CS261: A Second Course in Algorithms
Lecture #9: Linear Programming Duality (Part 2)*

Tim Roughgarden†

February 2, 2016

1 Recap

This is our third lecture on linear programming, and the second on linear programming duality. Let's page back in the relevant stuff from last lecture.

One type of linear program has the form

$$\max \sum_{j=1}^{n} c_j x_j$$

subject to

$$\begin{aligned}
\sum_{j=1}^{n} a_{1j} x_j &\leq b_1 \\
\sum_{j=1}^{n} a_{2j} x_j &\leq b_2 \\
\vdots &\leq \vdots \\
\sum_{j=1}^{n} a_{mj} x_j &\leq b_m \\
x_1, \ldots, x_n &\geq 0.
\end{aligned}$$

Call this linear program (P), for "primal." Alternatively, in matrix-vector notation it is

$$\max \mathbf{c}^T \mathbf{x}$$

†Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.

subject to

$$\begin{aligned} \mathbf{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq 0, \end{aligned}$$

where $\mathbf{c}$ and $\mathbf{x}$ are n-vectors, $\mathbf{b}$ is an m-vector, $\mathbf{A}$ is an $m \times n$ matrix (of the a_{ij}'s), and the inequalities are componentwise.

We then discussed a method for generating upper bounds on the maximum-possible objective function value of (P): take a nonnegative linear combination of the constraints so that the result dominates the objective $\mathbf{c}$, and you get an upper bound equal to the corresponding nonnegative linear combination of the right-hand side $\mathbf{b}$. A key point is that the tightest upper bound of this form is the solution to another linear program, known as the "dual." We gave a general recipe for taking duals: the dual has one variable per primal constraint and one constraint per primal variable; "max" and "min" get interchanged; the objective function and the right-hand side get interchanged; and the constraint matrix gets transposed. (There are some details about whether decision variables are nonnegative or not, and whether the constraints are equalities or inequalities; see the table last lecture.)

For example, the dual linear program for (P), call it (D), is

$$\min \mathbf{y}^T \mathbf{b}$$

subject to

$$\begin{aligned} \mathbf{A}^T \mathbf{y} &\geq \mathbf{c} \\ \mathbf{y} &\geq 0 \end{aligned}$$

in matrix-vector form. Or, if you prefer the expanded version,

$$\min \sum_{i=1}^{m} b_i y_i$$

subject to

$$\begin{aligned} \sum_{i=1}^{m} a_{i1} y_i &\geq c_1 \\ \sum_{i=1}^{m} a_{i2} y_i &\geq c_2 \\ \vdots &\geq \vdots \\ \sum_{i=1}^{m} a_{in} y_i &\geq c_n \\ y_1, \ldots, y_m &\geq 0. \end{aligned}$$

In all cases, the meaning of the dual is the tightest upper bound that can proved on the optimal primal objective function by taking suitable linear combinations of the primal constraints. With this understanding, we see that weak duality holds (for all forms of LPs), essentially by construction.

For example, for a primal-dual pair (P),(D) of the form above, for every pair $\mathbf{x}, \mathbf{y}$ of feasible solutions to (P),(D), we have

$$\underbrace{\sum_{j=1}^{n} c_j x_j}_{\mathbf{x}\text{'s obj fn}} \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j \tag{1}$$

$$= \sum_{i=1}^{m} y_i \left(\sum_{j=1}^{n} a_{ij} x_j \right) \tag{2}$$

$$\leq \underbrace{\sum_{i=1}^{m} y_i b_i}_{\mathbf{y}\text{'s obj fn}} . \tag{3}$$

Or, in matrix-vector notion,

$$\mathbf{c}^T \mathbf{x} \leq (\mathbf{A}^T \mathbf{y})^T \mathbf{x} = \mathbf{y}^T (\mathbf{A}\mathbf{x}) \leq \mathbf{y}^t \mathbf{b}.$$

The first inequality uses that $\mathbf{x} \geq 0$ and $\mathbf{A}^t \mathbf{y} \geq \mathbf{c}$; the second that $\mathbf{y} \geq 0$ and $\mathbf{A}\mathbf{x} \leq \mathbf{b}$.

