Online Algorithms for Spectral Hypergraph Sparsification

Yuichi Yoshida
(National Institute of Informatics)

Joint work with
Kam Chuen Tung (University of Waterloo)
Tasuku Soma (Institute of Statistical Mathematics)
Slides mostly by Soma. Mistakes mostly by Yoshida.

Graph Laplacian

$G = (V, E, w)$: (undirected) graph with edge weights $w \in \mathbb{R}_+^E$

Graph Laplacian Matrix

$$L_G = \sum_{e=\{i,j\} \in E} w_e (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top$$

Graph Laplacian

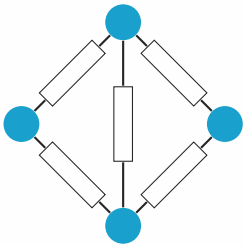
$G = (V, E, w)$: (undirected) graph with edge weights $w \in \mathbb{R}_+^E$

Graph Laplacian Matrix

$$L_G = \sum_{e=\{i,j\} \in E} w_e (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top$$

Quadratic form

$$Q_G(x) := \boldsymbol{x}^\top L_G \boldsymbol{x} = \sum_{e=\{i,j\} \in E} w_e (x_i - x_j)^2$$



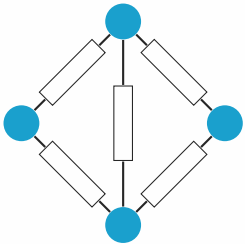
Energy of G (as a circuit) with potential $\boldsymbol{x}$

Graph Laplacian

$G = (V, E, w)$: (undirected) graph with edge weights $w \in \mathbb{R}_+^E$

Quadratic form

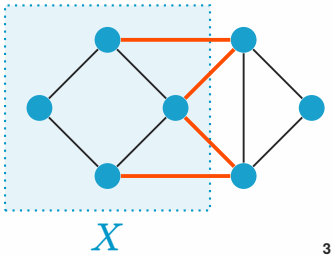
$$Q_G(x) := \boldsymbol{x}^\top L_G \boldsymbol{x} = \sum_{e=\{i,j\} \in E} w_e (x_i - x_j)^2$$



Cut function

For a vertex set $X \subseteq V$,

$$\kappa_G(X) = \sum_{\{i,j\} \in E : |\{i,j\} \cap X| = 1} w_e = Q_G(\mathbf{1}_X).$$

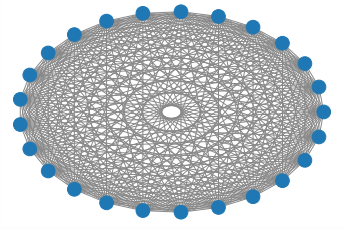


Spectral Graph Sparsification [Spielman, Teng 2011]

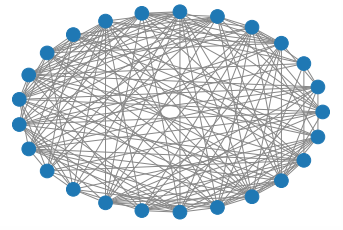
edge subset $\tilde{E} \subseteq E$ ····· ····· edge weight can be modified

A weighted subgraph $\tilde{G} = (V, \tilde{E}, \tilde{w})$ of G is an **ε-spectral sparsifier**

$$\overset{\triangle}{\iff} \quad (1-\varepsilon)Q_G(\boldsymbol{x}) \leq Q_{\tilde{G}}(\boldsymbol{x}) \leq (1+\varepsilon)Q_G(\boldsymbol{x}) \qquad (\forall \boldsymbol{x} \in \mathbb{R}^V)$$



G



$\tilde{G}$

- Spectral sparsifier $\implies$ all cuts are preserved (i.e., **cut sparsifier**)
- $O(n/\varepsilon^2)$ edges is sufficient [Batson, Spielman, Srivastava 2014; Y. T. Lee, Sun 2015]

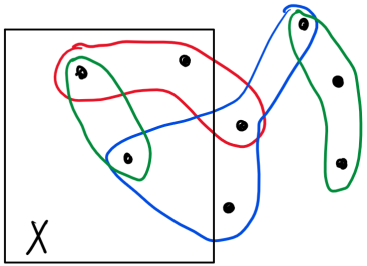
Hypergraph Cut

$H = (V, \mathcal{E}, w)$: hypergraph with edge weight $w \in \mathbb{R}_+^{\mathcal{E}}$

Hypergraph cut function

For vertex subset $X \subseteq V$,

$$\kappa_H(X) = \sum_{e \in \mathcal{E} : 0 < |e \cap X| < |e|} w_e$$



Hypergraph cut function is **NOT** a quadratic form in the usual sense!

