CS 598: Spectral Graph Theory. Lecture 5

Graph Cutting and Cheeger's Inequality

Alexandra Kolla

Today

- Why do we cut graphs?
- Cut ratio, and integer programming formulation
- Integer programming relaxation, easy direction of Cheeger
- Difficult direction of Cheeger

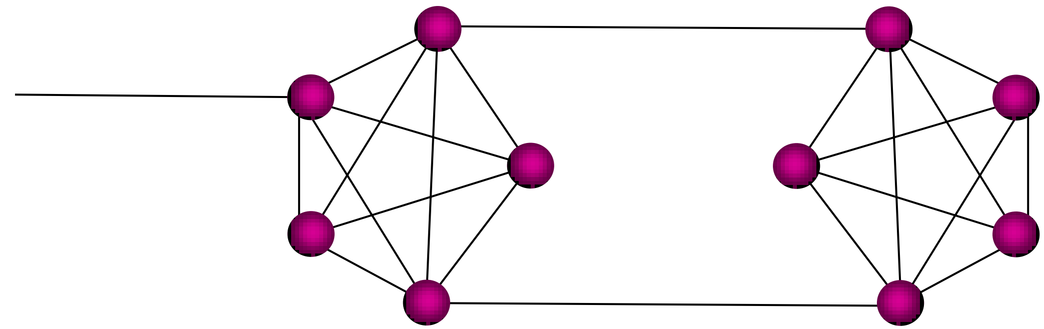
Why Cut?

- One of the inspirations of spectral graph theory is graph partitioning
- Want to cut a graph in two approximately equally sized pieces while minimizing the number of edges cut.
- Applications like divide-and conquer algorithms, clustering etc
- Concentrate on two-piece partitions

Some Notation

- Graph G=(V,E)
- $S \subseteq V$ a set of vertices of G
- |S| = the number of vertices in S
- $\bar{S} = V \backslash S$ the complement of S
- $e(S) = e(\bar{S})$ = the number f edges between $S, \bar{S}$

First Instinct: Min Cut

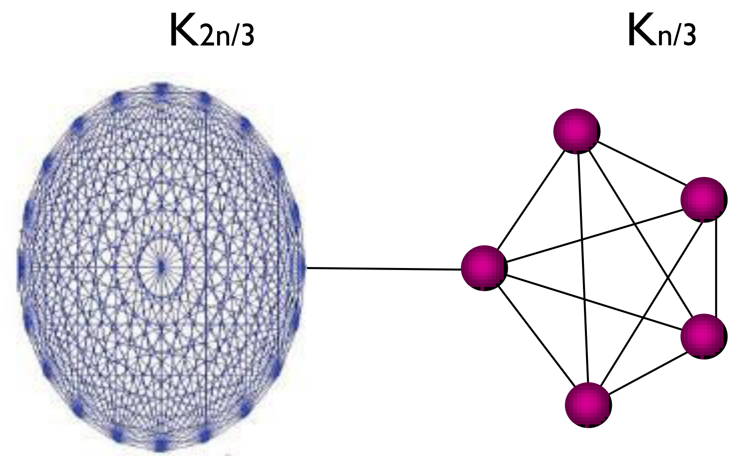


- Min Cut: divide G into

two parts as to minimize e(S)

- Would cut the one edge on the left and not in the middle

Second Instinct: Approximate Bisection



- Cut in equal size

pieces while minimizing e(S)

- Would cut the clique on the left to achieve balance but would cut too many edges

A Good Tradeoff: Cut Ratio

- Cut ratio : $$\phi(S) = \frac{e(S, \bar{S})}{\min(|S||\bar{S}|)}$$
- Sparsest Cut is the one that minimizes cut ratio. Also called isoperimetric number of G: $$\phi(G) = \min_{S \subseteq V} \phi(S)$$
- Nice property that if S_1, S_2 disjoint and $|S_1 \cup S_2| \leq n/2$ then

$$\phi(S_1 \bigcup S_2) \leq \max\{\phi(S_1), \phi(S_2)\}$$

An Integer Program for Cut Ratio

- How to find the optimal cut fast? Integer program for cut ratio.
- Associate every cut $S - \bar{S}$ with a vector $x \in \{-1,1\}^n$, where

$$x_i = \begin{cases} -1, i \in S \\ 1, i \in \bar{S} \end{cases}$$

  - We can now write

$$e(S) = \frac{1}{4}\sum_{(i,j)\in E}(x_i - x_j)^2$$

$$|S| \cdot |\bar{S}| =$$

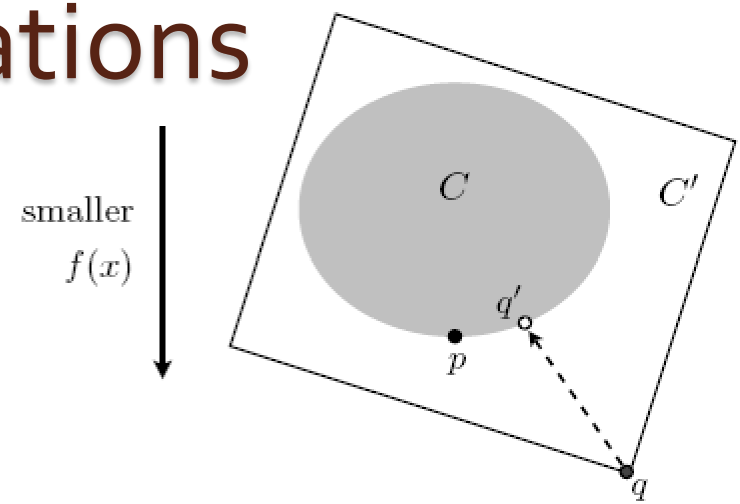
$$(\sum_{i\in V}[i \in S])(\sum_{j\in V}[j \in \bar{S}]) = \sum_{i,j\in V}[i \in S, j \in \bar{S}] = \frac{1}{2}\sum_{i,j\in V}[x_i \neq x_j] = \frac{1}{4}\sum_{i<j}(x_i - x_j)^2$$

[A] is the characteristic function of boolean event A. It is 1 if A true, zero otherwise.

