Lecture 22

Plan: 1) Finish matroid union
2) Ellipsoid

Ellipsoid Algorithm

- General purpose convex opt. alg.
- Poly time in many situations, e.g., linear programming
  (not necessarily strongly polynomial).
For \( \max \{c^T x : A x \leq b \} \) can solve in \( \mathcal{O}(\langle A \rangle \langle b \rangle) \leq \log |a_{ij}|, 3 \log |b| \) if \( a_{ij}, b \in \mathbb{R} \).
- Slow for LP in practice.
- Contrast of Simplex which is fast in practice but not provably poly.

- May consequences for complexity of combinatorial opt problems.

- There is also interior point methods (Karmarkar '84) that solve LP in poly time, & fast in practice, but not as versatile for theory.
Consequences

Given convex set \( P \subseteq \mathbb{R}^n \), (e.g. a polyhedron), consider two problems:

- **Separation (SEP):**

  Given \( y \in \mathbb{R}^n \), decide if \( y \in P \), if not return separately hyperplane \( i.e. c \in \mathbb{R}^n \) s.t. \( c^T y > \max \{ c^T x : x \in P \} \).
**Optimization (OPT)**

Given vector $c \in \mathbb{R}^n$, find $x$ maximizing $c^T x$ on $P$.

**Examples**

- **Linear programming:**

  $P = \{ x : A x \leq b \}$

  How to solve SEP?

  $P = \{ x : a_i^T x \leq b_i \}$, so just check for each $i$ if $a_i^T x \leq b_i$;
  if not for some $i$, output $a_i^T x \leq b_i$ as separating hyperplane.
efficient if $A$ is part of the input.

\[ \text{OPT for } P = \{ x : Ax \leq b \} \]

is just LP.

\[ \text{max } c^T x \]
subject to \( Ax \leq b \)

(SEP easy, OPT seems hard.)

Matroid polytope:
$M = (E, I)$ matroid,

$P = \text{conv}(\{1_I : 1 \leq I \leq 3\})$

matroid polytope

we know a face characterization.

Thus

$P = \{x \in \mathbb{R}^E : x(S) \leq 1_I(S) \forall S \in E$

$x \geq 0 \forall e \in E\}.$

However, exponentially many constraints!

Even if we can compute
rank function \( r_M \), SEP not obviously efficient!

- OPT for \( P \) is just greedy algorithm for the matroid.
  \( \text{OPT} = \max \text{ cost indep set} \).
  (\text{OPT easy, SEP seems hard.})

- Matroid intersection polytope:
  \( \text{OPT} \text{?? SEP??} \)
- Amazing Result:

  Ellipsoid method & consequences in combo, opt.

Theorem (Grötschel, Lovász, Schrijver '81) For a family $P$ of convex bodies,

\[ \text{SEP for } P \text{ is poly-time solvable } \iff \text{OPT for } P \text{ is poly-time solvable.} \]

Proof idea:

Ellipsoid algorithm: can solve OPT using calls to SEP.
Reduces to \( \equiv \) using "polar" \( P^* \) of \( P \); we won't cover.

- Actually, if \( P \) is "nondegenerate" enough, don't need \( SEP \), just need membership (MEM).

\textbf{MEM!} decide if \( x \in P \).

\textbf{Thm (GLS '88):} Given ball of radius \( E \) contained in \( P \), ball of radius \( R \) containing \( P \), (and a MEM oracle to \( P \)) can solve \( SEP \) with

\[
poly (\log (\frac{1}{\epsilon}), \log (R), n) \]

Book

\textit{last time I didn't say} if not given the \( E \)-ball, might never hit \( P \)!
Calls to \textsc{mem}.
Actually, is about approximate versions of \textsc{sep} & \textsc{mem}.

\textbf{Proof:} not covered. \textsc{sep}.

\underline{OPT vs. feasibility}

\begin{itemize}
\item First we solve simpler problem:
\end{itemize}

\underline{\textsc{feas} (feasibility)} Given \textsc{sep} oracle for \textsc{p} find some $x \in \textsc{p}$ or decide $\textsc{p} = \emptyset$.

\textsc{feas} tells you if \textsc{p} empty or not.

\item OPT reduces to \textsc{feas}:

\underline{binary search}:

\[ \max \{ c^T x : x \in \textsc{p} \} \geq L \text{ if} \]
Given a-priori bound
\[-C \leq L \leq C\]
Binary search to find \(\max L : P_L \neq \emptyset\).
Find $\mathcal{P} \cap \mathcal{P}_A$ using FEAS alg.

2) test for $L_1$ between $L_0/C$.

- Optimize to $\epsilon$-precision in $\log\left(\frac{2C}{\epsilon}\right)$ calls to FEAS.

- For LP, can solve exact OPT with $C/\epsilon \leq$ exponential in bitsize of $A, b$.

(details later.)
(finally!)

The Algorithm

- Solves FEAS in time $\text{poly} \left( \log \left( \frac{1}{\varepsilon} \right), \log R, n \right)$ assuming given ball $B(x_0, R)$ of radius $R$ containing $P$, and either $P$ contains ball of radius $\varepsilon$ or $P = \emptyset$.

- $\varepsilon, R$ dependence not a big deal:
  (& actually necessary).  
  For LP with $P = \{ x : Ax \leq b \}$, $\varepsilon, R$ can be assumed exponential in bitsize of $A, b$. 

$\square$ or $P = \emptyset$. 

$P$
using some tricks (we'll see these tricks for a special case.)

**Algorithm idea:**

- Set $E = E_0$, ellipsoid guaranteed to contain $P$.
  (e.g. $E_0$ = outer ball $B(0, R)$.)

- **While** $e \notin P$:
  (if so, just return $e$ & done)
D get separating hyperplane

\[ c^T x \leq d \] (valid for \( P \) but not \( \text{fore} \)).

(actually, assume \( d = c^T e \).

by translating the hyperplane).

From \( \text{SEP oracle} \).

\( E \cap \{ x : c^T x \leq c^T e \} \)
Let $E'$ "smaller ellipse" containing $E \cap \{x : c^T x \leq c^T e^3\}$. 

$E' 
\cap \{x : c^T x \leq c^T e^3\}$

(Can take $E'$ to be minimum volume ellipsoid containing $E \cap \{x : c^T x \leq c^T e^3\}$, can find $E'$ efficiently!)
0 Set $E \leftarrow E'$.

Runtime:

- Volume Lemma:
  \[
  \text{vol}(E') \leq \frac{1}{2^{(n+1)^2}} \text{vol}(E).
  \]

- As $E$ always contains $P$, alg. must terminate in
  \[
  \leq 2(n+1) \log \left( \frac{\text{vol}(E_0)}{\text{vol}(P)} \right)
  \]
  iterations.

(Cif $P$ contains ball of radius $E$, contain ball of radius $R$, \leq 2(n+1) \log \left( \frac{R}{E} \right)$)
If after $2(n+1)n\log(n \log(R/E))$ iterations the algorithm hasn't terminated, output $P=E$.

Proof of lemma

Bounding $R,E$. How to compute $E$. Issues