Math 482: Linear Programming[1] Mikhail Lavrov

Lecture 29: The Primal-Dual Algorithm I

April 17, 2020 *University of Illinois at Urbana-Champaign*

1 Motivation

The primal-dual algorithm is a method for solving linear programs inspired by the Ford–Fulkerson method.

Instead of applying the simplex method directly, we start at a feasible solution and then compute the direction which is most likely to improve that solution. This direction is analogous to finding an augmenting path for a network flow. Then we go in that direction as far as possible.

This has its advantages and disadvantages:

- Each such "augmenting" step takes longer to do than a simple step of the simplex method.
- However, an augmenting step might skip across the feasible region in a way that would require many pivot steps.

2 The primal-dual algorithm

2.1 Setup

We will work with problems in equational form, where the primal-dual pair is

$$(\mathbf{P}) \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases} \qquad (\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^m}{\text{maximize}} & \mathbf{u}^\mathsf{T}\mathbf{b} \\ \text{subject to} & \mathbf{u}^\mathsf{T}A \leq \mathbf{c}^\mathsf{T} \\ & \mathbf{u} \text{ unrestricted} \end{cases}$$

We will require $\mathbf{b} \geq \mathbf{0}$, for technical reasons we'll see later. We can always put $(\mathbf{P})$ into this form, so it's not an issue. We'll also assume that $(\mathbf{P})$ is a minimization problem and $(\mathbf{D})$ is a maximization problem.

Although we're calling $(\mathbf{P})$ the primal program and $(\mathbf{D})$ the dual program, it's not unreasonable to start in a situation where $(\mathbf{D})$ is the program we're actually interested in, and $(\mathbf{P})$ is the one we construct as an auxiliary tool to use the primal-dual algorithm.

We will consider the example

$$(\mathbf{P}) \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^3}{\text{minimize}} & 2x_1 + 2x_2 + x_3 \\ \text{subject to} & 2x_1 + x_2 - 4x_3 = 3 \\ & 4x_1 - x_2 + x_3 = 3 \\ & x_1, x_2, x_3 \geq 0 \end{cases} \qquad (\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^2}{\text{maximize}} & 3u_1 + 3u_2 \\ \text{subject to} & 2u_1 + 4u_2 \leq 2 \\ & u_1 - u_2 \leq 2 \\ & -4u_1 + u_2 \leq 1 \end{cases}$$

[1] This document comes from the Math 482 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/482-spring-2020.html

2.2 The augmenting linear program

Suppose we're at the dual solution $(u_1, u_2) = (1, 0)$ and, looking around from this point, we want to pick the best direction to move in. The problem "pick the best direction" is itself an optimization problem. But what are its constraints and objective function?

We want to pick a direction $\mathbf{v} \in \mathbb{R}^2$ to move in: from $\mathbf{u}$, we will follow the ray $\{\mathbf{u} + t\mathbf{v} : t > 0\}$ for as far as we can. The change in objective function from $\mathbf{u}^\mathsf{T}\mathbf{b}$ to $(\mathbf{u} + t\mathbf{v})^\mathsf{T}\mathbf{b}$ is $t \cdot (\mathbf{v}^\mathsf{T}\mathbf{b})$. So in order to increase the objective function as quickly as possible, we'd like to maximize $\mathbf{v}^\mathsf{T}\mathbf{b}$: the same objective function as in (**D**).

What are the constraints on $\mathbf{v}$? For this, we should look at (**D**) and at the point $\mathbf{u} = (1, 0)$ more closely. We see that at $\mathbf{u} = (1, 0)$, we have $2u_1 + 4u_2 = 2$ (the first constraint of (**D**) is tight) but $u_1 - u_2 = 1 < 2$ and $-4u_1 + u_2 = -4 < 1$ (the second and third constraints of (**D**) are slack).

