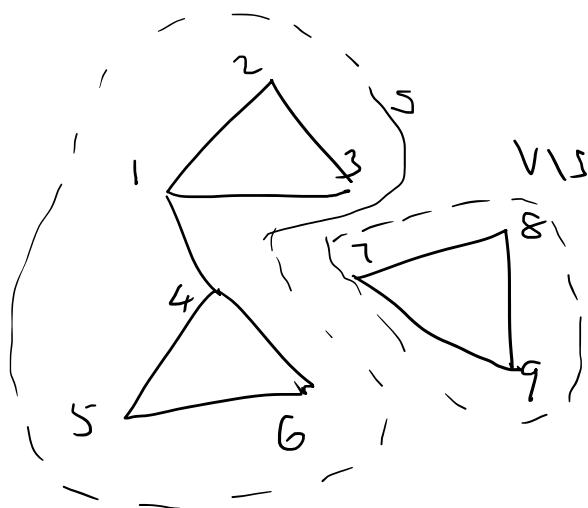
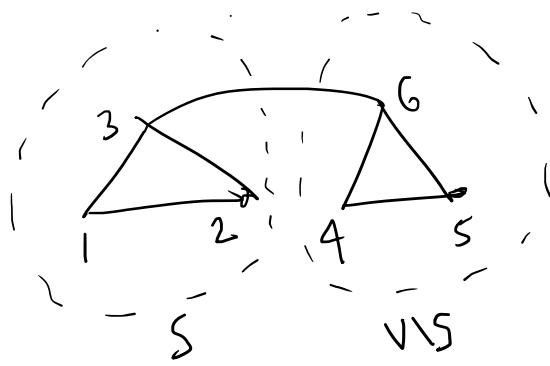


MIN CUT

Given  $G = (V, E)$  unweighted, undirected graph  
 Partition into  $(S, V \setminus S)$  to minimize  
 # edges "cut" (or) crossing  $(S, \bar{S})$



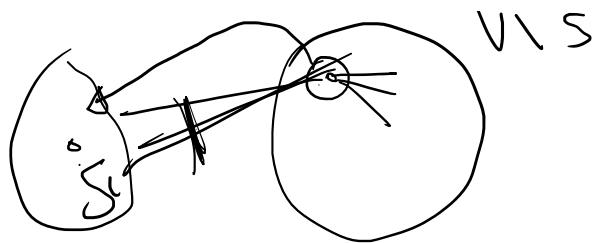
Q: How to solve it?

Idea ① : LP formulation?  
 ↳ We'll get to this

Idea ② : Randomly put each vertex  
 ↴ in  $S$  or  $\bar{S}$   
 UNCLEAR if any good!

Idea ③

Greedy : Include  $v$  from  $V \setminus S$   
 with max # edges to  $S$ .



Idea ④ "local Search"



if try shifting any  
v from S to V \ S  
(or vice versa)

the cut improves,  
do it

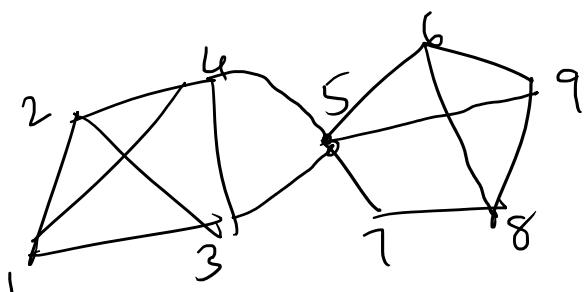
Idea ⑤

Karger's Randomized contraction Algorithm :-

OPERATION

Graph Contraction ?

$$G_1 = \text{graph} \quad G_1 = (V, E)$$

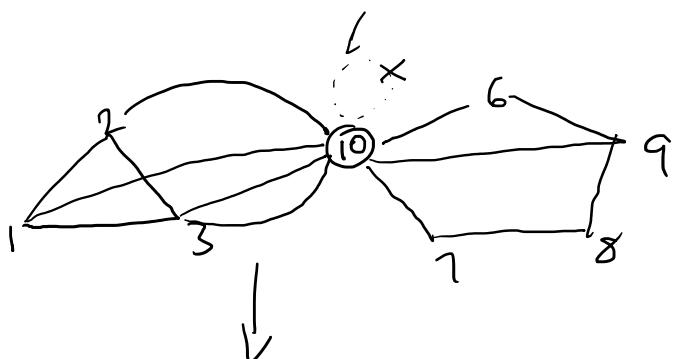


↑ let u & v be 2 vertices

let  $u$  &  $v$  be 2 vertices  
 $G_r \setminus \{u, v\}$  = contraction of  $G_r$   
 by "fusing  $u$  &  $v$ "

