CS38
Introduction to Algorithms
Lecture 16
May 22, 2014

Outline

• Linear programming
  – LP duality
  – ellipsoid algorithm
  * slides from Kevin Wayne

• coping with intractibility
  – NP-completeness

LP Duality

Primal problem.

(P) \( \max 13x + 23y \)
\[ \begin{align*}
& \text{s.t.} \\
& 5x + 15y \leq 480 \\
& 4x + 4y \leq 160 \\
& 15x + 20y \leq 1190
\end{align*} \]
\( x, y \geq 0 \)

Idea. Add nonnegative combination \((C, H, M)\) of the constraints s.t.

\( 13x + 23y \leq (5C + 4H + 35M) x + (5C + 4H + 20M) y \)
\( \leq 480C + 160H + 1190M \)

Dual problem. Find best such upper bound.

(D) \( \min 480C + 160H + 1190M \)
\[ \begin{align*}
& \text{s.t.} \\
& 5C + 4H + 35M \geq 13 \\
& 15C + 4H + 20M \geq 23 \\
& C, H, M \geq 0
\end{align*} \]

LP Duals

Canonical form.

(P) \( \max c^T x \)
\[ \begin{align*}
& \text{s.t.} \\
& Ax \leq b \\
& x \geq 0
\end{align*} \]

(D) \( \min y^T b \)
\[ \begin{align*}
& \text{s.t.} \\
& A^T y \geq c \\
& y \geq 0
\end{align*} \]

Double Dual

Canonical form.

(P) \( \max x^T y \)
\[ \begin{align*}
& \text{s.t.} \\
& x \geq 0 \\
& y \geq 0
\end{align*} \]

(D) \( \min y^T b \)
\[ \begin{align*}
& \text{s.t.} \\
& A^T y \geq c \\
& y \geq 0
\end{align*} \]

Property. The dual of the dual is the primal.

Pf. Rewrite (D) as a maximization problem in canonical form; take dual.

(D') \( \max -y^T b \)
\[ \begin{align*}
& \text{s.t.} \\
& -A^T y \leq -c \\
& y \geq 0
\end{align*} \]

DD \( \min -c^T z \)
\[ \begin{align*}
& \text{s.t.} \\
& -A^T z \leq -b \\
& z \geq 0
\end{align*} \]

Taking Duals

LP dual recipe.

<table>
<thead>
<tr>
<th>( \text{Primal (P)} )</th>
<th>( \text{maximize} )</th>
<th>( \text{minimize} )</th>
<th>( \text{Dual (D)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{constraints} )</td>
<td>( \alpha \geq \beta )</td>
<td>( \gamma \leq \delta )</td>
<td>( \beta, \delta )</td>
</tr>
<tr>
<td>( \text{variables} )</td>
<td>( \gamma \geq \delta )</td>
<td>( \gamma \leq \delta )</td>
<td>( \delta )</td>
</tr>
<tr>
<td>( \text{nonnegative} )</td>
<td>( \gamma \leq \delta )</td>
<td>( \gamma \leq \delta )</td>
<td>( \gamma \leq \delta )</td>
</tr>
</tbody>
</table>

Pf. Rewrite LP in standard form and take dual.
Strong duality

LP Strong Duality

Theorem. [Gale-Kuhn-Tucker 1951; Dantzig von Neumann 1947]
For $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, c \in \mathbb{R}^n$ if (P) and (D) are nonempty, then $\max = \min$.

\[(P) \quad \max c^T x \quad \text{s. t.} \quad Ax \leq b, \quad x \geq 0\]
\[(D) \quad \min y^T b \quad \text{s. t.} \quad Ay \geq c, \quad y \geq 0\]

Generalizes:
- Dilworth’s theorem.
- König-Egerváry theorem.
- Max-flow min-cut theorem.
- von Neumann’s minimax theorem.
- ...

Pf. (ahead)

LP Weak Duality

Theorem. For $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, c \in \mathbb{R}^n$ if (P) and (D) are nonempty, then $\max \leq \min$.

\[(P) \quad \max x^T \quad \text{s. t.} \quad dx \leq b \quad x \geq 0\]
\[(D) \quad \min y^T b \quad \text{s. t.} \quad A^y \geq c \quad y \geq 0\]

Pf. Suppose $x \in \mathbb{R}^n$ is feasible for (P) and $y \in \mathbb{R}^n$ is feasible for (D).
- $y \geq 0$, $Ax \leq b$ $\implies$ $y^T A x \leq y^T b$
- $x \geq 0$, $A^y \geq c$ $\implies$ $x^T A^y \geq y^T b$
- Combine: $x^T \leq y^T A x \leq y^T b$

Projection Lemma

Weierstrass’ theorem. Let $X$ be a compact set, and let $f(x)$ be a continuous function on $X$. Then $\min \{f(x) : x \in X\}$ exists.

