LECTURE 4

MAX-CUT, SDPs, ...

Lecture 4: Max-Cut, SDPs

The Problem:

Input: $G = (V, E)$: graph on n vertices, m edges

Goal: compute $S \subseteq V$ s.t $\frac{|E(S, \bar{S})|}{m}$ is maximized.

"normalized max-cut(G)."

We'll simply call it max-cut

Fact: Computing Max-cut(G) is NP-hard.

Pf: Reduction from Max Independent Set.

Remark: You may recall that min-cut(G) can be computed in polynomial time. What a difference min vs max makes!

Approximation Algorithms for Max-Cut

- $\frac{1}{2}$-approximation is easy.

Lemma: Let S be a uniformly random set of vertices of V. Equivalently, choose each $v \in V$ to be in S w.p. $\frac{1}{2}$ independently of others.

Then, $\mathbb{E}_S\left[\frac{|E(S,\bar{S})|}{m}\right] \geq \frac{1}{2}$

Since max-cut$(G) \leq 1$, this gives a $\frac{1}{2}$ approx.

"Surrogate for max-cut$(G) = m$"

Proof: $\mathbb{E}$[edge $\{u,v\}$ is cut] $= 1/2$.

Apply linearity of expectations.

don't need to use randomized algo.

Local Search: 1) Start from any cut S

2) If there's a v such that moving v to the other side of the cut improves cut size, do it.

3) Stop when there's no such v & return the resulting cut.

Analysis: Exercise.

Question: Is there a $> \frac{1}{2}$ factor approx. algo for Max-Cut?

Set Cover: LPs helped. What about here?

quadratic program

- QP for Max-Cut

$$\boxed{\begin{aligned} &\max \frac{1}{4m} \sum_{\substack{e=\{u,v\} \\ \in E}} (x_u - x_v)^2 \\ &\text{s.t.} \quad x_u \in \{\pm 1\} \text{ for all } u \in V. \end{aligned}}$$

Every $x = (x_u)_{u \in V} \in \{\pm 1\}^n$ is a "± 1" indicator of a set of vertices. The objective function computes the size of the cut defined by x.

Clearly, QPs are NP-hard to solve.

Can we relax & round them?

- LPs for Max-Cut

must "linearize" the objective.

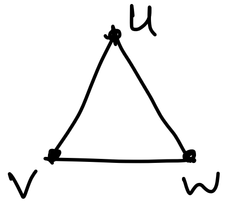
$$\frac{1}{m} \sum_{e=\{u,v\} \in E} y_{u,v}$$

$\hookrightarrow$ standing for $\frac{(x_u - x_v)^2}{4}$

$$y_{u,v} \in [0,1]$$

Clearly useless.

Can we add additional constraints?



no cut can "cut" all 3 edges.

- For every $\{u, v, w\} = \Delta$ in G,

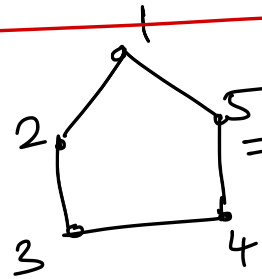
$$y_{uv} + y_{vw} + y_{uw} \leq 2.$$

Do they help?

In general, can add any linear constraint that is feasible...

<u>NOPE</u>.

<u>Integrality Gap</u>: Let



$= G$.

Then clearly $y_{1,2} = y_{2,3} = \ldots = y_{5,1} = 1$

is a solution.

In fact, we know that for some $c \in [0,1]$, we need at least

2^{n^c} constraints to improve on $\frac{1}{2}$ approx.

Fact: [K, Meka, Raghavendra '18]

Beating $\frac{1}{2}$ for Max-Cut requires $> 2^{n^c}$ size "extended formulations" for some constant $c > 0$.

Brief Intuition: There are d-regular graphs G_1, G_2 s.t. max-cut$(G_1) \sim 1$ (almost bipartite)

max-cut$(G_2) \sim \frac{1}{2}$ ("minimal")

but local neighborhood around every vertex in both G_1 & G_2 looks exactly the same. Both are $(d-1)$-regular trees.

