6.854 Advanced Algorithms

Lecture 1: 10/20/2006 Lecturer: David Karger

Scribes: Jay Kumar Sundararajan, and Aaron Bernstein

Duality

This lecture covers weak and strong duality, and also explains the rules for finding the dual of a linear program, with an example. Before we move on to duality, we shall first see some general facts about the location of the optima of a linear program.

1.1 Structure of LP solutions

1.1.1 Some intuition in two dimensions

Consider a linear program -

Maximize $y^T b$
subject to $y^T A \leq c$

The feasible region of this LP is in general, a convex polyhedron. Visualize it as a polygon in 2 dimensions, for simplicity. Now, maximizing $y^T b$ is the same as maximizing the projection of the vector y in the direction represented by vector b. For whichever direction b we choose, the point y that maximizes $y^T b$ cannot lie strictly in the interior of the feasible region. The reason is that, from an interior point, we can move further in any direction, and still be feasible. In particular, by moving along b, we can get to a point with a larger projection along b. This intuition suggests that the optimal solution of an LP will never lie in the interior of the feasible region, but only on the boundaries. In fact, we can say more. We can show that for any LP, the optimal solutions are always at the "corners" of the feasible region polyhedron. This notion is formalized in the next subsection.

1.1.2 Some definitions

Definition 1 (Vertex of a Polyhedron) *A point in the polyhedron which is uniquely optimal for some linear objective, is called a vertex of the polyhedron.*

Definition 2 (Extreme Point of a Polyhedron) *A point in the polyhedron which is not* *a convex combination of two other points in the polyhedron is called an extreme point of the polyhedron.*

Definition 3 (Tightness) *A constraint of the form $a^T x \leq b$, $a^T x = b$ or $a^T x \geq b$ in a linear program is said to be tight for a certain point y, if $a^T y = b$.*

Definition 4 (Basic Solution) *For an n-dimensional linear program, a point is called a basic solution, if n linearly independent constraints are tight for that point.*

Definition 5 (Basic Feasible Solution) *A point is a basic feasible solution, iff it is a basic solution that is also feasible.*

Note: If x is a basic feasible solution, then it is in fact, the unique point that is tight for all its tight constraints. This is because, there can be only one solution for a set of n linearly independent equalities, in n-dimensional space.

Theorem 1 *For a polyhedron P and a point $x \in P$, the following are equivalent:*

1. *x is a basic feasible solution*
2. *x is a vertex of P*
3. *x is an extreme point of P*

Proof: Assume the LP is in the canonical form.

1. **Vertex⇒ Extreme Point**
   Let v be a vertex. Then for some objective function c, $c^T x$ is uniquely minimized at v. Assume v is not an extreme point. Then, v can be written as $v = \lambda y + (1 - \lambda) z$ for some y, z neither of which is v, and some λ satisfying $0 \leq \lambda \leq 1$.

   Now, $c^T v = c^T [\lambda y + (1 - \lambda) z] = \lambda c^T y + (1 - \lambda) c^T z$

   But note that since $c^T x$ is uniquely minimized at v, we must have $c^T y > c^T v$ and $c^T z > c^T v$. Thus, $c^T v = \lambda c^T y + (1 - \lambda) c^T z > c^T v(\lambda + 1 - \lambda) = c^T v$. This is clearly a contradiction.

2. **Extreme Point ⇒ Basic Feasible Solution**
   Let x be an extreme point. By definition, it lies in the polyhedron and is therefore feasible. Assume x is not a basic solution. Let T be the set of rows of the constraint matrix A for which the constraints are tight at x. Let a_i (a $1 \times n$ vector) denote the

i^{th} row of A. For $a_i \notin T$, $a_i.x > b_i$. Since x is not a basic solution, T does not span $\mathcal{R}^n$. So, there is a vector $d \neq 0$ such that $a_i.d = 0 \ \forall a_i \in T$.