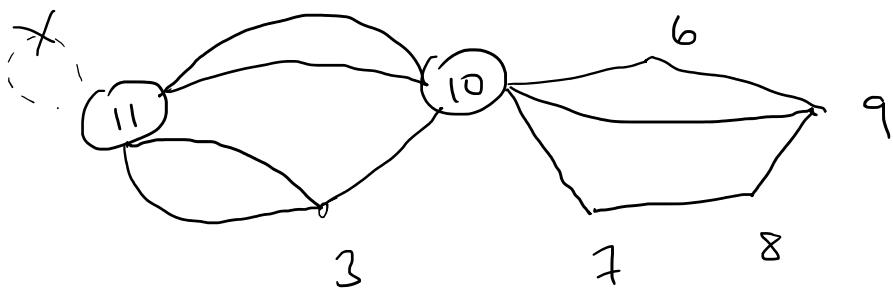
let's say we contract 4 & 5

let 10 be the contracted/fused vertex.



remove self-loop if any.

contract 1 & 2



Imp: retain duplicate/parallel edges,  
remove self-loop (if any)

### KARGER ALGO

if  $G_r = 2$  vertices =  $\{u, v\}$   
Min Cut

=  $u$  on one side  
 $v$  on other side

All parallel edges b/w  $u$  &  $v$  will be cut.

If  $|G| > 2$  vertices:

Pick a uniformly random edge  
and contract it !



With reasonable prob ( $\geq \frac{1}{n^2}$ ),  
the resulting cut is MINIMUM !!

Overall  
repeat  $\mathcal{O}(n^2 \log n)$  times and pick  
the best

↓  
Min cut is in P [poly-time solvable]  
let's look @ LP formulation [Idea ①]

## Variables

$x_u = 0 / 1$  to denote if  
 $u \in S$  or  $V \setminus S$

What about edges? How to determine  
 if edge is cut or not?

$y_{uv}$  for every edge  $(u, v) \in E$

Want  $y_{uv} = 0$  if  $e$  is not cut  
 $1$  if  $e$  is cut.

$$\text{Min } \sum_{(u,v) \in E} y_{uv}$$

$$0 \leq x_u \leq 1 \quad \forall (u, v)$$

Want to encode

$$y_{uv} \geq 1 - \{x_u + x_v\}$$

with linear constraint

Attempt

$$y_{uv} \geq |x_u - x_v|$$

is actually a linear constraint  
 because it is equivalent to

$$\begin{cases} y_{uv} \geq (x_u - x_v) \\ y_{uv} \geq -(x_u - x_v) \end{cases}$$

$$\left\{ \begin{array}{l} y_{uw} \geq 0 \\ y_{uv} \geq -(x_u - x_v) \end{array} \right.$$

LP

Min

$$\sum_{(u,v) \in E} y_{uv}$$

$$\begin{cases} x_u \geq 0 \\ x_u \leq 1 \end{cases} \quad \forall u \in V$$

tries to capture  
 $y_{uv} \geq \min(x_u, x_v)$

$$\begin{cases} y_{uv} \geq (x_u - x_v) \\ y_{uv} \geq - (x_u - x_v) \end{cases} \quad \forall u, v$$

Lemma ①

LP is a valid relaxation of  
 Min Cut problem

$\Rightarrow$  if  $(x^*, y^*)$  is an optimal soln  
 then  $\sum_{uv \in E} y_{uv}^* \leq \text{OPT}(\text{Min Cut})$ .

Now, we'll need to "round" it, since  
 it might be fractional

Idea ①

Use threshold (0.5)

If  $x_u \geq 0.5$ , put  $u \in S$

< 0.5 put in V<sub>S</sub>

Idea ② Think of  $X_u$  as probability of being in S

We'll use Idea ③, a combination of ① & ②.

