Advanced Algorithms

南京大学

尹一通

Constraint Satisfaction Problem

- variables: $X = \{x_1, x_2, \ldots, x_n\}$
- domain: Ω, usually $\Omega = [q]$ for a finite q
- constraints: $c = (\psi, S)$ where ψ: $\Omega^k \to \{0,1\}$ and scope $S \subseteq X$ is a subset of k variables
- CSP instance: a set of constraints defined on X
- assignment: $\sigma \in \Omega^X$ assigns values to variables
- a constraint $c = (\psi, S)$ is *satisfied* if $\psi(\sigma_S) = 1$
- examples:
  - max-cut: q=2, constraints are $\neq$
  - k-SAT: q=2, constraints are k-clauses
  - matching/cover: q=2, constraints are $\sum\leq 1$ (or $\sum\geq 1$)
  - k-coloring: q=k, constraints are $\neq$
  - graph homomorphism: constraint is adjacency matrix

Constraint Satisfaction Problem

- variables: $X = \{x_1, x_2, \ldots, x_n\}$
- domain: Ω, usually $\Omega = [q]$ for a finite q
- constraints: $c = (\psi, S)$ where $\psi: \Omega^k \to \{0,1\}$ and scope $S \subseteq X$ is a subset of k variables
- CSP instance: a set of constraints defined on X
- assignment: $\sigma \in \Omega^X$ assigns values to variables
- a constraint $c = (\psi, S)$ is *satisfied* if $\psi(\sigma_S) = 1$
- examples:
  - unique games: $\Omega = [q]$, each constraint is an arbitrary binary *bijective* predicate:
    $\psi: \Omega^2 \to \{0,1\}$ where $\forall a \in \Omega$, $\exists$ unique $b \in \Omega$, $\psi(a,b)=1$

Algorithmic Problems for CSP

Given a CSP instance I:

- satisfiability: decide whether ∃ an assignment satisfying all constraints
  - search: find such a satisfying assignment
- optimization: find an assignment satisfying as many constraints as possible
  - refutation (dual): find a "proof" of "no assignment can satisfy $>m$ constraints" for m as small as possible
- counting: estimate the number of satisfying assignments
  - sampling: random sample a satisfying assignments
  - inference: observing part of a satisfying assignment, guess the value of an unobserved variable

Algorithmic Problems for CSP

CSP	Satisfiability	optimization	counting
2SAT	**P**	**NP**-hard	**#P**-complete
3SAT	**NP**-complete	**NP**-hard	**#P**-complete
matching	perfect matching **P**	max matching **P**	**#P**-complete
cut (2-coloring)	bipartite test **P**	max-cut **NP**-hard	**FP** (poly-time)
3-coloring	**NP**-complete	max-3-cut **NP**-hard	**#P**-complete

A Wishlist for Optimization Algorithms

- Nonlinear, non-convex objectives.
- Powerful enough to tackle hard problems in a systematic way, and meanwhile is still practical.
- Becoming more accurate as we're paying more (but certainly won't beat the inapproximability).
- A generic framework that can be applied obviously to various problems.

sum-of-squares (SoS) SDP, Lasserre hierarchy, Lovász-Schrijver hierarchy, ...

A Polynomial Program?

Given an n-variate polynomial f, determine:

- whether $f(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \{0,1\}^n$.

polynomial:

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{d}=(d_1,\ldots,d_n)\in\mathbb{N}^n} a_{\boldsymbol{d}} \cdot x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$$

degree of monomial $a_{\boldsymbol{d}} \cdot x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$: $d_1 + \cdots + d_n$

degree of f: largest degree for non-zero monomials

multilinear: $f(\boldsymbol{x}) = \sum_{\boldsymbol{d}=(d_1,\ldots,d_n)\in\{0,1\}^n} a_{\boldsymbol{d}} \cdot x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$

A Polynomial Program?

Given an n-variate polynomial f, determine:

- whether $f(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \{0,1\}^n$.