Consider $y = x + \epsilon d$ and $z = x - \epsilon d$. If $a_i \in T$, then $a_i.y = a_i.z = b_i$. If $a_i \notin T$, then, by choosing a sufficiently small ϵ: $0 < \epsilon \leq min_{i \notin T} \frac{a_i.x - b_i}{|a_i.d|}$, we can ensure that $a_i.y \geq b_i$ and $a_i.z \geq b_i$. Thus y and z are feasible. Since $x = y/2 + z/2$, x cannot be an extreme point – a contradiction.

3. **Basic Feasible Solution ⇒ Vertex**
Let x be a basic feasible solution. Let $T = \{i \mid a_i.x = b_i\}$. Consider the objective as minimizing $c.y$ for $c = \sum_{i \in T} a_i$. Then, $c.x = \sum_{i \in T}(a_i.x) = \sum_{i \in T} b_i$.
For any $x' \in \mathcal{P}$, $c.x' = \sum_{i \in T}(a_i.x') \geq \sum_{i \in T} b_i$ with equality only if $a_i.x' = b_i \ \forall i \in T$. This implies that $x' = x$ and that x uniquely minimizes the objective $c.y$.

This proves that vertex, extreme point and basic feasible solution are equivalent terms. ■

Theorem 2 *Any bounded LP in standard form has an optimum at a basic feasible solution.*

Proof: Consider an optimal x which is not a basic feasible solution. Being optimal, it is feasible, hence it is not basic. As in the previous proof, let T be the set of rows of the constraint matrix A for which the constraints are tight at x. Since x is not a basic solution, T does not span $\mathcal{R}^n$. So, there is a vector $d \neq 0$ such that $a_i.d = 0 \ \forall a_i \in T$. For a scalar ϵ with sufficiently small absolute value, $y = x + \epsilon d$ is feasible, and represents a line containing x in the direction d. The objective function at y is $c^T x + \epsilon c^T d$. Since x is optimal, $c^T d = 0$, as otherwise, an ϵ of the opposite sign can reduce the objective. This means, all feasible points on this line are optimal. One of the directions of motion on this line will reduce some x_i. Keep going till some x_i reduces to 0. This results in one more tight constraint than before.

This technique can be repeated, till the solution becomes basic. ■

Thus, we can convert any feasible solution to a basic feasible solution of no worse value. In fact, this proof gives an algorithm for solving a linear program: evaluate the objective at all basic feasible solutions, and take the best one. Suppose there are m constraints and n variables. Since a set of n constraints defines a basic feasible solution, there can be upto $\binom{m}{n}$ basic feasible solutions. For each set of n constraints, a linear system of inequalities has to be solved, which by Gaussian elimination, takes $O(n^3)$ time. This is in general an exponential complexity algorithm in n. Note that the output size is polynomial in n, since the optimal solution is just the solution of a system of linear equalities.

1.2 The dual of a linear program

Given an LP in the standard form:

$$\text{Minimize } c.x$$
$$\text{subject to: } Ax = b; x \geq 0$$

We call the above LP the primal LP. The decision version of the problem is: Is the optimum $c.x \leq \delta$? This problem is in NP, because, if we find a feasible solution with optimum value $\leq \delta$, we can verify that it satisfies these requirements, in polynomial time. A more interesting question is whether this problem is in *co-NP*. We need to find an easily verifiable proof for the fact that there is no x which satisfies $c.x < \delta$. To do this, we require the concept of duality.

1.2.1 Weak Duality

We seek a lower bound on the optimum. Consider a vector y (treat is as a row vector here). For any feasible x, $yAx = yb$ holds. If we require that $yA \leq c$, then $yb = yAx \leq cx$. Thus, yb is a lower bound on cx, and in particular on the optimum cx. To get the best lower bound, we need to maximize yb. This new linear program:

$$\text{Maximize } yb$$
$$\text{subject to: } yA \leq c$$

is called the *dual linear program.* (Note: The dual of a dual program is the primal). Thus primal optimum is lower bounded by the dual optimum. This is called *weak duality.*

Theorem 3 (Weak Duality) *Consider the LP* $z = Min\{c.x \mid Ax = b, x \geq 0\}$ *and its dual* $w = max\{y.b \mid yA \leq c\}$. *Then* $z \geq w$.

