CS 217 – Algorithm Design and Analysis

Shanghai Jiaotong University, Fall 2020

Yujie Lu, Tong Chen, Jiabao Ji

Handed out on Friday, 2020-06-05
First submission and questions due on Thursday, 2020-06-12
You will receive feedback from the TA.
Final submission due on Thursday, 2020-06-19

7 Farkas Lemma and LP Duality

7.1 Different Versions of Farkas Lemma

In the following, let $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, and let $\mathbf{x} = (x_1, \dots, x_n)^T$ be a column vector of n variables and $\mathbf{y} = (y_1, \dots, y_m)$ be a row vector of m variables.

Exercise 1. Show that the three versions of Farkas Lemma presented in class are all equivalent (I actually did not present the third version in class):

$$(\neg\exists \mathbf{x} : A\mathbf{x} \le \mathbf{b}) \iff (\exists \mathbf{y} \ge \mathbf{0} : \mathbf{y}^T A = \mathbf{0}, \mathbf{y}^T\mathbf{b} < 0) . \tag{1}$$

$$(\neg\exists \mathbf{x} \ge \mathbf{0} : A\mathbf{x} \le \mathbf{b}) \iff (\exists \mathbf{y} \ge \mathbf{0} : \mathbf{y}^T A \ge \mathbf{0}, \mathbf{y}^T\mathbf{b} < 0) . \tag{2}$$

$$(\neg\exists \mathbf{x} \ge \mathbf{0} : A\mathbf{x} = \mathbf{b}) \iff (\exists \mathbf{y} \quad : \mathbf{y}^T A \ge \mathbf{0}, \mathbf{y}^T\mathbf{b} < 0) . \tag{3}$$

Note that the direction "$\Longleftarrow$" is easy in each case. We will show the "$\Longrightarrow$" of (1) in class using a technique called *Fourier-Motzkin Elimination.* This exercise is actually not that hard. The hardest part is keeping track of what you want to prove and what you can assume.

Proof. First we prove (1) and (2) are equivalent. First consider (1) $\implies$ (2). If $\neg\exists \mathbf{x} \ge \mathbf{0} : A\mathbf{x} \le \mathbf{b}$, notice that $\mathbf{x} \ge \mathbf{0}$ iff $(-I_n)\,\mathbf{x} \le \mathbf{0}$, we can construct a new maxtrix A'

$$A' = \begin{pmatrix} -I \\ A \end{pmatrix}, \mathbf{b}' = \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}.$$

such that our assumption turns into

$$\neg \exists \mathbf{x} : A'\mathbf{x} \leq \mathbf{b}'.$$

And by (1), if follows that $\exists \mathbf{y}' \geq \mathbf{0}$ s.t. $(\mathbf{y}')^\top A' = 0$ and $(\mathbf{y}')^\top \mathbf{b}' < 0$. For convenience, we wirte $\mathbf{y}' = (z_1, z_2 \ldots, z_n, \mathbf{y})$ while $z_i \geq 0$, therefore

$$-z_i + \mathbf{y}^\top A_i = 0 \implies \mathbf{y}^\top A_i = z_i \geq 0.$$

Which leads that $\mathbf{y}^\top A \geq 0$. Similarly $(\mathbf{y}')^\top \mathbf{b}' = \mathbf{y}^\top \mathbf{b} < 0$, thus (2) is true.

Next we prove (2) $\implies$ (3). If $\neg \mathbf{x} \geq 0, A\mathbf{x} = \mathbf{b}$, which is $A\mathbf{x} \leq \mathbf{b}$ plus $(-A)\,\mathbf{x} \leq -\mathbf{b}$, similarly we construct

$$A' = \begin{pmatrix} A \\ -A \end{pmatrix}, \mathbf{b}' = \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}.$$

By (2), $\exists \mathbf{y}' = (\mathbf{y}_1, \mathbf{y}_2)$, such that

$$\mathbf{y}_1 A - \mathbf{y}_2 A \geq 0, \mathbf{y}_1 \mathbf{b} - \mathbf{y}_2 \mathbf{b} < 0.$$

Choose $\mathbf{y} = \mathbf{y}_1 - \mathbf{y}_2$ we are done.

Now we prove (2) $\implies$ (1).

We know that $\neg \mathbf{x} \geq 0, A\,(\mathbf{x}) \leq \mathbf{b}$. And it's also obvious that $\neg \exists \mathbf{x} \geq 0, A\,(-\mathbf{x}) = (-A)\,\mathbf{x} \leq \mathbf{b}$, we can build a new matrix $A' = (A, -A)$, hence

$$\neg \mathbf{x}' = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \geq 0, (A, -A) \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \leq 2\mathbf{b}.$$

By (2) we have

$$\exists \mathbf{y}, \mathbf{y}^\top A \geq 0, \mathbf{y}^\top (-A) \geq 0, \mathbf{y}^\top \mathbf{b} < 0 \implies \mathbf{y}^\top A = 0.$$

Thus (2) $\implies$ (1) is true.

