Brief Introductory Note on Linear Programming Duality

Jasper Lee

(Updated: November 17, 2018)

This note is a brief introduction to the concept of *duality* in linear programming: given a *primal* (maximisation/minimisation) linear program P, the *dual* D of P is a (minimisation/maximisation *resp.*) linear program that is in a certain sense an *equivalent* formulation of P. Often, formulating an optimisation problem as a linear program (whenever possible) and taking its dual will reveal insights about the problem structure. The aim of this note is to motivate and illustrate the definition/construction of such duality. In particular, following the steps provided in these notes, you will be able to re-derive the "duality" table instead of having to memorise it for the purposes of the exam. (Memorising the table is still useful, since you can use it to take duals quicker.)

There is a small number of simple exercises throughout the note. Solutions are included in the appendix.

1 What's the goal?

As a general rule of thumb, when we give a mathematical definition, we also give corresponding characterisations and guarantees about the definition. In this case, since the dual D is constructed from the primal P, we shall aim for theorems relating the two programs. Thus, we are going to start by identifying the theorem we want to prove, and from that we derive the construction of the dual program.

Since the dual program D is meant to be an equivalent formulation of the primal program P, it is natural that we desire the guarantee of *strong duality*, which does in fact hold for linear programs.

Theorem 1 (Strong duality for linear programs)**.** *Given a primal linear program P that is feasible, let D be its LP dual. Then D is feasible and $Opt(P) = Opt(D)$, where $Opt()$ maps a linear program to its optimal objective value.*

Whilst strong duality is true of linear programs, in the general context of optimisation (namely for many kinds of non-linear programs), it is too strict a condition to hold. Instead, we aim for *weak duality*, a more general duality principle.

Theorem 2 (Weak duality for linear programs)**.** *Given a maximisation/minimisation linear program P that is feasible, let D be its dual (that is a minimisation/maximisation resp. problem). Assuming D is also feasible, then $Opt(P) \leq Opt(D)$ ($Opt(P) \geq Opt(D)$ resp.), where $Opt()$ maps a linear program to its optimal objective value.*

Weak duality is a weaker but still meaningful theorem, since it says that even the minimum feasible point of the dual program is an upper bound on the maximum solution of the primal problem. Equivalently, every dual feasible point gives an upper bound on the objective at every primal feasible point. We shall derive the construction of LP duality, keeping in mind the goal of achieving weak duality.

2 The derivation

To simplify the exposition, consider the following maximisation linear program P with 2 variables and 2 constraints:

$$\begin{array}{lll} \text{maximise} & c_1x_1 + c_2x_2 & \\ \text{subject to} & a_{11}x_1 + a_{12}x_2 \leq b_1 & \\ & a_{21}x_1 + a_{22}x_2 \leq b_2 & \\ \text{where} & x_1 \geq 0 & \\ & x_2 \geq 0 & \end{array}$$

Since every dual feasible point is to be an upper bound on the objective at every primal feasible point, we should first come up with inequalities satisfied by the primal points. The only inequalities we have for these points are the *primal constraints*, which are linear inequalities in the primal variables. How can we aggregate the primal constraints into one single bound? Given that they are linear inequalities, the most natural approach is to take their *weighted sum*. That is, we multiply each constraint with a multiplier (in this case, all multipliers are non-negative) and sum them up. Thus, we know that every primal feasible point (x_1, x_2) satisfies the following inequality, for every $y_1, y_2 \geq 0$:

$$y_1(a_{11}x_1 + a_{12}x_2) + y_2(a_{21}x_1 + a_{22}x_2) \leq b_1y_1 + b_2y_2 \tag{1}$$

First, note that y_1 and y_2 *have* to be non-negative. If we multiply both sides of an inequality with a negative number, the direction of the inequality changes, which is undesirable in this case. Second, also note that if we fix the values of y_1 and y_2, the left hand side of the inequality $((a_{11}y_1 + a_{21}y_2)x_1 + (a_{12}y_1 + a_{22}y_2)x_2)$ is a linear function of x_1 and x_2, and the right hand side is a constant.

