Linear programming

Huy L. Nguyễn

In the course of our studies, we have encountered many optimizations problems: finding the shortest paths, finding the minimum spanning trees, finding the minimum cuts, etc. We can model the decisions we need to make using variables and the constraints of the problems as equations involving these variables. Amazingly enough, the objective and constraints of many problems can be expressed as a system of linear inequalities.

The feasible region (the set of points satisfying the linear inequalities) is called a *polytope*. The region is *convex*: for any two points x and y in it, the line segment connecting x and y lies entirely in the region.

In *linear programming*, the goal is to optimize (maximize or minimize) a linear objective function over the feasible region. The general form of a linear program is

$$\begin{aligned} \min c^T x \\ Ax \geq b \\ x \geq 0 \end{aligned}$$

Here $\geq$ denotes component-wise greater than or equal.

This general form can express all types of linear programs. To maximize rather than minimize an objective, simply negate the coefficients. To include an inequality such as $5x + y \leq 3$, rewrite it as $-5x - y \geq -3$. To include an equality $3x = y + 1$, rewrite it as two inequalities $3x - y - 1 \geq 0$ and $-3x + y + 1 \geq 0$. To include an unconstrained variable x, one can replace x with $x^+ - x^-$ where x^+, x^- are two new non-negative variables.

An important fact about linear programs is that they can be solved in polynomial time and in fact, there are efficient softwares for this task. We will mostly use this fact as a tool to solve our problems but in due time, we will also see a glimpse of the theory behind algorithms for linear programs.

1 Modeling tasks as linear programs

1.1 Assignment Problem

Suppose we have n jobs and n machines. Since the machines are different and jobs have different requirements (memory, disk space, CPU, etc), the running time of different jobs on different machines are different. Let $c_{i,j}$ be the running time of job i on machine j. We would like to assign one job per machine so as to minimize the sum of the processing times across all machines.

Let $x_{i,j}$ be the variable indicating whether we assign job i to machine j. We hope that this variable will be either 0 or 1 but that is not expressible in an LP so we relax it to the constraints

$$0 \leq x_{i,j} \leq 1 \ \forall i, j$$

Since each machine j can process 1 job, we have the constraint:

$$\sum_{i=1}^{n} x_{i,j} = 1 \ \forall j$$

Each job i is assigned to one machine so we have the constraint:

$$\sum_{j=1}^{n} x_{i,j} = 1 \ \forall i$$

Finally, we would like to minimize the total processing time:

$$\min \sum_{i,j} c_{i,j} x_{i,j}$$

Interestingly this linear program always has an integral optimal solution i.e. all variables $x_{i,j}$ are either 0 or 1. Thus, solving the LP actually gives the optimal assignment.

1.2 Shortest path

Suppose we have a graph $G = (V, E)$. The edge from i to j has weight $w_{i,j}$. We would like to find the shortest path from vertex s to vertex t.

We use a variable $x_{i,j}$ to indicate whether an edge (i, j) is used in the shortest path. Again we hope that it is either 0 or 1 but we need to relax it to the constraints

$$0 \leq x_{i,j} \leq 1 \ \forall (i, j) \in E$$

Exactly one edge going out of s must be used in the shortest path:

$$\sum_{(s,j)\in E} x_{s,j} = 1; \quad \sum_{(i,s)\in E} x_{i,s} = 0$$

Exactly one edge going into t must be used in the shortest path:

$$\sum_{(i,t)\in E} x_{i,t} = 1; \quad \sum_{(t,j)\in E} x_{t,j} = 0$$

For all other vertices u, the number of incoming and outgoing edges in the shortest path must be the same:

$$\sum_{(i,u)\in E} x_{i,u} - \sum_{(u,j)\in E} x_{u,j} = 0 \ \forall u \neq s, t$$

Interestingly this LP also has an integral optimal solution. Thus, solving the LP gives the shortest path from s to t.