We concluded last lecture with the following sufficient condition for optimality.[1]

Corollary 1.1 *Let (P),(D) be a primal-dual pair of linear programs. If* $\mathbf{x}, \mathbf{y}$ *are feasible solutions to (P),(D), and* $\mathbf{c}^T \mathbf{x} = \mathbf{y}^T \mathbf{b}$*, then both* $\mathbf{x}$ *and* $\mathbf{y}$ *are both optimal.*

For the reason, recall Figure 1 — no "x" can be to the right of an "o", so if an "x" and "o" are superimposed it must be the rightmost "x" and the leftmost "o." For an analogy, whenever you find a flow and s-t cut with the same value, the flow must be maximum and the cut minimum.

Figure 1: Illustrative figure showing feasible solutions for the primal (x) and the dual (o).

[1] We also noted that weak duality implies that whenever the optimal objective function of (P) is unbounded the linear program (D) is infeasible, and vice versa.

2 Complementary Slackness Conditions

2.1 The Conditions

Next is a corollary of Corollary 1.1. It is another sufficient (and as we'll see later, necessary) condition for optimality.

Corollary 2.1 (Complementary Slackness Conditions) *Let (P),(D) be a primal-dual pair of linear programs. If* $\mathbf{x}, \mathbf{y}$ *are feasible solutions to (P),(D), and the following two conditions hold then both* $\mathbf{x}$ *and* $\mathbf{y}$ *are both optimal.*

(1) Whenever $x_j \neq 0$, $\mathbf{y}$ *satisfies the* j*th constraint of (D) with equality.*

(2) Whenever $y_i \neq 0$, $\mathbf{x}$ *satisfies the* i*th constraint of (P) with equality.*

The conditions assert that no decision variable and corresponding constraint are simultaneously "slack" (i.e., it forbids that the decision variable is not 0 and also the constraint is not tight).

Proof of Corollary 2.1: We prove the corollary for the case of primal and dual programs of the form (P) and (D) in Section 1; the other cases are all the same.

The first condition implies that

$$c_j x_j = \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j$$

for each $j = 1, \ldots, n$ (either $x_j = 0$ or $c_j = \sum_{i=1}^{m} y_i a_{ij}$). Hence, inequality (1) holds with equality. Similarly, the second condition implies that

$$y_i \left(\sum_{j=1}^{n} a_{ij} x_i \right) = y_i b_i$$

for each $i = 1, \ldots, m$. Hence inequality (3) also holds with equality. Thus $\mathbf{c}^T \mathbf{x} = \mathbf{y}^T \mathbf{b}$, and Corollary 1.1 implies that both $\mathbf{x}$ and $\mathbf{y}$ are optimal. ■

2.2 Physical Interpretation

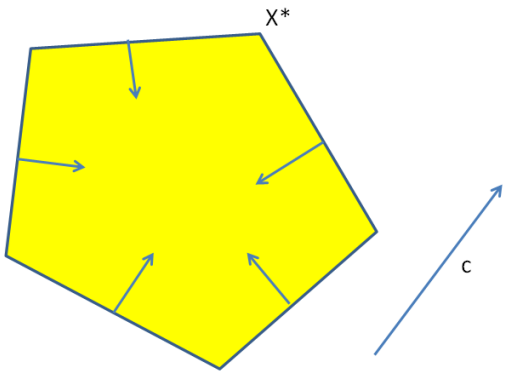

Figure 2: Physical interpretation of complementary slackness. The objective function pushes a particle in the direction c until it rests at x^*. Walls also exert a force on the particle, and complementary slackness asserts that only walls touching the particle exert a force, and sum of forces is equal to 0.

We offer the following informal physical metaphor for the complementary slackness conditions, which some students find helpful (Figure 2). For a linear program of the form (P) in Section 1, think of the objective function as exerting "force" in the direction $\mathbf{c}$. This pushes a particle in the direction $\mathbf{c}$ (within the feasible region) until it cannot move any further in this direction. When the particle comes to rest at position $\mathbf{x}^*$, the sum of the forces acting on it must sum to 0. What else exerts force on the particle? The "walls" of the feasible region, corresponding to the constraints. The direction of the force exerted by the ith constraint of the form $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ is perpendicular to the wall, that is, $-\mathbf{a}_i$, where $\mathbf{a}_i$ is the ith row of the constraint matrix. We can interpret the corresponding dual variable y_i as the magnitude of the force exerted in this direction $-\mathbf{a}_i$. The assertion that the sum of the forces equals 0 corresponds to the equation $\mathbf{c} = \sum_{i=1}^{n} y_i\mathbf{a}_i$. The complementary slackness conditions assert that $y_i > 0$ only when $\mathbf{a}_i^T\mathbf{x} = b_i$ — that is, only the walls that the particle touches are allowed to exert force on it.

