LP Basis and LP Rounding-based Approximation

Qiyu Liu

Hong Kong University of Science and Technology

qliuau@cse.ust.hk

May 27, 2018

Overview

1. Linear Programming Basis
   - Standard Form
   - Dual Form
   - Duality Theory
2. LP Rouding-based Approximation
   - Weighted Vertex Cover
   - Set Cover

Linear Programming

LP Standard Form

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} c_j x_j \\
s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad & i = 1, 2, \cdots, m \\
& x_j \geq 0 \quad & j = 1, 2, \cdots, n
\end{aligned} \tag{1}
$$

An equivalent form represented by maxtix is:

$$
\begin{aligned}
\max \quad & \mathbf{c}^T \mathbf{x} \\
& \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0}
\end{aligned} \tag{2}
$$

Transformation to Standard Form

- For objective function, change min to max;
- For variables without non-negative constrant, change x_j to $x_j' - x_j''$ and add constraints $x_j' \geq 0$ and $x_j'' \geq 0$;
- For equality constraints, change $\sum_{j=1}^{n} a_{ij}x_j = b_i$ to $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ and $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$;
- For inequality constraints with a $\geq$, change $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ to $-\sum_{j=1}^{n} a_{ij}x_j \leq -b_i$.

Linear Programming

LP Dual Form

For the LP standard form shown in Eq. (1), the corresponding dual form is given by:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} b_i y_i & & \\
s.t. \quad & \sum_{i=1}^{m} a_{ij} y_i \geq c_j & & j = 1, 2, \cdots, n \\
& y_i \geq 0 & & i = 1, 2, \cdots, m
\end{aligned}
\tag{3}
$$

where y_i are dual variables.

The LP dual form can be derived by introducing *Lagrangians* for the constraints and eliminate all the variables except dual variables by setting derivetives to 0.

LP Dual Form Derivation

According to the standard LP shown in Eq. (1), we can write down the Lagrangian as follows:

$$\mathcal{L}(\{x_j\},\{y_i\},\{\lambda_j\}) = \sum_{j=1}^{n} c_j x_j - \sum_{i=1}^{m} y_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right) - \sum_{j=1}^{n} \lambda_j x_j \quad (4)$$

Then, Eq. (1) is equivalent to:

$$\begin{aligned} \min \quad & \mathcal{L}(\{x_j\},\{y_i\},\{\lambda_j\}) \\ s.t. \quad & \forall y_i \geq 0 \\ & \forall \lambda_j \geq 0 \end{aligned} \quad (5)$$

Then, we try to eliminate the original variables (primal variables) x_j. The basic idea is setting the derivative of $\mathcal{L}$ w.r.t. x_j to 0.

Duality Theory

Theorem (Waek Duality Theory)

Let $\bar{x}$ and $\bar{y}$ denote any feasible solution to the standard LP and its corresponding dual LP, it always holds that:

$$\sum_{j=1}^{n} c_j \bar{x}_j \leq \sum_{i=1}^{n} b_i \bar{y}_i. \tag{6}$$

Theorem (Strong Duality Theory)

Let x^* and y^* denote the optimal solution to the standard LP and its corresponding dual LP, it always holds that:

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{n} b_i y_i^*. \tag{7}$$

Algorithms for LP

Existing Algorithms:

- Simplex alogirithm [Dantzig 1947];
- Ellipsoid algorithm [Khachian 1979].

Open Problem: Is there a strongly polynomial-time algorithm for LP?

Integral Linear Programming

Integral Linear Programming (ILP)

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} c_j x_j \\
s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad & i = 1, 2, \cdots, m \\
& x_j \ge 0 & j = 1, 2, \cdots, n \\
& x_j \text{ is integral} & j = 1, 2, \cdots, n
\end{aligned} \tag{8}
$$

- 0-1 ILP is a special case of ILP where x_i takes value from 0 and 1;
- Solving an ILP is NP-hard (observe that many NPC problems, e.g., vertex cover, are special cases of ILP).

Integral Linear Programming – Cont.

Exact Solutions to ILP:

- Branch and Bound;
- Cutting plane [Gomory 1958].

There is no polynomial exact algorithm for ILP, which is the reason we use LP relaxation and LP rounding instead of directly solving ILP.

LP – Weighted Vertex Cover

Weighted Vertex Cover

Given an undirected graph $G = (V, E)$ with vertex weights $w_i > 0$, find a minimum weight subset of nodes S such that every edge is incident to at least one vertex in S.

