# LINEAR PROGRAMMING

## 1. Introduction

Optimization is a central objective in computer science. Linear programming (LP) is a clean framework for optimization with the following "good" properties:
— Many problems fit into this framework.
— The problems can be efficiently solved.
— Comes with a geometric picture and a theory.
— Allows to approximately solve discrete optimization problems.

An optimization problem is usually of the form "maximize $f(x)$ where $x$ satisfies the following...". In LP, both the objective $f$ and the constraints are linear. We start by introducing the framework. It is helpful to keep a geometric picture in mind; in our running example we shall work over the plane.

We work with $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. This is $n$-dimensional Euclidean space (it has an algebraic structure, a geometric structure, a topological structure, etc.).

**Example 1.** *The point $x = (x_1, x_2)$ in the plane $\mathbb{R}^2$.*

We have an inner product structure (a.k.a. dot product) on $\mathbb{R}^n$:

$$\langle x, y \rangle = \sum_{i \in [n]} x_i y_i.$$

**Example 2.** *The inner product between $(1, 0)$ and $(0, 1)$ is zero; these vectors are orthogonal.*

A hyperplane $h$ in $\mathbb{R}^n$ is defined by a non-zero vector $a \in \mathbb{R}^n$ and a translation in $b \in \mathbb{R}$:

$$h = \{x \in \mathbb{R}^n : \langle a, x \rangle = b\}.$$

**Example 3.** *The line defined by $a = (1, 1)$ and $b = 2$.*

A half-space $H$ in $\mathbb{R}^n$ is defined by a non-zero vector $a \in \mathbb{R}^n$ and a translation in $b \in \mathbb{R}$:

$$H = \{x \in \mathbb{R}^n : \langle a, x \rangle \leq b\}.$$

**Example 4.** *The half-plane defined by $a = (1, 1)$ and $b = 2$.*

Given half-spaces $H_1, \ldots, H_k$ in $\mathbb{R}^n$, we can consider their intersection:

$$\bigcap_{i \in [k]} H_i.$$

**Example 5.** *The intersection of the three half-planes:*
- $a = (1, 1)$ *and* $b = 2$
- $a = (0, -1)$ *and* $b = 0$
- $a = (-1, 0)$ *and* $b = 0$

*is a triangle with vertices* $(0,0),(0,2),(2,0)$.

**Remark.** *The intersection of half-spaces is a convex set; that is, if* $x, y$ *belong to this set, then the whole line segment connecting* $x, y$ *belong to the set.*

**Remark.** *We can replace* $\langle a, x\rangle \geq b$ *by* $\langle -a, x\rangle \leq -b$. *We can add equality constraints* $\langle a, x\rangle = b$ *via* $\langle a, x\rangle \leq b$ *and* $\langle -a, x\rangle \leq -b$.

We can represent this more succinctly as follows. For two vectors $x, y \in \mathbb{R}^n$, write

$$x \leq y \iff \forall j \in [n] \ x_j \leq y_j.$$

Let $a^{(i)} \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ be the ones defining $H_i$ for $i \in [k]$. Represent the $a^{(i)}$ by a $k \times n$ matrix:

$$A_{i,j} = a_j^{(i)}$$

and the $b_i$ by a $k$-dimensional vector $b$. We can finally write

$$\bigcap_{i \in [k]} H_i = \{x \in \mathbb{R}^n : Ax \leq b\}.$$

**Example 6.** *The three previous half-planes*

$$A = \begin{bmatrix} 1 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}$$

*and*

$$b = (2,0,0).$$

**Definition.** *The LP problem defined by* $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{k \times n}$ *and* $b \in \mathbb{R}^k$ *is*

$$\max \ \langle c, x\rangle : Ax \leq b.$$

**Example 7.** *The previous three previous half-planes with* $c = (1,2)$*; the value is* $\langle (1,2),(0,2)\rangle = 4$.

We optimize a linear function over a convex polytope. There are three options:
— The set $\{x : Ax \leq b\}$ is empty.
— The set $\{x : Ax \leq b\}$ is non-empty and the maximum is infinite.
— The set $\{x : Ax \leq b\}$ is non-empty and the maximum is finite.

**Exercise.** *Draw an example in the plane for each of the three options.*

We shall focus on the third option, which we call the "feasible and bounded" case.

**Remark.** *If the LP is feasible and bounded, then the maximum is attained at a vertex of the polytope. See example.*

## 2. Reductions

LP appears naturally in many settings, like the distribution of resources. In addition, many algorithmic problems can be reduced to LP. Some examples are the Max Flow problem, and the Shortest Path problem. Let us consider Shortest Path for example.

