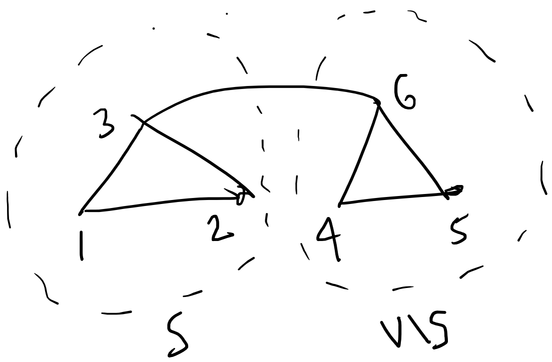
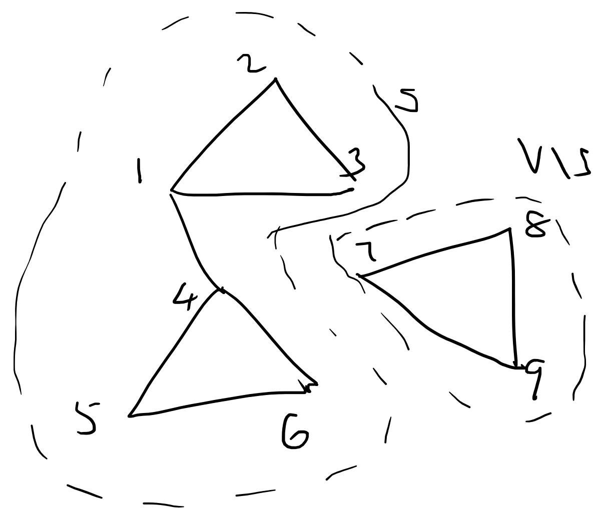
Min-Cut, Max-Cut and SDPs

19 April 2021 10:03

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 ?

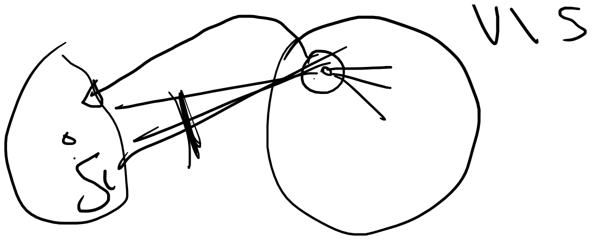
↳ We'll get to this

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"

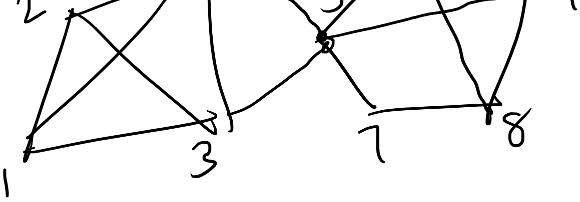

if by shifting any $v$ from $S$ to $V \setminus S$ (or vice versa) the cut improves, do it.

Idea (5)
Karger's Randomized contraction Algorithm :-

OPERATION

Graph Contraction ?

$G$ = graph $G = (V, E)$




Let $u$ & $v$ be 2 vertices

Let $u$ & $v$ be 2 vertices

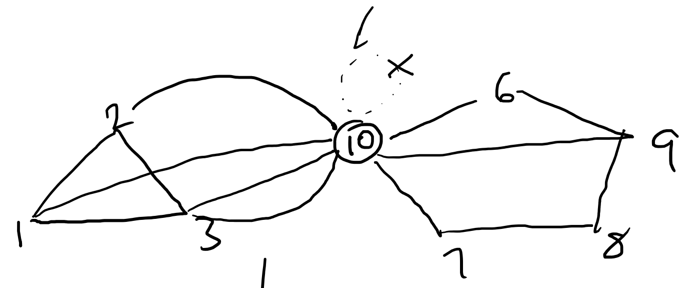
$G \setminus \{u, v\}$ = contraction of $G$ by "fusing $u$ & $v$"