Corollary 1 *If the primal is feasible and unbounded, then the dual is infeasible.*

1.3 Strong Duality

In fact, if either the primal or the dual is feasible, then the two optima are equal to each other. This is known as *strong duality.* In this section, we first present an intuitive explanation of the theorem, using a gravitational model. The formal proof follows that.

1.3.1 A gravitational model

Consider the LP $\min\{y.b|yA \geq c\}$. We represent this feasible region as a hollow polytope, with the vector b pointing "upwards". If a ball is dropped into the polytope, it will settle down at the lowest point, which is the optimum of the above LP. Note that any minimum is a global minimum, since the feasible region of an LP is a convex polyhedron. At the equilibrium point, there is a balance of forces – the gravitational force and the normal reaction of the floors (constraints). Let x_i represent the amount of force exerted by the i^{th} constraint. The direction of this force is given by the i^{th} column of A. Then the total force exerted by all the constraints Ax balances the gravity b: $Ax = b$.

The physical world also gives the constraints that $x \geq 0$, since the floors' force is always outwards. Only those floors which the ball touches exert a force. This means that for the constraints which are not tight, the corresponding x_i's are zero: $x_i = 0$ if $yA_i > c_i$. This can be summarized as

$$(c_i - yA_i)x_i = 0$$

. This means x and y satisfy:

$$y.b = \sum yA_ix_i = \sum c_ix_i = c.x$$

But weak duality says that $yb \leq cx$, for every x and y. Hence the x and y are the optimal solutions of their respective LP's. This implies strong duality – the optima of the primal and dual are equal.

1.3.2 A formal proof

Theorem 4 (Strong Duality) *Consider $w = min\{y.b \mid yA \geq c\}$ and $z = min\{c.x \mid Ax = b, x \geq 0\}$. Then $z = w$.*

Proof: Consider the LP $\min\{y.b|yA \geq c\}$. Consider the optimal solution y^*. Without loss of generality, ignore all the constraints that are loose for y^*. If there are any redundant constraints, drop them. Clearly, these changes cannot alter the optimal solution. Dropping these constraints leads to a new A with fewer columns and a new shorter c. We will prove that the dual of the new LP has an optimum equal in value to the primal. This dual optimal solution can be extended to an optimal solution of the dual of the original LP, by filling in zeros at places corresponding to the dropped constraints. The point is that we do not need those constraints to come up with the dual optimal solution.

After dropping those constraints, at most n tight constraints remain (where n is the length of the vector y). Since we have removed all redundancy, these constraints are linearly independent. In terms of the new A and c, we have new constraints $yA = c$. y^* is still the optimum.

Claim: There exists an x, such that $Ax = b$.
Proof: Assume such an x does not exist, i.e. $Ax = b$ is infeasible. Then "duality" for linear equalities implies that there exists a z such that $zA = 0$, but $zb \neq 0$. Without loss of generality, assume $z.b < 0$ (otherwise, just negate the z). Now consider $(y^* + z)$. $A(y^* + z) = Ay^* + Az = Ay^*$. Hence, it is feasible. $(y^* + z).b = y^*.b + z.b < y^*.b$, which is better than the assumed optimum – a contradiction. So, there is an x such that $Ax = b$. Let this be called x^*.