NP-hard

graph $G(V,E)$: $\forall \boldsymbol{x} \in \{0,1\}^n$ represents a cut of size

$$cut(\boldsymbol{x}) = \sum_{ij \in E} (x_i - x_j)^2$$

$$\boxed{\text{let } f(\boldsymbol{x}) = C - cut(\boldsymbol{x})}$$

- $\exists\, \boldsymbol{x} \in \{0,1\}^n$, $f(\boldsymbol{x}) < 0 \Leftrightarrow$ found a cut of size $>C$;
- $\forall \boldsymbol{x} \in \{0,1\}^n$: $f(\boldsymbol{x}) \geq 0 \Leftrightarrow$ no cut of size $>C$;

decision of non-negativity of f $\Rightarrow$ optimization of cut by binary search for C

Sum-of-Squares (SoS) Proofs

Given an n-variate polynomial f , find:

- either an $\boldsymbol{x} \in \{0,1\}^n$ such that $f(\boldsymbol{x}) < 0$;
- or a "proof" of $f(\boldsymbol{x}) \geq 0$ over all $\boldsymbol{x} \in \{0,1\}^n$.

a proof of non-negativity over hypercube:

$f(\boldsymbol{x})$ can be represented as a sum of squares

$$f(\boldsymbol{x}) = g_1(\boldsymbol{x})^2 + g_2(\boldsymbol{x})^2 + \cdots + g_r(\boldsymbol{x})^2$$

for all $\boldsymbol{x} \in \{0,1\}^n$

$\Rightarrow$ $f(\boldsymbol{x}) \geq 0$ over all $\boldsymbol{x} \in \{0,1\}^n$

Sum of Squares (SoS) Polynomials

$f\colon\mathbb{R}^n\to\mathbb{R}$ is an n-variate non-negative polynomial

$$\forall \boldsymbol{x}\in\mathbb{R}^n,\ f(\boldsymbol{x})\geq 0$$

Minkowaski: "Can any such f be represented as a sum of squares of other polynomials?"

- Hilbert (1888): No. (nonconstructive)
- Hilbert's 17th problem (1900):
  "Can such f be represented as SoS of *rational* functions?"
- Artin (1927): Yes. (to Hilbert's question)
- Motzkin (1967):

$$f(x,y,z)=z^6+x^4y^2+x^2y^4-3x^2y^2z^2$$

nonnegative but not SoS of polynomials

David Hilbert (1862-1943)

SoS Proofs

degree-d SoS proof for $f:\{0,1\}^n \to \mathbb{R}$:

n-variate polynomials $g_1, g_2, \ldots, g_r: \{0,1\}^n \to \mathbb{R}$
of degree $\leq d/2$ such that

$$f(\boldsymbol{x}) = g_1(\boldsymbol{x})^2 + g_2(\boldsymbol{x})^2 + \cdots + g_r(\boldsymbol{x})^2$$

for all $\boldsymbol{x} \in \{0,1\}^n$.

For nonnegative polynomial $f:\{0,1\}^n \to \mathbb{R}$

- degree-$2n$ SoS proof always exists (by interpolation);
- a degree-d SoS proof needs at most $r = n^{O(d)}$ length;
- a degree-d SoS proof $f = \sum_i g_i^2$ over $\{0,1\}^n$, can be verified in $n^{O(d)}$ time (by reducing to multilinear polynomials);
- if f has degree-d SoS proof, then it can be found in $n^{O(d)}$ time (by SDP).

A Polynomial Program?

