Linear Programming in SI152

Jinxi Xiao

March 2024

1 Some Words At the Beginning

As the title goes: this is my notebook for Linear Programming(LP) in the course SI152 at ShanghaiTech. Due to the schedule and plan of the course, not all aspects of linear programming are covered(in fact we have ignored quite a lot for the reason that LP is not as popular as other methods like gradient descent in computer science). As a result, I will only record some key notes prof. teaches in class, as well as some of my understandings of LP(thus this essay is more useful as a function of *review*).

In addition, I strongly suggest reading this book: *Introduction to Linear Optimization*[1], and I have learnt a lot from this book. Thus, I will directly mention some contents within it.

2 Introduction

The first thing to consider is what does *linear* mean. Maybe we can take a look at the general form of an optimization problem:

$$\begin{aligned} &\min_x \; f_0(x) \\ &s.t. \; f_i(x) \;\le\; 0, \quad i = 1, \; ..., \; m \\ &\qquad h_i(x) = 0, \quad i \;=\; 1, \; ..., \; n \end{aligned}$$

From my point of view, linear means that

$$f(ax + by) = af(x) + bf(y)$$

for all related equations. And recall that linear operations are closely related to matrices, we derive a general form for Linear Programming:

$$\begin{aligned} &\min_x \; c^T x \\ &s.t. \; Ax \;=\; b \\ &\qquad x \ge 0 \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n, \; A \in \mathbb{R}^{m \times n}, \; b \in \mathbb{R}^m$.

However, problems that we encounter in real life will not simply be in the standard form, which means that we have got to convert a problem into the standard form. Typically, there are two methods

- *Elimination of free variables:* Given an unrestricted variable x_j in a problem in general form, we replace it by $x_j^+ - x_j^-$,where x_j^+ and x_j^- are new variables on which we impose the sign constraints $x_j^+ \geq 0$ and $x_j^- \geq 0$. The underlying idea is that any real number can be written as the difference of two nonnegative numbers.

- *Elimination of inequality constraints:* Given an inequality constraint of the form

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i,$$

we introduce a new variable s_i and the standard form contains

$$\sum_{j=1}^{n} a_{ij} x_j + s_i = b_i,$$
$$s_i \geq 0.$$

Such a variable s_i is called a *slack* variable. Similarly, an inequality constraint $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$ can be put in standard form by introducing a *surplus* variable s_i and the constraints $\sum_{j=1}^{n} a_{ij} x_j - s_i = b_i, \quad s_i \geq 0$.

The above two methods indicate a very basic way of converting any problem into the standard form. For more details and examples, please see book [1] starting from page 5.

Strict inequality?

It would be easy to notice that x is required to be $x \geq 0$, and what if the constraint changes to $x > 0$? Well, the answer may become a bit more complex. In Section 3 we will see that if the polyhedron(set of constraints) is nonempty and bounded, the optimal solution is a vertex. If equality does not hold, we cannot guarantee that the optimal solution(vertex) exists. As a result, what we normally do is to add a small margin into the strict inequality, i.e. $x \geq \epsilon$, where ϵ is extremely small. Of course this is wrong mathematically, but very practical in real life.

3 Geometry of Linear Programming

It is suggested to look Chapter 2 in [1] for this part.
Only some of the key notes are recorded here.

3.1 Polyhedron

Definition 3.1 A polyhedron is a set that can be described in the form $\{x \in \mathbb{R}^n | Ax \geq b\}$, where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$.

Notice that polyhedra are convex sets, and they are in fact constraints of Linear Programming. And this also suggests that we do not consider strict inequalities.

3.2 Extreme points, verteces and basic feasible solutions

Definition 3.2 Let P be a polyhedron. A vector $x \in P$ is an extreme point of P if we cannot find two vectors $y, z \in P$ both different from x, and scalar $\lambda \in [0, 1]$, such that $x = \lambda y + (1 - \lambda)z$.