2.3 A General Algorithm Design Paradigm

So why are the complementary slackness conditions interesting? One reason is that they offer three principled strategies for designing algorithms for solving linear programs and their special cases. Consider the following three conditions.

A General Algorithm Design Paradigm

1. $\mathbf{x}$ is feasible for (P).

2. $\mathbf{y}$ is feasible for (D).

3. $\mathbf{x}, \mathbf{y}$ satisfy the complementary slackness conditions (Corollary 2.1).

Pick two of these three conditions to maintain at all times, and work toward achieving the third.

By Corollary 2.1, we know that achieving these three conditions simultaneously implies that both $\mathbf{x}$ and $\mathbf{y}$ are optimal. Each choice of a condition to relax offers a disciplined way of working toward optimality, and in many cases all three approaches can lead to good algorithms. Countless algorithms for linear programs and their special cases can be viewed as instantiations of this general paradigm. We next revisit an old friend, the Hungarian algorithm, which is a particularly transparent example of this design paradigm in action.

3 Example #2: The Hungarian Algorithm Revisited

3.1 Recap of Example #1

Recall that in Lecture #8 we reinterpreted the max-flow/min-cut theorem through the lens of LP duality (this was "Example #1"). We had a primal linear program formulation of the maximum flow problem. In the corresponding dual linear program, we observed that s-t cuts translate to 0-1 solutions to this dual, with the dual objective function value equal to the capacity of the cut. Using the max-flow/min-cut theorem, we concluded two interesting properties: first, we verified strong duality (i.e., no gap between the optimal primal and dual objective function values) for primal-dual pairs corresponding to flows and (fractional) cuts; second, we concluded that these dual linear programs are always guaranteed to possess an integral optimal solution (i.e., fractions don't help).

3.2 The Primal Linear Program

Back in Lecture #7 we claimed that all of the problems studied thus far are special cases of linear programs. For the maximum flow problem, this is easy to believe, because flows can be fractional. But for matchings? They are suppose to be integral, so how could they be modeled with a linear program? Example #1 provides the clue — sometimes, linear programs are guaranteed to have an optimal integral solution. As we'll see, this also turns out to be the case for bipartite matching.

Given a bipartite graph $G = (V \cup W, E)$ with a cost c_e for each edge, the relevant linear program (P-BM) is

$$\min \sum_{e \in E} c_e x_e$$

subject to

$$\sum_{e \in \delta(v)} x_e = 1 \qquad \text{for all } v \in V \cup W$$
$$x_e \geq 0 \qquad \text{for all } e \in E,$$

where $\delta(v)$ denotes the edges incident to v. The intended semantics is that each x_e is either equal to 1 (if e is in the chosen matching) or 0 (otherwise). Of course, the linear program is also free to use fractional values for the decision variables.[2]

In matrix-vector form, this linear program is

$$\min \mathbf{c}^T \mathbf{x}$$

subject to

$$\mathbf{A}\mathbf{x} = \mathbf{1}$$
$$\mathbf{x} \geq 0,$$

where $\mathbf{A}$ is the $(V \cup W) \times E$ matrix

$$\mathbf{A} = \left[a_{ve} = \begin{cases} 1 \text{ if } e \in \delta(v) \\ 0 \text{ otherwise} \end{cases} \right] \tag{4}$$

3.3 The Dual Linear Program

We now turn to the dual linear program. Note that (P-BM) differs from our usual form both by having a minimization objective and by having equality (rather than inequality) constraints. But our recipe for taking duals from Lecture #8 applies to all types of linear programs, including this one.

When taking a dual, usually the trickiest point is to understand the effect of the transpose operation (on the constraint matrix). In the constraint matrix $\mathbf{A}$ in (4), each row (indexed by $v \in V \cup W$) has a 1 in each column (indexed by $e \in E$) for which e is incident to v (and 0s in other columns). Thus, a column of $\mathbf{A}$ (and hence row of $\mathbf{A}^T$) corresponding to edge e has 1s in precisely the rows (indexed by v) such that e is incident to v — that is, in the two rows corresponding to e's endpoints.

