

Constrained Optimization

Linear Programming - The Simplex Algorithm

Contents

1	Linear Programming	1
1.1	Standard form	1
1.2	Optimality and duality	2
1.2.1	Optimality	2
1.2.2	Duality	3
1.3	Geometry of the feasible set	4
1.3.1	Basic feasible points	4
1.3.2	Vertices of the feasible polytope	5
2	The simplex algorithm	5
2.1	Intuition	5
2.2	One-step	7
2.3	Practical details	7
2.3.1	Selection of the entering index	7
2.3.2	Starting the simplex method	7
2.4	When to use the simplex ?	8

1 Linear Programming

We'll consider here a rather easy optimization program - that is a linear objective function with linear constraints. Therefore, the feasible set is a *polytope*, and the contours of the objective function are planar. However, we might encounter some misspecified cases : the problem can be infeasible or unbounded.

1.1 Standard form

The standard linear program writes under the following form :

$$\begin{aligned}
 \min_x \quad & c^T x \\
 \text{s.t.} \quad & x \geq 0, \\
 & Ax = b
 \end{aligned} \tag{1}$$

where :

$$\begin{cases} c, x \in \mathbb{R}^n \\ A \in \mathcal{M}_{m,n}(\mathbb{R}) \\ b \in \mathbb{R}^m \end{cases}$$

Remark : This form is achievable for every linear program, thanks to the use of *slack variables*. For instance, let us consider the program :

$$\begin{aligned}
 \min_x \quad & c^T x \\
 \text{s.t.} \quad & Ax \leq b
 \end{aligned} \tag{2}$$

Let us introduce a slack variable $z = Ax - b \geq 0$. We'll also use the notations :

$$\begin{aligned}
 x^+ &= \max(x, 0) \\
 x^- &= \max(-x, 0)
 \end{aligned}$$

so that $x = x^+ - x^-$.

Therefore the problem (2) is equivalent to :

$$\begin{aligned} \min_x \quad & (c \quad c \quad 0) \begin{pmatrix} x^+ & x^- & z \end{pmatrix}^T \\ \text{s.t.} \quad & (A \quad -A \quad I) \begin{pmatrix} x^+ \\ x^- \\ z \end{pmatrix} = b, \\ & x^+, x^-, z \geq 0 \end{aligned} \tag{3}$$

which is the standard form.

By a similar reasoning, one can transform any linear programming problem into an equivalent one, under the standard form (1).

In all that follows, we'll assume that $m \leq n$, to avoid degenerate cases (infeasible, unique feasible point, etc).

1.2 Optimality and duality

1.2.1 Optimality

Optimality conditions for (1) can be obtained from the KKT. We'll define :

$$\begin{cases} \lambda \in \mathbb{R}^m & \text{the Lagrange multiplier for the equality constraint} \\ s \in \mathbb{R}^n & \text{the Lagrange multiplier for the inequality constraint} \end{cases}$$

Therefore the Lagrangian of (1) writes :

$$\mathcal{L}(x, \lambda, s) = c^T x - \lambda^T (Ax - b) - s^T x \tag{4}$$

and the first order necessary conditions writes that for an optimal feasible point x , $\exists(\lambda, s)$ such that :

$$c = A^T \lambda + s \tag{5a}$$

$$Ax = b \tag{5b}$$

$$x \geq 0 \tag{5c}$$

$$s \geq 0 \tag{5d}$$

$$x^T s = 0 \tag{5e}$$

Now let (x^*, λ^*, s^*) a triple that satisfy such equations. Therefore we have that :

$$\begin{aligned} c^T x^* &= \lambda^{*T} A x^* + s^{*T} x^* \\ &= b^T \lambda^* \end{aligned}$$

We'll see later that $\lambda \rightarrow \lambda^T b$ is actually the objective of the dual problem. More importantly, we can actually show the following theorem.

Sufficient optimality conditions

The KKT first-order necessary conditions (5) are sufficient optimality conditions.

PROOF : Let \bar{x} be a feasible point, hence such that

$$A\bar{x} = b$$

$$\bar{x} \geq 0$$

Therefore

$$\begin{aligned} c^T \bar{x} &= (\lambda^T A + s^T) \bar{x} \\ &= \lambda^{*T} b + s^{*T} \bar{x} \\ &\geq c^T x^* \end{aligned}$$

with equality only if the complementary condition (5e) holds. This therefore means that no feasible point has a lower objective than x^* . The necessary KKT conditions are therefore sufficient conditions.