Projection lemma. Let $X \subset \mathbb{R}^n$ be a nonempty closed convex set, and take $y$ not in $X$. Then there exists $x' \in X$ with minimum distance from $y$.
Moreover, for all $x \in X$ we have $(y - x')^T(y - x') \leq 0$.

Pf. Suppose $x \in \mathbb{R}^n$ is feasible for (P) and $y \in \mathbb{R}^n$ is feasible for (D).
- $y \geq 0$, $Ax \leq b$ $\implies$ $y^T A x \leq y^T b$
- $x \geq 0$, $A^y \geq c$ $\implies$ $x^T A^y \geq y^T b$
- Combine: $x^T \leq y^T A x \leq y^T b$

Theorem. For $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, c \in \mathbb{R}^n$ if (P) and (D) are nonempty, then $\max = \min$.

\[(P) \quad \max c^T x \quad \text{s. t.} \quad Ax \leq b, \quad x \geq 0\]
\[(D) \quad \min y^T b \quad \text{s. t.} \quad Ay \geq c, \quad y \geq 0\]
**Separating Hyperplane Theorem**

**Theorem.** Let \( X \subseteq \mathbb{R}^n \) be a nonempty closed convex set, and take \( y \not\in X \). Then there exists a hyperplane \( H = \{ x \in \mathbb{R}^n : a^T x = \alpha \} \) where \( a \in \mathbb{R}^n \), \( \alpha \in \mathbb{R} \) that separates \( y \) from \( X \).

**Pf.** Let \( s^* \) be closest point in \( X \) to \( y \).

- By projection lemma, \((a - a^*)^T (x - s^*) \leq 0 \) for all \( x \in X \).
- Choose \( a - a^* \) not equal to \( 0 \) and \( \alpha = a^T s^* \).
- If \( x \in X \), then \( (a - a^*)^T (x - s^*) \geq 0 \).
- Thus \( a^T x \geq a^T s^* = \alpha \).
- Also, \( a^T y - a^T s^* = \alpha - \| a \|^2 \alpha < \alpha \).

**Farkas’ Lemma**

**Theorem.** For \( a \in \mathbb{R}^{n^*} \), \( b \in \mathbb{R}^n \) exactly one of the following two systems holds:

\[
(i) \exists x \in \mathbb{R}^n \quad a^T x = b, \quad x \geq 0
\]

\[
(ii) \exists y \in \mathbb{R}^n \quad y^T b = 0, \quad y^T a \leq 0
\]

**Pf.** (not both) Suppose \( x \) satisfies (i) and \( y \) satisfies (ii).

Then \( 0 \geq y^T b = y^T a \geq 0 \), a contradiction.

**Pf.** (at least one) Suppose (i) infeasible. We will show (ii) feasible.

- Consider \( S = \{ x \in \mathbb{R}^n : a^T x = 0 \} \) and note that \( b \) not in \( S \).
- Let \( y \in \mathbb{R}^n \), \( a \in \mathbb{R} \) be a hyperplane that separates \( b \) from \( S \):
  \begin{align*}
  y^T b < \alpha, \quad y^T x \geq \alpha \quad \text{for all} \quad x \in S.
  \end{align*}

\( 0 \in S \Rightarrow a \not\in S \Rightarrow y^T a < 0 \)

\( y^T a \geq 0 \) for all \( x \geq 0 \Rightarrow y^T a > 0 \), since \( a \) can be arbitrarily large.

**Another Theorem of the Alternative**

**Corollary.** For \( a \in \mathbb{R}^{n^*} \), \( b \in \mathbb{R}^n \) exactly one of the following two systems holds:

\[
(i) \exists x \in \mathbb{R}^n \quad a^T x = b, \quad x \geq 0
\]

\[
(ii) \exists y \in \mathbb{R}^n \quad y^T b = 0, \quad y^T a \leq 0
\]

**Pf.** Apply Farkas’ lemma to:

- \( (i) \exists y \in \mathbb{R}^n, z \in \mathbb{R}^n \) s.t. \( x^T y = b, x, y \geq 0 \)
- \( (ii) \exists y \in \mathbb{R}^n \) s.t. \( y^T b = 0, y^T a \leq 0, y \geq 0 \)

