18.409: An Algorithmist’s Toolkit
Lecture 9

Jonathan Kelner
Today

- Sparsification
Motivation and What We’ll Cover

- Suppose you have a graph $G$ with $m = \Theta(n^2)$ and want to approximately solve a cut problem (e.g., sparsest cut, min cut, s-t min cut)
- Running time of most algorithms grow with $m$ and are much slower for dense graphs than sparse ones
- So would be really nice if we could somehow throw out a lot of edges and get a similar answer
  - Then running time would be like that of a sparse graph
  - But, of course, you only get an approximate answer
- Idea: randomized sampling
- This isn’t spectral, but later we’ll see how to use spectral techniques to improve it
Create a new graph $G'$ by sampling every edge of $G$ with probability $p$

Expected number of edges = $mp$

Suppose have a cut $V = S \cup \overline{S}$ with $e_G(S) = c$

Each edge in cut is kept with prob. $p$

So expected value of $e_{G'} = pc$

By Chernoff bound,

$$\Pr [ |e_{G'}(S) - pc| \geq \epsilon pc] \leq e^{-\epsilon^2 pc/2}$$

So very likely to get about the right answer for sufficiently large cuts
More on First Cut at Algorithm

\[ \Pr \left[ |e_{G'}(S) - pc| \geq \epsilon pc \right] \leq e^{-\epsilon^2 pc/2} \]

- **Theorem (Karger):** If \( G \) has min cut \( c \), number of cuts less than \( \alpha c \) is less than \( n^{2\alpha} \)

- If \( p = \Omega \left( \frac{d \log n}{\epsilon^2 c} \right) \)

  cut of size \( \alpha c \) is within \((1+\epsilon)\) factor of expectation with probability \( n^{-O(1)\alpha} \) for whatever constant in the exponent we want

  - So can choose constants so every cut of size \( \alpha c \) is right
  - So can show every cut is within \((1+\epsilon)\) factor of correct value with probability \( 1-n^{-d} \)

- If \( c \) is small all bets are off
  - A small cut may get badly distorted
  - So need to take very large \( p \)
How to Fix the Problem [B-K]

- Suppose graph has small cut $c$, but edge $e$ is only involved in cuts of size at least $k$
  - Will actually need a slightly stronger assumption
- Somehow should only need to sample $e$ as if graph had min cut of size $k$
  - Then every edge in a cut of size $k$ would be sampled at least enough to give small probability of failure
- So maybe we should sample nonuniformly
  - See picture on blackboard
- **Problem:** Expectation isn’t right anymore
- **Solution:** When you do sample an edge, give it a weight of $1/p$
- Importance sampling
A Slightly More General Chernoff Bound

- We previously had:

- **Theorem (Chernoff Bound):** Let \( X_1, \ldots, X_n \) be independent \( \{0,1\} \) random variables and \( X = \sum_i X_i \). Then:

  \[
  \Pr[|X - E[X]| \geq \epsilon X] \leq 2e^{-\Theta(1)\epsilon^2 E[X]}
  \]

- If you look at a proof (or do the homework problem!), can replace \( X_i \in \{0,1\} \) with \( X_i \in [0,1] \) and theorem’s still true
  - Can’t make the \( X_i \)’s too big, or else one can dominate the sum

- Can scale everything up w/o changing anything:
  - Let \( Y_i = c X_i \), \( Y = \sum_i Y_i \)

  \[
  \Pr[|Y - E[Y]| \geq \epsilon Y] \leq 2e^{-\Theta(1)\epsilon^2 E[Y]/c}
  \]

- So just need to know bound \( c \) s.t. all \( Y_i \in [0,c] \)
Our Case

• Every edge is random variable $Y_e$
• $e$ gets a weight $w_e$
• If $e$ is in a cut of size $c$, require $w_e \leq c$
• Earlier version:
  ◦ Take all weights = size of min cut
  ◦ Let $p = \Omega(\log n/\varepsilon^2)$
  ◦ Let $Y_e = 1$ with prob. $p/w_e$
  ◦ Guess that cut size $= \sum_i w_e Y_e$
• Chernoff bound says get $\varepsilon$-approx for given cut with prob $n^{-\Theta(\sum 1/w_e)}$ where sum is over edges in cut
• If have many more edges in cut than min cut, could take bigger $w_e$’s and smaller probs and still get this cut right with good prob.
Our Case (cont.)

