ADVANCED ALGORITHMS (II)

CHIHAO ZHANG

1. MaxSAT

We already mentioned in class that we can obtain a $\frac{3}{4}$-approximation algorithm by *choosing the best of two*, i.e., returning the better results of the LP rounding based algorithm and the "tossing fair coins" algorithm. We now show that a $\frac{3}{4}$-approximation algorithm can be obtained directly by rounding.

Recall that our linear relaxation of MaxSAT is

$$\begin{aligned} \max \quad & \sum_{j=1}^{m} z_j \\ \text{subject to} \quad & \sum_{i \in P_j} y_i + \sum_{k \in N_j} (1 - y_k) \geq z_j, \quad \forall C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{k \in N_j} \bar{x}_k \\ & 0 \leq z_j \leq 1, \quad \forall j \in [m] \\ & 0 \leq y_i \leq 1, \quad \forall i \in [n] \end{aligned}$$

Let $\left\{y_i^*\right\}_{i \in [n]}, \left\{z_j^*\right\}_{j \in [m]}$ be an optimal solution of the LP. Instead of tossing y_i^*-biased coins for each variable, we set up an increasing function $f : [0,1] \to [0,1]$ and toss $f(y_i^*)$-biased coins. Following this idea, the probability that $C_j = \bigvee_{i \in P_j} x_j \wedge \bigvee_{k \in N_j} \bar{x}_k$ is not satisfied is

$$\prod_{i \in P_j} (1 - f(y_i^*)) \prod_{k \in N_j} f(y_k^*). \tag{1}$$

Remember that we want to relate (1) with constraints $z_j^* \leq \sum_{i \in P_j} y_i^* + \sum_{k \in N_j} (1 - y_k^*)$, in which each y_i^* is linear. Therefore, exponential function might be a good choice for $f(\cdot)$.

We now assume that for some $\beta > 1$, it holds that $1 - \beta^{-y} \leq f(y) \leq \beta^{y-1}$ for every $y \in [0,1]$. Then

$$(1) \leq \beta^{-\sum_{i \in P_j} y_i^* + \sum_{k \in N_j} (y_k^* - 1)} \leq \beta^{-z_j^*}.$$

Therefore, we have

$$\mathbf{Pr}\left[C_j \text{ is satisfied}\right] \geq 1 - \beta^{-z_j^*} \overset{①}{\geq} (1 - \beta^{-1}) z_j^*,$$

where ① is due to the fact that the function $h(z) = 1 - \beta^{-z}$ is concave on $[0,1]$. We now have

$$\mathbf{E}\left[X\right] = \sum_{j \in [m]} \mathbf{Pr}\left[C_j \text{ is satisfied}\right] \geq (1 - \beta^{-1}) \sum_{j \in [m]} z_j^* = (1 - \beta^{-1}) \cdot \mathbf{OPT}(LP) \geq (1 - \beta^{-1}) \cdot \mathbf{OPT}.$$

The approximation ratio is increasing in β, therefore, we want to find a maximum β so that the function $f(y)$ exists, namely $1 - \beta^{-y} \leq \beta^{y-1}$ for every $y \in [0,1]$. An easy calculation yields $\beta \leq 4$. We plot the function $1 - 4^{-y}$ and 4^{y-1} in Figure 1.

2. Integrality Gap

Can we find a more clever way to round the LP and beat the $\frac{3}{4}$ bound? The answer is no if we still use the upper bound $\mathbf{OPT} \leq \mathbf{OPT}(LP)$ in the analysis.

Consider the following CNF formula:

$$\phi = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2). \tag{2}$$

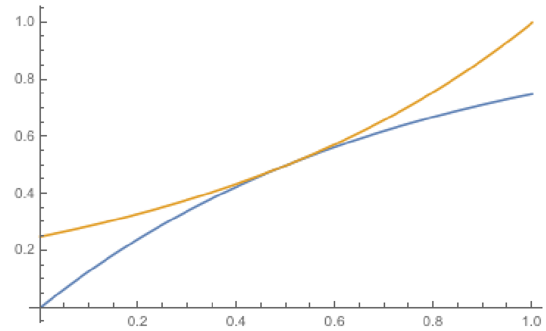

FIGURE 1. The function $1 - 4^{-y}$ and 4^{y-1}.