**LP Strong Duality**

**Theorem.** [strong duality] For \( a \in \mathbb{R}^{n^*} \), \( b \in \mathbb{R}^n \), \( c \in \mathbb{R}^n \), if \( (P) \) and \( (D) \) are nonempty then \( \max = \min \).

\[
(P) \quad \max c^T x \quad \text{s.t.} \quad a^T x \leq b, \quad x \geq 0
\]

\[
(D) \quad \min y^T b \quad \text{s.t.} \quad y^T a \geq c, \quad y \geq 0
\]

**Pf.** (max \( \leq \) min) Weak LP duality.

**Pf.** (min \( \leq \) max) Suppose \( \max < \alpha \). We show \( \min < \alpha \).

- By definition of \( \alpha \), \( (i) \) infeasible \( \Rightarrow (ii) \) feasible by Farkas’ Corollary.

**Ellipsoid algorithm**

Let \( y, z \) be a solution to (ii).

**Case 1.** \( x = 0 \)

- Then, \( (y \in \mathbb{R}^n : y^T b = 0, y^T a = 0) \) is feasible.
- Farkas Corollary \( \{ x \in \mathbb{R}^n : a^T x = 0 \} \) is infeasible.
- Contradiction since by assumption \( (P) \) is nonempty.

**Case 2.** \( x > 0 \)

- Scale \( y, z \) so that \( y \) satisfies (ii) and \( x = 1 \).
- Resulting \( y \) feasible to \( (D) \) and \( y^T b < \alpha \).
To find a point in $P$:
- Maintain ellipsoid $E$ containing $P$.
- If center of ellipsoid $z$ is in $P$ stop;
- otherwise find hyperplane separating $z$ from $P$.
- Find smallest ellipsoid $E'$ containing half-ellipsoid.
- Repeat.
**Ellipsoid Algorithm**

**Goal.** Given $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, find $x \in \mathbb{R}^n$ such that $Ax \leq b$.

**Ellipsoid algorithm.**
- Let $E_k$ be an ellipsoid containing $P$.
- Fix $b = 0$.
- While center $z$ of ellipsoid $E$ is not in $P$:
  - Find a constraint, say $D_2 = \{ x : a_2 x \leq a \}$, that is violated by $z$.
  - Let $E_{k+1}$ be min volume ellipsoid containing $E_k \cap \{ x : a_2 x \leq a \}$.
  - $k = k+1$.

**Shrinking Lemma**

**Ellipsoid.** Given $D \in \mathbb{R}^{n \times n}$ positive definite and $x \in \mathbb{R}^n$, then

$$E = \{ x : y^T D^{-1} (y - x) \leq 1 \}$$

is an ellipsoid centered on $x$ with $\text{vol}(E) = \sqrt{\text{det}(D)}$.

**Key lemma.** Every half-ellipsoid $\frac{1}{2} E$ is contained in an ellipsoid $E'$ with $\text{vol}(E') / \text{vol}(E) \leq e^{-\frac{1}{2} \delta^2/2}$.

**Ellipsoid algorithm terminates after at most $2(n+1) \ln (\text{vol}(E_k)/\text{vol}(P))$ steps.
Ellipsoid Algorithm

**Theorem.** Linear Programming problems can be solved in polynomial time.

**Pf sketch.**
- Shrinking lemma.
- Set initial ellipsoid \( E_0 \) so that \( \text{vol}(E_0) \leq 2^m \).
- Perturb \( Ax \cdot b \) to \( Ax \cdot b + (\delta) \) either \( P \) is empty or \( \text{vol}(P) \geq 2^m \).
- Bit complexity (to deal with square roots).
- Purify to vertex solution.

**Caveat.** This is a theoretical result. Do not implement.

\[ \text{Time complexity } O\left(\frac{m^3L}{\epsilon^2}\right) \]

where \( L \) = number of bits to encode input.

---

### Decision problems + languages

- A problem is a function: \( f: \Sigma^* \rightarrow \Sigma^* \)
- Simple. Can we make it simpler?
- Yes. **Decision problems:** \( f: \Sigma^* \rightarrow \{\text{accept}, \text{reject}\} \)
- Does this still capture our notion of problem, or is it too restrictive?

---

### Decision problems + languages

- Example: factoring:
  - given an integer \( m \), find its prime factors
  - decision version:
    - given 2 integers \( m, k \), accept iff \( m \) has a prime factor \( p < k \)
- Can use one to solve the other and vice versa. True in general.