- So, perhaps, take $w_e = \text{size of min cut containing } e$
- Several problems
  - Won’t cover details of how to fix
    - Will post original paper on web site
  - Will give details of stronger scheme later today

**Problem:** Want to union bound over all cuts to show every cut is right
  - Doesn’t quite work, since have a lot of cuts of large size

**Also:**
  - Don’t know $w_e$!
  - Need to prove don’t take too many edges

- Rough ideas:
  - Can use approx $w_e$’s instead of exact ones, and can compute these quickly
  - Can use a slightly more conservative weighting scheme
When an edge appears, we count it with weight $w_i$

Size of every cut in $G'$ is $\varepsilon$-approx of size in $G$

So we’ve produced a weighted graph $G'$ with $O(n \log n/\varepsilon^2)$ edges in which every cut is approx the same as in $G$

Call this a *combinatorial sparsifier* of $G$
Let $G$ be our original graph, $G'$ our (weighted) combinatorial sparsifier.

We didn’t really do weighted Laplacians, but I claim that everything is similar:

- Think of just having $w_e$ multiple edges
- Off-diags $= -w_e$
- $j$th diag $= \sum_{(i,j) \in E} w_{i,j}$

Condition that all cuts are $\varepsilon$-approximated is just:

$$(1 - \varepsilon)x^T L_{G'} x \leq x^T L_G x \leq (1 + \varepsilon)x^T L_{G''} x$$

for all $x = \{-1,1\}^n$

Would be even better if works for all $x$

- By scaling, same as working for all $x \in [-1,1]^n$

Call this a spectral sparsifier of $G$

Reasonable conjecture: combinatorial sparsifiers are spectral sparsifiers

- Reasonable, but false!
A Counterexample

- Edge \((i,j)\) if \(|i-j| \leq k \text{ mod } n\)
- Add edge from 0 to \(n/2\)
- Min cut = 2\(k\)
- So removing 1 edge gives combinatorial sparsifier with \(\varepsilon = 1/k\)

Let \(x = (0, 1, \ldots, n/2, n/2-1, \ldots, 1, 0)\)

\[
x^T L_{G'} x = \sum_{(i,j) \in E} (x_i - x_j)^2 = \Theta(nk^3)
\]

\[
x^T L_G x = \sum_{(i,j) \in E} (x_i - x_j)^2 + \left(\frac{n}{2}\right)^2 = \Theta(nk^3) + \frac{n^2}{4}
\]

So need \(\varepsilon = \Omega(n/k^3)\), which can be very big (e.g., for constant \(k\))
Useful Tool: Matrix Inequalities

- **Question:** How do you define the notion that one symmetric matrix is “bigger” than another?
  - Everything is symmetric unless otherwise noted
- **Most obvious answer:**
  - \[ M \succeq N \text{ if } m_{i,j} \geq n_{i,j} \quad \forall \ i,j \]
- **Not great for our purposes**
  - Not basis independent
  - Says nothing about eigenvalues
Matrix Inequalities (cont.)

- Another try:
- $M \succeq N$ if the $i^{th}$ eigenvalue of $M$ is $\geq i^{th}$ eigenvalue of $N$ for all $i$
- Better, but too basis independent
- Do we really want to say

\[
\begin{pmatrix}
1 & 0 \\
0 & -1 \\
\end{pmatrix} \succeq \frac{1}{\sqrt{2}} \begin{pmatrix}
1 & 1 \\
1 & -1 \\
\end{pmatrix}
\]

and

\[
\frac{1}{\sqrt{2}} \begin{pmatrix}
1 & 1 \\
1 & -1 \\
\end{pmatrix} \succeq \begin{pmatrix}
1 & 0 \\
0 & -1 \\
\end{pmatrix}
\]
A Good Definition

