CS261: A Second Course in Algorithms
Lecture #8: Linear Programming Duality (Part 1)*

Tim Roughgarden†

January 28, 2016

1 Warm-Up

This lecture begins our discussion of linear programming duality, which is the really the heart and soul of CS261. It is the topic of this lecture, the next lecture, and (as will become clear) pretty much all of the succeeding lectures as well.

Recall from last lecture the ingredients of a linear program: decision variables, linear constraints (equalities or inequalities), and a linear objective function. Last lecture we saw that lots of interesting problems in combinatorial optimization and machine learning reduce to linear programming.

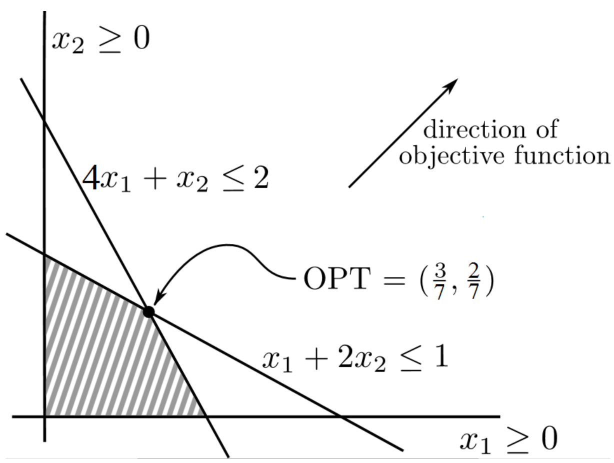

Figure 1: A toy example to illustrate duality.

*©2016, Tim Roughgarden.

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

To start getting a feel for linear programming duality, let's begin with a toy example. It is a minor variation on our toy example from last time. There are two decision variables x_1 and x_2 and we want to

$$\max x_1 + x_2 \tag{1}$$

subject to

$$4x_1 + x_2 \leq 2 \tag{2}$$
$$x_1 + 2x_2 \leq 1 \tag{3}$$
$$x_1 \geq 0 \tag{4}$$
$$x_2 \geq 0. \tag{5}$$

(Last lecture, the first constraint of our toy example read $2x_1 + x_2 \leq 1$; everything else is the same.)

Like last lecture, we can solve this LP just by eyeballing the feasible region (Figure 1) and searching for the "most northeastern" feasible point, which in this case is the vertex (i.e., "corner") at $(\frac{3}{7}, \frac{2}{7})$. Thus the optimal objective function value if $\frac{5}{7}$.

When we go beyond three dimensions (i.e., decision variables), it seems hopeless to solve linear programs by inspection. With a general linear program, even if we are handed on a silver platter an allegedly optimal solution, how do we know that it is really is optimal?

Let's try to answer this question at least in our toy example. What's an easy and convincing proof that the optimal objective function value of the linear program can't be too large? For starters, for any feasible point (x_1, x_2), we certainly have

$$\underbrace{x_1 + x_2}_{\text{objective}} \leq 4x_1 + x_2 \leq \underbrace{2}_{\text{upper bound}},$$

with the first inequality following from $x_1 \geq 0$ and the second from the first constraint. We can immediately conclude that the optimal value of the linear program is at most 2. But actually, it's obvious that we can do better by using the second constraint instead:

$$x_1 + x_2 \leq x_1 + 2x_2 \leq 1,$$

giving us a better (i.e., smaller) upper bound of 1. Can we do better? There's no reason we need to stop at using just one constraint at a time, and are free to blend two or more constraints. The best blending takes $\frac{1}{7}$ of the first constraint and $\frac{3}{7}$ of the second to give

$$x_1 + x_2 \leq \tfrac{1}{7}\underbrace{(4x_1 + x_2)}_{\leq 2 \text{ by (2)}} + \tfrac{3}{7}\underbrace{(x_1 + 2x_2)}_{\leq 1 \text{ by (3)}} \leq \frac{1}{7} \cdot 2 + \frac{3}{7} \cdot 1 = \frac{5}{7}. \tag{6}$$

(The first inequality actually holds with equality, but we don't need the stronger statement.) So this is a convincing proof that the optimal objective function value is at most $\frac{5}{7}$. Given the feasible point $(\frac{3}{7}, \frac{2}{7})$ that actually does realize this upper bound, we can conclude that $\frac{5}{7}$ really is the optimal value for the linear program.

