Lecture 8: Graph Sparsification

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science

rafael.oliveira.teaching@gmail.com

October 5, 2020

Overview

- Introduction
  - Why Sparsify?
  - Warm-up Problem
- Main Problem
  - Graph Sparsification
- Acknowledgements

Why do we sparsify?

Often times graph algorithms for graphs $G(\overset{\text{vertices}}{V}, \overset{\text{edges}}{E})$ have runtimes which depend on $|E|$. If the graph is dense, i.e. $|E| = \omega(n^{1+c})$ then this may be *too slow*.

$|V| = n \qquad |E| = m$

Want something that is nearly-linear

$$O(n \cdot \underbrace{\text{polylog}(n)}_{\log^{\beta}(n)}) \qquad \beta > 0$$

Approximate answers

Why do we sparsify?

To sparsify is to make $|\tilde{E}|$ nearly linear

Often times graph algorithms for graphs $G(V, E)$ have runtimes which depend on $|E|$. If the graph is dense, i.e. $\underline{|E| = \omega(n^{1+c})}$ then this may be *too slow*.

- Used as primitives in many other algorithms (for instance, max-flow, sparsest cut, etc.)

Why do we sparsify?

Often times graph algorithms for graphs $G(V, E)$ have runtimes which depend on $|E|$. If the graph is dense, i.e. $|E| = \omega(n^{1+c})$ then this may be *too slow*.

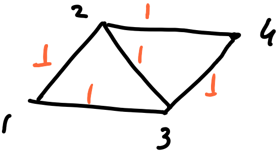
- Used as primitives in many other algorithms (for instance, max-flow, sparsest cut, etc.)
- Applications in network connectivity

Graph Cuts

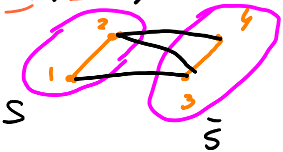
Definition (Graph Cut)

If $G(\underline{V}, \underline{E}, \underline{w})$ is a weighted graph, a *cut* is a partition of the vertices into two non-empty sets $V = S \sqcup \underset{:= V\setminus S}{\overline{S}}$. The *value* of a cut is the quantity

$$\underline{w(S,\overline{S})} := \sum_{e\in E(S,\overline{S})} w_e.$$



$(\{1,2\},\{3,4\})$



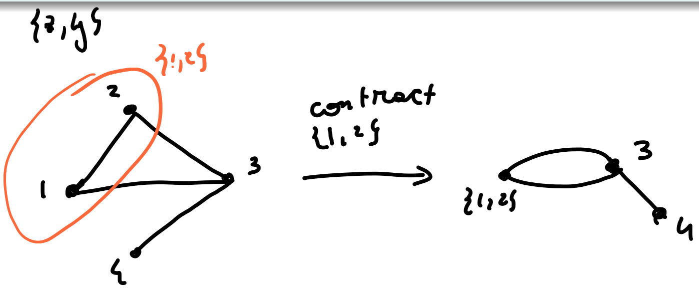
$w(S,\bar{S}) = 3 = |E(S,\bar{S})|$

Contraction of Edges

Definition (Edge Contraction)

Let $G(V, E)$ be a graph. If $e = \{u, v\} \in E$ is an edge of G, then the *contraction* of e is a new graph $H(V \cup \underline{\{z\}} \setminus \underline{\{u, v\}}, \underline{F})$ where we replace the vertices u, v by *one* vertex z, and any edge $\{u, x\} =: f \in E \setminus \{e\}$ is replaced by $\{z, x\} \in F$.