1.2.2 Duality

Let us define the dual problem :

$$\begin{aligned} \max_{\lambda} \quad & b^T \lambda \\ \text{s.t.} \quad & A^T \lambda \leq c \end{aligned} \tag{6}$$

for which the corresponding Lagrangian writes :

$$\bar{\mathcal{L}}(\lambda, x) = b^T \lambda - x^T (A^T \lambda - c) \tag{7}$$

The corresponding KKT equations are :

$$Ax = b \tag{8a}$$

$$A^T \lambda \leq c \tag{8b}$$

$$x \geq 0 \tag{8c}$$

$$x^T (c - A^T \lambda) = 0 \tag{8d}$$

which are exactly the same conditions (5) as for the primal problem ! Also, as we did for the primal problem, we can show that those are also sufficient optimality conditions.

The *weak duality* results tells us that for a given feasible vector x for the primal problem, and a feasible vector λ for the dual, we have an inequality control over the respective objective functions :

$$c^T x \geq b^T \lambda \tag{9}$$

making the dual objective function a lower bound for the primal objective :

$$\min_{x \in \Omega_1} c^T x \geq \max_{\lambda \in \Omega_2} b^T \lambda \tag{10}$$

The equality between the two objective function we derived lead us to formulate the *strong duality* theorem.

Strong Duality

- If either (1) or (6) has a finite solution, then so does the other, and the objective values at the respective solutions are equal.
- If either (1) or (6) is unbounded, then the other is unfeasible.

PROOF :

- Since both KKT conditions are equivalent and sufficient conditions, we have the equivalence between the existence of a solution for one problem and the other.
- Let us suppose that the primal problem is unbounded : therefore $\exists \{x_k\}_{k \in \mathbb{N}}$ so that

$$\begin{aligned} c^T x_k &\rightarrow -\infty \\ Ax_k &= b, & \forall k \in \mathbb{N} \\ x_k &\geq 0, & \forall k \in \mathbb{N} \end{aligned}$$

Now, if the dual is feasible we can find $\bar{\lambda}$ such that $A^T \bar{\lambda} \leq c$ and therefore $\forall k \in \mathbb{N}$:

$$\begin{aligned} \bar{\lambda}^T (Ax_k) &\leq c^T x_k \text{ so} \\ b^T \bar{\lambda} &\leq c^T x_k \end{aligned}$$

This inequality holding for all k , we therefore have that

$$b^T \bar{\lambda} < -\infty$$

which is impossible. The dual must therefore be unfeasible ! A similar reasoning can be applied for the other side of the equivalence.

1.3 Geometry of the feasible set

1.3.1 Basic feasible points

We assume in all that follows that A has **full row-rank**, and introduce the notion of basic feasible points.

Basic feasible point

A vector x is said to be a basic feasible point of (1) if it is feasible and if there exists a subset \mathcal{B} of the index set $\{1, \dots, n\}$ so that :

- \mathcal{B} contains exactly m indices.
- $i \notin \mathcal{B} \Rightarrow x_i = 0$
- The $m \times m$ matrix B defined as a collection of A 's columns :

$$B = [A_i]_{i \in \mathcal{B}}$$

is non-singular.

\mathcal{B} is called the basis of the primal problem.

As we'll see, the simplex's strategy is to examine only such points, which leads to the optimum if and only if such points exists and if the optimal is a basic feasible point. Such an assumption is true - under certain conditions - as the following theorem indicates.

Fundamental theorem of linear programming

- (i) If (1) has a non-empty feasible region, then there is at least one basic feasible point.
- (ii) If (1) has solutions, then at least one of them is a basic optimal point.
- (iii) If (1) is feasible and bounded, it has an optimal solution.

PROOF :

(i) : Among all feasible vector x , let us choose the one with the minimal number of non-zero components. Without loss of generality, we'll note that number p and write down :

$$x = (x_1 \quad x_2 \quad \dots \quad x_p \quad 0 \quad \dots \quad 0)^T \quad (11)$$

Hence we'll have that :

$$\sum_{i=1}^p A_i x_i = b \quad (12)$$

If we assume that (A_1, \dots, A_p) is linearly dependent, therefore we'll have that there exists $(z_i)_{i \in \{1, \dots, p-1\}}$ such that :

$$A_p = \sum_{i=1}^{p-1} z_i A_i \quad (13)$$

If we now introduce, for some $\varepsilon > 0$:

$$x(\varepsilon) = x + \varepsilon (z_1 \quad \dots \quad z_{p-1} \quad -1 \quad 0 \quad \dots \quad 0)^T \quad (14)$$

we have thanks to (12) and (13) that $x(\varepsilon)$ is a feasible point :

$$\begin{aligned} Ax(\varepsilon) &= b \quad \forall \varepsilon \in \mathbb{R} \\ x(\varepsilon) &\geq 0 \quad \text{for } \varepsilon \text{ small enough} \end{aligned}$$

From the definition of $x(\varepsilon)$ we know that $\exists \bar{\varepsilon}$ such that for it exists $i \in \{1, \dots, p\}$ so that

$$x_i(\bar{\varepsilon}) = 0$$

Or the vector $x(\bar{\varepsilon})$ is feasible and as therefore a maximum of $(p - 1)$ non-zero components, which contradicts our initial hypothesis. Therefore we know that (A_1, \dots, A_p) **is linearly independent**. This said, we have that $p \leq m$. If $p = m$ the vector x is a feasible point. On the other hand, if $p < m$, because A has full row rank we know that we can choose $m - p$ independent column from A_{p+1}, \dots, A_n and add their indexes to \mathcal{B} . This provides the existence of a basic feasible point.

