Spectral Sparsification of Hypergraph Laplacian-type Operators

Shiv Kampani*

Spring 2024

Abstract

This is a research project for Prof. Alexandr Andoni. I focus on the problem of constructing spectral sparsifiers for hypergraphs by edge-sampling based methods. Most of the existing literature focuses on the non-linear hypergraph Laplacian defined by [CLTZ16]. In addition to analyzing sparsifiers in literature, I make progress in a new research direction: constructing a $O(nr^3\varepsilon^{-2}\log n)$-hyperedge sparsifier for the new Laplacian-type operator on hypergraphs recently introduced by [MS21], with applications in efficient algorithms for identifying bipartite components in hypergraphs.

1 Introduction

We will focus on the problem of constructing spectral sparsifiers for hypergraphs. Given an undirected graph G, a sparsifier, denoted $\tilde{G}$, is a graph that captures "important information" about G with significantly fewer edges. Benczur and Karger studied sparsifiers that preserve the cut-structure of G and constructed cut-sparsifiers with $O(n\varepsilon^{-2}\log n)$-edges that preserves every cut upto a multiplicative factor of $(1 \pm \varepsilon)$. This has been improved to $O(n\varepsilon^{-2})$-edges (optimal).

[ST08] generalized the notion of a cut-sparsifier to a **spectral sparsifier**, which preserves the quadratic form $x^T L_G x \; \forall x \in \mathbb{R}^n$ upto a multiplicative factor of $(1 \pm \varepsilon)$. [SS09] presented a method to construct, with high probability, better spectral sparsifiers with $O(n\varepsilon^{-2}\log n)$-edges via an edge-sampling method. As a result of better sampling methods and more efficient Laplacian solvers, the sparsification procedure of [SS09] has been improved. The current state of the art is sparsifiers with $O(n\varepsilon^{-2}\log n)$-edges that can be constructed in $O(m\log^2 n)$-time ([KPPS16]).

In graphs, every edge connects a pair of vertices, which enables it to model only dyadic relationships. Hypergraphs are the generalization of graphs that contain hyperedges. Each hyperedge $e \subseteq V$ can connect two or more vertices. A non-linear Laplacian operator for hypergraphs was proposed by [CLTZ16] based on diffusion processes. Sparsification for hypergraphs is an arguably more challenging task because they can contain upto $2^{|V|}$ hyperedges, whereas graphs only contain $O(|V|^2)$

*svk2118@columbia.edu. Columbia University.

edges. Still, spectral sparsifiers generalizing the techniques of [SS09] have been constructed for hypergraphs. [BST19] presented an efficient hyperedge-sampling-based algorithm that constructs a spectral sparsifier containing $O(nr^3\varepsilon^{-2}\log n)$-hyperedges with high probability. r is the rank of the hypergraph, or the size of the largest hyperedge. Recently, [Lee22] and [JLS22] independently improved this result to $O(n\log r\varepsilon^{-2}\log n)$-hyperedges, which is the current state-of-the-art.

Spectral sparsifiers have been studied for directed (hyper)graphs and in streaming settings. However, in this project, we will only consider weighted undirected (hyper)graphs.

1.1 Motivation

Spectral sparsifiers for graphs are applicable in several approximation algorithms that are dependent on the number of edges, including approximating cuts, flows, paritioning, and clustering. For hypergraphs, spectral sparsifiers are applicable in efficient algorithms for diffusion on hypergraphs, clustering, semi-supervised learning, etc. We will defer to [KKTY21] for additional discussion.

A newer application is the use of hypergraph spectral sparsifiers for computing dense-cuts and identifying bipartite components. It turns out that the well-studied hypergraph Laplacian operator for which sparsifiers have been constructed is not suited to such applications. Recently, [MS21] defined an alternative, non-linear Laplacian-type operator that captures information about the "bipartiteness" of hypergraph components. We will explore this operator further and make progress towards the goal of constructing a sparsifier for it.