Definition 3.3 Let P be a polyhedron. A vector $x \in P$ is a vertex of P if there exists some c such that $c^T x < c^T y$ for all y satisfying $y \in P$ and $y \neq x$.

Definition 3.4 Consider a polyhedron P defined by linear equality and inequality constraints, and let x^* be an element of $\mathbb{R}^n$.

1. The vector x^* is a basic solution if:
   (a) All equality constraints are active;
   (b) Out of the constraints that are active at x^*, there are n of them that are linearly independent.
2. If x^* is a basic solution that satisfies all of the constraints, we say that it is a basic feasible solution(BFS).

All three concepts are equivalent in the context of Linear Programming.

Adjacent basic solutions

Two distinct basic solutions to a set of linear constraints in $\mathbb{R}^n$ are said to be adjacent if we can find $n - 1$ linearly independent constraints that are active at both of them. If two adjacent basic solutions are also feasible, then the line segment that joins them is called an edge of the feasible set.

3.3 Something of polyhedron A

Recall in standard form 1, we have $Ax = b$. And not it's time to discuss something necessary of $A \in \mathbb{R}^{m \times n}$.

- We assume that A is in full rank, that is, we will think of it in a very concise form without duplicate constraints.

- We assume $m < n$. Otherwise, the polyhedron of this LP problem is empty or contains only one vertex. Then what we need to do is to find such vertex or report that the problem is infeasible.

3.4 Degeneracy

Definition 3.5 A basic solution $x \in \mathbb{R}^n$ is said to be degenerate if more than n of the constraints are active at x.

A very easy way to understand this is to think of the situation where some linear equalities also leads to $x_j = 0$. And this will make Simplex methods difficult to iterate. However, we will not consider degeneracy in the following sections!

3.5 Optimality

Corollary 3.1 Every nonempty bounded polyhedron and every nonempty polyhedron in standard form has at least one basic feasible solution.

Corollary 3.2 Consider the linear programming problem of minimizing $c^T x$ over a nonempty polyhedron. Then, either the optimal cost is equal to $-\infty$ or there exists an optimal solution.

4 Simplex Method

Also, please read Chapter 3 in [1].

4.1 General Idea

From Section 3, we know that optimal solutions are located at verteces, thus we need to find out verteces(basic feasible solutions). Then it is the problem of solving linear system $Ax = b$. Since there are more unknowns than equations in A, we need to set some inequalities active, i.e. there are $n - m$ of x to be zero. In total, there are $N = \binom{n}{m}$ possible choices, thus there are at most N verteces. Well, up to now, a brute-force method can be found: iterate N BFS one by one and find the minimum. However, this method is too naive and takes too much time!

Can we be smarter? Yes, we can design a strategy to find the next BFS. To be more specific, we can find a direction and search for the next **adjacent** BFS. Then if a certain BFS satisfies some requirements, we can say that this is the optimal one.

Suppose that we have selected m out of n x to be zero, and the corresponding basis matrix is $B \in M_m$ with basis index set $I = \{B_{(1)}, ..., B_{(m)}\}$ Then rewrite $A = [B, N], \quad x^T = [x_B, 0]^T, \quad c^T = [c_B, c_N]^T$ with objective function

$$c^T x = c_B^T x_B$$

and constraints

$$Bx_B = b \Rightarrow x_B = B^{-1}b$$

Consequently, we have obtained the first BFS $x^0 = [x_B, 0]$. In order to move to the adjacent BFS, we define a direction $d \in \mathbb{R}^n = [d_B, 0, ..., 1, ..., 0]^T$ where the j-th entry in the non-basis index set to be 1 and elsewhere in the non-basis index set to be 0 with a step-length $\theta \in \mathbb{R} > 0$. To satisfy the equality constraints, we have

$$A(x + \theta d) = b \Rightarrow Ad = 0 \Rightarrow Bd_B + A_j = 0 \Rightarrow d_B = -B^{-1}A_j$$

where A_j is the j-th column of N. Again, find the objective to be

$$c^T x = [c_B, ..., c_j, ...]^T [x_B + \theta d_B, ..., \theta, ...] = c_B^T x_B + \theta(c_j + c_B^T d_B)$$

Notice that we want to find the minimum objective, we want **reduced cost** $\bar{c}_j$ such that $\bar{c}_j = c_j + c_B^T d_B$ to be negative. As a result, we normally choose index j where $\mathbf{c_j < 0}$. In fact, we can prove further:

Theorem 4.1 Consider a basic feasible solution x associated with a basis matrix B, and let $\bar{c}$ be the corresponding vector of reduced costs.

