Lecture 4: Spectral sparsification in dynamic streams

Michael Kapralov[1]

[1]EPFL

May 26, 2017

Algorithms for massive graphs

Massive networks ubiquitous in data processing

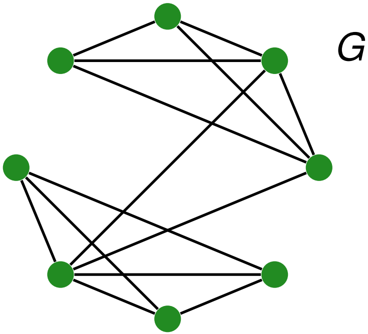
- $\geq$ 100 billion edges
- graph does not fit into memory of single computer
- with metadata, does not fit on a single hard drive

Social distance between nodes, community detection,...

Compress the network while preserving useful properties?

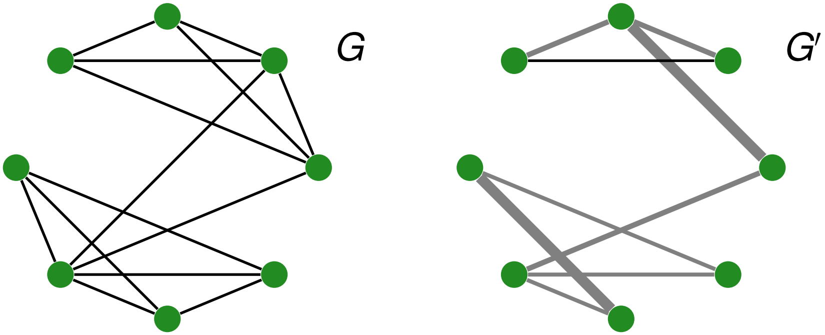
Sparsification

- Let $G = (V, E)$ be an undirected graph, where $|V| = n, |E| = m$.
- Find a smaller subgraph G' of G that approximates G



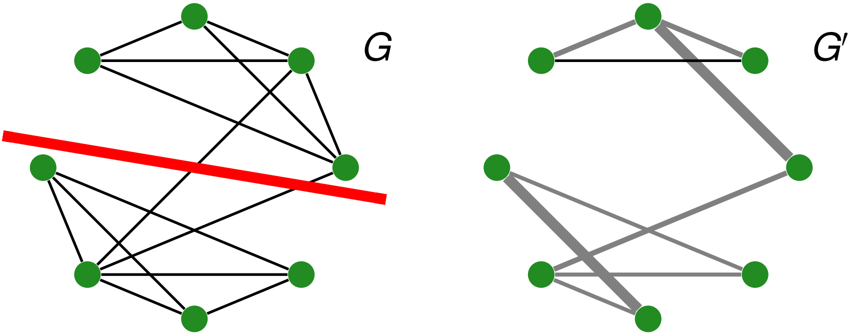
Sparsification

- Let $G = (V, E)$ be an undirected graph, where $|V| = n, |E| = m$.
- Find a smaller subgraph G' of G that approximates G



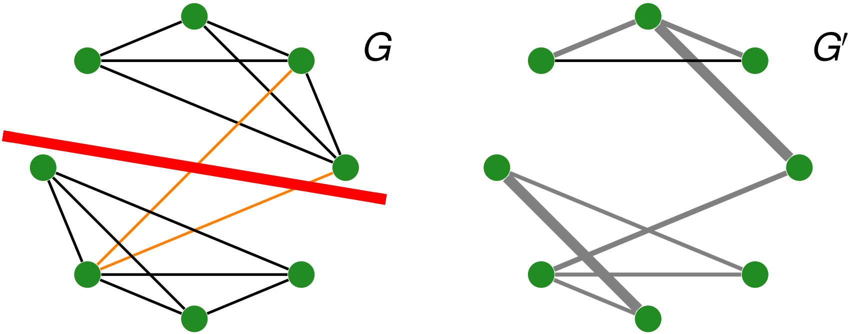
Sparsification

- Let $G = (V, E)$ be an undirected graph, where $|V| = n, |E| = m$.
- Find a smaller subgraph G' of G that approximates G



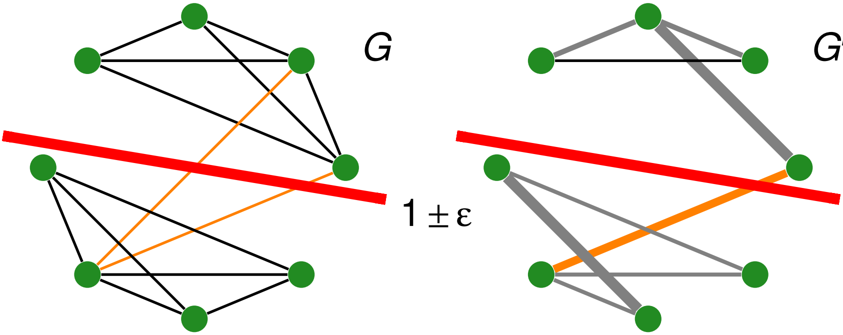
Sparsification

- Let $G = (V, E)$ be an undirected graph, where $|V| = n, |E| = m$.
- Find a smaller subgraph G' of G that approximates G



Sparsification

- Let $G = (V, E)$ be an undirected graph, where $|V| = n, |E| = m$.
- Find a smaller subgraph G' of G that approximates G

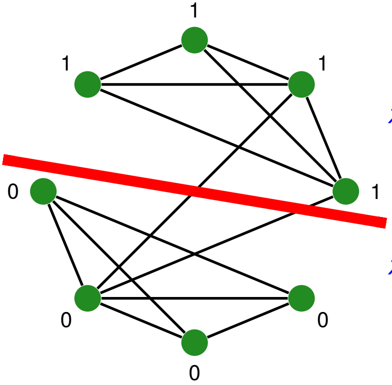


1. Spectral sparsification
2. Streaming model of computation
3. Dynamic streaming and linear sketching
4. Spectral sparsification via linear sketches

1. **Spectral sparsification**
2. Streaming model of computation
3. Dynamic streaming and linear sketching
4. Spectral sparsification via linear sketches

Sparsification

$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2$$

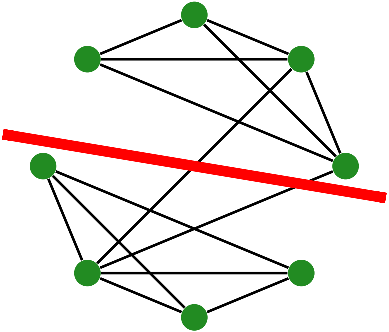


$x_u = 1$

$x_u = 0$

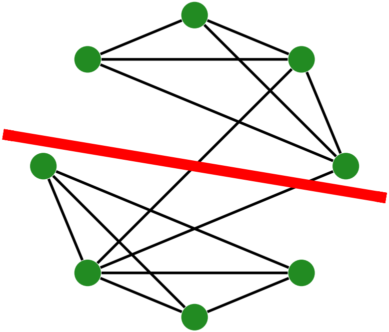
Sparsification

$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2$$



Sparsification

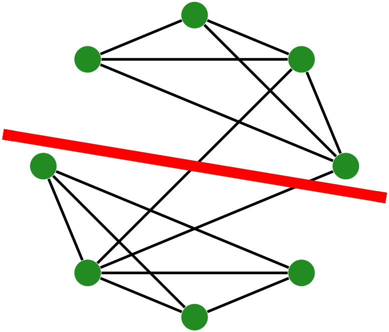
$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2 = ||z||^2$$



$$z = \begin{bmatrix} 0 \\ x_1 - x_2 \\ 0 \\ x_4 - x_3 \\ 0 \\ x_4 - x_5 \\ \vdots \end{bmatrix}$$

