Outline:

- LP Duality
- Weak and Strong Duality
- Dual-Fitting
- Primal-Dual Schema

This pair of lectures covers linear programming duality and its applications. We start by defining the dual of a linear program, present complementary slackness conditions, and establish the easily proved fact of weak duality. We then also present the main ingredients of the proof of strong duality. In the second half of this document, we present applications of duality, including dual-fitting—a method for analyzing approximation algorithms—and the primal-dual method—a method for designing algorithms (which are often combinatorial) using the structure of primal and dual LPs. Some useful references for the material covered in this lecture on the geometry of LP are [Goe94, WS11].

1 LP Duality

Consider the following linear program, which we refer to as the primal LP:

$$\begin{array}{lrcl} \min & 3x_1 + 2x_2 + 8x_3 & & \\ \text{s.t.} & x_1 - x_2 + 2x_3 & \geq & 5 \\ & x_1 + 2x_2 + 4x_3 & \geq & 10 \\ & x_1, x_2, x_3 & \geq & 0 \end{array}$$

Let OPT_P be the optimal value. By definition $\text{OPT}_\text{P} \geq 3x_1 + 2x_2 + 8x_3$, for every $x_1, x_2, x_3 \in P$, where P denotes the polytope of feasible solutions, i.e. the set of values x_1, x_2, and x_3 that satisfy the three ineqaulities. By adding the first two inequalities, we arrive at

$$2x_1 + x_2 + 6x_3 \geq 15.$$

Since $x_1, x_2, x_3 \geq 0$, we get:

$$\text{OPT}_\text{P} \geq 3x_1 + 2x_2 + 8x_3 \geq 2x_1 + x_2 + 6x_3 \geq 15.$$

We can use another linear program to find the largest possible lowerbound one can obtain for OPT_P in such a manner, i.e. by using linear combinations of the constraints on x:

$$\begin{array}{lrcl} \max & 5y_1 + 10y_2 & & \\ \text{s.t.} & y_1 + y_2 & \leq & 3 \\ & -y_1 + 2y_2 & \leq & 2 \\ & 2y_1 + 4y_2 & \leq & 8 \\ & y_1, y_2 & \geq & 0 \end{array}$$

This is called the dual LP, and we use $\mathrm{OPT_D}$ to denote its optimal value. Here y_i denotes the coefficient of the i-th constraint in the linear combination. It is natural to ask how tight this lowerbound is. The strong duality theorem states that if the primal LP is feasible and bounded, then the $\mathrm{OPT_D}$ is exactly equal to $\mathrm{OPT_P}$.

1.1 Weak Duality

Weak duality states that $\mathrm{OPT_D} \leq \mathrm{OPT_P}$. This is not surprising since we designed the dual LP to provide a lowerbound for the primal LP. Let us define the primal and dual formally:

$$
\begin{array}{llll|llll}
P: & & & & D: & & & \\
\min & c^T x & & & \max & b^T y & & \\
\text{s.t.} & Ax & \geq & b & \text{s.t.} & A^T y & \leq & c \\
 & x & \geq & 0 & & y & \geq & 0
\end{array}
$$

Let x and y be any feasible solutions to P and D. We show that $c^T x \geq b^T y$:

$$c^T x \geq y^T A x \geq y^T b.$$

Where the first inequality follows from $x \geq 0$ and $A^T y \leq c$, and the second inequality follows from $y \geq 0$ and $Ax \geq b$.

1.2 Strong Duality

We now establish the strong duality for linear programs. A crucial ingredient of the proof is Farkas Lemma, which is presented in different incarnations. The lemma essentially provides a certificate of infeasibility for a set of linear inequalities. It uses a fact referred to as the *Separating Hyperplane Theorem*, which captures the intuitive claim that for any convex and closed body K and a point $x \notin K$, there is a hyperplane that separates K from x. We will use the Separating Hyperplane Theorem without proof.

Lemma 1 (Farkas Lemma)**.** For any $m \times n$ real matrix A', $m \times 1$ vector b', exactly one of the following two holds:

1. There exists an x' such that $A'x' \geq b'$.
2. There exists $y' \geq 0$ such that $A'^T y' = 0$ and $b'^T y' > 0$.

Proof. We consider the easy direction first. Suppose both the conditions hold. Then, we have the contradiction

$$0 < b'^T y' \leq x'^T A'^T y' \leq 0.$$

Suppose there is no x' such that $A'x' \geq b'$. Then, consider the convex and closed body $K = \{A'x' - s : x' \in \Re^n, s \in \Re^m, s \geq 0\}$. We first note that b' does not belong to this body, since otherwise there exist x', $s \geq 0$ such that $A'x' - s = b'$, implying that $A'x' \geq b'$, contradicting our assumption. Therefore, there is a hyperplane separating b' from K. That is, there exists $y' \neq 0$ such that $y'^T b' > 0$ and $y'^T A' x' \leq y'^T s$ for every $x' \in \Re^n, s \in \Re^m, s \geq 0$. By setting s to 0 and considering different values of x' (since the claim holds for all $x' \in \Re^n$, we can obtain $A'^T y' = 0$. $\square$

