CSCI-B609: A Theorist's Toolkit, Fall 2016 Dec 5, 2016

Solving LPs/SDPs via Multiplicative Weighs

Lecturer: Yuan Zhou *Scribe: Ruiyu Zhu*

From the previous lecture, we have proved the following lemma.

Lemma 1. *If the cost vector* $\vec{c^t} \in [-\rho, \rho]^n$, *and* $T \geq \frac{4\rho^2 \ln n}{\epsilon^2}$, *then average loss*

$$\frac{1}{T}\sum_{t=1}^{T} \vec{p^t}\vec{c^t} \leq \frac{1}{T}\sum_{t=1}^{T} \vec{c^t}(i) + \epsilon, \forall i \in [n]$$

In this lecture, we are going to see the application of multi-weight algorithm for solving LPs and SDPs. For illustration purpose, we will be working on an LP example (the LP relaxation for set cover) and an SDP example (the SDP relaxation for maximum cut). The algorithms should be able to extended to general LPs and SDPs with a few modifications.

1 Set cover LP

First, we are going to show how to use multi-weight algorithm to solve LPs. We will use set-cover LP as an example.

Let U denotes the universe set, $|U| = n$. And there is a family of sets $\mathcal{F} = \{S_1, S_2, \cdots, S_m\}$. We want to choose as less as possible sets from $\mathcal{F}$ to cover the whole set.

We could interpret the problem as such a IP:

$$\begin{aligned} \min \sum_{S \in \mathcal{F}} \chi_S \\ s.t. \forall e \in U, \sum_{\substack{S \in \mathcal{F} \\ e \in S}} \chi_S \geq 1 \\ \chi_S \in \{0, 1\} \forall S \in \mathcal{F} \end{aligned}$$

where each χ_S indicates whether this set is chosen or not.

The corresponding LP relaxation is:

$$\min \sum_{S \in \mathcal{F}} \chi_S$$
$$s.t. \forall e \in U, \sum_{\substack{S \in \mathcal{F} \\ e \in S}} \chi_S \geq 1$$
$$\chi_S \geq 0, \forall S \in \mathcal{F}$$

Here, our goal is to decide whether the solution of this LP is smaller than a certain value. A.k.a.

$$(LP) \leq L \tag{1}$$

We will then set up a sequential game to demonstrate how to decide if there is a solution that "almost" satisfies (1).

1.1 Game setting

For $t = 1, 2, \cdots, T$:

1. The allocater picks distribution of $\vec{p^t}$ over to (1), which gives $\sum_{e \in V} \vec{p^t}(\sum_{e \in S} \chi_S) \geq 1$.
2. The adversary finds a solution χ_S^T s.t.
$$\sum_{e \in U} \vec{p^t}(\sum_{e \in S} \chi_S^t) \geq 1 \tag{2}$$
$$\chi_S^t \geq 0 \tag{3}$$
3. cost vector $\vec{c^t}(e) = \sum_{e \in S} \chi_S - 1$

1.2 Observation

The key here is that if at some time t, adversary asserts no solution, then there will be no solution for $(LP) \leq L$. That's to say, we could decide if there exists a solution of (1) based on whether the game continues to the very end.

Lemma 2. *If the game reaches the final day, then*

$$\frac{1}{T} \sum_{t=1}^{T} \vec{p^t} \vec{c^t} \leq \frac{1}{T} \sum_{t=1}^{T} \vec{c^t}(e) + \epsilon$$

then there exists a solution to $(LP) \leq \frac{L}{1-\epsilon}$

Proof. $\forall S$, Let $\chi_S = \frac{1}{T}\sum_1^T \chi_S^t$.

Then, we will have:

$$\sum_{e\in S}\chi_S - 1 = \frac{1}{T}(\sum_{t=1}^{T}\chi_S^t - 1) = \frac{1}{T}\sum_{t=1}^{T}\vec{c^t}(e) \geq \frac{1}{T}\sum_{t=1}^{T}\vec{p^t}\vec{c^t} - \epsilon \geq -\epsilon$$

This is because $\frac{1}{T}\sum_{t=1}^{T}\vec{p^t}\vec{c^t} = \frac{1}{T}\sum_{t=1}^{T}\vec{p^t}(\sum_{e\in S}\chi_S - 1) \geq 0$.

There for we could see that $\sum_{e\in S}\chi_S \geq 1-\epsilon$. Then we let $\chi'_S = \chi_S/(1-\epsilon)$. Note that here χ'_S is the solution that almost satisfies (1). □

1.3 Deciding the value of ρ and T

Now what we need to do is to solve (2) under the constraint that $\sum_S \chi_S \leq L$.