1.2 Organization

This is a survey/research project organized as follows. In the survey component, we will discuss the effective-resistance-based sampling algorithm introduced by [SS09]. We will then generalize this approach to hypergraphs and sketch the proof of the following result of [BST19]:

Theorem 1 ([BST19])**.** *Given a hypergraph $H = (V, E, w)$, there exists a polynomial-time algorithm, that with probability $1 - 1/n$ constructs a ϵ-spectral-sparsifier $\tilde{H} = (V, \tilde{E}, \tilde{w})$ with $|\tilde{E}| = O(n\varepsilon^{-2}r^3\log n)$.*

Finally, we will use a similar sampling approach to construct a sparsifier for the Laplacian-type operator J_H defined by [MS21]. We sketch the proof of the following version of Theorem 1 for the new operator:

Theorem 2 (Research result)**.** *Given a hypergraph $H = (V, E, w)$, there exists a polynomial-time algorithm, that with probability $1 - 1/n$ constructs a sparsifier $\tilde{H}$ that satisfies:*

$$\forall x \in \mathbb{R}^n, \quad (1-\varepsilon)J_{\tilde{H}}(x) \leq J_H(x) \leq (1+\varepsilon)J_{\tilde{H}}(x)$$

1.3 Preliminaries

Let $G = (V, E, w)$ be a weighted undirected graph with $|V| = n, |E| = m$ and $w : E \to \mathbb{R}_{\geq 0}$ (non-negative weights). To capture the notion of cuts, we define the following function: $\mathrm{cut} : 2^V \times 2^V \to$

$\mathbb{R}_{\geq 0}$ such that $\text{cut}(L, R) = \sum_{(u,v)\in E} w_e \cdot \mathbb{1}[(u,v) \cap L \neq \varnothing \wedge (u,v) \cap R \neq \varnothing]$. That is, it is the total weight of all edges that "go between" $L, R \subseteq V$. Note that when $L = S, R = V \setminus S$, the cut is equivalent to $|\partial S|$.

We can define a cut-sparsifier using this function. A ε-cut-sparsifier is defined as a graph $\tilde{G} : \forall S \subseteq V,\ \text{cut}_{\tilde{G}}(S, V \setminus S) \in (1 \pm \epsilon)\text{cut}_G(S, V \setminus S)$. We will now define the Laplacian operator on graphs and prove that it is related to the cut-structure.

Definition 1 (Graph Laplacian)**.** *The graph Laplacian, $L_G = D_G - A_G$, where D_G, A_G are the weighted degree and adjacency matrices of G respectively. When discussing the Laplacian, we are interested in the following quadratic form:*

$$\forall x \in \mathbb{R}^n, \quad x^T L_G x = \sum_{(u,v)\in E} w_e \cdot (x_u - x_v)^2$$

Definition 2 (Graph spectral sparsifier)**.** *Let $G = (V, E, w)$ be a weighted undirected graph. $\tilde{G} = (V, \tilde{E}, \tilde{w})$ is a ε-spectral-sparsifier of G if $(1-\varepsilon)L_{\tilde{G}} \preceq L_G \preceq (1+\varepsilon)L_{\tilde{G}}$. Equivalently:*

$$\forall x \in \mathbb{R}^n, \quad (1-\varepsilon)x^T L_{\tilde{G}} x \leq x^T L_G x \leq (1+\varepsilon)x^T L_{\tilde{G}} x$$

Proposition 1. *Every ε-spectral-sparsifier is a ε-cut-sparsifier.*

Proof. For any set $S \subseteq V$, define the vector χ_S as follows. $\chi_{S_i} = 1 \iff i \in S$, else $\chi_S = 0$. Clearly, $\chi_S^T L_G \chi_S = \text{cut}_G(S, S \setminus V)$. By definition of a spectral sparsifier:

$$\begin{aligned} &\forall S \subseteq V, \quad (1-\varepsilon)x^T L_{\tilde{G}} x \leq x^T L_G x \leq (1+\varepsilon)x^T L_{\tilde{G}} x \\ \implies &\forall S \subseteq V, \quad (1-\varepsilon)\text{cut}_{\tilde{G}}(S, S \setminus V) \leq \text{cut}_G(S, S \setminus V) \leq (1+\varepsilon)\text{cut}_{\tilde{G}}(S, S \setminus V) \\ \implies &\tilde{G} \text{ is a cut-sparsifier for } G \end{aligned}$$

Thus, we have proved that spectral sparsifiers are generalizations of cut-sparsifiers. ∎

Let $H = (V, E, w)$ be a weighted undirected hypergraph with $|V| = n, |E| = m$ and $w : E \to \mathbb{R}_{\geq 0}$ (non-negative weights). The difference is that $\forall e \in E, e \subseteq V$, that is, hyperedges may include more than two nodes and $|E| = O(2^|V|)$. To capture the notion of cuts on hypergraphs, we generalize the cut function for graphs. Namely, $\text{cut} : 2^V \times 2^V \to \mathbb{R}_{\geq 0}$ such that $\text{cut}(L, R) = \sum_{e\in E} w_e \cdot \mathbb{1}[e \cap L \neq \varnothing \wedge e \cap R \neq \varnothing]$. We also define the rank of the hypergraph $r = \max_{e\in E} |e|$.