1. If $\bar{c}^T = c^T - c_B^T B^{-1} A \geq 0$, then x is optimal. [a]

2. If x is optimal and nondegenerate, then $\bar{c} \geq 0$.

[a] For any index $i \in I$, we have $c_i - c_B^T B^{-1} A_i = c_i - c_B^T e_i = 0$, where e_i is a standard basis vector for $\mathbb{R}^m$ with only 1 on the i-th entry and elsewhere 0.

Consequently, we need to find the value for θ. Since costs decrease along the direction d, it is desirable to move as far as possible.

- If $d_B \geq 0$, then we can set $\theta \to +\infty$ (since $x_B + \theta d_B$ is always nonnegative), which indicates that the polyhedron is unbounded. And notice that the current objective would be $c_B^T x_B + \theta \bar{c}_j \to -\infty$.

- If $d_i < 0$ for some i, the constraint $x_i + \theta d_i \geq 0$ becomes $\theta \leq -x_i/d_i$. This constraint on θ must be satisfied for every i with $d_i < 0$. Thus, the largest possible value of θ is

$$\theta^* = \min_{\{i|d_i<0\}} (-\frac{x_i}{d_i})$$

  Recall that if x_i is a nonbasic variable, then either x_i is the entering variable and $d_i = 1$, or else $d_i = 0$. In either case, d_i is nonnegative. Thus, we only need to consider the basic variables and we have the equivalent formula

$$\theta^* = \min_{\{i=1,...,m|d_{B(i)}<0\}} (-\frac{x_{B(i)}}{d_{B(i)}})$$

  Note that $\theta^* > 0$, because $x_{B(i)} > 0$ for all i, as a consequence of nondegeneracy.

Up to now, we have complete a general process of Simplex method. A pseudo code for this trivial implementation can be found at page 90 in [1].

Why do we ignore degeneracy?

Let's consider an index $l \in I$ such that $x_l = 0$. And if the l-th value of corresponding direction $d_{B(l)} < 0$, for any $\theta > 0$, we will make $x_l = \theta d_{B(l)} < 0$. Thus a failure in iteration! As a result, we will ignore such kind of situation.

4.2 Full Tableau

To make life easier, we can write down the linear programming problem into a tableau, as the following:

$-c_B^T x_B$	$c - c_B^T B^{-1} A$
$B^{-1}b$	$B^{-1}A$

or to be more specific:

$-c_B^T x_B$	0	$\cdots$	0	$\cdots$	0	$\cdots$	$\bar{c}_j$	$\cdots$
$x_{B(1)}$	1							
$\vdots$		$\ddots$					$B^{-1}A_j$	
$x_{B(l)}$			1					
$\vdots$				$\ddots$				
$x_{B(m)}$					1			

Notice that when initializing the table, we **must** make $\mathbf{B^{-1}b} > \mathbf{0}$ (if nondegenerate, else ≥ 0 is sufficient), otherwise the solution is infeasible. And we do row operations based on the pivot row, say $B(l)$, only the $B(l)$-th entry on the zeroth row will be affected. Thus just as what we expected. For more details, read pages 98-105 in [1]. And this is the typical method to remember for **hand calculations**.

4.3 Complexity

Assume that there are n unknowns and m equations. Generally, the average time complexity is within polynomial time and simplex algorithm performs quite well in many real problems. However, the worst case of simplex is $O(n^2 2^n)$, the construction of the problem is [2]. In the worst case, we have to go through every vertex, which will give $\binom{2n}{n}$ choices (a lower bound and a upper bound can be found at [3]). In sum, the worst case of simplex is the brute-force algorithm.

