Lecture 16: Introduction to Linear Programming

Lecturer: Shayan Oveis Gharan *11/29/21*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

16.1 Linear Programming

Linear systems are easiest class of optimization problems. Roughly speaking, any linear system of equations can be solved efficiently. The easiest examples are linear systems of equalities. This systems can be solved by matrix inversion. Given a (full rank) matrix A, to solve $Ax = b$, it is enough to compute $A^{-1}b$. In linear programming we study system of linear inequalities.

In the first few lectures we study linear systems of inequalities, also known as Linear Programming (LP). We also allow for a linear cost (or objective) function. LPs can be used to solve problems in a wide range of disciplines from engineering to nutrition and the stock market, At some point it was estimated that half of all computational tasks in the world correspond to solving linear programs.

A general LP can be formalized as follows:

$$\begin{aligned} &\min_x \langle c, x\rangle \\ &\text{s.t. } Ax \leq b, \end{aligned} \tag{16.1}$$

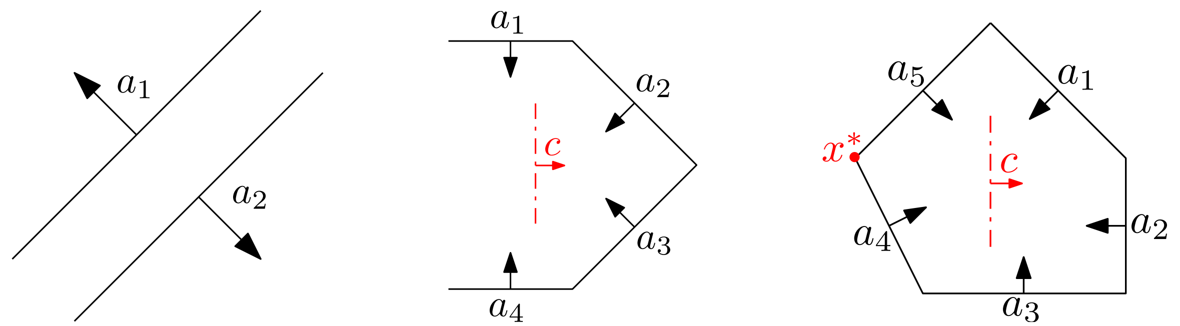
where $x \in \mathbb{R}^n$ is represents the variables, $c \in \mathbb{R}^n$ defines the objective function, and $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ define the constraints. The above form is fairly general; one can model various types of constraints in this form. For example, a constraint $\langle a_1, x\rangle \geq b_1$ can be written as $\langle -a_1, x\rangle \leq -b_1$. Or, a constraint $\langle a_1, x\rangle = b_1$ can be written as $\langle a_1, x\rangle \leq b_1$ and $\langle -a_1, x\rangle \leq -b_1$.

The objective function can be viewed as a hyperplane in $\mathbb{R}^n$ with normal vector c. Further, one can express the constraint matrix A as a series of row vectors:

$$A = \begin{bmatrix} a_1^{\mathrm{T}} \\ a_2^{\mathrm{T}} \\ \vdots \\ a_m^{\mathrm{T}} \end{bmatrix}.$$

When viewed from this perspective, it is easy to see the constraint set $a_i^{\mathrm{T}} x \leq b_i,\ i \in \{1, 2, \ldots, m\}$ as the intersection of finitely many half-spaces. The feasible set an LP is called a polyhedron. Thus, the solution will always be one of the following three cases shown in Figure 16.1.

There are many well known ways of solving LPs, but two of the most well known and used are Interior Point Methods (IPMs) and the simplex method. The simplex method is based on a very geometrically intuitive idea: start at a vertex of the domain of x and move to an adjacent vertex with lower objective - continue until an optimal solution is reached. Theoretically, the worst case time complexity of the simplex method is exponential; however, the simplex method is very fast in practice, and toolboxes such as CPLEX and Gurobi can solve very large LPs (millions of solution variables) in a few seconds. Interior point methods have better theoretical bounds; roughly speaking, using an interior point method, one can solve a general linear program with n variables by solving $\widetilde{O}(\sqrt{n})$ many systems of linear equalities.