Summarizing, for the linear program (1)–(5), there is a quick and convincing proof that the optimal solution has value at least $\frac{5}{7}$ (namely, the feasible point $(\frac{3}{7}, \frac{2}{7})$) and also such a proof that the optimal solution has value at most $\frac{5}{7}$ (given in (6)). This is the essence of linear programming duality.

2 The Dual Linear Program

We now generalize the ideas of the previous section. Consider an arbitrary linear program (call it (P)) of the form

$$\max \sum_{j=1}^{n} c_j x_j \tag{7}$$

subject to

$$\sum_{j=1}^{n} a_{1j} x_j \le b_1 \tag{8}$$

$$\sum_{j=1}^{n} a_{2j} x_j \le b_2 \tag{9}$$

$$\vdots \le \vdots \tag{10}$$

$$\sum_{j=1}^{n} a_{mj} x_j \le b_m \tag{11}$$

$$x_1, \ldots, x_n \ge 0. \tag{12}$$

This linear program has n nonnegative decision variables $x_1, \ldots, x_n$ and m constraints (not counting the nonnegativity constraints). The a_{ij}'s, b_i's, and c_j's are all part of the input (i.e., fixed constants).[1]

You may have forgotten your linear algebra, but it's worth paging the basics back in when learning linear programming duality. It's very convenient to write linear programs in matrix-vector notation. For example, the linear program above translates to the succinct description

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

subject to

$$\mathbf{A}\mathbf{x} \le \mathbf{b}$$
$$\mathbf{x} \ge 0,$$

[1] Remember that different types of linear programs are easily transformed to each other. A minimization objective can be turned into a maximization objective by multiplying all c_j's by -1. An equality constraint can be simulated by two inequality constraints. An inequality constraint can be flipped by multiplying by -1. Real-valued decision variables can be simulated by the difference of two nonnegative decision variables. An inequality constraint can be turned into an equality constraint by adding an extra "slack" variable.

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.

Remember our strategy for deriving upper bounds on the optimal objective function value of our toy example: take a nonnegative linear combination of the constraints that (componentwise) dominates the objective function. In general, for the above linear program with m constraints, we denote by $y_1, \ldots, y_m \geq 0$ the corresponding multipliers that we use. The goal of dominating the objective function translates to the conditions

$$\sum_{i=1}^{m} y_i a_{ij} \geq c_j \tag{13}$$

for each objective function coefficient (i.e. for $j = 1, 2, \ldots, m$). In matrix notation, we are interested in nonnegative m-vectors $\mathbf{y} \geq 0$ such that

$$\mathbf{A}^T \mathbf{y} \geq \mathbf{c};$$

note the sum in (13) is over the rows i of A, which corresponds to an inner product with the jth column of $\mathbf{A}$, or equivalently with the jth row of $\mathbf{A}^T$.

By design, every such choice of multipliers $y_1, \ldots, y_m$ implies an upper bound on the optimal objective function value of the linear program (7)–(12): for every feasible solution $(x_1, \ldots, x_n)$,

$$\underbrace{\sum_{j=1}^{n} c_j x_j}_{\mathbf{x}\text{'s obj fn}} \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j \tag{14}$$

$$= \sum_{i=1}^{m} y_i \cdot \left(\sum_{j=1}^{n} a_{ij} x_j \right) \tag{15}$$

$$\leq \underbrace{\sum_{i=1}^{m} y_i b_i.}_{\text{upper bound}} \; . \tag{16}$$

In this derivation, inequality (14) follows from the domination condition in (13) and the nonnegativity of $x_1, \ldots, x_n$; equation (15) follows from reversing the order of summation; and inequality (16) follows from the feasibility of $\mathbf{x}$ and the nonnegativity of $y_1, \ldots, y_m$.

Alternatively, the derivation may be more transparent in matrix-vector notation:

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

The upshot is that, whenever $\mathbf{y} \geq 0$ and (13) holds,

$$\text{OPT of (P)} \leq \sum_{i=1}^{m} b_i y_i.$$

In our toy example of Section 1, the first upper bound of 2 corresponds to taking $y_1 = 1$ and $y_2 = 0$. The second upper bound of 1 corresponds to $y_1 = 0$ and $y_2 = 1$ The final upper bound of $\frac{5}{7}$ corresponds to $y_1 = \frac{1}{7}$ and $y_2 = \frac{3}{7}$.