• **Definition:** We’ll say \( M \succeq N \) if
  \[
  x^T M x \geq x^T N x \quad \forall x \in \mathbb{R}^n
  \]

• **Properties:** (some were on problem set)
  1. If \( M \succeq N \) and \( N \succeq M \) then \( M = N \)
  2. \( M \succeq 0 \) iff \( M \) is positive semidefinite
  3. \( M \succeq N \) iff \( M - N \) is positive semidefinite
  4. If \( M_1 \succeq N_1 \) and \( M_2 \succeq N_2 \) then \( M_1 + M_2 \succeq N_1 + N_2 \)
  5. Other nice properties, as needed
Matrix Inequalities and Eigenvalues

- Let M have eigenvalues $\lambda_1, ..., \lambda_n$ and let N have eigenvalues $\mu_1, ..., \mu_n$
- **Claim:** If $M \succeq N$, then
  $$\lambda_i \geq \mu_i \quad \forall i$$
- **Proof:** Was on problem set
We now know how to define inequalities on graphs $G$ and $H$ (on the same vertex set):

**Definition:** We say $G \succeq H$ if $L_G \succeq L_H$

**Claim:** Let $G=(V,E_G,w_G)$ and $H=(V,E_H,w_H)$ be weighted graphs on the same vertex set such that $w_G(i,j) \geq w_H(i,j)$ $\forall (i,j) \in E$. Then $G \succeq H$.

Why?
Spectral Sparsification

- Have dense graph G, want to produce a sparse graph H s.t.
  \[ L_H \leq L_G \leq (1 + \varepsilon)L_H \]
- Sparse will mean \( n \cdot \text{polylog}(n) \) edges
- Recall that Benczur-Karger sparsifier didn’t quite get this
  - Got \( x^T L_H x \leq x^T L_G x \leq (1+\varepsilon)x^T L_H x \) for \( \{\pm 1\}\)-vectors
  - Need it for all vectors
- In B-K, easy case was “well-connected” graphs
- Same is true for us, but with a slightly different notion of connectedness
  - Any guesses?
  - Instead of have no small cuts, we’ll look at graphs with no sparse cuts
  - Or, equivalently-ish, look at graphs with big \( \lambda_2 \)
- It turns out that uniform random sampling works for graphs with good expansion, but problems analogous to B-K appear when there are sparse cuts
  - Well, almost. Need to sample with prob. inversely proportional to the degree.
  - Will follow from proof of general scheme
What We’ll Show (and What is True)

- We’ll show how to construct spectral sparsifiers with $O(n \log n)$ edges in polynomial time
  - Will be cleaner and more geometric than B-K analysis
  - Good example of how generalizing can sometimes make things easier, not harder
- Can actually construct them in nearly linear time
  - Will probably develop the tools for this later
    - The main one is an algorithm for quickly solving linear systems involving graph Laplacians
    - Also uses a theorem on concentration of measure that we’ll prove during convex geometry unit
- In poly time, you can actually construct spectral sparsifiers with $O(n)$ edges [BSS]!
  - Note that such a sparsifier of a complete graph is an expander
    - Well, almost. Expanders have constant max degree, whereas this just guarantees constant average degree
  - Would be very nice to get this in nearly linear time
The Algorithm

- Algorithm is very simple
  - Same structure as B-K, just different probabilities

- Algorithm:
  - Compute a probability $p_e$ for each edge $e$
  - Sample each edge uniformly with probability $p_e$, and,
  - If edge is selected, include it with weight $1/p_e$.