It is clear that $\mathbf{OPT} = 3$. On the otherhand, $\mathbf{OPT}(LP) = 4$ since we can set $y_1 = y_2 = \frac{1}{2}$. Therefore, let Z be the cost of *any* solution of this instance, it must be the case that

$$\frac{Z}{\mathbf{OPT}(LP)} \le \frac{\mathbf{OPT}}{\mathbf{OPT}(LP)} = \frac{3}{4},$$

or equivalently

$$Z \le \frac{3}{4} \cdot \mathbf{OPT}(LP). \tag{3}$$

Now consider an α-approximation algorithm $\mathbb{A}$ for MaxSAT, and assume that we establish the α-approximation by using the upper bound $\mathbf{OPT} \le \mathbf{OPT}(LP)$ (like all LP based examples we met so far), namely we prove for any ψ, it holds that

$$\mathbb{A}(\psi) \ge \alpha \cdot \mathbf{OPT}(LP) \ge \alpha \cdot \mathbf{OPT}.$$

Then we must have $\alpha \le \frac{3}{4}$, otherwise, there is a contradiction with (3).

The $\frac{3}{4}$ here is called the *integrality gap* of our LP relaxation, it is formally defined to be

$$\min_{\text{instance } I} \frac{\mathbf{OPT}(LP)(I)}{\mathbf{OPT}(I)},$$

where $\mathbf{OPT}(I)$ and $\mathbf{OPT}(LP)(I)$ are the optimal cost of the instance I and the cost of LP relaxation of I respectively.

3. Minimum Label Cut

The second example we are going to consider is the problem of *minimum label s-t cut*, which is formally defined as follows:

> MINIMUM LABEL s-t CUT
>
> *Input:* A graph $G = (V, E)$; a set of labels $[L] = \{1, 2, \ldots, L\}$ such that each $e \in E$ is labelled with one $\ell(e) \in [L]$; two vertices $s, t \in V$.
>
> *Problem:* Compute a minimum set of labels $L' \subseteq [L]$ such that the removal of all edges with label in L' disconnects s and t.

We can directly write down a linear programming relaxation of the problem: For each $j \in [L]$, we introduce a variable z_j indicating whether the label j is chosen, then the following LP relaxes the problem,

$$\begin{aligned} \min \quad & \sum_{j \in [L]} z_j \\ \text{subject to} \quad & \sum_{e \in P} z_{\ell(e)} \ge 1, \quad \forall P \in \mathcal{P}_{s,t} \\ & z_j \in [0,1], \quad \forall j \in [L], \end{aligned}$$

where we use $\mathcal{P}_{s,t}$ to denote the collection of simple paths connecting s and t, and we represent each path $P \in \mathcal{P}$ as a set of edges.

We need to notice that the set $\mathcal{P}_{s,t}$ might be very large and hence the number of constraints of the LP might be exponential. This raise the issue that how can we solve the LP in polynomial-time. In fact, we can solve this LP using *ellipsoid method* provided an efficient *separation oracle*.

First, we can reduce the LP to the problem of deciding whether a collection of linear constraints is feasible: we can use binary search to guess an *optimal solution*, say S, and check the feasibility of

$$
\begin{aligned}
S &\leq \sum_{j \in [L]} z_j \\
\sum_{e \in P} z_{\ell(e)} &\geq 1, \quad \forall P \in \mathcal{P}_{s,t} \\
z_j &\in [0,1], \quad \forall j \in [L].
\end{aligned}
$$

An efficient separation oracle is a polynomial-time algorithm that given as input a point $(\hat{S}, \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_L)$, can

- check the feasibility of this point; and if it is not feasible,
- output a constriant that the point violates.

The ellipsoid method can well approximate the optimal of an LP provided an efficient separation oracle. In this lecture, we treat this result as a black-box, therefore, we only need to construct a separation oracle.

The separation oracle for our LP relaxation of minimum label s-t cut is straightforward. Given a point $(\hat{S}, \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_L)$, we first check whether $\hat{S} \leq \sum_{j \in [L]} \hat{z}_j$. If not, the oracle returns this constraint as the violated one. We then assign a weight $z_{\ell(e)}$ for each edge e. After this, we check the length of the shortest path between s and t with respect to these weights. If the length of the shortest path is at least one, the oracle return "feasible", otherwise, return the constraint corresponding to the shortest path as the violated one.

In the last section of this lecture, we introduced the notion of integrality gap, which measures how "relax" the LP is. What is the integrality gap of our LP relaxation for minimum label s-t cut? It is at least $|E|$, as shown by the instance described in Figure 2.

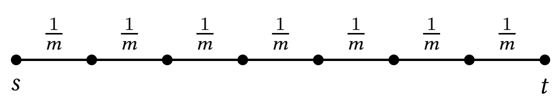

FIGURE 2. A path of length $m = 7$.

Formally, we can let G be a path and let s and t be the two ends respectively. Moreover, we let $L = 1$ so that every edge in G is of the same label. Then it is clear that the optimal cost of the problem is one. On the otherhand, we can let $z_1 = \frac{1}{|E|}$ and the path withnesses that our integrality gap is at least $|E|$.

