Math 482: Linear Programming[1] Mikhail Lavrov

Lecture 19: Fourier–Motzkin Elimination

October 14, 2019 *University of Illinois at Urbana-Champaign*

1 The feasibility problem

A related question to the linear program optimization problems we've been solving is the existence question: given A and $\mathbf{b}$, is the set $\{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \le \mathbf{b}\}$ nonempty? In other words, does there exist a *feasible* solution to a linear program?

We already have methods to solve this problem. Right now, one reasonable approach is to use the dual simplex method:

1. If necessary, put the constraints into equational form.
2. Find a basic solution (not necessarily a feasible one) by row reduction.
3. Invent an objective function that will give us a dual feasible tableau.
4. Use the dual simplex method to try to make the tableau primal feasible as well.

This subproblem captures most of the important features of linear programming as well: it's not significantly easier than an optimization problem. For example, we could use the feasibility problem to answer questions like: "Given a linear program $\min\{\mathbf{c}^\mathsf{T}\mathbf{x} : A\mathbf{x} \ge \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$, does the optimal solution have objective value at least z_0?" To do so, just add $\mathbf{c}^\mathsf{T}\mathbf{x} \ge z_0$ as a constraint and ask if any point $\mathbf{x} \in \mathbb{R}^n$ satisfies all of the constraints including this one.

2 Fourier–Motzkin elimination

There is another algorithm for solving the feasibility problem, which is very different from the simplex method. Here, "very different" is kind of like "worse", for large problems. But this algorithm, called Fourier–Motzkin elimination, is still occasionally interesting. It's simpler, so it's easier to prove things about. And for small problems, it might not even be that bad to use.

2.1 The algorithm

The idea of the algorithm is just this: first, we find a procedure to reduce an n-variable problem to an equivalent $(n-1)$-variable problem. (Equivalent means that one system of inequalities has a feasible solution if and only if the other one does.) Then, repeat this procedure, eliminating variables one at a time.

Eventually, we'll be left with a 1-variable problem, and these are easy to solve: just check if there are any numbers between the greatest lower bound and the least upper bound. We will even be

[1] This document comes from the Math 482 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/482-fall-2019.html

able to trace back our steps, using a solution to the 1-variable problem to find solutions to the 2-variable, 3-variable, and eventually the original n-variable problem.

So how do we eliminate a variable? Well, let's suppose we have variables $x_1, x_2, \dots, x_n$, and we want to eliminate x_n.

Our first step is to solve for x_n. For each inequality such as

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$$

we will get one of the two inequalities

$$x_n \leq \frac{b - a_1x_1 - a_2x_2 - \cdots - a_{n-1}x_{n-1}}{a_n} \quad \text{or} \quad x_n \geq \frac{b - a_1x_1 - a_2x_2 - \cdots - a_{n-1}x_{n-1}}{a_n}$$

depending on whether $a_n > 0$ or $a_n < 0$. (If $a_n = 0$, we'll leave the inequality alone, since it doesn't involve x_n to begin with.)

We'll end up with a collection of lower and upper bounds:

$$x_n \geq L_1, x_n \geq L_2, \dots, x_n \geq L_k \qquad x_n \leq U_1, x_n \leq U_2, \dots, x_n \leq U_m.$$

Here, each L_i and U_j is an expression in the $n-1$ variables $x_1, x_2, \dots, x_{n-1}$.

It's possible to choose a value of x_n that satisfies all these bounds if and only if

$$\max\{L_1, L_2, \dots, L_k\} \leq \min\{U_1, U_2, \dots, U_m\}$$

because thern there's something in between all the lower bounds and all the upper bounds. Unfortunately, we can't just write down this single equation, because the lower and upper bounds are all in terms of unknown values $x_1, \dots, x_{n-1}$, so we can't tell which ones are larger or smaller. So we compromize by writing down $k \cdot m$ inequalities:

$$\begin{array}{cccc} L_1 \leq U_1 & L_1 \leq U_2 & \cdots & L_1 \leq U_m \\ L_2 \leq U_1 & L_2 \leq U_2 & \cdots & L_2 \leq U_m \\ \vdots & \vdots & \ddots & \vdots \\ L_k \leq U_1 & L_k \leq U_2 & \cdots & L_k \leq U_m \end{array}$$

If an x_n exists that satisfies all the constraints, then for every i and j, $L_i \leq x_m \leq U_j$, so all of these inequalities hold. Conversely, if all of these inequalities hold, then the max-min inequality holds as well, and we can choose x_n between $\max\{L_1, \dots, L_k\}$ and $\min\{U_1, \dots, U_m\}$.

This results in our new system in only $n-1$ variables. It consist of the $k \cdot m$ inequalities above, plus any of the original inequalities which didn't involve x_n in them to begin with.