Applying our recipe for duals to (P-BM), initially in matrix-vector form for simplicity, yields

$$\max \mathbf{p}^T \mathbf{1}$$

subject to

$$\mathbf{A}^T \mathbf{p} \leq \mathbf{c}$$
$$\mathbf{p} \in \mathbb{R}^E.$$

[2] If you're tempted to also add in the constraints that $x_e \leq 1$ for every $e \in E$, note that these are already implied by the current constraints (why?).

We are using the notation p_v for the dual variable corresponding to a vertex $v \in V \cup W$, for reasons that will become clearly shortly. Note that these decision variables can be positive or negative, because of the equality constraints in (P-BM).

Unpacking this dual linear program, (D-BM), we get

$$\max \sum_{v \in V \cup W} p_v$$

subject to

$$\begin{aligned} p_v + p_w &\leq c_{vw} \qquad \text{for all } (v,w) \in E \\ p_v &\in \mathbb{R} \qquad \text{for all } v \in V \cup W. \end{aligned}$$

Here's the punchline: the "vertex prices" in the Hungarian algorithm (Lecture #5) correspond exactly to the decision variables of the dual (D-BM). Indeed, without thinking about this dual linear program, how would you ever think to maintain numbers attached to the *vertices* of a graph matching instance, when the problem definition seems to only concern the graph's edges?[3]

It gets better: rewrite the constraints of (D-BM) as

$$\underbrace{c_{vw} - p_v - p_w}_{\text{reduced cost}} \geq 0 \tag{5}$$

for every edge $(v,w) \in E$. The left-hand side of (5) is exactly our definition in the Hungarian algorithm of the "reduced cost" of an edge (with respect to prices $\mathbf{p}$). Thus the first invariant of the Hungarian algorithm, asserting that all edges have nonnegative reduced costs, is exactly the same as maintaining the dual feasibility of $\mathbf{p}$!

To seal the deal, let's check out the complementary slackness conditions for the primal-dual pair (P-BM),(D-BM). Because all constraints in (P-BM) are equations (not counting the nonnegativity constraints), the second condition is trivial. The first condition states that whenever $x_e > 0$, the corresponding constraint (5) should hold with equality — that is, edge e should have zero reduced cost. Thus the second invariant of the Hungarian algorithm (that edges in the current matching should be "tight") is just the complementary slackness condition!

We conclude that, in terms of the general algorithm design paradigm in Section 2.3, the Hungarian algorithm maintains the second two conditions ($\mathbf{p}$ is feasible for (D-BM) and complementary slackness conditions) at all times, and works toward the first condition (primal feasibility, i.e., a perfect matching). Algorithms of this type are called *primal-dual algorithms*, and the Hungarian algorithm is a canonical example.

[3] In Lecture #5 we motivated vertex prices via an analogy with the vertex labels maintained by the push-relabel maximum flow algorithm. But the latter is from the 1980s and the former from the 1950s, so that was a pretty ahistorical analogy. Linear programming (and duality) were only developed in the late 1940s, and so it was a new subject when Kuhn designed the Hungarian algorithm. But he was one of the first masters of the subject, and he put his expertise to good use.

3.4 Consequences

We know that

$$\text{OPT of (D-PM)} \leq \text{OPT of (P-PM)} \leq \text{min-cost perfect matching.} \tag{6}$$

The first inequality is just weak duality (for the case where the primal linear program has a minimization objective). The second inequality follows from the fact that every perfect matching corresponds to a feasible (0-1) solution of (P-BM); since the linear program minimizes over a superset of these solutions, it can only have a better (i.e., smaller) optimal objective function value.

In Lecture #5 we proved that the Hungarian algorithm always terminates with a perfect matching (provided there is at least one). The algorithm maintains a feasible dual and the complementary slackness conditions. As in the proof of Corollary 2.1, this implies that the cost of the constructed perfect matching equals the dual objective function value attained by the final prices. That is, both inequalities in (6) must hold with equality.

As in Example #1 (max flow/min cut), both of these equalities are interesting. The first equation verifies another special case of strong LP duality, for linear programs of the form (P-BM) and (D-BM). The second equation provides another example of a natural family of linear programs — those of the form (P-BM) — that are guaranteed to have 0-1 optimal solutions.[4]

4 Strong LP Duality

4.1 Formal Statement

Strong linear programming duality ("no gap") holds in general, not just for the special cases that we've seen thus far.