LPs, local algorithms seem unable to distinguish between such pairs.

Moving On :

Can we hope to add some non-linear constraints on $y_{u,v}$ that provide more power?

Idea: Objective: $\frac{1}{4m} \sum_{\substack{e=\{u,v\} \\ \in E}} (x_u - x_v)^2$

$$= \frac{1}{4m} \sum_{e=\{u,v\} \in E} 2 - 2 x_u \cdot x_v .$$

$$= \frac{1}{2} - \frac{1}{2m} \cdot \sum_{e=\{u,v\} \in E} x_u \cdot x_v .$$

Observation:

$$M = \begin{array}{c} \\ u \\ \\ \end{array}\begin{array}{c} v \\ \left[\begin{array}{cc} \cdot\, x_u^2 = 1 & \\ & x_u \cdot x_v \\ & \end{array}\right] \end{array} = XX^T \succeq 0$$

$n \times n$ matrix with $\text{diag} = \vec{1}$

positive semidefinite

A new relaxation:

$$\max \ \frac{1}{2} - \frac{1}{2m} \sum_{\substack{e \in \{u,v\} \\ \in E}} X_{u,v}$$

s.t. $\text{diag}(X) = 1$ → linear constraint

"non linear constraint" ← $X \succeq 0$

relaxation because we forgot "rank 1" constraint

Detour: Semidefinite Programs

a class of convex programs.

linear objective $\leftarrow \max \sum c_i x_i$

$$\text{s.t. } x \in K$$

Convex program

K: convex subset of $\mathbb{R}^N$

CAUTION: Convex programming in general is NP-hard.

Convex $\neq$ easy

but some convex programs are "solvable" in polynomial time:

e.g. Linear Programs

K = intersection of linear inequalities

$= \{ x \mid A_i x \leq b_i \ \forall\, 1 \leq i \leq m \}$

Def (Semidefinite Programs)

An SDP in $n \times n$ matrix valued variable X is a convex program where

$$K = \{ X \succeq 0 \mid \langle A_i, X \rangle_F \leq b_i \}$$

for $1 \leq i \leq m$

$\sum_{j,k} A_i(j,k) \cdot X(j,k)$ = Frobenius inner product (HWO)

SDPs can be solved approximately via ellipsoid method. ??
We'll study this in more detail in last two weeks of this course.

SDPs for Max-Cut

$G=(V,E)$

$$\max \ \frac{1}{2} - \frac{1}{2m} \sum_{\{u,v\} \in E} X_{u,v}.$$

$$\text{s.t } \operatorname{diag}(X) = 1$$

$$X \succeq 0$$

$\longrightarrow \boxed{SDP(G)}$

Rounding (Goemans-Williamson '95)

Theorem: There's a poly time randomized algorithm that takes input X: $X \succcurlyeq 0$ & $X_{ii} = 1 \; \forall i$ and outputs an $x \in \{\pm 1\}^n$ s.t. for every $1 \le i, j \le n$,

if $\left(\frac{1}{2} - \frac{1}{2} X_{uv}\right) \geq 1 - \varepsilon$

then $\mathbb{E}_x\left[\frac{1}{2} - \frac{1}{2} x_u \cdot x_v\right] \geq 1 - 2\sqrt{\varepsilon}$

Further: $$\frac{\mathbb{E}_x\left[\frac{1}{2} - \frac{1}{2} x_u x_v\right]}{\frac{1}{2} - \frac{1}{2} X_{uv}} \geq 0.878\ldots$$

Corollary: There's a (c,s) approx. algo for Max-Cut for $c = 1-\varepsilon$ & $s = 1-2\sqrt{\varepsilon}$ for every $\varepsilon > 0$. Further, the approx. ratio of this algo is $\geq 0.878\ldots$

Proof: Let G be a graph and $SDP(G) = c$ with optimal solution X.

Then: $SDP(G) \geq \text{Max-Cut}(G)$.