These are (uncountably) infinitely many inequalities we have just inferred, where $y_1, y_2 \geq 0$ are values that we can pick however we want. Ultimately, though, we only want one inequality that gives an upper bound on the optimal objective value for the primal problem. Equivalently, the inequality should give an upper bound on the objective value on every primal feasible point ((x_1, x_2) satisfying the primal constraints). Which inequality do we want then? We need to first determine for what values of y_1 and y_2 does the inequality actually upper bound the primal objective function. We observe that the inequality is useful only if the left hand side is itself an upper bound on the objective, $c_1x_1 + c_2x_2$, for all non-negative x_1, x_2. What constraint should we impose on y_1, y_2 in order to enforce the condition? Think for a minute before reading the (simple) exercise on the next page.

Exercise 1. Show that $\alpha_1 x_1 + \alpha_2 x_2 \leq \beta_1 x_1 + \beta_2 x_2$ is true for all $x_1, x_2 \geq 0$ if and only if both 1) $\alpha_1 \leq \beta_1$ and 2) $\alpha_2 \leq \beta_2$. Note that all of $\alpha_1, \alpha_2, \beta_1$ and β_2 could be any real value, positive or negative or 0.

With the result in the exercise, we now know that we should enforce, for all variables x_i, the constraint that the coefficient of x_i in Equation 1 should be at least its coefficient c_i in the objective. This ensures that, for every y_1, y_2 satisfying the dual constraints, $b_1 y_1 + b_2 y_2$ is indeed an upper bound on the objective function for every x_1, x_2 in the primal feasible region. The weak duality theorem thus follows from the construction of the dual constraints.

Lastly, going back to our question of which inequality we want, we (obviously) want the tightest upper bound possible. That is, we shall choose to *minimise* $b_1 y_1 + b_2 y_2$, subject to the constraints we just inferred.

Interpreting $\{y_i\}$ as a set of variables, we have now derived the *dual* linear program D of P:

$$\begin{array}{lrcl} \text{minimise} & b_1 y_1 + b_2 y_2 & & \\ \text{subject to} & a_{11} y_1 + a_{21} y_2 & \geq & c_1 \\ & a_{12} y_1 + a_{22} y_2 & \geq & c_2 \\ \text{where} & y_1 \geq 0 & & \\ & y_2 \geq 0 & & \end{array}$$

In general, the number of dual variables is equal to the number of primal constraints, since the dual variables are multipliers to the primal constraints. Furthermore, the number of dual constraints is equal to the number of primal variables, since the dual constraints apply to coefficients of the primal variables in the inequality (Equation 1) we derived.

3 Reconstructing the LP duality table

We are now in a position to re-derive the duality table concerning how 1) primal variable domains are related to the direction of the dual constraint inequalities and 2) the direction of primal constraints are similarly related to the domains of the dual variables. To do so, we shall change parts of the primal and think about how the dual should correspondingly change, such that the reasoning in our previous derivation still holds.

The domain of x_1 is now $x_1 \leq 0$: We can repeat Exercise 1 with $x_1 \leq 0$ by modifying condition 1) to $\alpha_1 \geq \beta_1$. In the context of the dual program, we now have $a_{11} y_1 + a_{21} y_2 \leq c_1$ as our first dual constraint.

The domain of x_1 is now unrestricted: In order to ensure that the left hand side of Equation 1 is an upper bound of the objective, we know that the direction of $\geq$ is "safe" in the first dual constraint when $x_1 \geq 0$, and $\leq$ is similarly "safe" when

$x_1 \leq 0$. Thus, when x_1 is unrestricted, we have to be conservative and enforce both inequalities, resulting in the *equality* $a_{11}y_1 + a_{21}y_2 = c_1$. Again, Exercise 1 can be repeated, modifying the condition to $\alpha_1 = \beta_1$ when x_1 has unrestricted domain.

The first constraint becomes $a_{11}x_1 + a_{12}x_2 \geq b_1$: Recall that we need an upper bound on the primal objective function, thus the derived inequality (Equation 1) should be of the form "linear function of $x_1, x_2 \leq$ some constant". We should therefore multiply this primal constraint with a non-positive dual variable in order to change the sign of the inequality. Thus, the domain of the dual variable y_1 becomes non-positive ($y_1 \leq 0$).