(a) Infeasible domain of x: No solution.

(b) Unbounded domain of x: $x^* = \pm\infty$

(c) Closed, bounded and compact domain of x: at least one vertex of the domain of x is in the solution set x^*.

Figure 16.1: The possible domains of the solution variable x for an LP, along with the solution.

16.2 Convex Programming

Definition 16.1 (Convex Functions)**.** *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is* convex *on a set* $S \subseteq \mathbb{R}^n$ *if for any two points* $x, y \in S$*, we have*

$$f\left(\frac{x+y}{2}\right) \leq \frac{1}{2}(f(x) + f(y)).$$

We say f is *concave* if for any such $x, y \in S$, we have

$$f\left(f(\alpha x + (1-\alpha)y\right) \geq \alpha f(x) + (1-\alpha) f(y),$$

for any $0 \leq \alpha \leq 1$. There is an equivalent definition of convexity: For a function $f : \mathbb{R}^n \to \mathbb{R}$, the Hessian of f, $\nabla^2 f$ is a $n \times n$ matrix defined as follows:

$$(\nabla^2 f)_{i,j} = \partial_{x_i} \partial_{x_j} f$$

for all $1 \leq i, j \leq n$. We can show that f is convex over S if and only if for all $a \in S$,

$$\nabla^2 f \Big|_{x=a} \succeq 0.$$

For example, consider the function $f(x) = x^T A x$ for $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. Then, $\nabla^2 f = A$. So, f is convex (over $\mathbb{R}^n$) if and only if $A \succeq 0$.

For another example, let $f : \mathbb{R} \to \mathbb{R}$ be $f(x) = x^k$ for some integer $k \geq 2$. Then, $f''(x) = k(k-1)x^{k-2}$. If k is an even integer, $f''(x) \geq 0$ over all $x \in \mathbb{R}$, so f is convex over all real numbers. On the other hand, if k is an odd integer then $f''(x) \geq 0$ if and only if $x \geq 0$. So, in this f is convex only over non-negative reals.

Similarly, f is concave over S, if $\nabla^2 f \Big|_{x=a} \preceq 0$ for all $a \in S$. For example, $x \mapsto \log x$ is concave over all positive reals.

A norm $\|\cdot\|$ is defined as a function that maps $\mathbb{R}^n$ to $\mathbb{R}$ and satisfies the following three properties,

i) $\|x\| \geq 0$ for all $x \in \mathbb{R}^n$,

ii) $\|\alpha x\| = \alpha \|x\|$ for all $\alpha \geq 0$ and $x \in \mathbb{R}^n$,

iii) Triangle inequality: $\|x+y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$.

It is easy to see that any norm function is a convex function: This is because for any $x, y \in R^n$, and $0 \leq \alpha \leq 1$,

$$\|\alpha x + (1-\alpha)y\| \leq \|\alpha x\| + \|(1-\alpha)y\| = \alpha \|x\| + (1-\alpha) \|y\| .$$

Definition 16.2 (Convex Set)**.** *We say a set $S \subseteq \mathbb{R}^n$ is convex if for any pair of points $x, y \in S$, the line segment connecting x to y is in S.*

For example, let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function over a set $S \subseteq \mathbb{R}^n$. Let $t \in \mathbb{R}$, and define

$$T = \{x \in \mathbb{R}^n : f(x) \leq t\}.$$

Then, T is convex. This is because if $x, y \in T$, then for any $0 \leq \alpha \leq 1$,

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \leq \alpha t + (1-\alpha)t = t$$

where the first inequality follows by convexity of f. So, $\alpha x + (1-\alpha) \in T$ and T is convex.

