MIT 6.7220/15.084 — Nonlinear Optimization (Spring '25) Tue, Feb 11th 2025

Lecture 3
More on normal cones, and a first taste of duality

Instructor: Prof. Gabriele Farina (✉ gfarina@mit.edu)[*]

In this lecture, we continue our investigation of normal cones to points in feasible sets that occur frequently. In Lecture 2, we have considered the normal cone to a hyperplane, and we have seen that the normal cone at a point x in a hyperplane is the set of all vectors that are orthogonal to the hyperplane. We now consider the normal cone at a point in a *halfspace*.

An interesting consequence of the characterization of normal cones at the intersection of halfspaces will be to allow us to derive linear programming duality—one of the "crown jewels" of linear optimization—as a direct corollary.

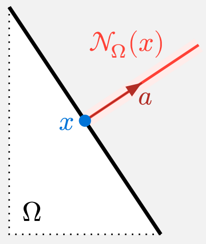
L3.1 Normal cone at a point in a halfspace

Let's start from considering a point in a single halfspace.

Example L3.1 (Normal cone to a halfspace)**.** Consider a halfspace

$$\Omega := \{y \in \mathbb{R}^n : \langle a, y\rangle \le b\}, \quad \text{where } a \in \mathbb{R}^n, a \neq 0$$

and a point $x \in \Omega$.



We already know that if x is in the interior of Ω (*i.e.*, $\langle a, x\rangle < b$), then $\mathcal{N}_\Omega(x) = \{0\}$. On the other hand, if x is on the boundary of Ω, that is, $\langle a, x\rangle = b$, then it is pretty intuitive from the picture that

$$\mathcal{N}_\Omega(x) = \{\lambda \cdot a : \lambda \in \mathbb{R}_{\ge 0}\}.$$

The crucial difference between the above result and that of Theorem L2.2 (Normal cone to a hyperplane), is that the normal cone to a *halfspace* only points in one direction ($\lambda \ge 0$), as opposed to the case of the hyperplane where $\lambda \in \mathbb{R}$. The formal proof can be obtained by adapting the arguments we used in Theorem L2.2. [▷ You should try to work out the details!]

L3.2 Normal cone at the intersection of halfspaces

[*]These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

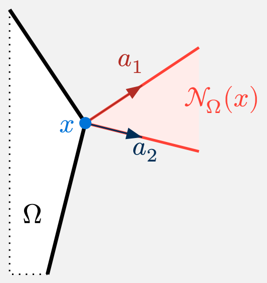
We now consider the case of the *intersection* of two halfspaces.

Example L3.2 (Normal cone to the intersection of two halfspaces)**.** Consider the intersection Ω of two halfspaces:

$$\Omega := \left\{ y \in \mathbb{R}^n : \begin{matrix} \langle a_1, y\rangle \le b_1 \\ \langle a_2, y\rangle \le b_2 \end{matrix} \right\}, \quad \text{where } a_1, a_2 \in \mathbb{R}^n, a_1, a_2 \neq 0,$$

and a point $x \in \Omega$.

If x is in the interior of Ω, or if x is on the boundary of one of the halfspaces but not both, then the normal cone follows from our prior results. So, consider x at the intersection of the boundaries of the halfspaces, that is, $\langle a_1, x\rangle = b_1, \langle a_2, x\rangle = b_2$.



It is pretty intuitive from the picture that the normal cone at x is the *conic hull* of a_1 and a_2, that is, the set obtained from summing the directions in all possible ways:

$$\mathcal{N}_\Omega(x) = \{\lambda_1 \cdot a_1 + \lambda_2 \cdot a_2 : \lambda_1, \lambda_2 \in \mathbb{R}_{\ge 0}\}.$$

Formally proving the above result requires a bit of simple machinery that we have not seen yet; it will be the topic of Lecture 5. For now, we take the result as a given. An important takeaway to remember is that, at least in the case of halfspaces, *the normal cone is obtained by* summing *the normal cones given by all the constraints that are active at the point.*

The generalization of the result to the case of m halfspaces is equally intuitive, and we present it next.

