Notes on Approximation Algorithms

Bhargav Samineni

Abstract

A collection of some notes on the design and analysis of approximation algorithms and approximation techniques. Based mainly off of [1, 2].

1 Introduction to LP-Duality

For a minimization linear program in canonical form, the goal is to find a non-negative, rational vector x that minimizes a given linear objective function in x subject to some linear constraints on x. If there are n decision variables and m linear constraints, then $x \in \mathbb{Q}^n$, the coefficients of the linear objective function can be represented by a vector $c \in \mathbb{Q}^n$, the coefficients of the linear constraints can be represented by a matrix $A = (a_{ij}) \in \mathbb{Q}^{m \times n}$, and the values of the linear constraints can be represented by a vector $b \in \mathbb{Q}^m$.

Definition 1 (Primal and Dual)

Given a linear programming problem in canonical form denoted as (P), we can induce a problem denoted by (D) with the following form:

$$
\begin{array}{llll}
(P) & \text{minimize} & \displaystyle\sum_{j=1}^{n} c_j x_j & \\
& \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j \geq b_i & i = 1, \dots, m \\
& & x_j \geq 0 & j = 1, \dots, n
\end{array}
\qquad
\begin{array}{llll}
(D) & \text{maximize} & \displaystyle\sum_{i=1}^{m} b_i y_i & \\
& \text{subject to} & \displaystyle\sum_{i=1}^{m} a_{ij} y_i \leq c_j & j = 1, \dots, n \\
& & y_i \geq 0 & i = 1, \dots, m
\end{array}
$$

where a_{ij}, b_i, and c_i are given rational numbers and y_i corresponds to the ith inequality of (P).

Definition 2 (Primal)

The problem (P) is referred to as the *primal*.

Definition 3 (Dual)

The *dual* of the primal is problem (D).

Every feasible solution for the dual serves as a lower bound on the optimal objective function value of the primal. The reverse also holds in that every feasible solution to the primal serves as an upper bound on the optimal objective function value of the dual.

Theorem 1 (Weak Duality)**.** *If $x = (x_1, \dots, x_n)$ is a feasible solution to the LP (P) and $y = (y_1, \dots, y_m)$ a feasible solution to the LP (D), then $\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i$.*

Proof.

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \geq \sum_{i=1}^{m} b_i y_i$$

■

As a consequence of Theorem 1, if we find that there exist some feasible solutions to (P) and (D) that have matching objection function values, then these solutions must be optimal.

Theorem 2 (Strong Duality/LP-Duality)**.** *If the LPs (P) and (D) are both feasible, and $x^* = (x_1^*, \ldots, x_n^*)$ and $y^* = (y_1^*, \ldots, y_m^*)$ are optimal solutions to (P) and (D), respectively, then $\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*$.*

As a corollary of Theorem 2, we find that optimal solutions to (P) and (D) must satisfy a set if conditions.

Corollary 1 (Complementary Slack Conditions)**.** *Let x and y be feasible solutions to (P) and (D), respectively. Then x and y are both optimal iff the following are satisfied:*

1. *For each $1 \leq j \leq n$, either $x_j = 0$ or $\sum_{i=1}^{m} a_{ij} y_i = c_j$*
2. *For each $1 \leq i \leq m$, either $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$.*

1.1 Algorithm Design Techniques

Definition 4 (Extreme Point Solution)

Consider the polyhedron defining the set of feasible solutions to an LP. A feasible solution is called an *extreme point solution* if it is a vertex of the polyhedron.

Theorem 3. *For any objective function, there is an extreme point solution that is optimal.*

Combinatorial optimization problems can often be stated as integer programs (IP), which can be relaxed to an LP. For the case of NP-Hard problems, however, the polyhedron that defines the set of feasible solutions to the LP-relaxation may not have integer vertices. With the LP-relaxation, the goal is to find near optimal integral solutions.

One method to get an approximation algorithm is through *LP-rounding*. Rounding involves solving the LP-relaxation and the converting the fractional solution to an integral solution while ensuring the in the process the cost does not increase dramatically. The approximation ratio is given by comparing the cost of the integral and fractional solutions.

Another method is the *primal-dual schema*. Consider an LP-relaxation as the primal program. Under the scheme, an integral solution to the primal and a feasible solution to the dual are constructed iteratively. Any feasible solution to the dual also provides a lower bound on OPT. The approximation ratio is given by comparing the two solutions.

1.1.1 Integrality Gap

Definition 5 (Integrality Gap)

Given an LP-relaxation for a minimization problem, let $\text{OPT}_f(I)$ be the cost of an optimal fractional solution to an instance I of the problem. Similarly, let $\text{OPT}(I)$ denote the cost of an optimal solution to the original IP for an instance I of the problem. The *integrality gap* of the relaxation is

$$\sup_I \frac{\mathrm{OPT}(I)}{\mathrm{OPT}_f(I)}.$$

In the case of a maximization problem, the integrality gap is given by the infimum of the ratio.

When the integrality gap of an LP-relaxation is exactly 1, such a relaxation is called an *exact relaxation*. If the cost of a solution found by an algorithm is compared to the cost of an optimal fractional solution (or feasible dual solution), the best approximation factor one can hope to achieve is the integrality gap of the relaxation.

2 Set Cover

Problem 1 (Set Cover)

Given a ground set $U = \{e_1, \ldots, e_n\}$, a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of U, and a cost function $c\colon \mathcal{S} \to \mathbb{Q}^+$, find a minimum cost subcollection of $\mathcal{S}$ that covers all elements in U.