Let's say we contract 4 & 5

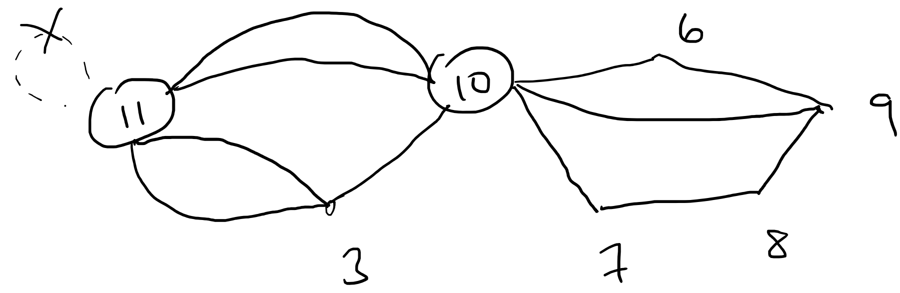
Let 10 be the contracted/fused vertex.

↓

Remove self-loop if any.




contract 1 & 2




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

---

KARGER ALGO

if $G$ = 2 vertices = $\{u, v\}$

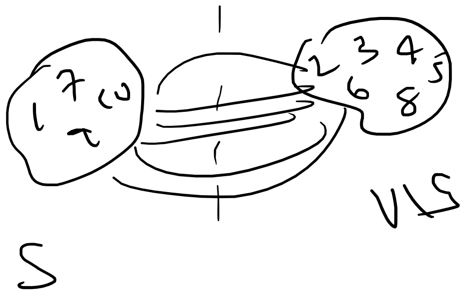
Min cut

= u on one side
v on other side

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

If $|G| > 2$ vertices:
Pick a <u>uniformly</u> random edge
and contract it !

---




output of random contraction algorithm

<u>THM</u>: with reasonable prob $\left(\geq \frac{1}{n^2}\right)$,
the resulting cut is <u>MINIMUM</u> !!

<u>Overall</u>
repeat $\Omega(n^2 \log n)$ times and pick the best

---

Min CUT is in P [poly-time solvable]
Let's look @ LP formulation [idea (1)]

<u>Variables</u>

$x_u = 0/1$ 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 $(u,v) \in E$

<u>Want</u> $y_{uv} = 0$ if $e$ is not cut
$\quad\quad\quad\quad 1$ if $e$ is cut.

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

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

Want to encode
$y_{uv} \geq \mathbb{1}\{x_u \neq x_v\}$
with linear constraint

<u>Attempt</u> $y_{uv} \geq |x_u - x_v|$.

is Actually a linear constraint because it is equivalent to

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

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

---

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

$$\left.\begin{array}{l} x_u \geq 0 \\ x_u \leq 1 \end{array}\right\} \forall u \in V$$

tries to capture $y_{uv} \geq \mathbb{1}(x_u \neq x_v)$.

$$\left.\begin{array}{l} y_{uv} \geq (x_u - x_v) \\ \quad\;\; \geq -(x_u - x_v) \end{array}\right\} \forall u, v.$$

---

Lemma 1

LP is a valid relaxation of Min Cut problem.

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

---

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 in S

< 0.5 put in V\S

Idea ② Think of $x_u$ as probability of being in S.

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

Algo:

① Choose random $\alpha$ between 0 & 1.

② For all $u \in V$,
put $u$ in S if $x_u^* \geq \alpha$.

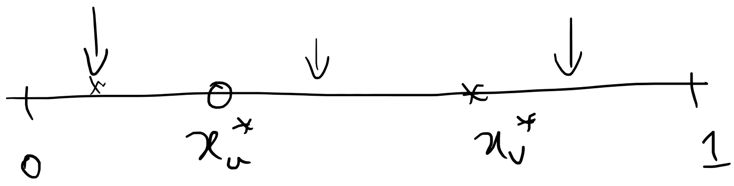
Need to show
$E[\#\text{ cut edges}]$ is small.

