COMS 4995-8: Advanced Algorithms (Spring'21) Mar 23, 2021

Lecture 19: Linear Programming Duality, Ellipsoid Algorithm

Instructor: *Alex Andoni* Scribes: *Qianjun Chen, Cheng Zhang*

1 Overview

Today's lecture is about LP duality, and ellipsoid algorithm, which is the first poly-time algorithm for LP.

2 Linear Programming Duality

Standard form:

$$v^* = min\, c^T x$$

$$s.t.\ Ax = b$$

$$x \geq 0$$

which $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$.

1. If we want to prove optimal value $v^* \leq v$, just show some $x \in F$, $c^T x \leq v$.
2. $v \geq ...?$

Definition 1. *Dual of standard form linear programming:*

$$\omega^* = max\, b^T y$$

$$s.t.\ A^T y \leq c$$

$$y \geq \mathbb{R}^m$$

You'll notice that conveniently the dual of the standard form is what we call the general form of linear program.

Theorem 2 (weak duality)**.**

$$\omega^* \leq v^*$$

So in a sense, these dual linear program, will provide such lower bound. It's basically coming from the weak duality theorem. Let's prove this, and basically, this will motivate how we came up with a linear program.

Proof. Fix $x \in F$: $Ax = b$ and $x \geq 0$. Fix any $y \in \mathbb{R}^m$, $y^T Ax = y^T b$, which $y^T A$ is a vector $\in \mathbb{R}^n$. Remember, in standard form, x is positive, and the standard form is minimizing the product of x with some vector c.
Suppose y such that $y^T A \leq c^T$ (each coordinate). This implies that:

1. $y^T b = y^T Ax \leq c^T x$
2. $max\, y^T b \leq min\, c^T x$. y: $y^T A \leq c^T$, which is ω^*. x: $Ax = b$, $x \geq 0$, which is υ^*.

□

So there are two questions. The first is, is this a good lower bound?

3 Dual of Dual

If we take a dual, and take its dual, we get the primal:

$$\upsilon^* = min\, c^T x$$

$$s.t.\ Ax = b$$

$$x \geq 0$$

This is the dual:

$$\omega^* = max\, b^T y$$

$$s.t.\ A^T y \leq c$$

Mechanically, once you put the primal in the standard form, then there is an. automatic way to get the dual. We need to take our dual, put it in a standard form.
Dual return to the standard form:

$$-\omega^* = min\ - b^T y, A^T y \leq c$$

$$= min\ - b^T (y^+ - y^-)$$

$$A^T (y^+ - y^-) + \delta$$

which y^+, y^-, $\delta \geq 0$, y^+, $y^- \in \mathbb{R}^m$, $\delta \in \mathbb{R}^n$. This is our dual return to the standard form.

Dual of the dual:
unknowns: $z \in \mathbb{R}^n$.
Objective function: $max\, c^T z$.
Constraints:
The constraint matrix of dual return to the standard form is :

A^T	$-A^T$	I_n
y^+	y^-	δ

So the constraints will look like this:

A		$-b$
-A	$\cdot z \leq$	$+b$
I		0

Each row here we have:

$$A_i^T z = b_i \Leftarrow \begin{cases} A_i^T \cdot z \leq -b_i \\ -A_i^T \cdot z \leq +b_i \\ z_i \leq 0 \end{cases}$$

which $A_i = ith$ row of A.
Let's replace the variable $z' = -z$. What we obtain is that Dual of (-Dual):

$$max\, c^T(-z')$$

$$s.t.\; Az' = b$$

$$z' \geq 0$$

So Dual of the Dual:

$$-max\, -c^T(-z') = min\, c^T z'$$

$$s.t.\; Az' = b$$

$$z' \geq 0$$

which is exactly equal to primal.