Why? $SDP(G)$ is a relaxation.
[$X = x^* \cdot x^{*T}$ is "feasible"]

On the other hand, theorem implies that we can find x such that

$$\mathbb{E}_{x} \frac{1}{2} - \frac{1}{2} x_u \cdot x_v \geq 1 - 2\sqrt{\varepsilon}$$

take average of LHS over $\{u,v\} \in E$ of G.

then $\mathbb{E}_{x} \, \mathrm{cut}_G(x) \geq 1 - 2\sqrt{\varepsilon}$

Similar argument for approx. ratio. $\square$

BASIC FACTS (may not prove in lecture)

Lemma (Cholesky Factorization)

For every $X \in \mathbb{R}^{n \times n}$, $X \succcurlyeq 0$, there is a $Z \in \mathbb{R}^{n \times n}$ s.t.

$$X = ZZ^T$$

Proof: $X = \underbrace{U \cdot \Lambda U^T}_{\text{Eigenvalue decomposition}}$

Λ: diagonal matrix

$\underline{X \succcurlyeq 0} \iff \Lambda$ has non-neg diags.

Let $\Lambda^{1/2}$ = entrywise square root of Λ.

Set $Z = (U \cdot \Lambda^{1/2})$.

Then $X = ZZ^T$ $\square$.

$\boxed{\text{Def. (Gaussian).}}$

Std. Gaussian distⁿ: $PDF(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

Std. Gaussian vector: $(g_1, \ldots, g_n)$

independent std. gaussians.

Prop: (Rotation Invariance)

$H \in \mathbb{R}^{n \times n}$, orthogonal (ie. $H^T H = HH^T = I$)

g : std. gaussian vector in $\mathbb{R}^n$.

Then, Hg has same distⁿ as g.

Pf: PDF of $g = \frac{1}{(\sqrt{2\pi})^n} e^{-\frac{\sum_i x_i^2}{2}}$

$$= \frac{1}{(\sqrt{2\pi})^n} \cdot e^{-\frac{\|x\|_2^2}{2}}$$

$\|x\|_2^2 = \|Hx\|_2^2$ (l_2 norm is rot. invariant).

Corollary 1: Let (g_1, g_2) be std. 2D gaussian vector. Then the point $\left(\frac{g_1}{\|g\|_2}, \frac{g_2}{\|g_2\|}\right)$ is uniformly distributed on the unit circle.

Pf: Let $u = (u_1, u_2)$, $v = (v_1, v_2)$ s.t. $u_1^2 + u_2^2 = v_1^2 + v_2^2 = 1$.

Then there's a H, orthogonal s.t. $Hu = v$. By rotation invariance PDF at u = PDF at v. $\square$.

<u>Corollary 2</u>: Let $z_1, z_2 \in \mathbb{R}^n$ be <u>unit</u> vectors s.t. $\beta = \langle z_1, z_2 \rangle$

$g = (g_1, \ldots, g_n)$: n-D std. gaussian.

Then
$$\begin{pmatrix} \langle z_1, g \rangle \\ \langle z_2, g \rangle \end{pmatrix} \sim \begin{pmatrix} g_1 \\ \sqrt{1-\beta^2} \cdot g_2 + \beta \cdot g_1 \end{pmatrix}$$

<u>Proof</u>: There's an orthogonal matrix H s.t.
$$H z_1 = e_1$$
$$H z_2 = \sqrt{1-\beta^2} \cdot e_2 + \beta \cdot e_1$$

use rotation invariance. $\square$

Proof of Theorem

Rounding:

1) Compute Z : $X = ZZ^T$
2) Generate $g = (g_1 \cdots g_n)$ from std. gaussian distⁿ.
3) For each u, set

$$x_u = \text{sign}(\langle g, z_i \rangle)$$

Analysis of Rounding

$g = (g_1, \ldots, g_n)$: std. gaussian vector

Fix $u, v \in V$. then

(X_u, X_v) has same dist^n as $\mathrm{sgn}(\langle g, z_u\rangle, \langle g, z_v\rangle)$

We care about the random variable

$$\frac{1}{2} - \frac{1}{2} x_u \cdot x_v = \begin{cases} 0 & \text{if } X_u = X_v \\ 1 & \text{if } X_u \neq X_v. \end{cases}$$

$\downarrow$

expectation $= \Pr[\text{value} = 1]$.