It is not a straightforward task to come up with a useful generalization of the graph Laplacian. [CLTZ16] defined the following non-linear hypergraph Laplacian operator that is well-studied.

Definition 3 (Hypergraph Laplacian)**.** *Define the following non-linear operator $Q_H(x)$:*

$$\forall x \in \mathbb{R}^n, \quad Q_H(x) = \sum_{e\in E} w_e \max_{(u,v)\in e} (x_u - x_v)^2$$

Using this operator in-place of the graph Laplacian quadratic form, we can define a notion for spectral sparsification for hypergraphs. Similar to the graph Laplacian quadratic form, it is easy to see that the operator $Q_H(x)$ contains information about the cut-structre of the hypergraph, when x is taken to be a cut-representing vector as in the proof of Proposition 1.

Definition 4 (Hypergraph spectral sparsifier)**.** *Let $H = (V, E, w)$ be a weighted undirected hypergraph. $\tilde{H} = (V, \tilde{E}, \tilde{w})$ is a ε-spectral-sparsifier of H if:*

$$\forall x \in \mathbb{R}^n, \quad (1-\varepsilon)Q_{\tilde{H}}(x) \leq Q_H(x) \leq (1+\varepsilon)Q_{\tilde{H}}(x)$$

Physical Intuition. Consider the Laplacian quadractic form $x^T L_G x$ for some graph G. We can choose to interpret the graph G as a one-dimensional system of springs, wherein vertex u is a peg located at position x_u and every edge with weight w_e defines a spring connecting the endpoints with spring-constant w_e. The Laplacian quadratic form represents the total energy of this system. In the same way, we can come up with a physical interpretation for the non-linear hypergraph Laplacian operator $Q_H(x)$ for some hypergraph H. Interpret the hypergraph H as a one-dimensional system of elastic-bands, wherein vertex u is a peg located at position x_u and every hyperedge e with weight w_e is an elastic-band stretched around all vertices in e with elastic-constant w_e. Clearly, the energy of the elastic-band depends on the pegs that are furthest apart, namely on $\max_{(u,v)\in e}(x_u - x_v)^2$. It is clear that $Q_H(x)$ represents the total energy of this system.

It is also possible to study hypergraphs by reducing them to graphs. One such reduction is a clique-reduction, wherein every hyperedge is replaced with a clique.

Definition 5 (Clique-reduction)**.** *Let $H = (V, E, w)$ be a weighted undirected hypergraph. Define clique-reduction $G = (V, E', w')$ as the graph with every edge $e \in E$ replaced by a clique of edges in E' connecting every pair of vertices in $\binom{e}{2}$. In this clique, each new edge has weight w_e.*

Definition 6 (r-uniform hypergraph)**.** *Let $H = (V, E, w)$ be a hypergraph with rank r. H is r-uniform if $\forall e \in E, |e| = r$. In other words, all edges have the same cardinality.*

Proposition 2. *Let $H = (V, E, w)$ be an r-uniform hypergraph and $G = (V, E', w')$ be its clique-reduction. We have:*

$$\forall x \in \mathbb{R} \quad \frac{r}{2} Q_H(x) \leq x^T L_G x \leq \binom{r}{2} Q_H(x)$$

Proof. Consider the Laplacian quadratic form for the clique-reduction:

$$x^T L_G x = \sum_{e\in E} \sum_{(u,v)\in e} (x_u - x_v)^2$$

Essentially, we are comparing $\sum_{(u,v)\in e}(x_u - x_v)^2$ with $\max_{(u,v)\in e}(x_u - x_v)^2$ as is present in the hypergraph operator $Q_H(x)$. For each edge, $|e| = r$. There are $\binom{r}{2}$ pairs $(u, v) \in e, \forall e$. This gives us the upper-bound:

$$\forall (u,v) \in e \quad (x_u - x_v)^2 \leq \max_{(u,v)\in e}(x_u - x_v)^2 \implies \sum_{(u,v)\in e}(x_u - x_v)^2 \leq \binom{r}{2} \max_{(u,v)\in e}(x_u - x_v)^2$$

It takes a little more work to prove the lower-bound. Namely, by expanding the sum for the graph Laplacian and comparing terms, it is evident that the value of $\max_{(u,v)\in e}(x_u - x_v)^2$ can be constructed from the expanded terms $\frac{r}{2}$-times over. This gives us the lower-bound. ■

Proposition 2 provides some intuition for as to why the problem of sparsifying hypergraphs isn't merely reducible to that of sparsifying graphs. Namely, the clique-reduction (in fact, any such reduction) of a hypergraph can only approximate the hypergraph Laplacian and equivalently, its cut-structure, upto a factor of $O(r) = O(n)$.

