Lecture 3: Linear Programming

Rajat Mittal

IIT Kanpur

We will foucs on the definition and concept of optimization. This lecture will introduce you to the field of linear programming with some examples.

1 Mathematical optimization

Optimization is a process of maximizing or minimizing a quantity under given constraints. Most of the problems in this world are optimization. You have to maximize (happiness/peace/money) or minimize (poverty, grief, wars etc.). Unfortunately, we are not solving any of those problems.

On a smaller scale, there are many real world problems where we need to optimize mathematical quantities and the constraints can also be represented as mathematical functions. For example, optimizing time in the production cycle of an industry, optimizing tax in a tax-return, optimizing length in a tour are mathematical optimization problems we encounter in our daily life.

Formally, any problem of the form:

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \; & f_i(x) \leq b_i \; i = 1, 2, \cdots, m \end{aligned}$$

is called a mathematical optimization problem. Here f_0 is the objective/optimization function and $f_i \leq b_i$ are called constraints. The task here is to find the max/min value of $f_0(x)$, s.t., x satisfies all the constraints.

An x satisfying all the constraints is called a *feasible solution*. The set of x's satisfying all the constraints is called the *feasible region*.

$$S_f = \{x : f_i(x) \leq b_i \;\; \forall\, i \in [m]\}$$

A feasible solution x^* is called an *optimal* solution if it has the smallest objective value among all the feasible solutions. So, for any feasible $z \in S_f$, we know that $f_0(z) \geq f_0(x^*)$. Let's consider some examples,

– take our/mine favorite example of world peace: x-actions, f_0- peace function, f_i's - there are many (for example, don't kill everyone/anyone),
– satisfiability: given a boolean satisfiability formula, $(x_1 \vee x_2 \vee x_4), (\bar{x_2} \vee x_4 \vee x_1), \cdots,$

$$\begin{aligned} & \max \# \text{ of clauses satisfied} \\ & \text{s.t.} \qquad x \in \{0,1\}^n \end{aligned}$$

– portfolio optimization: Every variable represents amount spend in each asset. Constraints might be on budget/availability/expected return. Objective is to minimize risk,
– data fitting: find some model from some class of models which fit the data. What are the constraints, objective function, variables?

Note 1. Remember that optimal solution need not be unique. One special case is: when variables have symmetry, in this case some kind of permutation can be applied to get multiple optimal solutions.

1.1 Classes of optimization problems

It is quite clear from the previous discussion that general optimization problems seem to be really hard. Hence, we are interested in classes of optimization problems which can be solved easily or have specific properties. These different classes differ in the kind of constraints and objective functions that are allowed to be included in these problems. One example is Linear programming, where constraints and objective function have to be linear.

A natural question might be, What kind of classes should be studied? A class of problems is interesting if:

- Many real world problems can be modeled as a problem in that class.
- Problems in the class are *easily/efficiently* solved.
- Problems in the class have nice properties (e.g., duality), which can give us more information about the structure, solution of the problem (this will become clear later).

Linear programming satisfies all the above properties and hence a natural candidate to be studied. Our emphasis will be to understand why linear programming can be solved efficiently and see some applications of them in the field of theoretical computer science. Linear programming solvers (like simplex, interior point method) will not be covered in this course.

Convex optimization is a generalization of linear programming where the constraints and objective function are convex. It is interesting because most of the algorithms for linear programming can be generalized to convex optimization too. More importantly, many more problems can be expressed in this framework than linear programming. Many subclasses of convex optimization like semidefinite programming and least square problem are also widely used and have important applications in various fields. Later, we will study semidefinite programming in detail.

I would like to emphasize that very simple additional constraints can make the these problems hard. One of those constraints which is notorousily difficult is, when the variables are restricted to be integers. These are called *integer programming problems*.

Exercise 1. Show that the minimum weight vertex cover problem can be formulated as an LP with additional constraint that variables belong to $\{0, 1\}$.

2 Linear Programming

Linear programming is one of the well studied classes of optimization problem. We already discussed that a linear program is one which has linear objective and constraint functions. A linear constraint is a linear expression with equalities or inequalities.

Exercise 2. What is a linear expression?

This implies that a linear program looks like

$$\begin{aligned} \min \sum_i c_i x_i & \\ \text{subject to } a_i^T x_i \leq b_i & \quad \forall i \in \{1, \cdots, m_1\} \\ a_i^T x_i \geq b_i & \ \forall i \in \{m_1 + 1, \cdots, m_2\} \\ a_i^T x_i = b_i & \ \ \forall i \in \{m_2 + 1, \cdots, m\} \end{aligned}$$

Here the vectors $c, a_1, \cdots, a_m \in \mathbb{R}^n$ and scalars $b_i \in \mathbb{R}$ are the problem parameters. Notice that $\sum_i c_i x_i$ can also be written as $c^T x$ in vector notation.

Let us take an example. In the *max flow* problem, we are given a graph, start(s) and end node (t), capacities on every edge. We need to find out the maximum flow possible through edges.