1.3 ℓ_1 regression

Suppose you are trying to build a model to explain student's performance. The performance of a student in a course is a function of 1) the difficulty of the course, 2) the aptitude of the student, and 3) random noise. Thus, we hypothesize that the student i's grade in course j is modeled by the following equation:

$$G_{i,j} = a_i + d_j + \varepsilon_{i,j}$$

where a_i is the aptitude of student i, d_j is the difficulty of course j and $\varepsilon_{i,j}$ is a noise term.

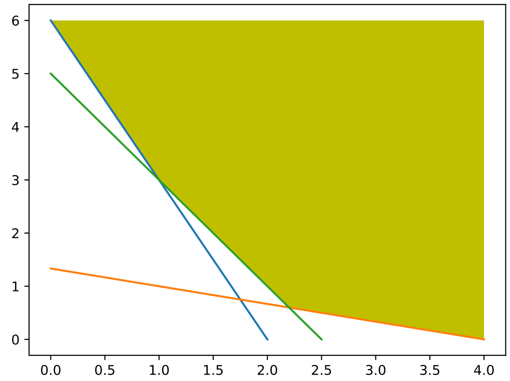
We would want the model to have as small noise as possible. Thus we have the following optimization problem, which is an LP.

$$\min \sum_{i,j} |\varepsilon_{i,j}| \ :$$
$$G_{i,j} = a_i + d_j + \varepsilon_{i,j} \ \forall i, j$$

2 Duality

An important question in linear programming is to be able to certify the optimality of the solution. Consider a small example:

$$\begin{aligned} &\min x + y \text{ subject to} \\ &3x + y \geq 6 \\ &x + 3y \geq 4 \\ &2x + y \geq 5 \\ &x, y \geq 0 \end{aligned}$$



If the software gives us the solution $x = 11/5, y = 3/5$, how good is this solution? This question is intimately connected to the question of giving an lower bound on the solution.

One possible lower bound comes from adding up 1/4 times the first inequality plus 1/4 times the second inequality:

$$x + y = \frac{1}{4}(3x + y) + \frac{1}{4}(x + 3y) \geq \frac{1}{4} \cdot 6 + \frac{1}{4} \cdot 4 = \frac{5}{2}$$

Question: can you find a better lower bound?

This question brings us to the concept of duality. For each linear program, there is an associated dual linear program:

Primal	**Dual**
$\min c^T x$	$\max b^T y$
$Ax \geq b$	$A^T y \leq c$
$x \geq 0$	$y \geq 0$

Notice that for each constraint in the original problem, there is a corresponding variable in the dual problem. Similarly, for each variable in the original problem there is a corresponding constraint in the dual problem.

Question. Write the dual linear programs for the assignment problem and the shortest path problem.

Notice that every solution for the dual linear program gives a lower bound on the value of the primal problem.

Theorem 2.1 (Weak duality). *Consider an arbitrary feasible solution y for the dual LP and an arbitrary feasible solution x for the primal LP. we have*

$$x^T c \geq y^T b$$

Proof. Because x is a feasible solution for the primal LP:

$$Ax \geq b$$

Because $y \geq 0$, we can take dot product of both sides with y and obtain $y^T Ax \geq y^T b$.

Similarly, because y is a feasible solution for the dual LP:

$$A^T y \leq c$$

Because $x \geq 0$, we can take dot product of both sides with x and obtain $x^T A^T y \leq x^T c$

Combining the two inequalities, we obtain $x^T c \geq y^T b$ □

Theorem 2.2 (Strong duality). *If the primal and dual problems are feasible then their optimal values are equal.*

To prove the theorem, we will make use of a useful theorem:

Lemma 2.3 (Separating hyperplane theorem). *Let P be a closed convex set and x be a point not in P. There exists a vector w such that $w^T x > \max_{z \in P} w^T z$.*

The theorem is intuitive but proving it requires some formal math so we will skip it. We now proceed to prove the duality theorem.