Theorem L3.1. Let $\Omega \subseteq \mathbb{R}^n$ be given as the intersection of m halfspaces $\langle a_j, x\rangle \le b_j$. Then, the normal cone at any point $x \in \Omega$ is obtained by taking nonnegative combinations of all those a_j's for which $\langle a_j, x\rangle = b_j$. In symbols:

$$\mathcal{N}_\Omega(x) = \left\{ \sum_{j \in I(x)} \lambda_j \cdot a_j : \lambda_j \in \mathbb{R}_{\ge 0} \right\}, \qquad \text{where } I(x) = \{j \in \{1, ..., m\} : \langle a_j, x\rangle = b_j\}.$$

The constraints j in $I(x)$ are often called the "active constraints" at $x \in \Omega$.

L3.2.1 A remark on equality constraints

Note that the above characterization of normal cones immediately recovers what we saw in Lecture 2 for hyperplanes and affine subspaces. Indeed, the constraint

$$\langle a, x\rangle = 0$$

is equivalent to the intersection of the halfspaces

$$\langle a, x\rangle \leq 0 \quad \wedge \quad \langle -a, x\rangle \leq 0.$$

No matter the point x considered, the two inequality constraints must be active at the same time. So, by Theorem L3.1, the normal cone is given by the set

$$\begin{aligned}
\mathcal{N}_\Omega(x) &= \{\lambda_1 a + \lambda_2(-a) : \lambda_1, \lambda_2 \geq 0\} \\
&= \{(\lambda_1 - \lambda_2)a : \lambda_1, \lambda_2 \geq 0\} \\
&= \{\lambda a : \lambda \in \mathbb{R}\} \\
&= \operatorname{span}(a).
\end{aligned}$$

Example L3.3. As an example application, let's write the normal cone to a point in the probability simplex

$$\Delta^n := \{(x_1, ..., x_n) : x_1 + \cdots + x_n = 1,\ x_1 \geq 0,\ ...,\ x_n \geq 0\}.$$

The probability simplex is the intersection of the hyperplane $1^\top x = 1$, and the halfspaces $-e_i^\top x \leq 0$, for all $i = 1, ..., n$, where e_i is the i-th unit vector. Hence, by Theorem L3.1, the normal cone at any point $x \in \Delta^n$ is given by

$$\mathcal{N}_{\Delta^n}(x) = \left\{\alpha 1 - \sum_{i : x_i = 0} \lambda_i e_i : \alpha \in \mathbb{R}, \lambda_i \geq 0\right\}.$$

L3.2.2 Complementary slackness reformulation

The previous result can be reformulated in a more compact way by using the concept of *complementary slackness*, as we detail in the next theorem. This reformulation is a simple algebraic trick, and it does not really bring anything fundamentally new to the table. However, the final form after the reformulation is very pleasantly concise, and often leads to nice manipulations.

The key idea is that, in the definition of $\mathcal{N}_\Omega(x)$ in Theorem L3.1, instead of summing over $j \in I(x)$, one could equivalently sum over all $j = 1, ..., m$, and then impose the constraint that $\lambda_j = 0$ for all $j \notin I(x)$:

$$\mathcal{N}_\Omega(x) = \left\{\sum_{j=1}^m \lambda_j \cdot a_j : \lambda_j \geq 0, \lambda_j = 0 \text{ if } \langle a_j, x\rangle < b_j\right\}.$$

Because $\lambda_j \geq 0$, the condition that $\lambda_j = 0$ whenever $\langle a_j, x\rangle < b_j$ can be succintly rewritten as

$$\sum_{j=1}^m \lambda_j (b_j - \langle a_j, x\rangle) = 0.$$

The above condition is usually called "complementary slackness".

The reason why the above reformulation is often preferred, is that it lends especially well to being written in matrix form. In particular, we have the following.

Remark L3.1. Given the intersection of halfspaces expressed in vectorial form as

$$Ax \leq b,$$

the normal cone at any feasible point x given in Theorem L3.1 can be expressed equivalently as

$$\mathcal{N}_\Omega(x) = \{A^\top \lambda : \lambda \geq 0, \lambda^\top(b - Ax) = 0\}.$$

There is nothing "deep" regarding this rewriting. The important point is that only the constraints that are active at x participate in defining the normal cone. Using the complementary slackness formulation, or the form in Theorem L3.1 is just a matter of convenience in how to formalize what it means for a constraint to be active.