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

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

$(u,v) \in E$ Expectation

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




$$E[Z_{uv}] = \Pr[\,(u,v) \text{ is cut}\,]$$
$$= \Pr[\,u \ \& \ v \text{ are on diff sides}\,]$$
$$= \Pr_{\alpha}[\,\alpha \in (x_u^*, x_v^*)\,]$$
$$= |x_u^* - x_v^*|.$$
$$\leq y_{uv}^* \quad (\text{from LP constraint}).$$

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

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

OPT = 5

| | | | |
|---|---|---|---|
| Algo output | $(S_1, V \setminus S_1)$ | w.p. 0.25. | 4 |
| | $(S_2, V \setminus S_2)$ | w.p. 0.5. | 6 |
| ~~CANT~~ | | | 4 |

( ... )

~~CAN'T HAPPEN!~~ $(S_3, V \setminus S_3)$ w.p. 0.25 4

$\Rightarrow$ All events have to yield value 5

Issue Pointed Out:

All $x_u$ can be the same, LP will have value 0

How to fix this?

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

and enforce $x_{u_L} = 0$, $x_{u_R} = 1$ in the LP

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.

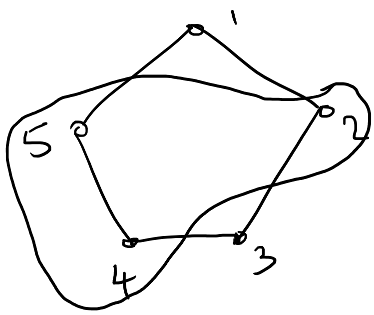
Min Cut is in P (exact polytime algorithm)

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

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

where $E(A, B) = \{ (u, v) \in E$ st $u \in A, v \in B \}$ for disjoint sets A, B

Example




What is the Max Cut?
# edges cut = 4

- Why is this Max Cut?

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

G here has an odd cycle.

---

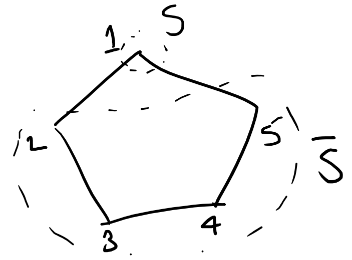
Max Cut for bipartite graphs is easy

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.

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

# edges cut = 2.

Lets swap 3 over.

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

# 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 $(S, \bar{S})$

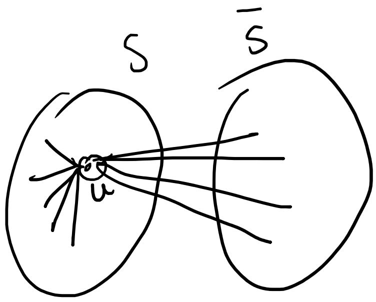
ANS: Quite good! Final solution $(S, \bar{S})$ cuts $\geq \frac{m}{2}$ edges of $|E| = m$.

Proof:

Let $(S, \bar{S})$ be the final soln.

$\forall u$, # edges incident to u crossing the partition

$\geq$ # edges incident to u within u's side.




$\Rightarrow$ u can swap

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

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

SUM OVER ALL u

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

$$= m.$$

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

$\left\{\frac{1}{2}\text{ - approximation}\right\}$

↑ ... maximization problems,

[$\frac{1}{2}$ - approximation]

For maximization problems,
$c$ approx $\Rightarrow$
Val(Alg) $\geq$ $c \cdot$ 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 using linear programming which ... using linear programming.

FUNNY: (LP will have infinitely many constraints)

but still "solvable" in poly-time

---

$$\left\{\begin{array}{l} \text{Max} \sum_{(u,v)\in E} y_{uv} \\ 0 \le x_u \le 1 \quad \forall u \\ \left[ y_{uv} \ge |x_u - x_v| \quad \forall u,v \right. \end{array}\right.$$

for Min Cut we used this

expressible as linear constraint

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

whereas we need

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

Problem ☹ Not a linear constraint

$y_{uv} \le \max\left(x_u - x_v \,,\, -(x_u - x_v)\right)$.

