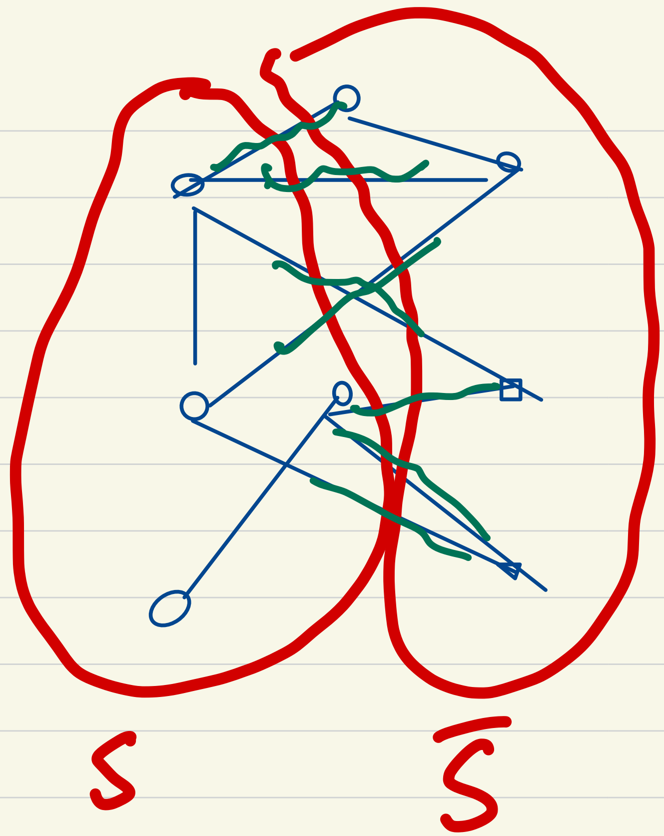
LAST LECTURE

Plan: "SDP" Based Approximation Algo for MaxCut.

MAXIMUM CUT: is NP-hard

INPUT: Graph $G=(V,E)$

GOAL: Find a cut $(S,\bar{S})$ $S \subseteq V$ that maximizes # of crossing edges.



Randomized Alg:

Randomly assign every vertex v to S, or $\bar{S}$.

$cut(S,\bar{S}) = 7$

$\rightarrow$ Every edge (u,v) is cut with prob $1/2$.

$\Rightarrow \mathbb{E}[\text{edges cut}] = 1/2 \cdot (\text{Total \# of edges})$

Max Cut

INPUT: Graph $G = (V, E)$

Goal: Find $(S, \bar{S})$

a partition of vertices

maximizing # of edges cut.

Graph G

Semidefinite Program

optimal Sol $\Rightarrow$ n Vectors $v_1 \dots v_n$

Rounding Algo

Quadratic Program:

Variable $x_i = \begin{cases} +1 & i \in S \\ -1 & i \in \bar{S} \end{cases}$

for each vertex i

Constraints

$$x_i^2 = 1 \iff x_i = \pm 1$$

Objective Function

$$\sum_{\substack{\text{edges} \\ (i,j) \in E}} \left(\begin{array}{c} \text{edge } i,j \\ \text{is cut} \end{array} \right) = \frac{(x_i - x_j)^2}{4}$$

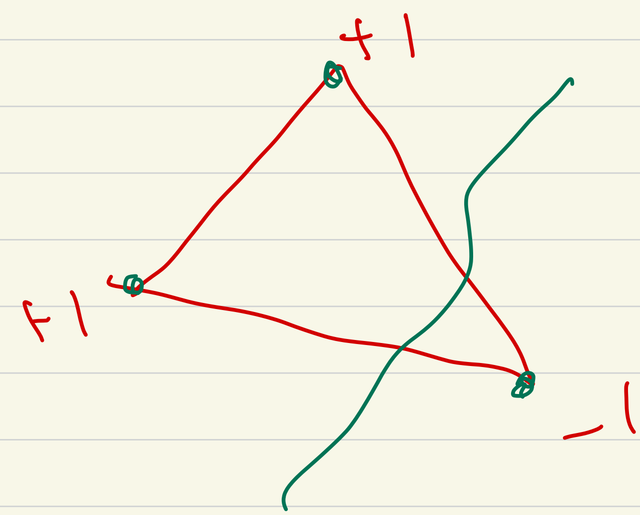
Max Cut

INPUT: Graph $G = (V, E)$

Goal: Find $(S, \bar{S})$

a partition of vertices

maximizing # of edges cut.