Our toy example illustrates that there can be many different ways of choosing the y_i's, and different choices lead to different upper bounds on the optimal value of the linear program (P). Obviously, the most interesting of these upper bounds is the tightest (i.e., smallest) one. So we really want to range over all possible $\mathbf{y}$'s and consider the minimum such upper bound.[2]

Here's the key point: *the tightest upper bound on OPT is itself the optimal solution to a linear program.* Namely:

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

subject to

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

Or, in matrix-vector form:

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

subject to

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

This linear program is called the *dual* to (P), and we sometimes denote it by (D).

For example, to derive the dual to our toy linear program, we just swap the objective and the right-hand side and take the transpose of the constraint matrix:

$$\min 2y_1 + y_2$$

[2]For an analogy, among all s-t cuts, each of which upper bounds the value of a maximum flow, the minimum cut is the most interesting one (Lecture #2). Similarly, in the Tutte-Berge formula (Lecture #5), we were interested in the tightest (i.e., minimum) upper bound of the form $|V| - (\text{oc}(S) - |S|)$, over all choices of the set S.

subject to

$$4y_1 + y_2 \geq 1$$
$$y_1 + 2y_2 \geq 1$$
$$y_1, y_2 \geq 0.$$

The objective function values of the feasible solutions $(1,0)$, $(0,1)$, and $(\frac{1}{7}, \frac{3}{7})$ (of 2, 1, and $\frac{5}{7}$) correspond to our three upper bounds in Section 1.

The following important result follows from the definition of the dual and the derivation (14)–(16).

Theorem 2.1 (Weak Duality) *For every linear program of the form (P) and corresponding dual linear program (D),*

$$\textit{OPT value for (P)} \leq \textit{OPT value for (D)}. \tag{17}$$

(Since the derivation (14)–(15) applies to any pair of feasible solutions, it holds in particular for a pair of optimal solutions.) Next lecture we'll discuss *strong duality*, which asserts that (17) always holds with equality (as long as both (P) and (D) are feasible).

3 Duality Example #1: Max-Flow/Min-Cut Revisited

This section brings linear programming duality back down to earth by relating it to an old friend, the maximum flow problem. Last lecture we showed how this problem translates easily to a linear program. This lecture, for convenience, we will use a different linear programming formulation. The new linear program is much bigger but also simpler, so it is easier to take and interpret its dual.

3.1 The Primal

The idea is to work directly with path decompositions, rather than flows. So the decision variables have the form f_P, where P is an s-t path. Let $\mathcal{P}$ denote the set of all such paths. The benefit of working with paths is that there is no need to explicitly state the conservation constraints. We do still have the capacity (and nonnegativity) constraints, however.

$$\max \sum_{P \in \mathcal{P}} f_P \tag{18}$$

subject to

$$\underbrace{\sum_{P \in \mathcal{P} \,:\, e \in P} f_P}_{\text{total flow on } e} \leq u_e \qquad \text{for all } e \in E \tag{19}$$

$$f_P \geq 0 \qquad \text{for all } P \in \mathcal{P}. \tag{20}$$

Again, call this (P). The optimal value to this linear program is the same as that of the linear programming formulation of the maximum flow problem given last lecture. Every feasible solution to (18)–(20) can be transformed into one of equal value for last lecture's LP, just by setting f_e equal to the left-hand side of (19) for each e. For the reverse direction, one takes a path decomposition (Problem Set #1). See Exercise Set #4 for details.

3.2 The Dual

The linear program (18)–(20) conforms to the format covered in Section 2, so it has a well-defined dual. What is it? It's usually easier to take the dual in matrix-vector notation:

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

subject to

$$\begin{aligned} \mathbf{A}\mathbf{f} &\leq \mathbf{u} \\ \mathbf{f} &\geq 0, \end{aligned}$$

where the vector $\mathbf{f}$ is indexed by the paths $\mathcal{P}$, $\mathbf{1}$ stands for the ($|\mathcal{P}|$-dimensional) all-ones vector, $\mathbf{u}$ is indexed by E, and $\mathbf{A}$ is a $\mathcal{P} \times E$ matrix. Then, the dual (D) has decision variables indexed by E (denoted $\{\ell_e\}_{e \in E}$ for reasons to become clear) and is

$$\min \mathbf{u}^T \ell$$

$$\begin{aligned} \mathbf{A}^T \ell &\geq \mathbf{1} \\ \ell &\geq 0. \end{aligned}$$

Typically, the hardest thing about understanding a dual is interpreting what the transpose operation on the constraint matrix ($\mathbf{A} \mapsto \mathbf{A}^T$) is doing. By definition, each row (corresponding to an edge e) of A has a 1 in the column corresponding to a path P if $e \in P$, and 0 otherwise. So an entry a_{eP} of $\mathbf{A}$ is 1 if $e \in P$ and 0 otherwise. In the column of $\mathbf{A}$ (and hence row of $\mathbf{A}^T$) corresponding to a path P, there is a 1 in each row corresponding an edge e of P (and zeroes in the other rows).

