①

Lecture 15 - Semidefinite Programming

[Scribe - Grötschel-Lovász-Schrijver]

Ellipsoid Alg: Recall: LP ≤ emptiness checking w/ big box constrs:

Given $K = \left\{ x : \begin{array}{l} Ax \le b \\ -R \le x_i \le R \ \forall i \end{array} \right\}$:

$K \ne \emptyset \rightarrow$ output $x \in K$

$K = \emptyset \rightarrow$ [nothing. alg can just output?]

Claim: ≤ "robust version": $K \supseteq$ a little box of side r $\rightarrow$ output $x \in K$

r input to alg

not input!

[obv, if can solve exact ver, can solve robust ver. Reverse?]

"Proof": [See hawk for details]

Convert to $Ax = b$, $x \ge 0$ form.

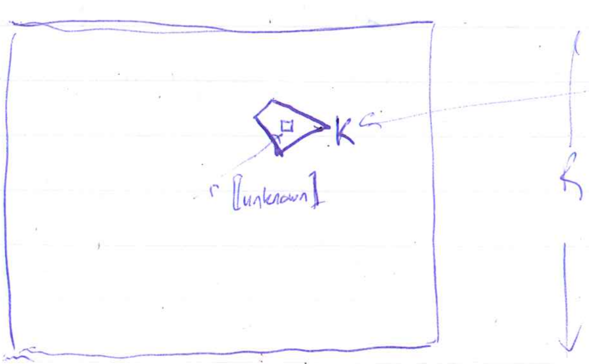
$K = \emptyset$ $\ge 2^{-\mathrm{poly}(\langle A \rangle, \langle b \rangle)}$

$K \ne \emptyset$.

[For the usual granularity reasons.]

$\rightsquigarrow b - r' \le Ax \le b + r'$.

[Pic]



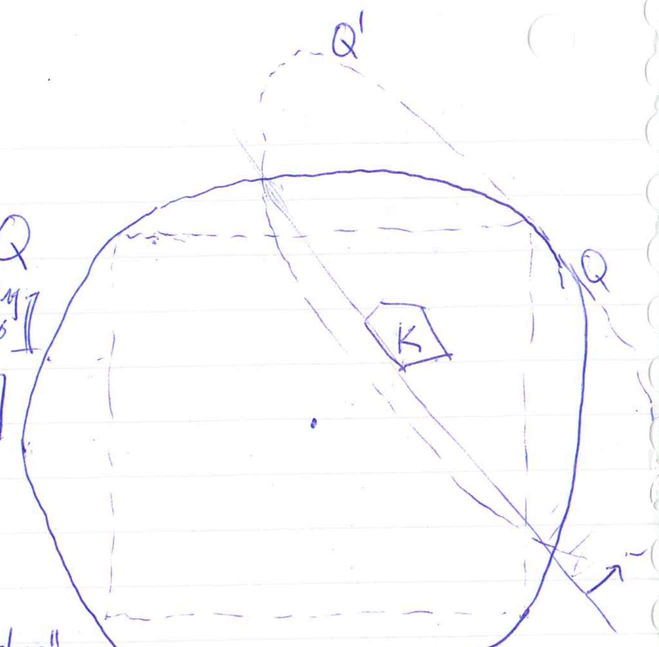
"☐"

[locale unknown, can assume $K \supseteq$ ☐]

[don't have to worry abt $K = \emptyset$ case].

["hunting for K"]

Ellipsoid Alg:



Invar: Maintain an ellipsoid Q containing K [[assuming $K \neq \emptyset$]]

[[Initially a round ball around big box.]]

Alg:
- Ask: center$(Q) \in K$?
- If yes, ☺
- Else get "separating hyperplane" [[violated constr.]]. $a \cdot x \geq b$
- Set $Q' :=$ smallest ellipsoid containing $Q \cap \{a x \geq b\}$

Analysis: (easy) lemma: $\mathrm{vol}(Q') \leq \left(1 - \frac{1}{3n}\right) \mathrm{vol}(Q)$.

