Chapter 5, Lecture 8: Deriving Dual Constraints

March 29, 2019 *University of Illinois at Urbana-Champaign*

1 Linear programming

Consider the following linear program as an example:

$$(P) \qquad \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} & \mathbf{c}\cdot\mathbf{x} \\ \text{subject to} & A\mathbf{x} \ge \mathbf{b}. \end{cases}$$

where A is an $m \times n$ matrix, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$. (In standard form, the constraint looks like $\mathbf{b} - A\mathbf{x} \le \mathbf{0}$.)

Here, the dual objective function is $h(\boldsymbol{\lambda}) = \inf\{\mathbf{c}\cdot\mathbf{x} + \boldsymbol{\lambda}\cdot(\mathbf{b} - A\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$. We can rewrite the Lagrangian as

$$\mathbf{c}\cdot\mathbf{x} + \boldsymbol{\lambda}\cdot(\mathbf{b} - A\mathbf{x}) = \left(\mathbf{c} - A^\mathsf{T}\boldsymbol{\lambda}\right)\cdot\mathbf{x} + \boldsymbol{\lambda}\cdot\mathbf{b}.$$

We're minimizing a linear function of $\mathbf{x}$ to determine $h(\boldsymbol{\lambda})$, and this pretty much always results in $-\infty$. If $\mathbf{c} - A^\mathsf{T}\boldsymbol{\lambda}$ has any negative component, just make the corresponding component of $\mathbf{x}$ be arbitrarily large. If $\mathbf{c} - A^\mathsf{T}\boldsymbol{\lambda}$ has any positive component, just make the corresponding component of $\mathbf{x}$ be arbitrarily negative. It is only when $\mathbf{c} - A^\mathsf{T}\boldsymbol{\lambda} = \mathbf{0}$ that we get a value for the dual: in this case, the contribution from $\mathbf{x}$ cancels, and $h(\boldsymbol{\lambda}) = \boldsymbol{\lambda}\cdot\mathbf{b}$.

We can write the dual as

$$(D) \qquad \begin{cases} \underset{\boldsymbol{\lambda}\in\mathbb{R}^m}{\text{maximize}} & h(\boldsymbol{\lambda}) = \begin{cases} \boldsymbol{\lambda}\cdot\mathbf{b} & \text{if } A^\mathsf{T}\boldsymbol{\lambda} = \mathbf{c}, \\ -\infty & \text{otherwise} \end{cases} \\ \text{subject to} & \boldsymbol{\lambda} \ge \mathbf{0}. \end{cases}$$

But this is silly: if we're at all capable of solving $A^\mathsf{T}\boldsymbol{\lambda} = \mathbf{c}$ for $\boldsymbol{\lambda}$, we will never want to pick the $-\infty$ option. In such cases, we think of the condition $A^\mathsf{T}\boldsymbol{\lambda} = \mathbf{c}$ as an additional constraint in the dual program, and write the dual as

$$(D) \qquad \begin{cases} \underset{\boldsymbol{\lambda}\in\mathbb{R}^m}{\text{maximize}} & \boldsymbol{\lambda}\cdot\mathbf{b} \\ \text{subject to} & A^\mathsf{T}\boldsymbol{\lambda} = \mathbf{c}, \\ & \boldsymbol{\lambda} \ge \mathbf{0}. \end{cases}$$

This is the linear programming dual, which sees approximately infinitely many applications if you take a class on linear programming—but for us, the important thing is the general principle at work. **When we derive necessary conditions to have $h(\boldsymbol{\lambda}) > -\infty$, we turn them into dual constraints.**

[1]This document comes from the Math 484 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/484-spring-2019.html

2 Constrained geometric programming

Recall that a *posynomial term* in variables $t_1, \dots, t_m$ is a product $Ct_1^{\alpha_1} \cdots t_m^{\alpha_m}$, where $\alpha_1, \dots, \alpha_m$ are arbitrarily real numbers and $C > 0$. A posynomial is a sum of one or more posynomial terms.

Earlier in this course, we learned to solve unconstrained geometric programs: to minimized an arbitrary posynomial, subject to all the variables remaining positive.

Now, we can also look at constrained geometric programs. The constraints we allow will have the form $P(t_1, t_2, \dots, t_m) \le 1$, where P can be any posynomial. (Later on, we will discuss slightly more general constraints.)

The general procedure for deriving the dual of such programs is painful and involves lots of notation. So we will consider a specific example: the geometric program

$$(GP) \qquad \begin{cases} \underset{t_1,t_2>0}{\text{minimize}} & \dfrac{4}{t_1} + \dfrac{9}{t_2} \\ \text{subject to} & t_1 + t_2 \le 1. \end{cases}$$

The ideas we see here apply to all constrained geometric programs, and we will see the general form of the dual in the next lecture.

