EXACT RECOVERY IN THE HYPERGRAPH STOCHASTIC BLOCK MODEL: A SPECTRAL ALGORITHM

SAM COLE AND YIZHE ZHU

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 we define to be the 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 $k = \Theta(\sqrt{n})$.

1. Introduction

1.1. **Hypergraph clustering.** Clustering is an important topic in data mining, network analysis, machine learning and computer vision [JMF99]. 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 [ZHS07, PM07]. 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 [ZHS07, LS12, Vaz09, GD15b, BP09, HSJR13, AIV15]. Hypergraph clustering has found a wide range of applications [HKKM98, DBRL12, BG05, GG13, KNKY11].

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 [GD14] 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.

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 $H(n, d, \mathcal{C}, p, q)$.

Date: November 19, 2018.

Hypergraphs are closely related to symmetric tensors. We give a definition of symmetric tensors below, see [KB09] 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 is symmetric if $T_{i_1,i_2,\dots i_d} = T_{\sigma(i_1),\sigma(i_2)\dots,\sigma(i_d)}$ for any $i_1,\dots,i_d \in [n]$ 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,

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

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

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

We set $T_{i_1,\dots i_d} = 0$ if any two of the indices in $\{i_1,\dots i_d\}$ coincide, and we set $T_{\sigma(i_1),\sigma(i_2)\dots,\sigma(i_d)} = T_{i_1,\dots i_d}$ for any permutation σ. Hence, T is symmetric, and $T_{i_1,\dots,i_d}$ is determined entirely by the index set $\{i_1,\dots,i_d\}$, regardless of order. If we have two distinct index sets $\{i_1,\dots,i_d\} \neq \{j_1,\dots,j_d\}$, then the random variables $T_{i_1,\dots,i_d}$ and $T_{j_1,\dots,j_d}$ are independent. Furthermore, we may abuse notation and write T_e in place of $T_{i_1,\dots,i_d}$, where $e = \{i_1,\dots,i_d\}$.

The HSBM recovery problem is to find the ground truth clusters $\mathcal{C} = \{C_1,\dots,C_k\}$ 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 [Abb18] for further details:

(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 $\hat{\mathcal{C}}$ such that the portion of mislabelled vertices is $1/2-\varepsilon$ for some positive ε.

For exact recovery with two blocks, it was shown that the phase transition occurs in the regime of logarithmic average degree in [LCW17, CLW18b, CLW18a] by analyzing the minimax risk, and the exact threshold was given in [KBG18], by a generalization of the techniques in [ABH16]. An exact recovery threshold of the censored block model for hypergraphs was characterized in [ALS16]. For detection, [ACKZ15] proposed a belief propagation algorithm and conjectured a threshold point.

Several methods have been considered for exact recovery of HSBMs. In [GD14], 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, for exact, almost exact and partial recovery [GD15a, GD15b, GD17]. Spectral methods along with local refinements were considered in [CLW18a, ALS16]. A semidefinite programing approach was analyzed in [KBG18].

1.3. **This paper.** In this paper, we focus on exact recovery. Rather than dealing with sparsity as in [KBG18], we approach the problem from a different direction: we attempt to construct algorithms that succeed on dense hypergraphs (with $0 \le q < p \le 1$ constant) when the number of blocks k increases with n. Our algorithm works when $k = \Omega(\sqrt{n})$, which is believed to be the barrier for exact recovery in the dense graph case with clusters of equal size [CSX14, Ame14, OH11]. To the best of our knowledge, our algorithm is the first to guarantee exact recovery when the number of clusters $k = \Theta(\sqrt{n})$. In addition, in contrast to [KBG18, CLW18a, ALS18, GD17], our algorithm is purely spectral. While we focus on the dense case, our algorithm can be adapted to the sparse case as well; however, it does not perform as well as previously known algorithms [KBG18, LCW17, CLW18b, CLW18a, GD17] in the sparse 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 3.1 below for precise statement.

Our algorithm compares favorably with other known algorithms for HSBM recovery in the dense case with $k = \omega(1)$ clusters; see Section 11. It is based on the *iterated projection* technique developed in [CFR17, Col18] for the graph case. We apply this approach to the *adjacency matrix* of the random hypergraph H—the $n \times n$ symmetric matrix A whose (u, v)-th entry is the number of hyperedges containing both u and v (or 0 if $u = v$). In the process, we prove a concentration result for the spectral norm of A, which may be of independent interest (Theorem 5.4).