The first constraint becomes $a_{11}x_1 + a_{12}x_2 = b_1$: Equality constraints hold and can be added to any inequality when they are multiplied by any constant, regardless of whether the constant is positive or negative or 0. Thus, the dual variable y_1 can now have an unrestricted domain.

4 Example problem

We now go through a past exam problem, concerning the shortest path problem on graphs with positive or negative edge weights.

Problem 1. Given a directed graph, with positive or negative edge weights but without any negative weight cycles, we know how to find the distance between two nodes s and t via the Bellman-Ford algorithm. However, in this problem your task is to find this distance using linear programming. We strongly suggest you have 1 variable x_i for each node i in the graph, representing the distance from s to i.

1. Given an edge $i \to j$ of length ℓ, describe a constraint on how the two distances x_i and x_j must relate to each other (where x_i represents the length of the shortest path from s to i, and x_j represents the length of the shortest path from s to j).

2. Describe and explain the remaining details of a linear program that, when solved, will have objective function equal to the distance from s to t. (Hint: if there are two ways to get to node i, we want the value of x_i to equal the minimum of these two ways.) You do not have to prove that your linear program computes the correct distance, but describe why its parts make sense.

3. Compute the *dual* of your linear program.

4. Interpret the dual of your linear program and explain **from the dual alone** why the value of its objective function will correctly compute the distance from s to t. (**Hint:** the dual variables correspond to a "flow" and you are trying to minimize its "cost".)

Solution guide/explanation:

You are not expected to write nearly as much for the exam. The following is meant to be significantly more explanatory, and motivates the calculations and derivations.

1. We know that the variable x_i define the shortest distance from s to i. There are a few inequalities concerning distances, and the most important one is the *triangle inequality*. That is, for every edge $i \to j$, we know that $x_j \leq x_i + w_{i\to j}$ where $w_{i\to j}$ is the weight of the edge $i \to j$. The triangle we are considering is defined by s, i and j.

2. The variables of our linear program are x_i for each node i in the graph, with unrestricted domain since there are possibly negative weights and thus the shortest distance may be negative.

   Part 1 already gives us most of the constraints in our linear program. One last constraint we need is that $x_s = 0$, to provide a well-defined reference point in the distances instead of just inequalities.

   As for the objective, we are really only concerned about the distance from s to t, so it is natural to have an objective function that is simply x_t. Here's the more tricky question: is the problem a minimisation or a maximisation problem? Since it is a *shortest* path problem, it is natural to try minimisation. However, minimisation does not work, as illustrated by the following exercise.

   Exercise 2. Suppose we minimise the objective x_t subject to the above constraints. Give an example weighted directed graph (and an s-t pair), and an upper bound to the corresponding minimisation linear program, showing that the minimisation problem clearly does *not* compute the shortest distance from s to t.

   The only alternative then is maximisation. Why does it work? A simple answer is that, by induction (on the number of vertices along a shortest path from s, taking the minimum number if there are multiple possibilities), any feasible solution has x_i is less than or equal to the true shortest distance d_i from s to i. When we maximise x_t, then the maximisation stops exactly when x_t hits the distance from s to t. More formally, $x_i = d_i$ is a feasible point to the maximisation problem, whereas we have also proved that for any feasible point, $x_i \leq d_i$ for all node i. In particular, $x_t \leq d_t$. Thus $x_t = d_t$ is the optimal solution. (Note: a formal proof was not needed for the exam.)

   The full linear program P_G (depending on the graph G) is as follows:

$$
\begin{array}{lrcl}
\text{maximise} & x_t & & \\
\text{subject to} & x_j - x_i & \leq & w_{i\to j} \text{ for all directed edge } i \to j \\
 & x_s & = & 0 \\
\text{where} & \multicolumn{3}{l}{x_i \text{ is unrestricted for all node } i}
\end{array}
$$

As an extra exercise, let us reconsider the assumption that the graph does not have any negative weight cycles. What can we say about the linear program if the assumption does not hold?