1.4 [MS21] Laplacian-type operator

As mentioned previously, [MS21] introduced a new non-linear hypergraph Laplacian-type operator that encodes information about the bipartite components of a hypergraph H. To define "bipartiteness" on a hypergraph, we will augment the definition of the cut function to include cuts that "leave out" a set of nodes. Namely, we define: cut : $2^V \times 2^V \times 2^V \to \mathbb{R}_{\geq 0}$ as follows, $\text{cut}(A, B|C) = \sum_{e\in E} w_e \cdot \mathbb{1}[e \cap A \neq \varnothing \wedge e \cap B \neq \varnothing \wedge e \cap C = \varnothing]$.

Definition 7 (Bipartiteness)**.** *Let $H = (V, E, w)$ and $L, R \subseteq V$. The bipartiteness of L and R denoted $\beta(L, R)$ is defined as:*

$$\beta(L,R) = \frac{2\mathit{cut}(L, V\setminus L) + 2\mathit{cut}(R, V\setminus R) + \mathit{cut}(L, V\setminus(L\cup R)|R) + \mathit{cut}(R, V\setminus(L\cup R)R|L)}{\mathit{cut}(L\cup R, L\cup R)}$$

Note that $\mathit{cut}(L\cup R, L\cup R) = \mathit{vol}(L\cup R)$.

We will now define the Laplacian-type operator introduced by [MS21] through studying a heat-diffusion process on hypergraphs.

Definition 8 (Hypergraph Laplacian-type operator)**.** *Define the non-linear operator $J_H(x)$:*

$$\forall x \in \mathbb{R}^n, \quad J_H(x) = \sum_{e\in E} w_e \left(\max_{u\in e} x_u + \min_{v\in e} x_v\right)^2$$

[MS21] shows that, similar to the proof of Proposition 1, the operator J_H contains information (upto factor 2) of the bipartiteness $\beta(L, R)$ $\forall L, R \subseteq V$. By the definition of bipartiteness, we see that sets L, R with low bipartiteness form bipartite components of H. There is also a dense-cut between them, which relates this operator to the problem of max-cut on hypergraphs. This connection, in graphs, was explored in [Tre08].

2 Effective-resistance sampling

In this section, we provide a brief overview of effective-resistance-based sampling approach used to construct graph sparsifiers by [SS09]. Although this project is more focused on hypergraphs, the notion of effective-resistance is useful. The key idea is to sample proportionally to effective-resistance. Thus, effective-resistance is a proxy for the importance of each edge in the sparsifier.

Definition 9 (Effective-resistance)**.** *Let $G = (V, E, w)$. $\forall e \in E, R_e = ||L_G^{-1/2} L_e L_G^{-1/2}||_{op}$. The matrix L_e denotes the edge-Laplacian, or the Laplacian of the graph consisting only of the edge e. $L_G^{-1/2}$ satisfies $L_G^{-1/2} \cdot L_G^{-1/2} = L_G^{-1}$, which is the pseudo-inverse of L_G. $|| \cdot ||_{op}$ denotes the operator-norm on matrices.*

Effective-resistances are also equivalent to the probability that a given edge e appears in a random spanning tree of the graph ([SS09]). Thus, it is reasonable to perform importance-sampling based on effective-resistances, as follows:

Algorithm 1 [SS09]

Input: graph $G = (V, E, w)$
Compute effective-resistances $R_e = ||L_G^{-1/2} L_e L_G^{-1/2}||_{\text{op}}\ \forall e \in E$
Initialize sparsifier $\tilde{G} = (V, \varnothing)$
Let $\ell \leftarrow O(n\varepsilon^{-2} \log n)$. For iteration $i = 1 \ldots \ell$:
  Choose independently a random edge $e \in E$ with probability $p_e \propto w_e R_e$
  Add edge e to $\tilde{G}$ with weight w_e/p_e if $e \notin \tilde{G}$. Else, add to weight of edge $e \in \tilde{G}$ by w_e/p_e.

