the *adjacency matrix* of H, which is a symmetric $n \times n$ matrix whose (u, v)-th entry is the number of hyperedges containing both u and v. To the best of our knowledge, our algorithm is the first to guarantee exact recovery when the number of clusters $k = \Theta(\sqrt{n})$.

* Corresponding authors.
E-mail addresses: samuel.cole@umanitoba.ca (S. Cole), yiz084@ucsd.edu (Y. Zhu).

https://doi.org/10.1016/j.laa.2020.01.039
0024-3795/

1. Introduction

1.1. Hypergraph clustering

Clustering is an important topic in data mining, network analysis, machine learning and computer vision. Many clustering methods are based on graphs, which represent pairwise relationships among objects. However, in many real-world problems, pairwise relations are not sufficient, while higher order relations between objects cannot be represented as edges on graphs. Hypergraphs can be used to represent more complex relationships among data, and they have been shown empirically to have advantages over graphs; see [55,51]. Thus, it is of practical interest to develop algorithms based on hypergraphs that can handle higher-order relationships among data, and much work has already been done to that end; see, for example, [55,41,52,28, 13,33,6]. Hypergraph clustering has found a wide range of applications ([32,21,12,25, 38]).

The stochastic block model (SBM) is a generative model for random graphs with community structures which serves as a useful benchmark for the task of recovering community structure from graph data. It is natural to have an analogous model for random hypergraphs as a testing ground for hypergraph clustering algorithms.

1.2. Hypergraph stochastic block models

The *hypergraph stochastic block model*, first introduced in [26], is a generalization of the SBM for hypergraphs. We define the hypergraph stochastic block model (HSBM) as follows for d-uniform hypergraphs.

Definition 1.1 *(Hypergraph).* A d-uniform hypergraph H is a pair $H = (V, E)$ where V is a set of vertices and $E \subset \binom{V}{d}$ is a set of subsets with size d of V, called hyperedges. When $d = 2$, it is the same as an ordinary graph.

Definition 1.2 *(Hypergraph stochastic block model (HSBM)).* Let $\mathcal{C} = \{C_1, \dots C_k\}$ be a partition of the set $[n]$ into k sets of size $s = n/k$ (assume n is divisible by k), each $C_i, 1 \leq i \leq k$ is called a cluster. For constants $0 \leq q < p < 1$, we define the d-uniform hypergraph SBM as follows:

For any set of d distinct vertices $i_1, \dots i_d$, generate a hyperedge $\{i_1, \dots i_d\}$ with probability p if the vertices $i_1, \dots i_d$ are in the same cluster in $\mathcal{C}$. Otherwise, generate the hyperedge $\{i_1, \dots i_d\}$ with probability q. We denote this distribution of random hypergraphs as $H(n, d, \mathcal{C}, p, q)$. When $d = 2$, it is the same as the stochastic block models for random graphs.

Hypergraphs are closely related to symmetric tensors. We give a definition of symmetric tensors below. See, e.g., [39], for more details on tensors.

Definition 1.3 *(Symmetric tensor).* Let $T \in \mathbb{R}^{n\times\cdots\times n}$ be an order-d tensor. We call T symmetric if $T_{i_1,i_2,\ldots i_d} = T_{\sigma(i_1),\sigma(i_2)\ldots,\sigma(i_d)}$ for any $i_1,\ldots,i_d \in [n]$ and any permutation σ in the symmetric group of order d.

Formally, we can use a random symmetric tensor to represent a random hypergraph H drawn from this model. We construct an *adjacency tensor* T of H as follows. For any distinct vertices $i_1 < i_2 < \cdots < i_d$ that are in the same cluster,

$$T_{i_1,\ldots,i_d} = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1-p. \end{cases}$$

For any distinct vertices $i_1 < \cdots < i_d$, if any two of them are not in the same cluster, we have

$$T_{i_1,\ldots,i_d} = \begin{cases} 1 & \text{with probability } q, \\ 0 & \text{with probability } 1-q. \end{cases}$$

We set $T_{i_1,\ldots i_d} = 0$ if any two of the indices in $\{i_1,\ldots i_d\}$ coincide, and we set $T_{\sigma(i_1),\sigma(i_2)\ldots,\sigma(i_d)} = T_{i_1,\ldots i_d}$ for any permutation σ. Furthermore, we may abuse notation and write T_e in place of $T_{i_1,\ldots,i_d}$, where $e = \{i_1,\ldots,i_d\}$.

The HSBM recovery problem is to find the ground truth clusters $\mathcal{C} = \{C_1,\ldots,C_k\}$ either approximately or exactly, given a sample hypergraph from $H(n,d,\mathcal{C},p,q)$. We may ask the following questions about the quality of the solutions; see [1] for further details in the graph case:

(1) **Exact recovery (strong consistency):** Find $\mathcal{C}$ exactly (up to a permutation) with probability $1-o(1)$.
(2) **Almost exact recovery (weak consistency):** Find a partition $\hat{\mathcal{C}}$ such that $o(1)$ portion of the vertices are mislabeled.
(3) **Detection:** Find a partition $\hat{\mathcal{C}}$ which is correlated with the true partition $\mathcal{C}$.

We are typically interested in one of two regimes:

- **The dense regime.** In this regime p and q are constant, and the number of clusters k is allowed to grow with n. We then ask: how small can we make the cluster size $s = n/k$ while still being able to guarantee recovery?
- **The sparse regime.** In this regime k is constant, $s = \Theta(n)$, and $p,q = o(1)$. We then ask: how small can we make p and q while still being able to guarantee recovery?

Several methods have been considered for exact recovery of HSBMs. In [26], the authors used spectral clustering based on the hypergraph's Laplacian to recover HSBMs that are dense and uniform. Subsequently, they extended their results to sparse,

non-uniform hypergraphs in [27–29]. Spectral methods along with local refinements were considered in [16,4]. A semidefinite programing approach was analyzed in [37].

For different sparsity regimes, the efficient algorithms are not the same and there is no 'universal' algorithm that works optimally for all different sparsity regimes and different problems. For example, the detection problems of SBMs with bounded expected degrees are analyzed by algorithms based on self-avoiding walks [45] or non-backtracking walks [47,11]. However, for the exact recovery problems in the logarithmic degree regime, the algorithms that achieve the information theoretical threshold are based on semidefinite programming [2] or spectral clustering based on eigenvectors of the adjacency matrices [3]. In this paper we will only focus on the exact recovery problem and our algorithm might not work well for the almost exact recovery or detection problems.

1.3. Our results

In this paper, we present a spectral algorithm for exact recovery which compares well with previously known algorithms in the dense regime. Our main result is the following:

Theorem 1.4. *Let p, q, d be constant. For sufficiently large n, there exists a deterministic, polynomial time algorithm which exactly recovers d-uniform HSBMs with probability* $1 - \exp(-\Omega(\sqrt{n}))$ *if* $s = \Omega(\sqrt{n})$.

See Theorem 2.1 below for the precise statement. Our algorithm is based on the *iterative projection* algorithm developed in [19,18] for the graph case. We apply this approach to the *adjacency matrix* A of the random hypergraph H (see Definition 3.1). The challenge is that the adjacency matrix constructed from the adjacency tensor used in the algorithm does not have independent entries. In the process, we prove a non-asymptotic concentration result for the spectral norm of A, which may be of independent interest (Theorem 4.3) for other random hypergraph problems.

1.4. Why dense HSBMs?

While sparse (H)SBMs typically have more applications in data science, the dense case is nonetheless of theoretical importance. The SBM recovery problem is known alternately as the *planted partition* problem and can be seen as a variant of the *planted clique* problem originally posed in [36]. In the latter, one generates an Erdős-Rényi random graph $G\left(n, \frac{1}{2}\right)$ and adds edges deterministically to form a clique on an arbitrary subset of the vertices; the goal is then to determine exactly which vertices were members of the "planted" clique w.h.p. If the planted clique is too small, then there is no way to distinguish it from a randomly occurring clique in $G(n, \frac{1}{2})$; thus, the central question is how big the clique must be in order to guarantee (efficient) recovery.

For both planted clique and planted partition, it is statistically impossible to recover the clique or partition w.h.p. if the size of the clique or parts of the partition is

$O(\log n)$ [15]. On the other hand, the best known polynomial time algorithms in both problems require the size to be $\Omega(\sqrt{n})$ in order to guarantee recovery w.h.p. [7,8,14,19, 49], and there is evidence that this is best that can be done efficiently for exact recovery [22,23,36]. If the size of the partition or clique is $s = \Omega(\sqrt{n \log n})$, a simple counting algorithm similar to the one we present in Appendix A would work [40,15]. However, when the cluster sizes are only required to be $\Theta(\sqrt{n})$, the problem becomes more difficult because the error terms introduced by simple counting arguments are too large. It's still a major open problem in theoretical computer science to find polynomial time algorithms which succeed w.h.p. in the regime where the size of the clique or the partition is $o(\sqrt{n})$. See Section 1.4.1 in [15] for more discussion.

Thus, Theorem 1.4 is consistent with the state of the art for planted problems on graphs; moreover, our algorithm for HSBM recovery compares favorably with other known algorithms in the dense case with $k = \omega(1)$ clusters (see Section 1.5). To the best of our knowledge, our algorithm is the first to guarantee exact recovery when all clusters are size $\Theta(\sqrt{n})$. In Section 9, we include a more thorough discussion of limitations of SBM recovery algorithms.

As discussed at the end of Section 1.2, one should not expect one algorithm that works optimally for all sparsity regimes. While we focus on the dense case, our algorithm can be adapted to the sparse case as well (see Appendix B); however, it does not perform as well as previously known algorithms in [37,43,17,16,29] in the sparse regime. In fact, it is even outperformed by the simple hyperedge counting algorithm presented in Appendix A. The main obstacle is the lack of concentration for sparse hypergraphs: in our spectral algorithm (Algorithm 1), to make sure the iterative procedure succeed at every step, one needs to take a union bound over exponentially many events, which requires a concentration bound of the spectral norm with exponentially decaying tail probabilities, but sparse random matrices do not concentrate as well as dense random matrices. Optimizing our algorithm for sparse HSBMs is a possible direction for future work.