Exercise 3. Given a graph G with a negative cycle, in what way does the linear program P_G "break"? What goes wrong?

Here is another extra exercise. What if we did not include the constraint $x_s = 0$? What happens to the optimal solution of the linear program?

Exercise 4. If we removed the constraint $x_s = 0$ from the linear program P_G, what can we say about its optimum?

3. To compute the dual program D_G of P_G, let us start by identifying and naming the dual variables. Recall that there should a dual variable for each primal constraint. We have one primal constraint for each directed edge, and an extra constraint $x_s = 0$. We shall therefore have dual variables $f_{i \to j}$ for each directed edge $i \to j$, and an additional dual variable S.

   Now we can write down the objective of the dual program. Since the primal is a maximisation, the dual becomes a minimisation program. The objective function is the sum, over all primal constraints, of the dual variable multiplied by the constant in the corresponding primal constraint. Thus, in our case, noting that the extra constraint has a constant of 0, the objective function is $\sum_{i \to j} w_{i \to j} f_{i \to j}$.

   As for the domains of the dual variables, for the variable S it is unrestricted, since S corresponds to an equality constraint. The variables $f_{i \to j}$ correspond to $\leq$ constraints, with a linear function of primal variables on the left and a constant on the right. Recall that, since the primal is a maximisation, we want an upper bound of the objective for the dual, and so we wish to preserve the direction of the inequalities. Thus the domains of $f_{i \to j}$ are the non-negative real numbers ($f_{i \to j} \geq 0$).

   We now come to the more difficult (and the only remaining) part of computing the dual: determining the dual constraints. There should be one dual constraint for each primal variable, that is, one dual constraint per node i on the graph. The dual constraints are all equalities, since the primal variables have unrestricted domains. We examine three cases of the dual constraints separately: 1) if i is neither the source s nor the destination t, 2) if i is s and 3) if i is t.

   First consider the general case for a node i, where $s \neq i \neq t$. We wish to compute the coefficient of x_i in the inequality analogous to Equation 1, derived from 1) multiplying each primal constraint corresponding to edge e with the dual variable f_e and 2) summing them up. A primal constraint would contribute to the coefficient of x_i if and only if e is incident to i (and since $i \neq s$, the extra constraint of "$x_s = 0$" does not matter). If the edge e is outgoing from i, namely $e = i \to j$ for some j, then that constraint contributes $-f_{i \to j}$ to x_i's coefficient.

Conversely, if the edge e is $k \to i$ for some k, then the constraint contributes $f_{k\to i}$ to the coefficient. Thus the coefficient of x_i is $(\sum_{k\to i} f_{k\to i} - \sum_{i\to j} f_{i\to j})$. Since the coefficient of x_i in the primal objective is 0 (for $i \neq t$), we derive the constraint that "$\sum_{k\to i} f_{k\to i} - \sum_{i\to j} f_{i\to j} = 0$", or equivalently, "$\sum_{k\to i} f_{k\to i} = \sum_{i\to j} f_{i\to j}$".

Now for the case if node i is actually the source s. The only difference from the general case is that the extra constraint "$x_s = 0$" also contributes to the coefficient of x_s in the derived inequality. Thus the corresponding dual constraint is "$S + \sum_{k\to s} f_{k\to s} = \sum_{s\to j} f_{s\to j}$".

Lastly, for the case if node i is the destination t, the difference from the general case is that the coefficient of x_t in the primal objective is 1 instead of 0, and so we get the dual constraint "$\sum_{k\to t} f_{k\to t} = \sum_{t\to j} f_{t\to j} + 1$".