- Probs based on linear algebraic notion of importance (instead of graph theoretic one)
  - Have very nice interpretation in terms of effective resistances of circuits

- Need to develop two main tools:
  - Effective resistances
  - A matrix analogue of the Chernoff bound

- Bibliographic note: many of the following slides/graphics are taken with minor modifications from a talk by Dan Spielman
(Moore-Penrose) Pseudoinverses

- How do you invert a non-invertible matrix?
- Let $M$ be $n \times n$ and symmetric
- Diagonalize:
  \[ M = \sum_{i=1}^{n} \lambda_i v_i v_i^T \]
- If all eigenvalues are nonzero, \[ M^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} v_i v_i^T \]
- When degenerate, get pseudoinverse by throwing away zero eigenvalues \[ M^+ = \sum_{i \mid \lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^T \]
- Can define for nonsquare/nonsymmetric too, but we won’t need it
- Has lots of nice properties. We’ll use:
  \[ \ker(L) = \ker(L^+) \]
  \[ MM^+ = \sum_{i \mid \lambda_i \neq 0} v_i v_i^T = \text{proj. onto nonzero eigenvects} \]
  \[ \text{So } MM^+ \text{ equals identity when restricted to } \text{image}(M) \]
Effective Resistance

Treat each edge as a resistor of resistance 1
(If had capacity c, would use 1/c)

$R_{eff}(e)$ is resistance between endpoints of e
Effective Resistance

Treat each edge as a resistor of resistance 1
(If had capacity c, would use 1/c)

$R_{\text{eff}}(e)$ is resistance between endpoints of e

Resistance of path is 2
Effective Resistance

Treat each edge as a resistor of resistance 1
(If had capacity $c$, would use $1/c$)

$R_{eff}(e)$ is resistance between endpoints of $e$

Resistance of path is 2

$\frac{1}{1/2 + 1/1} = \frac{2}{3}$
Effective Resistance

Treat each edge as a resistor of resistance 1 (If had capacity c, would use 1/c)

\( R_{\text{eff}}(e) \) is resistance between endpoints of edge

\[ i(u, v) = \frac{2}{3} \]

\[ i(u, a) = \frac{1}{3} \]

\[ i(a, v) = \frac{1}{3} \]

\[ v = i r \]

= potential difference between endpoints when flow one unit from one endpoint to other
Laplacians and Electrical Flow

Orient edges arbitrarily.

\[ U = \text{edge-vertex adj matrix}, \quad C = \text{diag}(c), \quad r_e = 1/c_e \]

\[
U(e, v) = \begin{cases} 
1 & \text{if } v \text{ is the head of } e \\
-1 & \text{if } v \text{ is the tail of } e \\
0 & \text{otherwise}
\end{cases}
\]

\[ L = U^T C U \]
Laplacians and Electrical Flow

Orient edges arbitrarily.

\[ U = \text{edge-vertex adj matrix}, \ C = \text{diag}(c), \ r_e = 1/c_e \]

\[ U(e, v) = \begin{cases} 
1 & \text{if } v \text{ is the head of } e \\
-1 & \text{if } v \text{ is the tail of } e \\
0 & \text{otherwise} 
\end{cases} \]

\[ L = U^T C U \]

Ohm's law: \( i = CUv, \ i \in \mathbb{R}^E, v \in \mathbb{R}^v \)

Conservation: \( i_{\text{ext}} = U^T i, \ i_{\text{ext}} \in \mathbb{R}^V \)

\[ i_{\text{ext}} = Lv \quad v = L^+ i_{\text{ext}} \]
Laplacians and Effective Resistance

potential difference between endpoints when flow one unit from one endpoint to other

\[ U(e, v) = \begin{cases} 
1 & \text{if } v \text{ is the head of } e \\
-1 & \text{if } v \text{ is the tail of } e \\
0 & \text{otherwise}
\end{cases} \]

\( u_e = \text{e}^{\text{th}} \text{ row} \quad v = L^+ i_{\text{ext}} \)

\[ R_{\text{eff}}(e) = u_e L^+ u_e^T \]

\[ R_{\text{eff}}(e) = (UL^+U^T)(e, e) \]
A Matrix Tail Bound

- We’ll use the following theorem (in which k is a universal constant):

\[
\text{Thm [RV]} \text{ For distribution on vectors } y \text{ s.t. }
\|y\| \leq t \text{ and } \left\| Eyy^T \right\|_2 \leq 1
\]

\[
E \left\| Eyy^T - \frac{1}{q} \sum_{i=1}^{q} y_i y_i^T \right\|_2 \leq kt \sqrt{\log \frac{q}{q}}.
\]

- It’s a “concentration of measure” theorem
  - We’ll talk about how to prove these when we study convex geometry

- Why do we need \( \|y\| \leq t \)?