Sparsification

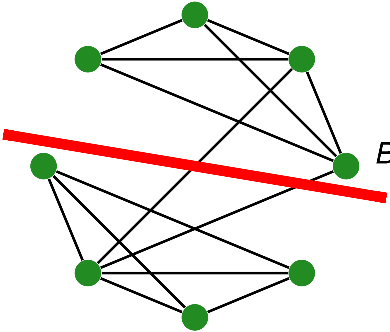
$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2 = ||z||^2$$



$$z = \begin{bmatrix} 0 \\ x_1 - x_2 \\ 0 \\ x_4 - x_3 \\ 0 \\ x_4 - x_5 \\ \vdots \end{bmatrix} = Bx$$

Sparsification

$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2 = ||Bx||^2$$

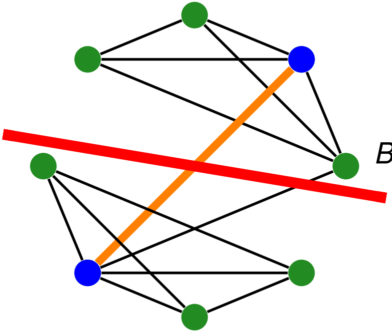


$$Bx = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$\binom{n}{2} \times n$

Sparsification

$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2 = ||Bx||^2$$

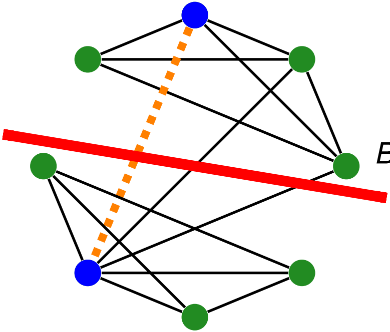


$$Bx = \begin{bmatrix} \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$\binom{n}{2} \times n$

Sparsification

$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2 = \color{red}{||Bx||^2}$$



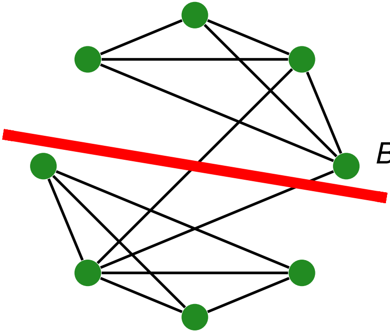
$$Bx = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$\color{red}{\binom{n}{2} \times n}$

Sparsification

$$\text{value of cut} = \sum_{e=(u,v)\in E} (x_u - x_v)^2 = ||Bx||^2 = x^T B^T B x$$

$L = B^T B$ is the Laplacian of G



$$Bx = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$\binom{n}{2} \times n$

Definition (Karger'94, Cut sparsifiers)

G' is an ε-cut sparsifier of G if

$$(1-\varepsilon)x^T Lx \leq x^T L' x \leq (1+\varepsilon)x^T Lx$$

for all $x \in \{0,1\}^V$ (all cuts).

Theorem (Karger'94, Benczur-Karger'96)

For any G there exists an ε-cut sparsifier G' with $O(\frac{1}{\varepsilon^2} n \log n)$ edges, and it can be constructed in $\widetilde{O}(m)$ time.

Definition (Spielman-Teng'04, Spectral sparsifiers)

G' is an ε-spectral sparsifier of G if

$$(1-\varepsilon)x^T Lx \le x^T L' x \le (1+\varepsilon)x^T Lx$$

for all ~~$x \in \{0,1\}^V$ (all cuts).~~ all $x \in \mathbb{R}^V$.

Equivalently, $(1-\varepsilon)L \prec L' \prec (1+\varepsilon)L$

Theorem (Spielman-Teng'04, Spielman-Srivastava'09)

For any G there exists an ε-spectral sparsifier G' with $O(\frac{1}{\varepsilon^2} n \log n)$ edges, and it can be constructed in $\tilde{O}(m)$ time.

Definition (Spielman-Teng'04, Spectral sparsifiers)

G' is an ε-spectral sparsifier of G if

$$(1-\varepsilon)x^T Lx \le x^T L' x \le (1+\varepsilon)x^T Lx$$

for all ~~$x \in \{0,1\}^V$ (all cuts).~~ all $x \in \mathbb{R}^V$.

Equivalently, $(1-\varepsilon)L \prec L' \prec (1+\varepsilon)L$

Theorem (Spielman-Teng'04, Spielman-Srivastava'09)

For any G there exists an ε-spectral sparsifier G' with $O(\frac{1}{\varepsilon^2} n \log n)$ edges, and it can be constructed in $\widetilde{O}(m)$ time.

Karger'94, Benczur-Karger'96, Fung-Hariharan-Harvey-Panigrahi'11

Spielman-Teng'04, Spielman-Srivastava'08, Batson-Spielman-Srivastava'09,

Kolla-Makarychev-Saberi-Teng'10, Koutis-Levin-Peng'12, Kapralov-Panigrahy'12

Implications for numerical linear algebra, combinatorial optimization etc

Constructing spectral sparsifiers

Theorem (Spielman-Srivastava'09)

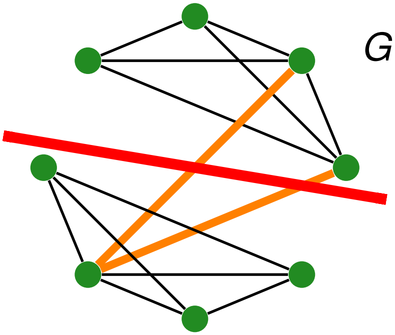
Let $G = (V, E)$ be an undirected graph. Let G' be obtained by including every edge $e \in E$ independently with probability proportional to its effective resistance:

$$p_e \geq \min\{1, \frac{C \log n}{\varepsilon^2} R_e\}.$$

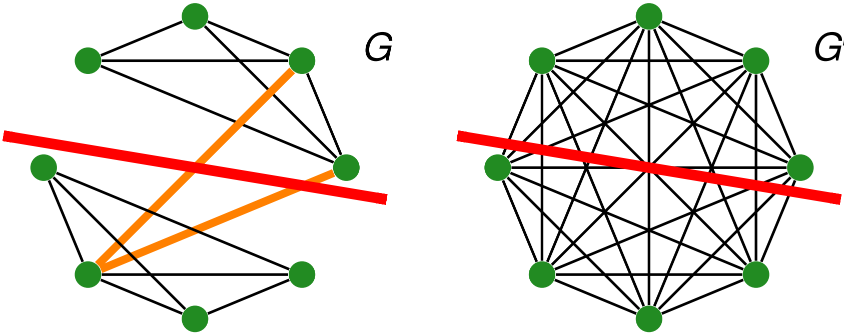
Assigning weight $1/p_e$ if sampled. Then $(1-\varepsilon)L \prec L' \prec (1+\varepsilon)L$ whp.

Sample edges according to a measure of importance, assign weights to make estimate unbiased

Sparsification



Sparsification



1. Spectral sparsification

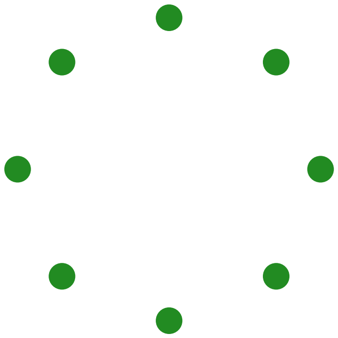
2. **Streaming model of computation**

3. Dynamic streaming and linear sketching

4. Spectral sparsification via linear sketches

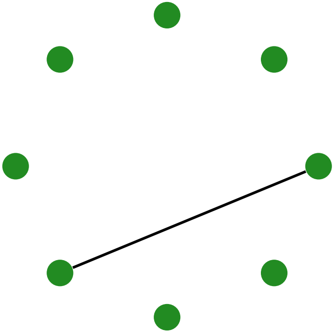
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



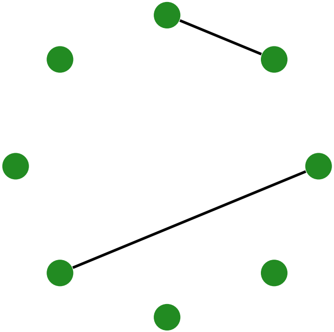
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