Farkas' Lemma can be used to prove the strong duality theorem for LPs. Let the primal and dual LPs be the following.

$$\begin{array}{lll} P: & \min c^T x & \text{subject to } Ax \geq b; x \geq 0 \\ D: & \max b^T y & \text{subject to } A^T y \leq c; y \geq 0 \end{array}$$

Theorem 1. If the primal P and dual D are both feasible, then the optimal value z^* of the primal equals the optimal value w^* of the dual.

Proof. Let x^* and y^* denote, respectively, optimal solutions for P and D. By weak duality, $z^* \geq w^*$. We now show that $z^* \leq w^*$. The proof is by contradiction. If $z^* > w^*$, then there does not exist a y such that

$$A^T y \leq c; y \geq 0; b^T y \geq z^*.$$

We apply Farkas Lemma with the following substitutions.

$$A' = \begin{pmatrix} -A^T \\ I \\ b^T \end{pmatrix} \; b' = \begin{pmatrix} -c \\ 0 \\ z^* \end{pmatrix} \; x' = y; \; y' = \begin{pmatrix} x \\ \delta \\ \lambda \end{pmatrix}.$$

It follows from Farkas Lemma that there exists an y' of the above form such that $y' \geq 0$, $A'^T y' = 0$, and $b'^T y' > 0$. This implies that there exists an $x \geq 0$, $\lambda \geq 0$ such that $Ax = \lambda b$ and $c^T x < \lambda z^*$.

We consider two cases. First, if $\lambda = 0$, then we have found an $x \geq 0$ such that $Ax = 0$ and $c^T x < 0$. This implies that $x^* + x$ is a feasible solution with $c^T(x^* + x) < c^T x^*$, contradicting the optimality of x^*. Second, if $\lambda > 0$, we can assume by scaling x appropriately that $\lambda = 1$, which implies that x satisfies $x \geq 0$, $Ax \geq b$, and $c^T x < z^*$, again contradicting the optimality of x^*. $\square$

1.3 Max-Flow Min-Cut Theorem

As an example, let us apply the concept of duality to the max-flow problem. Given two vertices s and t, the source and the sink, the goal is to find a flow over the edges such that for every vertex except s and t the incoming and outgoing flow are equal, each edge (u, v) has flow at most $c(u, v)$, and the flow going from s to t is maximized. Below is a natural LP that expresses this problem, we use $\delta(u)$ to denote the neighbors of u, and variable $f(u, v)$ to represent the flow going from u to v.

$$\begin{array}{llcll} \max & \sum_{v \in \delta(s)} f(u,v) & & & \\ \text{s.t.} & \sum_{u \in \delta(v)} f(u,v) & = & \sum_{u \in \delta(v)} f(v,u) & \forall v \in V \\ & f(u,v) & \leq & c(u,v) & \\ & f(u,v) & \geq & 0 & \end{array}$$

While this perfectly formulates the max-flow problem, working with the dual can be hard. Instead, we use another linear program that sends flows over paths and we obtain the dual. We use $\mathcal{P}$ to denote the set of s-t

paths:

$$
\begin{array}{lllll}
P: & & & & \\
\max & \sum_{p\in\mathcal{P}} x_p & & & \\
\text{s.t.} & \sum_{p\ni e} x_p & \le & c_e & \forall e\in E \\
 & x_p & \ge & 0 &
\end{array}
\quad\Bigg|\quad
\begin{array}{lllll}
D: & & & & \\
\min & \sum_e y_e\cdot c_e & & & \\
\text{s.t.} & \sum_{e\in p} y_e & \ge & 1 & \forall p\in\mathcal{P} \\
 & y_e & \ge & 0 &
\end{array}
$$

Now, we study the dual program. First, notice that if y was restricted to have 0-1 values, then the resulting integer program would clearly formulate the min-cut problem, where $y_e = 1$ denotes that e is in the cut. One can also prove that the relaxed LP where y can take any nonnegative value (i.e. D above) is defined over an integral polytope (the proof is nontrivial). This would imply that OPT_D is exactly equal to the cost of min-cut. Then, strong duality implies that max-flow is equal to min-cut, while weak duality only implies that max-flow is at most as large as min-cut.