1.5. Comparison with previous results

We compare our spectral algorithm, as well as the simple counting algorithm presented in Appendix A, with previous exact recovery algorithms, with p, q, d being constant. In [4,16] the regime where k grows with n is not explicitly discussed, so we only include $k = O(1)$ case.

Paper	Number of clusters	Algorithm type
[37]	$O(1)$	Semidefinite programming
[5]	$O(1)$	Spectral + local refinement
[16]	$O(1)$	Spectral + local refinement
[29], Corollary 5.1	$o(\log^{\frac{-1}{2d}}(n) n^{\frac{d-4}{2d}})$	Spectral + k-means
Our result (Algorithm 2)	$O(\log^{\frac{-1}{2d-4}}(n) n^{0.5})$	Simple counting
Our result (Algorithm 1)	$O(n^{0.5})$	Spectral

2. Spectral algorithm and main results

Our main result is that Algorithm 1 below recovers HSBMs with high probability, given certain conditions on n, k, p, q, and d. It is an adaptation of the *iterated projection* algorithm for the graph case introduced by [19,18]. The algorithm can be broken down into three main parts:

(1) Construct an "approximate cluster" using spectral methods (Steps 1-4)
(2) Recover the cluster exactly from the approximate cluster by counting hyperedges (Steps 5-6)
(3) Delete the recovered cluster and recurse on the remaining vertices (Step 7).

Algorithm 1

Given $H = (V, E)$, $|V| = n$, number of clusters k, and cluster size $s = n/k$:

(1) Let A be the adjacency matrix of H (as defined in Section 3).
(2) Let $P_k(A) = (P_{uv})_{u,v \in V}$ be the dominant rank-k projector of A (as defined in Section 5).
(3) For each column v of $P_k(A)$, let $P_{u_1,v} \geq \ldots \geq P_{u_{n-1},v}$ be the entries other than P_{vv} in non-increasing order. Let $W_v := \{v, u_1, \ldots, u_{s-1}\}$, i.e., the indices of the $s-1$ greatest entries of column v of $P_k(A)$, along with v itself.
(4) Let $W = W_{v^*}$, where $v^* := \arg\max_v ||P_k(A)\mathbf{1}_{W_v}||_2$, i.e. the column v with maximum $||P_k(A)\mathbf{1}_{W_v}||_2$. It will be shown that W has small symmetric difference with some cluster C_i with high probability (Section 6).
(5) For all $v \in V$, let $N_{v,W}$ be the number of hyperedges e such that $v \in e$ and $e \setminus \{v\} \subseteq W$, i.e., the number of hyperedges containing v and $d-1$ vertices from W.
(6) Let C be the s vertices v with highest $N_{v,W}$. It will be shown that $C = C_i$ with high probability (Section 7).
(7) Delete C from H and repeat on the remaining sub-hypergraph. Stop when there are $< s$ vertices left.

Theorem 2.1. *Let H be sampled from $H(n, d, \mathcal{C}, p, q)$, where p and q are constant, $\mathcal{C} = \{C_1, \ldots, C_k\}$ and $|C_i| = s = n/k$ for $i = 1, \ldots, k$. If $d = o(s)$ and*

$$\frac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}} \leq \varepsilon \leq \frac{p-q}{32d}, \tag{2.1}$$

then for sufficiently large n, Algorithm 1 exactly recovers $\mathcal{C}$ with probability $\geq 1 - 2^k \cdot \exp(-s) - nk \cdot \exp\left(-\varepsilon^2\binom{s-1}{d-1}\right)$.

In the theorem above, the size d of the hyperedges is allowed to grow with n. The special case in which d is constant follows easily:

Theorem 2.2. *Let H be sampled from $H(n, d, \mathcal{C}, p, q)$, where p and q, and d are constant, $\mathcal{C} = \{C_1, \ldots, C_k\}$ and $|C_i| = s = n/k$ for $i = 1, \ldots, k$. If*

$$s \geq \frac{c_0\sqrt{nd}}{(p-q)^{\frac{2}{d-1}}},$$

then Algorithm 1 recovers $\mathcal{C}$ w.h.p., where c_0 is an absolute constant.

Proof. Observe that if

$$\frac{18d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s} \leq \frac{p-q}{32d}, \tag{2.2}$$

then we have

$$\frac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}} \leq \frac{18d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s} \leq \frac{p-q}{32d}.$$

Hence, if (2.2) is satisfied, then it is possible to choose ε satisfying (2.1), so Theorem 2.1 guarantees that we can recover $\mathcal{C}$ w.h.p. in this case. Recall that for nonnegative integers $a \geq b$ we can bound the binomial coefficient $\binom{a}{b}$ by $\left(\frac{a}{b}\right)^b \leq \binom{a}{b} \leq \left(\frac{ae}{b}\right)^b$. The conclusion follows by applying these bounds in (2.2) and solving for s. Note that we want the failure probability in Theorem 2.1 to be $o(1)$, so we require

$$\exp\left(-\varepsilon^2\binom{s-1}{d-1}\right) = o((nk)^{-1}).$$

It is easy to verify that this is satisfied if d is constant. □

In Appendix A we present a trivial hyperedge counting algorithm. Algorithm 1 beats this algorithm by a factor of $(\log n)^{\frac{1}{2d-4}}$. See Section 1.5 for comparison with other known algorithms.

The remainder of this paper is devoted to proving Theorem 2.1. Sections 3-5 introduce the linear algebra tools necessary for the proof; Section 6 shows that Step 4 with high probability produces a set with small symmetric difference with one of the clusters; Section 7 proves that Step 6 with high probability recovers one of the clusters exactly; and Section 8 proves inductively that the algorithm with high probability recovers all clusters.

2.1. Running time

In contrast to the graph case, in which the most expensive step is constructing the projection operator $P_k(A)$ (which can be done in $O(n^2k)$ time via truncated SVD [30,31]), for $d \geq 3$ the running time of Algorithm 1 is dominated by constructing the adjacency matrix A, which takes $O(n^d)$ time (the same amount of time it takes to simply read the input hypergraph). Thus, the overall running time of Algorithm 1 is $O(kn^d)$.

3. Reduction to random matrices

Since we do not have many linear algebra and probability tools for random tensors, it would be convenient if we could work with matrices instead of tensors. We

propose to analyze the following adjacency matrix of a hypergraph, originally defined in [24].

Definition 3.1 *(Adjacency matrix).* Let H be a random hypergraph generated from $H(n, d, \mathcal{C}, p, q)$ and let T be the adjacency tensor of H. For any hyperedge $e = \{i_1, \ldots, i_d\}$, let T_e be the entry in T corresponding to $T_{i_1,\ldots,i_d}$. We define the adjacency matrix A of H by

$$A_{ij} := \sum_{e:\{i,j\}\in e} T_e. \tag{3.1}$$

Thus, A_{ij} is the number of hyperedges in H that contains vertices i, j. Note that in the summation (3.1), each hyperedge is counted once.

From our definition, A is symmetric, and $A_{ii} = 0$ for $1 \le i \le n$. However, the entries in A are not independent. This presents some difficulty, but we can still get information about the clusters from this adjacency matrix A.

4. Eigenvalues and concentration of spectral norms

It is easy to see that for $d \ge 2$,

$$\mathbb{E}A_{ij} = \begin{cases} \binom{s-2}{d-2}(p-q) + \binom{n-2}{d-2}q, & \text{if } i \ne j \text{ are in the same cluster,} \\ \binom{n-2}{d-2}q, & \text{if } i, j \text{ are in different clusters.} \end{cases}$$

Let

$$\tilde{A} := \mathbb{E}A + \left(\binom{s-2}{d-2}(p-q) + \binom{n-2}{d-2}q\right) I,$$

then $\tilde{A}$ is a symmetric matrix of rank k. The eigenvalues for $\tilde{A}$ are easy to compute, hence by a shifting, we have the following eigenvalues for $\mathbb{E}A$. Note that we are using the convention $\lambda_1(X) \ge \ldots \ge \lambda_n(X)$ for a $n \times n$ self-adjoint matrix X.

Lemma 4.1. *The eigenvalues of* $\mathbb{E}A$ *are*

$$\begin{aligned} \lambda_1(\mathbb{E}A) &= \binom{s-2}{d-2}(p-q)(s-1) + \binom{n-2}{d-2}q(n-1), \\ \lambda_i(\mathbb{E}A) &= \binom{s-2}{d-2}(p-q)(s-1) - \binom{n-2}{d-2}q, \quad 2 \le i \le k, \\ \lambda_i(\mathbb{E}A) &= -\binom{s-2}{d-2}(p-q) - \binom{n-2}{d-2}q, \quad k+1 \le i \le n. \end{aligned}$$

We can use an ε-net chaining argument to prove a concentration inequality for the spectral norm of $A - \mathbb{E}A$.

Definition 4.2 *(ε-net).* An *ε-net* for a compact metric space $(\mathcal{X}, d)$ is a finite subset $\mathcal{N}$ of $\mathcal{X}$ such that for each point $x \in \mathcal{X}$, there is a point $y \in \mathcal{N}$ with $d(x, y) \leq \varepsilon$.

Theorem 4.3. *Let $\|\cdot\|_2$ be the spectral norm of a matrix, we have*

$$\|A - \mathbb{E}A\|_2 \leq 6d\sqrt{d\binom{n}{d-1}} \tag{4.1}$$

with probability at least $1 - e^{-n}$.

When $d = 2$, Theorem 4.3 is a concentration result for Wigner matrices. Our result includes the case where d is growing with n. Lemma 2 in [5] is a similar concentration result for the adjacency matrix A for random hypergraphs, but with probability $1 - O(1/n)$. However, to make our Algorithm 1 succeed with high probability with $k = \Theta(\sqrt{n})$ many clusters, we need a concentration bound of the spectral norm of A with exponentially small failure probability since we need to take a union bound over 2^k events in order to guarantee the algorithm's success (see Section 8.2).

