Lecture 3

Monday, 2 February 2026 11:11 PM

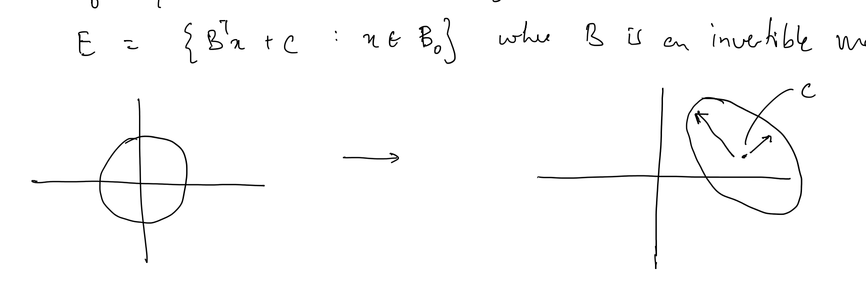
Ellipsoid Algo

First known polynomial-time algo for solving LPs (simplex is not known to be polynomial-time)
Very impt. in theoretical computer science, can solve LPs in time poly(n), even if the # constraints is exponential in n
Practically, both simplex & interior-pt. algos perform much better.

Basic problem solved by ellipsoid algo is <u>feasibility</u>:
Given a bounded convex set $P \subseteq \mathbb{R}^n$, find $x \in P$, if it exists.

Will show later that we can reduce solving a linear program (i.e., optimization) to this feasibility problem.

So what is an ellipsoid?
An ellipsoid is an affine transformation of the unit ball.
i.e., $B_0 = \{x \in \mathbb{R}^n : x^T x \leq 1\}$
Then $E = \{B^T x + c : x \in B_0\}$ where B is an invertible matrix



where the axes now correspond to the eigenvectors of B^T, and the lengths correspond to the eigenvalues.

Claim: the ellipsoid $E = \{B^T x + c : x \in B_0\}$ is equivalent to the set $\{x \in \mathbb{R}^n : (x-c)^T B^T B (x-c) \leq 1\}$
(example, check if B is a diagonal matrix)

Now any positive definite matrix B can be written as
$A = B^T B$, where B is also positive definite

Hence we write an ellipsoid as
$$E(c, A) = \{x \in \mathbb{R}^n : (x-c)^T A^{-1} (x-c) \leq 1\}$$
so that, if $A^{-1} = B^T B$,
$$E(c, A) = \{B^T x + c : x \in B_0\}$$

Volume of an ellipsoid:
- volume of ball of radius $R = \dfrac{\pi^{n/2} R^n}{\Gamma(\frac{n}{2}+1)}$
  (gamma fn, $\sim (\frac{n}{2}+1)!$ if n is even)
- volume of ellipsoid $E(c, A) = Vol(B_0) \times \det(B)$, where $A^{-1} = B^T B$
- what about polyhedra?

<u>Defn</u>: Vectors $v_0, v_1, \ldots, v_k \in \mathbb{R}^n$ are <u>affinely ind</u> if the vectors $v_1 - v_0, v_2 - v_0, \ldots, v_k - v_0$ are linearly independent.

Then if $v_0, v_1, \ldots, v_k$ are affinely independent,
$$vol(CH(v_0, \ldots, v_k)) = \frac{1}{n!} \left| v_1 - v_0 \quad v_2 - v_0 \quad \cdots \quad v_k - v_0 \right|$$

So if $v_0, v_1, \ldots, v_k$ are binary vectors $\in \{0,1\}^n$,
$$vol(CH(v_0, \ldots, v_k)) \geq 1/n!$$

Basic idea of ellipsoid.

Given a polytope of the form $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, we maintain an ellipsoid that contains the feasible region, iteratively shrink this ellipsoid

- Let E_0 be an ellipsoid s.t. $P \subseteq E_0$
- For $k = 0, 1, \ldots$
  - if center c_k of $E_k = E(c_k, A_k) \notin P$,
  - find a "violated constraint" / "separating hyperplane" $ax \leq b$ s.t. $a \cdot c_k > b$
  - find a smaller ellipsoid $E_{k+1}(c_{k+1}, A_{k+1})$ s.t. $E_k \cap \{x : ax \leq b\} \subseteq E_{k+1}$
  - $k \leftarrow k+1$

Lemma: Given $E_k = E(c_k, A_k)$ s.t. $P \subseteq E_k$ and a violated constraint $ax \leq b$ (s.t. $ac_k > b$), can find an ellipsoid E_{k+1} s.t.
(i) $P \subseteq E_k \cap \{x : ax \leq b\} \subseteq E_{k+1}$
(ii) $\dfrac{vol(E_{k+1})}{vol(E_k)} < e^{-1/(2n+1)}$

Thus after K iterations, $vol(E_k) \leq e^{-K/2n+1}$
Thus to run ellipsoid, we need:
1. An initial ellipsoid $E_0 \supseteq P$
2. A lower bound ε on the volume of P, if non-empty
   (note that P must be full-dimensional)
3. A "separation oracle": given $\hat{x} \in \mathbb{R}^n$, either accept $\hat{x} \in P$
   or return a, b s.t.
   $ax \leq b \quad \forall x \in P$
   $a\hat{x} > b$

Note that we don't need to know the entire LP in advance! Just need E_0, ε, a separation oracle.

Step 1: Getting E_0
LP-specific. E.g., if $P \subseteq [0,1]^n$,
then $E_0 = E(0, nI)$

Step 2: Lower bound ε
For combinatorial optimization, $P \subseteq [0,1]^n$.
If P is full-dimensional (contains $n+1$ affinely independent binary vectors) then
$vol(P) \geq \dfrac{1}{n!}$

Step 3: Separating hyperplane: application specific.

This completes the description of ellipsoid <u>for feasibility</u>.

Ok, what about optimization?

Problem is: $\min\{c^T x ; Ax \geq b\}$
Basic idea: find smallest λ s.t. $P_\lambda = \{x \in \mathbb{R}^n : Ax \geq b, c^T x \leq \lambda\}$
is feasible
by binary search.

Q1. How many steps of binary search?
Q2. What about lower bound on size of P_λ?

Assume A, b, c are integral (by scaling). & $\|A\|_\infty, \|b\|_\infty \leq M$.
Claim: For a polytope $P = \{x : Ax \leq b\}$, at any extreme pt.,
$$\# \text{ bits in } x^* \leq 2n \log(Mn)$$
(without proof, but since for any extreme pt. $x^* = A_B^{-1} b_B$)

Thus any extreme pt. $x^* = \dfrac{p}{q}$ where $p, q \in \mathbb{Z}_+$, $q \neq 0$,
$$p, q \in \{-2^{2n \log Mn}, 2^{2n \log Mn}\}$$

(Q1) Let $C_{max} = \|c\|_\infty$ (integral by assumption). Then the optimal soln. for $\{\min c^T x : Ax \leq b\}$ also has bit complexity $2n \log(Mn C_{max})$.
Thus can find OPT in $O(2n \log Mn C_{max})$ iterations.

(Q2) What about lower bound on size of P_λ?
We know that if P_λ is feasible, then an extreme pt. of P is feasible.

Claim: An n-dimensional hypercube with sides of length $2^{-2n \log Mn}$ contains at most one point of bit complexity $O(2n \log Mn)$

So if $\varepsilon < 2^{-2n \log Mn}$, then either P_λ is feasible, or it has a single extreme pt. of P, which can be found (using Diophantine approximations)