As a result, our direction $\mathbf{v}$ had better satisfy $2v_1 + 4v_2 \leq 0$, so that when we move from $\mathbf{u}$ to $\mathbf{u} + t\mathbf{v}$, the constraint $2u_1 + 4u_2 \leq 2$ is not violated. This is essentially asking for the left-hand side of the inequality not to increase. On the other hand, $\mathbf{v}$ doesn't need to worry about the other constraints in (**D**): they're not tight, so no matter which direction we pick, we can go in that direction for a while and not violate the constraints.

Finally, it is important to remember that $\mathbf{v}$ is a direction. Picking the direction $2\mathbf{v}$ or $\frac{1}{2}\mathbf{v}$ would be equivalent to picking $\mathbf{v}$, because we're going to go from $\mathbf{u}$ to $\mathbf{u} + t\mathbf{v}$ for some yet-to-be-determined multiple t, and so the magnitude of $\mathbf{v}$ doesn't matter. Just maximizing $3v_1 + 3v_2$ would care a lot about the magnitude, so we should add some "normalizing" constraint to avoid this.

In nonlinear programming, it would be typical to add the constraint $\|\mathbf{v}\| = 1$: $\mathbf{v}$ lies on the unit sphere. Here, we will add the constraints $v_1 \leq 1$ and $v_2 \leq 1$. Any direction $\mathbf{v}$ can be scaled down to satisfy those constraints.

(Note: because we assumed $\mathbf{b} \geq \mathbf{0}$, we don't have to worry about putting lower bounds down on v_1 and v_2. If there were a $\mathbf{v}$ with all negative components, then scaling $\mathbf{v}$ to $2\mathbf{v}$ or $3\mathbf{v}$ or $100\mathbf{v}$ would produce much, much worse values of $\mathbf{v}^\mathsf{T}\mathbf{b}$.)

Putting these together, we get the linear program below on the left:

$$\begin{cases} \underset{\mathbf{v} \in \mathbb{R}^2}{\text{maximize}} & 3v_1 + 3v_2 \\ \text{subject to} & 2v_1 + 4v_2 \leq 0 \\ & v_1 \leq 1 \\ & v_2 \leq 1 \end{cases} \qquad \begin{cases} \underset{\mathbf{v} \in \mathbb{R}^m}{\text{maximize}} & \mathbf{v}^\mathsf{T}\mathbf{b} \\ \text{subject to} & \mathbf{v}^\mathsf{T}A_J \leq \mathbf{0}^\mathsf{T} \\ & v_1, \ldots, v_m \leq 1 \end{cases}$$

In general, the linear program we get will have the form above on the right. Here, I'm taking J to be the set of constraints that are tight at a particular point $\mathbf{u}$ we're trying to augment: A_J just takes the columns of A corresponding to those constraints, and $\mathbf{c}_J$ just takes the entries of $\mathbf{c}$ corresponding to those constraints.

If we solve the direction-finding linear program in our example, we'll get the vector $\mathbf{v} = (1, -\frac{1}{2})$. Then, our next goal is to figure out the step size, t.

We are moving from $\mathbf{u} = (1, 0)$ to $\mathbf{u} + t\mathbf{v} = (1 + t, -\frac{1}{2}t)$. We know this will improve our objective function for as long as we do it, so we'd like to choose t as large as possible. However, our constraints come into play.

- The first constraint, $2u_1 + 4u_2 \leq 2$, will not be an issue. Since we included $2v_1 + 4v_2 \leq 0$ as a constraint in the direction-finding problem, we know that the new point $\mathbf{u} + t\mathbf{v}$ will satisfy $2(u_1 + tv_1) + 4(u_2 + tv_2) \leq 2$ for any value of t.
- The second constraint, $u_1 - u_2 \leq 2$, tells us that $(1 + t) - (-\frac{1}{2}t) \leq 2$, or $t \leq \frac{2}{3}$.
- The third constraint, $-4u_1 + u_2 \leq 1$, tells us that $-4(1 + t) + (-\frac{1}{2}t) \leq 1$, or $t \geq -\frac{10}{9}$.

  This kind of deduction isn't useful because we already want to choose as large a t as possible; lower bounds aren't doing much.

(As a sanity check: the value $t = 0$ should always be in the allowable range, because going from $\mathbf{u}$ to $\mathbf{u} + 0\mathbf{v}$ is just staying put, and we started out at a feasible $\mathbf{u}$.)