$\{v, y\}$



Randomized Minimum Cut

- **Input:** undirected unweighted graph $G(V, E)$
- **Output:** minimum cut $(S, \overline{S})$, with high probability

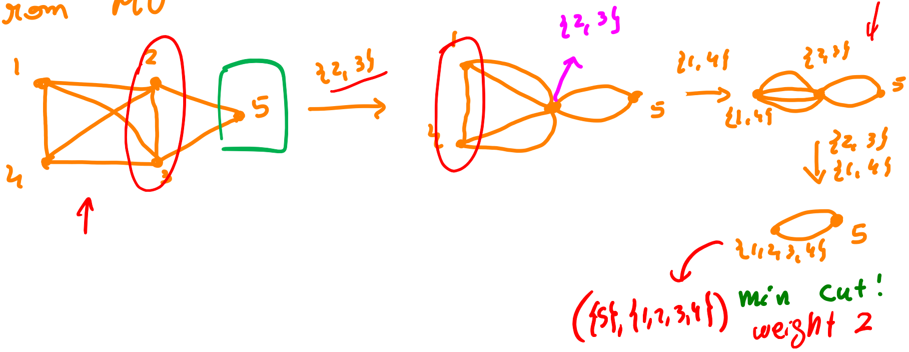
Randomized Minimum Cut

- **Input:** undirected unweighted graph $G(V, E)$
- **Output:** minimum cut $(S, \overline{S})$, with high probability
- While there are more than 2 vertices in the graph:
    - Pick uniformly random edge and contract it

Randomized Minimum Cut

- **Input:** undirected unweighted graph $G(V, E)$
- **Output:** minimum cut $(S, \overline{S})$, with high probability
- While there are more than 2 vertices in the graph:
  - Pick uniformly random edge and contract it
- Output the two subsets encoded by the two remaining vertices.

From MU



({5}, {1, 2, 3, 4}) min cut!
weight 2

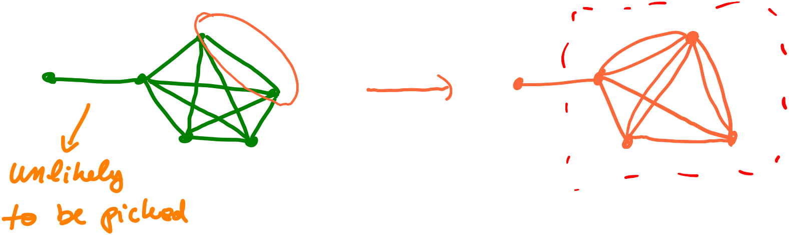
Analysis

Why does this work?

Analysis

Why does this work?

Intuition: picking a random edge uniformly at random “favours” *small cuts* (i.e. preserves them) with higher probability.



Analysis

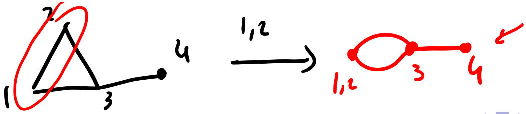
Why does this work?

Intuition: picking a random edge uniformly at random “favours” *small cuts* (i.e. preserves them) with higher probability.

Remark

The value of the minimum cut only increases or stays the same after contraction.

After contracting 1 and 2, only cuts that remain are cuts $(S, \bar{S})$ where $\underline{1,2} \in S$ (and these have same value).



Analysis

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

Analysis

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

$= \omega(S, \bar{S})$

- Let $(S, \overline{S})$ be a minimum cut, and $k := |E(S, \overline{S})|$. If we never contract an edge from $\underline{E(S, \overline{S})}$, the algorithm succeeds.

Analysis

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

- Let $(S, \overline{S})$ be a minimum cut, and $k := |E(S, \overline{S})|$. If we never contract an edge from $E(S, \overline{S})$, the algorithm succeeds.
- Probability that an edge from $E(S, \overline{S})$ is contracted in the i^{th} iteration (conditioned on cut still alive)

Analysis

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

- Let $(S, \overline{S})$ be a minimum cut, and $k := |E(S, \overline{S})|$. If we never contract an edge from $E(S, \overline{S})$, the algorithm succeeds.
- Probability that an edge from $E(S, \overline{S})$ is contracted in the i^{th} iteration (conditioned on cut still alive)
- Each vertex is a cut, so each vertex has degree $\geq \underbrace{k}_{\text{size of min-cut}} \Rightarrow$