To formulate Set Cover as an IP, denote the binary decision variables by x_j for each $S_j \in \mathcal{S}$. If S_j is in the cover, then $x_j = 1$.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} c(S_j) x_j \\ \text{subject to} \quad & \sum_{j\colon e_i \in S_j} x_j \geq 1 \quad && i = 1, \ldots, n \\ & x_j \in \{0, 1\} && j = 1, \ldots, m \end{aligned} \tag{1}$$

The LP-relaxation to this IP is given by letting $x_j \geq 0$. A solution to the LP-relaxation can be viewed as a fractional set cover.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} c(S_j) x_j \\ \text{subject to} \quad & \sum_{j\colon e_i \in S_j} x_j \geq 1 \quad && i = 1, \ldots, n \\ & x_j \geq 0 && j = 1, \ldots, m \end{aligned} \tag{2}$$

Introducing a variable y_i for each $e_i \in U$, we get the dual program.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} y_i \\ \text{subject to} \quad & \sum_{i\colon e_i \in S_j} y_i \leq c(S_j) \quad && j = 1, \ldots, m \\ & y_i \geq 0 && i = 1, \ldots, n \end{aligned} \tag{3}$$

2.1 Dual-Fitting Analysis

Algorithm 1 defines the dual program's value for each variable as $y_i = p(e_i)$. However, the solution y constructed by this approach is infeasible as it may not satisfy the first constraint. To see this, note that if a set S_j is chosen by the algorithm, it distributes its cost across all its uncovered elements. If $S_j \cap C \neq \emptyset$, then this means that it contains some covered elements that already have some other set's cost distributed to them, which would cause $\sum_{i\colon e_i \in S_j} y_i \geq c(S_j)$. Instead, we may be able to show that some scaled solution $y' = \frac{y}{\kappa}$ where $\kappa > 1$ is feasible for the dual.

Algorithm 1 Greedy Set Cover Algorithm

```
1: C, I ← ∅
2: while C ≠ U do
3:     i = argmin_{j: S_j ∈ 𝒮} c(S_j)/|S_j \ C|
4:     p(e_i) = c(S_i)/|S_i \ C|        ∀e_i ∈ S_i \ C
5:     C ← C ∪ S_i, I ← I ∪ i, 𝒮 ← 𝒮 \ S_i
6: return C
```

Lemma 1. *The vector $y' = \frac{y}{H_g}$ is a feasible solution for the dual program Eq. (3), where $g = \max_i |S_i|$ and H_g is the gth harmonic number.*

Proof. Assume the algorithm runs for ℓ iterations. Let $a_{j,k}$ be the number of elements in S_j that are still uncovered at the start of the kth iteration (hence $a_{1,j} = |S_j|), a_{\ell+1,j} = 0)$. Let $A_{j,k}$ denote the set of uncovered elements in S_j that are covered at the end of the kth iteration. If S_i is chosen in the kth iteration, for each covered element $e_i \in A_{i,k}$,

$$y'_i = \frac{c(S_i)}{H_g a_{i,k}} \leq \frac{c(S_j)}{H_g a_{j,k}} \qquad \forall S_j \in \mathcal{S}$$

since S_i minimizes the ratio $\frac{S_i}{a_{i,k}}$. Then for any set S_j,

$$\begin{aligned}
\sum_{i:\, e_i \in S_j} y'_i = \sum_{k=1}^{\ell} \sum_{i:\, e_i \in A_{j,k}} y'_i &\leq \sum_{k=1}^{\ell} (a_{j,k} - a_{j,k+1}) \frac{c(S_j)}{H_g a_{j,k}} \\
&= \frac{c(S_j)}{H_g} \sum_{k=1}^{\ell} \frac{a_{j,k} - a_{j,k+1}}{a_{j,k}} \\
&= \frac{c(S_j)}{H_g} \sum_{k=1}^{\ell} \sum_{l=a_{j,k+1}+1}^{a_{j,k}} \frac{1}{a_{j,k}} \\
&\leq \frac{c(S_j)}{H_g} \sum_{k=1}^{\ell} \left(\frac{1}{a_{j,k}} + \frac{1}{a_{j,k}-1} + \ldots + \frac{1}{a_{j,k+1}+1} \right) \\
&\leq \frac{c(S_j)}{H_g} \sum_{l=1}^{|S_j|} \frac{1}{l} = \frac{c(S_j)}{H_g} H_{|S_j|} \leq c(S_j)
\end{aligned}$$

which satisfies the first constraint of Eq. (3). ■

Theorem 4. *Algorithm 1 gives a H_g approximation to Set Cover.*

Proof. Since the greedy algorithm distributes the cost of the sets it chooses across all the elements, $\sum_{j \in I} c(S_j) = \sum_{i=1}^{n} y_i$. By Lemma 1, we know that y' is a feasible solution to the dual, so using Theorem 1 we get that

$$\sum_{j \in I} c(S_j) = \sum_{i=1}^{n} y_i = H_g \sum_{i=1}^{n} y'_i \leq H_g \mathrm{OPT}_f \leq H_g \mathrm{OPT}.$$

■

3 Exercises

3.1 LP-Duality

Exercise 3.1.1
The dual of the dual is the primal.

Solution.
The dual (*D*) is equivalent to the minimization problem

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{m} -b_i y_i & \\ \text{subject to} & \displaystyle\sum_{i=1}^{m} -a_{ij} y_i \geq -c_j & j = 1, \dots, n \\ & y_i \geq 0 & i = 1, \dots, m. \end{array}$$

Taking the dual of this LP gives

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{j=1}^{n} -c_j x_j & \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} -a_{ij} x_j \leq -b_i & i = 1, \dots, m \\ & x_j \geq 0 & j = 1, \dots, n \end{array} \iff \begin{array}{lll} \text{minimize} & \displaystyle\sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j \geq b_i & i = 1, \dots, m \\ & x_j \geq 0 & j = 1, \dots, n \end{array}$$

which is the primal (*P*). ■

References

[1] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[2] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.