The linear program looks like:

$$\begin{aligned} \max \quad & \textstyle\sum_{\{s,u\}} f(s, u) \\ \text{s.t.} \ & \textstyle\sum_{\{u,v\}} f(u, v) = \sum_{\{v,u\}} f(v, u) \ \forall v \neq s, t \\ & 0 \leq f(u, v) \leq c(u, v) \end{aligned}$$

Note 2. There exist another linear program for the same problem, which can be made using the flow through paths.

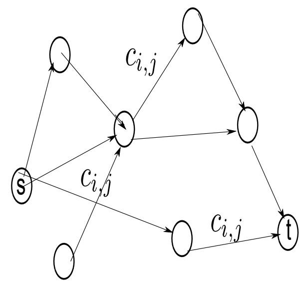

Fig. 1. Max flow problem: there will be capacities for every edge in the problem statement

2.1 Solving linear programs

You might have already had a course on linear optimization. So you might know that there are many known algorithms for solving linear programs: like simplex, ellipsoid and interior point method. Simplex method was one of the first methods to solve these programs. But almost all initial versions have examples which will take too long (exponential time) to solve. It is an open question if some version of simplex can run in polynomial time for all the instances. Since it is efficient in practice, it is used in many places.

The first polynomial time algorithm was Ellipsoid algorithm. It is not found to be very efficient in practice. Few years later, interior point method was developed and shown to be in polynomial time. Since it is efficient in practice and is provably fast, it is implemented in a lot of places.

Because of the abundance of algorithms to solve linear programs, researchers were really excited about this paradigm. There were many attempts to solve even NP hard problems (like traveling salesman problem) using linear programming. Notice that this will prove one of the most fundamental questions of complexity theory, P=NP. This is because we know that linear programs can be solved in polynomial time.

Recently there was a big result by Wolf et. al., where they showed that most of these techniques are bound to fail. They showed that the traveling salesman polytope or its extension will require exponential number of constraints.

2.2 Another example of a linear program

Suppose we have a company which makes two kinds of laptop, Apple and Dell. Every Apple gives a profit of 10 Rs. and every Dell 5 Rs.. It is clear that to maximize the profit the company should make as many Apple computers as possible (assuming they can sell everything they build).

Though, life is not so simple, every Apple computer takes 20 people to build, on the contrary Dell just takes 13. Similarly, an Apple needs 4 chips, but the Dell needs only 1. At any particular day, the company has at most 95 people and 28 chips for their disposal. How many Apple's and Dell's should the company make? This problem is known as *resource allocation problem.*

From the mathematical point of view, the problem is quite clear,

$$\begin{aligned} \max \quad & 10x_1 + 5x_2 \\ \text{s.t.} \quad & 20x_1 + 13x_2 \leq 95 \\ & 4x_1 + x_2 \leq 28 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Here, x_1 is the number of Apple's and x_2 is the number of Dell's. In a real scenario, we want these to be integers. Let's not worry about this constraint yet, though we have seen that these constraints (that variables should be integer) make certain problems really hard.

In any case, the above optimization approach can be generalized to the following resource allocation problem.

Suppose, a manufacturing unit wants to produce items $i = 1, \cdots, n$ using raw materials $j = 1, \cdots m$. The cost of raw material j is γ_j and the price of item i is ρ_i. There is only b_j amount of raw material j available. If a single unit of item i requires a_{ij} amount of raw material j, the manager's job is,

$$\begin{array}{rl} \max & \sum_i (\rho_i - \sum_j a_{ij}\gamma_j) x_j \\ \text{s.t.} & \forall\, j \quad \sum_i a_{ij} x_i \le b_j \\ & \forall\, i \quad x_i \ge 0. \end{array}$$

Notice that $\rho_i - \sum_j a_{ij}\gamma_j$ can be thought of as the profit for item i, we call it c_i. Suppose c is the vector with co-ordinates c_i, x with co-ordinates x_i and so on, then

$$\begin{array}{rl} \max & c^T x \\ \text{s.t.} & \forall\, j \quad a_j^T x \le b_j \\ & x \ge 0. \end{array}$$

Let us look at the same resource allocation problem from a pessimist's point of view. Suppose, he wants to assign some cost y_j to every raw material so that the cost of his inventory is minimized (for budget purposes). Though the catch is, he should be willing to sell the raw material at the same price to some other competitor manufacturing unit.