Proposition 3 ([SS09])**.** *Let $G = (V, E, w)$ be a graph. There exists a polynomial time algorithm for computing effective-resistances $R_e\ \forall e \in E$.*

Theorem 3. *Given a graph $G = (V, E, w)$, Algorithm 1, with probability $\geq 1/2$ constructs a ε-spectral-sparsifier $\tilde{G}$ with $O(n\varepsilon^{-2} \log n)$ edges.*

Theorem 3 can be proved using a straightforward application of matrix Chernoff bounds. Since our focus is on hypergraphs, we omit the proof in the interest of space. For a complete proof see [SS09] and [Ver20] for more information on matrix Chernoff bounds.

3 Bounding random processes

Since the hypergraph Laplacian is a non-linear operator, analyzing effective-resistance-based sampling sparsifiers requires additional machinery compared to a simple application of Matrix Chernoff bounds as in the case of graphs. Namely, we will study how to bound sub-Gaussian processes.

3.1 Sub-Gaussian processes

Let $\mathcal{U}$ be a set. Define a distribution $\mathcal{D}$ over functions $F : \mathcal{U} \to \mathbb{R}$. Our goal is to upper-bound the expected value of the following quantity: $\sup_{x \in \mathcal{U}} F(x)$. We refer to $\{F(x)\}_{x \in \mathcal{U}}$ as a **random process**. While, in general, it is challenging to bound the supremum of random processes, it is more straightforward to do so when the process is Gaussian, through techniques like generic chaining ([Tal21]) and Gaussian matrix Chernoff bounds.

The process that we will analyze, however, is a **sub-Gaussian** process. Each variable is a sub-Gaussian variable, defined as follows:

Theorem 4 (Sub-gaussian variable)**.** *Random variable X is sub-Gaussian if there is a centered Gaussian variable whose tails dominate the tails of X. Formally, $\exists \sigma : \mathbb{E}[e^{X^2/\sigma^2}] = O(1)$.*

We can define the following norm on sub-Gaussian variables:

Theorem 5 (Sub-Gaussian norm)**.** *Let X be a sub-Gaussian random variable. The sub-Gaussian norm $||\cdot||_{\Psi_2}$ is defined as follows: $||X||_{\Psi_2} = \inf\{\sigma : \mathbb{E}[e^{X^2/\sigma^2}] \leq 2\}$.*

Fact 1 (Symmetric Bernoulli sum)**.** *$\forall i$ let $\alpha_i \in \mathbb{R}, X_i$ be an independent symmetric Bernoulli variable which is defined as taking values in $\{\pm 1\}$ with uniform probability $1/2$. $X = \sum \alpha_i X_i$ is sub-Gaussian with $||X||_{\Psi_2} = O\left(\sqrt{\sum_i \alpha_i^2}\right)$.*

Finally, we will introduce the notion of a Gaussian random process dominating a sub-Gaussian process. We will use this notion to later "upper bound" the sub-Gaussian process (that is difficult to analyze directly) with a Gaussian process (that is easier to analyze via matrix Chernoff bounds). The following section on Talagrand's comparison inequality will formalize this notion.

Theorem 6 (Dominates)**.** *Let $\{F(x)\}_{x\in\mathcal{U}}$ be a centered sub-Gaussian random process, and $\{\hat{F}(x)\}_{x\in\mathcal{U}}$ be a centered Gaussian process. We say that $\hat{F}$ $O(1)$-dominates F if:*

$$\forall x, y \in \mathcal{U} \quad ||F(x) - F(y)||_{\Psi_2}^2 \leq O(1) \cdot \mathbb{E}_{\hat{F}}[(\hat{F}(x) - \hat{F}(y))^2]$$

3.2 Talagrand's comparison inequality

The following theorem is key to our analysis of the spectral hypergraph sparsifier. It allows us to bound a sub-Gaussian process by bounding a related dominating Gaussian process.