$(\{v\}, V-\{v\})$

$$\geq \frac{(n-i+1)\cdot k}{2} \quad \text{edges remain.}$$

vertices in graph after i-1 iterations

Analysis

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

- Let $(S, \overline{S})$ be a minimum cut, and $k := |E(S, \overline{S})|$. If we never contract an edge from $E(S, \overline{S})$, the algorithm succeeds.
- Probability that an edge from $E(S, \overline{S})$ is contracted in the i^{th} iteration (conditioned on cut still alive)
- Each vertex is a cut, so each vertex has degree $\geq k \Rightarrow$

$$\geq \boxed{\frac{(n-i+1) \cdot k}{2}} \text{ edges remain.}$$

- Contracting random edge, probability we kill cut $(S, \overline{S})$ is

$$= \underbrace{|E(S, \overline{S})|}_{\text{\#edges in my cut}} \cdot \frac{1}{(\#\text{ edges})} \leq k \cdot \frac{2}{(n-i+1) \cdot k} = \boxed{\frac{2}{n-i+1}}$$

Analysis

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

- Let $(S, \overline{S})$ be a minimum cut, and $k := |E(S, \overline{S})|$. If we never contract an edge from $E(S, \overline{S})$, the algorithm succeeds.
- Probability that an edge from $E(S, \overline{S})$ is contracted in the i^{th} iteration (conditioned on cut still alive)
- Each vertex is a cut, so each vertex has degree $\geq k \Rightarrow$

$$\geq \frac{(n-i+1)\cdot k}{2} \quad \text{edges remain.}$$

$1 - \frac{2}{n-i+1}$

survive

- Contracting random edge, probability we kill cut $(S, \overline{S})$ is

$$= |E(S, \overline{S})| \cdot \frac{1}{(\#\text{ edges})} \leq k \cdot \frac{2}{(n-i+1)\cdot k} = \frac{2}{n-i+1}$$

- $\Pr[(S, \overline{S}) \text{ survives}] \geq \underline{(1 - 2/n)} \cdot \underline{(1 - \overset{2/n-1}{\cancel{3/n}})} \cdots (1 - 2/3) = \underline{2/n(n-1)}$

Hmmmmm, this is not with high probability...

- To improve success probability, repeat this randomized procedure t times (for which t?)

Hmmmmm, this is not with high probability...

- To improve success probability, repeat this randomized procedure t times (for which t?)
- If we repeat for t times, failure probability is

$$\leq \left(\underbrace{1 - \frac{2}{n(n-1)}}_{\text{Probability that my iteration fails}}\right)^t$$

Hmmmmm, this is not with high probability...

- To improve success probability, repeat this randomized procedure t times (for which t?)
- If we repeat for t times, failure probability is

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^t$$

- setting $t = 2n(n-1)$ then

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^t \leq \exp\left(-\frac{2t}{n(n-1)}\right) = e^{-4} < \frac{1}{3}$$

Hmmmmm, this is not with high probability...

- To improve success probability, repeat this randomized procedure t times (for which t?)
- If we repeat for t times, failure probability is

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^t$$

- setting $t = 2n(n-1)$ then

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^t \leq \exp\left(-\frac{2t}{n(n-1)}\right) = e^{-4}$$

- **Running time:** One execution implemented in $O(n^2)$, so t executions in time $O(n^2 t) = O(n^4)$.

Hmmmmm, this is not with high probability...

- To improve success probability, repeat this randomized procedure t times (for which t?)
- If we repeat for t times, failure probability is

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^t$$

- setting $t = 2n(n-1)$ then

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^t \leq \exp\left(-\frac{2t}{n(n-1)}\right) = e^{-4}$$

- **Running time:** One execution implemented in $O(n^2)$, so t executions in time $O(n^2 t) = O(n^4)$.
- For running time improvements, see [Motwani & Raghavan 2007, Chapter 10.2]

Combinatorial Application

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

Combinatorial Application

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

Corollary

There are at most $O(n^2)$ minimum cuts in an undirected graph.