Now that we understand $\mathbf{A}^T$, we can unpack the dual and write it as

$$\min \sum_{e \in E} u_e \ell_e$$

subject to

$$\begin{aligned} \sum_{e \in P} \ell_e &\geq 1 \qquad \text{for all } P \in \mathcal{P} & (21) \\ \ell_e &\geq 0 \qquad \text{for all } e \in E. \end{aligned}$$

3.3 Interpretation of Dual

The duals of natural linear programs are often meaningful in their own right, and this one is a good example. A key observation is that every s-t cut corresponds to a feasible solution to this dual linear program. To see this, fix a cut (A, B), with $s \in A$ and $t \in B$, and set

$$\ell_e = \begin{cases} 1 & \text{if } e \in \delta^+(A) \\ 0 & \text{otherwise.} \end{cases}$$

(Recall that $\delta^+(A)$ denotes the edges sticking out of A, with tail in A and head in B; see Figure 2.) To verify the constraints (21) and hence feasibility for the dual linear program, note that every s-t path must cross the cut (A, B) as some point (since it starts in A and ends in B). Thus every s-t path has at least one edge e with $\ell_e = 1$, and (21) holds. The objective function value of this feasible solution is

$$\sum_{e \in E} u_e \ell_e = \sum_{e \in \delta^+(A)} u_e = \text{capacity of } (A, B),$$

where the second equality is by definition (recall Lecture #2).

s-t-cuts correspond to one type of feasible solution to this dual linear program, where every decision variable is set to either 0 or 1. Not all feasible solutions have this property: any assignment of nonnegative "lengths" ℓ_e to the edges of G satisfying (21) is feasible. Note that (21) is equivalent to the constraint that the shortest-path distance from s to t, with respect to the edge lengths $\{\ell_e\}_{e \in E}$, is at least 1.[3]

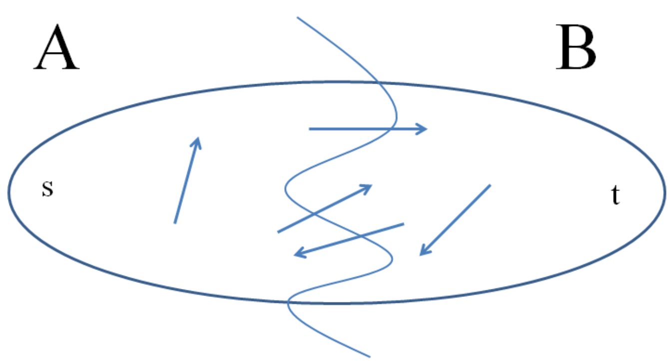

Figure 2: $\delta^+(A)$ denotes the two edges that point from A to B.

3.4 Relation to Max-Flow/Min-Cut

Summarizing, we have shown that

$$\text{max flow value} = \text{OPT of (P)} \leq \text{OPT of (D)} \leq \text{min cut value}. \qquad (22)$$

[3]To give a simple example, in the graph $s \to v \to t$, one feasible solution assigns $\ell_{sv} = \ell_{vt} = \frac{1}{2}$. If the edge (s, v) and (v, t) have the same capacity, then this is also an optimal solution.

The first equation is just the statement the maximum flow problem can be formulated as the linear program (P). The first inequality is weak duality. The second inequality holds because the feasible region of (D) includes all (0-1 solutions corresponding to) s-t cuts; since it minimizes over a superset of the s-t cuts, the optimal value can only be less than that of the minimum cut.

In Lecture #2 we used the Ford-Fulkerson algorithm to prove the maximum flow/minimum cut theorem, stating that there is never a gap between the maximum flow and minimum cut values. So the first and last terms of (22) are equal, which means that both of the inequalities are actually equalities. The fact that

$$\text{OPT of (P)} = \text{OPT of (D)}$$

is interesting because it proves a natural special case of strong duality, for flow linear programs and their duals. The fact that

$$\text{OPT of (D)} = \text{min cut value}$$

is interesting because it implies that the linear program (D), despite allowing fractional solutions, always admits an optimal solution in which each decision variable is either 0 or 1.