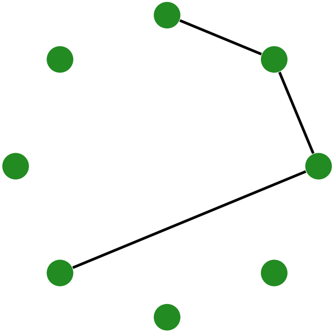
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



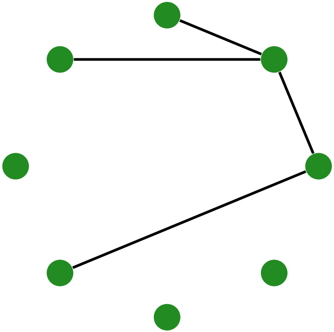
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



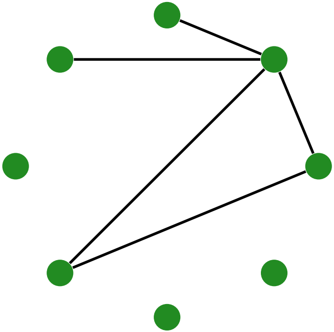
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



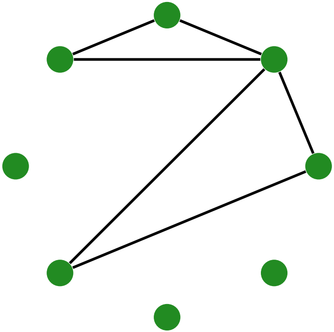
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



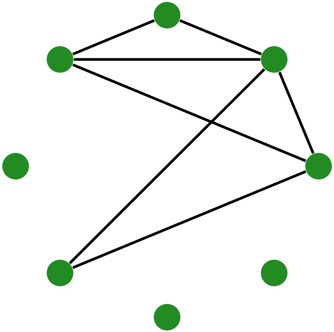
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



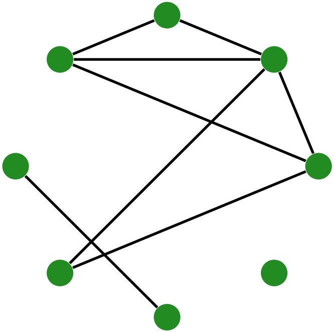
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



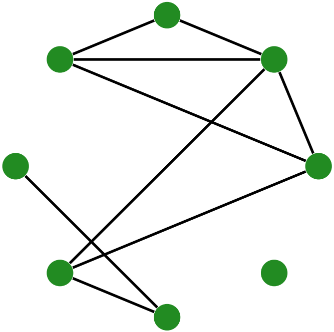
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



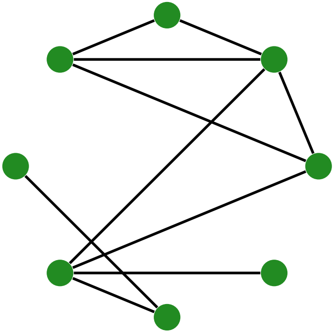
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



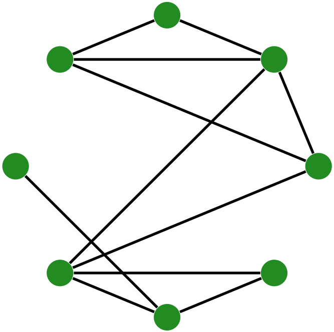
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



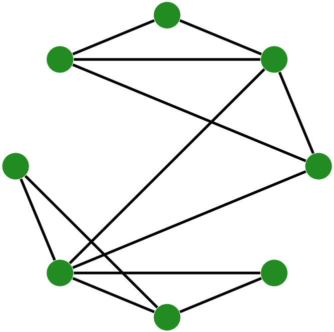
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



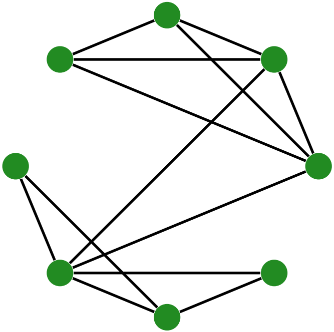
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



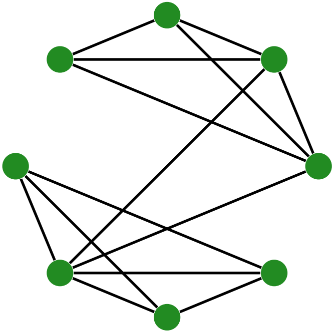
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



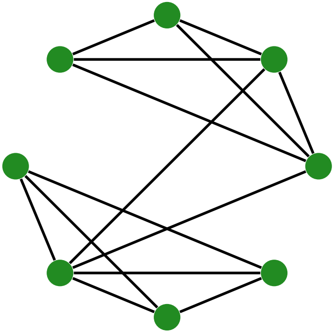
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



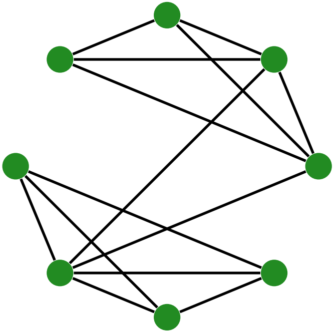
Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream



Streaming model

- streaming model: edges of G arrive in an arbitrary order in a stream;
- algorithm can only use $\widetilde{O}(n)$ space
- several passes over the stream **(ideally one pass)**



Insertion-only stream

These algorithms are streamable: just keep resparsifiying the graph as edges come in.

Ahn-Guha'09: $O(\frac{1}{\varepsilon^2} n \log^2 n)$ space for cut sparsifiers

Kelner-Levin'11: $O(\frac{1}{\varepsilon^2} n \log n)$ space for spectral sparsifiers

These algorithms are streamable: just keep resparsifiying the graph as edges come in.

Ahn-Guha'09: $O(\frac{1}{\varepsilon^2} n \log^2 n)$ space for cut sparsifiers

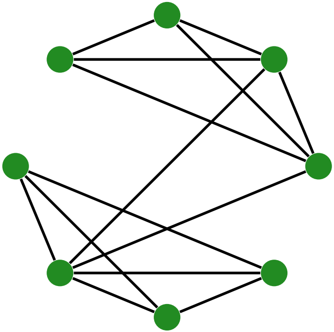
Kelner-Levin'11: $O(\frac{1}{\varepsilon^2} n \log n)$ space for spectral sparsifiers

Many modern networks evolve over time, edges both inserted and deleted

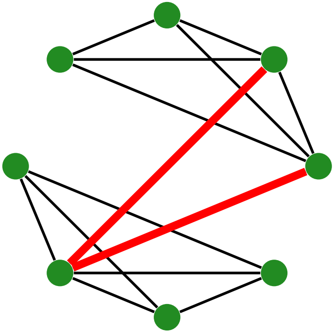
Construct sparsifiers in dynamic streams in small space?

1. Spectral sparsification
2. Streaming model of computation
3. **Dynamic streaming and linear sketching**
4. Spectral sparsification via linear sketches (main result)

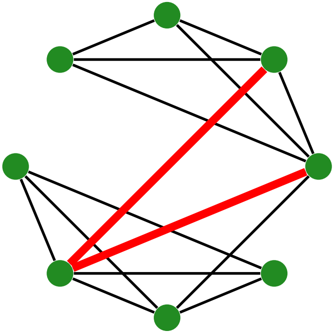
What if we have deletions?



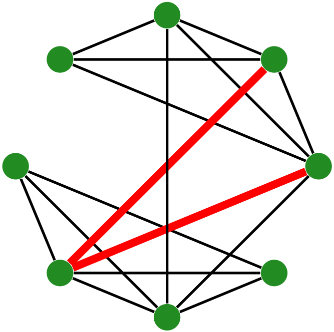
What if we have deletions?



