Lecture 16: Linear Programming Duality

Objectives of this lecture

- Standard form of linear programs and its dual.
- Derive max-flow min-cut using linear program duality.
- Transform linear programs to flow problems.

1 Philosophy

Linear programs (or more generally convex optimization problems) have dual instances. These instances are formed around the transpose of the constraints matrix. Duals have deep connections with high dimensional geometry / real analysis, but can be obtained from standard forms: the dual of $\min_{x:Ax=b,x\geq 0} c^\top x$ is $\max_{y:A^\top y\leq c} b^\top y$.

In particular, a linear programming algorithm can be viewed as taking $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ as input, and outputs **a pair** of solutions $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ such that:

1. $Ax = b$, $x \geq 0$ (x is feasible for the primal)
2. $A^\top y \leq c$ (y is feasible for the dual)
3. $c^\top x = b^\top y$ (the objective values are equal)

This algebraic fact, with some manipulations of inequalities, give some fairly surprising connectiosn between problems. One such connection is max-flow/min-cut.

2 Assumed Knowledge

1. Linear algebra:
   (a) A is an $m \times n$ matrix
   (b) x is $n \times 1$ vector
   (c) Ax is $m \times 1$ vector: for each $1 \leq i \leq m$, $(Ax)_i = \sum_{j=1}^n A_{ij} x_j$.
2. Linear inequalities, $a^\top x \leq b$ as short hand for $\sum_j a_j x_j \leq b$.

3. Overloading $\leq$ for vectors: $a \leq b$ for two vectors of the same length means $a_i \leq b_i$ for all entries i.
4. Max-flow and min-cut (will refresh)
5. Reducing problems to linear programs (will refresh)
6. Minimum cost flows

3 Linear Programs in Standard Forms

Recall linear programming in standard form:

$$\begin{aligned} \min \quad & c^\top x \\ \text{subject to:} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Many problems can be written as linear programs. Some examples are:

1. Shortest path: this is almost a special case of min-cost flow, in that edge have no upper capacities. So we just want to minimize the cost of a flow vector $f \geq 0$,

   $$c^\top f = \sum_e c_e f_e$$

   subject to the flow sending 1 unit from s to t,

   $$\forall u \qquad \sum_{v \to u} f_{v \to u} - \sum_{u \to v} f_{u \to v} = \begin{cases} \text{-1} & \text{if } u = s, \\ 1 & \text{if } u = t, \\ 0 & \text{otherwise.} \end{cases}$$

   The last of these can be abbreviated via the edge-vertex incidence matrix to $B^\top f = \mathbb{1}(t) - \mathbb{1}(s)$, where $B \in \mathbb{R}^{m \times n}$ is the edge-vertex incidence matrix given by

   $$B_{e=(u \to v),x} = \begin{cases} 1 & \text{if } x = v \\ -1 & \text{if } x = u \\ 0 & \text{otherwise} \end{cases}$$

   and the indicator vector $\mathbb{1}_i$ is the vector with 1 in location i, and 0 everywhere else:

   $$\mathbb{1}(i)_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

   So $s \to t$ shortest path with costs c can be written in standard form as

   $$\begin{aligned} \min \quad & c^\top f \\ \text{subject to:} \quad & B^\top f = \mathbb{1}(t) - \mathbb{1}(s) \\ & f \geq 0 \end{aligned}$$

2. Maximum independent set, which is finding the maximum number of vertices such that no edge has more than 1 end point taken, becomes maximizing $\mathbb{1}^\top x$ for $x \in \mathbb{R}^V$ subject to $x \geq 0$ and $x_u + x_v \leq 1$ for all edges $uv \in E$. To make it into standard form, introduce slack variables on each edge s_{uv} so that $x_u + x_v + s_{uv} = 1$. This leads to

$$\begin{aligned} \min \quad & \sum_{u \in V} x_V \\ \text{subject to:} \quad & x_u + x_v + s_{uv} = 1 \qquad \forall uv \in E \\ & x, s \geq 0 \end{aligned}$$

or in matrix notation, define the (positive) edge vertex incidence matrix $P \in \mathbb{R}^{m \times n}$ to be the matrix

$$P_{e=uv,x} = \begin{cases} 1 & \text{if } x = v \text{ or } x = v \\ 0 & \text{otherwise} \end{cases}$$

and overloading notations for 1 to be the all 1s vector of appropriate dimensions,

$$\begin{aligned} \min \quad & \begin{bmatrix} 1 \\ 0 \end{bmatrix}^\top \begin{bmatrix} x \\ s \end{bmatrix} \\ \text{subject to:} \quad & \begin{bmatrix} P & I \end{bmatrix}^\top \begin{bmatrix} x \\ s \end{bmatrix} = 1 \\ & x, s \geq 0 \end{aligned}$$

Note however the optimum answer for this may be fractional: consider for example a triangle. The max independent set has size 1, but a valid solution is 1/2 at all vertices.