5 Duality

Also, please read Chapter 4 in [1].

The first time we faced the problem of duality was back in the course of *Signal and Systems*. In fact, for most optimization problems, we can construct the dual problems of the primal ones.

5.1 A Trivial Visit to Lagrange Dual

Consider a general optimization problem with **linear** objective and constraints:

$$\begin{aligned}&\min_x \ f_0(x)\\ &s.t.\ f_i(x) \ \le \ 0, \quad i=1,\ ...,\ m\\ &\quad\ h_i(x)=0, \quad i \ = \ 1,\ ...,\ n\end{aligned}$$

Construct the Lagrange dual function:

$$L(x,\lambda,\nu)=f_0(x)+\sum_{i=1}^{m}\lambda_i f_i(x)+\sum_{j=1}^{n}\nu_j h_j(x), \quad \lambda_i\ge 0$$

Define $g(u)=\inf_x L(x,u)$ (u stands for λ,ν), and notice that

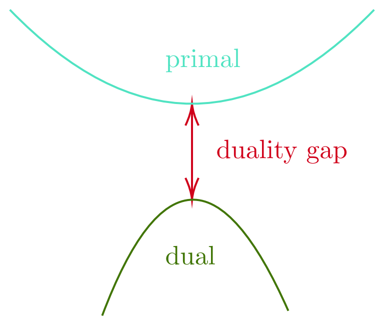
$$\begin{aligned}g(\theta a \ + \ (1-\theta)b) &= \ \inf_x \ L(x,\theta a \ + \ (1-\theta)b)\\ &= \ \inf_x[\theta L(x,a) \ +(1-\theta)L(x,b)]\\ &\ge \ \inf_x[\theta L(x,a)]+\inf_x[(1-\theta)L(x,b)]\\ &= \ \theta g(a)+(1-\theta)g(b)\end{aligned}$$

we can claim that $g(u)$ is concave, thus exists a maximum value $g(u^*)$.

Notice that $L(x,\lambda,\nu)\le f_0(x)$ for every x, and if we define x^* to be the optimal solution(if exists) , we can find that

$$g(\lambda,\nu)=\inf_x L(x,\lambda,\nu)\le g(u^*)\le f_0(x^*)\le f_0(x)$$

The aim of LP is to find the minimum value of the objective. By constructing the Lagrange dual function, it seems that we have found a lower bound for the objective, and also an upper bound for the dual function. And notice that the objective is convex, we can draw something like the following:



and if we are lucky enough to find the gap $f_0(x^*)-g(u^*)=0$, we can say that the two optimization problems are equivalent. And to be more specific, if the gap is zero, we have to make $\sum_{i=1}^{m}\lambda_i f_i(x)$ to zero.

5.2 Duality for LP

Recall the standard form of LP:

$$\begin{aligned}\min_x \ & c^T x \\ s.t. \ Ax &= b \\ -\mathbf{x} &\leq \mathbf{0}\end{aligned}$$

Construct the Lagrange dual as:

$$L(x, \lambda, \nu) = c^T x - \lambda^T x + \nu^T (b - Ax) = (c^T - \lambda^T - \nu^T A)x + \nu^T b$$

where $\lambda \geq 0$, and notice that

$$\inf_x L(x, \lambda, \nu) = \begin{cases} \nu^T b, & c^T - \lambda^T - \nu^T A = 0 \\ -\infty & \text{otherwise} \end{cases}$$

we can conclude the dual problem as:

$$\begin{aligned}\max_\nu \ & \nu^T b \\ s.t. \ & \lambda^T = c^T - \nu^T A \geq 0\end{aligned}$$

How to understand this dual problem?

We can find some nice explanations in [4]. Personally, I will give some explanations of mine in later sections. And I do believe to fully answer this question requires some knowledge of the concept of *dual* mathematically. I have mentioned that there are dual transformations in *Signal and Systems*, and their ideas are quite similar as those here. In sum, *duality* is a universal property just like symmetry.