Theorem 4.1 (Strong LP Duality) *When a primal-dual pair (P),(D) of linear programs are both feasible,*

$$\textit{OPT for (P)} = \textit{OPT for (D)}.$$

Amazingly, our simple method of deriving bounds on the optimal objective function value of (P) through suitable linear combinations of the constraints is always guaranteed to produce the tightest-possible bound! Strong duality can be thought of as a generalization of the max-flow/min-cut theorem (Lecture #2) and Hall's theorem (Lecture #5), and as the ultimate answer to the question "how do we know when we're done?"[5]

[4] See also Exercise Set #4 for a direct proof of this.

[5] When at least one of (P),(D) is infeasible, there are three possibilities, all of which can occur. First, (P) might have unbounded objective function value, in which case (by weak duality) (D) is infeasible. It is also possible that (P) is infeasible while (D) has unbounded objective function value. Finally, sometimes both (P) and (D) are infeasible (an uninteresting case).

4.2 Consequent Optimality Conditions

Strong duality immediately implies that the sufficient conditions for optimality identified earlier (Corollaries 1.1 and 2.1) are also necessary conditions — that is, they are *optimality conditions* in the sense derived earlier for the maximum flow and minimum-cost perfect bipartite matching problems.

Corollary 4.2 (LP Optimality Conditions) *Let* $\mathbf{x}, \mathbf{y}$ *are feasible solutions to the primal-dual pair (P),(D) be a = primal-dual pair, then*

$\mathbf{x}, \mathbf{y}$ *are both optimal* *if and only if* $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$
if and only if *the complementary slackness conditions hold.*

The first if and only if follows from strong duality: since both (P),(D) are feasible by assumption, strong duality assures us of feasible solutions $\mathbf{x}^*, \mathbf{y}^*$ with $\mathbf{c}\mathbf{x}^* = (\mathbf{y}^*)^T\mathbf{b}$. If $\mathbf{x}, \mathbf{y}$ fail to satisfy this equality, then either $\mathbf{c}^T\mathbf{x}$ is worse than $\mathbf{c}^T\mathbf{x}^*$ or $\mathbf{y}^T\mathbf{b}$ is worse than $(\mathbf{y}^*)^T\mathbf{b}$ (or both). The second if and only if does not require strong duality; it follows from the proof of Corollary 2.1 (see also Exercise Set #4).

4.3 Proof Sketch: The Road Map

We conclude the lecture with a proof sketch of Theorem 4.1. Our proof sketch leaves some details to Problem Set #3, and also takes on faith one intuitive geometric fact. The goal of the proof sketch is to at least partially demystify strong LP duality, and convince you that it ultimately boils down to some simple geometric intuition.

Here's the plan:

$$\underbrace{\text{separating hyperplane}}_{\text{will assume}} \Rightarrow \underbrace{\text{Farkas's Lemma}}_{\text{will prove}} \rightarrow \underbrace{\text{strong LP duality}}_{\text{PSet \#3}}.$$

The "separating hyperplane theorem" is the intuitive geometric fact that we assume (Section 4.4). Section 4.5 derives from this fact Farkas's Lemma, a "feasibility version" of strong LP duality. Problem Set #3 asks you to reduce strong LP duality to Farkas's Lemma.

4.4 The Separating Hyperplane Theorem

In Lecture #7 we discussed separating hyperplanes, in the context of separating data points labeled "positive" from those labeled "negative." There, the point was to show that the computational problem of finding such a hyperplane reduces to linear programming. Here, we again discuss separating hyperplanes, with two differences: first, our goal is to separate a convex set from a point not in the set (rather than two different sets of points); second, the point here is to prove strong LP duality, not to give an algorithm for a computational problem.

We assume the following result.

Theorem 4.3 (Separating Hyperplane) *Let C be a closed and convex subset of $\mathbb{R}^n$, and $\mathbf{z}$ a point in $\mathbb{R}^n$ not in C. Then there is a* separating hyperplane*, meaning coefficients $\alpha \in \mathbb{R}^n$ and an intercept $\beta \in \mathbb{R}$ such that:*

(1)

$$\underbrace{\alpha^T \mathbf{x} \geq \beta \qquad \text{for all } \mathbf{x} \in C}_{\text{all of } C \text{ on one side of hyperplane}};$$

(2)

$$\underbrace{\alpha^T \mathbf{z} < \beta}_{\mathbf{z} \text{ on other side}} .$$

See also Figure 3. Note that the set C is not assumed to be bounded.