Theorem 7. *([Ver20, Tal21]) Let $\{F(x)\}_{x\in\mathcal{U}}$ be a centered sub-Gaussian random process, and $\{\hat{F}(x)\}_{x\in\mathcal{U}}$ be a centered Gaussian process that $O(1)$-dominates $\{F(x)\}_{x\in\mathcal{U}}$. Then:*

$$\mathbb{E} \sup_{x\in\mathcal{U}} F(x) \leq \mathbb{E} \sup_{x\in\mathcal{U}} \hat{F}(x)$$

Talagrand's comparison inequality allows us to upper-bound the expectation of the supremum of a sub-Gaussian process $\{F(x)\}_{x\in\mathcal{U}}$, via the analysis of a related Gaussian process $\{\hat{F}(x)\}_{x\in\mathcal{U}}$. In informal terms, we are able to analyze a complicated sub-Gaussian process by analyzing the simpler "upper-bounding" Gaussian process.

Remark 1. *A stronger version of Talagrand's comparison inequality exists that can be used to also prove concentration of the sub-Gaussian process around the expected value. This is shown in [BST19].*

We can apply this machinery for analyzing sub-Gaussian processes to analyzing the hypergraph spectral sparsifiers constructed by [BST19].

4 Sparsifier for Hypergraph Laplacian

Given a hypergraph H, in order to prove that $\tilde{H}$ is an ε-spectral-sparsifier for H, it is sufficient to show that $|Q_H(x) - Q_{\tilde{H}}(x)| \leq \varepsilon \frac{2}{r^2} x^T L_G x$, where G is the clique-reduction of H. By the upper-bound in Proposition 2, this implies $|Q_H(x) - Q_{\tilde{H}}(x)| \leq \varepsilon Q_H(x)$, which implies that $\tilde{H}$ is an ε-spectral sparsifier. Thus, our goal is to prove the following bound: $\mathbb{E} \sup_{x \in \mathbb{R}^n} |Q_H(x) - Q_{\tilde{H}}(x)| \leq \frac{2\varepsilon}{r^2} x^T L_G x$, or equivalently (scaling by $x^T L_G x$), $\mathbb{E} \sup_{\{x : x^T L_G x = 1\}} |Q_H(x) - Q_{\tilde{H}}(x)| \leq \frac{2\varepsilon}{r^2}$.

4.1 [BST19] algorithm

A generalization of the effective-resistance-sampling approach used in Algorithm achieves our goal.

Algorithm 2 [BST19]

Input: hypergraph $H = (V, E, w)$
Compute the clique-reduction $G = (V, E', w')$
Compute effective-resistances for the clique-reduction $R^G_{e'} = ||L_G^{-1/2} L_{e'} L_G^{-1/2}||_{\text{op}}\ \forall e' \in E'$
$\forall$ hyperedges $e \in E$, define the effective-resistance of the hyperedge as $R^H_e = \sum_{e' \in \binom{e}{2}} R^G_{e'}$
Initialize sparsifier $\tilde{H} = (V, \varnothing)$
Independently, for each hyperedge $e \in E$:
    With probability $p_e = O(r^3 \varepsilon^{-2} \log n) \cdot R^H_e$, add hyperedge e to $\tilde{H}$ with weight w_e / p_e.

4.2 Analysis

Based on the properties of effective-resistances, it can be shown that the expected number of hyperedges in the sparsifier, $\sum_e p_e = O(Bn) = O(nr^3 \varepsilon^{-2} \log n)$. Given the independent sampling process, the number of edges is concentrated close to the expectation. It can be shown that, with high probability, Algorithm 2 produces a graph $\tilde{H}$ with $O(nr^3 \varepsilon^{-2} \log n)$-hyperedges. All that remains to be shown is that $\tilde{H}$ is a ε-spectral-sparsifier with high probability.

At a high-level, our proof will proceed as follows: first, we will define a sub-Gaussian process that corresponds to our sampling procedure. Second, we will define a related Gaussian process which dominates the sub-Gaussian process we are interested in bounding. Third, we apply Gaussian matrix Chernoff bounds to bound the Gaussian process. Finally, we apply Theorem 7 to carry-over bounds from the Gaussian process to the sub-Gaussian process of interest.

To make the analysis simpler, we will analyze our construction in rounds. For every p_e, round to the nearest non-positive power of two. Let the smallest such probability be rounded to $2^{-\ell}$. We will analyze the construction in ℓ rounds. Let the hypergraph Laplacian operator at the end of round k be $Q_H^{(k)}(x)$. The round-sampling is analyzed as follows:

- In round 1, $Q_H^{(1)}(x) = Q_H(x)$.
- In round k, sample all hyperedges with $p_e \leq 2^{k-1-\ell}$. In each sampling step, choose to either delete the hyperedge or double its weight with equal probability.