[[hard part is just numerics, $\sqrt{\ }$, etc.]] $\Rightarrow$ Repeating $O(n)$ times, volume halves

$$\mathrm{vol}(Q_0) = R^n \cdot n^{O(n)}$$
$$\mathrm{vol}(K) \geq r^n.$$

$\therefore$ #steps at most $O(n) \cdot \log\left(\frac{R^n \cdot n^{O(n)}}{r^n}\right) = \mathrm{poly}(n)\left(\log\left(\frac{R}{r}\right)\right)$

$\leq \mathrm{poly}$(input size)

$\square$

KEY OBS: Alg. doesn't need to "know" $K = \{Ax \leq b\}$

Just needs: R, r, "separation oracle"

[[doesn't have to be a facet]]

$z \longrightarrow$ [[black box]] $\longrightarrow$ $z \in K$!

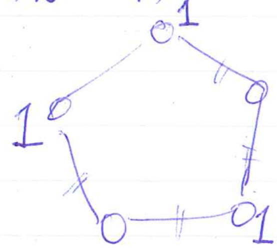
$\longrightarrow$ "$a \cdot x \geq b$" which K sats, z doesn't.

SDP: Crucial

[[Motivating example is Max-Cut problem]]

[[Last time: Saw Min(-st-)Cut solved exactly by L.P.]]
[[Max-cut is NP-compl., so can't solve w/ LP. Saw a factor-½ approx in second lec. Try LP anyway...]]

Max-Cut?



[[Sol'n? Can cut 4 out of 5 edges. Not 5 ∵ not bip. → odd cycle.]]

4/5 [[We focus on frac. of edges cut.]]

~~[[We'll be convenient to use ±1 notation for cut, not 0/1.]]~~

Find $f : V \to \{0,1\}$

to max: $\Pr_{v \sim w}[f(v) \neq f(w)]$ [[Agree?]]

[[try to use (v,w) not (u,v) throughout!]]

LP relaxation? [[well, if you recall min-cut.....]]

$x_v \in \{0,1\} \longrightarrow 0 \le x_v \le 1$ [[intent: $f(v)$]]

$y_{uv} \in \{0,1\} \longrightarrow 0 \le y_{uv} \le 1$ intent: $1[f(u) \neq f(v)]$

max $\operatorname{avg}_{u \sim v} \{y_{uv}\}$ [[that's a lin obj.]]

[[Well, so far can take $y_{uv} \equiv 1$]]

$y_{uv} \stackrel{?}{=} 1[x_u \neq x_v]$ (?)

For min-cut, $y_{uv} \geq |x_u - x_v|$ is cool → linear [[why?]]

Max-cut, $y_{uv} \le |x_u - x_v|$. ? ☹

Smart Delorme Poljak '90 e.a. Idea: [[Unimportant, yet highly elegant, little notational switch → ±1]]

4

[[Exact]]

$x_v \in \{-1,1\}$ $\forall v$. $y_{uv} = x_u x_v$ [[0/1 indic that $x_u \neq x_v$]]

$\max \operatorname{avg}_{u \sim v} \{ \frac{1}{2} - \frac{1}{2} y_{uv} \}$

(LP)

$y_{uv} = y_{vu}$ [[can just have half the vbls]]

$y_{vv} = 1$ $\forall v$.

[[Looks like a ridic LP. Could just take $y_{uv} \equiv -1$. Or $y_{uv} \equiv -100$!]]

(S) D.P. idea: Try to "enforce" $\boxed{\text{"}\exists (x_v) \text{ s.t. } y_{uv} = x_u x_v\text{"}}$ MOMENT CONSTR.
using linear constrs.

semi def. [[elaborate later]]

(NB: $\because$ we have $y_{vv}=1$, can drop $x_v \in \{\pm 1\}$)

MOMENT + LP = exact

[[How we gonna do that?]]

[[Leave up]]

Exact.

$x_v \in \{-1,1\} \quad \forall v \qquad y_{uv} = x_u x_v \quad \forall u,v$

$\max \ \operatorname{avg}_{u \sim v} \left\{ \frac{1}{2} - \frac{1}{2} y_{uv} \right\}$ [[0/1 indic. that $x_u \neq x_v$]]

[[kinda ridic, though. Let's put in a few more condits...]]