Proof of Theorem 4.3. Consider the centered matrix $M := A - \mathbb{E}A$, then each entry M_{ij} is a centered random variable. Let $M_e = T_e - \mathbb{E}[T_e]$. Let $\mathbb{S}^{n-1}$ be the unit sphere in $\mathbb{R}^n$ (using the l_2-norm). By the definition of the spectral norm,

$$\|M\|_2 = \sup_{\mathbf{x} \in \mathbb{S}^{n-1}} |\langle M\mathbf{x}, \mathbf{x}\rangle|.$$

Let $\mathcal{N}$ be an ε-net on $\mathbb{S}^{n-1}$. Then for any $\mathbf{x} \in \mathbb{S}^{n-1}$, there exists some $\mathbf{y} \in \mathcal{N}$ such that $\|\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon$. Then we have

$$\|M\mathbf{x}\|_2 - \|M\mathbf{y}\|_2 \leq \|M\mathbf{x} - M\mathbf{y}\|_2 \leq \|M\|_2\|\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon\|M\|_2.$$

For any $\mathbf{y} \in \mathcal{N}$, if we take the supremum over $\mathbf{x}$, we have

$$(1 - \varepsilon)\|M\|_2 \leq \|M\mathbf{y}\|_2 \leq \sup_{\mathbf{z} \in \mathcal{N}} \|M\mathbf{z}\|_2.$$

Therefore

$$\|M\|_2 \leq \frac{1}{1 - \varepsilon} \sup_{\mathbf{x} \in \mathcal{N}} \|M\mathbf{x}\|_2 = \frac{1}{1 - \varepsilon} \sup_{\mathbf{x} \in \mathcal{N}} \langle M\mathbf{x}, \mathbf{x}\rangle \tag{4.2}$$

Now we fix an $\mathbf{x} = (x_1, \ldots, x_n)^\top \in \mathbb{S}^{n-1}$ first, and prove a concentration inequality for $\|M\mathbf{x}\|_2$. Let E be the hyperedge set in a complete d-uniform hypergraph on $[n]$. We have

$$\langle M\mathbf{x},\mathbf{x}\rangle = \sum_{i\neq j} M_{ij}x_ix_j = 2\sum_{i<j} M_{ij}x_ix_j = 2\sum_{i<j}(\sum_{e\in E: i,j\in e} M_e)x_ix_j$$
$$= 2\sum_{e\in E}(\sum_{i,j\in e, i<j} x_ix_j)M_e.$$

Let $Y_e := (\sum_{ij\in e, i<j} x_ix_j)M_e$, then

$$\langle M\mathbf{x},\mathbf{x}\rangle = 2\sum_{e\in E} Y_e$$

where $\{Y_e\}_{e\in E}$ are independent. Note that $|M_e| \leq 1$, so we have

$$|Y_e| = |\sum_{ij\in e, i<j} x_ix_jM_e| \leq |\sum_{ij\in e, i<j} x_ix_j|$$

By Hoeffding's inequality,

$$\mathbb{P}(|\sum_{e\in E} Y_e| \geq t) \leq 2\exp\left(-\frac{2t^2}{4\sum_{e\in E}|\sum_{ij\in e, i<j} x_ix_j|^2}\right). \tag{4.3}$$

From Cauchy's inequality, we have

$$\sum_{e\in E}|\sum_{ij\in e, i<j} x_ix_j|^2 \leq \binom{d}{2}\sum_{e\in E}\sum_{ij\in e, i<j} x_i^2x_j^2 \leq \binom{d}{2}\binom{n-2}{d-2}\sum_{1\leq i<j\leq n} x_i^2x_j^2$$
$$\leq \binom{d}{2}\binom{n-2}{d-2}\frac{1}{2}\left(\sum_i x_i^2\right)^2 \leq \frac{1}{4}d^2\binom{n}{d-2}. \tag{4.4}$$

Therefore from (4.3) and (4.4),

$$\mathbb{P}(|\langle M\mathbf{x},\mathbf{x}\rangle| \geq 2t) \leq 2\exp\left(-\frac{2t^2}{\binom{n}{d-2}d^2}\right).$$

Taking $t = \frac{3}{2}d\sqrt{d\binom{n}{d-1}}$, we have

$$\mathbb{P}\left(|\langle M\mathbf{x},\mathbf{x}\rangle| \geq 3d\sqrt{d}\sqrt{\binom{n}{d-1}}\right) \leq 2\exp\left(-\frac{9d\binom{n}{d-1}}{2\binom{n}{d-2}}\right) \leq \exp(-3n).$$

Since $|\mathcal{N}| \leq (\frac{2}{\varepsilon}+1)^n$ (see Corollary 4.2.11 in [53] for example), we can take $\varepsilon = 1/2$ and by a union bound, we have

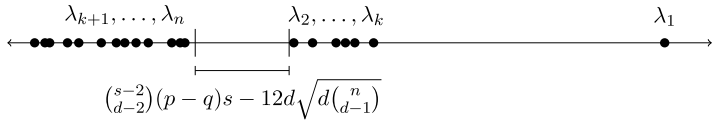

Fig. 1. The distribution of eigenvalues of A.

$$\mathbb{P}\left(\sup_{\mathbf{x}\in\mathcal{N}} |\langle M\mathbf{x}, \mathbf{x}\rangle| \geq 3d\sqrt{d}\sqrt{\binom{n}{d-1}}\right) \leq 5^n \exp(-3n) \leq e^{-n}. \tag{4.5}$$

So we have from (4.2), (4.5)

$$\mathbb{P}\left(\|M\|_2 \geq 6d\sqrt{d\binom{n}{d-1}}\right) \leq e^{-n}. \quad \square$$

Since $|\lambda_i(A) - \lambda_i(\mathbb{E}A)| \leq \|A - \mathbb{E}A\|_2$ for $1 \leq i \leq n$, we see that the largest k eigenvalues of A are separated from the remaining $n-k$ by at least $\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}$. Fig. 1 depicts this separation in the eigenvalues, which is necessary to bound the difference in the dominant rank-k projectors of A and $\mathbb{E}A$ in the next section.

5. Dominant eigenspaces and projectors

Our recovery algorithm is based on the *dominant rank-k projector* of the adjacency matrix A.

Definition 5.1 *(Dominant eigenspace).* If X is a $n \times n$ Hermitian or real symmetric matrix, the dominant r-dimensional eigenspace of X, denoted $\mathbf{E}_r(X)$, is the subspace of $\mathbb{R}^n$ or $\mathbb{C}^n$ spanned by eigenvectors of X corresponding to its r largest eigenvalues.

Note that by this definition, if $\lambda_r(X) = \lambda_{r+1}(X)$, then $\mathbf{E}_r(X)$ actually has dimension $> r$, but that will never be the case in this analysis.

Definition 5.2 *(Dominant rank-r projector).* If X is a $n \times n$ Hermitian or real symmetric matrix, the dominant rank-r projector of X, denoted $P_r(X)$, is the orthogonal projection operator onto $\mathbf{E}_r(X)$.

$P_r(X)$ is a rank-r, self-adjoint operator which acts as the identity on $\mathbf{E}_r(X)$. It has r eigenvalues equal to 1 and $n - r$ equal to 0. If $\mathbf{v}_1, \ldots, \mathbf{v}_r$ is an orthonormal basis for $\mathbf{E}_r(X)$, then

$$P_r(X) = \sum_{i=1}^{r} \mathbf{v}_i \mathbf{v}_i^*, \tag{5.1}$$

where $\mathbf{v}^*$ denotes either the transpose or conjugate transpose of $\mathbf{v}$, depending on whether we are working over $\mathbb{R}$ or $\mathbb{C}$. Let us define Y to be the *incidence matrix* of $\mathcal{C}$; i.e.,

$$Y_{uv} := \begin{cases} 1 & \text{if } u, v \text{ are in the same part of } \mathcal{C}, \\ 0 & \text{else} \end{cases} \tag{5.2}$$

Thus, it is our goal to reconstruct Y given $H \sim H(n, d, \mathcal{C}, p, q)$.

Theorem 5.3. *Let $A, \mathbb{E}A$, and $\tilde{A}$ be defined as in Sections 3 and 4. Then*

$$P_k(\mathbb{E}A) = P_k(\tilde{A}) = P_k(Y) = \frac{1}{s}Y.$$

Proof. Let $\mathbf{1}_{C_i} \in \{0,1\}^n$ denote the indicator vector for cluster C_i and J_n the $n \times n$ all ones matrix. Then we can write

$$Y = \sum_{i=1}^{k} \mathbf{1}_{C_i}\mathbf{1}_{C_i}^\top, \quad \tilde{A} = (a-b)Y + bJ_n, \quad \mathbb{E}A = \tilde{A} - aI_n,$$

for some constants $a > b > 0$. Thus, $\left\{\frac{1}{\sqrt{s}}\mathbf{1}_{C_i} : i = 1, \ldots, k\right\}$ is an orthonormal basis for the column space of both Y and $\tilde{A}$, and hence, in accordance with (5.1),

$$P_k(Y) = P_k(\tilde{A}) = \sum_{i=1}^{k} \frac{1}{s}\mathbf{1}_{C_i}\mathbf{1}_{C_i}^\top = \frac{1}{s}Y.$$

Now, observe that the eigenvalues of $\mathbb{E}A$ are those of $\tilde{A}$ shifted down by a, and $\mathbf{v}$ is an eigenvector of $\mathbb{E}A$ if and only if it is an eigenvector of $\mathbb{E}A$; hence, the dominant k-dimensional eigenspace of $\mathbb{E}A$ is the same as the column space of $\tilde{A}$, and therefore $P_k(\mathbb{E}A) = P_k(\tilde{A})$. □

Thus, $P_k(\mathbb{E}A) = P_k(\tilde{A})$ gives us all the information we need to reconstruct Y. Unfortunately, a SBM recovery algorithm doesn't have access to $\mathbb{E}A$ or $\tilde{A}$ (if it did the problem would be trivial), but the following theorem shows that the random matrix $P_k(A)$ is a good approximation to $P_k(\mathbb{E}A)$ and thus reveals the underlying rank-k structure of A:

Theorem 5.4. *Assume (4.1) holds. Then*

$$\|P_k(A) - P_k(\mathbb{E}A)\|_2 \leq \varepsilon$$

and

$$\|P_k(A) - P_k(\mathbb{E}A)\|_{\mathrm{F}} \leq \sqrt{2k}\varepsilon$$

for any $\varepsilon \geq \dfrac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}}$.