Combinatorial Application

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

Corollary

There are at most $O(n^2)$ minimum cuts in an undirected graph.

- Each minimum cut survives with probability $\Omega(1/n^2)$
- Events that two different cuts survive are disjoint

Combinatorial Application

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

Corollary

There are at most $O(n^2)$ minimum cuts in an undirected graph.

- Each minimum cut survives with probability $\Omega(1/n^2)$
- Events that two different cuts survive are disjoint
- Non-trivial statement to prove using other arguments!

Combinatorial Application

Theorem (Karger)

The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$, where $n = |V|$.

Corollary

There are at most $O(n^2)$ minimum cuts in an undirected graph.

- Each minimum cut survives with probability $\Omega(1/n^2)$
- Events that two different cuts survive are disjoint
- Non-trivial statement to prove using other arguments!

This is all good, but we haven't “sparsified” anything so far!

- Introduction
  - Why Sparsify?
  - Warm-up Problem
- Main Problem
  - Graph Sparsification
- Acknowledgements

Graph Sparsification

Definition (Weight of a cut)

Let $G(V, E, w)$ be undirected weighted graph. For any cut $(S, \overline{S})$, let the weight of $(S, \overline{S})$ be

$$w(S, \overline{S}) := \sum_{e \in E(S, \overline{S})} w(e).$$

If all weights are 1 then graph said to be "unweighted"

$$w(S, \bar{S}) = |E(S, \bar{S})|$$

Graph Sparsification

Definition (Weight of a cut)

Let $G(V, E, w)$ be undirected weighted graph. For any cut $(S, \overline{S})$, let the weight of $(S, \overline{S})$ be

$$w(S, \overline{S}) := \sum_{e \in E(S,\overline{S})} w(e).$$

Definition (Sparse Graph)

We say that a graph $G(V, E)$ is *sparse* if $|E| = \tilde{O}(|V|)$.

$\tilde{O}(n^\alpha)$ means any function $f: \mathbb{N} \to \mathbb{N}$ s.t. $f(n) = O(n^\alpha \cdot \log^\beta n)$ for some $\beta > 0$ (just hiding the log factors)

Graph Sparsification

Definition (Weight of a cut)

Let $G(V, E, w)$ be undirected weighted graph. For any cut $(S, \overline{S})$, let the weight of $(S, \overline{S})$ be

$$w(S, \overline{S}) := \sum_{e \in E(S, \overline{S})} w(e).$$

Definition (Sparse Graph)

We say that a graph $G(V, E)$ is *sparse* if $|E| = \tilde{O}(|V|)$.

Question

How to make a graph sparse (nearly linear # edges) while approximating the value of every cut of a graph?

Graph Sparsification

- **Input:** graph $G(V, E, w_G)$, $\varepsilon > 0$.

$$n = |V|, \ m = |E|.$$

- **Output:** graph $H(V, F, w_H)$ such that *for every cut* $(S, \overline{S})$, we have

$$(1-\varepsilon) \cdot w_G(S, \overline{S}) \leq w_H(S, \overline{S}) \leq (1+\varepsilon) \cdot w_G(S, \overline{S})$$

Graph Sparsification

- **Input:** graph $G(V, E, w_G)$, $\varepsilon > 0$.

$$n = |V|, \; m = |E|.$$

- **Output:** graph $H(V, F, w_H)$ such that *for every cut $(S, \overline{S})$*, we have

$$(1-\varepsilon) \cdot w_G(S, \overline{S}) \leq w_H(S, \overline{S}) \leq (1+\varepsilon) \cdot w_G(S, \overline{S})$$

- *Assumption (for this class):* the input graph $G(V, E)$ is unweighted and has minimum cut value $\Omega(\log n)$ (i.e., a large-ish cut)