2. Simple counting algorithm

Before we introduce our spectral algorithm for recovering HSBMs, let us observe that 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, we will see that our spectral algorithm provides better performance guarantees than this simple counting algorithm.

Algorithm 1

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 2.1. *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 1 recovers $\mathcal{C}$ with probability $\geq 1 - 1/n$ if*

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

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

$$A_{uv} \geq \binom{n-2}{d-2}q + \binom{s-2}{d-2}(p-q) - \sqrt{\frac{3}{2}\binom{n-2}{d-2}\log n} \tag{2.1}$$

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

$$A_{uv} \leq \binom{n-2}{d-2}q + \sqrt{\frac{3}{2}\binom{n-2}{d-2}\log n} \tag{2.2}$$

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 (2.1) is greater than the upper bound in (2.2), for each v the $s-1$ vertices with highest A_{uv} will be the other vertices in v's cluster. $\square$

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.

3. Spectral algorithm and main results

Our main result is that Algorithm 2 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 [CFR17, Col18]. 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 2

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 4).
(2) Let $P_k(A)=(P_{uv})_{u,v\in V}$ be the dominant rank-k projector of A (as defined in Section 6).
(3) For each column v of $P_k(\hat{A})$, 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 $P_k(\hat{A})$, 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 7).
(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 8).
(7) Delete C from H and repeat on the remaining sub-hypergraph. Stop when there are $< s$ vertices left.

The remainder of this paper is devoted to proving correctness of this algorithm.

Theorem 3.1. *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*

$$\frac{6d\sqrt{d\binom{n}{d-1}}}{\binom{s-2}{d-2}(p-q)s-12d\sqrt{d\binom{n}{d-1}}} \leq \varepsilon < \frac{p-q}{32d}, \tag{3.1}$$

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

Sections 4-6 introduce the linear algebra tools necessary for the proof; Section 7 shows that Step 4 with high probability produces a a set with small symmetric difference with one of the clusters; Section 8 proves that Step 6 with high probability recovers one of the clusters exactly; and Section 9 proves inductively that the algorithm with high probability recovers all clusters.

3.1. **Performance guarantees.** Observe that if the numerator is less than the denominator in (3.1), then we can upper bound the left hand side by

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

Thus, Theorem 3.1 guarantees that we can recover $\mathcal{C}$ w.h.p. if

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

(this is a slightly stronger condition than (3.1)). Recall that for nonnegative integers $a \geq b$ we can bound the binomial coefficient $\binom{a}{b}$ by $\left(\frac{a}{b}\right)^b \leq \binom{a}{b} \leq \left(\frac{ae}{b}\right)^b$. Using these bounds and solving for s, we get the following as a corollary to Theorem 3.1:

Theorem 3.2 (Dense case). *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*

$$s \geq \frac{c_3\sqrt{nd}}{(p-q)^{\frac{2}{d-1}}}$$

then Algorithm 2 recovers $\mathcal{C}$ w.h.p., where c_3 is an absolute constant.

Note that Theorem 3.1 requires p and q to be constant, but d is allowed to vary with n. However, we want the failure probability to be $o(1)$, so we require $\exp\left(\varepsilon^2\binom{s-1}{d-1}\right) = o((nk)^{-1})$. In particular, this is satisfied if d is constant.

Thus, we see that Algorithm 2 beats Algorithm 1 by a factor of $(\log n)^{\frac{1}{2d-4}}$. However, observe that the guarantee for Algorithm 1 approaches that of Algorithm 2 as d increases. See Section 11 for comparison with other known algorithms.

3.2. **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 [GVL96, Gu15]), for $d \geq 3$ the running time of Algorithm 2 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 2 is $O(kn^d)$.

4. Reduction to random matrices

Working with random tensors is hard [HL13], since we do not have as many linear algebra and probability tools as for random matrices. 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 [FL96].

Definition 4.1 (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 $e=\{i_1,\dots,i_d\}$, let T_e be the entry in T corresponding to $T_{i_1,\dots,i_d}$. We define the adjacency matrix A of H by

$$A_{ij} := \sum_{e:\{i,j\}\in e} T_e. \tag{4.1}$$

Thus, A_{ij} is the number of hyperedges in H that contains vertices i,j. Note that in the summation (4.1), 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.