To prove Theorem 5.4, we use the following Lemma from [18, Lemma 4].

Lemma 5.5. *Let* $X, Y \in \mathbb{R}^{n\times n}$ *be symmetric. Suppose that the largest* k *eigenvalues of both* X, Y *are at least* β*, and the remaining* $n-k$ *eigenvalues of both* X, Y *are at most* α*, where* $\alpha < \beta$*. Then*

$$\|P_k(X) - P_k(Y)\|_2 \leq \frac{\|X-Y\|_2}{\beta-\alpha}, \tag{5.3}$$

$$\|P_k(X) - P_k(Y)\|_F \leq \frac{\sqrt{2k}\|X-Y\|_2}{\beta-\alpha}. \tag{5.4}$$

Proof of Theorem 5.4. Apply Lemma 5.5 with $X = A$, $Y = \mathbb{E}A$ and

$$\alpha = \binom{s-2}{d-2}(p-q)(s-1) - \binom{n-2}{d-2}q - 6d\sqrt{d\binom{n}{d-1}},$$

$$\beta = -\binom{s-2}{d-2}(p-q) - \binom{n-2}{d-2}q + 6d\sqrt{d\binom{n}{d-1}}.$$

Note that in order for this to work we need $\alpha > \beta$, i.e.

$$\binom{s-2}{d-2}(p-q)s > 12d\sqrt{d\binom{n}{d-1}}. \quad \square$$

6. Constructing an approximate cluster

In this section we show how to use $P_k(A)$ to construct an "approximate cluster", i.e. a set with small symmetric difference with one of the clusters. We will show that

- If $|W| = s$ and $\|P_k(A)\mathbf{1}_W\|_2$ is large, then W must have large intersection with some cluster (Lemma 6.1)
- Such a set W exists among the sets $W_1, \ldots, W_n$, where W_v is the indices of the $s-1$ largest entries in column v of $P_k(A)$, along with v itself (Lemma 6.2).

The intuition is that if $\|P_k(A) - P_k(\mathbb{E}A)\|_2 \leq \varepsilon$, then

$$\|P_k(A)\mathbf{1}_W\|_2^2 \approx \|P_k(\mathbb{E}A)\mathbf{1}_W\|_2^2 = \frac{1}{s}\sum_{i=1}^{k}|W \cap C_i|^2,$$

and this quantity is maximized when W comes mostly from a single cluster C_i.

Lemmas 6.1 and 6.2 below are essentially the same as Lemmas 18 and 17 in [19]. As $P_k(A) = \frac{1}{s}\sum_i \mathbf{1}_{C_i}\mathbf{1}_{C_i}^\top$ as in the graph case (Theorem 5.3), we can import their proofs directly from the graph case. However, we present a simpler proof for Lemma 6.1.

Lemma 6.1. *Assume* (4.1) *holds and* $\dfrac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}} \le \varepsilon \le \dfrac{1}{12}$. *Let* $|W| = s$ *and* $||P_k(A)\mathbf{1}_W||_2 \ge (1-2\varepsilon)\sqrt{s}$. *Then* $|W \cap C_i| \ge (1-6\varepsilon)s$ *for some* i.

Proof. By Theorem 5.4,

$$\|(P_k(A) - P_k(\mathbb{E}A))\mathbf{1}_W\|_2 \le \varepsilon\|\mathbf{1}_W\|_2 = \varepsilon\sqrt{s}.$$

And by the triangle inequality,

$$\|P_k(\mathbb{E}A)\mathbf{1}_W\|_2 \ge \|P_k(A)\mathbf{1}_W\|_2 - \varepsilon\sqrt{s} \ge (1-2\varepsilon)\sqrt{s} - \varepsilon\sqrt{s} = (1-3\varepsilon)\sqrt{s}. \tag{6.1}$$

We will show that in order for this to hold, W must have large intersection with some cluster.

Fix t such that $\frac{s}{2} \le t \le s$. Assume by way of contradiction that $|W \cap C_i| \le t$ for all i. Observe that by Theorem 5.3

$$\|P_k(\mathbb{E}A)\mathbf{1}_W\|_2^2 = \frac{1}{s}\sum_{i=1}^{k} |W \cap C_i|^2. \tag{6.2}$$

Let $x_i = |W \cap C_i|$ and consider the optimization problem

$$\begin{aligned} \max\ & \frac{1}{s}\sum_{i=1}^{k} x_i^2 \\ \text{s.t.}\ & \sum_{i=1}^{k} x_i = s, \\ & 0 \le x_i \le t \text{ for } i = 1, \dots, k. \end{aligned}$$

It is easy to see that the maximum occurs when $x_i = t, x_j = s - t$ for some i, j, $x_l = 0$ for all $l \ne i, j$, and the maximum is $\frac{t^2}{s} + \frac{(s-t)^2}{s}$. Thus, by (6.1) and (6.2) we have

$$(1-3\varepsilon)^2 s \le \|P_k(\mathbb{E}A)\mathbf{1}_W\|_2^2 \le \frac{t^2}{s} + \frac{(s-t)^2}{s}.$$

Solving for t, this implies that

$$t \ge \left(\frac{1}{2} + \frac{1}{2}\sqrt{1 - 12\varepsilon + 18\varepsilon^2}\right) s > (1-6\varepsilon)s.$$

Thus, if we choose $t \in [s/2, (1-6\varepsilon)s]$ we have a contradiction. Let us choose t to be as large as possible, $t = (1-6\varepsilon)s$. Then it must be the case that $|W \cap C_i| \geq t = (1-6\varepsilon)s$ for some i. Note that for the proof to go through we require $\frac{1}{2} \leq 1-6\varepsilon$, which is satisfied if $\varepsilon \leq 1/12$. □

This lemma gives us a way to identify an "approximate cluster" using only A; however, it would take $\Omega(n^s)$ time to try all sets W of size s. However, if we define W_v to be v along with the indices of the $s-1$ largest entries of column v of $P_k(A)$ (as in Step 3 of Algorithm 1), then Lemma 6.2 below will show that one of these sets satisfies the conditions of Lemma 6.1; thus, we can produce an approximate cluster in polynomial time by taking the W_v that maximizes $||P_k(A)\mathbf{1}_{W_v}||_2$.

Lemma 6.2. *Assume (4.1) holds and* $\varepsilon \geq \dfrac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}}$. *For* $v = 1, \ldots, n$, *let* W_v *be defined as in Step 3 of Algorithm 1. Then there exists a column* v *such that*

$$\|P_k(A)\mathbf{1}_{W_v}\|_2 \geq (1 - 8\varepsilon^2 - \varepsilon)\sqrt{s} \geq (1-2\varepsilon)\sqrt{s}.$$

Lemmas 6.1 and 6.2 together prove that, as long as (4.1) holds, Steps 2-4 successfully construct a set W such that $|W| = s$ and $|W \cap C_i| \geq (1-6\varepsilon)s$ for some i. In the following section we will see how to recover C_i exactly from W.

7. Exact recovery by counting hyperedges

Suppose we have a set $W \subset [n]$ such that $|W \triangle C_i| \leq \varepsilon s$ for some i ($\triangle$ denotes symmetric difference). In the graph case ($d = 2$) we can use W to recover C_i exactly w.h.p. as follows:

(1) Show that w.h.p. for any $u \in C_i$ will have at least $(p-\varepsilon)s$ neighbors in C_i, while any $v \notin C_i$ will have at most $(q+\varepsilon)s$ neighbors in C_i. This follows from a simple Hoeffding argument.
(2) Show that, if these bounds hold for any u, v, then (deterministically) any $u \in C_i$ will have at least $(p-2\varepsilon)$ neighbors in W, while any $v \notin C_i$ will have at most $(q+2\varepsilon)$ neighbors in W. Thus, we can use number of vertices in W to distinguish between vertices in C_i and vertices in other clusters.

See [19, Lemmas 19-20] for details. The reason we cannot directly apply a Hoeffding argument to W is that W depends on the randomness of the instance A, thus the number of neighbors a vertex has in W is not the sum of $|W|$ fixed random variables.

To generalize to hypergraphs with $d > 2$, an obvious analogue of the notion of number of neighbors a vertex u has in a vertex set W is to define the random variable

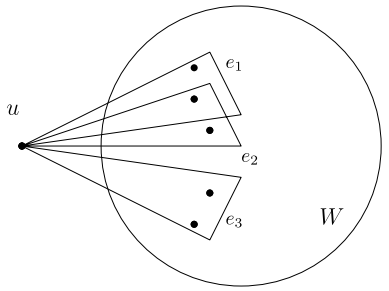

Fig. 2. When $d = 3$, $N_{u,W}$ is the number of hyperedges containing u and 2 vertices in W. In the figure above, $N_{u,W} = 3$.

$$N_{u,W} := \sum_{e:u\in e, e\setminus\{u\}\subseteq W} T_e,$$

i.e. the number of hyperedges containing u and $d-1$ vertices from W. When $d = 2$ this is simply the number of neighbors u has in W (see Fig. 2 for the case $d = 3$). We get the following analogue to [19, Lemma 19].

Lemma 7.1. *Consider cluster C_i and vertex $u \in [n]$, and let $\varepsilon > 0$. If $u \in C_i$, then for n sufficiently large and $d = o(s)$,*

$$N_{u,C_i} \geq (p-\varepsilon)\binom{s}{d-1} \tag{7.1}$$

with probability $\geq 1 - \exp\left(-\varepsilon^2\binom{s-1}{d-1}\right)$, and if $u \notin C_i$, then

$$N_{u,C_i} \leq (q+\varepsilon)\binom{s}{d-1} \tag{7.2}$$

with probability $\geq 1 - \exp\left(-\varepsilon^2\binom{s-1}{d-1}\right)$.