is not linearly expressible

whereas

$y_{uv} \ge \max\left( \quad " \quad , \quad " \quad \right)$

is expressible as

$y_{uv} \ge x_u - x_v$ and

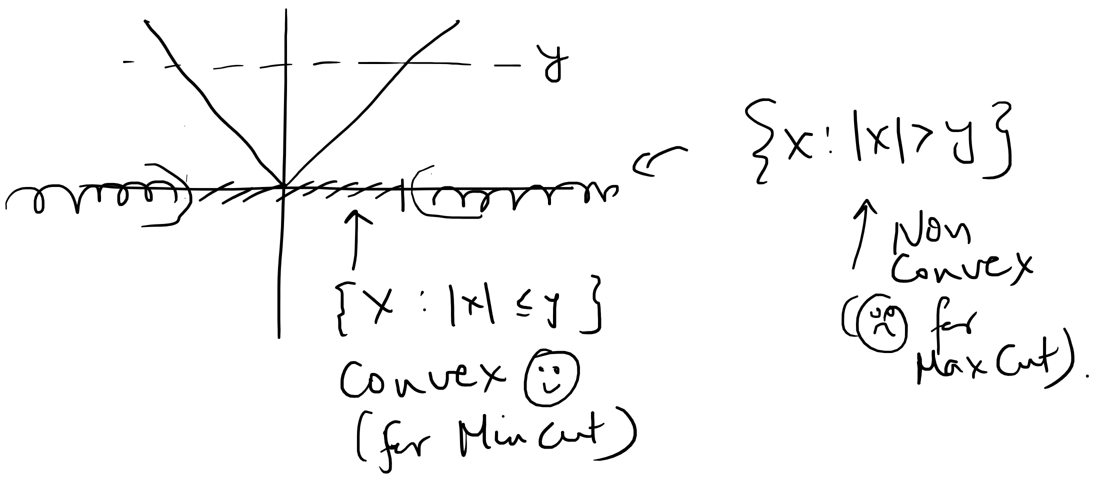
$y_{uv} \ge -(x_u - x_v)$.

Another way to see this issue :-

for any fixed $y$

$|x| \le y$ is a convex set

$|x| \ge y$ is not a convex set.




---

Can we "express" $y_{uv} \le |x_u - x_v|$

(or approximately)

using a linear program?

(or approx ... ))
using a linear program?

Hmmm...

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

This also captures $\mathbb{1}\{x_u \neq x_v\}$

Really, we're trying to capture $\mathbb{1}(x_u \neq x_v)$ 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
Lets think of them as $\pm 1$ variables.

---

Max Cut (exact formulation).

$$\left[\begin{array}{lr} \max \sum_{(u,v) \in E} \frac{1}{2}(1 - y_{uv}) & \\ x_u \in \{\pm 1\} & \forall u \\ y_{uv} = x_u \cdot x_v & \forall u, v \end{array}\right]$$

$x_u \in \{\pm 1\}$ is equivalent

$x_u \in \{\pm 1\}$ is equivalent to $x_u^2 = 1$, (ie) $Y_{uu} = 1$.

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

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

Want to express this using linear constraints.

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

Ex

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

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

$\Rightarrow$ we can enforce

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

$\Rightarrow \boxed{Y_{uv} \geq -1}$

$\because Y_{uu} = 1$

Similarly

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

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

$\boxed{Y_{uv} \leq 1}$ $\Leftarrow$

These are nice linear constraints in the $Y$ variables in the $Y$ variables ☺

$\underline{Q_N}$: what is a nice family of linear constraints we can impose on $Y_{uv}$, which the unknown soln satisfies?

SEMIDEFINITE PROGRAMMING

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

Let's look at

$$\left(\sum c_i x_i\right)^2 \geq 0 \quad \leftarrow \text{true always}$$

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

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

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

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

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

<u>Lemma ①</u>

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

$$\boxed{\text{This LP is actually called a semi-definite program}}$$

How to solve the LP if it has

① 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

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & & & \\ y_{n1} & y_{n2} & \cdots & y_{nn} \end{bmatrix}$$

(1) Then we can easily check

$y_{ii} = 1$ and
$y_{ij} = y_{ji}$ in $n^2$ time.

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

let's imagine c is a column vector

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}.$$

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

A real-symmetric matrix $Y$ is called a "positive-semidefinite matrix" if

$$c^T Y c \geq 0 \qquad \forall c \in \mathbb{R}^n$$

Checking ② $\iff$ 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 eigenpair if $\boxed{Av = \lambda v}$

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]?

$$\text{Max} \sum \tfrac{1}{2}(1 - y_{ij})$$

$(i,j) \in E$

$$Y_{ii} = 1 \quad \forall i$$

can be avoided because PSD will imply

$$Y_{ij} = Y_{ji} \quad \forall (i,j)$$

$$Y \succeq 0 \quad \left[ Y \text{ is a PSD matrix} \right]$$

- equivalent to $\forall c \;\; c^T Y c \geq 0$
- equivalent to all eigs of $Y \geq 0$
- equivalent to $\exists\, w_i$ vectors st $Y_{ij} = \langle w_i, w_j \rangle$.