3.5 Take-Aways

The example in this section illustrates three general points.

1. The duals of natural linear programs are often natural in their own right.

2. Strong duality. (We verified it in a special case, and will prove it in general next lecture.)

3. Some natural linear programs are guaranteed to have integral optimal solutions.

4 Recipe for Taking Duals

Section 2 defines the dual linear program for primal linear programs of a specific form (maximization objective, inequality constraints, and nonnegative decision variables). As we've mentioned, different types of linear programs are easily converted to each other. So one perfectly legitimate way to take the dual of an arbitrary linear program is to first convert it into the form in Section 2 and then apply that definition. But it's more convenient to be able to take the dual of any linear program directly, using a general recipe.

The high-level points of the recipe are familiar: dual variables correspond to primal constraints, dual constraints correspond to primal variables, maximization and minimization get exchanged, the objective function and right-hand side get exchanged, and the constraint matrix gets transposed. The details concern the different type of constraints (inequality vs. equality) and whether or not decision variables are nonnegative.

Here is the general recipe for maximization linear programs:

Primal	Dual
variables $x_1, \ldots, x_n$	n constraints
m constraints	variables $y_1, \ldots, y_m$
objective function $\mathbf{c}$	right-hand side $\mathbf{c}$
right-hand side $\mathbf{b}$	objective function $\mathbf{b}$
$\max \mathbf{c}^T\mathbf{x}$	$\min \mathbf{b}^T\mathbf{y}$
constraint matrix $\mathbf{A}$	constraint matrix $\mathbf{A}^T$
ith constraint is "$\leq$"	$y_i \geq 0$
ith constraint is "$\geq$"	$y_i \leq 0$
ith constraint is "$=$"	$y_i \in \mathbb{R}$
$x_j \geq 0$	jth constraint is "$\geq$"
$x_j \leq 0$	jth constraint is "$\leq$"
$x_j \in \mathbb{R}$	jth constraint is "$=$"

For minimization linear programs, we define the dual as the reverse operation (from the right column to the left). Thus, by definition, the dual of the dual is the original primal.

5 Weak Duality

The above recipe allows you to take duals in a mechanical way, without thinking about it. This can be very useful, but don't forget the true meaning of the dual (which holds in all cases): *feasible dual solutions correspond to bounds on the best-possible primal objective function value (derived from taking linear combinations of the constraints), and the optimal dual solution is the tightest-possible such bound.*

If you remember the meaning of duals, then it's clear that weak duality holds in all cases (essentially by definition).[4]

Theorem 5.1 (Weak Duality) *For every maximization linear program (P) and corresponding dual linear program (D),*

$$\textit{OPT value for (P)} \leq \textit{OPT value for (D)};$$

for every minimization linear program (P) and corresponding dual linear program (D),

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

Weak duality can be visualized as in Figure 3. Strong duality also holds in all cases; see next lecture.

[4]Math classes often teach mathematical definitions as if they fell from the sky. This is not representative of how mathematics actually develops. Typically, definitions are reverse engineered so that you get the "right" theorems (like weak/strong duality).

Figure 3: visualization of weak duality. X represents feasible solutions for P while O represents feasible solutions for D.

Weak duality already has some very interesting corollaries.

Corollary 5.2 *Let (P),(D) be a primal-dual pair of linear programs.*

(a) If the optimal objective function value of (P) is unbounded, then (D) is infeasible.

(b) If the optimal objective function value of (D) is unbounded, then (P) is infeasible.

(c) If $\mathbf{x}, \mathbf{y}$ *are feasible for (P),(D) and* $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$*, then both* $\mathbf{x}$ *and* $\mathbf{y}$ *are both optimal.*

Parts (a) and (b) hold because any feasible solution to the dual of a linear program offers a bound on the best-possible objective function value of the primal (so if there is no such bound, then there is no such feasible solution). The hypothesis in (c) asserts that Figure 3 contains an "x" and an "o" that are superimposed. It is immediate that no other primal solution can be better, and that no other dual solution can be better. (For an analogy, in Lecture #2 we proved that capacity of every cut bounds from above the value of every flow, so if you ever find a flow and a cut with equal value, both must be optimal.)