$y_{uv} = y_{vu} \quad \forall u,v$ [[or could just use $\frac{1}{2}$ the vbls]]

$y_{uu} = 1 \quad \forall u$.. [[certainly $\because x_u \in \{\pm 1\}$]]

$\boxed{LP}$

[[at this point, can drop $x_v \in \{-1,1\}$]]

[[Smart Delorme-Poljak]]: $\exists (x_u)$ s.t. $y_{uv} = x_u x_v \quad \forall u,v.$

$\boxed{\text{Moment}}$

Rem: LP + MOMENT is exact.

S.D.P. idea: Try to enforce $\boxed{\text{MOMENT}}$ with linear constrs.

[[How we going to do that?]]

Hand off $\boxed{LP}$ to Ellipsoid: [[Actually unbounded, at a high level here, say it comes back with..]]

$y_{uv} \equiv -2^{\text{poly}(n)} \ (u \neq v), \quad y_{uu} \equiv 1.$

"Just a moment: Forgot the constr. $\sum_{u,v \in V} y_{uv} \geq 0$".

[[Why is that valid?]] Claim: $\boxed{\text{MOMENT}} \Rightarrow$

Pf: $\Downarrow \quad = \sum_{u,v \in V} x_u x_v = \left(\sum_u x_u \right)^2 \geq 0.$

[[You gave a "violated constr" to Ellipsoid. It comes back with a new opt. LP sol^n sat'ing constr.]]

[Maybe comes back with new sol^{n}...]

$$y_{11} = 1, \quad y_{12} = +100, \quad y_{21} = +100, \quad y_{22} = 1, \ldots$$

"Just a moment. Forgot constr. $y_{11} - y_{12} - y_{21} + y_{22} \geq 0$" (*)

Claim: $\boxed{\text{MOMENT}} \Longrightarrow$ ---

Pf: $X_1X_1 - X_1X_2 - X_2X_1 - X_2X_2 = (X_1 - X_2)^2 \geq 0$ □

Ellipsoid: Actually $y_{12} = -100$, $y_{21} = -100 \ldots$

$\boxed{\text{MOMENT}} \Rightarrow$ $\boxed{y_{11} + y_{12} + y_{21} + y_{22} \geq 0}$ (**) [Same pf...]

BTW: (*) + $\boxed{\text{LP}} \Rightarrow$ $1 - y_{12} - y_{21} + 1 \geq 0 \Rightarrow y_{12} \leq 1$. ☺

(**) + $\boxed{\text{LP}} \Rightarrow$ $y_{12} \geq -1$ ☺ [$\forall u,v$ of course]

[What are we doing here? What are these "just a moment" constrs...]

Generally: $\left(\sum_{u\in V} c_u X_u\right)^2 \geq 0 \Rightarrow \sum_{u,v} c_u c_v X_u X_v \geq 0$

$\therefore$ MOMENT $\Rightarrow \sum_{u,v} c_u c_v y_{uv} \geq 0.$

$\boxed{\text{S.D.P. constr.}}$:= add ALL of these constrs $\forall (c_v) \in \mathbb{R}^n$.

Key facts: ① We can do this, efficiently!

② $\boxed{\text{MOMENT}} \Longrightarrow \boxed{\text{SDP}}$ but $\not\Longleftarrow$

[SDP constraints don't totally enforce MOMENT. If they did, we'd solve Max-Cut in P!]

③ Still, $\boxed{\text{LP}}$ + $\boxed{\text{SDP}}$ quite good relaxation for Max-Cut.

[[Let's take these points in order]]

def: A symmetric mtx $Y = \begin{pmatrix} y_{uv} \end{pmatrix} \in \mathbb{R}^{n \times n}$ is <u>positive semidefinite (PSD)</u>
if $\begin{bmatrix} - c - \end{bmatrix} \begin{bmatrix} Y \end{bmatrix} \begin{bmatrix} | \\ c \\ | \end{bmatrix} \geq 0 \quad \forall c \in \mathbb{R}^n. \quad (Y \succeq 0)$
$\left(\sum_{u,v} c_u c_v y_{uv} \right)$

thm: $\exists$ poly($\langle Y \rangle$)-time alg which, given Y, either finds c s.t. $c^T Y c < 0$ or outputs "Y is PSD".
(see Hmwk #3.3)

pf sketch: Do "symmetric Gaussian elim." ("Cholesky Decomp."), trying to write
$$Y = \underbrace{\begin{bmatrix} 1 & & 0 \\ & 1 & \\ * & & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} * & & 0 \\ & * & \\ 0 & & * \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & & * \\ & 1 & \\ 0 & & 1 \end{bmatrix}}_{L^T}$$
where D's entries ≥ 0.
Success $\to$ PSD. Failure $\Longrightarrow$ easily extract c. $\square$