---

$$\boxed{\text{Beautiful Technique for solving Very Large LPs}}$$

## 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 sol$^n$ is feasible or not

↓

then we can actually solve the LP efficiently!! and "almost optimally" (Negligible error)

(Negligible error)

Efficient = poly (#variables) · time of checker

---

OK We have solved the LP.
Gotten ourselves a $Y$ matrix and $w_i$ vectors st $Y_{ij} = \langle w_i, w_j \rangle$.

$$\boxed{\text{How to get a cut from this ?}}$$

<u>Next class</u> !

<u>ROUNDING ALGO</u> :-

Let $Y^*$ be the optimal SDP sol$^n$.

$$\text{SDP Obj} = \sum \tfrac{1}{2}(1 - Y^*_{ij})$$

From $Y^*$, (because $Y^* \succeq 0$ is PSD)
we can recover vectors
$w_1^*, w_2^*, \ldots, w_n^* \in \mathbb{R}^n$
such that $Y^*_{ij} = \langle w_i^*, w_j^* \rangle$.

Moreover these are unit vectors
b/c $1 = Y^*_{ii} = \langle w_i^*, w_i^* \rangle = \|w_i^*\|^2$

NOTE

| |
|---|
| If $w_i^*$ vectors were in 1 dimensional space, they would be $\pm 1$ scalars and $Y^* = x_i^* \cdot x_j^*$ would hold |

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

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

all $w_i^*$ are unit vectors.

**Real goal in ROUNDING**

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

**Idea** Random projection!

2 view points of this algo (equivalent)

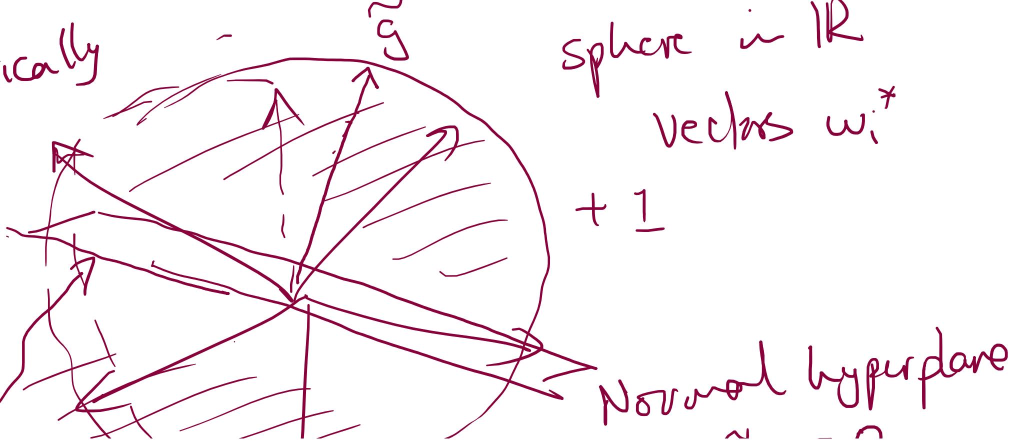
**VIEW ①**

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

Let $x_i = 1$ if $\langle w_i^*, g \rangle \geq 0$
$= -1$ otherwise.