Spectral Hypergraph Sparsification [Soma, Yoshida 2019]

Energy function

$$Q_H(\boldsymbol{x}) := \sum_{e \in \mathcal{E}} w_e \max_{i,j \in e} (x_i - x_j)^2$$

In particular, $Q_H(\mathbf{1}_X) = \kappa_H(X)$ for $X \subseteq V$.

Spectral Hypergraph Sparsification [Soma, Yoshida 2019]

Energy function

$$Q_H(\boldsymbol{x}) := \sum_{e \in \mathcal{E}} w_e \max_{i,j \in e} (x_i - x_j)^2$$

In particular, $Q_H(\mathbf{1}_X) = \kappa_H(X)$ for $X \subseteq V$.

A weighted subhypergraph $\tilde{H}$ of H is an **ε-spectral sparsifier**

$$\overset{\triangle}{\iff} \quad (1-\varepsilon)Q_H(\boldsymbol{x}) \leq Q_{\tilde{H}}(\boldsymbol{x}) \leq (1+\varepsilon)Q_H(\boldsymbol{x}) \qquad (\forall \boldsymbol{x} \in \mathbb{R}^V)$$

spectral sparsifier $\implies$ cut sparsifier [Kogan, Krauthgamer 2015]

Bounds for offline hypergraph sparsification

reference	cut	spectral
Kogan, Krauthgamer (2015)	$O(\frac{n(r+\log n)}{\varepsilon^2})$	
Soma, Yoshida (2019)		$O(\frac{n^3}{\varepsilon^2}\log n)$
Bansal, Svensson, Trevisan (2019)		$O(\frac{nr^3}{\varepsilon^2}\log n)$
Chen, Khanna, Nagda (2020a)	$O(\frac{n}{\varepsilon^2}\log n)$	
Kapralov, Krauthgamer, Tardos, Yoshida (2021)		$\tilde{O}(\frac{nr}{\varepsilon^{O(1)}})$
Kapralov, Krauthgamer, Tardos, Yoshida (2022)		$O(\frac{n}{\varepsilon^4}\log^3 n)$
J. R. Lee (2023) Jambulapati, Liu, Sidford (2023)		$O(\frac{n}{\varepsilon^2}\log n \log r)$

$n = |V|, r = \max_{e\in\mathcal{E}} |e|$

Drawbacks of offline sparsification algorithms

To store the input hypergraph, we may need **exponential** space in n

But, the output sparsifier has only $O(\varepsilon^{-2} n \log n \log r)$ hyperedges, which is **nearly linear!**

Q. Can we construct a spectral sparsifier of a hypergraph using only polynomial space?

Online hypergraph spectral sparsification

- Hyperedges and weights $(e_1, w_1), (e_2, w_2), \ldots, (e_m, w_m)$ arrive in stream.
- When (e_i, w_i) arrives, we **irrevocably** decide whether to include e_i to $\tilde{H}$ along with its weight.
- Only $\mathrm{poly}(n)$ working space is available.

Online hypergraph spectral sparsification

- Hyperedges and weights $(e_1, w_1), (e_2, w_2), \ldots, (e_m, w_m)$ arrive in stream.
- When (e_i, w_i) arrives, we **irrevocably** decide whether to include e_i to $\tilde{H}$ along with its weight.
- Only $\mathrm{poly}(n)$ working space is available.