2 Dual Fitting

Generally, duality can be used to analyze approximation algorithms as follows. Let us say we have a minimization problem with optimal value OPT. We then obtain a primal LP which is a relaxation of our problem, i.e. $\text{OPT} \ge \text{OPT}_\text{P}$. Then, we obtain the dual LP and weak duality implies $\text{OPT}_\text{P} \ge \text{OPT}_\text{D}$. Consequently, if we prove that our algorithm's solution, ALG, satisfies $\text{ALG} \le \alpha\cdot\text{OPT}_\text{D}$, we can conclude that our algorithm is an α-approximation. Because:

$$\text{ALG} \le \alpha\cdot\text{OPT}_\text{D} \le \alpha\cdot\text{OPT}_\text{P} \le \alpha\cdot\text{OPT}.$$

Dual fitting is a strategy where one obtains a solution to the problem along with a corresponding set of values y. The values y are defined in the same domain as the dual variables. The algorithm's solution has cost b^Ty, where b^Ty is the objective function of the dual program. However, y is not necessarily feasible in the dual program. A feasible solution y' is then derived from y such that $\alpha\cdot b^Ty' \ge b^Ty$, for some $\alpha \ge 1$ (y is *fitted* to the dual program). This implies that the solution is an α approximation as follows:

$$\text{ALG} = b^Ty \le \alpha\cdot b^Ty' \le \alpha\cdot\text{OPT}_\text{D} \le \alpha\cdot\text{OPT}_\text{P} \le \alpha\cdot\text{OPT}.$$

2.1 Set Cover

As an example, we consider the set cover problem. Given a universe of elements and $\mathcal{U}$, a collection of subsets $\mathcal{S}$, and a cost function c over $\mathcal{S}$, the goal is to choose a subcollection S of $\mathcal{S}$ that covers $\mathcal{U}$ (i.e. $\bigcup_{s\in S} s = \mathcal{U}$) and the cost of S is minimized.

First, we give an algorithm. The algorithm starts with the empty collection, and in every step, it adds a set until all elements are covered. Let us say that the algorithm has so far picked a collection S. For a set $s\in\mathcal{S}$, its density is defined as its cost divided by the number of new elements it covers, that is:

$$\frac{c(s)}{\#\text{ elements in } s \text{ not covered by } S}.$$

The algorithm picks the set s with minimum density in each step, breaking ties arbitrarily.

To analyze the algorithm we obtain the primal and dual LPs:

$$\begin{array}{rlll} P: & & & \\ \min & \sum_{s\in\mathcal{S}} c(s)\cdot x_s & & \\ \text{s.t.} & \sum_{s\ni u} x_s & \geq 1 & \forall u\in\mathcal{U} \\ & x_s & \geq 0 & \end{array} \qquad \Bigg| \qquad \begin{array}{rlll} D: & & & \\ \max & \sum_{u\in\mathcal{U}} y_u & & \\ \text{s.t.} & \sum_{u\in s} y_u & \leq c(s) & \forall s\in\mathcal{S} \\ & y_u & \geq 0 & \end{array}$$

Note that restricting x to 0-1 values would exactly formulate the problem, therefore OPT $\geq$ OPT$_\text{P}$. We provide a set of values y that has the same cost as our algorithm's solution. When a set s is added to the collection and has density d at the time. we let $y_u = d$ for all newly covered elements u. This way, the sum of y_u's grows by $c(s)$, and when the algorithm terminates, $\sum_{u\in\mathcal{U}} y_u$ is equal to the cost of the collection.

To complete the analysis, we need to show $\frac{1}{\alpha}y$ is feasible in the dual LP for some (hopefully small) value $\alpha \geq 1$. Take any set $s \in \mathcal{S}$, we consider the left-hand side in the dual constraint corresponding to s, i.e. the left-hand side of

$$\sum_{u\in s} y_u \leq c(s).$$

Let $u_1, u_2 \dots, u_k$ be the elements of s in the order they were covered by the algorithm (breaking ties arbitrarily). Take any of these elements u_i. At the time u_i is covered, s has at least $k-i+1$ uncovered elements. Hence, it has density at most $\frac{c(s)}{k-i+1}$. Since the algorithm picks the set with the lowest density, the chosen set that covers u_i must have density at most $\frac{c(s)}{k-i+1}$. Therefore, it holds that $y_{u_i} \leq \frac{c(s)}{k-i+1}$. Thus we have:

$$\sum_{u\in s} y_u \leq \sum_{i=1}^{k} \frac{c(s)}{k-i+1} = c(s)\cdot H(k) = c(s)\cdot\Theta(\ln|s|).$$

Therefore, if we let $\alpha = \Omega(\ln|s|)$ the dual constraint corresponding to s is satisfied, and if we let $\alpha = \Theta(\ln n)$ all the constraints are satisfied, i.e. $\frac{1}{\alpha}y$ is dual-feasible. Following the discussion in the beginning of this section, our algorithm has $\Theta(\ln n)$ approximation ratio.