Graph Sparsification

- **Input:** graph $G(V, E, w_G)$, $\varepsilon > 0$.

$$n = |V|, \ m = |E|.$$

- **Output:** graph $H(V, F, w_H)$ such that *for every cut* $(S, \overline{S})$, we have

$$(1-\varepsilon) \cdot w_G(S, \overline{S}) \leq w_H(S, \overline{S}) \leq (1+\varepsilon) \cdot w_G(S, \overline{S})$$

- *Assumption (for this class):* the input graph $G(V, E)$ is unweighted and has minimum cut value $\Omega(\log n)$ (i.e., a large-ish cut)

Algorithm:

- Let $p \in (0, 1)$ be a parameter.
- For each edge $e \in E(G)$, with probability p, make e an edge of H with weight $w_H(e) = 1/p$.

Graph Sparsification

Idea:

- Set p to get correct expected value for both $\#$ edges in H and the value of each cut $(S, \overline{S})$ in H.

Graph Sparsification

Idea:

- Set p to get correct expected value for both $\#$ edges in H and the value of each cut $(S, \overline{S})$ in H.
- After that, need to prove concentration around expected values *for all cuts simultaneously*!

Graph Sparsification

Idea:

- Set p to get correct expected value for both $\#$ edges in H and the value of each cut $(S, \overline{S})$ in H.
- After that, need to prove concentration around expected values *for all cuts* *simultaneously*!
- Use Chernoff-Hoeffding and assumption that min-cut value is large.

Graph Sparsification

Idea:

- Set p to get correct expected value for both $\#$ edges in H and the value of each cut $(S, \overline{S})$ in H.
- After that, need to prove concentration around expected values *for all cuts simultaneously*!
- Use Chernoff-Hoeffding and assumption that min-cut value is large.

Theorem ([Karger, 1993])

Let c be the value of the min-cut of G. Set

$$p = \frac{15 \ln n}{\varepsilon^2 \cdot c}.$$

Graph H given by algorithm from previous slide **approximates all cuts of** *G and has $O(p \cdot |E|)$ edges with probability $\geq 1 - 4/n$.*

Graph Sparsification

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
  $$X_e = \begin{cases} 1, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$

Graph Sparsification

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
  $$X_e = \begin{cases} 1, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$
- $$\mathbb{E}[|F|] = \sum_{e \in E} \mathbb{E}[X_e] = \sum_{e \in E} \underbrace{(p \cdot 1 + (1-p) \cdot 0)}_{\mathbb{E}[X_e]} = p \cdot |E|$$

linearity expectation

Graph Sparsification

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
$$X_e = \begin{cases} 1, & \text{if edge } e \text{ \textbf{included in} } H \\ 0, & \text{otherwise} \end{cases}$$
- $$\mathbb{E}[|F|] = \sum_{e \in E} \mathbb{E}[X_e] = \sum_{e \in E} (p \cdot 1 + (1-p) \cdot 0) = p \cdot |E|$$
- $$\mathbb{E}[w_H(S, \overline{S})] \overset{\text{linearity expectation}}{=} \sum_{e \in E(S, \overline{S})} \mathbb{E}[w_H(e)] = \sum_{e \in E(S, \overline{S})} \overbrace{\left(p \cdot \frac{1}{p} + (1-p) \cdot 0\right)}^{\mathbb{E}[w_H(e)]}$$
$$= |E(S, \overline{S})| \underset{\text{G is unweighted (i.e. } w_G(e) \in \{0,1\})}{=} k = \underbrace{w_G(S, \overline{S})}_{\text{definition}}$$

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
  $$w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
  $$w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$
- $w_H(S, \overline{S})$ is a sum of independent random variables w_e

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
$$w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$
- $w_H(S, \overline{S})$ is a sum of independent random variables w_e
- Chernoff Bound: (why Chernoff if vars not in $\{0,1\}$?)