3. previous terms used $Ax \leq b$, instead of $= b$, and also maximize $c^\top x$ instead of minimizing. Converting this to standard form requires adding a slack, and flipping the sign on c:

$$\begin{aligned} \min \quad & \begin{bmatrix} -c \\ 0 \end{bmatrix}^\top \begin{bmatrix} x \\ s \end{bmatrix} \\ \text{subject to:} \quad & \begin{bmatrix} A & I \end{bmatrix}^\top \begin{bmatrix} x \\ s \end{bmatrix} = b \\ & x, s \geq 0 \end{aligned}$$

4 Linear Programming Duality

Linear programs have duals. The reason standard form is useful is its dual can also be described succinctly using algebra. Specifically, it is the following minimization problem

$$\begin{aligned} \max \quad & b^\top y \\ \text{subject to:} \quad & A^\top y \leq c \end{aligned}$$

Note that the dual does not have any restrictions on the sign of y.

Linear programming duality states that these two problems, when both feasible, have the same objective values.

Theorem 1: Linear Programming Duality

When both $\{x \geq 0 | Ax = b\}$ and $\{y | A^\top y \leq c\}$ are feasible,

$$\min_{x \geq 0, Ax = b} c^\top x = \max_{y, A^\top y \leq c} b^\top y.$$

The primal is infeasible if and only iff the dual is unbounded; the primal is unbounded if and only if the dual is infeasible.

We make sense of this in several steps. The first is to check weak duality, that is, for any x satisfying $Ax = b$, $x \geq 0$, and any y satisfying $A^\top y \leq c$, we have

$$b^\top y \leq c^\top x.$$

To see this, substitute $b = Ax$ into $b^\top y$. We get

$$\begin{aligned} b^\top y &= (Ax)^\top y \\ &= x^\top \left(A^\top y\right). \end{aligned}$$

But since x is non-negative, we have $x_i d_i \leq x_i c_i$ for any $d_i \leq c_i$. Note this statement does not require anything about the signs of c_i or d_i.

So applying $A^\top y \leq c$ gives $b^\top y \leq c^\top x$.

This means that whenever LP solvers return feasible primal x and feasible dual y such that $c^\top x = b^\top y$, those x and y are optimal for their respective problems.

The next item to realize is the correspondence between inequalities and variables. That is, each inequalty in the primal becomes a variable in the dual, while each inequality in the dual corresponds to a variable in the primal. This is exactly the swapping in dimensions in A and $A^\top$.

5 Max-Flow / Min-Cut as Special Case of LP Duality

LP duality gives another proof that maximum flow equals to minimum cut. Recall the reduction from max-flow to linear programs: we add an edge from t to s with infinite capacity but cost -1.

The linear program, on this new graph is given by:

1. $f \geq 0$ is the vector (on all edges) encoding the flows per edge.
2. $s = \text{cap} - f$ is the remaining capacities left on the edges. So we need $s \geq 0$ and $s + f = \text{cap}$.
3. $c = -\mathbb{1}_{t \to s}$, the vector that's -1 on the $t \to s$ edge, 0 everywhere else
4. B is the edge vertex incidence matrix, and we need $B^\top f = 0$.

So overall, the primal formulation is

$$\begin{aligned} \min \quad & -f_{t\to s} \\ \text{subject to:} \quad & B^\top f = 0 \\ & f + s = \text{cap} \\ & f, s \geq 0 \end{aligned}$$

the two linear equalities, written in matrix form, is

$$\begin{aligned} \min \quad & \begin{bmatrix} -\mathbb{1}(t \to s) \\ 0 \end{bmatrix}^\top \begin{bmatrix} f \\ s \end{bmatrix} \\ \text{subject to:} \quad & \begin{bmatrix} B^\top & 0 \\ I & I \end{bmatrix} \begin{bmatrix} f \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ \text{cap} \end{bmatrix} \\ & f, s \geq 0 \end{aligned}$$

so applying linear programming duality for standard forms as stated above with:

1. $A = \begin{bmatrix} B^\top & 0 \\ I & I \end{bmatrix}$,
2. $b = \begin{bmatrix} 0 \\ \text{cap} \end{bmatrix}$,
3. $c = \begin{bmatrix} -\mathbb{1}(t \to s) \\ 0 \end{bmatrix}$,
4. $x = \begin{bmatrix} f \\ s \end{bmatrix}$

gives that its dual, $\max_{y: A^\top y \leq c} b^\top y$, writes out to

$$\begin{aligned} \max \quad & \begin{bmatrix} 0 \\ \text{cap} \end{bmatrix}^\top y \\ \text{subject to:} \quad & \begin{bmatrix} B & I \\ 0 & I \end{bmatrix} y \leq \begin{bmatrix} -\mathbb{1}(t \to s) \\ 0 \end{bmatrix} \end{aligned}$$

We parse this by first breaking down y a bit. The number of rows of A is $n+m$, so is the number of columns of $A^\top$. So y is a length $n+m$ vector, and we can write it as

$$y = \begin{bmatrix} z \\ g \end{bmatrix}$$

where $z \in \mathbb{R}^n$ and $g \in \mathbb{R}^m$. Note that these blocks of sizes n and m line up well with the length n and m blocks in $b = \begin{bmatrix} 0 \\ \text{cap} \end{bmatrix}$. So replacing y with $\begin{bmatrix} z \\ g \end{bmatrix}$ turns the dual into

$$\begin{aligned} \max \quad & \text{cap}^\top g \\ \text{subject to:} \quad & \begin{bmatrix} B & I \\ 0 & I \end{bmatrix} \begin{bmatrix} z \\ g \end{bmatrix} \leq \begin{bmatrix} -\mathbb{1}(t \to s) \\ 0 \end{bmatrix} \end{aligned}$$

The notation of using g to denote the length m part is chosen to mirror $f \in \mathbb{R}^m$, it can also be interpreted as the 'gradient' of the $z \in \mathbb{R}^n$: we will show below each $g_{u\to v}$ is related to the difference between z_v and z_u.

Unraveling the blocks further, we get $g \le 0$, and also

$$Bz + g \le -\mathbb{1}(t \to s)$$

which we can now start to interpret graph theoretically again using the structure of B as the edge-vertex incidence matrix.

Specifically, for an edge $u \to v$, we get

$$z_v - z_u + g_{u\to v} \le \begin{cases} -1 & \text{if } u \to v = t \to s \\ 0 & \text{otherwise} \end{cases}$$

For some edge $u \to v$ that's not the $t \to s$ edge, if $z_v \le z_u$ then we can set $g_{u,v}$ to 0 and satisfy $z_v - z_u \le 0$. Otherwise, we need to set $g_{u\to v}$ to cancel out the increase. As the capacities are positive, we are better off setting $g_{u\to v}$ to be as 'not negative' as possible, aka. to $-\max\{0, z_v - z_u\}$. So then the objective that we are maximizing becomes

$$\sum_{u\to v} \text{cap}_{u\to v} - \max\{0, z_v - z_u\} = -\sum_{u\to v} \text{cap}_{u\to v} \max\{0, z_v - z_u\}$$

which is equivalent to minimizing its negation, $\sum_{u\to v} \text{cap}_{u\to v} \max\{0, z_v - z_u\}$.

On the other hand, because the capacity of the $t \to s$ edge is infinity, we need to have $g_{t\to s} = 0$ for the objective ($\text{cap}^\top g$ to not be $-\infty$. So we need

$$z_s - z_t \le -1$$

or equivalently

$$z_t \ge z_s + 1$$

In other words, the increase in z from s to t is at least 1. This corresponds to a cut because we can label everything on the side of s to be 0, everything on the side of t to be 1.

Note however that the cut can be fractional. So to fully show maxflow-mincut, we also need to show that any fractional cut solution leads to an integral / combinatorial one. This is a problem on Oral Homework 3.

One side consequence of the steps of this reduction, specifically the introduction and removal of slacks, is that we also get the dual of the $Ax \le b$ version.

Lemma 1: Dual of inequality version

The dual of $\max_{x\ge 0, Ax\le b} c^\top x$ is $\min_{y\ge 0, A^\top y \ge c} b^\top y$.