**Shortest Path.** We have a directed graph $G = (V, E)$ with edge weights given by $W : E \to \mathbb{R}$. We are also given two fixed vertices $s, t \in V$. Out goal is to compute the minimum weight of a path from $s$ to $t$ in $G$. This turns into an LP over $\mathbb{R}^V$ as follows:

$$\begin{aligned} \max x_t : \ & x_s = 0 \\ & \forall e = (u, v) \in E \ \ x_v \le x_u + W(e) \end{aligned}$$

**Exercise.** *What are the $A, b, c$ of this problem?*

**Remark.** *We are solving a minimization problem via a maximization problem!*

Here is some intuition for correctness (we shall not fully prove):

- If we set $x_v = \mathsf{dist}(s, v)$ then all constraints are satisfied (requires a proof), which implies that the maximum is at least $\mathsf{dist} s, t$.
- In the other direction, choose a shortest path $s = v_0 \to v_1 \to \ldots \to v_n = t$. Then, for all $i$,
$$x_{v_{i+1}} \le x_{v_i} + W(v_i, v_{i+1})$$
and get that the maximum is at most the shortest path.

## 3. Duality

How did we start with a minimization problem and ended up with a maximization problem? This basically "always happens". Every LP maximization problems comes with a dual minimization problem, and vice versa. This phenomenon appears in many places in CS and math.

**Definition.** *The standard form of an LP is*

$$\begin{aligned} \max \ \langle c, x \rangle \ : \ & Ax \le b \\ & x \ge 0 \end{aligned}$$

**Remark.** *We added the requirement that $x \ge 0$.*

**Definition.** *The dual program is*

$$\begin{aligned} \min \ \langle b, y \rangle \ : \ & A^T y \ge c \\ & y \ge 0 \end{aligned}$$

**Remark.** *The dual in the standard form is $(A, b, c) \mapsto (-A^T, -c, -b)$.*

**Exercise.** *The dual of the dual is the primal.*

**Lemma 8** (weak duality)**.** *If $x$ is feasible in the primal and $y$ is feasible in the dual then*

$$\langle c, x \rangle \le \langle b, y \rangle.$$

In words, the values of the dual which is a minimization problem provide upper bound on the values of the primal which is a maximization problem (and vice versa).

*Proof.* First, because

$$Ax \le b$$

and because $y \ge 0$, we have

$$\langle y, Ax \rangle \le \langle y, b \rangle;$$

indeed, for all $i$, we have $y_i(Ax)_i \leq y_i b_i$ and we can sum over $i$. Second, because $A^T y \geq c$ and $x \geq 0$, we similarly have

$$\langle c, x\rangle \leq \langle A^T y, x\rangle.$$

Third,

$$\langle y, Ax\rangle = y^T Ax = x^T A^T y = \langle x, A^T y\rangle.$$

Putting it together:

$$\langle c, x\rangle \leq \langle A^T y, x\rangle = \langle y, Ax\rangle \leq \langle y, b\rangle.$$

**Remark.** *The couple of left inner products are in* $\mathbb{R}^n$ *and the right are in* $\mathbb{R}^k$.

□

**Remark.** *Strong duality states that if the primal is feasible and bounded, then the value of the two programs is the same. Not just an inequality, but an equality!*

**Example 9.** *A special case of strong duality is the max-flow min-cut theorem. The dual of the "flow LP" is the "cut LP".*

**Remark.** *Duality is a powerful idea. It replaces a* $\forall$ *quantifier* $(\max \leq \ldots)$ *and a* $\exists$ *quantifier* $(\min \geq \ldots)$. *Instead of proving that somethings is always true, we just need to find an example.*

## 4. Algorithms

There are three general algorithms for LP, which we briefly discuss on a high-level.

**Simplex.** The simplex algorithm was historically first. It operates by walking on the vertices on the polytope (using so-called "pivoting rules") until finding an optimal solution. This algorithm is quite simple but does not always run in polynomial time. In many practical cases, however, it is efficient.

**Ellipsoid.** The algorithm is based on the computational equivalence between optimization and decision. The optimization problem is

"given $A, b, c$, what is $\max \langle c, x\rangle \ : \ Ax \leq b$?"

The decision problem is

"given $A, b$, is there $x$ so that $Ax \leq b$?"

There are several way to implement this equivalence. For example, we can do binary search on the value of $\langle c, x\rangle$ be adding constraints of the form $\langle c, x\rangle \geq v$.

The algorithm solves the decision problem as follows. It maintain an ellipsoid that contains the feasible region (the first is "large enough"). If the center of the ellipsoid is feasible, then we are done. If it is not, we "cut the ellipsoid to two parts by a row of $A$, and recurse". It is possible to show that the volume of the ellipsoid "goes down rapidly enough".

**Interior-point method.** These algorithms iteratively move a point inside the feasible region following various rules that are based on "gradients, etc." until an optimal solution is found.

**Remark.** *All these algorithms are bases on a "separation oracle"; an oracle that given $x$ and $A, b$, either say "in" (meaning that $Ax \le b$), or return $i$ so that $\langle a^{(i)}, x\rangle > b_i$ where $a^{(i)}$ is the $i$'th two of $A$.*

## 5. Summary

LP is an important class of optimization problems. There are efficient algorithms, so when we are able to reduce a problem to an LP then "we solved it". There is a lot of theory behind it; linear algebra, geometry, norms, etc. An important and powerful concept is duality, which replaces max by min.