2.2 Example

Suppose that we have a system of inequalities

$$\begin{cases} x - y \geq 1 \\ -x + 2y \geq 1 \\ 3x - 5y \geq 1 \\ x, y \geq 0 \end{cases}$$

We begin by eliminating y. First, we solve for y in each inequality

$$\begin{cases} \quad x - \quad y \ge 1 \implies y \le x - 1 \\ -x + 2y \ge 1 \implies y \ge \frac{x+1}{2} \\ \ 3x - 5y \ge 1 \implies y \le \frac{3x-1}{5} \\ y \ge 0 \\ x \ge 0 \end{cases}$$

Next, we pair up the lower bounds and upper bounds. This gives us four inequalities on x, together with a fifth one which is just $x \ge 0$:

$$\begin{cases} \frac{x+1}{2} \le x - 1 \\ 0 \le x - 1 \\ \frac{x+1}{2} \le \frac{3x-1}{5} \\ 0 \le \frac{3x-1}{5} \\ x \ge 0 \end{cases}$$

These inequalities can all be simplified into lower bounds or upper bounds on x:

$$\begin{cases} x \ge 3 \\ x \ge 1 \\ x \ge 7 \\ x \ge \frac{1}{3} \\ x \ge 0 \end{cases}$$

In this case, they all happen to be lower bounds, so we know that they can be satisfied: for instance, setting $x = 7$ works. Then our lower and upper bounds on y become constants: we have $y \le 6$, $y \ge 4$, $y \le 4$, and $y \ge 0$. So we can set $y = 4$, and find a feasible solution.

2.3 Complexity

The downside of this method is that, in typical cases, the number of inequalities grows very quickly. For instance, here's one possible way that Fourier–Motzkin elimination can go:

$$\begin{aligned} \text{8 inequalities in } x_1, x_2, x_3, x_4 &\implies \text{4 lower and 4 upper bounds on } x_4 \\ &\implies 4 \times 4 = 16 \text{ inequalities in } x_1, x_2, x_3 \\ &\implies \text{8 lower and 8 upper bounds on } x_3 \\ &\implies 8 \times 8 = 64 \text{ inequalities in } x_1, x_2 \\ &\implies \text{32 lower and 32 upper bounds on } x_2 \\ &\implies 32 \times 32 = 1024 \text{ inequalities in } x_1 \end{aligned}$$

The above is worst-case behavior: the types of bounds are split exactly in half every time. But we expect *roughly* half of each type of bound, so this is not too far off from typical behavior, either.

If we started with just 8 inequalities in n variables, then this pattern would continue to give us 2^{2^k+2} inequalities in $n - k$ variables all the way up until we get $2^{2^{n-1}+2}$ inequalities in 1 variable on the very last step. This is worse than exponential: this is doubly exponential.

3 Farkas's lemma

Nevertheless, Fourier–Motzkin elimination is useful. For one thing, we can get the following result from it very quickly:

Theorem 3.1 (Farkas's lemma)**.** *For any system of inequalities $A\mathbf{x} \leq \mathbf{b}$, either there exists a feasible solution $\mathbf{x}$, or we can take a linear combination of the inequalities to derive a contradiction: there exists a vector $\mathbf{u} \geq \mathbf{0}$ such that $\mathbf{u}^\mathsf{T} A = \mathbf{0}^\mathsf{T}$ but $\mathbf{u}^\mathsf{T}\mathbf{b} < 0$.*

(The existence of $\mathbf{u}$ is the same as being able to derive a contradiction, because, left-multiplying $A\mathbf{x} \leq \mathbf{b}$ by $\mathbf{u}^\mathsf{T}$, we deduce $\mathbf{0}^\mathsf{T}\mathbf{x} < 0$, or $0 < 0$.)

Proof. Throughout Fourier–Motzkin elimination, all the inequalities we get have the form $\mathbf{v}^\mathsf{T} A\mathbf{x} \leq \mathbf{v}^\mathsf{T}\mathbf{b}$ for *some* $\mathbf{v} \geq \mathbf{0}$.

To see this, note that we can do everything in the algorithm using only two kinds of steps:

- Instead of solving for x_n, just divide through by $|a_n|$, the absolute value of the coefficient. The new inequality is a nonnegative multiple of the old one, and has x_n in it with a coefficient of 1 or -1.
- Instead of combining a lower bound and an upper bound, just add together an inequality with $+x_n$ and an inequality with $-x_n$. The new inequality eliminates x_n, and is definitely a nonnegative linear combination of preceding inequalities.

If Fourier–Motzkin elimination fails to find a solution, then we get a lower bound on x_1 and an upper bound on x_1 that contradict each other: linear combinations of the original inequalities that look like $x_1 \leq U$ and $-x_1 \leq -L$ with $L > U$. Adding these together, we get $0 \leq U - L$, which is exactly the kind of contradiction we were looking for. □

Farkas's lemma is important because it is the equivalent of strong duality for the feasibility problem: the system "$\mathbf{u}^\mathsf{T} A = \mathbf{0}^\mathsf{T}$ and $\mathbf{u}^\mathsf{T}\mathbf{b} < 0$ and $\mathbf{u} \geq \mathbf{0}$" is a dual feasibility problem to the original. We could have used Farkas's lemma to prove strong duality for linear programs, without relying on the correctness of the simplex method to get there.