A weighted subhypergraph $\tilde{H}$ of H is an (ε, δ)**-spectral sparsifier**

$$\overset{\triangle}{\Longleftrightarrow} \quad (1-\varepsilon)Q_H(\boldsymbol{x}) - \delta\|\boldsymbol{x}\|_2^2 \le Q_{\tilde{H}}(\boldsymbol{x}) \le (1+\varepsilon)Q_H(\boldsymbol{x}) + \delta\|\boldsymbol{x}\|_2^2 \qquad (\forall \boldsymbol{x} \in \mathbb{R}^V)$$

Online hypergraph spectral sparsification

- Hyperedges and weights $(e_1, w_1), (e_2, w_2), \ldots, (e_m, w_m)$ arrive in stream.
- When (e_i, w_i) arrives, we **irrevocably** decide whether to include e_i to $\tilde{H}$ along with its weight.
- Only $\mathrm{poly}(n)$ working space is available.

A weighted subhypergraph $\tilde{H}$ of H is an (ε, δ)**-spectral sparsifier**

$$\overset{\triangle}{\iff} \quad (1-\varepsilon)Q_H(\boldsymbol{x}) - \delta\|\boldsymbol{x}\|_2^2 \le Q_{\tilde{H}}(\boldsymbol{x}) \le (1+\varepsilon)Q_H(\boldsymbol{x}) + \delta\|\boldsymbol{x}\|_2^2 \qquad (\forall \boldsymbol{x} \in \mathbb{R}^V)$$

Theorem (Soma, Tung, and Yoshida '24)

There is an online algorithm that outputs an (ε, δ)-spectral sparsifier with $O(\varepsilon^{-2} n \log n \log r \log \frac{\varepsilon W}{\delta n})$ many hyperedges using $O(n^2)$ space. (Here, $W = \sum_i w_i$)

Online hypergraph spectral sparsification

- Hyperedges and weights $(e_1, w_1), (e_2, w_2), \ldots, (e_m, w_m)$ arrive in stream.
- When (e_i, w_i) arrives, we **irrevocably** decide whether to include e_i to $\tilde{H}$ along with its weight.
- Only $\mathrm{poly}(n)$ working space is available.

A weighted subhypergraph $\tilde{H}$ of H is an (ε, δ)**-spectral sparsifier**

$$\overset{\triangle}{\iff} \quad (1-\varepsilon)Q_H(\boldsymbol{x}) - \delta\|\boldsymbol{x}\|_2^2 \le Q_{\tilde{H}}(\boldsymbol{x}) \le (1+\varepsilon)Q_H(\boldsymbol{x}) + \delta\|\boldsymbol{x}\|_2^2 \qquad (\forall \boldsymbol{x} \in \mathbb{R}^V)$$

Theorem (Soma, Tung, and Yoshida '24)

For unweighted r-uniform hypergraphs, there is an online algorithm that outputs an ε-spectral sparsifier with $O(\varepsilon^{-2} nr \log^2 n \log r)$ many hyperedges using $O(n^2)$ space.

Edge sampling algorithm

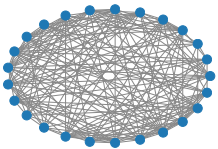
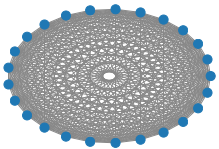
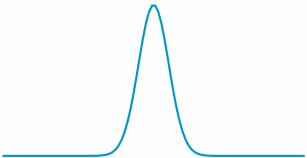
Algorithm

1: Compute edge sampling probability $p_e \in [0, 1]$ for every hyperedge $e \in \mathcal{E}$.
2: **for** each hyperedge e :
3: Add a copy of e to $\tilde{H}$ with weight w_e/p_e w.p. p_e.

Properties

- $\mathbf{E}_{\tilde{H}}[Q_{\tilde{H}}(\boldsymbol{x})] = Q_H(\boldsymbol{x})$ (i.e., unbiased)
- Expected total # of hyperedges $= \sum_{e \in \mathcal{E}} p_e$

Challenge

	small p_e	large p_e
# of edges	 sparse	 dense
Variance of cut weight	 large	 small

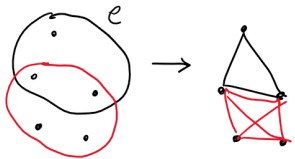
Need to carefully choose p_e!