Define 0/1 variable x_i as follows:

$$x_i = \begin{cases} 0, & \text{if vertex } i \text{ is not in vertex cover} \\ 1, & \text{if vertex } i \text{ is in vertex cover} \end{cases} \tag{9}$$

ILP formulation of weighted vertex cover problem:

$$\begin{aligned} \min \quad & \sum_{i \in V} w_i x_i \\ s.t. \quad & x_i + x_j \geq 1 \quad (i, j) \in E \\ & x_i \in (0, 1) \qquad i \in V \end{aligned} \tag{10}$$

LP Relaxation and LP Rounding

The basic framework of utilizing ILP to get an polynomial approximation algorithm for some NP-hard problem is to use LP relaxation and LP rounding (instead of directly solving ILP since no polynomial algorithm for general ILP).

Relaxation is the process that relaxes the integral constraints which transforms an ILP to LP.

Rounding is the process that using results of LP to obtain *appximated* solution to the corresponding ILP (introducing some error but sometimes can be bounded).

LP Relaxation and Rounding for Vertex Cover

The LP relaxation of Eq. (10) is:

$$\begin{aligned} \min \quad & \sum_{i \in V} w_i x_i \\ s.t. \quad & x_i + x_j \geq 1 \quad (i,j) \in E \\ & x_i \geq 0 \qquad\qquad i \in V \end{aligned} \tag{11}$$

LP rounding based algorithm for weighted vertex cover problem:

1. Solve LP relaxation of weighted vertex cover in Eq. (11). Denote the optimal solution as $\{x_i^*\}$;
2. Return a vertex set $S = \{i | x_i^* \geq \frac{1}{2}\}$.

Step 2 introduces error and leads to an approximation!

LP Relaxation and Rounding for Vertex Cover – Cont.

Lemma (S is a vertex cover.)

Proof. For any edge $(i, j) \in E$, since $x_i + x_j \geq 1$, either $x_i^* \geq \frac{1}{2}$ or $x_j^* \geq \frac{1}{2}$. Thus, edge must be covered by S.

Lemma ($w(S) \leq 2OPT$.)

Proof. Suppose S^* is the optimal vertex cover. Then we have,

$$OPT = \sum_{i \in S^*} w_i \geq \sum_{i \in V} w_i x_i^* \geq \sum_{i \in S} w_i x_i^* \geq \frac{1}{2} \sum_{i \in S} w_i. \tag{12}$$

The above proof utilizes an observation that the optimal value of LP is no larger than that of its corresponding ILP since LP has fewer constraints.

Integrality Gap

Linear programming relaxation is a standard technique for designing approximation algorithms for hard optimization problems. In this application, an important concept is the **Integrality Gap**, the maximum ratio between the solution quality of the integer program and of its relaxation.

Integrality Gap

Let $\mathcal{I}$ be any instance of a given minimization problem formulated by ILP, the a is given by:

$$IG = \sup_{\mathcal{I}} \frac{OPT_{ILP}(\mathcal{I})}{OPT_{LP}(\mathcal{I})}.$$

Note that, the integrality gap of weighted vertex cover problem is 2 (consider a complete graph).

Integrality Gap – Cont.

Typically, the integrality gap translates into the approximation ratio of an approximation algorithm. This is because an approximation algorithm relies on some rounding strategy that finds, for every relaxed solution of size OPT_{LP}, an integer solution of size at most $RR \cdot OPT_{LP}$ (where RR is the rounding ratio). If there is an instance with integrality gap IG, then every rounding strategy will return, on that instance, a rounded solution of size at least $OPT_{ILP} = IG \cdot OPT_{LP}$. Therefore necessarily $RR \geq IG$.

The rounding ratio RR is only an upper bound on the approximation ratio, so in theory the actual approximation ratio may be lower than IG, but this may be hard to prove. In practice, a large IG usually implies that the approximation ratio in the linear programming relaxation might be bad, and it may be better to look for other approximation schemes for that problem.

LP – Set Cover

Set Cover

Given a universe U of n elements, a list $S_1, S_2, \cdots, S_m$ of subsets of U with weights $w_1, w_2, \cdots, w_m$, find a collection of these subsets whose union is exactily U such that the total weight is minimized.