The strictest upper bound on t is $t \leq \frac{2}{3}$, so we set $t = \frac{2}{3}$. We go from $\mathbf{u}$ to $\mathbf{u} + \frac{2}{3}v = (\frac{5}{3}, -\frac{1}{3})$.

In its simplest form, the primal-dual algorithm would just repeat this augmenting procedure over and over again. This is not actually very good: we're forced to solve a separate linear program at every step, which is very slow. From here, we'll be looking for improvements that speed up this procedure.

2.3 The restricted primal

Let's look at the dual of the direction-finding linear program. In our example (still augmenting the point $\mathbf{u} = (1, 0)$) we'll get:

$$(\mathbf{RP}) \begin{cases} \underset{\mathbf{x}\in\mathbb{R},\mathbf{y}\in\mathbb{R}^2}{\text{minimize}} & y_1 + y_2 \\ \text{subject to} & 2x_1 + y_1 = 3 \\ & 4x_1 + y_2 = 3 \\ & x_1, y_1, y_2 \geq 0 \end{cases} \qquad (\mathbf{DRP}) \begin{cases} \underset{\mathbf{v}\in\mathbb{R}^2}{\text{maximize}} & 3v_1 + 3v_2 \\ \text{subject to} & 2v_1 + 4v_2 \leq 0 \\ & v_1 \leq 1 \\ & v_2 \leq 1 \end{cases}$$

In general, we'll get:

$$(\mathbf{RP}) \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} & y_1 + \cdots + y_m \\ \text{subject to} & A_J\mathbf{x}_J + I\mathbf{y} = \mathbf{b} \\ & \mathbf{x}, \mathbf{y} \geq \mathbf{0} \end{cases} \qquad (\mathbf{DRP}) \begin{cases} \underset{\mathbf{v}\in\mathbb{R}^m}{\text{maximize}} & \mathbf{v}^\mathsf{T}\mathbf{b} \\ \text{subject to} & \mathbf{v}^\mathsf{T}A_J \leq \mathbf{c}_J^\mathsf{T} \\ & v_1, \ldots, v_m \leq 1 \end{cases}$$

The latest linear program we've gotten is closely related to the primal program $(\mathbf{P})$ that we haven't looked at so far; it is called the "restricted primal" ($(\mathbf{RP})$) because we've taken $(\mathbf{P})$ and restricted our attention to only a subset of the variables: the variables indexed by J. The direction-finding linear program we looked at earlier is usually called $(\mathbf{DRP})$: the dual of the restricted primal.

We will use (**RP**) instead of (**DRP**) to make our augment-and-repeat procedure more efficient. There will be two ingredients to this:

1. First, when we use the simplex method to solve (**RP**), we will be able to get an optimal solution to (**DRP**) as well: the optimal direction $\mathbf{v}$. So it's fine to solve (**RP**) instead of (**DRP**).
2. Second, the different iterations of (**RP**) will not need to be solved independently. We will be able to take the optimal solution to (**RP**) at one iteration, and use it as a starting point for the next iteration.

2.4 Some motivation for (RP)

There is an independent argument for where (**RP**) comes from. It isn't important to how our method will work, but it's nice to know.

Suppose we want to know if $\mathbf{u}$ is an optimal solution to (**D**). We can find out by seeing if there's a solution $\mathbf{x}$ to (**P**) satisfying complementary slackness with $\mathbf{u}$.

Complementary slackness says that $x_i = 0$ whenever $\mathbf{u}^\mathsf{T} A_i < c_i$. So instead of solving $A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$, we want to solve $A_J \mathbf{x}_J = \mathbf{b}$, $\mathbf{x}_J \geq \mathbf{0}$, where J is the set of tight constraints for (**D**).

We can check if there is such a solution by adding artificial variables $y_1, y_2, \ldots, y_m$ to the constraints, and minimizing $y_1 + y_2 + \cdots + y_m$. If this artificial objective function can become 0, then there is such a solution, and $\mathbf{u}$ is optimal.