Proof. For $u \in C_i$, N_{u,C_i} is the sum of $\binom{s-1}{d-1}$ independent Bernoulli random variables with expectation p, so Hoeffding's inequality yields

$$\begin{aligned}
\mathbb{P}\left(N_{u,C_i} \leq (p-\varepsilon)\binom{s}{d-1}\right) &= \mathbb{P}\left(N_{u,C_i} \leq \left(p - \frac{\varepsilon s - (d-1)p}{s-d+1}\right)\binom{s-1}{d-1}\right) \\
&\leq \exp\left(-2\left(\frac{\varepsilon s-(d-1)p}{s-d+1}\right)^2\binom{s-1}{d-1}\right) \\
&\leq \exp\left(-\varepsilon^2\binom{s-1}{d-1}\right)
\end{aligned}$$

Note that the last inequality holds for $d = o(s)$ and n sufficiently large.

For $v \notin C_i$, N_{v,C_i} is the sum of $\binom{s}{d-1}$ independent Bernoulli random variables with expectation q. So by Hoeffding's inequality again

$$\mathbb{P}\left(N_{u,C_i} \geq (q+\varepsilon)\binom{s}{d-1}\right) \leq \exp\left(-2\varepsilon^2\binom{s-1}{d-1}\right) \leq \exp\left(-\varepsilon^2\binom{s-1}{d-1}\right). \quad \square$$

The difficulty is in going from N_{u,C_i} to $N_{u,W}$, where W is a set such that $|W \triangle C_i| \leq \varepsilon s$. We have the following estimate for $N_{u,W}$.

Lemma 7.2. *Let $W \subset [n]$ such that $|W| = s$ and $|W \cap C_i| \geq (1-6\varepsilon)s$ for some $i \in [k]$. Then for $\varepsilon < \frac{1}{16d}$, $d = o(s)$, and n sufficiently large, we have the following:*

(1) If $j \in C_i$ satisfies (7.1), then $N_{j,W} \geq (p - 16d\varepsilon)\binom{s}{d-1}$,

(2) If $j \notin C_i$ satisfies (7.2), then $N_{j,W} \leq (q + 16d\varepsilon)\binom{s}{d-1}$.

Proof. Assume $j \in C_i$ and j satisfies (7.1). As $|C_i| = s$, we have $|C_i \setminus W| \leq 6\varepsilon s$.

Let $\tilde{N}_{j,C_i\setminus W}$ be the number of hyperedges containing j and $d-1$ vertices from C_i, among which at least one vertex from $C_i \setminus W$. We then have

$$\begin{aligned} N_{j,W} &\geq N_{j,W\cap C_i} = N_{j,C_i} - \tilde{N}_{j,C_i\setminus W} \\ &\geq (p-\varepsilon)\binom{s}{d-1} - \sum_{m=1}^{d-1}\binom{\lceil 6\varepsilon s\rceil}{m}\binom{s}{d-1-m}. \end{aligned}$$

In the inequality above, we bound $\tilde{N}_{j,C_i\setminus W}$ by a deterministic counting argument, i.e. we count all possible hyperedges that include a vertex j, with m vertices from $C_i \setminus W$ and remaining $(d-1-m)$ vertices from C_i for $1 \leq m \leq d-1$.

Note that we can choose $\varepsilon < \dfrac{1}{16d}$ then for n sufficiently large, we have

$$\begin{aligned} \sum_{m=1}^{d-2}\binom{\lceil 6\varepsilon s\rceil}{m}\binom{s}{d-1-m} &\leq \sum_{m=1}^{d-2}\binom{7\varepsilon s}{m}\binom{s}{d-1-m} \\ &\leq \binom{s}{d-1}\sum_{m=1}^{d-1}(7\varepsilon s)^m\frac{(s-d+1)!(d-1)!}{(s-d+1+m)!(d-1-m)!} \\ &\leq \binom{s}{d-1}\sum_{m=1}^{d-1}\left(\frac{7\varepsilon sd}{s-d+2}\right)^m \\ &\leq \binom{s}{d-1}\frac{14\varepsilon sd}{s-d+2} \leq 15d\varepsilon\binom{s}{d-1} \end{aligned}$$

So we have

$$N_{j,W} \geq (p - 16d\varepsilon)\binom{s}{d-1}.$$

If $j \notin C_i$, let $\tilde{N}_{j,W\setminus C_i}$ be the number of hyperedges containing j and $d-1$ vertices from W, among which at least one vertex from $W \setminus C_i$. Recall $|W \setminus C_i| \leq 6\varepsilon s$.

$$\begin{aligned} N_{j,W} &\leq N_{j,C_i \cup W} = N_{j,C_i} + \tilde{N}_{j,W\setminus C_i} \\ &\leq (q+\varepsilon)\binom{s}{d-1} + \sum_{m=1}^{d-1}\binom{\lceil 6\varepsilon s\rceil}{m}\binom{s}{d-1-m} \\ &\leq (q+\varepsilon)\binom{s}{d-1} + \sum_{m=1}^{d-1}\binom{7\varepsilon s}{m}\binom{s}{d-1-m} \\ &\leq (q+16d\varepsilon)\binom{s}{d-1}. \quad \square \end{aligned}$$

Lemma 7.2 gives us a way to distinguish vertices $j \in C_i$ and $j \notin C_i$ provided $p - 16d\varepsilon > q + 16d\varepsilon$.

8. Proof of algorithm's correctness

We now have all the necessary pieces to prove the correctness of Algorithm 1 (Theorem 2.1). The proof is roughly the same as that of [19, Theorem 4].

8.1. Proof of correctness of first iteration

Lemmas 6.1-7.2 above prove that Steps 1-6 of Algorithm 1 correctly recover a single cluster in the first iteration.

Theorem 8.1. *Assume that* (4.1) *holds and that for* $i = 1, \ldots, k$, (7.1) *holds for all* $u \in C_i$ *and* (7.2) *holds for all* $u \notin C_i$ *with*

$$\frac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}} \leq \varepsilon \leq \frac{p-q}{32d}.$$

Then Steps 1-6 of Algorithm 1 exactly recover a cluster C_i *in the first iteration.*

Proof. By Lemma 6.2, the set W constructed in Step 4 has $||P_k(A)\mathbf{1}_W||_2 \geq (1-2\varepsilon)\sqrt{s}$. By Lemma 6.1 (noting that $\varepsilon \leq \frac{p-q}{32d} \leq \frac{1}{12}$), $|W \cap C_i| \geq (1-6\varepsilon)s$ for some i. And by Lemma 7.2, $N_{u,W} \geq (1-16\varepsilon)s$ for all $u \in C_i$, while $N_{u,W} \leq (q+16d\varepsilon)s$ for all $u \notin C_i$.

If $\varepsilon < \frac{p-q}{32d}$, then $(p-16d\varepsilon)s > (q+16d\varepsilon)s$. Thus, when we take the s vertices u with highest $N_{u,W}$ in Step 6, for each of them we have

$$N_{u,W} \geq (p-16d\varepsilon)s > (q+16d\varepsilon)s,$$

so none of them could possibly come from $[n] \setminus C_i$. Therefore, the set C constructed in Step 6 must be equal to C_i. □

8.2. The "delete and repeat" step

The difficulty with proving the success of Algorithm 1 beyond the first iteration is that the iterations cannot be handled independently: whether or not the t-th iteration succeeds determines which vertices will be left in the $(t+1)$-st iteration, which certainly affects whether or not the $(t+1)$-st iteration succeeds. However, notice that there is nothing probabilistic in the statement or proof of Theorem 8.1: if certain conditions are true, then the first iteration of Algorithm 1 will definitely recover a cluster. In fact, the only probabilistic statements thus far are in Theorem 4.3 and Lemma 7.1. Similar to the analysis in [19,18], we will show that if certain (exponentially many) conditions are met, then *all* iterations of Algorithm 1 will succeed. We will then show that all of these events occur simultaneously w.h.p.; hence, Algorithm 1 recovers all clusters w.h.p.

We begin by introducing some terminology:

Definition 8.2 *(Cluster subhypergraph, cluster subtensor).* We define a cluster subhypergraph to be a subhypergraph of H induced by a subset of the clusters $C_1, \ldots, C_k$. Similarly, we define a cluster subtensor to be the principal subtensor of T formed by restricting the indices to a subset of the clusters. For $J \subseteq [k]$, we denote by $H^{(J)}$ the subhypergraph of H induced by $\bigcup_{j\in J} C_j$, and we denote by $T^{(J)}$ the principal subtensor of T with indices restricted to $\bigcup_{j\in J} C_j$.

We now define two types of events on our probability space $H(n,d,\mathcal{C},p,q)$:

- *Spectral events* – For $J \subseteq [k]$, let E_J be the event that

$$\|B - \mathbb{E}B\|_2 \leq 6d\sqrt{d\binom{m}{d-1}},$$

  where B is the adjacency matrix of $H^{(J)}$ and $m = s|J|$ is the number of vertices in $H^{(J)}$. Note that B is *not* simply a submatrix of A, as only a subset of the edges of H are counted when computing the entries of B.
- *Degree events* – For $1 \leq i \leq k, 1 \leq u \leq n$, let $D_{i,u}$ be the event that $N_{u,C_i} \geq (p-\varepsilon)\binom{s}{d-1}$ if $u \in C_i$, or the event that $N_{u,C_i} \leq (q+\varepsilon)\binom{s}{d-1}$ if $u \notin C_i$. These

are the events that each vertex u has approximately the correct value of N_{u,C_i} for each cluster C_i.

Observe that there are 2^k spectral events and nk degree events. We will now show that if all of these events occur, then Algorithm 1 will definitely succeed in recovering all clusters. Again, there is nothing probabilistic in this theorem or its proof.