5. Eigenvalues and concentration of spectral norms

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

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

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. We have the following eigenvalues for $\tilde{A}$. 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 5.1. *The eigenvalues of $\tilde{A}$ are*

$$\begin{aligned} \lambda_1(\tilde{A}) &= \binom{s-2}{d-2}(p-q)s+\binom{n-2}{d-2}qn,\\ \lambda_i(\tilde{A}) &= \binom{s-2}{d-2}(p-q)s, \quad 2\le i\le k,\\ \lambda_i(\tilde{A}) &= 0, \quad k+1\le i\le n. \end{aligned}$$

Hence by a shifting, we have the following eigenvalues for $\mathbb{E}A$.

Lemma 5.2. *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 the following concentration inequality for the spectral norm of $A-\mathbb{E}A$.

Definition 5.3 (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 $d(x,y)\le\varepsilon$.

Theorem 5.4. *Let $\|\cdot\|_2$ be the spectral norm of a matrix, we have*

$$\|A - \mathbb{E}A\|_2 \le 6d\sqrt{d\binom{n}{d-1}} \tag{5.1}$$

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

Proof. 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 $\|\mathbf{x}-\mathbf{y}\|_2 \le \varepsilon$. Then we have

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

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

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

Therefore

$$\|M\|_2 \le \frac{1}{1-\varepsilon}\sup_{\mathbf{x}\in\mathcal{N}} \|M\mathbf{x}\|_2 = \frac{1}{1-\varepsilon}\sup_{\mathbf{x}\in\mathcal{N}}\langle M\mathbf{x}, \mathbf{x}\rangle \tag{5.2}$$

Now we fix an $\mathbf{x} = (x_1, \dots, x_n)^\top \in \mathbb{S}^{n-1}$ 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 $|M_e| \le 1$, so we have

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

By Hoeffding's inequality,

$$\mathbb{P}(|\sum_{e\in E} Y_e| \ge t) \le 2\exp\left(-\frac{2t^2}{4\sum_{e\in E}|\sum_{ij\in e, i<j} x_ix_j|^2}\right). \tag{5.3}$$

From Cauchy's inequality, we have

$$\begin{aligned}
\sum_{e\in E}|\sum_{ij\in e, i<j} x_ix_j|^2 &\le \binom{d}{2}\sum_{e\in E}\sum_{ij\in e, i<j} x_i^2x_j^2 \\
&\le \binom{d}{2}\binom{n-2}{d-2}\sum_{1\le i<j\le n} x_i^2x_j^2 \\
&\le \binom{d}{2}\binom{n-2}{d-2}\frac{1}{2}\left(\sum_i x_i^2\right)^2 \le \frac{1}{4}d^2\binom{n}{d-2}. \qquad (5.4)
\end{aligned}$$

Therefore from (5.3) and (5.4),

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

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

$$\mathbb{P}\left(|\langle M\mathbf{x},\mathbf{x}\rangle| \geq 3d\sqrt{d}\sqrt{\binom{n}{d-1}}\right) \leq 2\exp\left(-\frac{9d\binom{n}{d-1}}{2\binom{n}{d-2}}\right) \leq \exp(-3n). \tag{5.5}$$

Since $|\mathcal{N}| \leq (\frac{2}{\varepsilon}+1)^n$ (see Corollary 4.2.11 in [Ver18] for example), we can take $\varepsilon = 1/2$ and by a union bound, we have

$$\mathbb{P}\left(\sup_{\mathbf{x}\in\mathcal{N}} |\langle M\mathbf{x},\mathbf{x}\rangle| \geq 3d\sqrt{d}\sqrt{\binom{n}{d-1}}\right) \leq 5^n\exp(-3n) \leq e^{-n}. \tag{5.6}$$

So we have from (5.2), (5.6)