Remark 3. *It does not prove immediately the strong duality.*

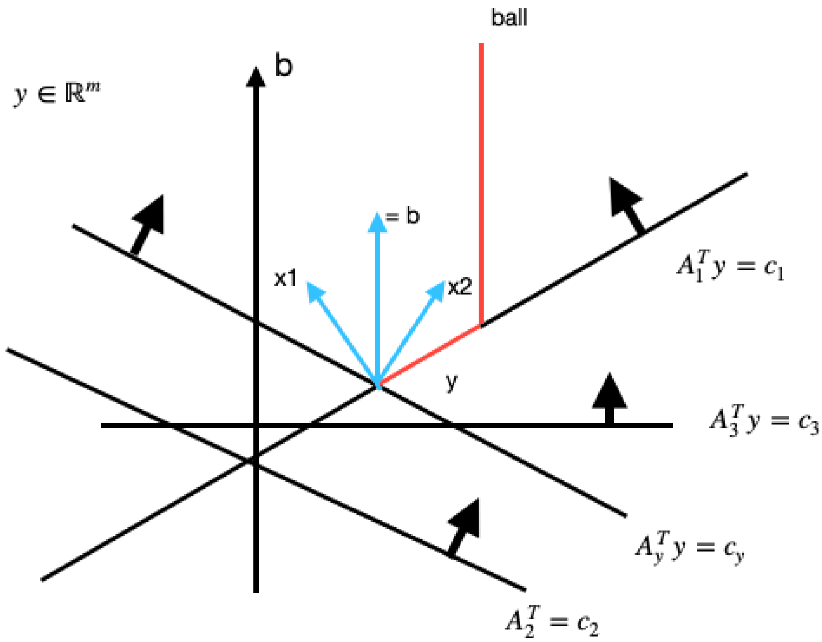
4 Strong Duality

Theorem 4. $w^* = v^*$

$$\boxed{\begin{aligned} v^* &= \min c^T x \\ &\quad Ax = b, x \geq 0 \\ w^* &= \max b^T y \\ &\quad A^T y \leq c \end{aligned}}$$

The actual proof of strong duality is painful, there are a lot case analysis, so we want to do pseudo proof by its intuition.
Intuition: Consider dual with $y \to -y$, $c \to -c$
$\Rightarrow -w^* = \min b^T y$
s.t, $A^T y \geq -c$, $y \in \mathbb{R}^m$

Consider a ball is pulled down by gravity in a m-dimensional space. Each equation $A_i^T y = c_i$ forms a hyper-plane in this space. The ball can only be dropped into the convex composed by these hyper-planes.



What does it mean that ball is resting?

At this point, there must exist the same force pushing it upward to against the gravity force pulling it down. This force is exactly these hyper-planes the ball is resting on.

$\Rightarrow$ a set S of constrains/half-spaces

(1) $i \in S \Rightarrow A_i y = c_i$ (tight constrains)

(2) $\exists$ coefficients $x_1, \cdots, x_n \in \mathbb{R}_+$
$\sum x_i A_i = b \Leftrightarrow Ax = b$

(2') if $i \notin S$, it is not a hyper-plane on which the ball is resting, then $x_i = 0$

Value of primal: $-c^T x$
Value of dual: $-b^T y$
To compute these two values based on part (1) and part (2) above:

$$\sum x_i A_i y = y^T A x \overset{①}{=} \sum x_i c_i = c^T x$$
$$\overset{②}{=} y^T b = b^T y$$

Therefore, we've proven the value of primal is equal to the value of dual. $\square$

Note: for every $i \in [n]$, we must have: $x_i = 0$, or $A_i y = c_i$, or both. (complementary slackness)

5 Ellipsoid Algorithm [Khachiyan '79]

It solves feasibility problem.
First, we need to define what feasibility problem is, feasibility problem (FP) is to find $x \in Q_v$ if exists, for such:

$$Q_v = \{x : Ax \geq b, c^T x \leq v\}$$

in general form, that is LP : $v^* = \min c^T x$, $Ax \geq b$.

The feasibility problem is definitely no harder than solving the LP, since once we solve the LP, we will find the optimal x, that can be used to solve FP.

Remark 5. *can solve LP using FP with ploy-time slow-down*

Proof. Do binary search on v.
Let R = upper bound on value of $|v^*|$
Without solving a linear program, it is easier to find an upper bound of absolute value of the V^*, in particular:

$$\lg_2 R \leq ploy(n, m) \cdot B$$

where m is the number of constraints, B is the number of bits to represent each input number

---

Algorithm 1 Binary Search

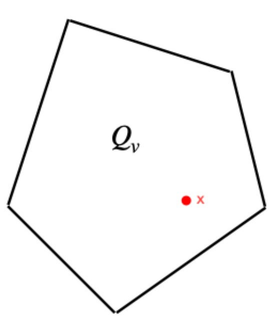
---

set $l = -R$, $r = +R$
$m = \frac{l+r}{2}$
if Q_m is feasible **then**
    recursive on $l = l$, $r = m$
else
    recursive on $l = m$, $r = r$
end if
stop when $r - l$ is sufficiently small, $\approx \frac{1}{R^{O(1)}}$
then optimal solution $v^* \approx l \approx r$

---

number of binary search iterations $= O(\log R) = O(poly(n, m)B)$,
solving $Q_v = \{x : Ax \geq b, c^T x \leq v\}$ □

From the perspective of this problem, there is no really difference between inequality $Ax \geq b$ and inequality $c^T x \leq v$.



The goal is to find any x point inside the space.

6 Next Class

How does Ellipsoid Algorithm find such x point, then we will start talking about gradient descent for general functions.

References

[Khachiyan '79] L. G. Khachiyan, "A polynomial algorithm in linear programming", Dokl. Akad. Nauk SSSR, 244:5 (1979), 1093–1096