Claim: $y^*.b = c.x^*$.
Proof: $y^*.b = y^*.(Ax^*) = (y^*A).x^* = c.x^*$ (since $Ax^* = b$ and $y^*A = c$)

Claim: $x^* \geq 0$
Proof: Assume the contrary. Then, for some i, $x_i^* < 0$. Let $c' = c + e_i$, where e_i is all 0's except at the i^{th} position, where it has a 1. Since A has full rank, $yA \geq c'$ has a solution, say y'. Besides, since $c' \geq c$, y' is feasible for the original constraints $yA \geq c$. But, $y'.b = y'Ax^* = c'x^* < cx^* = y^*b$ (since c'_i is now higher and $x_i < 0$). This means y' gives a better objective value than the optimal solution – a contradiction. Hence, $x^* \geq 0$.

Thus, there is an x^* which is feasible in the dual, and whose objective is equal to the primal optimum. Hence, x^* must be the dual optimal solution, using weak duality. Thus, the optima of primal and dual are equal.

■

Corollary 2 *Checking for feasibility of a linear system of inequalities and optimizing an LP are equally hard.*

Proof: Optimizer → Feasibility checker
Use the optimizer to optimize any arbitrary function with the linear system of inequalities as the constraints. This will automatically check for feasibility, since every optimal solution is feasible.

Feasibility checker → Optimizer
We construct a reduction from the problem of finding an optimal solution of LP_1 to the problem of finding a feasible solution of LP_2. LP_1 is $min\{c.x \mid Ax = b, x \geq 0\}$. Consider $LP_2 = min\{0.x|Ax = b, x \geq 0, yA \leq c, c.x = b.y\}$. Any feasible solution of LP_2 gives an optimal solution of LP_1 due to the strong duality theorem. Finding an optimal solution is thus no harder than finding a feasible solution. ■

1.4 Rules for duals

Usually the primal is constructed as a minimization problem and hence the dual becomes a maximization problem. For the standard form, the primal is given by:

$$\begin{aligned} z &= \min\ (c^T x) \\ Ax &\geq b \\ x &\geq 0 \end{aligned}$$

while the dual is given by:

$$\begin{aligned} w &= \max\ (b^T y) \\ A^T y &\leq c \\ y &\geq 0 \end{aligned}$$

For a mixed form of the primal, the following describes the dual:

Primal:

$$\begin{aligned} z &= \min\ c_1x_1 + c_2x_2 + c_3x_3 \\ A_{11}x_1 + A_{12}x_2 + A_{13}x_3 &= b_1 \\ A_{21}x_1 + A_{22}x_2 + A_{23}x_3 &\geq b_2 \\ A_{31}x_1 + A_{32}x_2 + A_{33}x_3 &\leq b_3 \\ x_1 &\geq 0 \\ x_2 &\leq 0 \\ x_3 & \quad \text{UIS} \end{aligned}$$

(UIS = unrestricted in sign)

Dual:

$$\begin{aligned} w &= \max\ y_1b_1 + y_2b_2 + y_3b_3 \\ y_1A_{11} + y_2A_{21} + y_3A_{31} &\leq c_1 \\ y_1A_{12} + y_2A_{22} + y_3A_{32} &\geq c_2 \\ y_1A_{13} + y_2A_{23} + y_3A_{33} &= c_3 \end{aligned}$$

$$\begin{array}{lcl} y_1 & & \text{UIS} \\ y_2 & \geq & 0 \\ y_3 & \leq & 0 \end{array}$$

These rules are summarized in the following table.

PRIMAL	Minimize	Maximize	DUAL
Constraints	$\geq b_i$	≥ 0	Variables
	$\leq b_i$	≤ 0	
	$= b_i$	Free	
Variables	≥ 0	$\leq c_j$	Constraints
	≥ 0	$\leq c_j$	
	Free	$= c_j$	

Each variable in the primal corresponds to a constraint in the dual, and vice versa. For a maximization, an upper bound constraint is a “natural” constraint, while for a minimization, a lower bound constraint is natural. If the constraint is in the natural direction, then the corresponding dual variable is non-negative.

