Semidefinite Programming:

Two different intuitions, both correct:

1) Fundamentally different kind of relaxation:

   vectors instead of fractions

2) Linear Programming + one non-linearity

Definition / Theorem: A symmetric matrix $X \in \mathbb{R}^{n \times n}$ is positive semidefinite (PSD) ($X \succeq 0$) if and only if:

1) All eigenvalues of X are ≥ 0
2) $y^T X y \geq 0 \quad \forall y \in \mathbb{R}^n$
3) $X = V^T V$ for some $V \in \mathbb{R}^{n \times n}$
4) $\forall i \in [n]$ there is some vector $v_i \in \mathbb{R}^n$ s.t.

   $$X_{ij} = v_i \cdot v_j = \langle v_i, v_j \rangle$$

Def: A semidefinite program (SDP) is an LP with the additional constraint that the matrix of variables is PSD

Ex: Variable X_{ij} $\forall i,j \in [n]$.

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} X_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ijk} X_{ij} \leq b_k \qquad \forall k \in [m]$$

$$X_{ij} = X_{ji} \qquad \forall i,j \in [n]$$

$$X = (x_{ij}) \succeq 0$$

"Thm": SDPs can be "solved" in poly time

- requires some "techical niceness" conditions
- additive error ε
- time $\text{poly}(\text{input}, \log \frac{1}{\varepsilon})$

Pf sketch:

Ellipsoid alg.

If X not PSD, $\exists\, y$ s.t. $y^T X y < 0$

$$\Rightarrow \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j X_{ij} < 0$$

Separating hyperplane: $\sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j X_{ij} = 0$

Equivalent:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \langle v_i, v_j \rangle$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ijk} \langle v_i, v_j \rangle \le b_k \qquad \forall k \in [m]$$

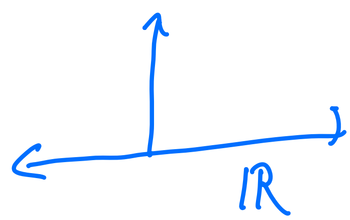
$$v_i \in \mathbb{R}^n$$

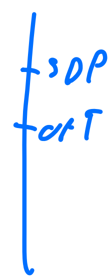
why do this?

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ijk} x_i x_j \le b_k \qquad \forall k \in [m]$$

$$x_i \in \mathbb{R} \qquad \forall i \in [n]$$





Can't solve: quadratic program!

But can solve vector (SDP) relaxation!

Relaxation: Given x feasible for QP, set $v_i = (x_i, \underbrace{0, 0, \ldots, 0}_{n-1})$

LPs: relaxation of ILPs where integer vars → fractional vars

SDPs: relaxation of strict quadratic programs, vars → vectors

strictness required! can't have some linear constraints, some quadratic

<u>LP approach</u>: - Write ILP

- Relax to LP (fractions)

- Solve, round

<u>SDP approach</u>: - Write (strict) quadratic program
- Relax to SDP (vectors)
- Solve, round

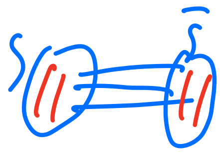
<u>Max-Cut</u>:

Input: - $G=(V,E)$
- $w: E \to \mathbb{R}^+$

Feasible solution: $S \subseteq V$

Objective: $\max \; w(\delta(S)) = \sum_{e \in \delta(S)} w(e)$

$V = [n]$



<u>SDP Approach</u> [Goemans-Williamson '95]:

First: Write strict quadratic program.

$$\max \; \frac{1}{2} \sum_{\{i,j\} \in E} w(i,j)\,(1 - x_i x_j)$$

$$\text{s.t.} \quad x_i^2 = 1 \qquad \forall i \in V$$

$$x_i \in \mathbb{R} \qquad \forall i \in V$$

<u>Thm</u>: This QP is exactly Max-Cut

<u>Pf</u>: Let x feasible QP solution

$x_i^2 = 1 \Rightarrow x_i \in \{-1, 1\} \quad \forall i \in V$

Let $S = \{i : x_i = 1\} \Rightarrow x_i x_j = -1$ if $\{i,j\} \in \delta(S)$

1 if $\{i,j\} \notin \delta(S)$

$$\Rightarrow \frac{1}{2} \sum_{\{i,j\} \in E} w(i,j)(1 - x_i x_j) =$$

$$= \frac{1}{2} \left(\sum_{\{i,j\} \in \delta(S)} w(i,j)(2) + \sum_{\{i,j\} \notin \delta(S)} w(i,j) \cdot 0 \right)$$

$$= \sum_{\{i,j\} \in \delta(S)} w(i,j) = w(\delta(S))$$

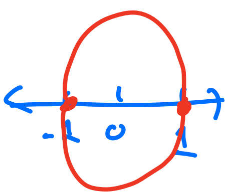
Other direction: let $S \subseteq V$.

Set $x_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{if } i \notin S \end{cases}$

$$\Rightarrow w(\delta(S)) = \frac{1}{2} \sum_{\{i,j\} \in E} w(i,j)(1 - x_i x_j)$$ (same calculation)

Second: can't solve QP, relax to vectors (SDP)

$$\max \quad \frac{1}{2} \sum_{\{i,j\} \in E} w(i,j)(1 - \langle v_i, v_j \rangle)$$