L3.3 Application #1: Derivation of linear programming duality

The previous characterization of normal cones at the intersection of halfspaces is natural, but hides quite the punch. In particular, we show how linear programming duality follows immediately as a corollary.

Consider the linear program

$$\begin{aligned} \max_x \quad & f(x) \coloneqq c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{R}^n, \end{aligned} \tag{P}$$

where the matrix A is assumed to have m rows. We will show that the first-order necessary optimality conditions imply that in order for x to be an optimal solution to (P), then the following "dual" problem also has an optimal solution, and the value matches that of (P):

$$\begin{aligned} \min_\lambda \quad & g(\lambda) \coloneqq b^\top \lambda \\ \text{s.t.} \quad & A^\top \lambda = c \\ & \lambda \geq 0 \end{aligned} \tag{D}$$

From the first-order necessary optimality conditions for (P), we know that any optimal x^* must satisfy the condition (note that the problem is a maximization rather then minimization)

$$\nabla f(x^*) \in \mathcal{N}_\Omega(x), \qquad \text{where } \Omega \coloneqq \{x \in \mathbb{R}^n : Ax \leq b\}. \tag{3}$$

From Theorem L3.1 and Remark L3.1, the normal cone $\mathcal{N}_\Omega(x)$ is a combination of the rows of A, that is,

$$\mathcal{N}_\Omega(x^*) = \{A^\top \lambda : \lambda^\top(b - Ax^*) = 0, \lambda \geq 0\}, \tag{4}$$

where the condition $\lambda^\top(b - Ax^*) = 0$ is the complementary slackness condition mentioned in Remark L3.1. Plugging (4) into (3), together with the fact that $\nabla f(x^*) = c$, we obtain that at optimality there exist $\lambda^* \in \mathbb{R}^n$ such that

$$c = A^\top \lambda^*, \quad (\lambda^*)^\top (b - Ax^*) = 0, \quad \lambda^* \geq 0.$$

Plugging in the fact that $A^\top \lambda^* = c$ in the middle equality, yields that

$$(\lambda^*)^\top b = c^\top x^*.$$

Hence, from the mere existence of an optimal point $x^* \in \Omega$ for (P), we deduced that there exists a point λ^* in the feasible set of (D) that achieves objective value $g(\lambda^*) = b^\top \lambda^* = c^\top x^* = f(x^*)$. To conclude that λ^* is optimal for (D), it just suffices to show that all other feasible points for (D) must achieve value $\geq c^\top x^*$. This is pretty straightforward. All feasible vectors λ for (D) satisfy $\lambda \geq 0$ and $A^\top \lambda = c$. Furthermore, $b \geq Ax^*$ since $x^* \in \Omega$. Hence, for any feasible λ of (D),

$$\begin{aligned} g(\lambda) &= b^\top \lambda \\ &\geq (Ax^*)^\top \lambda \\ &= (x^*)^\top A^\top \lambda \\ &= (x^*)^\top c = f(x^*). \end{aligned}$$

Since $g(\lambda) \geq f(x^*)$ for all feasible λ in (D), and we know that a feasible λ^* that achieves $g(\lambda^*) = f(x^*)$ exists, it follows that λ^* must be optimal for (D).

Thus, with just a simple application of corollary of a result that felt simple, we have shown one of the crown jewels of linear optimization.

Theorem L3.2 (Strong linear programming duality)**.** If (P) admits an optimal solution x^*, then (D) admits an optimal solution λ^*, such that:

- the values of the two problems coincide: $c^\top x^* = b^\top \lambda^*$; and
- λ^* satisfies the complementary slackness condition $(\lambda^*)^\top (b - Ax^*) = 0$.