3 The Primal-Dual Scheme

Primal-dual algorithms are a powerful class of optimization techniques based on LP-duality. Given an optimization problem and the primal and dual LPs, a primal-dual algorithm operates roughly as follows. The algorithm starts with an infeasible solution x to the primal LP, and a suboptimal (but feasible) solution y to the dual LP. In each step, changes are made so that x moves closer to being feasible and y moves closer to being optimal. The art lies in how exactly these changes are made. In the end, the algorithm is analyzed by looking at the relations between x and y. One way to do so is by using complementary slackness.

3.1 Complementary Slackness

Definition 1. Given the following primal and dual linear programs

$$
\begin{array}{rrcl|rrcl}
P: & & & & D: & & & \\
\min & c^T x & & & \max & b^T y & & \\
\text{s.t.} & Ax & \geq & b & \text{s.t.} & A^T y & \leq & c \\
 & x & \geq & 0 & & y & \geq & 0
\end{array}
$$

two feasible solutions x and y satisfy *complementary slackness* if:

1. $\forall i : x_i > 0 \implies (A^T)_i y = c_i$, and
2. $\forall j : y_j > 0 \implies A_j x = b_j$.

Simply put, if a variable takes a positive value, then the corresponding inequality must be tight. Alternatively, this can be stated as $(c^T - y^T A)x = 0$ and $y^T(Ax - b) = 0$.

Theorem 2. If feasible solutions x and y satisfy complementary slackness, then x is primal-optimal and y is dual-optimal.

Proof. Given that $(c^T - y^T A)x = 0$ and $y^T(Ax - b) = 0$, we have $c^T x = y^T Ax = y^T b$. The claim follows from weak duality. □

Definition 2. Given the same primal and dual linear programs, two feasible solutions x and y satisfy *relaxed complementary slackness*, with parameters $\alpha, \beta \geq 1$, if:

1. $\forall i : x_i > 0 \implies \frac{1}{\alpha} c_i \leq (A^T)_i y \leq c_i$, and
2. $\forall j : y_j > 0 \implies b_j \leq A_j x \leq \beta b_j$.

Theorem 3. If feasible solutions x and y satisfy (α, β)-complementary slackness, then x is $\alpha\beta$-primal-optimal and y is $\alpha\beta$-dual-optimal.

3.2 Shortest Path

Here we go over an application of complementary slackness. Consider the following linear program and its dual for finding the shortest s-t path in a weighted directed graph ($\delta^+(S)$ denotes the outgoing edges of a vertex set S):

$$
\begin{array}{rrcll|rrcll}
P: & & & & & D: & & & & \\
\min & \sum_e c_e x_e & & & & \max & \sum_{S: s \in S, t \notin S} y_S & & & \\
\text{s.t.} & \sum_{e \in \delta^+(S)} x_e & \geq & 1 & \forall S \subseteq V, s \in S, t \notin S & \text{s.t.} & \sum_{\delta^+(S) \ni e} y_S & \leq & c_e & \forall e \in E \\
 & x & \geq & 0 & & & y & \geq & 0 &
\end{array}
$$

Note that restricting x to have 0-1 values would result in an integer program that exactly formulates the problem.

We present a primal-dual algorithm. We start with infeasible primal solution $x = 0$ and feasible solution $y = 0$. Until x is feasible, there is a s-t cut S, such that the corresponding constraint is violated by x. Take the smallest such S. We increase the corresponding dual variable y_S, until one of the dual constraints becomes tight, say the dual constraint corresponding to e. Finally, we add e to the primal solution, i.e. we let $x_e = 1$ and proceed to the next step. Note that after this step, the constraint corresponding to S is no longer violated by x. When x becomes feasible, the algorithm outputs the s-t path in the added edges (i.e. the edges with $x_e = 1$). The claim below implies that there is exactly one such path.

Claim 1. At any point, the subgraph of edges e with $x_e = 1$, is a tree rooted at s.