$$\frac{(x_i - x_j)^2}{4}$$

x_i	x_j	$(x_i - x_j)^2/4$
+1	+1	0
-1	-1	0
+1	-1	1
-1	+1	1

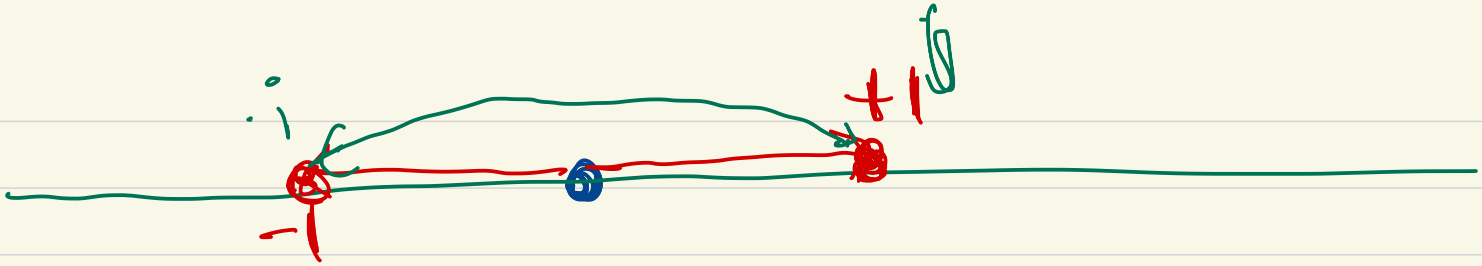
QP

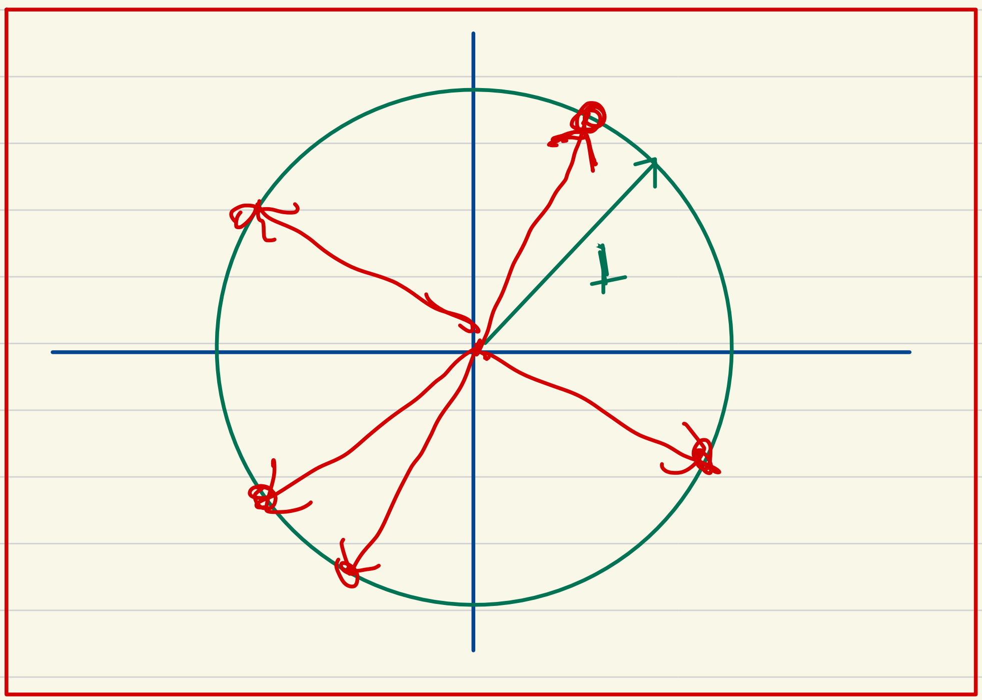
Maximize $\displaystyle\sum_{(i,j)\in E} (x_i - x_j)^2/4$ = # edges cut

Subj to $\rightarrow$ $x_i^2 = 1$ for all $i \in V$.

QP exactly captures MaxCut

But solving QP is NP-hard.





2-dim

← Each x_i is a unit vector

QP_2 is NP-hard

Maximise $\sum_{(i,j)} \|x_i - x_j\|^2$

subj $\|x_i\|^2 = 1$

$\|v\|$ = length of vector

QP_n can be solved in polynomial time (!)