Lemma 8.3. *Assume that E_J holds for all $J \subseteq [k]$ and $D_{i,u}$ holds for all $i \in [k], u \in [n]$ with*

$$\frac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{n}{d-1}}} \leq \varepsilon \leq \frac{p-q}{32d}. \tag{8.1}$$

Then Algorithm 1 recovers $C_1, \ldots, C_k$ exactly.

Proof sketch. We omit the full proof as it is analogous to the proof in [19, Section 7.3]. Essentially, we prove by induction that the t-th iteration succeeds for $t = 1, \ldots, k$. If the 1st through tth iterations succeed, then the $(t+1)$-st iteration receives as input a cluster sub-hypergraph $H^{(J)}$, for some $J \subseteq [k]$. Hence, E_J and $D_{i,u}$ for $i \in J$, $u \in \bigcup_{j \in J} C_j$ ensure the success of the $(t+1)$-st iteration. Note that if there are $m = |J|s$ vertices remaining, then Theorem 5.4 requires that

$$\varepsilon \geq \frac{6d\sqrt{d\binom{m}{d-1}}}{\binom{s-2}{d-2}(p-q)s - 12d\sqrt{d\binom{m}{d-1}}},$$

but this bound is largest when $m = n$; thus, the condition (8.1) is sufficient for all iterations. □

Finally, we show that all of the E_J and $D_{i,u}$ hold simultaneously w.h.p.

Lemma 8.4. *E_J and $D_{i,u}$ hold simultaneously for all $J \subseteq [k], i \in [k], u \in [n]$ with probability $\geq 1 - 2^k \cdot \exp(-s) - nk \cdot \exp\left(-\varepsilon^2 \binom{s-1}{d-1}\right)$ for any ε satisfying condition (8.1).*

Proof. For any *fixed* $J \subseteq [k]$, $H^{(J)}$ is simply an instance of a smaller HSBM; it has distribution $H(|J|s, d, \bigcup_{j \in J} C_j, p, q)$. Thus,

$$\mathbb{P}\left(\overline{E_J}\right) \leq \exp(-|J|s) \leq \exp(-s)$$

by Theorem 4.3. And for any $i \in [k], u \in [n]$,

$$\mathbb{P}\left(\overline{D_{i,u}}\right) \leq \exp\left(-\varepsilon^2 \binom{s-1}{d-1}\right)$$

by Lemma 7.1. The proof is completed by taking a union bound over all $J \subseteq [k], i \in [k], u \in [n]$. □

Theorem 2.1 follows as an immediate corollary to Lemmas 8.3 and 8.4.

9. Lower bounds for HSBM recovery

There have been many results which show that, for a fixed number of blocks, (H)SBM recovery becomes impossible if the edge probabilities are below a certain threshold. For exact recovery of HSBMs with two blocks, it was shown in [43,17,16] that the phase transition from impossible to possible occurs in the regime of logarithmic average degree by analyzing the minimax risk, and the exact threshold was given in [37] by a generalization of the techniques in [2] for graph SBMs. For the detection problem of HSBMs in the bounded expected degree regime, a phase transition was conjectured in [9] based on belief propagation and the non-backtracking operator. Very recently, a spectral method based on the self-avoiding walk was proved to achieve the conjectured threshold [50].

For graph SBMs, a phase transition for the detection problem in the bounded expected degree regime with finitely many blocks, called the Kesten-Stigum threshold, was conjectured in [20] and then proved in [47,45,46] for the 2-block case. Below the Kesten-Stigum threshold no algorithms (even with exponential running time) will solve the detection problem, while above the threshold detection is not only possible but can be done in polynomial time. See [1] for further details. The phase transition behavior for the exact recovery problem with two blocks was proved in [2]. Above the threshold, there are polynomial time algorithms that solve the problem [2,3,48]. Minimax lower bounds for general SBMs with finite or a growing number of blocks were given in [15,35,54].

Relatively little is known in the dense regime, even in the graph case, and most results focus on the related *planted clique* problem [7,36] rather than SBM recovery (a.k.a. planted partition). It is generally believed that these problems become intractable when the size of the clique/blocks is $o(\sqrt{n})$. In this case, one would ideally like to prove that these problems are distNP-complete (where distP and distNP are distributional analogues of P and NP [42]). However, showing the existence of a "natural" distNP-complete problem is itself a long-outstanding problem in complexity theory [10].

Instead, various authors have shown that certain types of algorithms will provably fail if the size of the planted clique/blocks is too small. The first such result dates back to the original paper introducing the planted clique problem [36], in which the author showed that the Metropolis-Hastings algorithm fails to recover planted cliques of size $n^{1/2-\varepsilon}$ for any $\varepsilon > 0$. It was subsequently shown in [15,22] that certain optimization-based approaches also fail in this regime, while it was shown in [23] that *statistical algorithms* also run into the same barrier. (A statistical algorithm is an algorithm which, instead of receiving samples from a distribution, can query an oracle for statistics on the distribution within some tolerance. Such algorithms can be used to simulate many standard algorithms on randomized input.)

Extending these results for planted clique to dense SBM and HSBM recovery appears to be a promising direction for future work. In the d-uniform hypergraph case, it is unclear whether the barrier should be $\sqrt{n}$, $n^{1/d}$ or something else. It seems plausible that the barrier could be less than $\sqrt{n}$ for $d > 2$, since d-uniform HSBMs have $\binom{n}{d}$ independent random variables in play compared with only $\binom{n}{2}$ in the graph case, and thus we may expect certain random variables (e.g. degrees) to be more tightly concentrated about their expectations. However, it seems doubtful that a spectral algorithm could do better than $\sqrt{n}$ using only the adjacency matrix (see Definition 3.1), as this is the error term introduced by concentration results for the spectral norm of a random symmetric $n \times n$ matrix (see, e.g., [53]); to break the $\sqrt{n}$ barrier, we suspect that one must use the spectral properties of the *adjacency tensor* and not simply reduce to the adjacency matrix.

While proving lower bounds for *efficient* HSBM recovery appears to be difficult, one can readily prove that exact recovery is impossible for *any* algorithm, regardless of running time, if the cluster size is small enough (i.e., an information theoretic lower bound). We will follow the proof ideas from [15,34] for graph SBM recovery to prove an information theoretic lower bound for HSBMs.

Recall the definition of incidence matrix of a partition (5.2). Conversely, let $\mathcal{C}(Y)$ denote the partition of $[n]$ whose incidence matrix is Y. Let $\mathcal{Y}$ be the set of all incidence matrices corresponding to partitions of $[n]$ into k parts of size s:

$$\mathcal{Y} := \{Y : \exists\ k \text{ clusters of size } s \text{ such that } Y \text{ is the corresponding incidence matrix}\}. \tag{9.1}$$

In addition, recall that the Kullback-Leibler (KL) divergence between two Bernoulli random variables with means u and v is given by

$$D(u\|v) = u \log \frac{u}{v} + (1-u) \log \frac{1-u}{1-v}. \tag{9.2}$$

We are now able to state our theorem for the lower bound on the minimax error probability of recovering Y^*.

Theorem 9.1. *If* $128 \le s \le n/2$ *and*

$$\binom{s-1}{d-1} \max\{D(p\|q), D(q\|p)\} \le \frac{1}{24} \ln(n-s), \tag{9.3}$$

then

$$\inf_f \sup_{\mathcal{Y}^* \in \mathcal{Y}} \mathbb{P}(f(T) \neq Y^*) \ge \frac{1}{2}.$$

The infimum is taken over all measurable functions $f : \mathbf{S}^d(\{0,1\}^n) \to \mathcal{Y}$, *where* $\mathbf{S}^d(\{0,1\}^n)$ *denotes the set of symmetric* d*-tensors in* $\{0,1\}^{n^d}$, *and the probability is*

taken over a random adjacency tensor T sampled from the HSBM distribution corresponding to Y^*, i.e. $T \sim H(n, \mathcal{C}(Y^*), p, q, d)$.

From this theorem we know that when p, q are constant, if $s = O\left(\log^{\frac{1}{d-1}} n\right)$, then for *any* algorithm there is some "bad" input on which the algorithm will fail with probability at least $1/2$. When $d = 2$, this is the result obtained in [15]. It is unclear at present whether the exponent $\frac{1}{d-1}$ can be improved.

Note that Theorem 9.1 is an information theoretical lower bound. On the other hand, our Theorem 2.1 considers only polynomial time solvability, and there is a considerable gap between the performance guarantee of Theorem 2.1 and the lower bound given in Theorem 9.1. Closing this gap remains an open problem.

Proof of Theorem 9.1. Let $m = n - s$ and $\overline{\mathcal{Y}} = \{Y_0, \ldots, Y_m\}$ be a subset of $\mathcal{Y}$ of size $m + 1$ defined as follows. Let Y_0 be the incidence matrix of the partition $C_1, \ldots, C_k$, where $C_l := \{(l-1)s + 1, \ldots, ls\}$. We then define Y_i for $i > 0$ by swapping the cluster membership of s and $s + i$. More formally, if $s + i \in C_l$, then Y_i is the incidence matrix of the partition $C'_1, \ldots, C'_k$, where $C'_1 := C_1 \cup \{s + i\} \setminus \{s\}$, $C'_l := C_l \cup \{s\} \setminus \{s + i\}$, and $C'_j := C_j$ for all $j \neq 1, l$.