Proof. We prove the claim using induction. At first the set of edges with $x_e = 1$ is empty so the claim is true. For the induction step, assume that so far the chosen edges form a tree T. If T contains the vertex t, x is already feasible and we are done. Otherwise, note that the next S the algorithm chooses (to increase y_S) is exactly the set of vertices of T. Therefore, the edge that becomes tight and is added to the solution is an edge from T to $V \setminus T$. This completes the induction step. □

To analyze the algorithm, we make a final change to x. Given that the edges with $x_e = 1$ form a tree, there is exactly one path from s to t. We delete every edge e that is not on the s-t path, i.e. we let $x_e = 0$. To show this path is optimal, we prove that the complementary slackness property holds for x and y.

The first condition is trivial the way we designed the algorithm. We only let $x_e = 1$, when the corresponding constraint becomes tight. Afterwards, the y_S's are not decreased, and since the constraint is tight y_S is not increased for any other cut S that contains e, i.e. the constraint remains tight.

The second condition of complementary slackness follows from the fact that the edges with $x_e = 1$ form a path from s-t. Therefore, for any s-t cut S, the corresponding constraint is tight. This completes the proof of optimality.

Remark 1. One can see that this primal-dual algorithm is equivalent to Dijkstra's algorithm.

3.3 Set Cover Revisited

Here we present an application of relaxed complementary slackness. Consider, again, the primal and dual LPs for the set cover problem.

$$\begin{array}{llcll} P: & & & & \\ \min & \sum_{s\in\mathcal{S}} c(s)\cdot x_s & & & \\ \text{s.t.} & \sum_{s\ni u} x_s & \geq & 1 & \forall u \in \mathcal{U} \\ & x_s & \geq & 0 & \end{array} \qquad \Bigg| \qquad \begin{array}{llcll} D: & & & & \\ \max & \sum_{u\in\mathcal{U}} y_u & & & \\ \text{s.t.} & \sum_{u\in s} y_u & \leq & c(s) & \forall s \in \mathcal{S} \\ & y_u & \geq & 0 & \end{array}$$

We give a primal-dual algorithm that achieves f-approximation, where f is the highest frequency of an element u, i.e. the maximum number of occurrences of an element in the collection $\mathcal{S}$.

We start with infeasible primal solution $x = 0$ and feasible dual solution $y = 0$. In every step, we take an uncovered element u, and increase y_u until some constraint becomes tight, say the constraint corresponding to s. Then, we add s to our solution, i.e. we let $x_s = 1$. The algorithm stops when every element is covered. We claim the x and y satisfy $(1, f)$-complementary slackness.

The first condition is trivial, since we let $x_s = 1$ only when the corresponding constraint is tight, and the constraint remains tight throughout the algorithm. The second condition holds, because each x_s is at most 1. Hence, for any element u the sum $\sum_{s \ni u} x_s$ is at most equal to the number of occurrences of u in the collection, which is at most f. Therefore, x and y satisfy $(1, f)$-complementary slackness and the algorithm outputs an f-approximation.

4 A Brief Overview of the Ellipsoid Method

Many naturally occurring linear programs, such as the shortest path LP in the previous section, have a super-polynomial number of constraints. One may wonder if there is an efficient way to solve such linear programs. In 1979, Khachyian showed how the ellipsoid method can be used to solve linear programs.

Note that any optimization can be reduced to a search problem, i.e. to find a point x in the polytope P that maximizes $c^T x$, one can binary-search on z^*, and look for a point in $P \cap \{x \mid c^T x \geq z^*\}$ if possible. If the constraints of the polytope satisfy certain reasonable conditions, then the approximate solution of the binary search can be rounded to an exact solution. A separation oracle is defined as follows. Given a point x and a polytope P, the oracle must decide if $x \in P$, and if it is not, the oracle must output a hyperplane that separates x from P.

Provided with a separation oracle, one can use the ellipsoid method to solve the search problem. Starting with a large ellipsoid E_0, centered at s_0, that contains the polytope, in every step the algorithm checks whether $s_i \in P$. If so, we are done and s_i is returned. Otherwise, there is a hyperplane $a^T x \leq b$ that separates s_i from P. This hyperplane can be shifted to include s_i, cutting the ellipsoid in half, with the polytope lying in one half. E_{i+1} is taken as the smallest ellipsoid that contains that half-ellipsoid. The algorithm will stop at some point since the volume of the ellipsoid decreases exponentially.

References

[Goe94] M. Goemans. Introduction to Linear Programming. Lecture notes, available from `http://www.cs.cmu.edu/afs/cs/user/glmiller/public/Scientific-Computing/F-11/RelatedWork/Goemans-LP-notes.pdf`, October 1994.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011.