Therefore, we cannot obtain any non-trivial approximation algorithm by simply rounding the fractional solution of the LP. Instead, we first obtain a *partial cut* via rounding, and then complement it combinatorially. We can therefore circumvent the integrality gap barrier.

Here is the algorithm

Let $\left\{z_j^*\right\}_{j \in [L]}$ be an optimal solution of the LP. Let $\beta > 0$ be a parameter.

(1) Let $L_1 \triangleq \left\{j \in L \,:\, z_j^* \geq \beta\right\}$.
(2) Let G' be the graph obtained from G by removing edges with label in L_1.
(3) Let F be the minimum s-t cut of G', L_2 be the labels of edges in F.
(4) Return $L_1 \cup L_2$.

In step (1), we construct a partial cut by choosing all the labels j with corresponding $z_j^* \geq \beta$ for some parameter β to be set. This rounding step simply follows the information given by the LP relaxation: larger z_j^* implies the label j is more likely to be in the optimal solution. Of course, we cannot guarantee the edges with label in L_1 separate s and t in G, so in step (3) and (4), we find a minimum s-t cut in the remaining graph and denote the set of labels in the cut by L_2. Our final solution is $L_1 \cup L_2$.

The correctness of the algorithm is straightforward. Therefore, we only need to bound $|L_1 \cup L_2|$. The construction of L_1 is rounding all z_j^* with $z_j^* \geq \beta$ to one, therefore we have

$$|L_1| \leq \sum_{j \in [L]} \frac{1}{\beta} \cdot z_j^* = \frac{1}{\beta} \cdot \mathbf{OPT}(LP) \leq \frac{1}{\beta} \cdot \mathbf{OPT}.$$

It remains to bound L_2, and we will turn to show that the graph G' has small s-t cut. If s and t are not connected, then $L_2 = \emptyset$. Otherwise, consider every simple path P connecting s and t in G'. An important observation here is that the path P must be quite long, this is because

- the path P survives after removing edges with label in L_1, so every $e \in P$ must satisfy $z_{\ell(e)}^* < \beta$; and
- the path P satisfies the constraint $\sum_{e \in P} z_{\ell(e)}^* \geq 1$.

The above discussion implies $|P| > \frac{1}{\beta}$. Does this property imply that the graph G' contains small s-t cut? The answer is Yes, and we have a few different ways to argument this. One easy way is to apply Menger's theorem from graph theory, which is a special case of the *max-flow min-cut theorem*.

Theorem 1 (Menger's Theorem)**.** *Let G ba a finite undirected graph and s, t be two distinct vertices. The size of the minimum edge cut for s and t is equal to the maximum number of pairwise edge-disjoint paths from s to t.*

In our case, since every path from s to t contains more than $\frac{1}{\beta}$ edges, the maximum number of pairwise edge-disjoint paths from s to t is less than $\beta\,|E|$. Therefore, by Menger's theorem, we have

$$|L_2| \leq |F| \leq \beta\,|E|\,.$$

Combining bounds for $|L_1|$ and $|L_2|$ together, we obtain

$$|L_1 \cup L_2| \leq |L_1| + |L_2| \leq \frac{1}{\beta} \cdot \mathbf{OPT} + \beta\,|E|\,.$$

So if we choose $\beta = \sqrt{\frac{\mathbf{OPT}}{|E|}}$, we have

$$|L_1 \cup L_2| \leq 2\left(\frac{|E|}{\mathbf{OPT}}\right)^{\frac{1}{2}} \cdot \mathbf{OPT},$$

in other words, we have an $O\left(\left(\frac{|E|}{\mathbf{OPT}}\right)^{\frac{1}{2}}\right)$-approximation algorithm.

There is still one problem remains: how can we implement the algorithm for $\beta = \sqrt{\frac{\mathbf{OPT}}{|E|}}$? We do not know **OPT** in advance, but we can run the algorithm with $\beta = \sqrt{\frac{i}{|E|}}$ for all $i = 1, \ldots, m$ and return the best solution. This can be accomplished in polynomial-time.

4. Remark

The presention on MaxSAT problem follows [WS11, Chapter 5], you are advised to check the book for more details about the problem. The algorithm for minimum s-t cut is from [TZ12]. In its journal version [ZFT18], the linear programming rounding part has been replaced by a purely combinatorial algorithm with the same performance.

References

[TZ12] Linqing Tang and Peng Zhang. Approximating minimum label s-t cut via linear programming. In *Latin American Symposium on Theoretical Informatics*, pages 655–666. Springer, 2012. 4

[WS11] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011. 4

[ZFT18] Peng Zhang, Bin Fu, and Linqing Tang. Simpler and better approximation algorithms for the unweighted minimum label st cut problem. *Algorithmica*, 80(1):398–409, 2018. 4