Thus we are interested in the following elementary question

Question: Let (g'_u, g'_v) be jointly distributed as a 2-D Gaussian with mean 0, $\mathrm{cov} = \begin{pmatrix} 1 & X_{uv} \\ X_{uv} & 1 \end{pmatrix}$

What's the chance that $x_u \neq x_v$?

Lemma (Sheppard's lemma)

Let z_u, z_v be n-dim unit vectors such that $\langle z_u, z_v \rangle = X_{uv} = -1 + \eta$

Let $(g_1 \cdots g_n)$ = std. gaussian vector

$$\Pr[x_u \neq x_v] = 1 - \frac{\sqrt{2\eta}}{\pi} + O(\eta^{3/2})$$
$$\geq 1 - 2\sqrt{\eta} \quad \forall \eta.$$

ASIDE: If X is PSD, $X_{uu} = X_{vv} = 1$ then $|X_{u,v}| \leq 1$. To see why use $w^T X w \geq 0$ for $w = e_u + e_v$ & $w = e_u - e_v$ and rearrange.

Proof: "reduce to 2d geometry".

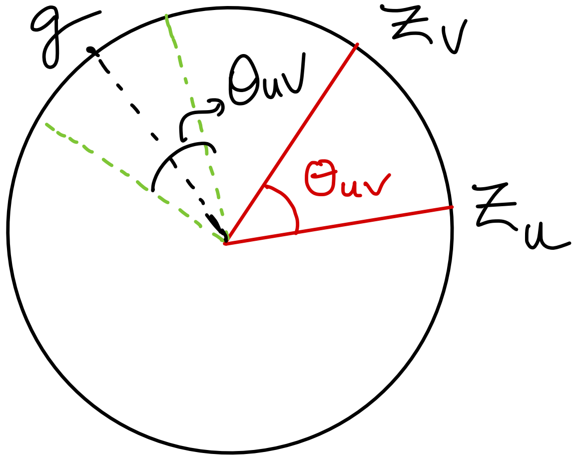
$$Z = (z_1, \dots, z_u, \dots, z_v, \dots, z_n)$$

rows of Z

$g = (g_1, \dots, g_n)$: std. gaussian

Then:

$$\begin{pmatrix} \langle z_u, g \rangle \\ \langle z_v, g \rangle \end{pmatrix} \sim \begin{pmatrix} g_1 \\ (-1+\eta) g_1 + \sqrt{2\eta - \eta^2} \cdot g_2 \end{pmatrix}$$



$$\Pr[X_u \neq X_v]$$

$$= \Pr\left[\mathrm{sign}(g_1) \neq \mathrm{sign}\left((1+\eta)g_1 + \sqrt{2\eta - \eta^2}\cdot g_2\right)\right]$$

$$= \boxed{\frac{2\theta_{uv}}{2\cdot\pi} = \frac{\theta_{uv}}{\pi}}.$$

Thus,

$$\Pr[x_u \neq x_v] = \frac{\theta_{uv}}{\pi}.$$

$$= \frac{\text{arc-cos}(X_{uv})}{\pi}$$

Parameterize $X_{uv} = -(1-\eta)$.

$\text{arc-cos}(-(1-\eta)) = \pi - \text{arc-cos}(1-\eta)$.

$$\text{arc-cos}(1-\eta) = \sqrt{2\eta} + \frac{(2\eta)^{3/2}}{24} + O(\eta^2)$$

plugging in:

$$\Pr[X_u \neq X_v] = 1 - \frac{\sqrt{2\eta}}{\pi} + O(\eta^{3/2})$$

Use calculus/mathematica to minimize

$$\frac{\theta_{uv}}{\pi \cos(\theta_{uv})} \text{ over } \theta_{uv}.$$

minimizing θ_{uv} : -0.69

min value : 0.878.

$\square$.

Can we improve GW?

Yes!

1) For bounded degree graphs, can beat 0.878

2) There is a rounding that does better than GW in some regimes. Can get a better (c,s)-approx. curve.

[O'donnell-Wu].

Even better?

Next time: limitations of GW algo

<u>Future</u>: UGC and "optimality" of the above alg