$QP_n \iff$ Semidefinite Program

$\underline{QP_n:}$ Maximize $\sum_{(i,j)\in E} \|v_i - v_j\|^2$

||

Semidefinite Program (SDP) $\underbrace{\|v_i\|^2 = 1}$ for each vertex $i \in \{1..n\}$

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

SDP can be optimally solved in polynomial time

$v_i = (v_i^{(1)}, v_i^{(2)} \ldots v_i^{(n)})$

$$\|v_i - v_j\|^2 = (v_i - v_j)\cdot(v_i - v_j)$$

$$= v_i \cdot v_i + v_j \cdot v_j - 2\, v_i \cdot v_j$$

$$\|v_i\|^2 = \boxed{\| \quad = v_i \cdot v_i}$$

$$\|v_i\|^2 = \sum_j \left(v_i^{(j)}\right)^2$$

What is an SDP?

Variables: n vectors in n-dimensions

$$v_i \circ v_j = \sum_a v_i^{(a)} \cdot v_j^{(a)}$$

Constraints: Linear constraints on the dot products $(v_i \circ v_j)$

Objective Function: Minimize / Maximize linear function of dot products

$$K = \{ \text{Set of } n \times n \text{ matrices } M \quad \text{all eigenvalues}(M) \geq 0 \} \in \mathbb{R}^{n^2}$$

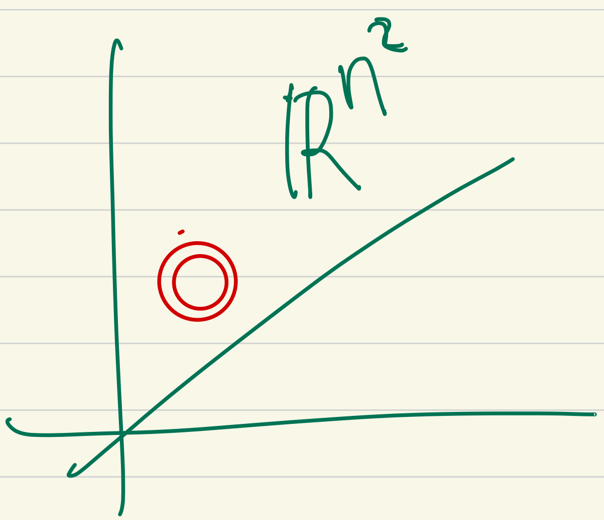
positive semidefinite matrices

$$\|$$

Convex Set

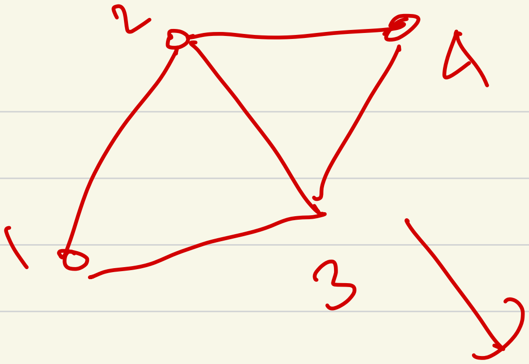
Lemma:

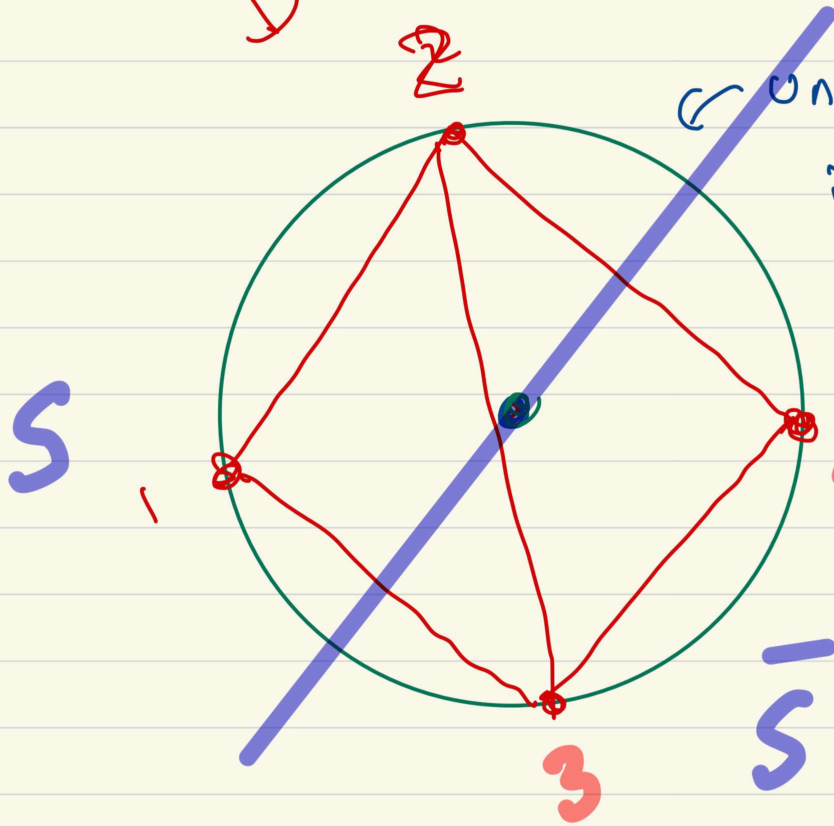
M is a positive semidefinite matrix $\iff$ $M_{ij} = v_i \cdot v_j$ $\exists$ vectors so that