[  
Alg ;  
① choose random  $\alpha$  between 0 & 1.  
② For all  $u \in V$ ,  
put  $u$  in S if  $X_u^* \geq \alpha$ .  
]

Need to show

$E[\# \text{ cut edges}]$  is small.

let

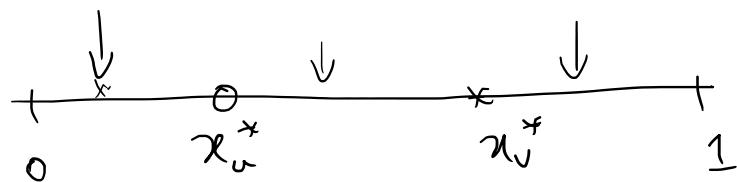
$Z_{u,v}$  = random variable  
= 1 if  $u, v$  is cut  
= 0 otherwise.

$$\begin{aligned} E[\# \text{ edges cut}] &= E\left[\sum_{u,v \in E} Z_{uv}\right] \\ &= \sum E[Z_{uv}] \leftarrow \text{linearity} \end{aligned}$$

$(u, v) \in E$

Expectation

What's  $E[Z_{uv}]$  ?



$$E[Z_{uv}] = \Pr[(u, v) \text{ is cut}]$$

$$= \Pr[u \& v \text{ are on diff sides}]$$

$$= \Pr[\alpha \in (x_u^*, x_v^*)]$$

$$= |x_u^* - x_v^*|.$$

$$\leq y_{uv} \quad (\text{from LP constraint}).$$

$$\Rightarrow E[\# \text{ edges cut}] \leq \sum_{(u, v) \in E} y_{uv} \leq \text{OPT}(\text{Min Cut})$$

$\Rightarrow$  It's random, can we make it deterministic  
OPT = 5

Alg Output  $(S_1, V \setminus S_1)$  up 0.25. 4 }  
~~OPT~~

$(S_2, V \setminus S_2)$  up 0.5. 6 }

~~CANT~~ .. n .. n .. 4 }

~~CANT  
HAPPEN!~~

(1L)  $\cup$  " 4  
 $(S_3, V \setminus S_3)$  up 0.25

$\Rightarrow$  All events have to yield value 5

Issue pointed out:

All  $x_u$  can be the same }  
Up will have }  
value 0 ]

How to fix this?

Ans:

"Gives 2 vertices in OPT Min Cut"  
 $u_L, u_R$   
which are on  
different sides,

and enforce

$x_{u_L} = 0, x_{u_R} = 1$   
in the Up

{ Solve many different LPs  
with different choices of }  
 $u_L$  &  $u_R$   
and output the non-degenerate one }

TOMORROW:

Max Cut, Max # edges crossing cut.



MinCut is in P (exact polytime algorithm)

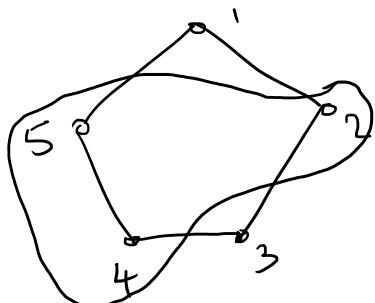
- What about MaxCut?
- Ans: NP-complete, resort to Approximation Algorithms

Qn:

Given  $G = (V, E)$ , partition  $V = S \cup \bar{S}$   
such that  $|E(S, \bar{S})|$  is maximum.

where  $E(A, B) = \{(u, v) \in E \text{ s.t. } u \in A, v \in B\}$   
for disjoint sets  $A, B$

Example



What is the Max Cut?

# edges cut = 4

- Why is this Max Cut?

Because if I cut with value 5,  
then it cuts all edges  
 $\Rightarrow$  Graph is bipartite

Gr here has an odd cycle.

---

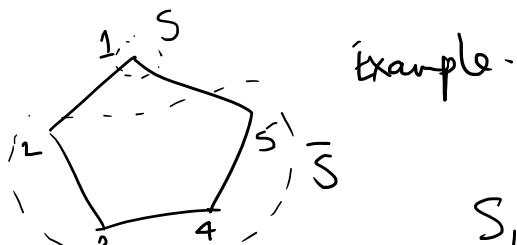
Max Cut for bipartite graphs is easy

just need to recover the partition.

Q: What about general graphs?

Idea ①: Local Search.

- Start with an arbitrary  $(S, \bar{S})$  partition.
- If by moving a vertex from one side to the other, we can increase # edges cut, we'll do it.



example.

$$S_1 = \{1\} \quad \bar{S}_1 = \{2, 3, 4, 5\}$$

$$\# \text{edges cut} = 2.$$

Let's swap 3 over.

$$S_2 = \{1, 3\} \quad \bar{S}_2 = \{2, 4, 5\}$$

$$\# \text{edges cut} = 4.$$

↳ LOCALLY OPTIMAL, NO SINGLE SWAP CAN IMPROVE OBJECTIVE VALUE.