**Geometric View**

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



choosing random hyperplane and partitioning

Normal hyperplane $\langle x, \tilde{g} \rangle = 0$

## Overall Algo

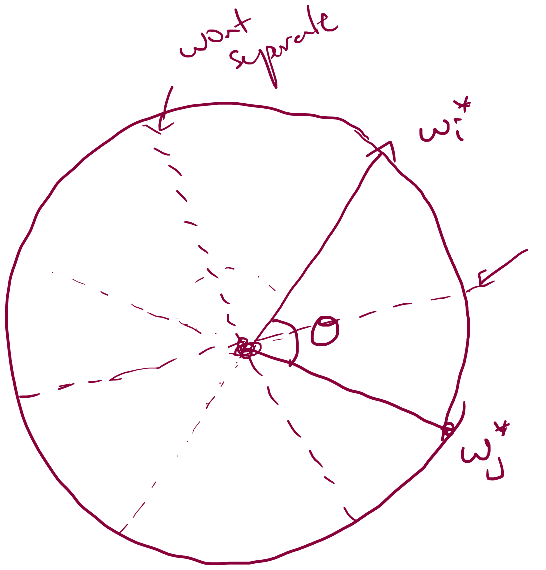
- Solve SDP optimally
- From $Y^*$, recover $w_1^*, w_2^*, \ldots w_n^*$
- Sample random gaussian vector $\tilde{g} = (g_1, g_2, \ldots, g_n)$
- Set $x_i = 1$ if $\langle w_i^*, \tilde{g} \rangle \geq 0$, $-1$ else
- Output $S = \{ i : x_i = 1 \}$ as cut.

Analysis:

Fix $(i,j)$: SDP is cutting this 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 $(i,j)$ if random hyperplane separates $w_i^*$ & $w_j^*$

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

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

---

① Our prob. of cutting edge $= \dfrac{\theta}{\pi}$

② SDP obj. $= \dfrac{1}{2}(1-\cos\theta)$

ANALYSIS QN:

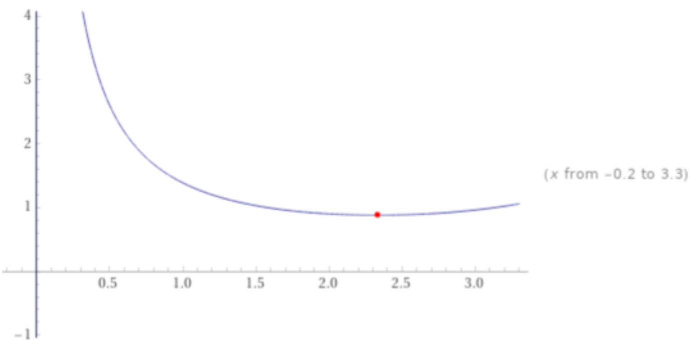
How small can ① be compared to ② ?

| | | |
|---|---|---|
| minimize | function | $\dfrac{2 \times \frac{x}{\pi}}{1-\cos(x)}$ |
| | domain | $0 \le x \le \pi$ |

Global minimum: More digits | Exact form | Step-by-step solution

$$\min\left\{\frac{2x}{\pi(1-\cos(x))} \,\middle|\, 0 \le x \le \pi\right\} \approx 0.878567 \text{ at } x \approx 2.33112$$

Plot:




$\Rightarrow$ ① $\geq 0.878$ ② $\forall\, \theta$.

$$\Rightarrow \quad (1) \geq 0.878 \; (2) \qquad \forall \theta$$

$$\Rightarrow E[\#\text{ cut edges}] = \sum_{(i,j)\in E} \Pr[(i,j) \text{ is cut}]$$

$$\geq \sum_{(i,j)\in E} 0.878 * \frac{1}{2}(1-\cos\theta_{ij})$$

$$= 0.878 \sum_{(i,j)\in E} \frac{1}{2}(1-\langle w_i^*, w_j^*\rangle)$$

$$= 0.878 \sum_{(i,j)\in E} \frac{1}{2}(1-y_{ij}^*).$$

$$= 0.878 \; \underline{\text{SDP OPT}}.$$

---

(A) Back to how to solve the SDP?

Ellipsoid Algorithm :-

Black box way to convert feasibility to optimization

SEPA-RATION 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 OR prove why $\tilde{y}$ is not feasible, (ie) output some constraint of LP/SDP which $\tilde{y}$ does not satisfy.

$\tilde{Y}$ 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 $\geq 0$, say feasible (also if all $\tilde{Y}_{ii} = 1$),

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

Output violated constraint $\boxed{v_i^T Y v_i \geq 0}$

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

$$v_i^T \tilde{Y} v_i = v_i^T \lambda_i v_i = \lambda_i \|v_i\|^2 < 0$$