We can generalize linear programming to convex programming as follows:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.,} & f_1(x) \leq b_1 \\ & \vdots \\ & f_m(x) \leq b_m \end{array}$$

where $f, f_1, \ldots, f_m$ are convex functions over the domain of x. Note that if f_i's are convex only over a convex set S of $\mathbb{R}^n$, we can limit x to be inside S.

Note that since each f_i is convex, the set of x satisfying $f_i(x) \leq b_i$ is a convex set, and the intersection of convex sets is also convex. Therefore the set of x feasible in the above program is a convex set. In general, we can do optimization over convex sets as long as they are well represented.

For a well-defined example, the program $\max x^T A x$ for a PSD matrix A returns the largest eigenvector and eigenvalue of A. Now, we may add arbitrary constraints. For example, we can say we want to find a vector x of largest quadratic form subject to $x_1 = 1$ and $x_2 = -3$.

As another side note, observe that for a convex function f, the set x where $f_i(x) \geq b_i$ is not necessarily a convex set. For a concrete example, if $f : \mathbb{R} \to \mathbb{R}$, which maps $x \mapsto |x|$, then $|x| \geq 1 = \{-\infty, 1\} \cup \{1, \infty\}$ is not even a "connected" set let alone being convex.

16.3 Semidefinite Programming

Suppose we have a convex set $S \subseteq \mathbb{R}^n$. We can represent S by infinitely many constraint as follows: For every point y in the boundary of S, let v_y be the unit vector orthogonal to the hyperplane that is tangent to S at y. Since S is a convex set all points in S are on one side of this hyperplane. Therefore, one of the following is true:

- For every point $x \in S$, $\langle x, v_y \rangle \geq \langle y, v_y \rangle$, or
- For every point $x \in S$, $\langle x, v_y \rangle \leq \langle y, v_y \rangle$.

So, we can represent S by by one such constraint for every point y in the boundary of S.

The above constraints defines a linear program with an infinitely many constraints. There is a well-known theorem which says we can do optimization over such a linear program as long as we have access to a *separation oracle*.

Theorem 16.3. *Suppose we have a linear program with infinitely many constraints. Suppose that for every given point x we can certify that either x is feasible and if x is not feasible we can efficiently output a constraint $\langle a, x\rangle \leq b$ for a vector a which is violated by x. Then we can find a point which $1 \pm \epsilon$ multiplicative factor of the optimum in time polynomial in the dimension n, and $\log(1/\epsilon)$.*

For example, suppose we have a symmetric matrix $X \in \mathbb{R}^{n\times n}$ of variables, and we want to enforce the constraint $X \succeq 0$, i.e., that X is a PSD matrix. We can represent this constraint with infinitely many *linear* constraints:

$$\sum_{i,j} a_i X_{i,j} a_j = a^T X a \geq 0, \ \forall a \in \mathbb{R}^n.$$

Now, let us discuss how to design a separation oracle for the above infinitely many constraints. Given a PSD matrix X we need to check if it feasible and if not we need to efficiently find a violated constraint. So, given a matrix X, i.e., an assignment to the underlying variables of X, we find the smallest eigenvector v and the corresponding eigenvalue λ of X. If $\lambda \geq 0$ we output that X is feasible; otherwise,

$$v^T X v < 0$$

is a violated constraint. This allows us to solve *semidefinite programs* by reducing them to linear programs with infinitely many constraints:

$$\begin{aligned} \max \quad & C \bullet X \\ \text{s.t.,} \quad & A_1 \bullet X \leq b_1 \\ & \vdots \\ & A_m \bullet X \leq b_m \\ & X \succeq 0. \end{aligned}$$

Note that any term $A_i \bullet X = \text{Tr}(A^T X)$ is a linear function in X so the above program is a bunch of linear inequalities in the underlying variables $X_{1,1}, X_{1,2} \ldots, X_{n,n}$ of X together with a PSD constraint $X \succeq 0$. This also can be solved efficiently in polynomial time.