Let $\mathcal{P}_{(Y^*,T)}$ be the joint distribution of (Y^*, T) where we first sample an incidence matrix Y^* from $\overline{\mathcal{Y}}$ uniformly at random and then sample a hypergraph adjacency tensor $T \sim H(n, \mathcal{C}(Y^*), p, q, d)$ (see Section 1.2). Then we have

$$\inf_f \sup_{Y^* \in \mathcal{Y}} \mathbb{P}(f(T) \neq Y^*) \geq \inf_f \mathbb{P}_{(Y^*,T)}(f(T) \neq Y^*) \geq 1 - \frac{I(Y^*; T) + 1}{\log |\overline{\mathcal{Y}}|}, \tag{9.4}$$

where the last inequality is by Fano's inequality and $I(Y^*; T)$ is the mutual information between Y^* and T. Let $\mathbb{P}_i$ be the probability distribution of the hypergraph H conditioned on $Y^* = Y_i$. By the convexity of KL-divergence we have

$$I(Y^*; T) \leq \frac{1}{(m+1)^2} \sum_{i,i'=0}^{m} D(\mathbb{P}_i \| \mathbb{P}_{i'}) \leq \max_{i,i'} D(\mathbb{P}_i \| \mathbb{P}_{i'}).$$

Note that $\mathbb{P}_i$ is the product of $\binom{n}{d}$ many Bernoulli distributions. Let $\mathbb{P}_i(e)$ be the probability distribution of the hyperedge e under the distribution $\mathbb{P}_i$, which is either $\mathrm{Ber}(p)$ or $\mathrm{Ber}(q)$. Then for any $i \neq i'$ we have

$$\begin{aligned} D(\mathbb{P}_i \| \mathbb{P}_{i'}) &\leq \sum_e D(\mathbb{P}_i(e) \| \mathbb{P}_{i'}(e)) \leq 3\binom{s-1}{d-1} D(p\|q) + 3\binom{s-1}{d-1} D(q\|p) \qquad (9.5) \\ &\leq 6\binom{s-1}{d-1} \max\{D(p\|q), D(q\|p)\}. \end{aligned}$$

Here the first inequality in (9.5) is due to the fact that KL-divergence is additive for products of independent distributions. The second inequality comes from counting terms

in which $\mathbb{P}_i(e) \neq \mathbb{P}_{i'}(e)$. In the worst case we have $s+i \in C_l$ and $s+i' \in C_{l'}$ for some $l' \neq l \neq 0$, in which we get a contribution of $D(p||q)$ from all e containing $s+i$ and $d-1$ indices from $C_1 \setminus \{s\}$, all e containing s and $d-1$ indices from $C_l \setminus \{s+i\}$, and all e containing $s+i'$ and $d-1$ indices from $C_{l'} \setminus \{s+i'\}$, a total of $3\binom{s-1}{d-1}$ terms; we get a contribution of $D(q||p)$ from the same number of terms.

If (9.3) holds, then $I(Y^*;T) \leq \frac{1}{4}\log(n-s) = \frac{1}{4}\log|\overline{\mathcal{Y}}|$. When $n \geq 128$, we have $\log|\overline{\mathcal{Y}}| \geq 4$. Then from (9.4) the minimax error probability is at least $1/2$. This completes the proof. □

Declaration of competing interest

No competing interest.

Acknowledgements

The authors would thank MSRI 2018 Summer School: Representations of High Dimensional Data, during which this project was initiated. The authors are grateful to anonymous referees for their detailed comments and suggestions, which have improved the quality of this paper. Y. Zhu is supported by NSF DMS-1712630.

Appendix A. Simple counting algorithm

One can recover HSBMs by simply counting the number of hyperedges containing pairs of vertices: with high probability, pairs of vertices in the same cluster will be contained in more hyperedges than pairs in different clusters. However, our spectral algorithm provides better performance guarantees than this simple counting algorithm.

Algorithm 2

Given $H=(V,E)$, $|V|=n$, number of clusters k, and cluster size $s=n/k$:

(1) For each pair of vertices $u \neq v$, compute $A_{uv} :=$ number of hyperedges containing u and v.
(2) For each vertex v, let W_v be the set of vertices containing v and the $s-1$ vertices $u \neq v$ with highest A_{uv} (breaking ties arbitrarily). It will be shown that w.h.p. W_v will be the cluster C_i containing v.

Theorem A.1. *Let H be sampled from $H(n,d,\mathcal{C},p,q)$, where $d \geq 3, \mathcal{C} = \{C_1,\ldots,C_k\}$ and $|C_i| = s = n/k$ for $i = 1,\ldots,k$. Then Algorithm 2 recovers $\mathcal{C}$ with probability $\geq 1-1/n$ if*

$$\binom{s-2}{d-2}(p-q) > \sqrt{6\binom{n-2}{d-2}\log n}.$$

A simple counting algorithm for graph SBMs was given in [15]. Our algorithm is modified from [15] for hypergraphs based on counting hyperedges and it requires $d \geq 3$.

Proof. For each $u \neq v$, $A_{uv} = \sum_{e:u,v\in e} T_e$ is the sum of $\binom{n-2}{d-2}$ independent Bernoulli random variables of expectation either p or q. Thus, it follows from a straightforward application of Hoeffding's inequality that

$$A_{uv} \geq \binom{n-2}{d-2}q + \binom{s-2}{d-2}(p-q) - \sqrt{\frac{3}{2}\binom{n-2}{d-2}\log n} \tag{A.1}$$

with probability $\geq 1 - 1/n^3$ if u and v are in the same cluster and

$$A_{uv} \leq \binom{n-2}{d-2}q + \sqrt{\frac{3}{2}\binom{n-2}{d-2}\log n} \tag{A.2}$$

with probability $\geq 1 - 1/n^3$ if u and v are in different clusters. Taking a union bound over all $\binom{n}{2}$ pairs, these bounds hold for all pairs $u \neq v$ with probability $\geq 1-1/n$. Thus, as long as the lower bound in (A.1) is greater than the upper bound in (A.2), for each v the $s-1$ vertices with highest A_{uv} will be the other vertices in v's cluster. □

In particular, if we bound the binomial coefficient $\binom{a}{b}$ by $\left(\frac{a}{b}\right)^b \leq \binom{a}{b} \leq \left(\frac{ae}{b}\right)^b$, we see that

$$s \geq c_1\sqrt{nd}\left(\frac{\sqrt{\log n}}{p-q}\right)^{\frac{1}{d-2}}$$

and

$$p - q \geq \frac{c_2(2end)^{\frac{d-2}{2}}\sqrt{\log n}}{s^{d-2}} = c_2\left(\frac{2ek^2d}{n}\right)^{\frac{d-2}{2}}\sqrt{\log n}$$

are both sufficient conditions for recovery, where c_1 and c_2 are absolute constants.

Appendix B. The sparse case

We can also analyze the performance of Algorithm 1 in the sparse case, in which we treat k, d as fixed and try to make p and q as small as possible. Our concentration bound (4.3) is not optimal in the sparse case. However, when $p = \frac{\omega(\log^4 n)}{n^{d-1}}$, we can still get a good concentration inequality of the adjacency matrix A using Lemma 5 in [44]. We include it here:

Lemma B.1. *If* $p = \frac{\omega(\log^4 n)}{n^{d-1}}$, *we have*

$$\|A - \mathbb{E}A\|_2 \leq 2d\sqrt{n^{d-1}p} \tag{B.1}$$

with probability $1 - o(1)$.

In this case, we get the following analog of Theorem (5.4).

Lemma B.2. *Assume* (B.1) *holds. Then*

$$\|P_k(A) - P_k(\mathbb{E}A)\|_2 \leq \varepsilon$$

and

$$\|P_k(A) - P_k(\mathbb{E}A)\|_{\mathrm{F}} \leq \sqrt{2k}\varepsilon$$

for any

$$\varepsilon \geq \frac{2d\sqrt{n^{d-1}p}}{\binom{s-2}{d-2}(p-q)s - 4d\sqrt{n^{d-1}p}}. \tag{B.2}$$

Proof. Apply Lemma 5.5 with $X = A$, $Y = \mathbb{E}A$, and

$$\alpha = \binom{s-2}{d-2}(p-q)(s-1) - \binom{n-2}{d-2}q - 2d\sqrt{n^{d-1}p},$$
$$\beta = -\binom{s-2}{d-2}(p-q) - \binom{n-2}{d-2}q + 2d\sqrt{n^{d-1}p}.$$

Note that we need

$$\binom{s-2}{d-2}(p-q)s > 4d\sqrt{n^{d-1}p} \tag{B.3}$$

in order for this to work. □

If we assume $p = \dfrac{\omega(\log^4 n)}{n^{d-1}}, p-q = \Theta(p)$, and k is fixed, condition (B.3) always holds. In addition, we want the failure probability to be $o(1)$, so we require

$$\exp\left(-\varepsilon^2\binom{s-1}{d-1}\right) = o((nk)^{-1}).$$

Putting $\varepsilon^2\binom{s-1}{d-1} \geq 3\log n$ suffices to accomplish this. Therefore, we require that $\varepsilon \geq \dfrac{c_4\sqrt{\log n}}{n^{(d-1)/2}}$ for some constant c_4 depending only on d, k as an additional lower bound on ε. On the other hand, to make the algorithm succeed, we need to have $\varepsilon < \dfrac{p-q}{32d}$ from the analysis in Section 8. Together we have the following constraint on ε:

$$\max\left\{\frac{c_4\sqrt{\log n}}{n^{(d-1)/2}}, \frac{2d\sqrt{n^{d-1}p}}{\binom{s-2}{d-2}(p-q)s - 4d\sqrt{n^{d-1}p}}\right\} < \varepsilon < \frac{p-q}{32d}. \tag{B.4}$$

To make (B.4) work, assuming $p - q > c_5 p$ for some constant $0 < c_5 < 1$, we have

$$p - q \geq \frac{c_6}{n^{(d-1)/3}}$$

for some constant $c_6 > 0$ depending on d, k and c_5. This yields the following corollary to Theorem 2.1:

Theorem B.3 *(Sparse case). Let k, d be constant and let H be sampled from $H(n, d, \mathcal{C}, p, q)$, where $\mathcal{C} = \{C_1, \ldots, C_k\}$ and $|C_i| = s = n/k$ for $i = 1, \ldots, k$. If $p - q > c_5 p$ for some constant $0 < c_5 < 1$ and*

$$p - q \geq \frac{c_6}{n^{(d-1)/3}} \tag{B.5}$$

for some constant c_6 depending on d, k and c_5, then Algorithm 1 recovers $\mathcal{C}$ w.h.p.