$$\Pr[|w_H(S, \overline{S}) - k| \geq \varepsilon \cdot k] \leq 2 \exp\left(-\frac{\varepsilon^2 kp}{3}\right) = 2n^{-5k/c}$$

$$w_e = \frac{1}{p} \cdot X_e \qquad X_S = \sum_{e \in E(S,\bar{S})} X_e = p \cdot \sum w_e = w_H(S,\bar{S}) \cdot p$$

$$\Pr[|w_H(S,\bar{S}) - k| \geq \varepsilon k] = \Pr[|X_S - pk| \geq \varepsilon pk]$$

$$\underset{\substack{\downarrow \\ \text{Chernoff} \\ \text{on } X}}{\leq} 2e^{-\left(\frac{\varepsilon^2 \cdot pk}{3}\right)}$$

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let $w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$
- $w_H(S, \overline{S})$ is a sum of independent random variables w_e
- Chernoff Bound:

$$\Pr[|w_H(S, \overline{S}) - k| \geq \varepsilon \cdot k] \leq 2 \exp\left(-\frac{\varepsilon^2 kp}{3}\right) = 2n^{-5k/c}$$

- Note that $k \geq c$, as c is the weight of the minimum cut

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let $w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$
- $w_H(S, \overline{S})$ is a sum of independent random variables w_e
- Chernoff Bound:

$$\Pr[|w_H(S, \overline{S}) - k| \geq \varepsilon \cdot k] \leq 2 \exp\left(-\frac{\varepsilon^2 kp}{3}\right) = 2n^{-5k/c}$$

- Note that $k \geq c$, as c is the <u>weight of the minimum cut</u>
- This is probability of *single cut* deviating from its mean... How can we handle the *exponentially many* cuts in the graph?

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
$$w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$
- $w_H(S, \overline{S})$ is a sum of independent random variables w_e
- Chernoff Bound:

$$\Pr[|w_H(S, \overline{S}) - k| \geq \varepsilon \cdot k] \leq 2 \exp\left(-\frac{\varepsilon^2 kp}{3}\right) = 2n^{-5k/c}$$

- Note that $k \geq c$, as c is the weight of the minimum cut
- This is probability of *single cut* deviating from its mean... How can we handle the *exponentially many* cuts in the graph?
- **Observation:** probability that large cut violated is *much smaller*, and there are *not many small cuts*!

Graph Sparsification - Concentration

- Take a cut $(S, \overline{S})$. Suppose $k := w_G(S, \overline{S})$. Let
  $$w_e = \begin{cases} 1/p, & \text{if edge } e \textbf{ included in } H \\ 0, & \text{otherwise} \end{cases}$$
- $w_H(S, \overline{S})$ is a sum of independent random variables w_e
- Chernoff Bound:

$$\Pr[|w_H(S, \overline{S}) - k| \geq \varepsilon \cdot k] \leq 2 \exp\left(-\frac{\varepsilon^2 kp}{3}\right) = 2n^{-5k/c}$$

- Note that $k \geq c$, as c is the weight of the minimum cut
- This is probability of *single cut* deviating from its mean... How can we handle the *exponentially many* cuts in the graph?
- **Observation:** probability that large cut violated is *much smaller*, and there are *not many small cuts*!
- So we can do a clever union bound!

Number of Cuts Lemma

Lemma (Number of small cuts)

The number of cuts with at most $\alpha \cdot c$ edges for $\alpha \geq 1$ is at most $n^{2\alpha}$.

Number of Cuts Lemma

Lemma (Number of small cuts)

The number of cuts with at most $\alpha \cdot c$ edges for $\alpha \geq 1$ is at most $n^{2\alpha}$.

Practice problem: generalize our earlier proof on the # minimum cuts to this case.