Lemma 2.4. *Let x^* be the optimal solution for the primal LP. Let S be the set of constraints j that are tight i.e. $(Ax^*)_j = b_j$. There exist $\{\lambda_j \geq 0\}_{j \in S}$ such that $c_i = \sum_{j \in S} \lambda_j A_{ji}$ for all i.*

Proof. Suppose for contradiction that no such $\{\lambda_j\}$ exist. Let A_j denote row j of the constraint matrix A. Let

$$P = \left\{ v \,\middle|\, v = \sum_{j \in S} \lambda_j A_j \text{ for some } \{\lambda_j \geq 0\}_{j \in S} \right\}$$

i.e. P is the set of all linear combinations with nonnegative coefficients of the rows of A in S.

Observe that P is closed and convex (why?) and by our assumption, $c \notin S$ so there exists some w such that $w^T c > \max_{v \in P} w^T v$. Note that this means $w^T c > 0$ and $w^T A_j \leq 0 \ \forall j \in S$ (why?).

Consider the vector $x - \varepsilon w$ for a tiny positive constant ε. We will show that this is a feasible solution with better objective value than x, which is a contradiction:

- For constraint $j \notin S$, because $A_j^T x > b_j$ and ε is sufficiently small, $A_j^T (x - \varepsilon w) > b_j$. For constraint $j \in S$, we have $A_j^T (x - \varepsilon w) = b_j - \varepsilon A_j^T w b_j \geq b_j$ because $A_j^T w \leq 0$.
- The objective value decreases since $c^T (x - \varepsilon w) = c^T x - \varepsilon c^T w < c^T x$.

□

Thus, the objective coefficients is a conic combination of the coefficients in the constraints in S. Consider a set of values for $\lambda_j \geq 0$ so that $c = \sum_j \lambda_j A_j$ and set $\lambda_j = 0 \ \forall j \notin S$.

Observe that

- $\lambda \geq 0$
- $A^T\lambda = \sum_j \lambda_j A_j = c$
- $b^T\lambda = \sum_{j\in S} b_j\lambda_j = \sum_{j\in S}(x^T A_j)\lambda_j = x^T c$

Thus, λ is a solution to the dual problem with dual objective value exactly equal to the optimal primal objective value.

3 Special cases of the duality theorem

There are many interesting special cases of the duality theorem for linear programming. We will mention an example, which many of you might have seen in an undergraduate course.

Consider the maximum flow problem. We are given a directed graph $G = (V, E)$ with source s and sink t. Each edge e has a capacity c_e. The flow on each edge must be at most its capacity and at any vertex other than s, t, the flow must be conserved: the total incoming flow must be equal to the total outgoing flow. We would like to maximize the total flow we can send from s to t.

Let's formulate this problem as a linear program. Let P be the set of directed simple paths from s to t. Let x_p be the variable measuring the amount of flow we are sending on the path p. We have

$$\begin{aligned} \max \sum_{p\in P} x_p &: \\ \sum_{p:e\in p} x_p \leq c_e &\ \forall e \in E \\ x_p \geq 0 &\ \forall p \in P \end{aligned}$$

Let's write the dual of this linear program.

$$\begin{aligned} \min \sum_{e\in E} c_e y_e &: \\ \sum_{e\in p} y_e \geq 1 &\ \forall p \in P \\ y_e \geq 0 &\ \forall e \in E \end{aligned}$$

Notice that this dual represents a fractional version of the minimum cut problem: each edge is picked up to a fraction y_e with the constraint that on any path from s to t, the total fraction of edges being picked is at least 1. The usual minimum cut problem restricts the variables y_e to be either 0 or 1. It turns out that this LP also has an optimal integral solution so its value is equal to the value of a cut in the graph.

Thus, LP duality implies that the maximum flow is equal to the capacity of the minimum cut.

4 Approximation algorithms with linear programs

We have seen several examples where the LP gives exact solution for discrete optimization problems. In general, there might be no integral optimal solution for an LP relaxation and we cannot obtain the optimal solution for the discrete problem. Nonetheless, we can use the fraction solution to construct an *approximation* solution that is close in quality compared with the optimal solution. We will explore a few examples of this approach.