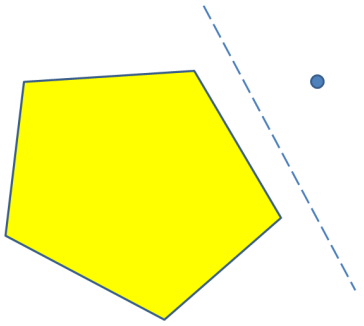

Figure 3: Illustration of separating hyperplane theorem.

If you've forgotten what "convex" or "closed" means, both are very intuitive. A convex set is "filled in," meaning it contains all of its chords. Formally, this translates to

$$\underbrace{\lambda \mathbf{x} + (1-\lambda)\mathbf{y}}_{\text{point on chord between } \mathbf{x}, \mathbf{y}} \in C$$

for all $\mathbf{x}, \mathbf{y} \in C$ and $\lambda \in [0, 1]$. See Figure 4 for an example (a filled-in polygon) and a non-example (an annulus).

A closed set is one that includes its boundary.[6] See Figure 5 for an example (the unit disc) and a non-example (the open unit disc).

[6] One formal definition is that whenever a sequence of points in C converges to a point $\mathbf{x}^*$, then $\mathbf{x}^*$ should also be in C.

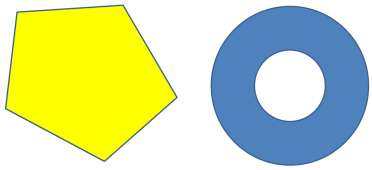

Figure 4: (a) a convex set (filled-in polygon) and (b) a non-convex set (annulus)

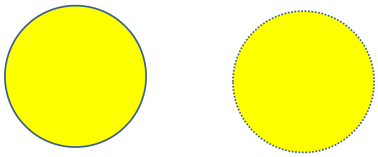

Figure 5: (a) a closed set (unit disc) and (b) non-closed set (open unit disc)

Hopefully Theorem 4.3 seems geometrically obvious, at least in two and three dimensions. It turns out that the math one would use to prove this formally extends without trouble to an arbitrary number of dimensions.[7] It also turns out that strong LP duality boils down to exactly this fact.

4.5 Farkas's Lemma

It's easy to convince someone whether or not a system of linear equations has a solution: just run Gaussian elimination and see whether or not it finds a solution (if there is a solution, Gaussian elimination will find one). For a system of linear *inequalities*, it's easy to convince someone that there is a solution — just exhibit it and let them verify all the constraints. But how would you convince someone that a system of linear inequalities has *no* solution? You can't very well enumerate the infinite number of possibilities and check that each doesn't work. *Farkas's Lemma* is a satisfying answer to this question, and can be thought of as the "feasibility version" of strong LP duality.

Theorem 4.4 (Farkas's Lemma) *Given a matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and a right-hand side* $\mathbf{b} \in \mathbb{R}^m$*, exactly one of the following holds:*

(i) There exists $\mathbf{x} \in \mathbb{R}^n$ *such that* $\mathbf{x} \geq 0$ *and* $\mathbf{A}\mathbf{x} = \mathbf{b}$*;*

(ii) There exists $\mathbf{y} \in \mathbb{R}^m$ *such that* $\mathbf{y}^T\mathbf{A} \geq 0$ *and* $\mathbf{y}^T\mathbf{b} < 0$*.*

[7] If you know undergraduate analysis, then even the formal proof is not hard: let $\mathbf{y}$ be the nearest neighbor to $\mathbf{z}$ in C (such a point exists because C is closed), and take a hyperplane perpendicular to the line segment between $\mathbf{y}$ and $\mathbf{z}$, through the midpoint of this segment (cf., Figure 3). All of C lies on the same side of this hyperplane (opposite of $\mathbf{z}$) because C is convex and $\mathbf{y}$ is the nearest neighbor of $\mathbf{z}$ in C.

To connect the statement to the previous paragraph, think of $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq 0$ as the linear system of inequalities that we care about, and solutions to (ii) as proofs that this system has no feasible solution.

Just like there are many variants of linear programs, there are many variants of Farkas's Lemma. Given Theorem 4.4, it is not hard to translate it to analogous statements for other linear systems of inequalities (e.g., with both inequality and nonnegativity constraints); see Problem Set #3.

