Optimization for Data Science

F. Rinaldi[1]

[1]  Università degli Studi di Padova | Dipartimento Matematica

Padova
2020

Outline

Optimization for Data Science

Linear Program

Definition

A *linear program* is an optimization problem where we seek to

- maximize/minimize a linear function;
- over a set of equality and/or inequality constraints.

A linear program in general form can hence be described as follows:

$$\begin{aligned} \min \text{ (or } \max) \quad & c_1 x_1 + \cdots + c_n x_n && (1)\\ \text{subject to} \quad & a_{11} x_1 + \cdots + a_{1n} x_n \sim b_1 && (2)\\ & \vdots \qquad\qquad \vdots \\ & a_{m1} x_1 + \cdots + a_{mn} x_n \sim b_m, && (3) \end{aligned}$$

where each "$\sim$" stands for one of the following operators "$\geq$", "$\leq$" or "$=$" (notice that no strict inequalities are allowed).

Basic Definitions

In problem (1)–(3),

- $x_1, \ldots, x_n$ are the *variables*;
- (1) is the *objective function*;
- (2)–(3) are the *constraints*;
- all data a_{ij}, b_i, c_j ($i = 1, \ldots, m, j = 1, \ldots, n$) are real numbers;
- variables $x_1, \ldots, x_n$ can assume any real value, and need to satisfy all constraints;
- Please notice that (1) can be rewritten as $c^\top x$.

Basic Definitions II

Closed half-space in $\mathbb{R}^n$

It is a set having the form $\{x \in \mathbb{R}^n : \ a^\top x \leq \beta\}$ where $a \in \mathbb{R}^n \setminus \{0\}$ and $\beta \in \mathbb{R}$. If we write it explicitly, we have $a_1 x_1 + \cdots + a_n x_n \leq \beta$.

Hyperplane in $\mathbb{R}^n$

It is a set having the form $\{x \in \mathbb{R}^n : \ a^\top x = \beta\}$ where $a \in \mathbb{R}^n \setminus \{0\}$ and $\beta \in \mathbb{R}$.

Polyhedron

It is the intersection of a finite number of closed half-spaces. Equivalently a polyhedron is a set P that can be written in the following form:

$$P = \{x \in \mathbb{R}^n : Ax \leq b\},$$

with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Connection between Polyhedra and Convex Sets

Proposition

The following properties hold:

1. every closed half-space is convex;
2. intersection of convex sets is a convex set;
3. every polyhedron is convex.

- A linear programming problem is also a convex problem.
- Results related to convex problems hold.
- When dealing with a polyhedron, an extreme point is usually called *vertex*.

Fundamental Theorem

Fundamental Theorem of Linear Programming

Given a linear program, one of the following conditions holds:

(a) the problem has an optimal solution (which is a vertex of the polyhedral feasible set) at least;

(b) problem is *unfeasible*,i.e., there are no feasible solutions;

(c) problem is *unbounded*, i.e., for all $K \in \mathbb{R}$ there exists a feasible x such that $c^\top x < K$ ($c^\top x > K$ if is a maximization problem).

Equivalence between Problems

- From now on, given a matrix $A \in \mathbb{R}^{m \times n}$ and a set of indices $I \subseteq \{1, \ldots, n\}$, we indicate with A_I the sub-matrix of A whose columns have indices in I.
- Given a vector $x \in \mathbb{R}^n$, x_I indicate the sub-vector whose components x_i are such that $i \in I$.
- Given a matrix $A \in \mathbb{R}^{m \times n}$ with components a_{ij}, we indicate with $a_i^\top$ the i-th row of A and with A_j the j-th column of A.

Equivalence between Problems

Given a feasible solution to one problem, we can construct a feasible solution for the other, with the same cost.

Standard Form

Equivalence with Standard Form Problems

Now we will see that every linear program is equivalent to a linear program in *standard form*, that is

$$\min \; c^\top x \tag{4}$$

$$\text{s.t. } Ax = b \tag{5}$$

$$x \geq \mathbf{0}, \tag{6}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and x is a vector of variables $\mathbb{R}^n$.