- By round ℓ, based on our round-sampling, $Q_H^{(\ell)}(x) = Q_{\tilde{H}}(x)$ (sparsified Laplacian).

The advantage of round-sampling is that:

$$F^{(k)}(x) \coloneqq Q_H^{(k)}(x) - Q_H^{(k-1)}(x) = \sum_{e \in E_k} b^{(k)} w_e^{(k-1)} \max_{(u,v)\in e} (x_u - x_v)^2$$

where $b^{(k)}$ are independent symmetric Bernoulli variables and E_k represents the edges being sampled in round k. Thus, $\forall k,\ F^{(k)}(x)$ is a sub-Gaussian process whose sub-Gaussian norm can be computed easily from Fact 1.

Thus, we can express the following bound:

$$\mathbb{E} \sup |Q_H(x) - Q_{\tilde{H}}(x)| \leq \sum_{i=2}^{\ell} \mathbb{E} \sup |Q_H^{(k)}(x) - Q_H^{(k-1)}(x)| = \sum_{i=2}^{\ell} \mathbb{E} \sup |F^{(k)}(x)|$$

where all supremums are taken over the domain $\{x : x^T L_G x = 1\}$.

Our next task is to bound the sub-Gaussian processes $F^{(k)}(x)$. It is in this step that our machinery for sub-Gaussian processes will be useful. Namely, we will see that it is easy to construct a Gaussian process that dominates this sub-Gaussian process of interest and then bound that.

Definition 10 (Related Gaussian process). $\forall k$, *define Gaussian process* $\hat{F}^{(k)}(x)$ *as follows:*

$$\hat{F}^{(k)}(x) = \sum_{e \in E_k} g^{(k)} w_e^{(k-1)} \sum_{(u,v)\in e} (x_u - x_v)^2$$

where $g^{(k)}$ *are independent standard normal variables.*

The operators defined in the Gaussian process $\forall k$ are linear in the Laplacian of the clique-reduction of the hypergraph in round k. Thus, they are easy to bound using the same techniques as for regular graphs.

Proposition 4. $\forall k, \quad \sup_{\{x : x^T L_G x = 1\}} |F^{(k)}(x)| \leq O(\frac{\varepsilon}{r^2} 2^{(k-\ell)/2})$

The proof of Lemma 4 can be done through the application of matrix Chernoff bounds, similar to the proof of Theorem 3, given the linear and Gaussian nature of $\hat{F}$. We will omit the proof for the same reason as for Theorem 3. A proof of this lemma can be found in [BST19].

Finally, in order to "carry over" the upper-bound from the Gaussian to the sub-Gaussian process by Talagrand's comparison inequality, we have to prove the following:

Proposition 5 (Key Proposition). $|\hat{F}^{(k)}(x)|$ $O(1)$-*dominates* $|F^{(k)}(x)|$.

Proof. According to the definition of domination:

$$\forall k,\ ||F^{(k)}(x) - F^{(k)}(y)||_{\Psi_2} = O(1) \sum_{e \in E_k} (w_e^{(k-1)})^2 \left(\max_{(a,b)\in e} (x_a - x_b)^2 - \max_{(c,d)\in e} (y_c - y_d)^2 \right)^2 \quad \text{(Fact 1)}$$

$$\leq O(1) \sum_{e \in E_k} (w_e^{(k-1)})^2 \max_{(a,b)\in e} \left((x_a - x_b)^2 - (y_a - y_b)^2\right)^2$$
$$\leq O(1) \sum_{e \in E_k} (w_e^{(k-1)})^2 \sum_{(a,b)\in e} \left((x_a - x_b)^2 - (y_a - y_b)^2\right)^2$$
$$= \mathbb{E}_{\hat{F}}[(\hat{F}(x) - \hat{F}(y))^2]$$

$\implies |\hat{F}^{(k)}(x)|$ $O(1)$-dominates $|F^{(k)}(x)|$. ■

Finally, we can put these pieces together to prove Theorem 1.