1.3.2 Vertices of the feasible polytope

The feasible set defined by the linear constraints is a polytope, which is fully defined by its set of vertex (it aligns with its convex hull)

$$\mathcal{V} = \{v_1, \dots, v_n\}$$

The $\{v_i\}$'s are visually recognizable. More importantly, algebraically speaking, they are the basic points that we were discussing earlier !

Theorem

All basic feasible points of

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0 \end{aligned} \tag{15}$$

are vertices of the feasible polytope

$$\{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \tag{16}$$

and vice-versa.

PROOF : Let x be a basic feasible point. We'll note $\mathcal{B} = \{1, \dots, m\}$ its basis without loss of generalization. Since

$$x_{m+1} = \dots = x_n = 0$$

if we assume that $\exists \alpha > 0$ and y, z two feasible vectors such that

$$x = \alpha y + (1 - \alpha)z$$

we will have that :

$$y_{m+1} = \dots = y_n = 0$$

$$z_{m+1} = \dots = z_n = 0$$

Therefore we have that $Ax = Ay = Az = b$ and that by noting, for any vector x , $x_{\mathcal{B}} = (x_1, \dots, x_m)$:

$$Bx_{\mathcal{B}} = By_{\mathcal{B}} = Bz_{\mathcal{B}}$$

However, B being non-singular, this implies that $x = y = z$. Therefore, we have that x is a vertex of the feasible polytope. The second implication relies on a similar reasoning that the one hold during the last proof.

2 The simplex algorithm

There exists a great deal of variant method to the simplex algorithm. In what follows, we present the *revised simplex method*.

2.1 Intuition

As said earlier, all iterates of the simplex are basic feasible points of (1), and therefore vertices of the feasible polytope. Most of the step of the simplex algorithm consist in moving from one vertex to another - an adjacent vertex, for which the basis differs in exactly one component.

The major issue in this approach is to know which of the index to remove from \mathcal{B} . Let us define \mathcal{N} the complementary set of \mathcal{B} and N equivalently to B :

$$N = [A_i]_{i \in \mathcal{N}}$$

Let us also partition x and s according to those sets- for instance :

$$x_{\mathcal{B}} = (x_i)_{i \in \mathcal{B}} \quad x_{\mathcal{N}} = (x_i)_{i \in \mathcal{N}}$$

From the KKT conditions, we have that

$$Ax = Bx_{\mathcal{B}} + Nx_{\mathcal{N}} = b \quad (17)$$

where and hence :

$$\begin{cases} x_{\mathcal{B}} = B^{-1}b \\ x_{\mathcal{N}} = 0 \end{cases} \quad (18)$$

We can now choose $s_{\mathcal{B}}$ to comply with the complementary conditions by setting :

$$s_{\mathcal{B}} = 0 \quad (19)$$

We find c and $s_{\mathcal{N}}$ by partitioning the first KKT conditions :

$$\begin{cases} B^T \lambda = c_{\mathcal{B}} \\ N^T \lambda + s_{\mathcal{N}} = c_{\mathcal{N}} \end{cases} \quad (20)$$

which eventually leads to :

$$s_{\mathcal{N}} = c_{\mathcal{N}} - (B^{-1}N)^T c_{\mathcal{B}} \quad (21)$$

In the previous computations, we never used the positivity of the inequality Lagrange multiplier. It is obvious that this condition is met if and only if :

$$s_{\mathcal{N}} \geq 0$$

If it is indeed the case, the algorithm can terminate since we discovered an optimal triplet (x, λ, s) . If $\exists q \in \mathcal{N}$ such that

$$s_q < 0$$

q is set to be the *entering index* (and is a candidate for entering \mathcal{B}).

The procedure for altering \mathcal{B} and changing x and s can be described as follows :

Pivoting

- Allow x_q to increase from 0 during next step
- Fix all other component of $x_{\mathcal{N}}$ to 0 and figure out the effect of increasing x_q on the current basis vector $x_{\mathcal{B}}$, given that we wish to stay feasible with respect to the equality constraint $Ax = b$.
- Keep increasing x_q until one of the component of x_p of \mathcal{B} is driven to 0.
- Remove p from \mathcal{B} and replace it with q .