Union Bound on # Cuts

Pr[some cut is violated] $\leq$

Union Bound on # Cuts

$$\Pr[\text{some cut is violated}] \leq$$

$$\leq \sum_{S \subseteq V} \Pr[(S, \overline{S}) \text{ is violated}]$$

union bound

Union Bound on # Cuts

$$\Pr[\text{some cut is violated}] \le$$

$$\le \sum_{S \subseteq V} \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\le \underbrace{\sum_{\alpha = 1,2,4,8,\dots}}_{\alpha = 2^i,\ i \in \{0,1,\dots,\log n\}} \underbrace{\sum_{\substack{S \subseteq V \\ \alpha c \le |w_G(S,\overline{S})| \le 2 \cdot \alpha c}}}_{\text{grouping cuts based on their weights}} \Pr[(S, \overline{S}) \text{ is violated}]$$

breaking up sum

$$2^i \cdot c \le w_G(S, \bar{S}) \le 2^{i+1} c$$

Remember: G unweighted, so

$$w_G(S, \bar{S}) = |E(S, \bar{S})|$$

Union Bound on # Cuts

$$\Pr[\text{some cut is violated}] \leq$$

$$\leq \sum_{S \subseteq V} \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\leq \sum_{\alpha=1,2,4,8,\ldots} \sum_{\substack{S \subseteq V \\ \alpha c \leq |w_G(S,\overline{S})| \leq 2\cdot\alpha c}} \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\leq \sum_{\alpha=1,2,4,8,\ldots} \underbrace{n^{4\alpha}}_{\downarrow} \cdot \Pr[(S, \overline{S}) \text{ is violated} \mid \alpha c \leq |w_G(S, \overline{S})| \leq 2 \cdot \alpha c]$$

upper bd. # cuts with weight between αc and 2αc (by our lemma)

(i.e. # edges in cut)

Union Bound on # Cuts

$$\Pr[\text{some cut is violated}] \leq$$

$$\leq \sum_{S \subseteq V} \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\leq \sum_{\alpha=1,2,4,8,\ldots} \quad \sum_{\substack{S \subseteq V \\ \alpha c \leq |w_G(S,\overline{S})| \leq 2 \cdot \alpha c}} \quad \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\leq \sum_{\alpha=1,2,4,8,\ldots} n^{4\alpha} \cdot \Pr[(S, \overline{S}) \text{ is violated} \mid \alpha c \leq |w_G(S, \overline{S})| \leq 2 \cdot \alpha c]$$

$$\leq \sum_{\alpha=1,2,4,8,\ldots} n^{4\alpha} \underbrace{\cdot\, 2n^{-5\alpha c/c}}_{\text{our Chernoff bound}}$$

Union Bound on # Cuts

$$\Pr[\text{some cut is violated}] \leq$$

$$\leq \sum_{S \subseteq V} \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\leq \sum_{\alpha = 1,2,4,8,\dots} \sum_{\substack{S \subseteq V \\ \alpha c \leq |w_G(S,\overline{S})| \leq 2\cdot\alpha c}} \Pr[(S, \overline{S}) \text{ is violated}]$$

$$\leq \sum_{\alpha = 1,2,4,8,\dots} n^{4\alpha} \cdot \Pr[(S, \overline{S}) \text{ is violated} \mid \alpha c \leq |w_G(S, \overline{S})| \leq 2 \cdot \alpha c]$$

$$\leq \sum_{\alpha = 1,2,4,8,\dots} n^{4\alpha} \cdot 2n^{-5\alpha c / c}$$

$$= \sum_{\alpha = 1,2,4,8,\dots} n^{-\alpha} \leq 4/n$$

↳ sum dominated by first term.