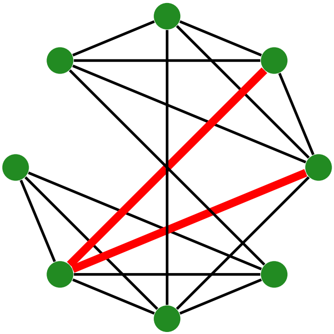
What if we have deletions?



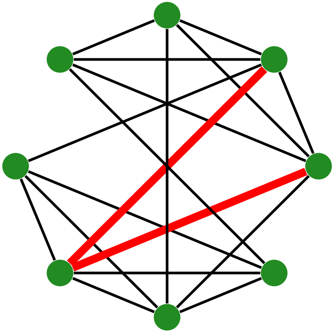
What if we have deletions?



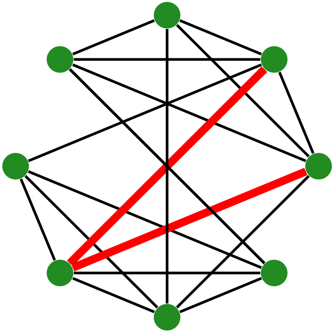
What if we have deletions?



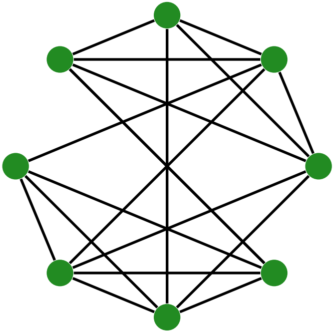
What if we have deletions?



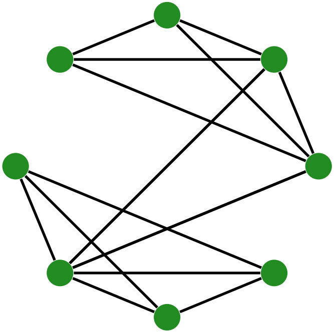
What if we have deletions?



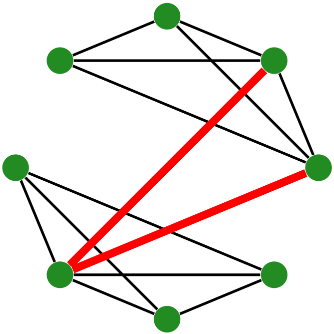
What if we have deletions?



What if we have deletions?



What if we have deletions?



Very different algorithms are needed...

Linear sketching

Classical data stream application: approximating frequency moments.

0	0	0	0	0	0	0	0	0	0	0

Goal: approximate $||x||_2^2 = \sum_i x_i^2$ using $\ll n$ space

Linear sketching

Classical data stream application: approximating frequency moments.

0	0	**1**	0	0	0	0	0	0	0	0

Goal: approximate $||x||_2^2 = \sum_i x_i^2$ using $\ll n$ space

Linear sketching

Classical data stream application: approximating frequency moments.

0	0	1	0	**1**	0	0	0	0	0	0

Goal: approximate $||x||_2^2 = \sum_i x_i^2$ using $\ll n$ space

Linear sketching

Classical data stream application: approximating frequency moments.

0	0	1	0	**2**	0	0	0	0	0	0

Goal: approximate $||x||_2^2 = \sum_i x_i^2$ using $\ll n$ space

Linear sketching

Classical data stream application: approximating frequency moments.

0	0	1	0	**2**	0	0	0	0	0	0

Goal: approximate $||x||_2^2 = \sum_i x_i^2$ using $\ll n$ space

Maintain $x^T v_i = 1, \ldots, O(1/\varepsilon^2)$ for random Gaussians $v_i \in \mathbb{R}^n$.
Output average of $(x^T v_i)^2$.

Linear sketching

Classical data stream application: approximating frequency moments.

0	0	1	0	**2**	0	0	0	0	0	0

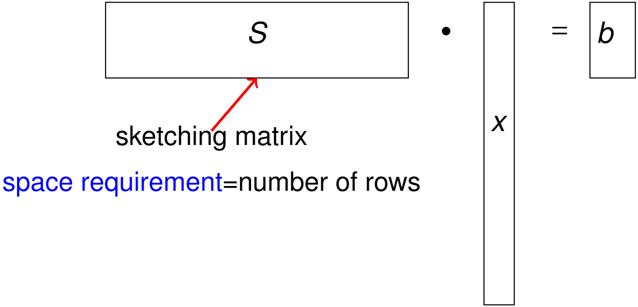
Goal: approximate $||x||_2^2 = \sum_i x_i^2$ using $\ll n$ space

Maintain $x^T v_i = 1, \ldots, O(1/\varepsilon^2)$ for random Gaussians $v_i \in \mathbb{R}^n$.
Output average of $(x^T v_i)^2$.

$(1 \pm \varepsilon)$-approximation with $O(\frac{1}{\varepsilon^2} \log n)$ space

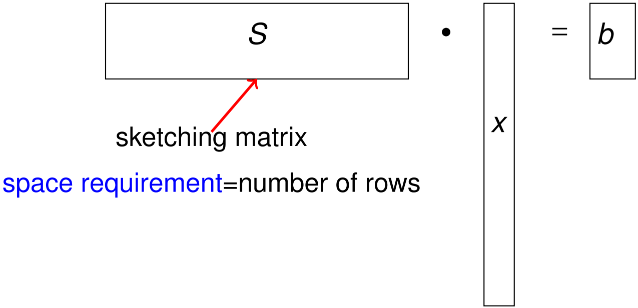
Linear sketching

Take (randomized) linear measurements of the input



Linear sketching

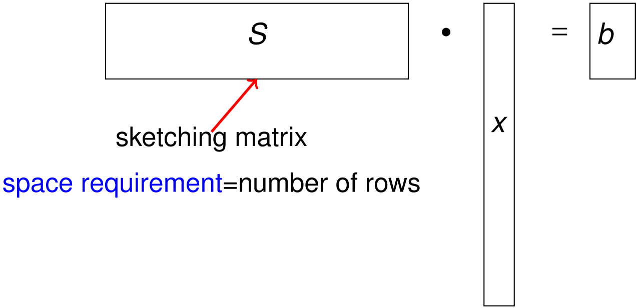
Take (randomized) linear measurements of the input



Can get $(1 \pm \varepsilon)$-approximation to $||x||^2$ with $\frac{1}{\varepsilon^2}\text{poly}(\log n)$ rows

Linear sketching

Take (randomized) linear measurements of the input



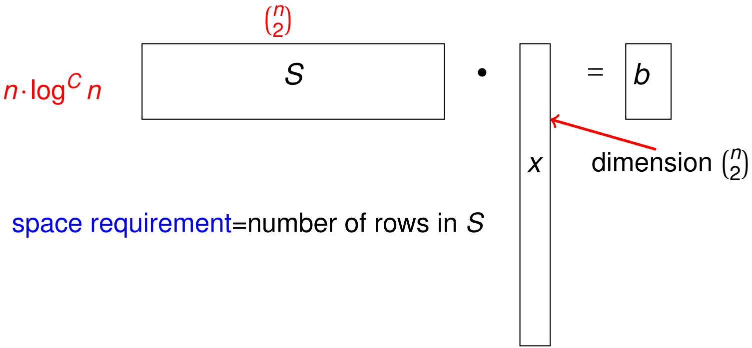
Can get $(1 \pm \varepsilon)$-approximation to $||x||^2$ with $\frac{1}{\varepsilon^2}\text{poly}(\log n)$ rows

Easy to maintain linear sketches in the (dynamic) streaming model

Graph sketching

Represent adjacency matrix of input graph G as a vector of dimension $\binom{n}{2}$, sketch the vector.

Ahn-Guha-McGregor'SODA12 – connectivity in $n \cdot poly(\log n)$ space.