Given an n-variate polynomial f, find:

- either an $\boldsymbol{x} \in \{0,1\}^n$ such that $f(\boldsymbol{x}) < 0$;
- or a "proof" of $f(\boldsymbol{x}) \geq 0$ over all $\boldsymbol{x} \in \{0,1\}^n$.

each $\boldsymbol{x} \in \{0,1\}^n \;\Leftrightarrow\;$ a cut in $G(V, E)$ of size:

$$cut(\boldsymbol{x}) = \sum_{ij \in E} (x_i - x_j)^2$$

let $\boxed{f(\boldsymbol{x}) = C - cut(\boldsymbol{x})}$

optimization

- $\exists\, \boldsymbol{x} \in \{0,1\}^n,\ f(\boldsymbol{x}) < 0 \Leftrightarrow$ found a cut of size $> C$;
- $\forall \boldsymbol{x} \in \{0,1\}^n :\ f(\boldsymbol{x}) \geq 0 \Leftrightarrow$ no cut of size $> C$;

refutation

decision of non-negativity of f $\Rightarrow$ optimization of cut by binary search for C

Pseudo-Distribution

degree-d pseudo-distribution over $\{0,1\}^n$:

μ: $\{0,1\}^n \to \mathbb{R}$ (could be negative) such that

- $$\sum_{\boldsymbol{x}\in\{0,1\}^n} \mu(\boldsymbol{x}) = 1$$
- for every polynomial f : $\{0,1\}^n \to \mathbb{R}$ of degree $\leq d/2$, the formal expectation:
$$\mathbb{E}_\mu f^2 = \sum_{\boldsymbol{x}\in\{0,1\}^n} \mu(\boldsymbol{x}) f(\boldsymbol{x})^2 \geq 0$$

- any degree-d pseudo-distribution μ can be represented as a multilinear polynomial of degree at most d;
- if an f of degree $\leq d$ has a pseudo-distribution μ such that $\mathbf{E}_\mu f < 0$, then μ can be found in $n^{O(d)}$ time (by SDP).

Duality

degree-d SoS proof for f: $\{0,1\}^n \to \mathbb{R}$:

$f = g_1^2 + \cdots + g_r^2$ for $g_1, \ldots, g_r$ for of degree $\leq d/2$

degree-d pseudo-distribution μ: $\{0,1\}^n \to \mathbb{R}$:

$\mathbb{E}_\mu \mathbf{1} = 1$ and $\mathbb{E}_\mu f^2 \geq 0$ for any f of degree $\leq d/2$

Duality of SoS proofs and pseudo-distributions:

(SDP) f has degree-d SoS proof $\Longleftrightarrow$ $\mathbb{E}_\mu f \geq 0$ for all degree-d pseudo-distribution μ

f has no degree-d SoS proof $\Leftrightarrow$ $\mathbf{E}_\mu f < 0$ for some degree-d pseudo-distribution μ (SDP)

- functions with degree-d proofs form a convex set; pseudo-distribution is a separating hyperplane from it.

The SoS Algorithm

a polynomial $f : \{0,1\}^n \to \mathbb{R}$ of degree $\leq k$, find:

- either a proof of $f(\boldsymbol{x}) \geq 0$ over all $\boldsymbol{x} \in \{0,1\}^n$;
- or an $\boldsymbol{x} \in \{0,1\}^n$ such that $f(\boldsymbol{x}) < 0$.

Duality of SoS proofs and pseudo-distributions:

f has degree-d SoS proof $\Longleftrightarrow$ $\mathbb{E}_\mu f \geq 0$ for all degree-d pseudo-distribution μ

SoS Algorithm:

try for every even $d \geq k$, to find (by SDP in $n^{O(d)}$ time):

- either a degree-d SoS proof for f;
- or a degree-d pseudo-dist. μ such that $\mathbf{E}_\mu f < 0$;

The SoS Algorithm

a polynomial $f : \{0,1\}^n \rightarrow \mathbb{R}$ of degree $\leq k$, find:

- either a proof of $f(\boldsymbol{x}) \geq 0$ over all $\boldsymbol{x} \in \{0,1\}^n$;
- or an $\boldsymbol{x} \in \{0,1\}^n$ such that $f(\boldsymbol{x}) < 0$.

