Announcements

- Start recording
- Scribe: work together to produce one set of notes
- My OLT: Mon 12:30-1:30, Thur 9-10
- TC hybrid setting: Mon 2-3, Wed 2-3, Wed 5-6, Thur 5:30-6:30
- PS 2 due Fri, including final project ideas/ problems
- If Zoom goes down, check Piazza
- Sync whiteboard
- Gather after class! Link in chat
- Jamboard link in chat
- FILL OUT FEEDBACK FORM

Agenda

- Recap
- Fiedler’s Algorithm
- The Power Method
Recap: \( \phi(s) = \frac{\omega(s)}{d(s)} \)

\[ \phi(G) = \min_{s: d(s) \leq d(V)/2} \max_{S \subseteq V} \left\{ \phi(s), \phi(V \setminus S) \right\} \]

\[ \phi_k(G) = \min_{S_1, \ldots, S_k \text{ disjoint}} \max_{i=1}^k \phi(s_i), \ldots, \phi(s_k) \]

Cheeger's Inequalities

\[ \sqrt{\frac{n}{k}} \leq \phi(G) \leq \sqrt{2\chi_2} \]

High-order Cheeger

\[ \sqrt{\frac{n}{k}} \leq \phi_k(G) \leq \text{poly}(k) \cdot \sqrt{\chi_k} \]

Fidler's Algorithm for Sparsest Cut

Given weighted, undirected \( G \), find \( S \subseteq V \) w/ \( d(S) \leq d(V)/2 \) and \( \phi(S) \leq \sqrt{2\chi_2} = O(\sqrt{\chi_2}) \)

1. Find a vector \( y \perp d \)

\[ \frac{y^T L y}{y^T D y} \leq \chi_2 \]

2. Sort \( y(1) \leq y(2) \leq \cdots \leq y(n) \)

3. Find \( k \in \{1, \ldots, n\} \) minimizing

\[ \max_{i \leq k} \min_{1 \leq j \leq m} d(i) \]

4. By eigen-decomposition of \( N \),

\[ 2 \leq \omega \leq 2.378 \ldots \]

5. \( O(n \log n) \)

6. \( O(\chi_2 \log n) \)
How to speed up Step 1? 

Goal: time $O(n+m)$

**Power Method**

Estimate:

1. **Largest $\lambda$-Value of a PSD matrix $M$**

(1) Choose $x \in \mathbb{R}^n$ uniformly at random

(2) Let $y = M^k x$, for $k = O\left(\frac{\log(n/\epsilon)}{\epsilon}\right)$

\[ y^T y \text{ is a good proxy for } \lambda_{\text{max}} \]

\[ \text{time } O(k \cdot m) \]

\[ \text{cf. related squaring } \text{time } O((\log k) \cdot n^2) \]

Thus: \[ M_i = \frac{y^T M y}{y^T y} \geq (1-\epsilon) \lambda_i \]

**Pf:** Write $x = c_1 v_1 + \cdots + c_n v_n$

\[ y = M^k x = c_1 y^T v_1 + \cdots + c_n y^T v_n \]

\[ y^T y = c_1^2 M_i + \cdots + c_n^2 M_n \]

Claim: \[ \text{with high prob., } c_i \geq \sqrt{2} \]

Then: \[ y^T y = \sum_{i=1}^{n} c_i^2 M_i \geq (1-\epsilon) \sum_{i=1}^{n} c_i^2 M_i \]

\[ c_i^2 M_i \leq \frac{1}{\epsilon} \frac{1}{\epsilon} \sum_{i=1}^{n} c_i^2 M_i \]

\[ c_i^2 M_i \leq \frac{1}{\epsilon} \sum_{i=1}^{n} c_i^2 (1-\epsilon) M_i \]

\[ c_i^2 M_i \leq (1-\epsilon) M_i \]

Let $\ell$ be s.t. $M_\ell \geq (1-\epsilon) M_i$
\[ \|x\|_2^2 = n \]

\[ \|x\|_2^2 = \sum_{i=1}^{n} c_i^2 u_i^2 + n \cdot u_k^2 \]

\[ \leq \sum_{i=1}^{n} c_i^2 u_i^2 + 4 \varepsilon c_k^2 u_k^2 \]

\[ \leq (1 + 4 \varepsilon) \cdot \left( c_1^2 u_1^2 + \ldots + c_k^2 u_k^2 \right) \]

\[ \geq (1 - \varepsilon) \cdot u_1 \cdot \left( c_1^2 u_1^2 + \ldots + c_k^2 u_k^2 \right) \]

\[ \text{Ratio} \geq \frac{1 - \varepsilon}{1 + 4 \varepsilon} = 1 - O(\varepsilon) \]

**Proof of claim:**

\[ C = x^T v_1 = \sum_{a} x(a) v_1(a) \]

\[ E[C_1] = \sum_{a} x(a) v_1(a) \]

\[ E[C_1^2] = \sum_{a,b} E[x(a) x(b) v_1(a) v_1(b)] \]

\[ E[C_1^4] \leq 3 \]

\[ \text{Var}[C_1^2] \leq 2 \]

Q: Why not use Hoeffding or Chebychev? Instead: Rademacher-Zygmund
Breakout Q: Why doesn't the above theorem/proof show that the lazy random walk on the undirected n-cycle mixes in time $O(\log n)$?

### Reasons it might apply
- $W$ on regular graph is PSD
- $\langle x, v_1 \rangle \leq \frac{\|x\|_1}{\sqrt{n}}$ also holds for any prob. dist $x$ and $v_1 = \frac{1}{\sqrt{n}}$

### Reasons it doesn't apply
- $\frac{y^T M y}{y^T y} \geq (1-\varepsilon) m_1$
- $y$ is close to $v_1$ e.g. $y = v_2$ and $m_2 \geq (1-\varepsilon) m_1$

## 1. 2nd largest eigenvalue of a PSD matrix

1. Let $x = e_{1 \pm \beta n}$
2. $y = M^k x_0$ for $k = O\left(\frac{(\log n)^{\alpha}}{\varepsilon^2}\right)$

### Cor: w/ constant probability

$$m_2 \geq \frac{y^T M y}{y^T y} \geq m_1 \cdot (1-\varepsilon)$$

and $y \perp v_1$
Second Smallest $\varepsilon$-value of Normalized Laplacian

Apply $\Pi$ to matrix

Evolves:

Obtain vector $y \perp$ s.t.

$$\frac{y^Ty}{y^TNy} \geq 1$$

i.e.

$$\frac{y^TNy}{y^Ty} \leq \frac{\varepsilon}{y^2}$$

Take $\varepsilon = \frac{y^2}{n}$

$\Rightarrow$ time $O(m \cdot \frac{\log(n \cdot y^2)}{y^2})$

Not good
Better: apply power meth to "pseudoinverse" $N^+$

eigenvalues $0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \ldots, \frac{1}{\sqrt{n}}$

Later in course: algorithms that gain $G$ and $x$, compute $N^+ x$ in the $O(m+n)$,

$\Rightarrow$ find cut of conductance $O(\sqrt{\lambda_2})$

in time $O(m+n)$

nearly linear time