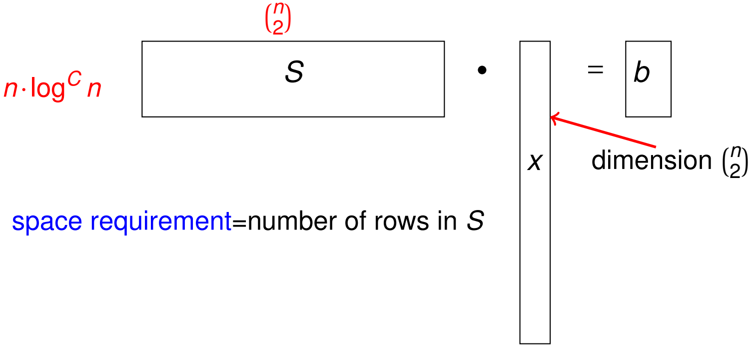


space requirement=number of rows in S

Graph sketching

Represent adjacency matrix of input graph G as a vector of dimension $\binom{n}{2}$, sketch the vector.

Ahn-Guha-McGregor'SODA12 – connectivity in $n \cdot poly(\log n)$ space.



Sketch the adjacency matrix, then reconstruct edges of a sparsifier from the sketch?

	Cut sparsifiers:	Spectral sparsifiers:
Streaming	Ahn-Guha'09	Kelner-Levin'11
Dynamic streaming	Ahn-Guha-McGregor'12 Goel-Kapralov-Post'12 $O(\frac{1}{\varepsilon^2} n\text{poly}(\log n))$ space	Ahn-Guha-McGregor'14 Kapralov-Woodruff'14 $\widetilde{O}(\frac{1}{\varepsilon^2} n^{5/3})$ space $O(\text{poly}(\frac{1}{\varepsilon}) n^{1+o(1)})$ space, two passes

	Cut sparsifiers:	Spectral sparsifiers:
Streaming	Ahn-Guha'09	Kelner-Levin'11
Dynamic streaming	Ahn-Guha-McGregor'12 Goel-Kapralov-Post'12 $O(\frac{1}{\epsilon^2} n \mathrm{poly}(\log n))$ space	Ahn-Guha-McGregor'14 Kapralov-Woodruff'14 $\widetilde{O}(\frac{1}{\epsilon^2} n^{5/3})$ space $O(\mathrm{poly}(\frac{1}{\epsilon}) n^{1+o(1)})$ space, two passes

Theorem (K.-Lee-Musco-Musco-Sidford'14)

There exists a single-pass streaming algorithm that constructs a spectral sparsifier of a graph given as a dynamic stream of edges using $\widetilde{O}(\frac{1}{\epsilon^2} n \mathit{poly}(\log n))$ space and $\mathit{poly}(n)$ runtime.

	Cut sparsifiers:	Spectral sparsifiers:
Streaming	Ahn-Guha'09	Kelner-Levin'11
Dynamic streaming	Ahn-Guha-McGregor'12 Goel-Kapralov-Post'12 $O(\frac{1}{\varepsilon^2} n \text{poly}(\log n))$ space	Ahn-Guha-McGregor'14 Kapralov-Woodruff'14 $\widetilde{O}(\frac{1}{\varepsilon^2} n^{5/3})$ space $O(\text{poly}(\frac{1}{\varepsilon}) n^{1+o(1)})$ space, two passes

Theorem (K.-Lee-Musco-Musco-Sidford'14)

There exists a single-pass streaming algorithm that constructs a spectral sparsifier of a graph given as a dynamic stream of edges using $\widetilde{O}(\frac{1}{\varepsilon^2} n \, poly(\log n))$ space and $poly(n)$ runtime.

Essentially optimal space complexity, oblivious compression scheme

1. Spectral sparsification
2. Streaming model of computation
3. Dynamic streaming and linear sketching
4. **Spectral sparsification via linear sketches**

Constructing spectral sparsifiers

Theorem (Spielman-Srivastava'09)

Let $G = (V, E)$ be an undirected graph. Let G' be obtained by including every edge $e \in E$ independently with probability proportional to its effective resistance:

$$p_e \geq \min\{1, \frac{C \log n}{\varepsilon^2} R_e\}.$$

Assign weight $1/p_e$ if sampled. Then $(1-\varepsilon)G \prec G' \prec (1+\varepsilon)G$ whp.

Sample edges according to a measure of importance, assign weights to make estimate unbiased

Note: edges e with resistance $R_e = \Omega(1/\log n)$ included with probability 1

Constructing spectral sparsifiers offline

Maintain: sketch $S \cdot B$ of the incidence matrix B

Step 1. Compute sampling probabilities p_e for each $e \in E$

Step 2. Sample edges independently with probability p_e, give weight $1/p_e$.

Constructing spectral sparsifiers offline

Maintain: sketch $S \cdot B$ of the incidence matrix B

Step 1. Compute sampling probabilities p_e for each $e \in E$

[Q] How? We do not know which edges are present in the graph...

Step 2. Sample edges independently with probability p_e, give weight $1/p_e$.

[Q] Sample from the sketch?

Refining a sparsifier

Goal: design a sketch S that allows sampling edges of G according to effective resistance given

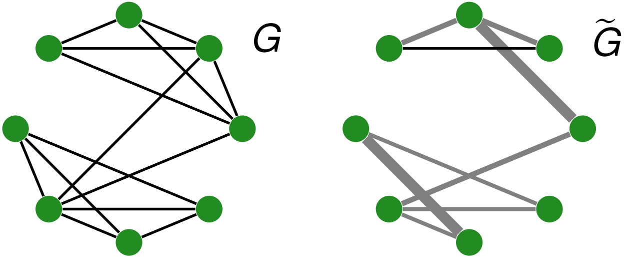
- $S \cdot B$ (sketch of edge incidence matrix)

Refining a sparsifier

Goal: design a sketch S that allows sampling edges of G according to effective resistance given

- $S \cdot B$ (sketch of edge incidence matrix)
- crude constant factor spectral sparsifier $\widetilde{G}$

$$\frac{1}{C} \cdot L \prec \widetilde{L} \prec L$$



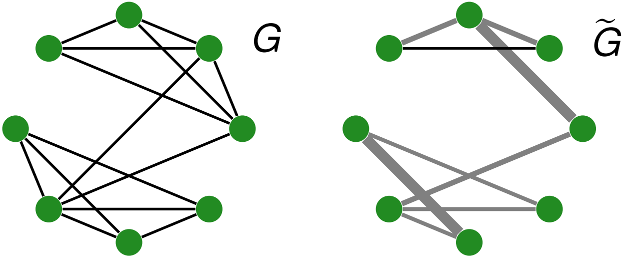
Construct a $1+\varepsilon$ sparsifier G' of G

Refining a sparsifier

Goal: design a sketch S that allows recovery of high resistance ($\geq 1/\log n$) edges of G given

- $S \cdot B$ (sketch of edge incidence matrix)
- crude constant factor spectral sparsifier $\widetilde{G}$

$$\frac{1}{C} \cdot L \prec \widetilde{L} \prec L$$

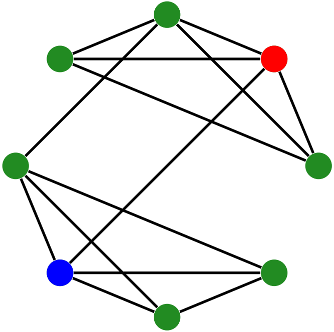


Construct a $1+\varepsilon$ sparsifier G' of G

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).

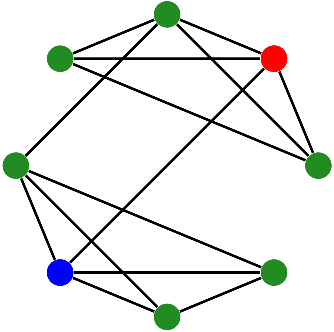


Inject current at u, take out at v.

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



Inject current at u, take out at v.