---

### Decision problems + languages

- For most complexity settings a problem is a **decision problem**: \( f: \Sigma^* \rightarrow \{\text{accept}, \text{reject}\} \)
- Equivalent notion: **language**
  - \( L \subseteq \Sigma^* \)
  - the set of strings that map to "accept"
- Example: \( L = \) set of pairs \((m,k)\) for which \( m \) has a prime factor \( p < k \)

---

### Search vs. Decision

- **Definition:** given a graph \( G = (V, E) \), an independent set in \( G \) is a subset \( V' \subseteq V \) such that for all \( u, w \in V' \) \( (u, w) \notin E \)
- A problem:
  - given \( G \), find the largest independent set
- This is called a **search problem**
  - searching for optimal object of some type
  - comes up frequently
Search vs. Decision

- We want to talk about languages (or decision problems)
- Most search problems have a natural, related decision problem by adding a bound \( k \); for example:
  - search problem: given \( G \), find the largest independent set
  - decision problem: given \((G, k)\), is there an independent set of size at least \( k \)

The class \( NP \)

**Definition:** \( \text{TIME}(t(n)) = \{ L : \text{there exists a TM } M \text{ that decides } L \text{ in time } O(t(n)) \} \)

\( P = \bigcup_{k \geq 1} \text{TIME}(n^k) \)

**Definition:** \( \text{NTIME}(t(n)) = \{ L : \text{there exists a NTM } M \text{ that decides } L \text{ in time } O(t(n)) \} \)

\( \text{NP} = \bigcup_{k \geq 1} \text{NTIME}(n^k) \)

Poly-time verifiers

- \( \text{NP} = \{ L : L \text{ decided by poly-time NTM} \} \)
- Very useful alternate definition of \( \text{NP} \):
  - **Theorem:** language \( L \) is in \( \text{NP} \) if and only if it is expressible as:
    \[ L = \{ x | \exists y, |y| \leq |x|^k, (x, y) \in R \} \]
    where \( R \) is a language in \( P \).
  - poly-time TM \( M_R \) deciding \( R \) is a “verifier”

Poly-time verifiers

- Example: 3SAT expressible as
  
  \[ 3\text{SAT} = \{ \varphi : \varphi \text{ is a 3-CNf formula for which } \exists \text{ assignment } A \text{ for which } (\varphi, A) \in R \} \]
  
  \[ R = \{ (\varphi, A) : A \text{ is a sat. assign. for } \varphi \} \]
  
  - satisfying assignment \( A \) is a “witness” of the satisfiability of \( \varphi \) (it “certifies” satisfiability of \( \varphi \))
  - \( R \) is decidable in poly-time

Poly-time reductions

- Type of reduction we will use:
  - “many-one” poly-time reduction

\[ \begin{array}{ccc}
\text{A} & \text{f} & \text{B} \\
\text{yes} & & \text{yes} \\
\text{no} & \text{f} & \text{no}
\end{array} \]

reduction from language A to language B
Poly-time reductions

Definition: A \leq_p B ("A reduces to B") if there is a poly-time computable function f such that for all w

• function f should be poly-time computable

\begin{align*}
\text{Definition: } f &: \Sigma^* \to \Sigma^* \text{ is poly-time computable if for some } g(n) = n^c(1) \text{ there exists a } g(n)-\text{time TM } M_f \text{ such that on every } w \in \Sigma^*, M_f \text{ halts with } f(w) \text{ on its tape.}
\end{align*}

Poly-time reductions

Theorem: if A \leq_p B and B \in P then A \in P.

Proof:
• a poly-time algorithm for deciding A:
• on input w, compute f(w) in poly-time.
• run poly-time algorithm to decide if f(w) \in B
• if it says "yes", output "yes"
• if it says "no", output "no"

Hardness and completeness

• Reasonable that can efficiently transform one problem into another.

• Surprising:
  • can often find a special language L so that every language in a given complexity class reduces to L!
  • powerful tool

Hardness and completeness

• Recall:
  • a language L is a set of strings
  • a complexity class C is a set of languages

Definition: a language L is C-hard if for every language A \in C, A poly-time reduces to L; i.e., A \leq_p L.

meaning: L is at least as "hard" as anything in C

Definition: a language L is C-complete if L is C-hard and L \in C

meaning: L is a "hardest" problem in C
Lots of NP-complete problems

- logic problems
  - 3-SAT = \{\phi : \phi \text{ is a satisfiable 3-CNF formula}\}
  - NAE3SAT, (3,3)-SAT
  - Max-2-SAT

- finding objects in graphs
  - independent set
  - vertex cover
  - clique

- sequencing
  - Hamilton Path
  - Hamilton Cycle and TSP

- problems on numbers
  - subset sum
  - knapsack
  - partition

- splitting things up
  - max cut
  - min/max bisection