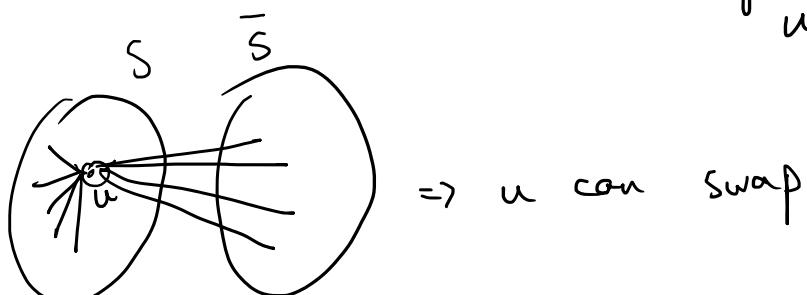
How good is this algo for general graphs?

Ans: Quite good! Final solution  $(S, \bar{S})$

Ans : Quite good! Final solution  $(S, \bar{S})$   
 cuts  $\geq \frac{m}{2}$  edges of  $|E| = m$ .

Proof

Let  $(S, \bar{S})$  be the final sol'.  
 If  $u$ ,  $\#$  edges incident to  $u$  crossing  
 the partition  
 $\geq \#$  edges incident to  $u$   
 within  $u$ 's side.



$$\Rightarrow 2(\# \text{ edges from } u \text{ crossing the partition}) \geq \delta(u)$$

$$\Rightarrow \boxed{\# \text{ edges from } u \text{ crossing} \geq \frac{\delta(u)}{2}}$$

SUM OVER ALL  $u$

$$2|E(S, \bar{S})| \geq \frac{\sum \delta(u)}{2}$$

$$= m.$$

$$\Rightarrow \boxed{|E(S, \bar{S})| \geq \frac{m}{2} \geq \frac{OPT}{2}}.$$

$\left\{ \begin{array}{l} \frac{1}{2} - \text{approximation} \\ \uparrow \downarrow \end{array} \right\}$   
 ... partition problems,

?

$\left[ \frac{1}{2} - \text{approx.} \right]$   
 For maximization problems,  
 if  $c$  approx.  $\rightarrow$   
 $\text{Val}(Lg) \geq c \cdot \text{Val(OPT)}$ .  
 for  $c \leq 1$  (larger  $c$  is better).

Q: Can we do a better analysis?  
 Ans: I think not 😞 (try showing?).

If we're trying to show that  
 local search can do  $\geq c \cdot m$   
 for  $c > \frac{1}{2}$ , that  
 may not be possible.  
 (complete graph).

Ans 2: Are there graphs where OPT-Cut  
 very close to  $m$ , but  
 local search close to  $m/2$ ?

Was long standing open problem to beat  
 factor  $\frac{1}{2}$  for Max Cut.

Seminal work [Goemans - Williamson]  
 which try to characterize Max Cut  
 linear programming

using linear programming

FUNNY: (LP will have infinitely many constraints)  
but still "solvable" in poly-time

$$\left\{ \begin{array}{l} \text{Max } \sum_{(u,v) \in E} y_{uv} \\ 0 \leq x_u \leq 1 \quad \forall u \\ y_{uv} \geq |x_u - x_v| \quad \forall u, v \\ \text{for min cut we used this} \end{array} \right.$$

expressible as linear constraint

- LP can cheat in above formulation
- All  $y_{uv}$  can be set to one!

whereas we need

$$\left\{ \begin{array}{l} \text{Max } \sum y_{uv} \\ y_{uv} \leq |x_u - x_v| \quad \forall u, v \\ 0 \leq x_u \leq 1 \quad \forall u \end{array} \right.$$