SoS Algorithm:

try for every even $d \geq k$, to find (by SDP in $n^{O(d)}$ time):

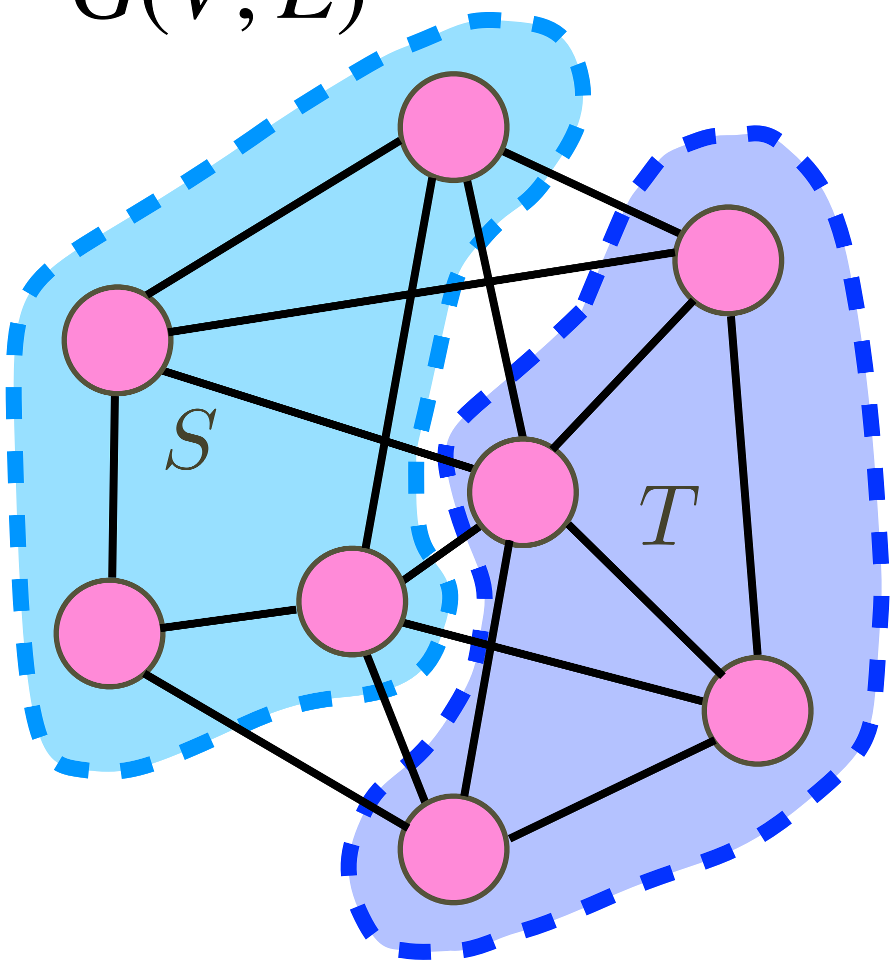
- either a degree-d SoS proof for f;
- or a degree-d pseudo-dist. μ such that $\mathbf{E}_\mu f < 0$;

for specific f — construct from μ → *efficiently sample-able* probability distribution ν over $\{0,1\}^n$ such that $\mathbf{E}_\nu f < 0$

- time complexity: for what d, f has degree-d SoS proof
- approximation ratio: gap between pseudo-dist. and dist.

Max-Cut

$G(V, E)$



$\forall \boldsymbol{x} \in \{0,1\}^n$:

$$cut(\boldsymbol{x}) = \sum_{ij \in E} (x_i - x_j)^2$$

Rounding pseudo-distributions:

$\forall G$ and $\forall$ degree-2 pseudo-dist. μ

$\exists$ an efficiently sample-able dist. ν

such that

$$\mathbb{E}_{\boldsymbol{x} \sim \nu} [cut(\boldsymbol{x})] \geq 0.878 \mathbb{E}_{\boldsymbol{x} \sim \mu} [cut(\boldsymbol{x})]$$

for every C: polynomial $f(\boldsymbol{x}) = C - 0.878 \cdot cut(\boldsymbol{x})$