Reweighting

[Kapralov, Krauthgamer, Tardos, Yoshida 2022; J. R. Lee 2023; Jambulapati, Liu, Sidford 2023]

Clique expansion:

$G = (V, F)$, each hyperedge being replaced with clique

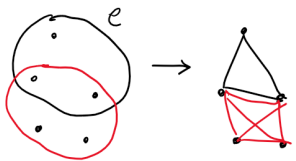


Reweighting

[Kapralov, Krauthgamer, Tardos, Yoshida 2022; J. R. Lee 2023; Jambulapati, Liu, Sidford 2023]

Clique expansion:

$G = (V, F)$, each hyperedge being replaced with clique



Lemma ([Kapralov, Krauthgamer, Tardos, Yoshida 2022; J. R. Lee 2023])

There are edge weights $c_{e,u,v} \geq 0$ on F such that

- $\sum_{u,v \in e} c_{e,u,v} = w_e \quad (e \in E)$
- $c_{e,u,v} > 0 \implies r_{u,v} = \max_{u',v' \in e} r_{u',v'}$, *where $r_{u,v}$ is the effective resistance between u, v.*

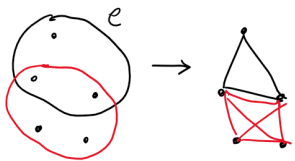
Effective resistance:

$r_{u,v} = (\boldsymbol{e}_u - \boldsymbol{e}_v)^\top L_G^+ (\boldsymbol{e}_u - \boldsymbol{e}_v)$

Reweighting

[Kapralov, Krauthgamer, Tardos, Yoshida 2022; J. R. Lee 2023; Jambulapati, Liu, Sidford 2023]

Clique expansion:
$G = (V, F)$, each hyperedge being replaced with clique



Lemma ([Kapralov, Krauthgamer, Tardos, Yoshida 2022; J. R. Lee 2023])

There are edge weights $c_{e,u,v} \geq 0$ on F such that

- $\sum_{u,v \in e} c_{e,u,v} = w_e \quad (e \in E)$
- $c_{e,u,v} > 0 \implies r_{u,v} = \max_{u',v' \in e} r_{u',v'}$, *where $r_{u,v}$ is the effective resistance between u, v.*

Effective resistance:
$r_{u,v} = (\boldsymbol{e}_u - \boldsymbol{e}_v)^\top L_G^+ (\boldsymbol{e}_u - \boldsymbol{e}_v)$

Given $\{c_{e,u,v}\}$, $p_e \propto w_e \max_{u,v \in e} r_{u,v}$ yields ε-spectral sparsifier with $O(\varepsilon^{-2} n \log n \log r)$ hyperedges. [J. R. Lee 2023; Jambulapati, Liu, Sidford 2023]

Graph case: effective resistance sampling [Spielman, Srivastava 2011]

Convex optimization for reweighting [J. R. Lee 2023]

The edge weight $\{c_{e,u,v}\}$ can be found by convex optimization:

$$\max \quad \log\det\left(\sum_{e\in E}\sum_{u,v\in V} c_{e,u,v}L_{u,v} + J\right)$$

$$\text{sub. to} \quad \sum_{u,v\in e} c_{e,u,v} = w_e \quad (e \in \mathcal{E})$$

$$c_{e,u,v} \geq 0 \quad (e \in E, u, v \in e)$$

$L_{u,v} := (\boldsymbol{e}_u - \boldsymbol{e}_v)(\boldsymbol{e}_u - \boldsymbol{e}_v)^\top$
J: all-one matrix

KKT condition $\Longrightarrow$ the required conditions for reweighting.

Our online algorithm

(cf. online spectral sparsification of graphs [Cohen, Musco, Pachocki 2020])

Let $\eta = \delta/\varepsilon > 0$. (regularization parameter)

Define matrix L_i ($i = 0, 1, \dots$) as follows:

- $L_0 = O_n$,
- $L_i = L_{i-1} + \sum_{u,v \in e_i} w_i c_{i,u,v} L_{u,v}$, where $c_{i,u,v}$ is the solution of the following convex optimization:

$$\max_{c_i \in \Delta_{e_i}} \log\det\left(L_{i-1} + \sum_{u,v \in e_i} w_i c_{i,u,v} L_{u,v} + \eta I_n \right)$$

Laplacian matrix up to $i-1$ .... ridge regularizer

Δ_e: simplex in $\mathbb{R}^{\binom{e}{2}}$

Our online algorithm

Let $\eta = \delta/\varepsilon > 0$.