Problem 😞 Not a linear constraint

$$y_{uv} \leq \max(x_u - x_v, -(x_u - x_v))$$

is not linearly expressible

whereas

$$y_{uv} \geq \max(a, " )$$

is expressible as

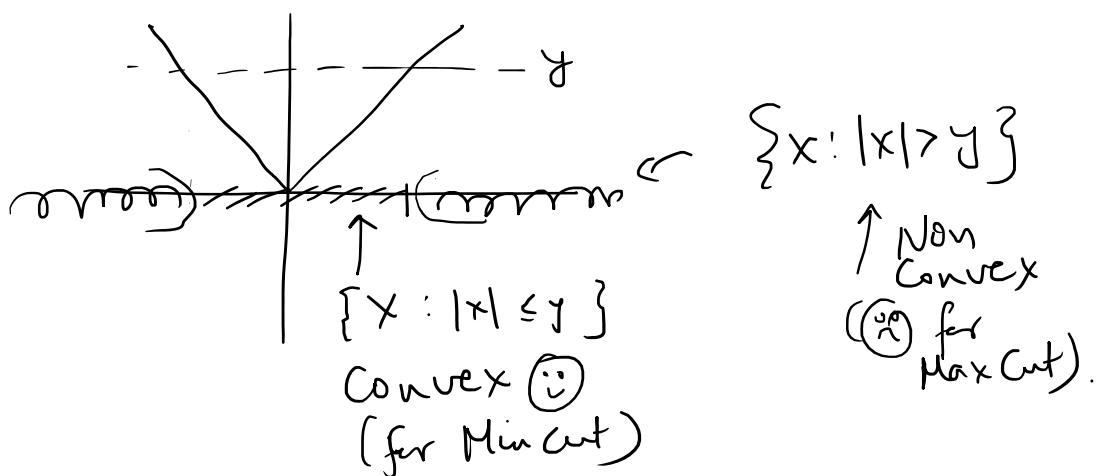
$$y_{uv} \geq x_u - x_v \quad \text{and}$$

$$y_{uv} \geq -(x_u - x_v),$$

Another way to see this issue :-

for any fixed  $y$   
 $|x| \leq y$  is a convex set

$|x| \geq y$  is not a convex set




---

Can we "express"  $y_{uv} \leq |x_u - x_v|$   
 (or approximately)  
 using a linear program?

using a linear program?

Human...

Maybe not fully well, but lets also try  $y_{uv} \leq (x_u - x_v)^2$   
This also captures  $\underline{1\{x_u \neq x_v\}}$

Really, we're trying to capture  $\underline{1\{x_u \neq x_v\}}$  using some nice constraints

To make life easier, let us use a slightly different notation.

don't think of  $x_u$  as 0/1 variables

Let's think of them as ±1 variables

---

Max Cut (exact formulation).

$$\begin{aligned} & \text{Max } \sum_{(u,v) \in E} \frac{1}{2}(1 - y_{uv}) \\ & x_u \in \{\pm 1\} \quad \forall u \\ & y_{uv} = x_u \cdot x_v \quad \forall u, v \end{aligned}$$

↓  
∴  $\vdash S+1?$  is equivalent

$\downarrow$   
 $x_u \in \{-1\}$  is equivalent  
 to  $x_u^2 = 1$ , (ie)  $y_{uu} = 1$ .

$$\Rightarrow \text{Max } \sum_{(u,v) \in E} \frac{1}{2}(-y_{uv})$$

$$\begin{cases} y_{uu} = 1 & \forall u \\ y_{uv} = x_u x_v & \forall u, v \end{cases}$$

Want to express this using  
linear constraints.

What sort of constraints do these  $y_{uv}$ 's  
satisfy?

We know  $(x_u + x_v)^2 \geq 0$

$$x_u^2 + x_v^2 + 2x_u x_v \geq 0$$

) we can enforce

$$y_{uu} + y_{vv} + 2y_{uv} \geq 0$$

$$\Rightarrow y_{uv} \geq -1$$

$\because y_{uu} = 1$

Similarly

$$(x_u - x_v)^2 \geq 0$$

$$y_{uu} + y_{vv} - 2y_{uv} \geq 0$$

$y_{uv} \leq 1$

These are nice linear constraints  
in the  $y$  variables

in the  $y$  variables  
😊

$\frac{Q_N}{}$  what is a nice family of linear constraints we can impose on  $y_{uv}$ ? which the  $\bullet$  unknown soln satisfies).

### SEMITDEFINITE PROGRAMMING

Consider any real vector  $(c_1, c_2, \dots, c_n)$   
 $c_i \in \mathbb{R}$

let's look at  $(\sum c_i x_i)^2 \geq 0$   $\Leftarrow$  true always

$$\Rightarrow \sum_{i,j} c_i c_j x_i x_j \geq 0$$

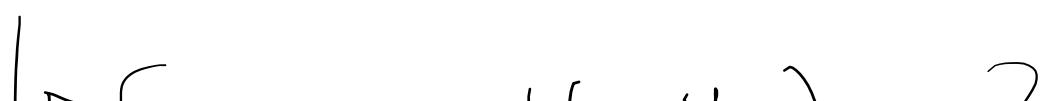
↑ ordered pairs (including  $i=j$ )

$\Rightarrow$  We can enforce

$$\boxed{\sum_{i,j} c_i c_j y_{ij} \geq 0}$$

Linear constraint on the  $y$  variables

Here is a relaxation for Max Cut



$\rightarrow \left\{ \begin{array}{l} \text{Max } \sum_{(i,j) \in E} \frac{1}{2}(1 - y_{ij}) \\ \text{st: } y_{ii} = 1; \\ \forall c = (c_1, \dots, c_n); \sum_i c_i^2 y_{ii} + \sum_{j \neq i} c_i c_j y_{ij} \geq 0 \\ \uparrow \forall (i,j); y_{ij} = y_{ji} \text{ (ordered)} \end{array} \right\}$

Vertices are labeled  $1, 2, \dots, n$ .