An interesting observation is that, the tighter the primal gets, the looser the dual gets. For instance, an equality constraint in the primal leads to an unrestricted variable in the dual. Adding more constraints in the primal leads to more variables in the dual, hence more flexibility. This makes sense because the dual bounds the primal, so since adding more constraints will lead to a worse optimum in the primal, this will be a reflected by a better optimum in the dual.

1.5 Shortest Path – an example

Consider the problem of finding the shortest path in a graph. Given a graph G, we wish to find the shortest path from a specified source node, to all other nodes. The following linear program finds the shortest distance from s to t:

$$w = \max \ (d_t - d_s)$$

$$\text{s.t. } d_j - d_i \leq c_{ij}, \qquad \forall i, j$$

In this formulation, d_i represents the distance of node i from the source node s. The c_{ij} constraints are essentially the triangle inequalities – the distance from the source to a node i should not be more than the distance to some node j plus the distance from j to i. Intuitively, one can imagine stretching the network physically, to increase the source-destination distance. When we cannot pull any further without breaking an edge, we have found a shortest path.

More formally, let p be a shortest path from s to t, and let δ(t) be the length of this path. We must have d(t) $\geq \delta$(t) because we are trying to maximize d(t), and setting d(i) = δ(i) for every vertex on p is obviously a feasible solution. Now, I will show that we cannot have d(t) > δ(t). For let v be the first vertex on p such that $d(v) > \delta(v)$, and let u be its predecessor. p is a shortest path to v, so $\delta(v) = \delta(u) + c_{u,v} \geq d(u) + c_{u,v} \geq d(v)$, contradictory to our assumption.

The dual to this program is found thus. The constraint matrix in the primal has a row for every pair of nodes (i, j), and a column for every node. The row corresponding to (i, j) has a +1 in the i^{th} column and a -1 in the j^{th} column, and zeros elsewhere.

1. Using this, we conclude that the dual has a variable for each pair (i, j), say y_{ij}.
2. It has a constraint for each node i. The constraint has a coefficient of +1 for each edge entering node i and a -1 for each edge leaving i. The right side for the constraints are -1 for the node s constraint, 1 for the node t constraint, and 0 for others, based on the objective function in the primal. Moreover, all the constraints are equality constraints, since the d_i variables were unrestricted in sign in the primal.
3. The dual variables will have to have a non-negativity constraint as well, since the constraints in the primal were "natural" (upper bounds for a maximization).
4. The objective is to minimize $\sum_{i,j} c_{ij} y_{ij}$, since the right side of the primal constraints are c_{ij}.

Thus the dual is:

$$z = \min \sum_{i,j} c_{ij} y_{ij}$$

$$\begin{aligned}
\sum_j (y_{js} - y_{sj}) &= -1 \\
\sum_j (y_{jt} - y_{tj}) &= 1 \\
\sum_j (y_{ji} - y_{ij}) &= 0, \forall i \neq s, t \\
y_{ij} &\geq 0, \forall i, j
\end{aligned}$$

This is precisely the linear program to solve the minimum cost unit flow, in a gross flow formulation. The constraints correspond to the flow conservation at all nodes except at the source and sink. The value of the flow is forced to be 1. This makes sense intuitively because if we treat edge weights as cost, then the shortest path is precisely the cheapest way to send a unit of flow.

1.6 Max Flow – Another Example

Consider the problem of finding a maximum flow from a source s to a sink t in a graph G. In order to formulate this as a LP, we will add an infinite capacity edge from t to s. Our objective is then to maximize the flow on this edge while perserving conservation constraints, capacity constraints, and the non-negative flow constraint. Think of non-existing edges as having capacity 0.

Let $\mathrm{x}_{v,w}$ be the flow value on edge (v,w). The linear program is then:

Primal: maximize $\mathrm{x}_{t,s}$ while preserving
$\mathrm{x}_{v.w} \geq 0 \;\; \forall(v,w)$ (a non-negative flow value for each edge)
$\mathrm{x}_{v,w} \leq u_{v,w} \;\; \forall(v,w)$ (capacity constraints)
$\sum_w x_{v,w} - x_{w,v} = 0 \;\; \forall v$ (conservation constraints).