These constraint imply, his assigned cost should not be smaller than the market price, $y_j \ge \gamma_j$ (else the competitors can directly buy from him instead of market) and also

$$\forall\, i \quad \sum_j a_{ij} y_j \ge \rho_i$$

Otherwise, the competitor can buy the raw material from our unit and make the items cheaper than the market price. Hence, the problem becomes,

$$\begin{array}{rl} \min & \sum_j b_j y_j \\ \text{s.t.} & \forall\, i \quad \sum_j a_{ij} y_j \ge \rho_i \\ & \forall\, j \quad y_j \ge \gamma_j. \end{array}$$

If we make a change of variable here $z_j = y_j - \gamma_j$, the life will be much simpler,

$$\begin{array}{rl} \min & \sum_j b_j z_j \\ \text{s.t.} & \forall\, i \quad \sum_j a_{ij} z_j \ge c_i \\ & \forall\, j \quad z_j \ge 0. \end{array} \tag{1}$$

You can see that both problems look similar. Definitely both have linear objective and constraint functions. Let's make this precise.

3 Standard format

A linear program is an optimization problem where both the constraints as well as objective function is linear in the variables to be optimized. Through this definition there can be inequalities, equalities or different signs on the variables.

To make the future analysis and description simple, we assume a standard form. All other kind of linear programs can be converted into this standard form. The standard form of a linear program is,

$$\begin{aligned} &\max \quad c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

Where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and A is an $m \times n$ matrix. The constraint $Ax = b$ should be interpreted as, every entry of Ax is equal to the corresponding entry of b. It is almost clear that the resource allocation problem is a linear program (an LP) in standard form.

Exercise 3. Show that the optimization problem 1 can be converted into an LP in standard form. What are c, b, A now?

3.1 Converting one LP into another

We have been talking informally about we can convert any LP into standard form. Intuitively it means that both the LP's are equivalent. What does it mean mathematically? Suppose we are given two LP's L_1 and L_2, when are they equivalent?

Two LP's (L_1 and L_2) are equivalent iff

- Any optimal solution of L_1 can be converted into a feasible solution of L_2 with *same* objective value.
- Any optimal solution of L_2 can be converted into a feasible solution of L_1 with *same* objective value.

Note 3. The solutions for two LP's having the *same* value can be defined in various ways, e.g., one could be a simple monotone function of another.

For an example, suppose for input we have $x \in \{0,1\}^n$ and there are sets $C_1, C_2, \cdots, C_m \subseteq \{0,1\}^n$. Consider the LP,

$$\begin{aligned} &\max \quad \textstyle\sum_x u_x + v_x \\ \text{s.t.} \ \forall\, i \in [m] \quad & \textstyle\sum_{x \in C_i} u_x - v_x \leq |C_i| \\ \forall\, x : \quad & u_x, v_x \in \mathbb{R}. \end{aligned}$$

Observe that by change of variable $y_x = u_x + v_x$ and $z_x = u_x - v_x$, the LP becomes

$$\begin{aligned} &\max \quad \textstyle\sum_x y_x \\ \text{s.t.} \ \forall\, i \in [m] \quad & \textstyle\sum_{x \in C_i} z_x \leq |C_i| \\ \forall\, x : \quad & y_x, z_x \in \mathbb{R}. \end{aligned}$$

Now it is clear that value of z_x doesn't matter (we can set it to zero) and y_x can be raised as high as possible.

Exercise 4. Show that above two LP's are equivalent. What if in the first LP, we had constraint $u_x, v_x \geq 0$ for all x?

3.2 Other formats

Let's talk about how to convert different kind of linear constraints into the standard form.

- inequality into equality: Use extra non-negative variables.
- Inequality in the opposite direction: A constraint like $d^T x \geq e$ can be converted to $(-d^T)x \leq (-e)$.

  Exercise 5. What if input variable is less than zero?

- No constraint on input variable: If x_i is unconstrained, then $x_i = y_i - z_i$, where $y_i, z_i \geq 0$.

  Exercise 6. Show that the two LP's in this case would be equivalent in the sense described above.

- Strict inequalities: Not allowed in LP's. Instead we solve the approximate version with inequalities.
- We don't need to consider sup/inf and can only work with max/min. This can be justified using Fourier-Motzkin elimination.

4 Assignment

Exercise 7. What is the least square optimization problem? Read about it.

Exercise 8. How will you convert an equality constraint into the standard format?

Exercise 9. Show that every linear program can be converted into this kind of standard form.

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

Exercise 10. Look at the investment problem given at,
https://www.utdallas.edu/ scniu/OPRE-6201/documents/LP02-Investment.pdf.
Try to formulate its LP before looking at the solution.

Exercise 11. Consider a two player game with a matrix M (of dimension $n \times n$). The two players, call them row player and column player, have n strategies each. Row player gets an output M_{ij} when she plays strategy i and column player strategy j. We want to find probabilities $p_1, p_2, \cdots, p_n$ for row player which optimizes her output.

Show that this problem can be formulated as a linear program.

References

1. G. Strang. Linear Algebra and Its Applications. *Cengage learning*, 2007.
2. S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge*, 2004.
3. D. Spielman. Course notes: Spectral Graph Theory. *http://www.cs.yale.edu/homes/spielman/561/*, 2015.
4. L. Trevisan. Course notes: Graph Partitioning, Expanders and Spectral Methods. *https://people.eecs.berkeley.edu/ luca/expanders2016/index.html.*