1: $\tilde{H}_0 := \emptyset$, $L_0 := O_n$ ····· Laplacian matrix for reweighted clique expansion

2: **for** $i = 1, 2, \dots$:

3: Solve the following convex optimization:

$$\max_{c_i \in \Delta_{e_i}} \log\det\left(L_{i-1} + \sum_{u,v \in e_i} w_i c_{i,u,v} L_{u,v} + \eta I_n \right)$$

····· ridge regularizer

Δ_e: simplex in $\mathbb{R}^{\binom{e}{2}}$

$L_{u,v} := (\boldsymbol{e}_u - \boldsymbol{e}_v)(\boldsymbol{e}_u - \boldsymbol{e}_v)^\top$

4: $L_i \leftarrow L_{i-1} + \sum_{u,v \in e_i} w_i c_{i,u,v} L_{u,v}$

5: $p_i \propto w_i \max_{u,v \in e_i} \|(L_i + \eta I)^{-1/2}(\boldsymbol{e}_u - \boldsymbol{e}_v)\|_2^2$. ····· ridged effective resistance

6: Add e_i with weight w_i/p_i to $\tilde{H}_{i-1}$ with probability p_i and obtain $\tilde{H}_i$.

Analysis

H_i: input hypergraph up to time i

- L_i is deterministically determined by H_i.
  $\longrightarrow$ Sampling of each e_i is independent
- Maintains only sparsifier $\tilde{H}_i$ and Laplacian matrix L_i.
  $\longrightarrow O(n^2)$ space
- $\tilde{H}_i$ is an (ε, δ)-spectral sparsifier of H_i w.h.p.
  Based on the **chaining** analysis by [J. R. Lee 2023].

Analysis

The expected # of hyperedges $= \tilde{O}(\varepsilon^{-2} n \log \frac{\varepsilon W}{\delta n})$

Consider the potential function $\Phi_i := \log\det L_i$.

- $\Phi_0 = \log\det \eta I = n \log \eta$.
- $\Phi_i - \Phi_{i-1} = \Omega(p_i)$ at every step i.
- $\Phi_m = \log\det(L_m + \eta I) \leq n \log\left(\frac{\mathrm{tr} L_m}{n} + \eta\right) = n \log\left(\frac{2W}{n} + \eta\right)$.

$\longrightarrow$ Expected # of hyperedges

$$\approx \sum_{i=1}^{m} p_i \leq \sum_{i=1}^{m} (\Phi_i - \Phi_{i-1}) = \Phi_m - \Phi_0 = n \log\left(\frac{2W}{\eta n} + 1\right)$$

Summary

- **Online algorithm:** $\tilde{O}(\epsilon^{-2} n \log \frac{\varepsilon W}{\delta n})$ hyperedges with $O(n^2)$ space
  - We need $O(\epsilon^{-2} n \log \frac{\varepsilon W}{\delta n})$ for graphs [Cohen, Musco, Pachocki 2020].
- Can we reduce the space complexity from $O(n^2)$ to, say, $O(nr)$?

- For the fully dynamic setting, $\tilde{O}(\epsilon^{-2} nr \log m)$-sized sketch exists for cut sparsification $\Rightarrow$ Putterman's talk on Wednesday!

Summary

- **Online algorithm:** $\tilde{O}(\epsilon^{-2} n \log \frac{\varepsilon W}{\delta n})$ hyperedges with $O(n^2)$ space
  - We need $O(\epsilon^{-2} n \log \frac{\varepsilon W}{\delta n})$ for graphs [Cohen, Musco, Pachocki 2020].
- Can we reduce the space complexity from $O(n^2)$ to, say, $O(nr)$?

- For the fully dynamic setting, $\tilde{O}(\epsilon^{-2} nr \log m)$-sized sketch exists for cut sparsification $\Rightarrow$ Putterman's talk on Wednesday!

Thanks! Questions?