4.1 Vertex cover

Consider a graph $G = (V, E)$ where each node u has a weight $w_u \geq 0$. The goal is to find a vertex cover, which is a subset of vertices that is adjacent to all edges in the graph. Furthermore, we would like to find the vertex cover of minimum total weight.

We can write this problem as an integer linear program as follows.

$$\begin{aligned} \min \sum_{u \in V} w_u x_u : \\ x_u + x_v \geq 1 \ \forall (u, v) \in E \\ x_u \in \{0, 1\} \end{aligned}$$

To obtain an LP, we can relax the integral constraints:

$$\begin{aligned} \min \sum_{u \in V} w_u x_u : \\ x_u + x_v \geq 1 \ \forall (u, v) \in E \\ 0 \leq x_u \leq 1 \end{aligned}$$

From the fractional solution, we can simply round all x_u up to 1 if $x_u \geq 1/2$ and down to 0 if $x_u < 1/2$. Let S be the set of vertices selected by the algorithm. We will show that this is a valid solution whose weight is at most twice that of the optimal solution.

Lemma 4.1. *S is a valid vertex cover.*

Proof. For each edge (u, v), we know $x_u + x_v \geq 1$ so either x_u or x_v is at least $1/2$. Therefore, either u or v is selected in S and the edge (u, v) is covered. □

Lemma 4.2. *The weight of S is at most twice that of the optimal solution.*

Proof. Because we only pick vertices with $x_u \geq 1/2$, the weight of S is at most 2 times $\sum_u x_u w_u$. Because the LP is a relaxation, the optimal integral solution is a valid solution for the LP. Thus, the LP value is at most the weight of the optimal integral solution. Thus, the weight of S is at most 2 times the weight of the optimal integral solution. □

4.2 MAX2SAT

A 2CNF formula consists of n Boolean variables $x_1, \ldots, x_n$ and m clauses of the form $y \vee z$, where each y, z is called a *literal*, which is either a variable or its negation. Given a formula, our goal is to set the variables so as to maximize the number of satisfied clauses.

We start with an integral formulation. We use variable z_j to indicate whether clause j is satisfied or not.

$$\max \sum_{j=1}^{m} z_j$$
$$y_{j1} + y_{j2} \geq z_j \ \forall j$$
$$z_j \leq 1 \ \forall j x_i \in \{0, 1\}$$

where y_{j1} is the shorthand for x_i if the first literal in clause j is variable i and for $1 - x_i$ if the literal is the negation of variable i.

We relax the integral formulation to obtain an LP by replacing $x_i \in \{0, 1\}$ with $0 \leq x_i \leq 1$.

Now to obtain an integral solution, independently for each variable i, we randomly set it to 1 with probability x_i and to 0 with probability $1 - x_i$.

Lemma 4.3. *The expected number of satisfied clause is at least $\frac{3}{4}$ times the optimal value.*

Proof. We will prove that the probability a given clause is satisfied is at least $3z_j/4$. The lemma then follows from linearity of expectation.

Suppose the clause is $x_a \vee x_b$. Notice that at the optimal solution, $z_j = \min(1, x_a + x_b)$ since the LP tries to maximize $\sum_j z_j$.

The probability that randomized rounding satisfies this clause is exactly

$$1 - (1 - x_a)(1 - x_b) = x_a + x_b - x_a x_b$$

Consider two cases. First, consider the case $x_a + x_b \leq 1$. We have $x_a x_b \leq (x_a + x_b)^2/4 \leq (x_a + x_b)/4$. Thus,

$$x_a + x_b - x_a x_b \geq \frac{3}{4}(x_a + x_b) \geq 3x_j/4$$

Next, consider the case $t = x_a + x_b \geq 1$. We have $x_a x_b \leq (x_a + x_b)^2/4 = t^2/4$. Thus,

$$x_a + x_b - x_a x_b \geq t - t^2/4 \geq 3/4 \ \forall t \in [1, 2]$$

Thus, in both cases, the probability that randomized rounding satisfies a clause is at least $3z_j/4$.

4.3 Circuit routing?

□