We modify GP slightly. First, since we know $t_i > 0$, we can let $x_i = \log t_i$ and work with the variables x_1, x_2 instead. Now, the domain of (x_1, x_2) is all of $\mathbb{R}^2$, and the objective function and constraints are convex functions of $\mathbf{x}$. We get:

$$(GP') \qquad \begin{cases} \underset{x_1,x_2\in\mathbb{R}}{\text{minimize}} & 4e^{-x_1} + 9e^{-x_2} \\ \text{subject to} & e^{x_1} + e^{x_2} \le 1. \end{cases}$$

Second, we want to be able to look at each of the posynomial terms in this problem (both the ones in the constraint and the ones in the objective function) separately. So we replace the exponents by z_1, z_2, z_3, z_4 and write down the problem

$$(GP'') \qquad \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^2,\mathbf{z}\in\mathbb{R}^4}{\text{minimize}} & e^{z_1} + e^{z_2} \\ \text{subject to} & e^{z_3} + e^{z_4} \le 1 \\ & z_1 \ge \log 4 - x_1 \\ & z_2 \ge \log 9 - x_2 \\ & z_3 \ge x_1 \\ & z_4 \ge x_2. \end{cases}$$

We write the constraint $z_1 \ge \log 4 - x_1$ rather than $z_1 = \log 4 - x_1$ (the natural choice to replace $4e^{-x_1}$ by e^{z_1}) for two reasons:

- The KKT theorem is best suited for dealing with inequalities, not equations.
- In all cases, we are happier when e^{z_i} and therefore z_i is smaller, so a lower bound on z_i is always going to be tight.

There are 5 constraints, so there are 5 dual variables. Call them $\mu, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ in order of the constraints; μ is special because it goes with the $e^{z_3} + e^{z_4} \le 1$ constraint.

The Lagrangian is

$$\begin{aligned} L(\mathbf{x}, \mathbf{z}, \mu, \boldsymbol{\lambda}) = e^{z_1} &+ e^{z_2} + \mu(e^{z_3} + e^{z_4} - 1) \\ &+ \lambda_1(\log 4 - x_1 - z_1) \\ &+ \lambda_2(\log 9 - x_2 - z_2) \\ &+ \lambda_3(x_1 - z_3) \\ &+ \lambda_4(x_2 - z_4). \end{aligned}$$

Remember, the dual objective function is $h(\mu, \boldsymbol{\lambda}) = \inf\{L(\mathbf{x}, \mathbf{z}, \mu, \boldsymbol{\lambda}) : \mathbf{x} \in \mathbb{R}^2, \mathbf{z} \in \mathbb{R}^4\}$.

Pretty much always, our first step—when we can do it—is to isolate the dependencies on the primal variables $\mathbf{x}$ and $\mathbf{z}$, which gives us

$$\begin{aligned} L(\mathbf{x}, \mathbf{z}, \mu, \boldsymbol{\lambda}) = (e^{z_1} &- \lambda_1 z_1) + (e^{z_2} - \lambda_2 z_2) \\ &+ (\mu e^{z_3} - \lambda_3 z_3) + (\mu e^{z_4} - \lambda_4 z_4) \\ &+ (-\lambda_1 + \lambda_3)x_1 \\ &+ (-\lambda_2 + \lambda_4)x_2 \\ &+ \lambda_1 \log 4 + \lambda_2 \log 9 - \mu. \end{aligned}$$

Now we can deduce some dual constraints by thinking about how we optimize our choices of $\mathbf{x}$ and $\mathbf{z}$.

- The dependence on x_1 is linear, so unless the slope $-\lambda_1 + \lambda_3$ is 0, we can make L as negative as we want and $h(\mu, \boldsymbol{\lambda}) = -\infty$. We add $-\lambda_1 + \lambda_3 = 0$ as a constraint and eliminate x_1 from the problem.
- Similarly, for x_2 we add the dual constraint $-\lambda_2 + \lambda_4 = 0$, then forget all about x_2.
- For $i = 1, 2$, the part of Lagrangian depending on z_i is $e^{z_i} - \lambda_i z_i$, which is minimized when $z_i = \log \lambda_i$. So we make this substitution (and remember this choice of z_i for later).
- For $i = 3, 4$, the part of the Lagrangian depending on z_i is $\mu e^{z_i} - \lambda_i z_i$, which is minimized when $z_i = \log \frac{\lambda_i}{\mu}$. So we make this substitution (and remember this choice of z_i for later).