Equivalent Transformations I

Inequality constraints elimination

Given an inequality of the form

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \tag{7}$$

we introduce a new variable s_i and we get the new constraints

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j + s_i &= b_i, \\ s_i &\geq 0. \end{aligned} \tag{8}$$

Such a variable is called *slack* variable.

- Easy to see that if x satisfies (7), then setting $s_i = b_i - \sum_{j=1}^{n} a_{ij}x_j$ conditions in (8) are satisfied.
- Viceversa, if x and s_i satisfy conditions (8), easy to see that constraint (7) is also satisfied.

Equivalent Transformations II

Inequality constraints elimination II

Similarly, we can consider an inequality of the form

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

we introduce a new variable s_i and we get the new constraints

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j - s_i &= b_i, \\ s_i &\geq 0. \end{aligned} \tag{9}$$

Such a variable is called *surplus* variable.

- Equivalence: same as slack variable case.

Equivalent Transformations III

Free variables elimination

Given a free (or unrestricted) variable x_j, we can replace it with the term

$$x_j = x_j^+ - x_j^-$$

and impose the signs $x_j^+ \geq 0, x_j^- \geq 0$.

- Main idea: any real number represented as sum of its positive and negative part.
- given a feasible solution of the new problem, by simply setting $x_j = x_j^+ - x_j^-$ for all the unrestricted variables, we obtain a feasible solution for the original problem (with same cost).
- consider a solution of the original problem, we can just build up a feasible solution (with same cost) for the new problem by setting $x_j^+ \geq 0, x_j^- \geq 0$ in such a way that

$$x_j = x_j^+ - x_j^-,$$

a simple example is $x_j^+ = \max\{0, x_j\}$ and $x_j^- = \max\{0, -x_j\}$.

Examples of Transformations

Example I

$$\begin{array}{lrcrcl} \min & 2x_1 & + & 4x_2 & & \\ s.t. & x_1 & + & x_2 & \geq & 3 \\ & 3x_1 & + & x_2 & = & 10 \\ & x_1 & & & \geq & 0 \end{array}$$

Equivalent standard form problem

It is equivalent to the following problem in standard form:

$$\begin{array}{lrcrcrcrcl} \min & 2x_1 & + & 4x_2^+ & - & 4x_2^- & & & & \\ s.t. & x_1 & + & x_2^+ & - & x_2^- & - & s_1 & = & 3 \\ & 3x_1 & + & x_2^+ & - & x_2^- & & & = & 10 \\ & x_1, & & x_2^+, & & x_2^-, & & s_1 \geq 0. & & \end{array}$$

Standard Form

- From now on we will only consider problems in standard form.
- If general problem in standard form is feasible, then rows in A are linearly independent (thus we get $m \leq n$).
- In standard form problems we assume A rows are linearly independent.

Basis

A subset of indices $B \subseteq \{1, \ldots, n\}$ such that $|B| = m$ and the square matrix A_B is invertible or non-singular (Notice that, since rank of A is m, at least one matrix B exists).

Non-Basis

The complement of B is called *non-basis* and indicated with $N = \{1, \ldots, n\} \setminus B$. We also say that sub-matrix A_B is a basis of A.

Basic Solution

Useful Equalities

Given a basis B, the following equalities hold

$$Ax = b \iff A_B x_B + A_N x_N = b \iff x_B = A_B^{-1} b - A_B^{-1} A_N x_N. \tag{10}$$

Basic Solution

Basic solution related to basis B is the only solution $\bar{x}$ of system $Ax = b$ obtained fixing $\bar{x}_N = \mathbf{0}$.

- Uniqueness follows from (10), which shows that the following holds: $\bar{x}_B = A_B^{-1} b$.
- Notice that a basic solution might be unfeasible, since constraints (6) might be violated.
- Basic solution $\bar{x}$ is feasible if and only if $\bar{x}_B = A_B^{-1} b \geq \mathbf{0}$.

Basic Feasible Solutions and Vertices

- Connection between vertices and basic solution of a linear program in standard form.
- Possible to prove that a point $\bar{x}$ of a polyhedron in standard form is a vertex if and only if $\bar{x}$ is a basic (feasible) solution.