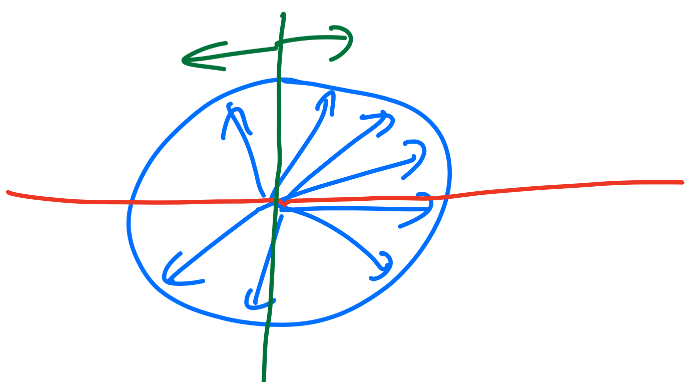


$$\text{s.t.} \quad \langle v_i, v_i \rangle = 1 \quad \forall i \in V$$ $\equiv v_i$ a unit vector

$$v_i \in \mathbb{R}^n \quad \forall i \in V$$

Valid relaxation: given solution x to QP, set

$$v_i = (x_i, \underbrace{0, 0, 0, \ldots, 0}_{n-1})$$



$$\Rightarrow OPT \leq OPT(SDP)$$

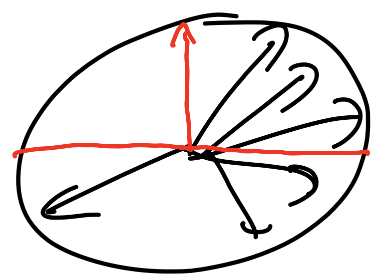
Three: solve SDP, get vectors v_i.

Round each vector to $\{-1, 1\}$, try not to lose too much in objective

<u>Rounding Algorithm</u>: random hyperplane rounding

- Choose $r \in \mathbb{R}^n$ uniformly at random from $\{v \in \mathbb{R}^n : \|v\| = 1\}$ (random unit vector)

  can do by choosing each coordinate independently from $N(0,1)$, rescaling to make unit



- Let $S = \{i \in V : \langle v_i, r \rangle \geq 0\}$
- Return S

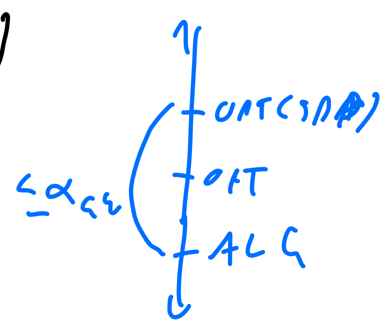
<u>Thm</u>: Random hyperplane rounding is a

$$\alpha_{GW} = \inf_{0 \leq \theta \leq \pi} \frac{2}{\pi} \cdot \frac{\theta}{1-\cos\theta} > 0.87856\text{-approximation}$$

<u>Pf</u>: <u>WTS</u>: $\Pr[\{i,j\} \in \delta(S)] \geq \alpha_{GW} \cdot \frac{1}{2}(1 - \langle v_i, v_j \rangle) \quad \forall \{i,j\} \in E$

$$\Rightarrow E[w(\delta(S))] = E\Big[\sum_{\{i,j\} \in E} w(i,j) \cdot \mathbb{1}[\{i,j\} \in \delta(S)]\Big]$$

$$= \sum_{\{i,j\} \in E} w(i,j) \Pr[\{i,j\} \in \delta(S)]$$



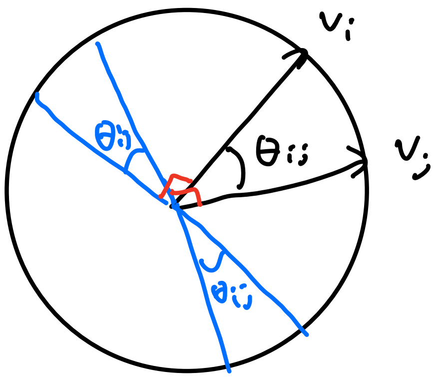
$$\geq \sum_{\{i,j\} \in E} w(i,j)\, \alpha_{GW} \frac{1}{2}\left(1 - \langle v_i, v_j \rangle\right)$$

$$= \alpha_{GW} \cdot OPT(SDP)$$

$$\geq \alpha_{GW} \cdot OPT$$

So look at some $\{i,j\} \in E$

Look at plane P spanned by v_i, v_j



whether $\{i,j\} \in \delta(S)$ determined by projection of r onto P (still uniformly distributed):

From perspective of $\{i,j\}$: choose random unit vector (line through origin in P, $\{i,j\} \in \delta(S)$ iff v_i, v_j on different sides of line

$$\Rightarrow \Pr[\{i,j\} \in \delta(S)] =$$

$$\frac{2\theta_{ij}}{2\pi} = \frac{\theta_{ij}}{\pi}$$

By def of α_{GW}: $\alpha_{GW} \leq \frac{2}{\pi} \cdot \frac{\theta_{ij}}{1 - \cos \theta_{ij}}$

$$\Rightarrow \frac{\theta_{ij}}{\pi} \geq \alpha_{GW} \cdot \frac{1}{2}(1 - \cos\theta_{ij})$$

$$\Rightarrow \Pr[\{i,j\} \in \delta(S)] \geq \alpha_{GW} \cdot \frac{1}{2}(1 - \cos\theta_{ij})$$

Recall (linear algebra or high school trig):

$$\langle a,b \rangle = \|a\| \cdot \|b\| \cdot \cos\theta_{ab}$$

angle b/w a,b

$$\Rightarrow \Pr[\{i,j\} \in \delta(S)] \geq \alpha_{GW} \cdot \frac{1}{2}(1 - \langle v_i, v_j \rangle) \qquad (\|v_i\| = \|v_j\| = 1)$$

Done!

__Thm__ [Håstad '01]: Assuming $P \neq NP$, no α-approximation for Max-Cut with $\alpha > \frac{16}{17} \approx 0.941$

__Thm__ [KKMO '07]: Assuming Unique Games Conjecture, no α-approx for Max-Cut with $\alpha > \alpha_{GW}$