- either has a degree-2 SoS proof of non-negativity (by duality);
- or produce a random cut $\sim\nu$ with $\mathbf{E}$[cut-size] $> 0.878 \cdot C$

for $C = \mathrm{OPT}(G)$, this means $\mathbf{E}$[cut-size] $> 0.878 \cdot \mathrm{OPT}$

Convex Programming

Convex programming:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{K} \end{aligned}$$

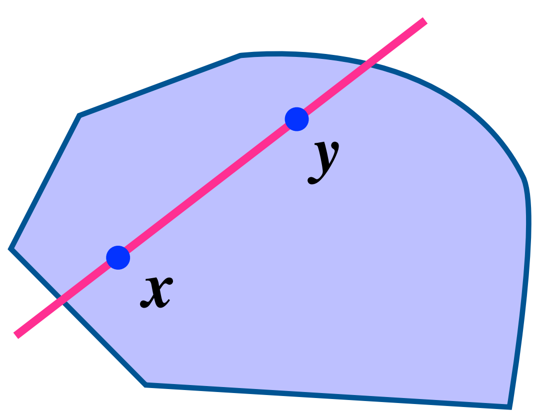
binary search

$$\exists \boldsymbol{x} \in \mathcal{K}'?$$

$$\mathcal{K}' = \mathcal{K} \cap \{\boldsymbol{x} : \boldsymbol{c}^T\boldsymbol{x} \le b\}$$

still convex

$\mathcal{K}$: convex body

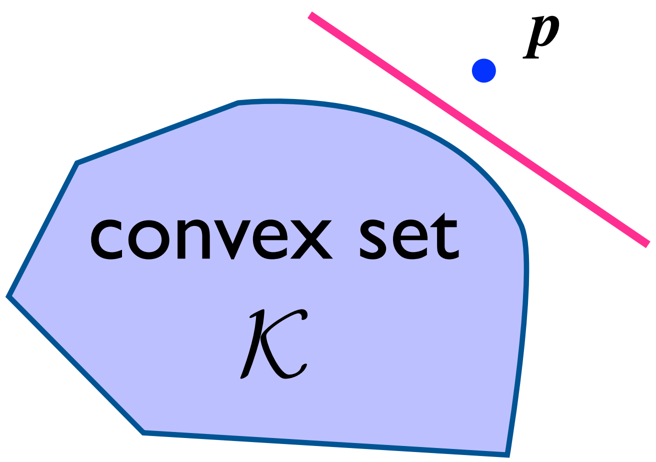


$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}, \quad \forall \lambda \in [0, 1]$$

$$\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in \mathcal{K}$$

convex polytope, set of positive semidefinite matrices

Separation Oracle



Farkas's Lemma:

convex set $\mathcal{K} \subseteq \mathbb{R}^n$, point $\boldsymbol{p} \notin \mathcal{K}$

⇨ $\exists$ a hyperplane separating $\boldsymbol{p}$ from $\mathcal{K}$

poly-time separation oracle for convex set $\mathcal{K}$:

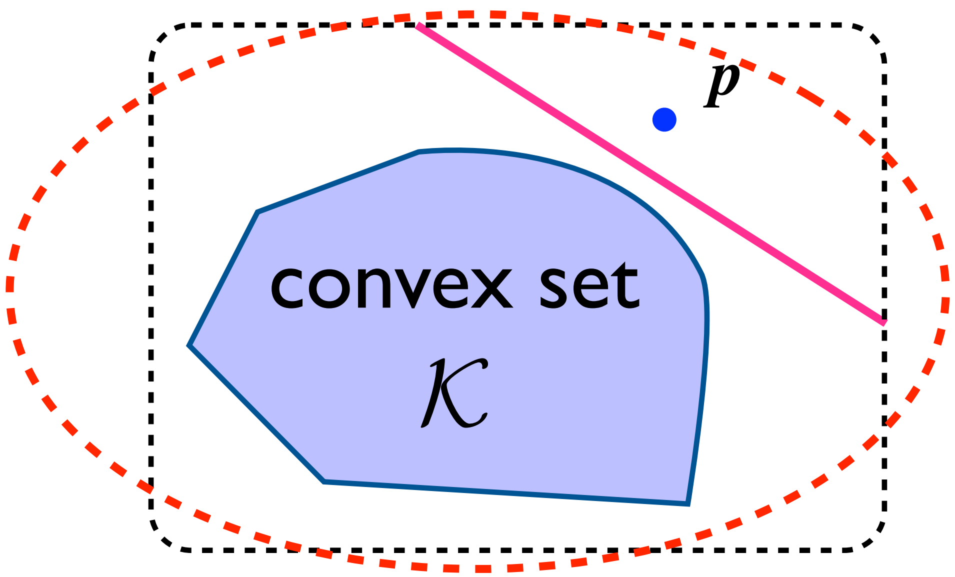
Given a point $\boldsymbol{p}$, either confirms that $\boldsymbol{p} \in \mathcal{K}$

or returns a hyperplane separating $\boldsymbol{p}$ from $\mathcal{K}$

within poly-time.

- even works for programs with infinite constraints: SDP
- LP: returns a violated linear constraint
- SDP: returns an eigenvector with negative eigenvalue

The Ellipsoid Method



Assume ∃ a poly-time Separation Oracle (SO)

bounding boxes are ellipsoids:

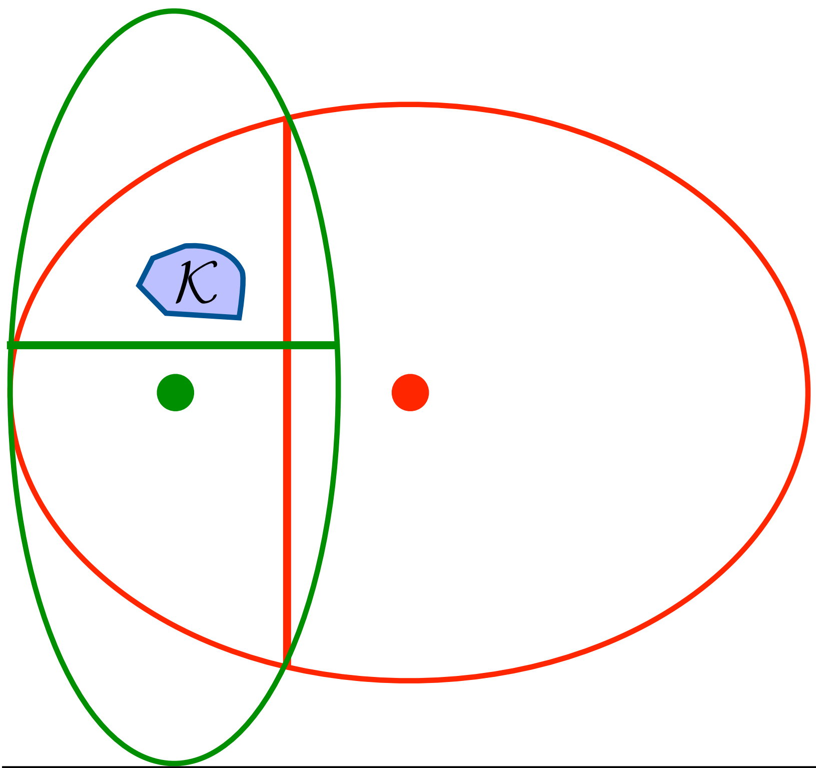
$$\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - a_i)b_{ij}(x_j - a_j) \leq 1$$

Divide & Conquer:

Start with a bounding box $\supseteq \mathcal{K}$;
pick a point $\boldsymbol{p} \in$ bounding box and asks SO if $\boldsymbol{p} \in \mathcal{K}$;

if SO says "yes", done;
else SO returns a half-space $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b$ containing $\mathcal{K}$ but $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{p} > b$;
continue with the minimum bounding box $\supseteq \{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b\}$;