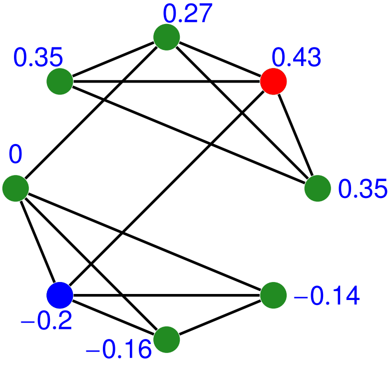
$\phi = L^+ b_{uv}$=vertex potentials

$$b_{uv} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



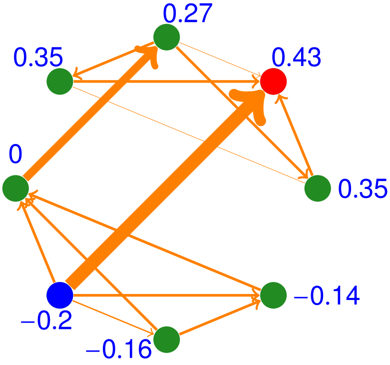
Inject current at u, take out at v.

$\phi = L^+ b_{uv}$=vertex potentials

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



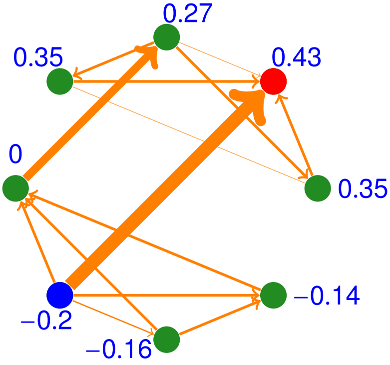
Inject current at u, take out at v.

$\phi = L^+ b_{uv}$=vertex potentials

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



Inject current at u, take out at v.

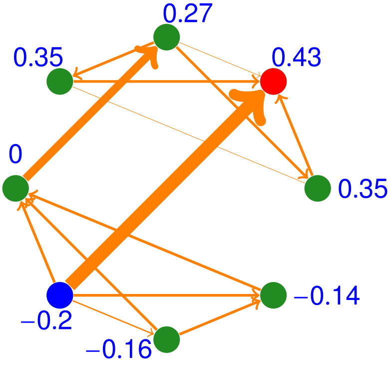
$\phi = L^+ b_{uv}$=vertex potentials

$f_{xy} = \phi_y - \phi_x = b_{xy}^T L^+ b_{uv}$

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



Inject current at u, take out at v.

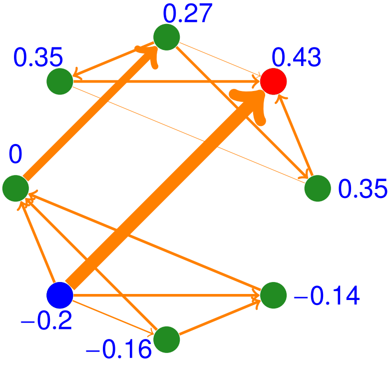
$\phi = L^+ b_{uv}$=vertex potentials

$f = B\phi$=currents on edges

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



Inject current at u, take out at v.

$\phi = L^+ b_{uv}$=vertex potentials

$f = B\phi$=currents on edges

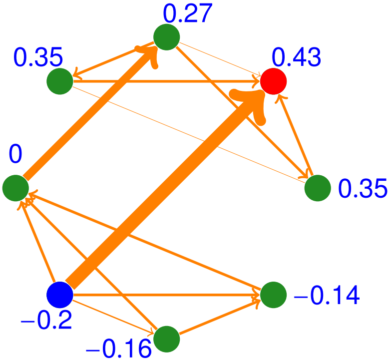
We have

$$R_e = \frac{f_e^2}{||f||^2}.$$

Effective resistance

$$R_{uv} = b_{uv}^T L^+ b_{uv}$$

Note: defined for any pair (u, v).



Inject current at u, take out at v.

$\phi = L^+ b_{uv}$=vertex potentials

$f = B\phi$=currents on edges

We have

$$R_e = \frac{f_e^2}{||f||^2}.$$

R_{uv}= fraction of $||f||_2^2$ contributed by $e = (u, v)$

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

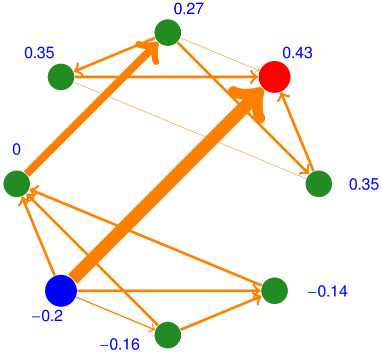
- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = L^{+} b_{uv}$ – vertex potentials

Compute $f = B\phi$=currents on edges

Check if

$$R_e = \frac{f_e^2}{||f||^2} = \Omega(1/\log n).$$

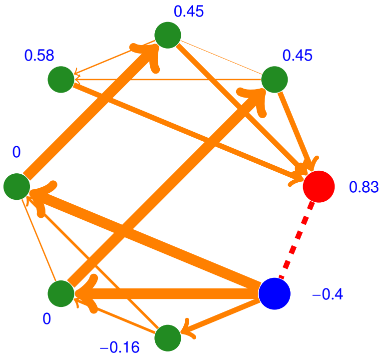
0	0	0.01	0	**200**	1	0	2	0	0	0

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = L^{+} b_{uv}$ – vertex potentials

Compute $f = B\phi$=currents on edges

Check if

$$R_e = \frac{f_e^2}{||f||^2} = \Omega(1/\log n).$$

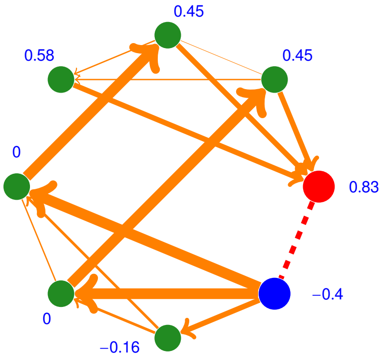
0	0	0.01	0	200	1	0	2	0	**0**	0

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = \mathbf{L}^{+} b_{uv}$ – vertex potentials

Compute $f = \mathbf{B}\phi$=currents on edges

Check if

$$R_e = \frac{f_e^2}{||f||^2} = \Omega(1/\log n).$$

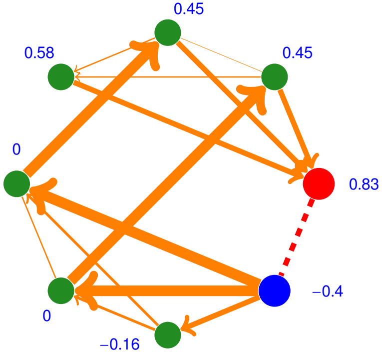
0	0	0.01	0	200	1	0	2	0	**0**	0

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = \widetilde{\mathbf{L}}^{+} b_{uv}$ – vertex potentials

Compute $\Delta = \mathbf{B}\phi$=potential differences

Check if

$$\widetilde{R}_e = \frac{\Delta_e^2}{||\Delta||^2} = \Omega(1/(C \log n)).$$

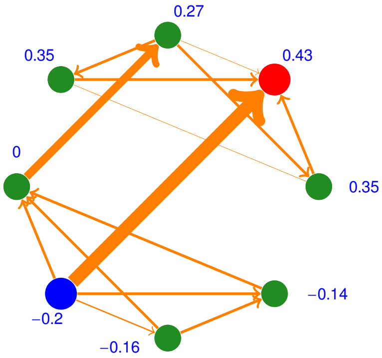
0	0	0.01	0	200	1	0	2	0	**0**	0

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = \widetilde{\mathbf{L}}^{+} b_{uv}$ – vertex potentials

Compute $\Delta = \mathbf{B}\phi$=potential differences

Check if

$$\widetilde{R}_e = \frac{\Delta_e^2}{||\Delta||^2} = \Omega(1/(C \log n)).$$

0	0	0.01	0	**200**	1	0	2	0	0	0

Linear sketching and sparse recovery

0	0	0.01	0	**200**	1	0	2	0	0	0

Let y be a vector of reals. Then $i \in [n]$ is an ℓ_2-heavy hitter if

$$y_i^2 \geq \eta ||y||_2^2.$$

Lemma (ℓ_2-heavy hitters)

For any $\eta > 0$ there exists a (randomized) sketch in dimension $\frac{1}{\eta} poly(\log n)$ from which one reconstruct all η-heavy hitters. The recovery works in time $O(\frac{1}{\eta} \cdot poly(\log n))$.