Proof of Theorem 4.4: First, we have deliberately set up (i) and (ii) so that it's impossible for both to have a feasible solution. For if there were such an $\mathbf{x}$ and $\mathbf{y}$, we would have

$$\underbrace{(\mathbf{y}^T\mathbf{A})}_{\geq 0}\underbrace{\mathbf{x}}_{\geq 0} \geq 0$$

and yet

$$\mathbf{y}^T(\mathbf{Ax}) = \mathbf{y}^T\mathbf{b} < 0,$$

a contradiction. In this sense, solutions to (ii) are proofs of infeasibility of the the system (i) (and vice versa).

But why can't both (i) and (ii) be infeasible? We'll show that this can't happen by proving that, whenever (i) is infeasible, (ii) is feasible. Thus the "proofs of infeasibility" encoded by (ii) are all that we'll ever need — whenever the linear system (i) is infeasible, there is a proof of it of the prescribed type. There is a clear analogy between this interpretation of Farkas's Lemma and strong LP duality, which says that there is always a feasible dual solution proving the tightest-possible bound on the optimal objective function value of the primal.

Assume that (i) is infeasible. We need to somehow exhibit a solution to (ii), but where could it come from? The trick is to get it from the separating hyperplane theorem (Theorem 4.3) — the coefficients defining the hyperplane will turn out to be a solution to (ii). To apply this theorem, we need a closed convex set and a point not in the set.

Define

$$Q = \{\mathbf{d} : \exists \mathbf{x} \geq 0 \text{ s.t. } \mathbf{Ax} = \mathbf{d}\}.$$

Note that Q is a subset of $\mathbb{R}^m$. There are two different and equally useful ways to think about Q. First, for the given constraint matrix $\mathbf{A}$, Q is the set of all right-hand sides $\mathbf{d}$ that are feasible (in $\mathbf{x} \geq 0$) with this constraint matrix. Thus by assumption, $\mathbf{b} \notin Q$. Equivalently, considering all vectors of the form $\mathbf{Ax}$, with $\mathbf{x}$ ranging over all nonnegative vectors in $\mathbb{R}^n$, generates precisely the set of feasible right-hand sides. Thus Q equals the set of all nonnegative linear combinations of the columns of $\mathbf{A}$.[8] This definition makes it obvious that Q is convex (an average of two nonnegative linear combinations is just another nonnegative linear combination). Q is also closed (the limit of a convergent sequence of nonnegative linear combinations is just another nonnegative linear combination).

[8]Called the "cone generated by" the columns of $\mathbf{A}$.

Since Q is closed and convex and $\mathbf{b} \notin Q$, we can apply Theorem 4.3. In return, we are granted a coefficient vector $\alpha \in \mathbb{R}^m$ and an intercept $\beta \in \mathbb{R}$ such that

$$\alpha^T \mathbf{d} \geq \beta$$

for all $\mathbf{d} \in Q$ and

$$\alpha^T \mathbf{b} < \beta.$$

An exercise shows that, since Q is a cone, we can take $\beta = 0$ without loss of generality (see Exercise Set #5). Thus

$$\alpha^T \mathbf{d} \geq 0 \tag{7}$$

for all $\mathbf{d} \in Q$ while

$$\alpha^T \mathbf{b} < 0. \tag{8}$$

A solution $\mathbf{y}$ to (ii) satisfies $\mathbf{y}^T \mathbf{A} \geq 0$ and $\mathbf{y}^T \mathbf{b} < 0$. Suppose we just take $\mathbf{y} = \alpha$. Inequality (8) implies the second condition, so we just have to check that $\alpha^T \mathbf{A} \geq 0$. But what is $\alpha^T \mathbf{A}$? An n-vector, where the jth coordinate is inner product of α^T and the jth column $\mathbf{a}^j$ of $\mathbf{A}$. Since each $\mathbf{a}^j \in Q$ — the jth column is obviously one particular nonnegative linear combination of $\mathbf{A}$'s columns — inequality (7) implies that every coordinate of $\alpha^T \mathbf{A}$ is nonnegative. Thus α is a solution to (ii), as desired. ■

4.6 Epilogue

On Problem Set #3 you will use Theorem 4.4 to prove strong LP duality. The idea is simple: let $OPT_{(D)}$ denote the optimal value of the dual linear program, add a constraint to the primal stating that the (primal) objective function value must be equal to or better than $OPT_{(D)}$, and use Farkas's Lemma to prove that this augmented linear program is feasible.

In summary, strong LP duality is amazing and powerful, yet it ultimately boils down to the highly intuitive existence of a separating hyperplane between a closed convex set and a point not in the set.