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

□

6. Dominant eigenspaces and projectors

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

Definition 6.1 (Dominant eigenspace)**.** If X is a $n\times n$ Hermitian 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 6.2 (Dominant rank-r projector)**.** If X is a $n\times n$ Hermitian 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 $\mathbf{v}_1,\dots,\mathbf{v}_r$ is an orthonormal basis for $\mathbf{E}_r(X)$, then

$$P_r(X) = \sum_{i=1}^{r}\mathbf{v}_i\mathbf{v}_i^*, \tag{6.1}$$

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.,

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

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

Theorem 6.3. *Let $A, \mathbb{E}A$, and $\tilde{A}$ be defined as in Sections 4 and 5. 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

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

for some constants $a > b > 0$. Thus, $\left\{\frac{1}{\sqrt{s}}\mathbf{1}_{C_i} : i = 1, \dots, k\right\}$ is an orthonormal basis for the column space of both Y and $\tilde{A}$, and hence, in accordance with (6.1),

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

Now, observe that the eigenvalues of $\mathbb{E}A$ are those of $\tilde{A}$ shifted down by $a-b$, and $\mathbf{v}$ is an eigenvector of $\mathbb{E}A$ iff. 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 $P_k(\mathbb{E}A) = P_k(\tilde{A})$. $\square$

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 6.4. *Assume* (5.1) *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 6.4, we use the following Lemma from [Col18, Lemma 4].

Lemma 6.5. *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 $n-r$ eigenvalues of both X, Y are at most α, where $\alpha < \beta$. Then*

$$\|P_k(X) - P_k(Y)\|_2 \leq \frac{\|X-Y\|_2}{\beta - \alpha}, \tag{6.2}$$

$$\|P_k(X) - P_k(Y)\|_F \leq \frac{\sqrt{2k}\|X-Y\|_2}{\beta - \alpha}. \tag{6.3}$$

Proof of Theorem 6.4. Apply Lemma 6.5 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.

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

$\square$

7. 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 7.1)
- Such a set W exists among the sets $W_1, \dots, W_n$, where W_v is the indices of the $s-1$ largest entries in column v of $P_k(A)$, along with v itself (Lemma 7.2).

The intuition is that if $||P_k(A) - P_k(\mathbb{E}A)||_2 \le \varepsilon$, 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 7.1 and 7.2 below are essentially the same as Lemmas 18 and 17 in [CFR17]. As $P_k(A) = \frac{1}{s}\sum_i \mathbf{1}_{C_i}\mathbf{1}_{C_i}^\top$ as in the graph case (Theorem 6.3), we can import their proofs directly from the graph case. However, we present a simpler proof for Lemma 7.1.

Lemma 7.1. *Assume (5.1) holds. 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 6.4,

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

And by the triangle inequality,

$$\|P_k(\mathbb{E}A)\mathbf{1}_W\|_2 \ge \|P_k(A)\mathbf{1}_W\|_2 - \varepsilon\sqrt{s} \ge (1-2\varepsilon)\sqrt{s} - \varepsilon\sqrt{s} = (1-3\varepsilon)\sqrt{s}. \tag{7.1}$$

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 6.3

$$||P_k(\mathbb{E}A)\mathbf{1}_W||_2^2 = \frac{1}{s}\sum_{i=1}^{k} |W \cap C_i|^2. \tag{7.2}$$

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

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

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 (7.1) and (7.2) 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 $\varepsilon \leq 1/12$. $\square$

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 2), then Lemma 7.2 below will show that one of these sets satisfies the conditions of Lemma 7.1; 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 7.2. *Assume (5.1) holds. For $v = 1, \ldots, n$, let W_v be defined as in Step 3 of Algorithm 2. Then there exists a column v such that*

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

The proof is exactly the same as that of [CFR17, Lemma 17].

Lemmas 7.1 and 7.2 together prove that, as long as (5.1) 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.

8. 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 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 [CFR17, 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 set W is to define the random variable

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

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. We get the following analogue to [CFR17, Lemma 19]:

Lemma 8.1. *Consider cluster C_i and vertex $u \in [n]$, and let $\varepsilon > 0$. If $u \in C_i$, then for n sufficiently large and $d = o(s)$,*

$$N_{u,C_i} \geq (p-\varepsilon)\binom{s}{d-1} \tag{8.1}$$

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

$$N_{u,C_i} \leq (q+\varepsilon)\binom{s}{d-1} \tag{8.2}$$

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

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

$\square$

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 8.2. *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 (8.1), then $N_{j,W} \geq (p-16d\varepsilon)\binom{s}{d-1}$,

(2) If $j \notin C_i$ satisfies (8.2), then $N_{j,W} \leq (q+16d\varepsilon)\binom{s}{d-1}$.