Linear Program with  $\infty$  many constraints !!!

Lemma

If  $(S, \bar{S})$  is any cut with cut value  $v$  then there is a feasible soln to above LP with obj value  $v$ .

This LP is actually called a semi-definite program

How to solve the LP if it has infinitely many constraints?

② Given a "claimed"  $\{y\}$  solution how do we even check if it is a feasible solution to LP?

Idea for ②  
express  $y_{ij}$  values as a matrix

$$Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \vdots & & & \\ y_{nn} & y_{n1} & \dots & y_{nn} \end{bmatrix}$$

① Then we can easily check  
 $y_{ii} = 1$  and  
 $y_{ij} = y_{ji}$  in  $n^2$  time.

② How to check  $\left( \sum_i c_i^2 y_{ii} + \sum_{i \neq j} c_i c_j y_{ij} \right) \geq 0$

let's imagine  $c$  is a column vector

$$\boxed{\text{② is equivalent to } c^T Y c \geq 0 \quad \forall c.}$$

$\left( \begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_n \end{array} \right)$

A real-symmetric matrix  $Y$  is called a "positive-semidefinite matrix" if  $c^T Y c \geq 0 \quad \forall c \in \mathbb{R}^n$

Checking ⑦  $\Leftrightarrow$  checking if  $Y$  is PSD or not.

### THEOREM

Real symmetric Matrix  $Y$  is PSD:  
[ 3 equivalent forms]

- ①  $c^T Y c \geq 0 \quad \forall c \in \mathbb{R}^n$
- ② All eigenvalues of  $Y$  are  $\geq 0$
- ③ There exist vectors  $w_1, w_2, \dots, w_n$   
such that  $Y_{ij} = \langle w_i, w_j \rangle$ .

Given  $A \in \mathbb{R}^{n \times n}$ ,  $(\lambda, v)$  is eigenvector

If

$$Av = \lambda v$$

Now back to checking if a given  $Y$  satisfies LP or not?  
(simply check if all eigen values  $\geq 0$ )

Checking feasibility is ok, but how  
to solve this LP [called a semi-definite  
program]?

$$\text{Max } \sum \frac{1}{2}(1 - y_{ii})$$

can be avoided because with  $C^T C \succ 0$

$C_i, j \in \mathbb{R}$

$$Y_{ii} = 1 + \epsilon_i \\ Y_{ij} = y_{ji} + (i, j)$$

- equivalent to  $\gamma \succ 0$  [ $\gamma$  is a PSD matrix]
- $\#C^T C \succ 0$
- equivalent to all eigs of  $\gamma \succ 0$
- equivalent to  $\exists w_i$  vectors s.t.  $y_{ij} = \langle w_i, w_j \rangle$ .

Beautiful Technique for solving  
Very Large LPs

## Ellipsoid Method / Separation Oracle

Given an LP, and a "claimed" solution for the LP,  
if there is an algorithm which can efficiently tell if the claimed soln is feasible or not

then we can actually solve the LP efficiently !!,  
and "almost optimally"  
(Negligible error)  
 $\dots x = 1 / \text{variables}$

(Negligible error)  
Efficient poly. time of check (#variables)

OK we have solved the LP.  
Gotten ourselves a  $\gamma$  matrix and  
 $w_i$  vectors s.t.  $\gamma_{ij} = \langle w_i, w_j \rangle$ .

How to get a cut from this?

Next class |

ROUNDING ACCO :-

let  $\gamma^*$  be the optimal SDP soln.

$$\text{SDP Obj} = \sum \frac{1}{2}(1 - \gamma_{ij}^*)$$

From  $\gamma^*$ , (because  $\gamma^* \gamma_0$  is PSD)

we can recover vectors

$$w_1^*, w_2^*, \dots, w_n^* \in \mathbb{R}^n$$

such that  $\gamma_{ij}^* = \langle w_i^*, w_j^* \rangle$ .