Remark

Using **Fundamental Theorem of Linear Programming**, we can focus on basic feasible solutions!!!

Comments

- IDEA: focus only on basic feasible solutions when looking for an optimum of a linear program.
- Linear programming is thus a classic topic in *discrete optimization* (even if the problem is mainly continuous).
- Problem having n variables and m constraints. There are at most

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

  basic solutions (corresponding to the number of ways of selecting m of n columns).
- only a finite number of possibilities!!!

A Brute Force Approach

Brute force approach

Fundamental theorem yields an obvious, but clearly inefficient, finite search technique:

- Generate all basic solutions.
- Choose the best one in terms of o.f. value.

Too many Basic solutions!!!

Better approach

Simplex procedure

- First proposed by G. Dantzig in 1947.
- It is one of the most popular methods for linear programming.
- We give here the basic ideas.

Simplex Algorithm

Phase I: Generate a basic (feasible) solution (if there are no feasible solutions, Phase I proves this fact);

Phase II:

1. At each iteration, move from the current basic (feasible) solution to another basic (feasible) solution, obtained replacing one index in basis with a new index not in the basis;
2. At each iteration, new basic solution has a value $c^\top x$ not worse than the previous one;
3. After a finite number of iterations the methods gives an optimal basic solution (or it gives proof that problem is unbounded).

Degeneracy

Degenerate solution $\bar{x}$

Number of positive components of $\bar{x}$ is smaller than m. In this case, we have that the corresponding basis might not be unique.

- We might generate *degenerate basic solutions*!
- Often they can be handled as a non-degenerate basic feasible solution.
- BAD CASE: after a new index is selected to enter the basis, the new basic feasible solution could also be degenerate (and with the same objective function value).
- Switching between degenerate solutions could continue until, finally, the original degenerate solution is again obtained!
- The result is a loop that could be repeated indefinitely!!!
- SOLUTION: There exist a few rules (like e.g. Bland's rule) that guarantee to avoid such loops in practice.

Complexity of the Simplex Method

We consider Classic Complexity Analysis here.

- Consider linear program in standard form (with A an $m \times n$ matrix, b and c respectively m and n vectors).
- Single iteration done in polynomial time (consists of simple operations over matrices and vectors).
- By means of the simplex method, The number of iterations needed to solve a linear program is usually a multiple of m (Average case).
- QUESTION: is the simplex method a polynomial time algorithm?
- ANSWER: worst-case complexity of simplex method (as formulated by Dantzig) is exponential time!!!
- In 1972, Klee and Minty gave an example, the Klee-Minty cube (i.e., a hypercube of variable dimension).
- For almost every variation on the method, we have exponential time worst case complexity.
- It is still an open question if there is a variation with polynomial time worst-case complexity.

Klee-Minty Cube

Klee-Minty Example

One form of the Klee-Minty example is

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} 10^{n-j} x_j \\ s.t. \quad & 2\sum_{j=1}^{n} 10^{n-j} x_j + x_i \leq 100^{i-j}, \quad i = 1, \ldots, n \\ & x_j \geq 0 \quad j = 1, \ldots, n \end{aligned}$$

This problem takes $2^n - 1$ standard simplex iterations (i.e., the number of vertices minus one, which is the starting vertex)!!!

- To get an idea of how bad this can be, consider the case where n = 50.
- We have $2^{50} - 1 \approx 10^{15}$.
- In a year, there are approximately $3 \cdot 10^7$ seconds.
- If a computer ran continuously, performing a million simplex iterations per second, it would take approximately **33 years** to get the job done!

Klee-Minty Cube

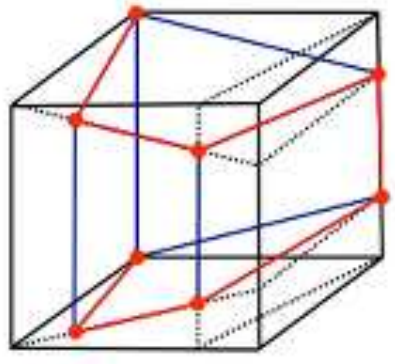

Figure: Example of Klee-Minty cube