What is the dual of this linear program? Well, we have two types of constraints. We have a constraint for each vertex v, which will become a variable z_v in the dual. We also have capacity constraints for each edge (v,w), which will become variables $\mathrm{y}_{v,w}$. Note that the conservation constraints are equalities, so by the cookbook, z_v is unrestricted in sign (∀ v). Capacity constraints are in the "natural direction", so we must have $\mathrm{y}_{v,w} \geq 0$ ∀ (v,w).

What are the constraints of the dual? The LHS (left hand side) of the constraints is derived from the primal variables, so there will be a constraint for each edge. To find the constraints, we must examine the constraint matrix in the primal. In the primal, the rows correspond to constraints, while the columns corresond to variables. In the dual, we will transpose this matrix, so the columns will now correspond to constraints.

There are m + n rows in the constraint matrix: m for the capacity constraints ($\mathrm{y}_{v,w}$ in the dual), and n for the circulation constraints (z_v in the dual). There are m columns: one for each variable $\mathrm{x}_{v,w}$. The capacity constraint corresponding to an edge (v,w) is simple: there is a 1 in the $\mathrm{x}_{v,w}$ column, and a zero everywhere else. The circulation constraint for a node v is $\sum_w x_{v,w} - x_{w,v} = 0$, so we will have a one in every column $\mathrm{x}_{v,w}$, a -1 in every column $\mathrm{x}_{w,v}$, and a zero everywhere else.

If we transpose this matrix, we get that a column $\mathrm{x}_{v,w}$ has a 1 in the row z_v, a -1 in the row z$_w$, a 1 in the row y$_{v,w}$, and a zero everywhere else. Thus, the LHS of a constraint (in the dual) corresponding to x$_{v,w}$ is just z$_v$ - z$_w$ + y$_{w,v}$. The direction of the inequality is $\geq$ because we had x$_{i,j}$ ≥ 0 (see cookbook).

Now, we must find the value on the RHS. To do this, we look at the objective function. For (v,w) $\neq$ (s,t), x$_{v,w}$ is not even in the objective, so we have a 0. For (v,w) = (t,s), we have a 1.

Finally, the objective of the dual is y$_1$b$_1$ + y$_2$b$_2$ + y$_3$b$_3$. The y's are variables in the dual (y$_1$ UIS, y$_2$ ≥ 0, y$_3$ ≤ 0), while the b's are constraints in the primal (b$_1$ =, b$_2$ $\geq$, b$_3$ $\leq$). In our example, b$_2$ (the capacity constraints) are the only non-zero constraints, and so our objective is y$_2$b$_2$ = $\sum y_{v,w}u_{v,w}$. Thus, we have:

Dual: Minimize $\sum u_{v,w}y_{v,w}$, while preserving
y$_{v,w}$ ≥ 0
z$_v - z_w + y_{v,w} \geq 0 \;\; \forall(v,w) \neq (t,s)$
z$_v - z_w + y_{v,w} \geq 1$ for (v,w) = (t,s).

Note that u$_{t,s}$ = ∞, so an optimal solution to the dual will have y$_{t,s}$ = 0. By using this, and moving some equations around, we get the constraints:
y$_{v,w}$ ≥ 0
z$_w \leq z_v + y_{v,w} \;\; \forall(v,w) \neq (t,s)$
z$_t - z_s \geq 1$

We can interpret the dual in the following way. View u$_{v,w}$ as the cross-sectional area of (v,w), and y$_{v,w}$ as the length of (v,w). Our goal is to minize the total volume, while maintaining a distance of at least 1 between s and t.

Duality is a very useful concept, especially because it helps to view the optimization problem on hand from a different perspective, which might be easier to handle.