## Appendix A. Strong duality

Here we explain strong duality, and provide some geometric picture. We have our data $A, b, c$ and we interested in

$$\max \langle c, x\rangle : Ax \le b, x \ge 0.$$

Let us see how to relate this optimization problem to a decision problem. For each value $v \in \mathbb{R}$, we can look at

$$P_v := \{x : Ax \le b, x \ge 0, \langle c, x\rangle \ge v\}.$$

Assume that the LP is feasible and bounded; that is, for small enough $v$ we have $P_v \neq \emptyset$ and for large enough $v$ we have $P_v = \emptyset$. Due to monotonicity (i.e., if $v < v'$ then $P_v \supseteq P_{v'}$), the set $\{v : P_v \neq \emptyset\}$ is a ray of the form

$$\{v : P_v \neq \emptyset\} := (-\infty, v_P].$$

The polytope $P_v$ is non-empty iff the maximization problem has solution $\ge v$. So,

$$v_P = \max\{\langle c, x\rangle : Ax \le b, x \ge 0\}.$$

For the dual problem, look at

$$Q_v := \{y : A^T x \ge c, y \ge 0, \langle b, y\rangle \le v\}.$$

Now, if $v < v'$ then $Q_v \subseteq Q_{v'}$. First, we claim that the dual is feasible and bounded. We already say that (weak duality)

$$\{v : Q_v \neq \emptyset\} \subset [v_P, \infty)$$

so for small enough $v$ we have that $Q_v = \emptyset$. Strong duality, which we shall prove next, states that for all $v > v_P$, we have that

$$Q_v \neq \emptyset.$$

This would imply that the dual is also bounded, and that in fact

$$Q_v = [v_Q, \infty)$$

with

$$v_Q = v_P = v_*.$$

**Remark.** *It is a priori not clear how to represent both the primal LP and the dual LP in the same picture. For example, the primal and the dual deal with spaces of different dimensions. In the geometric picture we shall draw next, we look only at the dual space. The columns of the matrix $A$ are interpreted as points in the dual space. The variable vector $x$ is thought of as defining a cone (i.e., a convex set which is also closed under products by positive numbers).*

To prove that, fix $v > v_P$. By definition,

$$\emptyset = P_v = \{x : Ax \leq b, x \geq 0, \langle -c, x\rangle \leq -v\}.$$

Let $A'$ be the matrix $A$ when we add to it the row $-c$, and let $b'$ the the vector $b$ when we add to it $-v$ at the end so that

$$P_v = \{x : A'x \leq b', x \geq 0\}.$$

Denote by $a'(1), \ldots, a'(n) \in \mathbb{R}^{k+1}$ the $n$ columns of $A'$. Consider the two convex sets

$$K = \mathsf{cone}(a'(1), \ldots, a'(n)) = \{A'x : x \geq 0\} \subseteq \mathbb{R}^{k+1}$$

and

$$B = \{y' \in \mathbb{R}^{k+1} : y' \leq b'\}.$$

It follows that

$$K \cap B = \emptyset.$$

Here we use convexity; two convex sets are disjoint iff there is a separating hyperplane. There is $\alpha' \in \mathbb{R}^{k+1}$ and $\beta \in \mathbb{R}$ so that

$$K \subseteq \{y' : \langle \alpha', y'\rangle \geq \beta\}$$

and

$$B \subseteq \{y' : \langle \alpha', y'\rangle \leq \beta\}.$$

**Remark.** *To geometrically see the reason behind the convex separation theorem, if $K, B$ are disjoint and convex, then look at the closest point $p$ in $K$ to $B$ and the closest point $q$ in $B$ to $K$, and consider the hyperplane that is orthogonal to $p - q$ and bisects the interval $[p, q]$ to two parts.*

**Remark.** *The convex separation theorem can also be proved by analyzing the perceptron algorithm.*

Because $K$ is a cone, we can choose $\beta = 0$. In other words, for all $x \geq 0$,

$$\langle \alpha', A'x\rangle \geq 0$$

and for all $y' \leq b'$,

$$\langle \alpha', y'\rangle \leq 0.$$

We claim that

$$\alpha' \geq 0,$$

because otherwise some $\alpha'_i < 0$ and so for for some $y' \leq b'$ we have $\langle \alpha', y'\rangle > 0$ which is false. In addition, $\alpha'_{k+1} \neq 0$, because otherwise $P_v$ is empty for all $v$, which is false. So, $\alpha'_{k+1} > 0$ and we can normalize $\alpha'$ so that

$$\alpha'_{k+1} = 1.$$

Writing $\alpha' = (\alpha, 1)$ we see that for all $x > 0$,

$$0 \leq \langle \alpha', A'x\rangle = \langle \alpha, Ax\rangle - \langle c, x\rangle = \langle A^T\alpha - c, x\rangle.$$

and
$$\langle \alpha, b \rangle \leq v.$$
Because the first inequality holds for all $x \geq 0$,
$$A^T \alpha \geq c.$$
It follows that
$$\alpha \in Q_v,$$
as needed.