Having cleaned everything up, we get

$$\begin{aligned} h(\mu, \boldsymbol{\lambda}) &= (\lambda_1 - \lambda_1 \log \lambda_1) + (\lambda_2 - \lambda_2 \log \lambda_2) + (\lambda_3 - \lambda_3 \log \frac{\lambda_3}{\mu}) + (\lambda_4 - \lambda_4 \log \frac{\lambda_4}{\mu}) + \lambda_1 \log 4 + \lambda_2 \log 9 - \mu \\ &= (\lambda_1 + \lambda_2) + (\lambda_3 + \lambda_4) \log \mu - \mu + \log \left[\left(\frac{4}{\lambda_1}\right)^{\lambda_1} \left(\frac{9}{\lambda_2}\right)^{\lambda_2} \left(\frac{1}{\lambda_3}\right)^{\lambda_3} \left(\frac{1}{\lambda_4}\right)^{\lambda_4} \right] \end{aligned}$$

with the constraints

$$-\lambda_1 + \lambda_3 = -\lambda_2 + \lambda_4 = 0, \qquad \mu, \lambda_1, \lambda_2, \lambda_3, \lambda_4 \ge 0.$$

The dependence on μ is particularly simple: $(\lambda_3 + \lambda_4) \log \mu - \mu$ is maximized[2] when $\mu = \lambda_3 + \lambda_4$. So we can make that substitution, and then forget about μ forever.

The result is the dual program

$$(D) \qquad \begin{cases} \underset{\boldsymbol{\lambda} \in \mathbb{R}^4}{\text{maximize}} & h(\boldsymbol{\lambda}) = \lambda_1 + \lambda_2 + \log \left[\left(\frac{4}{\lambda_1}\right)^{\lambda_1} \left(\frac{9}{\lambda_2}\right)^{\lambda_2} \left(\frac{1}{\lambda_3}\right)^{\lambda_3} \left(\frac{1}{\lambda_4}\right)^{\lambda_4} (\lambda_3 + \lambda_4)^{\lambda_3 + \lambda_4} \right] \\ \text{subject to} & -\lambda_1 + \lambda_3 = 0 \\ & -\lambda_2 + \lambda_4 = 0 \\ & \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0. \end{cases}$$

Take a deep breath.

By now, you should be starting to recognize the geometric programming dual we derived back in Chapter 2. The two constraints are exactly the "power of t_1" and "power of t_2" constraints we usually have.

We don't have a "things add up to 1" constraint yet, so let's deduce that one next!

Write the objective function as $h(\boldsymbol{\lambda}) = \lambda_1 + \lambda_2 + \log v(\boldsymbol{\lambda})$, where $v(\boldsymbol{\lambda})$ is shorthand for the function we have inside the log. The key observation is that right now, if any $\boldsymbol{\lambda}$ satisfies our constraints, so does $2\boldsymbol{\lambda}$, $3\boldsymbol{\lambda}$, or any other nonnegative multiple of $\boldsymbol{\lambda}$.

So write $\boldsymbol{\lambda} = s \cdot \boldsymbol{\delta}$ where $\boldsymbol{\delta}$ is a scaled version of $\boldsymbol{\lambda}$ with the extra constraint $\delta_1 + \delta_2 = 1$. (Why this one? Because it gives a nice final expression; we could have picked anything else.) Then

$$h(\boldsymbol{\lambda}) = h(s\boldsymbol{\delta}) = s + s \log v(\boldsymbol{\delta}) - s \log s.$$

(This equation hides a lot of simplification: you should think about how $v(s\boldsymbol{\delta})$ relates to $v(\boldsymbol{\delta})$ yourself, but the main change is that a factor of s appears in every exponent.)

Taking the derivative with respect to s gives $\frac{\partial h}{\partial s}(s\boldsymbol{\delta}) = \log v(\boldsymbol{\delta}) - \log s$, so the critical point is $s = v(\boldsymbol{\delta})$. This is happily a global maximizer again (the function is concave in s) and $s \log v(\boldsymbol{\delta}) - s \log s$ cancels to leave the objective function $h(s\boldsymbol{\delta}) = v(\boldsymbol{\delta})$. The final form of the GP dual is

$$(D) \qquad \begin{cases} \underset{\boldsymbol{\delta} \in \mathbb{R}^4}{\text{maximize}} & \left(\frac{4}{\delta_1}\right)^{\delta_1} \left(\frac{9}{\delta_2}\right)^{\delta_2} \left(\frac{1}{\delta_3}\right)^{\delta_3} \left(\frac{1}{\delta_4}\right)^{\delta_4} (\delta_3 + \delta_4)^{\delta_3 + \delta_4} \\ \text{subject to} & \delta_1 + \delta_2 = 1 \\ & -\delta_1 + \delta_3 = 0 \\ & -\delta_2 + \delta_4 = 0 \\ & \delta_1, \delta_2, \delta_3, \delta_4 \geq 0. \end{cases}$$

[2] It's a concave function with a single critical point, so that critical point is its global maximizer