Proof. Assume $j \in C_i$ and j satisfies (8.1). 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 vertices 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

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

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 vertices from $W \setminus C_i$. Recall $|W \setminus C_i| \leq 6\varepsilon s$.

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

□

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

9. Proof of algorithm's correctness

We now have all the necessary pieces to prove the correctness of Algorithm 2 (Theorem 3.1). The proof is roughly the same as that of [CFR17, Theorem 4].

9.1. Proof of correctness of first iteration.

Lemmas 7.1-8.2 above prove that Steps 1-6 of Algorithm 2 correctly recover a single cluster in the first iteration.

Theorem 9.1. *Assume that (5.1) holds and that for $i = 1, \ldots, k$, (8.1) holds for all $u \in C_i$ and (8.2) holds for all $u \notin C_i$. Then Steps 1-6 of Algorithm 2 exactly recover a cluster C_i in the first iteration if*

$$\varepsilon < \frac{p-q}{32d}.$$

Proof. By Lemma 7.2, the set W constructed in Step 4 has $||P_k(A)\mathbf{1}_W||_2 \geq (1-2\varepsilon)\sqrt{s}$. By Lemma 7.1, $|W \cap C_i| \geq (1-6\varepsilon)s$ for some i. And by Lemma 8.2, $N_{u,W} \geq (1-16\varepsilon)s$ 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. □

9.2. **The "delete and repeat" step.** The difficulty with proving the success of Algorithm 2 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 9.1: if certain conditions are true, then the first iteration of Algorithm 2 will definitely recover a cluster. In fact, the only probabilistic statements thus far are in Theorem 5.4 and Lemma 8.1. Mirroring the analysis in [CFR17, Col18], we will show that if certain (exponentially many) conditions are met, then *all* iterations of Algorithm 2 will succeed. We will then show that all of these events occur simultaneously w.h.p.; hence, Algorithm 2 recovers all clusters w.h.p.

We begin by introducing some terminology:

Definition 9.2 (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 $m = s|J|$ 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 2 will definitely succeed in recovering all clusters. Again, there is nothing probabilistic in this theorem or its proof.

Lemma 9.3. *Assume that E_J holds for all $J \subseteq [k]$ and $D_{i,u}$ holds for all $i \in [k], u \in [n]$. Then Algorithm 2 recovers $C_1, \ldots, C_k$ exactly if*

$$\varepsilon < \frac{p-q}{32d}. \tag{9.1}$$

Proof sketch. We omit the full proof as it is analogous to the proof in [CFR17, 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 6.4 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 the this bound is largest when $m = n$; thus, the condition (9.1) is sufficient for all iterations. □

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

Lemma 9.4. *E_J and $D_{i,u}$ hold simultaneously for all $J \subseteq [k], i \in [k], u \in [n]$ with probability $\geq 1 - 2^k \cdot \exp(-s) - nk \cdot \exp\left(-\varepsilon^2 \binom{s-1}{d-1}\right)$.*

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 5.4. 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 8.1. The proof is completed by taking a union bound over all $J \subseteq [k], i \in [k], u \in [n]$. □

Theorem 3.1 follows as an immediate corollary to Lemmas 9.3 and 9.4.

10. The sparse case

We can also analyze the performance of Algorithm 2 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 (5.4) is not optimal in the sparse case. However, when $p = \dfrac{\omega(\log^4 n)}{n^{d-1}}$, we can still get a good concentration inequality of the adjacency matrix A using Lemma 5 in [LP12]. We include it here:

Lemma 10.1. *If $p = \dfrac{\omega(\log^4 n)}{n^{d-1}}$, we have*

$$\|A - \mathbb{E}A\|_2 \leq 2d\sqrt{n^{d-1}p} \tag{10.1}$$

with probability $1 - o(1)$.

In this case, we get the following analog of Theorem (6.4).