Linear sketching and sparse recovery

0	0	**0.01**	0	**200**	1	0	2	0	0	0

Let y be a vector of reals. Then $i \in [n]$ is an ℓ_2-heavy hitter if

$$y_i^2 \geq \eta ||y||_2^2.$$

Lemma (ℓ_2-heavy hitters)

For any $\eta > 0$ there exists a (randomized) sketch in dimension $O(\frac{1}{\eta} \log n)$ from which one reconstruct all η-heavy hitters. The recovery works in time $O(\frac{1}{\eta} \cdot poly(\log n))$.

Linear sketching and sparse recovery

Need to recover 'heavy' coordinates of

$$S\Delta := (SB)\phi = S \cdot \begin{bmatrix} \phi_1 - \phi_2 \\ \phi_2 - \phi_3 \\ 0 \\ \phi_4 - \phi_3 \\ \phi_3 - \phi_6 \\ 0 \\ \phi_3 - \phi_1 \\ 0 \\ \vdots \end{bmatrix}$$

A coordinate $e \in \binom{n}{2}$ is heavy if $\Delta_e^2 = \Omega(||\Delta||_2^2/(C \log n))$

This is the ℓ_2 heavy hitters problem!

Problem: we do not know Δ in advance!

Sketching the edge incidence matrix

$$S \cdot B = S \cdot \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix}$$

Apply ℓ_2-heavy hitters sketch S to every column $b_u, u \in V$ of B

Store the n sketches, $n \cdot \log^C n$ space.

Sketching the edge incidence matrix

$$S \cdot B = S \cdot \begin{bmatrix} \mathbf{1} & -1 & 0 & 0 & 0 & 0 \\ \mathbf{0} & 1 & -1 & 0 & 0 & 0 \\ \mathbf{0} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{-1} & 0 & 1 & 0 & 0 & 0 \\ \mathbf{0} & 0 & 1 & 0 & 0 & -1 \\ \mathbf{0} & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix}$$

Apply ℓ_2-heavy hitters sketch S to every column $b_u, u \in V$ of B

Store the n sketches, $n \cdot \log^C n$ space.

Sketching the edge incidence matrix

$$S \cdot B = S \cdot \begin{bmatrix} 1 & \mathbf{-1} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & -1 & 0 & 0 & 0 \\ 0 & \mathbf{0} & 0 & 0 & 0 & 0 \\ -1 & \mathbf{0} & 1 & 0 & 0 & 0 \\ 0 & \mathbf{0} & 1 & 0 & 0 & -1 \\ 0 & \mathbf{0} & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix}$$

Apply ℓ_2-heavy hitters sketch S to every column $b_u, u \in V$ of B

Store the n sketches, $n \cdot \log^C n$ space.

Sketching the edge incidence matrix

$$S \cdot B = S \cdot \begin{bmatrix} 1 & -1 & \mathbf{0} & 0 & 0 & 0 \\ 0 & 1 & \mathbf{-1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 & 0 & 0 \\ -1 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & -1 \\ 0 & 0 & \mathbf{0} & 0 & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix}$$

Apply ℓ_2-heavy hitters sketch S to every column $b_u, u \in V$ of B

Store the n sketches, $n \cdot \log^C n$ space.

Sketching the edge incidence matrix

$$S \cdot B = S \cdot \begin{bmatrix} 1 & -1 & 0 & \mathbf{0} & 0 & 0 \\ 0 & 1 & -1 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0} & 0 & 0 \\ -1 & 0 & 1 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 1 & \mathbf{0} & 0 & -1 \\ 0 & 0 & 0 & \mathbf{0} & 0 & 0 \\ \vdots & & & & & \vdots \end{bmatrix}$$

Apply ℓ_2-heavy hitters sketch S to every column $b_u, u \in V$ of B

Store the n sketches, $n \cdot \log^C n$ space.

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?

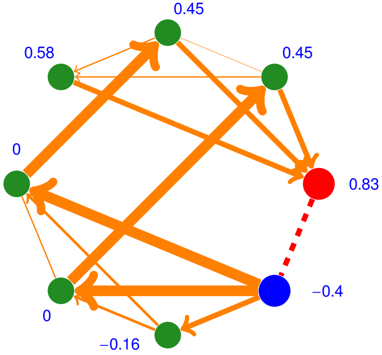
Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?

Compute $\phi = \widetilde{\mathbf{L}}^{+} b_{uv}$ – vertex potentials

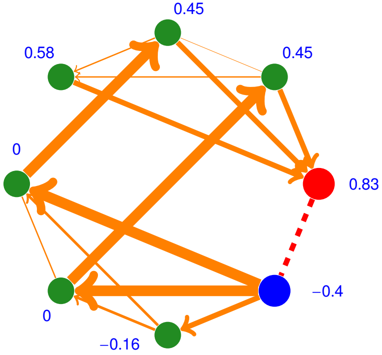


Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = \widetilde{\mathbf{L}}^{+} b_{uv}$ – vertex potentials

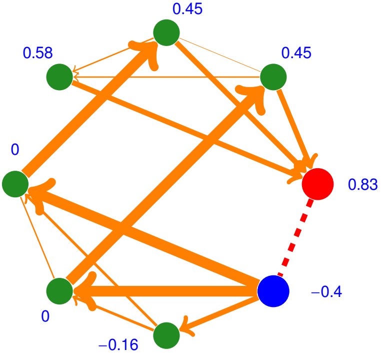
Compute $S \cdot \Delta = \mathbf{S} \cdot \mathbf{B}\phi$=potential differences

Given:

- a sketch $S \cdot B$ of G
- crude sparsifier $\widetilde{G}$
- pair $(u, v) \in V \times V$

Need:

- is (u, v) an edge in G of resistance $\Omega(1/\log n)$?



Compute $\phi = \widetilde{\mathbf{L}}^{+} b_{uv}$ – vertex potentials

Compute $S \cdot \Delta = \mathbf{S} \cdot \mathbf{B} \phi$=potential differences

Check if

$$\widetilde{R}_e = \frac{\Delta_e^2}{||\Delta||^2} = \Omega(1/(C \log n))$$

using heavy-hitters sketch S

0	0	0.01	0	200	1	0	2	0	**0**	0

What about an edge of resistance $r \approx 2^{-j} = o(1)$? it only contributes a $\approx 2^{-j}$ fraction of ℓ_2 mass...

What about an edge of resistance $r \approx 2^{-j} = o(1)$? it only contributes a $\approx 2^{-j}$ fraction of ℓ_2 mass...

Need to recover 'heavy' coordinates of

$$S\Delta := (SB)\phi = S \cdot \begin{bmatrix} \phi_1 - \phi_2 \\ \phi_2 - \phi_3 \\ 0 \\ \boldsymbol{\phi_4 - \phi_3} \\ \phi_3 - \phi_6 \\ 0 \\ \phi_3 - \phi_1 \\ 0 \\ \vdots \end{bmatrix}$$

What about an edge of resistance $r \approx 2^{-j} = o(1)$? it only contributes a $\approx 2^{-j}$ fraction of ℓ_2 mass...

Need to recover 'heavy' coordinates of

$$S\Delta := (SB)\phi = S \cdot \begin{bmatrix} \phi_1 - \phi_2 \\ \phi_2 - \phi_3 \\ 0 \\ \boldsymbol{\phi_4 - \phi_3} \\ \phi_3 - \phi_6 \\ 0 \\ \phi_3 - \phi_1 \\ 0 \\ \vdots \end{bmatrix}$$

Sample edges with probability 2^{-j}

If edge (a,b) is in G and is sampled, it contributes $\Omega(\frac{1}{C \log n})$ fraction of mass whp – can recover.