Moreover these are unit vectors

$$\text{b/c } 1 = \gamma_{ii}^* = \langle w_i^*, w_i^* \rangle = \|w_i^*\|^2$$

**NOTE** If  $w_i^*$  vectors were in 1 dimensional space, they would be  $\pm 1$  scalars and  $\gamma^* = w_i^* w_j^*$  would hold

$$\text{SDP objective} \leftarrow \frac{1}{2} \sum_{i,j} (1 - \langle w_i^*, w_j^* \rangle)$$

$$\text{SDP objective} \leftarrow \frac{1}{2} \sum_{(i,j) \in E} (1 - \langle w_i^*, w_j^* \rangle)$$

all  $w_i^*$  are unit vectors.

Real goal in ROUNDING

How to convert high-dimensional vectors into 1 dimension while roughly preserving inner products/distances?

Idea Random projection!

2 view points of this algo (equivalent)

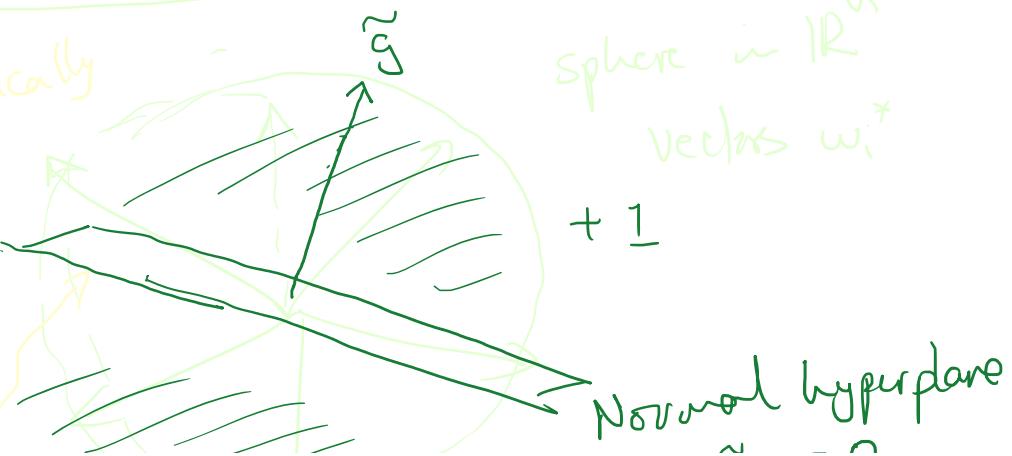
View ①

Pick  $\tilde{g} \in \mathbb{R}^n : (g_1, g_2, \dots, g_n)$  random gaussian vector

let  $x_i = 1$  if  $\langle w_i^*, \tilde{g} \rangle \geq 0$   
 $= -1$  otherwise.

Geometric View

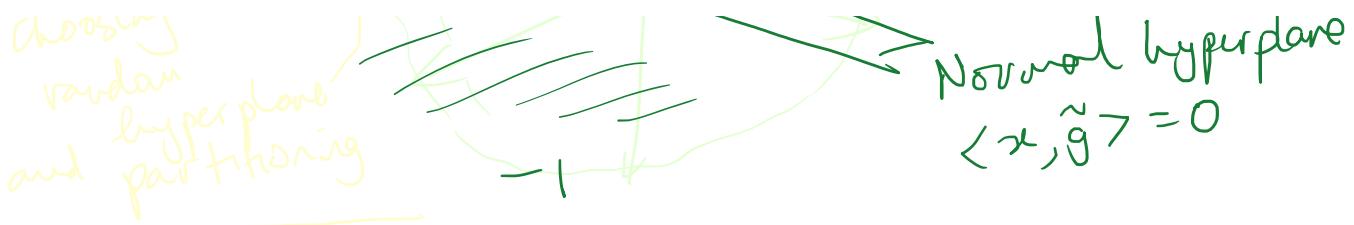
b/c  $\tilde{g}$  is spherically symmetric  
 it is same as choosing random ...



sphere in  $\mathbb{R}^n$   
 vectors  $w_i^*$

+ 1

Normal hyperplane



### Overall Alg

- Solve SDP Optimally
- From  $Y^*$ , recover  $w_1^*, w_2^*, \dots, w_n^*$
- Sample random gaussian vector  
 $\tilde{g} = (g_1, g_2, \dots, g_n)$
- Set  $x_i = 1$  if  $\langle w_i^*, \tilde{g} \rangle \geq 0$ , -1 else
- Output  $S = \{i : x_i = 1\}$  as cut.