Proof. We have, as desired:

$$\mathbb{E} \sup_{\{x : x^T L_G x = 1\}} |Q_H(x) - Q_{\tilde{H}}(x)| \leq \sum_{i=2}^{\ell} \mathbb{E} \sup |F^{(k)}(x)| \leq \sum_{i=2}^{\ell} \mathbb{E} \sup |\hat{F}^{(k)}(x)| \leq \frac{2\varepsilon}{r^2}$$

Note that we have only shown that Algorithm 2 constructs a sparsifier **in expectation**. However, to prove that it does so with high probability $1 - 1/n$, we require a stronger version of Talagrand's comparison inequality (see Remark 1). ■

5 Sparsifier for [MS21] operator

We will define a Laplacian-type linear operator J_G for graphs corresponding to the non-linear hypergraph Laplacian operator J_H: $x^T J_G x = \sum_{(u,v)\in E} w_e \cdot (x_u + x_v)^2$. J_G is clearly positive-semi-definite, thus the effective-resistances computed using it are well-defined.

The idea is to repeat the same algorithm as above, except using J_G instead of L_G for clique-reductions and effective-resistances since the goal is to sparsify $J_H(x)$ rather than $Q_H(x)$.

5.1 Algorithm

Algorithm 3 Effective-resistance-type sampling

Input: hypergraph $H = (V, E, w)$

Compute the clique-reduction $G = (V, E', w')$ using J_G

Compute effective-resistances for the clique-reduction $R^G_{e'} = ||J_G^{-1/2} J_{e'} J_G^{-1/2}||_{\text{op}}\ \forall e' \in E'$

$\forall$ hyperedges $e \in E$, define the effective-resistance of the hyperedge as $R^H_e = \sum_{e' \in \binom{e}{2}} R^G_{e'}$

Initialize sparsifier $\tilde{H} = (V, \varnothing)$

Independently, for each hyperedge $e \in E$:

  With probability $p_e = O(r^3 \varepsilon^{-2} \log n) \cdot R^H_e$, add hyperedge e to $\tilde{H}$ with weight w_e/p_e.

The proof of Theorem 2 follows a similar structure to the proof of Theorem 1, with the exception that J_G is used instead of L_G. This proof has been omitted since it is a restatement of the previous proof with some more careful analysis.

5.2 Future directions

- Apply the techniques described in [Lee22] to construct an improved sparsifier for J_H with $O(n\varepsilon^{-2}\log r \log n)$-hyperedges.
- Apply the current sparsifier to improve the polynomial-dependence on $|E_H|$ of the diffusion algorithms described in [MS21] to linear-dependence via an initial sparsification step.
- Generalize the ideas of [Tre08] to hypergraphs using the J_H operator and sparsifier to efficiently compute approximations for hypergraph max-cut.

6 Conclusion

Although some details and proofs in this project have been omitted for the sake of concision, the purpose was to highlight the **key idea** that the sparsification of non-linear operators is more challenging to analyze than the sparsification of linear operators (where an application of matrix Chernoff bounds usually suffices). The machinery for analyzing sub-Gaussian processes is helpful for this and can also be applied in many other contexts in theoretical CS. Finally, we also presented a sparsifier for the Laplacian-like operator introduced by [MS21], which has several potential applications in efficient approximation algorithms for dense-cuts and bipartite components.

References

[BST19] Nikhil Bansal, Ola Svensson, and Luca Trevisan. New Notions and Constructions of Sparsification for Graphs and Hypergraphs. 2019.

[CLTZ16] T-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral Properties of Hypergraph Laplacian and Approximation Algorithms. 2016.

[JLS22] Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Chaining, Group Leverage Score Overestimates, and Fast Spectral Hypergraph Sparsification. 2022.

[KKTY21] Michael Kapralov, Robert Krauthgamer, Jakab Tardos, and Yuichi Yoshida. Towards Tight Bounds for Spectral Sparsification of Hypergraphs. 2021.

[KPPS16] Rasmus Kyng, Jakub Pachocki, Richard Peng, and Sushant Sachdeva. A Framework for Analyzing Resparsification Algorithms. 2016.

[Lee22] James R. Lee. Spectral hypergraph sparsification via chaining. 2022.

[MS21] Peter Macgregor and He Sun. Finding Bipartite Components in Hypergraphs. 2021.

[SS09] Daniel A. Spielman and Nikhil Srivastava. Graph Sparsification by Effective Resistances. 2009.

[ST08] Daniel A. Spielman and Shang-Hua Teng. Spectral Sparsification of Graphs. 2008.

[Tal21] Michel Talagrand. *Upper and Lower Bounds for Stochastic Processes: Decomposition Theorems*. 2021.

[Tre08] Luca Trevisan. Max Cut and the Smallest Eigenvalue. 2008.

[Ver20] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. 2020.