Thus, we see that our algorithm is far from optimal in the sparse case: the algorithms developed in [37,43,17,16,29] all provide better performance guarantees. In fact, even the trivial hyperedge counting algorithm (Algorithm 2) beats our spectral algorithm in the sparse case.

References

[1] Emmanuel Abbe, Community detection and stochastic block models: recent developments, J. Mach. Learn. Res. 18 (177) (2018) 1–86.

[2] Emmanuel Abbe, Afonso S. Bandeira, Georgina Hall, Exact recovery in the stochastic block model, IEEE Trans. Inf. Theory 62 (1) (2016) 471–487.

[3] Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, Yiqiao Zhong, Entrywise eigenvector analysis of random matrices with low expected rank, arXiv preprint, arXiv:1709.09565, 2017.

[4] Kwangjun Ahn, Kangwook Lee, Changho Suh, Community recovery in hypergraphs, in: Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on, IEEE, 2016, pp. 657–663.

[5] Kwangjun Ahn, Kangwook Lee, Changho Suh, Hypergraph spectral clustering in the weighted stochastic block model, IEEE J. Sel. Top. Signal Process. (2018).

[6] Dan Alistarh, Jennifer Iglesias, Milan Vojnovic, Streaming min-max hypergraph partitioning, in: Advances in Neural Information Processing Systems, 2015, pp. 1900–1908.

[7] Noga Alon, Michael Krivelevich, Benny Sudakov, Finding a large hidden clique in a random graph, Random Structures Algorithms 13 (3–4) (1998) 457–466.

[8] Brendan P.W. Ames, Guaranteed clustering and biclustering via semidefinite programming, Math. Program. 147 (1–2) (2014) 429–465.

[9] Maria Chiara Angelini, Francesco Caltagirone, Florent Krzakala, Lenka Zdeborová, Spectral detection on sparse hypergraphs, in: Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on, IEEE, 2015, pp. 66–73.

[10] Sanjeev Arora, Boaz Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009.

[11] Charles Bordenave, Marc Lelarge, Laurent Massoulié, Nonbacktracking spectrum of random graphs: community detection and nonregular Ramanujan graphs, Ann. Probab. 46 (1) (2018) 1–71.

[12] Alain Bretto, Luc Gillibert, Hypergraph-based image representation, in: International Workshop on Graph-Based Representations in Pattern Recognition, Springer, 2005, pp. 1–11.
[13] Samuel R. Bulò, Marcello Pelillo, A game-theoretic approach to hypergraph clustering, in: Advances in Neural Information Processing Systems, 2009, pp. 1571–1579.
[14] Yudong Chen, Sujay Sanghavi, Huan Xu, Improved graph clustering, IEEE Trans. Inf. Theory 60 (10) (2014) 6440–6455.
[15] Yudong Chen, Jiaming Xu, Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices, J. Mach. Learn. Res. 17 (1) (2016) 882–938.
[16] I. Chien, Chung-Yi Lin, I-Hsiang Wang, Community detection in hypergraphs: optimal statistical limit and efficient algorithms, in: International Conference on Artificial Intelligence and Statistics, 2018, pp. 871–879.
[17] I. Chien, Chung-Yi Lin, I-Hsiang Wang, On the minimax misclassification ratio of hypergraph community detection, arXiv preprint, arXiv:1802.00926, 2018.
[18] Sam Cole, Recovering nonuniform planted partitions via iterated projection, Linear Algebra Appl. 576 (2019) 79–107.
[19] Sam Cole, Shmuel Friedland, Lev Reyzin, A simple spectral algorithm for recovering planted partitions, Spec. Matrices 5 (1) (2017) 139–157.
[20] Aurelien Decelle, Florent Krzakala, Cristopher Moore, Lenka Zdeborová, Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications, Phys. Rev. E 84 (6) (2011) 066106.
[21] Aurélien Ducournau, Alain Bretto, Soufiane Rital, Bernard Laget, A reductive approach to hypergraph clustering: an application to image segmentation, Pattern Recognit. 45 (7) (2012) 2788–2803.
[22] Uriel Feige, Robert Krauthgamer, Finding and certifying a large hidden clique in a semirandom graph, Random Structures Algorithms 16 (2000) 195–208.
[23] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S. Vempala, Ying Xiao, Statistical algorithms and a lower bound for detecting planted cliques, J. ACM 64 (2) (April 2017) 8.
[24] Keqin Feng, Wen-Ching Winnie Li, Spectra of hypergraphs and applications, J. Number Theory 60 (1) (1996) 1–22.
[25] Suzanne Renick Gallagher, Debra S. Goldberg, Clustering coefficients in protein interaction hypernetworks, in: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics, ACM, 2013, p. 552.
[26] Debarghya Ghoshdastidar, Ambedkar Dukkipati, Consistency of spectral partitioning of uniform hypergraphs under planted partition model, in: Advances in Neural Information Processing Systems, 2014, pp. 397–405.
[27] Debarghya Ghoshdastidar, Ambedkar Dukkipati, A provable generalized tensor spectral method for uniform hypergraph partitioning, in: International Conference on Machine Learning, 2015, pp. 400–409.
[28] Debarghya Ghoshdastidar, Ambedkar Dukkipati, Spectral clustering using multilinear SVD: analysis, approximations and applications, in: AAAI, 2015, pp. 2610–2616.
[29] Debarghya Ghoshdastidar, Ambedkar Dukkipati, Consistency of spectral hypergraph partitioning under planted partition model, Ann. Statist. 45 (1) (2017) 289–315.
[30] Gene H. Golub, Charles F. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, MD, USA, 1996.
[31] Ming Gu, Subspace iteration randomization and singular value problems, SIAM J. Sci. Comput. 37 (3) (2015) A1139–A1173.
[32] Eui-Hong Han, George Karypis, Vipin Kumar, Bamshad Mobasher, Hypergraph based clustering in high-dimensional data sets: a summary of results, IEEE Data Eng. Bull. 21 (1) (1998) 15–22.
[33] Matthias Hein, Simon Setzer, Leonardo Jost, Syama Sundar Rangapuram, The total variation on hypergraphs-learning on hypergraphs revisited, in: Advances in Neural Information Processing Systems, 2013, pp. 2427–2435.
[34] Amin Jalali, Qiyang Han, Ioana Dumitriu, Maryam Fazel, Relative density and exact recovery in heterogeneous stochastic block models, arXiv preprint, arXiv:1512.04937, 2015.
[35] Amin Jalali, Qiyang Han, Ioana Dumitriu, Maryam Fazel, Exploiting tradeoffs for exact recovery in heterogeneous stochastic block models, in: Advances in Neural Information Processing Systems, 2016, pp. 4871–4879.
[36] Mark Jerrum, Large cliques elude the Metropolis process, Random Structures Algorithms 3 (4) (1992) 347–359.

[37] Chiheon Kim, Afonso S. Bandeira, Michel X. Goemans, Stochastic block model for hypergraphs: statistical limits and a semidefinite programming approach, arXiv preprint, arXiv:1807.02884, 2018.
[38] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, Chang D. Yoo, Higher-order correlation clustering for image segmentation, in: Advances in Neural Information Processing Systems, 2011, pp. 1530–1538.
[39] Tamara G. Kolda, Brett W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.
[40] Luděk Kučera, Expected complexity of graph partitioning problems, Discrete Appl. Math. 57 (2–3) (1995) 193–212.
[41] Marius Leordeanu, Cristian Sminchisescu, Efficient hypergraph clustering, in: Artificial Intelligence and Statistics, 2012, pp. 676–684.
[42] Leonid A. Levin, Average case complete problems, SIAM J. Comput. 15 (1) (feb 1986) 285–286.
[43] Chung-Yi Lin, I. Eli Chien, I-Hsiang Wang, On the fundamental statistical limit of community detection in random hypergraphs, in: Information Theory (ISIT), 2017 IEEE International Symposium on, IEEE, 2017, pp. 2178–2182.
[44] Linyuan Lu, Xing Peng, Loose Laplacian spectra of random hypergraphs, Random Structures Algorithms 41 (4) (2012) 521–545.
[45] Laurent Massoulié, Community detection thresholds and the weak Ramanujan property, in: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, ACM, 2014, pp. 694–703.
[46] Elchanan Mossel, Joe Neeman, Allan Sly, Reconstruction and estimation in the planted partition model, Probab. Theory Related Fields 162 (3–4) (2015) 431–461.
[47] Elchanan Mossel, Joe Neeman, Allan Sly, A proof of the block model threshold conjecture, Combinatorica 38 (3) (2018) 665–708.
[48] Elchanan Mossel, Joe Neeman, Allan Sly, et al., Consistency thresholds for the planted bisection model, Electron. J. Probab. 21 (2016).
[49] Samet Oymak, Babak Hassibi, Finding dense clusters via low rank+ sparse decomposition, arXiv preprint, arXiv:1104.5186, 2011.
[50] Soumik Pal, Yizhe Zhu, Community detection in the sparse hypergraph stochastic block model, arXiv preprint, arXiv:1904.05981, 2019.
[51] David A. Papa, Igor L. Markov, Hypergraph partitioning and clustering, in: Handbook of Approximation Algorithms and Metaheuristics, Chapman and Hall/CRC, 2007, pp. 959–978.
[52] Alexei Vazquez, Finding hypergraph communities: a Bayesian approach and variational solution, J. Stat. Mech. Theory Exp. 2009 (07) (2009) P07006.
[53] Roman Vershynin, High-Dimensional Probability: An Introduction with Applications in Data Science, vol. 47, Cambridge University Press, 2018.
[54] Anderson Y. Zhang, Harrison H. Zhou, Minimax rates of community detection in stochastic block models, Ann. Statist. 44 (5) (2016) 2252–2280.
[55] Denny Zhou, Jiayuan Huang, Bernhard Schölkopf, Learning with hypergraphs: clustering, classification, and embedding, in: Advances in Neural Information Processing Systems, 2007, pp. 1601–1608.