Solving the Integer Program

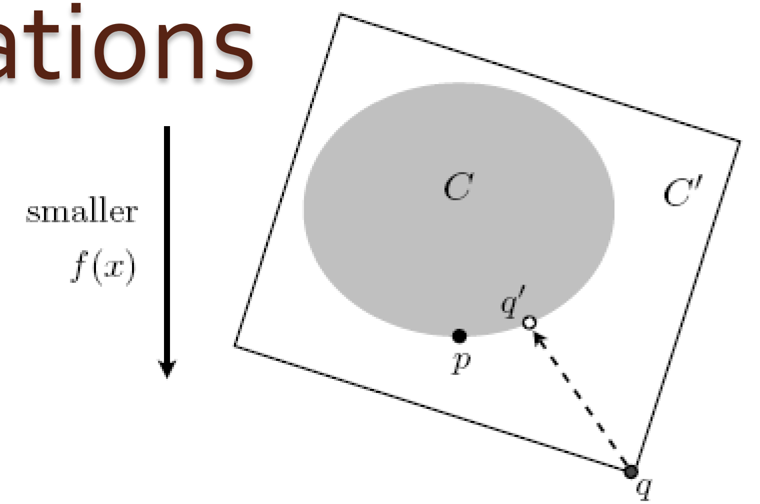
- $$min_{x\in\{-1,1\}^n} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2} = min_{S\subseteq V} \frac{e(S)}{|S|\cdot|\bar{S}|}$$
- $$\mathrm{n}/2\min\{S,\bar{S}\} \le |S|\cdot|\bar{S}| \le \mathrm{n}\min\{S,\bar{S}\}$$
- Solving the integer program approximates sparsest cut within 2.
  $$\frac{1}{n}\phi(G) \le min_{S\subseteq V} \frac{e(S)}{|S|\cdot|\bar{S}|} \le \frac{2}{n}\phi(G)$$
- NP-hard to solve
- Remove integrality constraint, get relaxation

A Note on Relaxations



- Often in approximation algorithms:
  - Want to solve NP-hard problem: "minimize f(x) subject to constraint x∈C"
  - Instead, we relax constraint and solve the problem:"minimize f(x) subject to constraint x∈C′ " for weaker C′.
  - Gives a lower minimum
  - Then need to round solution q to a feasible one, that is close to the optimal one p.

A Note on Relaxations



- Immediately, f(q)≤ f(p)
- To get a c-approximation (c>1) we need to round q to a point q′ and show

$$f(q') \leq cf(q) \leq c\, f(p)$$

Solving the Relaxation

$$min_{x \in R} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2}$$

- We use $min_{S \subseteq V} \frac{e(S)}{|S| \cdot |\bar{S}|} \leq \frac{2}{n} \phi(G)$
- Details on blackboard, and we obtain

$$\phi(G) \geq \frac{\lambda_2}{2}$$

  - Next Lecture, we will see more on relaxations and connections with λ_2

The Other Direction

- We just showed that $\phi(G) \geq \frac{\lambda_2}{2}$
- What about other direction? Need rounding method which will be a way to get a cut from λ_2 and v_2 together with upper bound on how much the rounding increases the cut ratio.
- **Cheeger's Inequality**:

$$\lambda_2 / 2 \leq \phi(G) \leq \sqrt{2d_{\max}} \sqrt{\lambda_2}$$

- Both upper and lower bounds are tight (up to constant), as seen by path graph and complete binary tree. Both have sparsest cut O(1/n), but P_n has $\lambda_2 = \Theta(1/n^2)$ and T_n has $\lambda_2 = \Theta(1/n)$, see lecture 4.
- We show the difficult direction next: $\dfrac{\phi(G)^2}{2d_{\max}} \leq \lambda_2$

The Proof of Cheeger’s Inequality

How to Get a Cut from λ_2 and v_2

- Algorithmic proof
- Let $x \in R^n$ be any vector such that $x \perp 1$
- Order vertices of x such that $x_1 \leq x_2 \leq \ldots \leq x_n$
- Let S={1,…,k} for some value of k. This will be our cut. Algorithm tries all values of k to find the best one, k depends on graph.
- We will next show something stronger

How to Get a Cut from λ_2 and v_2

Theorem

For any $x \perp 1$, such that $x_1 \leq x_2 \leq \ldots \leq x_n$, there is some i for which

$$\frac{\phi(\{1,\ldots,i\})^2}{2d_{\max}} \leq \frac{x^T L x}{x^T x}$$

This not only implies Cheeger by taking $x = v_2$ but also gives an actual cut. Also works if we only have good approximations of λ_2 and v_2

Proof: see blackboard