Let us formalize this process (known as *pivoting*). Since both the iterate x^+ and the current x should verify $Ax = b$:

$$Bx_{\mathcal{B}}^+ + A_q x_q^+ = Bx_{\mathcal{B}}$$

and therefore

$$x_{\mathcal{B}}^+ = x_{\mathcal{B}} - B^{-1}A_q x_q^+$$

The effect on the objective function is so that :

$$\begin{aligned} c^T x^+ &= c_{\mathcal{B}}^T x_{\mathcal{B}}^+ + c_q x_q^+ \\ &= c_{\mathcal{B}}^T x_{\mathcal{B}} - c_{\mathcal{B}}^T B^{-1} A_q x_q^+ + c_q x_q^+ \end{aligned}$$

Also, since $c_{\mathcal{B}}^T B^{-1} = \lambda^T$ and since $q \in \mathcal{N}$, we have that $A_q^T \lambda = c_q - s_q$. Finally :

$$c^T x^+ = c^T x + s_q x_q^+ \quad (22)$$

Given that $s_q < 0$ (by definition), the **objective function is therefore decreased** after such an iteration.

2.2 One-step

The summary of one step of the algorithm is detailed below.

Simplex - One-Step

- Given \mathcal{B} , \mathcal{N} , $x_{\mathcal{B}} = B^{-1}b$, $x_{\mathcal{N}} = 0$:
 - Solve $B^T \lambda = c_{\mathcal{B}}$ for λ
 - Compute $s_{\mathcal{N}} = c_{\mathcal{N}} - N^T \lambda$
- If $s_{\mathcal{N}} \geq 0$ **stop**
- Select $q \in \mathcal{N}$ with $s_q < 0$ (entering index)
- Solve $Bd = A_q$ for d
- If $d \leq 0$ **stop** (unbounded problem)
- Compute

$$x_q^+ = \min_{i \mid d_i > 0} \frac{(x_{\mathcal{B}})_i}{d_i} \quad (23)$$

with i the arg-minimum.

- Update $x_{\mathcal{B}}^+ = x_{\mathcal{B}} - dx_q +$
- Change \mathcal{B} by adding the index q and removing p

2.3 Practical details

2.3.1 Selection of the entering index

There are usually several negative components of $s_{\mathcal{N}}$ at each step. Question is : among them, how to choose the entering index, with the goal to minimize the number of steps to get to the optimum x^* . We therefore need to figure out a tradeoff between the search for that component and the actual decrease of the objective function.

Dantzig's original rule chooses q such that s_q is the most negative component of $s_{\mathcal{N}}$. This is clearly motivated by (22) - however a large reduction of $c^T x$ is not guaranteed.

Many different policies exist - we might want to evaluate the contribution of several indexes - at the cost of inverting B which can be high-dimensional. Such a policy is known as a *multiple pricing* strategy.

2.3.2 Starting the simplex method

The simplex requires a basic feasible point x to start with - as well as its corresponding initial basis \mathcal{B} - such that B is non-singular, and $x_{\mathcal{B}} = B^{-1}b \geq 0$. Finding such a point is non-trivial, and we describe hereinafter a two-phase approach for finding this point.

Phase 1 : Set-up an auxiliary linear program based on the data and solve it using the simplex method. Design it so that finding an initial basic feasible point is easy and so that its optimal point is a basic feasible point for the phase 2 program.

Phase 2 : Define a second linear problem, close to the original one, so that its solution can be used to extract one for (1).

In phase 1, we can introduce an artificial variable z and aim to solve :

$$\begin{aligned} \min_z \quad & e^T z \\ \text{s.t.} \quad & Ax + Ez = b, \\ & x, z \geq 0 \end{aligned} \tag{24}$$

where $e = (1 \dots 1)^T$, and $E = \text{diag}(\text{sign}(b_j))$. Then we immediately have that

$$\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ |b_j|_j \end{pmatrix}$$

is a basic feasible point for (24), with initial basis matrix given by E (which obviously is non-singular). In this program, z represent the degree of which the constraint $Ax = b$ is violated by the x component. If the optimal solution of (24) is superior to 0, then (1) is unfeasible.

Else, the optimal point provide a feasible point for a second linear program, which matches (1) by a mock variable z - we can therefore the optimal solution of (1) extremely easily.

2.4 When to use the simplex ?

The simplex method belongs to a general class of algorithms for constrained optimization known as the **active set method**. It explicitly maintains estimates of the active and inactive index sets.

Though highly efficient on almost all practical problem, the simplex has **exponential complexity**. The search for polynomial complexity recently led to intense research in the field of *interior-point* methods.