The Ellipsoid Method



Assume ∃ a poly-time
Separation Oracle (SO)

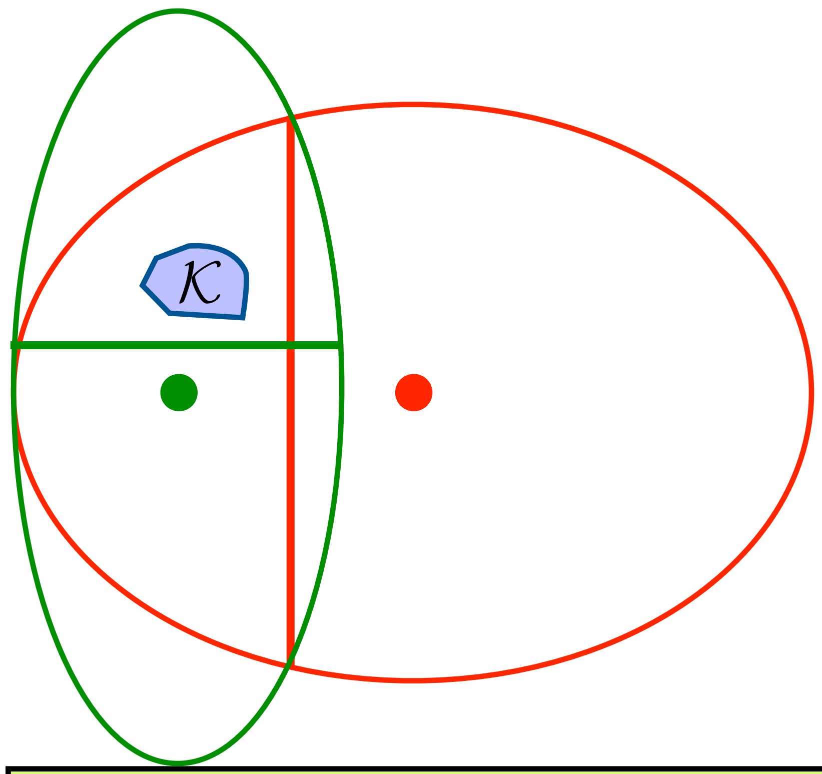
bounding boxes are ellipsoids:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - a_i)b_{ij}(x_j - a_j) \le 1$$

Ellipsoid Method:

Start with an ellipsoid $E_0 \supseteq \mathcal{K}$;
for i=0,1,2, ... until E_i is too small, then returns "$\mathcal{K}$ is empty";
pick the center $\boldsymbol{p}$ of E_i and asks SO if $\boldsymbol{p} \in \mathcal{K}$;
if SO says "yes", return $\boldsymbol{p}$;
else SO returns a half-space $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \le b$ containing $\mathcal{K}$ but $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{p} > b$;
E_{i+1} = the minimum ellipsoid $\supseteq E_i \cap \{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \le b\}$;

The Ellipsoid Method



$\mathcal{K} \subseteq \mathbb{R}^n$

Lemma:

$\mathrm{Vol}(E_{i+1}) \leq (1-1/2n)\ \mathrm{Vol}(E_i)$

Ellipsoid Method:

Start with an ellipsoid $E_0 \supseteq \mathcal{K}$;
for i=0,1,2, ... until E_i is too small, then returns "$\mathcal{K}$ is empty";
pick the center $\boldsymbol{p}$ of E_i and asks SO if $\boldsymbol{p} \in \mathcal{K}$;
if SO says "yes", return $\boldsymbol{p}$;
else SO returns a half-space $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b$ containing $\mathcal{K}$ but $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{p} > b$;
E_{i+1} = the minimum ellipsoid $\supseteq E_i \cap \{\boldsymbol{x} : \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b\}$;