What about an edge of resistance $r \approx 2^{-j} = o(1)$? it only contributes a $\approx 2^{-j}$ fraction of ℓ_2 mass...

Need to recover 'heavy' coordinates of

$$S\Delta := (SB)\phi = S \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ \phi_4 - \phi_3 \\ \phi_3 - \phi_6 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

Sample edges with probability 2^{-j}

If edge (a, b) is in G and is sampled, it contributes $\Omega(\frac{1}{C \log n})$ fraction of mass whp – can recover.

Store sketches of subsampled edge incidence matrix:

$$S\Pi_j B, j = 0, \ldots, \log_2 n.$$

Π_j is a diagonal matrix with Bernoulli$(0/1, 2^{-j})$ entries

REFINESPARSIFIER$(G, \widetilde{G}, \varepsilon, c)$

For $e = (a, b) \in \binom{V}{2}$

$\widetilde{R}_e \leftarrow b_e^T \widetilde{L}^+ b_e$ ▷ resistance in $\widetilde{G}$

Round: $\widetilde{R}_e \approx 2^{-j}$ ▷ determine sampling level

$x_e \leftarrow \widetilde{L}^+ b_e$

If TESTEDGE$(S\Pi_j B, x_e, e)$ **then**
add e to sparsifier with weight 2^j

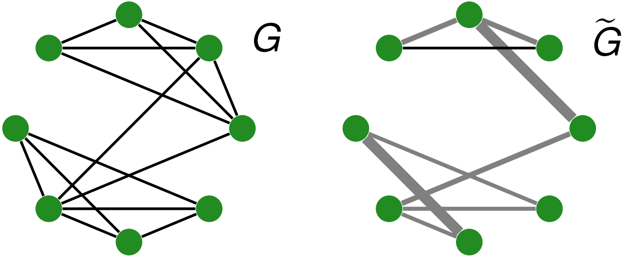
Repeat (C/ε^2) **times, take union**

Refining a sparsifier

Designed a sketch S that allows sampling edges of G according to effective resistance given

- $S \cdot B$ (sketch of edge incidence matrix)
- crude constant factor spectral sparsifier $\widetilde{G}$

$$\frac{1}{C} \cdot L \prec \widetilde{L} \prec L$$

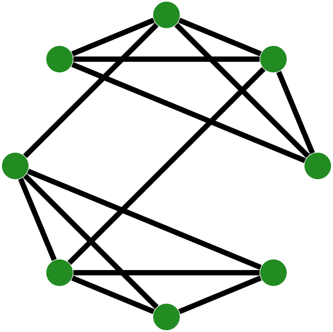


Chain of coarse sparsifiers

Approach of Miller and Peng in Iterative approaches to row sampling

Add a weighted complete graph to G:

$$G(\lambda) = G + \lambda K_n$$

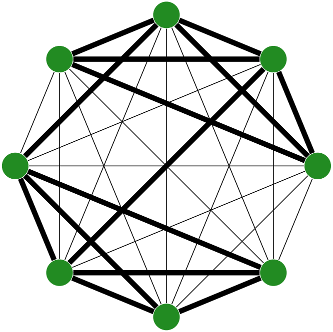


Chain of coarse sparsifiers

Approach of Miller and Peng in Iterative approaches to row sampling

Add a weighted complete graph to G:

$$G(\lambda) = G + \lambda K_n$$

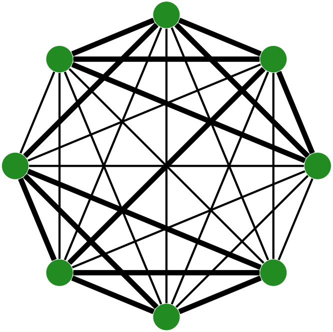


Chain of coarse sparsifiers

Approach of Miller and Peng in Iterative approaches to row sampling

Add a weighted complete graph to G:

$$G(\lambda) = G + \lambda K_n$$

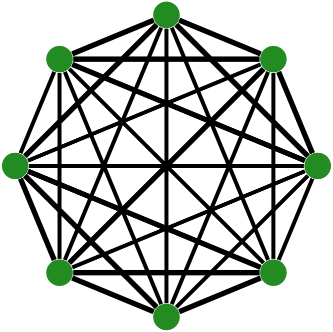


Chain of coarse sparsifiers

Approach of Miller and Peng in Iterative approaches to row sampling

Add a weighted complete graph to G:

$$G(\lambda) = G + \lambda K_n$$

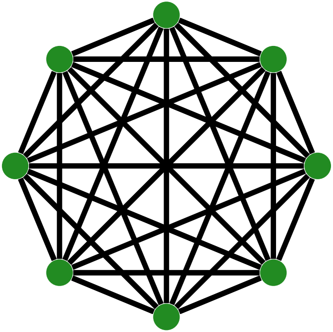


Chain of coarse sparsifiers

Approach of Miller and Peng in Iterative approaches to row sampling

Add a weighted complete graph to G:

$$G(\lambda) = G + \lambda K_n$$



$$L(\lambda) = L + \lambda \cdot nI^*$$

Nonzero eigenvalues of G are between n and $8/n^2$, so

- K_n is a C-spectral approximation to $G(1)$
- $G(1/\text{poly}(n))$ approximates G well spectrally.

Consider powers of 2 from 1 to $1/\text{poly}(n)$.

Two adjacent graphs in the chain are similar:

$$\frac{1}{2}G(\lambda) \prec G(\lambda/2) \prec G(\lambda)$$

This is exactly what we need for RefineSparsifier...

Final algorithm

$$\widetilde{G}_{1/2} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/2),\ \ K_n, \varepsilon, 3)$$

Final algorithm

$$\widetilde{G}_{1/2} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/2),\ \ K_n, \varepsilon, 3)$$
$$\widetilde{G}_{1/4} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/4), \widetilde{G}_{1/2}, \varepsilon, 3)$$

Final algorithm

$$\widetilde{G}_{1/2} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/2),\ \ K_n, \varepsilon, 3)$$
$$\widetilde{G}_{1/4} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/4), \widetilde{G}_{1/2}, \varepsilon, 3)$$
$$\widetilde{G}_{1/8} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/8), \widetilde{G}_{1/4}, \varepsilon, 3)$$
$$\vdots$$

Final algorithm

$\widetilde{G}_{1/2} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/2),\ \ K_n, \varepsilon, 3)$

$\widetilde{G}_{1/4} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/4), \widetilde{G}_{1/2}, \varepsilon, 3)$

$\widetilde{G}_{1/8} \leftarrow \textsc{RefineSparsifier}(S \cdot G(1/8), \widetilde{G}_{1/4}, \varepsilon, 3)$

$\vdots$

return $\widetilde{G}_{1/\mathrm{poly}(n)}$

Final algorithm

$\widetilde{G}_{1/2} \leftarrow \text{RefineSparsifier}(S \cdot G(1/2),\ \ K_n, \varepsilon, 3)$

$\widetilde{G}_{1/4} \leftarrow \text{RefineSparsifier}(S \cdot G(1/4), \widetilde{G}_{1/2}, \varepsilon, 3)$

$\widetilde{G}_{1/8} \leftarrow \text{RefineSparsifier}(S \cdot G(1/8), \widetilde{G}_{1/4}, \varepsilon, 3)$

$\vdots$

return $\widetilde{G}_{1/\text{poly}(n)}$

Space requirement

- $O(\log n)$ sampling levels, $O(\frac{1}{\varepsilon^2} \log n)$ repetitions
- $O(\log n)$ long chain of coarse sparsifiers
- an ℓ_2-heavy hitters sketch of $O(\text{poly}(\log n))$ size for each node.

Summary

- AMS sketch (approximating $||x||_2^2$)
- Heavy hitters (CountSketch)
- ℓ_0 samplers
- Graph connectivity
- Graph sparsification

Summary

- AMS sketch (approximating $||x||_2^2$)
- Heavy hitters (CountSketch)
- ℓ_0 samplers
- Graph connectivity
- Graph sparsification

Which other graph problems admit sketching solutions?