Define 0/1 variable x_i where $i = 1, 2, \cdots, m$ as:

$$x_i = \begin{cases} 1, & \text{if } S_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Then, the ILP formulation for set cover problem is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} w_i x_i \\ s.t. \quad & \sum_{j \in S_i} x_i \geq 1 \qquad\qquad j \in U \\ & x_i \in \{0, 1\} \quad i = 1, \cdots, m \end{aligned} \tag{14}$$

LP Relaxation and LP Rounding for Set Cover

The relaxation of ILP of set cover problem shown in Eq. (14) is:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} w_i x_i \\
s.t. \quad & \sum_{j \in S_i} x_i \geq 1 \qquad\qquad j \in U \\
& x_i \geq 0 \qquad\qquad i = 1, \cdots, m
\end{aligned}
\tag{15}
$$

The LP relaxation and rounding based algorithm for set cover is shown as follows:

1. Solve LP relaxation of set cover in Eq. (15). Denote the optimal solution as $\{x_i^*\}$.
2. Return a set cover $\mathcal{S} = \{S_i | x_i^* \geq \frac{1}{f}\}$ where f is the frequency of the most frequent element.

LP Relaxation and LP Rounding for Set Cover – Cont.

Lemma ($\mathcal{S}$ is a set cover to U.)

Proof. *Suppose item $j' \in U$ has the most frequency f, then, to satisfy the constraint $\sum_{j' \in S_i} x_i \geq 1$, there must be at least one $x_i^* \geq \frac{1}{f}$. Thus, item j' is covered by $\mathcal{S}$. Then, it is easy to show that if j' can be covered by $\mathcal{S}$, any $j \in U$ can be covered by $\mathcal{S}$.*

Lemma ($w(\mathcal{S}) \leq f \cdot OPT$.)

Proof. *Suppose $\mathcal{S}^*$ is the optimal set cover. Then, we have,*

$$OPT = \sum_{i \in \mathcal{S}^*} w_i \geq \sum_{i=1}^{m} w_i x_i^* \geq \sum_{i \in \mathcal{S}} w_i x_i^* \geq \frac{1}{f} \sum_{i \in \mathcal{S}} w_i. \tag{16}$$

LP Relaxation and LP Rounding for Set Cover – Cont.

My summary:

1. first step is to analyze the difference of solution structure between ILP and its LP realaxation;
2. then design proper rounding scheme to get feasible solution of the original ILP problem;
3. approxiamtion ratio highly depends on how you do rounding.

Randomized Rounding Technique

It is easy to know that $0 \leq x_i^* \leq 1$, which means we can regard x_i^* as a probability. Higher value of x_i^* means more possibility that S_i is selected, which gives the intuition of **randomized rounding**.

The LP relaxation and randomized rounding based algorithm for set cover is shown as follows:

1. Solve LP relaxation of set cover in Eq. (15). Denote the optimal solution as $\{x_i^*\}$.
2. Pick S_i with probability x_i^*.

Randomized Rounding Analysis

- The expected cost of randomized selection is:

$$\sum_{i=1}^{m} \Pr(S_i \text{ is picked}) \cdot w_i = \sum_{i=1}^{m} w_i x_i^*.$$

- For any item $j \in U$, the probability that j is not covered is:

$$\prod_{j \in S_i} (1 - x_i^*) \leq (1 - \frac{1}{k})^k \leq \frac{1}{e}$$

  where k is the occuring frequency of j.

- To make sure every item is covered, do selection $d \log n$ times independently, and take all sets as final result. Then, by taking large enough d, the probability that j is not covered is:

$$\left(\frac{1}{e}\right)^{d \log n} \leq \frac{1}{4n}.$$

Randomized Rounding Analysis – Cont.

- **Bad Event 1:** Some element is not covered.

$$Pr(\text{Bad Event 1}) \leq \frac{1}{4n} n = \frac{1}{4}$$

- **Bad Event 2:** Total cost is too high.

$$E[\text{cost of one round}] \leq \sum_{i=1}^{m} w_i x_i^*$$

$$E[\text{cost of } d \log n \text{ rounds}] \leq d \log n \sum_{i=1}^{m} w_i x_i^* \leq d \log n \cdot OPT$$

$$\Pr(\text{cost of } d \log n \text{ rounds} \geq 4d \log n \cdot OPT) \leq \frac{1}{4}$$

Thus, the probability that the algorithm succeeds is at least $1 - \frac{1}{4} - \frac{1}{4} = \frac{1}{2}$.

The End