Theorem

Theorem 5.1 (Weak duality) If x is a feasible solution to the primal problem and p is a feasible solution to the dual problem, then $p^T b \leq c^T x$.

Theorem 5.2 (Strong duality) If a linear programming problem has an optimal solution, so does its dual, and the respective optimal costs are equal.

Theorem 5.3 (Complementary slackness) Let x and p be feasible solutions to the primal and the dual problem, respectively. The vectors x and p are optimal solutions for the two respective problems if and only if:

$$\begin{aligned} p_i(a_i^T x - b_i) &= 0, \quad \forall i \in \{1, 2, ..., m\}, \\ (c_j - p^T A_j)x_j &= 0, \quad \forall j \in \{1, 2, ..., n\}. \end{aligned}$$

Let's take a closer look at the complementary slackness. If we define $u_i = p_i(a_i^T x - b_i)$ and $v_j = (c_j - p^T A_j)x_j$, we will find that $u_i \geq 0$, $v_j \geq 0$, and most importantly,

$$c^T x - p^T b = \sum_i u_i + \sum_j v_j \geq 0$$

which vividly explains why these terms should be zero.

In fact there's a more simple way to understand complementary slackness. Recall the primal and the dual problem

$$\begin{aligned} \min_x \ & c^T x \\ s.t. \ & Ax = b \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max_{p,\ s} \ & p^T b \\ s.t. \ & A^T p + s = c \\ & s \geq 0 \end{aligned}$$

and assume that there is an optimal basic solution x^* for primal problem, then we also find p^*, s^* such that the objective of dual problem reaches its optimal value. Thus, we can find out

$$\begin{aligned} c &= A^T p^* + s^* \\ b &= Ax^* \end{aligned}$$

which gives us

$$c^T x - p^T b = s^* x^*$$

notice that $s, x \geq 0$, therefore, we can imply that if $x_j \neq 0$, then s_j **must** be zero, and vice versa!

5.3 Duality in Simplex Tableau

Let's recall what's the strategy of simplex method: we iterate through some BFS with a right direction and step-length, starting from an initial BFS. This requires selected basis matrix to satisfy two requirements:

$$B^{-1} b \geq 0 \tag{2}$$

and

$$\bar{c} = c - c_B^T B^{-1} A \geq 0 \tag{3}$$

Then from **primal problem**'s view, in each step, primal simplex requires Equation 2 holds, means that in each iteration the solution must be feasible, while dual simplex requires the solution to be optimal(3), since $\nu^T A \leq c$ can be interpreted as $c_B^T B^{-1} A \leq c$. In other words, we make **zeroth row** to be satisfied in primal simplex, and make **zeroth column** to be satisfied in dual simplex. Notice that columns and rows are just the two basic elements of a matrix, ~~which makes me feel that any matrix is dual with respect to its transpose~~.

6 Sensitivity Analysis

Go and see Chapter 5.1 and 5.2 in [1], it is wonderful!

7 Interior Point Method

I DIDN'T GET IT, I WAS JUST IN AWE.

At present, I can not really tell that I have understood this method. But I would recommend some resources to be used: [5], [6]. And maybe I will update this section later some day.

References

[1] D. Bertsimas and J. N. Tsitsiklis, "Introduction to linear optimization," in *Athena scientific optimization and computation series*, 1997.

[2] "Klee–minty cube." `https://en.wikipedia.org/wiki/Klee%E2%80%93Minty_cube`.

[3] "C(2n,n)." `https://www.zhihu.com/question/494176389`.

[4] "Why to consider dual problem of lp." `https://www.zhihu.com/question/26658861`.

[5] "Zhiqin xu's tutorial on bilibili." `https://www.bilibili.com/video/BV1ST4y1u7Sk/?spm_id_from=333.337.search-card.all.click`.

[6] "A blog on zhihu which explans interior point method." `https://zhuanlan.zhihu.com/p/528000562`.