- Why is this similar to the Chernoff bound?

- What is the right equivalent to B-K’s importance sampling?
To prove approximation

$$1 - \epsilon \leq \frac{x^T L_H x}{x^T L_G x} \leq 1 + \epsilon, \quad \forall x$$

Suffices to show

$$1 - \epsilon \leq \frac{z^T M^T L_H M z}{z^T M^T L_G M z} \leq 1 + \epsilon, \quad \forall z$$

Provided \( x \perp \text{null}(L_G) \implies x \in \text{range}(M) \)
To prove approximation

\[ 1 - \epsilon \leq \frac{z^T M^T L_H M z}{z^T M^T L_G M z} \leq 1 + \epsilon, \quad \forall z \]

Choose \( M \) so that \( M^T L_G M \) is a projection

Then, suffices to show

\[ \| M^T L_H M - M^T L_G M \|_2 \leq \epsilon \]
The Projection

\[ L_G = U^T C U \]

\[ M = L_G^+ U^T C^{1/2} \]

\[ \Pi = M^T L_G M \]

\[ = C^{1/2} U L_G^+ U^T C^{1/2} \]

\[ = \Pi \Pi \Pi \]
Sampled, re-weighted Laplacian

\[ L_G = U^T C U \]

\( d_e \) is weight in sparsifier \( H \)

Set \( S(e, e) = \frac{d_e}{c_e} \)

So,

\[ L_H = U^T C S U = U^T C^{1/2} S C^{1/2} U \]

\[ M^T L_H M = \Pi S \Pi \]
Sampling, choosing $S$

Need to choose diagonal $S$ such that $\text{nnz}(S) \leq O(n \log n / \epsilon^2)$

$$\| \Pi S \Pi - \Pi \|_2 \leq \epsilon$$

Let $\pi_e = \Pi(\cdot, e)$, so $\Pi S \Pi = \sum S(e, e) \pi_e \pi_e^T$

$$\| \pi_e \|^2 = \Pi(e, e) = c_e R_{eff}(e)$$

As $\Pi = \Pi \Pi = C^{1/2} (U L_G^+ U^T) C^{1/2}$
Sampling, choosing $S$

Set \[ \tau_e = \sqrt{\frac{n - 1}{c_e R_{\text{eff}}(e)}} \pi_e \quad \|\tau_e\| = \sqrt{n - 1} \]

Choose with probability \[ p_e = \frac{c_e R_{\text{eff}}(e)}{n - 1} \]

\[
\left( \sum_e c_e R_{\text{eff}}(e) = \sum_e \Pi(e, e) = \text{tr}(\Pi) = n - 1 \right)
\]

And,

\[ E\tau_e\tau_e^T = \sum_e p_e \tau_e\tau_e^T = \sum_e \pi_e \pi_e^T = \Pi \]
Conclusion of proof

By theorem of Rudelson and Vershynin, if sample $q$ times with replacement,

set $S(e, e) = \frac{1}{qc_e R_{\text{eff}}(e)} \times$ number times $e$ is chosen

$$
E \| \Pi - \Pi S \Pi \|_2 \leq k \sqrt{n - 1} \sqrt{\frac{\log q}{q}} \leq \frac{\epsilon}{2}
$$

For $q = O(n \log n / \epsilon^2)$
Algorithmics of the Construction

- Easy to see that this is poly time
  - Whole procedure is constructive and just uses standard linear algebra operations
- Bottleneck is computing effective resistances
  - Involves a matrix (pseudo-)inversion and a few multiplications
- Can actually do it in nearly linear time
- Requires two main components:
  - Nearly linear algorithm for solve linear systems of the form \( Lx = b \) when \( L \) is a Laplacian
    - This lets you compute 1 effective resistance very quickly...
  - A way to compute all the effective resistances by solving logarithmically many linear systems
    - Uses Johnson-Lindenstrauss Lemma