Lemma 10.2. *Assume (10.1) 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

$$\varepsilon \geq \frac{2d\sqrt{n^{d-1}p}}{\binom{s-2}{d-2}(p-q)s - 4d\sqrt{n^{d-1}p}}. \tag{10.2}$$

Proof. Apply Lemma 6.5 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 in order for this to work we need

$$\binom{s-2}{d-2}(p-q)s > 4d\sqrt{n^{d-1}p}. \tag{10.3}$$

□

If we assume $p = \frac{\omega(\log^4 n)}{n^{d-1}}, p - q = \Theta(p)$, and k is fixed, condition (10.3) 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 $\varepsilon \geq \frac{c_4\sqrt{\log n}}{n^{(d-1)/2}}$ 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 $\varepsilon < \frac{p-q}{32d}$ from the analysis in Section 9. Together we have the following constraint on ε:

$$\max\left\{\frac{c_4\sqrt{\log n}}{n^{(d-1)/2}}, \frac{2d\sqrt{n^{d-1}p}}{\binom{s-2}{d-2}(p-q)s - 4d\sqrt{n^{d-1}p}}\right\} < \varepsilon < \frac{p-q}{32d} \tag{10.4}$$

To make (10.4) 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 3.1:

Theorem 10.3 (Sparse case)**.** *Let k, d be constant and let H be sampled from $H(n, d, \mathcal{C}, p, q)$, where $\mathcal{C} = \{C_1, \dots, C_k\}$ and $|C_i| = s = n/k$ for $i = 1, \dots, k$. If $p - q > c_5 p$ for some constant $0 < c_5 < 1$ and*

$$p - q \geq \frac{c_6}{n^{(d-1)/3}} \tag{10.5}$$

for some constant c_6 depending on d, k and c_5, then Algorithm 2 recovers $\mathcal{C}$ w.h.p.

Thus, we see that our algorithm is far from optimal in the sparse case: the algorithms developed in [KBG18, LCW17, CLW18b, CLW18a, GD17] all provide better performance guarantees. In fact, even the trivial hyperedge counting algorithm (Algorithm 1) beats our spectral algorithm in the sparse case.

11. Comparison with previous results

We compare Algorithms 1 and 2 with previous exact recovery algorithms, with p, q, d being constant. In [ALS16, CLW18a] 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
[KBG18]	$O(1)$	Semidefinite programming
[ALS18]	$O(1)$	Spectral + local refinement
[CLW18a]	$O(1)$	Spectral + local refinement
[GD17], Corollary 5.1	$o(\log^{\frac{-1}{2d}}(n) n^{\frac{d-4}{2d}})$	Spectral + k-means
Our result (Algorithm 1)	$O(\log^{\frac{-1}{2d-4}}(n) n^{0.5})$	Simple counting
Our result (Algorithm 2)	$O(n^{0.5})$	Spectral

Acknowledgement. The authors would thank MSRI 2018 Summer School: Representations of High Dimensional Data, during which this project was initiated. S.C. is supported by a PIMS Postdoctoral Fellowship. Y.Z. is supported by NSF DMS-1712630.

References

[Abb18] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[ABH16] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2016.

[ACKZ15] Maria Chiara Angelini, Francesco Caltagirone, Florent Krzakala, and Lenka Zdeborová. Spectral detection on sparse hypergraphs. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 66–73. IEEE, 2015.

[AIV15] Dan Alistarh, Jennifer Iglesias, and Milan Vojnovic. Streaming min-max hypergraph partitioning. In *Advances in Neural Information Processing Systems*, pages 1900–1908, 2015.

[ALS16] Kwangjun Ahn, Kangwook Lee, and Changho Suh. Community recovery in hypergraphs. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pages 657–663. IEEE, 2016.

[ALS18] Kwangjun Ahn, Kangwook Lee, and Changho Suh. Hypergraph spectral clustering in the weighted stochastic block model. *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[Ame14] Brendan PW Ames. Guaranteed clustering and biclustering via semidefinite programming. *Mathematical Programming*, 147(1-2):429–465, 2014.

[BG05] Alain Bretto and Luc Gillibert. Hypergraph-based image representation. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 1–11. Springer, 2005.

[BP09] Samuel R Bulò and Marcello Pelillo. A game-theoretic approach to hypergraph clustering. In *Advances in neural information processing systems*, pages 1571–1579, 2009.

[CFR17] Sam Cole, Shmuel Friedland, and Lev Reyzin. A simple spectral algorithm for recovering planted partitions. *Special Matrices*, 5(1):139–157, 2017.