$$(M) = V V^T$$

$$\left[\quad \right] \left[\quad \right]$$





← unit sphere in n-dimension

← Maximizes total squared distance

Randomize Rounding:

1) Pick a random hyperplane through the origin

$$\text{SDP-OPT} = \sum_{(i,j)\in E} \|v_i - v_j\|^2 \gg \text{Integer Max Cut.}$$

Claim: $\Pr[(i,j) \text{ is cut}] \geq (0.878) \cdot \|v_i - v_j\|^2$

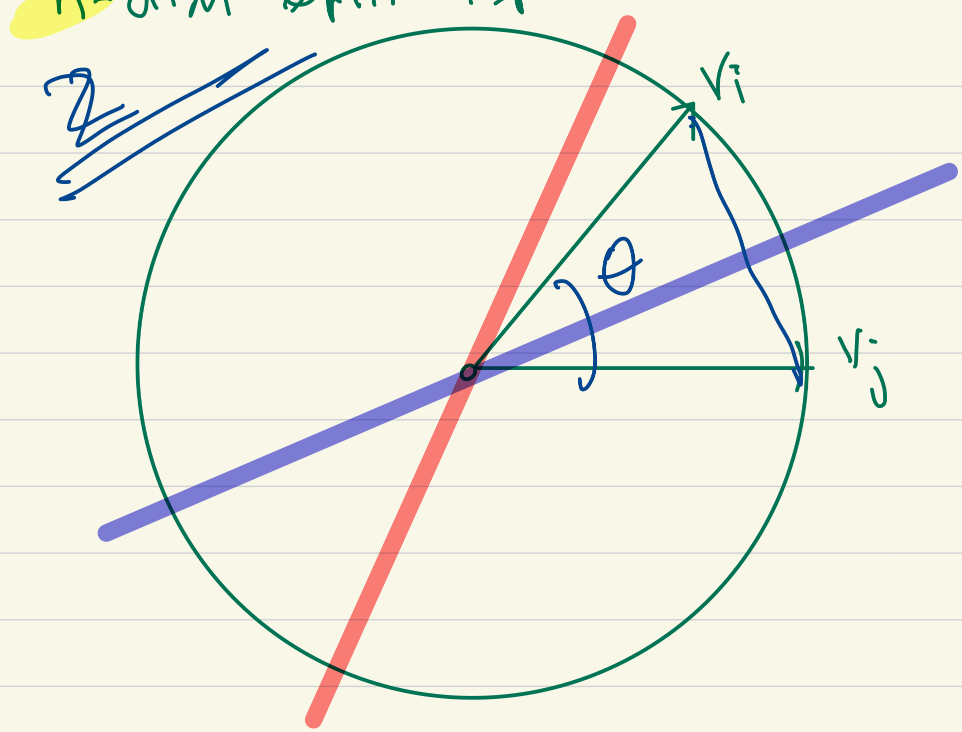
$= f(\theta)$ $\quad$ $= g(\varepsilon)$

Claim $\Rightarrow$ $\mathbb{E}[\text{size of cut}] \geq 0.878 \cdot \sum_{(i,j)} \|v_i - v_j\|^2$

$$= 0.878 \cdot \text{SDP-OPT}$$

$$\geq 0.878 \cdot \text{OPT}$$

n-dim unit sphere



$$\Pr[\, i \text{ and } j \text{ are separated} \,] = \frac{\theta}{\pi} \quad \pi = \text{total angle}$$

$$= \frac{\theta^\circ}{180^\circ}$$

Look at plane containing v_i and v_j

$$\|v_i - v_j\|^2 = 2 - 2\cos\theta$$

$$\frac{\Pr[i, j \text{ are separated}]}{\|v_i - v_j\|^2} = \frac{\theta/\pi}{2 - 2\cos\theta}$$

≥ 0.878 for all angles θ

$$\Pr[i, j \text{ are separated}] \geq \underline{\underline{(0.878)}} \cdot \|v_i - v_j\|^2$$