[[can output "proof" $Y = L^T D L^T$]] [Why?] $c^T Y c = c^T L D L^T c = c^T \underbrace{L\sqrt{D}}_{U^T} \underbrace{\sqrt{D} L^T}_{U} c = (\sqrt{D} L^T c)^T (\sqrt{D} L^T c) = \|\sqrt{D} L^T c\|_2^2 \geq 0.$

prop: (symm.) Y PSD $\iff Y = U^T U$ for some $U \in \mathbb{R}^{n \times n}$ [upper-Δ]
[[$\iff$ for some upper-Δ U]]

$\therefore$ $\exists$ poly-time "separation oracle" for PSD-ness
$\equiv \sum_{u,v} c_u c_v y_{uv} \geq 0 \ \forall c.$

$\therefore$ Can use Ellipsoid to solve* any
$\boxed{LP} + \boxed{SDP\ constr.} \leftarrow$ "$\begin{bmatrix} Y \end{bmatrix}$ is PSD".

* [[Slight annoyance: Ellipsoid needs small box constr / Full-dim. constr.]]
[[technically, must have the small & big box constrs.]]
~~$y_{uu} = 1$~~ $\to$ $1 - 2^{-n} \leq y \leq 1 + 2^{-n}$. [[Henceforth <u>ignored</u>]]

② $\boxed{\text{MOMENT}} \Longrightarrow \boxed{\text{PSD}}$

"$\exists (x_i) \in \mathbb{R}^n$ s.t. $y_{ij} = x_i x_j \ \forall u,v$"

$\sum_{i,j} c_i c_j y_{ij} \geq 0 \quad \forall c \in \mathbb{R}^n$

$\Longleftrightarrow Y = U^T U.$

ex: $\not\Longleftarrow$ is false.

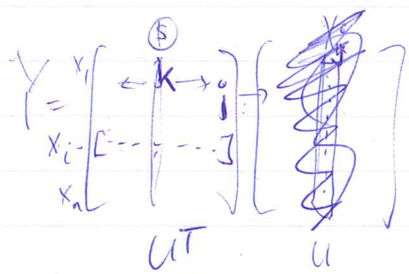
[But there is a similar condition which is equiv. to PSD.]

prop: (symm.) Y is PSD $\Longleftrightarrow$ $\exists$ jointly random variables $X_1, \ldots, X_n$ s.t.

$$y_{ij} = E[X_i X_j].$$

pf: $(\Longleftarrow)$: $\forall c$,
$$\sum_{i,j} c_i c_j y_{ij} = \sum_{i,j} c_i c_j E[X_i X_j]$$
$$= E\Big[\sum_{i,j} c_i X_i c_j X_j\Big]$$
$$= E\Big[\big(\sum_i c_i X_i\big)^2\Big] \geq 0. \ \square$$

$(\Longrightarrow)$ Say $Y = U^T U$. Define $(X_i)_{i \in [n]}$ via:
- Pick $k \sim [n]$ unif
- Set $X_i = \sqrt{n} \cdot U^T_{ik} = \sqrt{n} \cdot U_{ki}$



Now

$$E[X_i X_j] = \frac{1}{n} \sum_{k=1}^{n} \sqrt{n}\, U^T_{ik} \cdot \sqrt{n}\, U_{kj}$$
$$= \sum_{k=1}^{n} U^T_{ik} U_{kj}$$
$$= (U^T U)_{ij} = Y_{ij}. \ \square$$

prop: Y PSD $\Longleftrightarrow$ $\exists$ vectors $\vec{u}_1, \ldots, \vec{u}_n$ s.t.

$$Y_{ij} = \langle \vec{u}_i, \vec{u}_j \rangle \quad \forall ij$$

pf: Let $\vec{u}_i = \sqrt{n} \cdot (i^{th}$ col. of $U)$.