[CLW18a] I Chien, Chung-Yi Lin, and I-Hsiang Wang. Community detection in hypergraphs: Optimal statistical limit and efficient algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 871–879, 2018.

[CLW18b] I Chien, Chung-Yi Lin, and I-Hsiang Wang. On the minimax misclassification ratio of hypergraph community detection. *arXiv preprint arXiv:1802.00926*, 2018.

[Col18] Sam Cole. Recovering nonuniform planted partitions via iterated projection. *Linear Algebra and its Applications*, 2018.

[CSX14] Yudong Chen, Sujay Sanghavi, and Huan Xu. Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455, 2014.

[DBRL12] Aurélien Ducournau, Alain Bretto, Soufiane Rital, and Bernard Laget. A reductive approach to hypergraph clustering: An application to image segmentation. *Pattern Recognition*, 45(7):2788–2803, 2012.

[FL96] Keqin Feng and Wen-Ching Winnie Li. Spectra of hypergraphs and applications. *Journal of number theory*, 60(1):1–22, 1996.

[GD14] Debarghya Ghoshdastidar and Ambedkar Dukkipati. Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Advances in Neural Information Processing Systems*, pages 397–405, 2014.

[GD15a] Debarghya Ghoshdastidar and Ambedkar Dukkipati. A provable generalized tensor spectral method for uniform hypergraph partitioning. In *International Conference on Machine Learning*, pages 400–409, 2015.

[GD15b] Debarghya Ghoshdastidar and Ambedkar Dukkipati. Spectral clustering using multilinear svd: Analysis, approximations and applications. In *AAAI*, pages 2610–2616, 2015.

[GD17] Debarghya Ghoshdastidar and Ambedkar Dukkipati. Consistency of spectral hypergraph partitioning under planted partition model. *The Annals of Statistics*, 45(1):289–315, 2017.

[GG13] Suzanne Renick Gallagher and Debra S Goldberg. Clustering coefficients in protein interaction hypernetworks. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, page 552. ACM, 2013.

[Gu15] Ming Gu. Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):A1139–A1173, 2015.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[HKKM98] Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. Hypergraph based clustering in high-dimensional data sets: A summary of results. *IEEE Data Eng. Bull.*, 21(1):15–22, 1998.

[HL13] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[HSJR13] Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram. The total variation on hypergraphs-learning on hypergraphs revisited. In *Advances in Neural Information Processing Systems*, pages 2427–2435, 2013.

[JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[KB09] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[KBG18] Chiheon Kim, Afonso S Bandeira, and Michel X Goemans. Stochastic block model for hypergraphs: Statistical limits and a semidefinite programming approach. *arXiv preprint arXiv:1807.02884*, 2018.

[KNKY11] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang D Yoo. Higher-order correlation clustering for image segmentation. In *Advances in neural information processing systems*, pages 1530–1538, 2011.

[LCW17] Chung-Yi Lin, I Eli Chien, and I-Hsiang Wang. On the fundamental statistical limit of community detection in random hypergraphs. In *Information Theory (ISIT), 2017 IEEE International Symposium on*, pages 2178–2182. IEEE, 2017.

[LP12] Linyuan Lu and Xing Peng. Loose laplacian spectra of random hypergraphs. *Random Structures & Algorithms*, 41(4):521–545, 2012.

[LS12] Marius Leordeanu and Cristian Sminchisescu. Efficient hypergraph clustering. In *Artificial Intelligence and Statistics*, pages 676–684, 2012.

[OH11] Samet Oymak and Babak Hassibi. Finding dense clusters via low rank+ sparse decomposition. *arXiv preprint arXiv:1104.5186*, 2011.

[PM07] David A Papa and Igor L Markov. Hypergraph partitioning and clustering. In *Handbook of Approximation Algorithms and Metaheuristics*, pages 959–978. Chapman and Hall/CRC, 2007.

[Vaz09] Alexei Vazquez. Finding hypergraph communities: a bayesian approach and variational solution. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07006, 2009.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

[ZHS07] Denny Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*, pages 1601–1608, 2007.

Department of Mathematics, University of Manitoba, Winnipeg, MB R3T 2MB, Canada
E-mail address: samuel.cole@umanitoba.ca

Department of Mathematics, University of Washington, Seattle, Washington 98105, USA
E-mail address: yizhezhu@uw.edu