The dual program D_G is presented in full as follows:

$$
\begin{array}{lll}
\text{minimise} & \sum_{i\to j} w_{i\to j} f_{i\to j} & \\
\text{subject to} & \sum_{k\to i} f_{k\to i} = \sum_{i\to j} f_{i\to j} & \text{for all node } i \text{ where } s \neq i \neq t \\
 & S + \sum_{k\to s} f_{k\to s} = \sum_{s\to j} f_{s\to j} & \text{for node } s \\
 & \sum_{k\to t} f_{k\to t} = \sum_{t\to j} f_{t\to j} + 1 & \text{for node } t \\
\text{where} & f_{i\to j} \geq 0 \text{ for all directed edge } i \to j & \\
 & S \text{ is unrestricted} &
\end{array}
$$

4. To interpret the dual program D_G, we use the hint to interpret the dual variables as flows along the corresponding directed edges. Then, the general constraint on node i, where $s \neq i \neq t$, simply states the physical law that is the *conservation of flow*: the amount of flow into a node (left hand side of the equality) has to be equal to the amount of flow out of a node (right hand side). For the constraint on node s, essentially the same conservation holds, except we also inject a flow of S into the system via node s. Lastly, for node t, the constraint says the amount of flow out of t is the amount of flow into t, with 1 subtracted. That is, a flow of 1 is taken out of node t.

   By the conservation of flow at general nodes, we know that the variable S has to take value 1, meaning that we have to inject exactly a flow of 1 into the system in order to be able to take a flow of 1 out at t. Thus we are injecting a unit of flow into s and directing it to t to be taken out.

   The objective tells us that the cost incurred by a unit of flow is exactly the distance travelled by it. To minimise the objective, we simply direct a unit of flow from s to t through one of the shortest paths. Adding any extra flow (which has to go in cycles essentially) only add to the cost, and directing the flow through non-shortest paths also increases the cost.

   There is some degree of handwaving involved in the justification, but this is sufficient for the exam.

A Solutions to exercises

Exercise 1: "$\Rightarrow$": If $\alpha_1 x_1 + \alpha_2 x_2 \leq \beta_1 x_1 + \beta_2 x_2$ for all $x_1, x_2 \geq 0$, then consider the values of $x_1 = 1$ and $x_2 = 0$. We get $\alpha_1 \leq \beta_1$. Similarly, with $x_1 = 0$ and $x_2 = 1$, we get $\alpha_2 \leq \beta_2$.

"$\Leftarrow$": Consider an arbitrary pair of $x_1, x_2 \geq 0$. Assuming $\alpha_1 \leq \beta_1$, then $\alpha_1 x_1 \leq \beta_1 x_1$ (since $x_1 \geq 0$). Similarly, $\alpha_2 \leq \beta_2$ implies $\alpha_2 x_2 \leq \beta_2 x_2$ (since $x_2 \geq 0$). Summing the two inequalities gives $\alpha_1 x_1 + \alpha_2 x_2 \leq \beta_1 x_1 + \beta_2 x_2$.

Exercise 2: Consider any graph with positive weights only. If we changed the objective from a maximisation to a minimisation, then the optimum is upper bounded by 0. The value 0 is a feasible point, since the assignment of 0 to every single primal variable would satisfy all the triangle inequalities. If the optimum of x_t is upper bounded by 0, then clearly it will not be the shortest distance from s to t in the graph with positive weights.

Exercise 3: Suppose there is a negative cycle $i_1 \to i_2 \to \ldots i_n \to i_1$ in the graph. We claim that there is no feasible solution to the linear program P_G. For the sake of contradiction, assume there is a feasible solution $\{x_i\}$. Recall that P_G imposes the constraints $x_{i_{k+1}} - x_{i_k} \leq w_{i_k \to i_{k+1}}$ and $x_{i_1} - x_{i_n} \leq w_{i_n \to i_1}$. Summing these inequalities gives $0 \leq w$, where w is the weight of the negative cycle. By definition, w is strictly negative, and so $0 \leq w$ is a contradiction.

Exercise 4: If we removed the constraint $x_s = 0$, then the maximisation problem P_G is unbounded. To show this, consider an arbitrarily large constant c. Let d_i be the actual shortest distance from s to i. Clearly, $x_i = d_i$ satisfies all the triangle inequalities. Now consider the solution $x'_i = d_i + c$. For each directed $i \to j$, we have $x'_j - x'_i = d_j + c - d_i - c = d_j - d_i \leq w_{i \to j}$, and so $\{x'_i\}$ is also a feasible solution if we only had the triangle inequality constraints. The objective value of this solution is $d_t + c$. Since c could be made arbitrarily large for the above argument, the optimum of the maximisation is unbounded.