Union Bound on # Cuts

$$
\begin{aligned}
&\Pr[\text{some cut is violated}] \leq \\
&\leq \sum_{S \subseteq V} \Pr[(S,\overline{S}) \text{ is violated}] \\
&\leq \sum_{\alpha=1,2,4,8,\ldots} \sum_{\substack{S \subseteq V \\ \alpha c \leq |w_G(S,\overline{S})| \leq 2\cdot\alpha c}} \Pr[(S,\overline{S}) \text{ is violated}] \\
&\leq \sum_{\alpha=1,2,4,8,\ldots} n^{4\alpha} \cdot \Pr[(S,\overline{S}) \text{ is violated} \mid \alpha c \leq |w_G(S,\overline{S})| \leq 2\cdot\alpha c] \\
&\leq \sum_{\alpha=1,2,4,8,\ldots} n^{4\alpha} \cdot 2n^{-5\alpha c/c} \\
&= \sum_{\alpha=1,2,4,8,\ldots} n^{-\alpha} \leq 4/n
\end{aligned}
$$

Another application of Chernoff gives us that H has the right number of edges $|F| \approx p \cdot |E|$ (i.e., sparse)

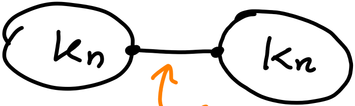
How to remove the assumption?

- Assumed that the graph has large min-cut value ($c = \Omega(\log n)$).

How to remove the assumption?

- Assumed that the graph has large min-cut value ($c = \Omega(\log n)$).
- Without min-cut assumption, uniform sampling won't work

Counterexample :



(dumbell graph)

cannot remove this edge!

How to remove the assumption?

- Assumed that the graph has large min-cut value ($c = \Omega(\log n)$).
- Without min-cut assumption, uniform sampling won't work
- **[Benczur, Karger 1996]:** without minimum cut assumption, just sample non-uniformly in clever way!

How to remove the assumption?

- Assumed that the graph has large min-cut value ($c = \Omega(\log n)$).
- Without min-cut assumption, uniform sampling won't work
- **[Benczur, Karger 1996]:** without minimum cut assumption, just sample non-uniformly in clever way!
- Sample edge with probability proportional to "connectivity" of two endpoints (i.e., how relevant is the edge between them?)

How to remove the assumption?

- Assumed that the graph has large min-cut value ($c = \Omega(\log n)$).
- Without min-cut assumption, uniform sampling won't work
- **[Benczur, Karger 1996]:** without minimum cut assumption, just sample non-uniformly in clever way!
- Sample edge with probability proportional to "connectivity" of two endpoints (i.e., how relevant is the edge between them?)
- **Strong Connectivity:** a k-strong component is a maximal induced subgraph that is k-edge-connected. For each edge e, let s_e be the maximum value k such that there exists a k-strong component containing e.

How to remove the assumption?

- Assumed that the graph has large min-cut value ($c = \Omega(\log n)$).
- Without min-cut assumption, uniform sampling won't work
- **[Benczur, Karger 1996]:** without minimum cut assumption, just sample non-uniformly in clever way!
- Sample edge with probability proportional to "connectivity" of two endpoints (i.e., how relevant is the edge between them?)
- **Strong Connectivity:** a k-strong component is a maximal induced subgraph that is k-edge-connected. For each edge e, let s_e be the maximum value k such that there exists a k-strong component containing e.
- Sample edge e with probability $p_e = \Theta\left(\frac{\log n}{\varepsilon^2 \cdot s_e}\right)$ and weight $1/p_e$.

Acknowledgement

- Lecture based largely on Lap Chi's notes.
- See Lap Chi's Lecture 1 notes at `https://cs.uwaterloo.ca/~lapchi/cs466/notes/L01.pdf`
- See Lap Chi's Lecture 3 notes at `https://cs.uwaterloo.ca/~lapchi/cs466/notes/L03.pdf`
- See Mohsen's notes for the general Benczur-Karger algorithm `https://people.inf.ethz.ch/gmohsen/AA18/Notes/S1.pdf`.

References I

Motwani, Rajeev and Raghavan, Prabhakar (2007)
Randomized Algorithms

Mitzenmacher, Michael, and Eli Upfal (2017)
Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis.
Cambridge university press, 2017.

Karger, David (1993)
Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm.
SODA 93, 21–30.

Benczur, Andras and Karger, David (1996)
Approximating st minimum cuts in $\tilde{O}(n^2)$ time.
Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 47 – 55.