Analysis:  
Fix  $(i, j)$ :

SDP is cutting the edge to extent of

$$\frac{1}{2}(1 - \langle w_i^*, w_j^* \rangle)$$

$$= \frac{1}{2}(1 - \cos\theta).$$



Our algo cuts  $(i, j)$  whenever the random hyperplane separates  $w_i^*$  &

Our algo cuts  $\mathbb{R}^n$   
 Random hyperplane separates  $w^*$  &  
 $w_j^*$

On if we choose a random  
 hyperplane, what's the  
 probability that it separates  
 $w^*$  and  $w_j^*$

$$\Pr[(i,j) \text{ is cut}] = \frac{\theta}{\pi}.$$

$$\textcircled{1} \quad \text{Our prob. of cutting edge} = \frac{\theta}{\pi}$$

$$\textcircled{2} \quad \text{SPP obj.} = \frac{1}{2}(1 - \cos)$$

### ANALYSIS ON :

How small can  $\textcircled{1}$  be compared  
 to  $\textcircled{2}$ ?

minimize	function	$\frac{2x}{\pi(1 - \cos(x))}$
	domain	$0 \leq x \leq \pi$

Global minimum:

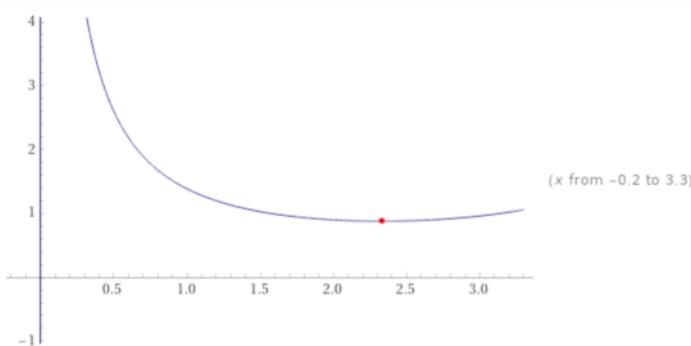
[More digits](#)

[Exact form](#)

[Step-by-step solution](#)

$$\min \left\{ \frac{2x}{\pi(1 - \cos(x))} \mid 0 \leq x \leq \pi \right\} \approx 0.878567 \text{ at } x \approx 2.33112$$

Plot:



$\Rightarrow \textcircled{1} \geq 0.878567 \textcircled{2}$   $\neq 0$

$$\begin{aligned}
 &\Rightarrow \textcircled{1} \geq 0.878 \textcircled{2} + 0 \\
 \Rightarrow E[\# \text{cut edges}] &= \sum_{(i,j) \in E} P\{(i,j) \in \text{cut}\} \\
 &\geq \sum_{(i,j) \in E} 0.878 * \frac{1}{2}(1 - \omega_i^* \omega_j^*) \\
 &= 0.878 \sum_{(i,j) \in E} \frac{1}{2}(1 - \langle \omega^*, \omega_j^* \rangle) \\
 &= 0.878 \sum_{(i,j) \in E} \frac{1}{2}(1 - y_i^* y_j^*) \\
 &= 0.878 \underline{\text{SDP OPT}}.
 \end{aligned}$$

④ Back to how to solve the SDP?

Ellipsoid Algorithm :-

Block box way to convert feasibility  
to optimization

SEPARATION ORACLE } Given a very large LP/SDP and a  $\tilde{y}$   
 A is a sep. oracle if it can  
 efficiently tell if  $\tilde{y}$  is feasible  
 for LP/SDP (OK) Prove why  
 $\tilde{y}$  is not feasible, (ie) output some  
 constraint of LP/SDP which  
 $\tilde{y}$  does not satisfy.

$\check{Y}$  does not satisfy.

## Ellipsoid Algo

If  $\exists$  a separation oracle, then the said LP/SDP can be optimally\* solved (up to any desired accuracy)

Q: What's the separation oracle for Max cut?

Ans: just the eigen solver!

Given the  $\tilde{Y}$  matrix, compute all eigen values and eigenvectors

If all  $\lambda_i$  are  $> 0$ , say fearable (also if all  $\tilde{Y}_{ii} = 1$ ),

If  $\exists \lambda_i < 0$ , with eigenvector  $v_i$ ,

Output violated constraint

$$v_i^T \tilde{Y} v_i > 0$$

This is enforced in the LP, but the given  $\tilde{Y}$  matrix won't satisfy

$$\tilde{v}_i^T \tilde{Y} v_i = v_i^T \lambda_i v_i = \lambda_i \|v_i\|^2 \leq 0$$