L3.4 Application #2: Projection onto a probability simplex

As a second application, we consider the problem of projecting a vector onto the probability simplex Δ^n, that is, the space of distribution over n actions

$$\Delta^n := \{(x_1, ..., x_n) \in \mathbb{R}^n_{\geq 0} : x_1 + ... + x_n = 1\}.$$

This problem arises in machine learning, where one often wants to figure out an optimal distribution over actions by repeatedly adjusting the distribution and projecting back onto the simplex. More formally, given a point $y \in \mathbb{R}^n$ to project, we are interested in computing the solution to

$$\begin{aligned} \min_x \quad & \tfrac{1}{2}\|x - y\|_2^2 \\ \text{s.t.} \quad & x \in \Delta^n. \end{aligned}$$

Incidentally, since the set is closed, we already know from Lecture 1 that the problem *has* a minimizer. As it turns out, unlike the applications we considered in Lecture 2, this problem *does not admit a closed-form solution.* However, the first-order optimality conditions will nonetheless point the way to a fast *algorithm* for computing the projection up to any desired approximation. In particular, we will see that an ϵ-approximate projection can be computed in $O\big(n \log \frac{1}{\epsilon}\big)$ time, and—with a little extra work—an exact solution can be computed in $O(n \log n)$ time.

From Example L3.3, we know that the normal cone at any point $x \in \Delta^n$ is given by

$$\mathcal{N}_{\Delta^n}(x) = \left\{ \alpha \mathbb{1} - \sum_{i : x_i = 0} \lambda_i e_i : \alpha \in \mathbb{R}, \lambda_i \geq 0 \right\}.$$

Hence, the first-order optimality conditions for the projection problem, $-(x - y) \in \mathcal{N}_{\Delta^n}(x)$, imply that at optimality there must exist multipliers $\alpha \in \mathbb{R}, \lambda_i \geq 0$ such that

$$x = y - \alpha \mathbb{1} + \sum_{i : x_i = 0} \lambda_i e_i.$$

Rewritten into complementary slackness form, this is the same as

$$x = y - \alpha \mathbb{1} + \lambda, \qquad \text{where} \quad \lambda_i x_i = 0 \text{ for all } i = 1, ..., n.$$

Looking at the vector equality along the generic row i, we have that

$$x_i = y_i - \alpha + \lambda_i, \qquad \lambda_i x_i = 0.$$

▸ **Removing the multipliers $\boldsymbol{\lambda_i}$.** As it turns out, we can solve for the multipliers λ_i in the previous equality and substitute them directly:

- Case I: $y_i - \alpha < 0$. Since $x_i \geq 0$, the only possibility is that $\lambda_i > 0$. But then, $x_i = 0$, which leads to $\lambda_i = \alpha - y_i \geq 0$.
- Case II: $y_i - \alpha \geq 0$. If we were to pick any $\lambda_i > 0$, this would result in $x_i > 0$, which would then imply $\lambda_i = 0$ by complementary slackness. So, the only possibility is that $\lambda_i = 0$. But then, $x_i = y_i - \alpha$.

We see that no matter the value of α, the only solution for λ_i results in the relationship

$$x_i = [y_i - \alpha]^+,$$

where $[z]^+ := \max\{z, 0\}$. This is not exactly obvious: basically, for a point x to aspire to be a projection of y onto the simplex, it better be the case that a value $\alpha \in \mathbb{R}$ exists, such that $x_i = [y_i - \alpha]^+$ for all $i = 1, ..., n$.

▸ **Solving for $\boldsymbol{\alpha}$.** How do we solve for α? The value of α is such that the sum of the x_i's is 1. Hence, we must have

$$\sum_{i=1}^{n} [y_i - \alpha]^+ = 1.$$

For $\alpha = \max\{y_i\}$, the left-hand side is 0, and reducing alpha results in the sum strictly increasing monotonically. Furthermore, for $\alpha = \min\{y_i\} - 1$, the function is ≥ 1. Hence, there exists a unique value of α that satisfies the equation. However, we do not have a closed-form expression for this value. Nonetheless, we can use a binary search in the range

$$\left[\min_i\{y_i\} - 1, \ \max_i\{y_i\}\right]$$

to compute the value of α to any desired precision ϵ in $O\big(n \log \frac{1}{\epsilon}\big)$ time.

Another approach would be to sort the y_i, and then compute the value of α by scanning the sorted list. This would take $O(n \log n)$ time.

Changelog

- Feb 11, 2025: Cosmetic changes.
- Feb 11, 2025: Typo fixes (thanks Jonathan Huang!)