Finally we prove (3) $\implies$ (2), if $\neg \exists \mathbf{x} \geq 0 : A\mathbf{x} \leq \mathbf{b}$, similarly construct

$$A' = (A, I).$$

We know $\neg \exists \mathbf{x} \geq 0, A'\mathbf{x}' = \mathbf{b}$. The result follows directly from (3). $\square$

7.2 A Linear Program for, well, for what?

Let $G = (V, E)$ be a directed graph, $s, t \in V$, and $c : E \to \mathbf{R}^+$ be a cost function. We want to find an $s - t$-flow f of value 1. Every edge e generates cost $f(e) \cdot c(e)$, and we want to minimize the overall cost. There are no capacity constraints. We can easily write this as a linear program MCF (Minimum Cost Flow):

$$\text{MCF}(G, s, t, c): \quad \begin{array}{ll} \text{minimize} & \sum_{e \in E} c(e) f(e) \\ & \\ \text{subject to} & \sum_{v \in V} f(v, t) = 1 \\ & \sum_{u \in V} f(u, v) - \sum_{w \in V} f(v, w) = 0 \quad \forall\, v \in V \setminus \{s, t\} \\ & \\ & f(e) \geq 0 \;\forall\; e \in E \end{array}$$

Note that we have m variables, one variable $f(e)$ for each edge e. The first constraint says that the value of the flow should be 1. The other constraints say that the inflow at v should equal the outflow.

Exercise 2. Let d be the shortest path distance from s to t in the directed graph G, where distance means sum of the $c(e)$ along the path. Show that $\text{opt}(MCF) = d$. **Hint.** Make sure you show both $\leq$ and $\geq$.

Proof. To prove $opt(MCF) \leq d$, we just need to prove that the shortest path is a solution to MCF. We set $f(e) = 1$ along all edges in the shortest path, since there is only one path with flow 1, The constraints are obviously satisfied. So it is a solution of MCF, and its value is $1 * d = d$, so $opt(MCF) \leq d$

To prove $opt(MCF) \geq d$, we need to prove that all solutions of MCF is not better than d. We try to improve the value of all possible solutions to d.

Suppose we have a solution with x different $s - t$ path. Define $b(path)$ be the smallest flow in all edges of $path$, $d(path)$ be the length of $path$. Let sp be the shortest $s - t$ path. We do as follows, choose any path p besides sp, put $b(p)$ units of flow on p to sp. Repeat it until there is only 1 unit flowing through sp.

We need to show in each turn, $val(MCF)$ is not worse than previous and no constraints are broke. We fist prove $val(MCF)$ is not worse. In each turn, $val(MCF)' = val(MCF) + d * b(p) - d(p) * b(p)$, $d \leq d(p)$, so $val(MCF)' \leq val(MCF)$. As for the constraints, inflow of t remains to be 1 since we just move $b(p)$ units between two different paths. Flow constraints remains since we modify the flow in one path, which means we move inflow and outflow of a single vertex at the same time. Since we only have x different $s - t$ path, and the flow value on each path is finite, the process terminates. So $val(MCF) \geq d$

In all $val(MCF) = d$ $\square$

Exercise 3. Write down the dual of MCF. This will be a maximization problem. Don't use any matrix notation.

Proof. We introduce a dual coefficient $g_v, v \in V$. The dual program is:

- Maximize g_t, subject to:
- $g_v - g_u \leq c(u, v), \forall (u, v) \in E$
- $g_v \in \mathbb{R}, v \in V$

$\square$

Exercise 4. Interpret the dual. Show that it is the LP formulation of a "natural" maximization problem on G.

Proof. If we set $g_s = 0$, then the g_v can be thought as the cost of some $s-t$-path. Since each edge (u, v) must satisfy $g_v - g_u \leq c(u, v))$, we can not just choose the maximal $s-t$-path as solution. Under this constraint, we can see that the solution must at first be a **safe** path, so the program is actually the shortest path problem. $\square$

Exercise 5. Describe an optimal solution of the dual program.

Proof. The optimal solution is the shortest $s-t$-path sp, and for each vertex along the path, we must set $g_v = g_u$ for $(u, v), u \in sp, v \notin sp$ accordingly to $g_v - g_u$ constraints. $\square$