We only need to find S^* such that $\sum_{e\in S}\vec{p^t}(e)$ is maximized. Then we let $\chi_S^t = \begin{cases} L & S = S^* \\ 0 & \text{o/w} \end{cases}$.

This should take $O(mn)$ time.

Now let's look at T and ρ. Note that $\vec{c^t}(e) = \sum_{e\in S}\chi_S - 1 \in [-1, L-1]$. Thus we could pick $\rho = L$. Then $T = \frac{4L^2\ln n}{\epsilon^2} = O(\frac{m^2\ln n}{\epsilon^2})$

2 MaxCut

Now we are going to look at another example, the Maxcut problem.

Given graph $G = (V, E)$, our goal is to find $S \subset V$ s.t. $\frac{\text{edge}(S,\bar{S})}{|E|}$ is maximized.

Corresponding QIP:

$$\begin{aligned} &\text{maximize} \sum_{(i,j)\in E}\frac{(x_i - x_j)^2}{4} \\ &\text{s.t. } x_i^2 = 1 \end{aligned}$$

Note that $\sum_{(i,j)\in E}\frac{(x_i-x_j)^2}{4}$ is equivalent to $\frac{1}{4m}L\cdot X$, where L is the graph's Laplace matrix, and $x_{ij} = x_i x_j$.

Then the problem is equivalent to:

$$\begin{aligned}&\text{maximize } \frac{1}{4m} L \cdot X\\&\text{s.t. } \forall i \in V, (e_i \cdot e_i^T) \cdot = 1\\&X \succeq 0\end{aligned}$$

Now we are going to determine whether there exists solution satisfies $(SDP) \geq b$. Actually we only need to decide the feasibility of the following constraints and scale X by $\frac{1}{n}$.

$$\begin{aligned}&\frac{n}{4b} L \cdot X \geq 1\\&\forall i \in V, n(e_i \cdot e_i^T) \cdot X = 1\\&X \succeq 0\end{aligned}$$

Let $K = \{X | \text{Tr}(X) = 1, X \succeq 0\}$, we then need to determine if there exists $X \in K$, s.t.

$$\frac{n}{4b} L \cdot X \geq 1 \tag{4}$$

$$ne_i \cdot e_i^T x \geq 1 \tag{5}$$

This is because $\text{Tr}(X) = 1$ and(5) suggests $x_{ii} = \frac{1}{n}$.

We will then set up a sequential game to demonstrate how to decide if there is a solution that can "almost" satisfies requirement .

2.1 Game setting

For $t = 1, 2, \cdots, T$

1. Allocator picks $\vec{p^t}$ over (4) and (5) for all i, which gives

$$(\vec{p^t}(0) \frac{n}{4b} L + \sum_{i=1}^{n} \vec{p^t}(i) \cdot n(e_i \cdot e_i^T)) \cdot X \geq 1. \tag{6}$$

2. Adversary finds a solution X^t in k satisfies (6) if there is a solution.

3. cost vector $\forall i \in \{0, 1, \cdots, n\}, \vec{c^t}_i = \begin{cases} \frac{n}{4b} L \cdot X^t - 1 & i = 0 \\ ne_i \cdot e_i^T \cdot X^t - 1 & i \geq 1 \end{cases}$

2.2 Observation

If the adversary asserts that there is no solution at some time t, then $(SDP) < b$.

2.3 Solver

To solve the system $\begin{cases}(6) \\ X \in K\end{cases}$, we let et $M^t \cdot X$ be the LHS of (6). We now observe that $\exists X \in K$ satisfies the constraints here is equivalent to $\lambda_{\max}(M^t) \geq 1$. If this is the case, we could then use power method to find a solution X^t s.t. $M^t \cdot X \geq 1 - \epsilon$ in $O(\frac{m \ln n}{\epsilon})$ time.

Lemma 3. *If* $\forall i \in \{0, 1, 2, \ldots, n\}$, $\frac{1}{T}\sum_{t=1}^{T} \vec{p^t} \cdot \vec{c^t} \leq \frac{1}{T}\sum_{t=1}^{T} \vec{c^t}(i) + \epsilon/n$, *then there is a solution to* $(SDP) \geq b - 3\epsilon$.

Proof. Now let us first assume that $\frac{1}{T}\sum_{t=1}^{T} \vec{p^t} \cdot \vec{c^t} \leq \frac{1}{T}\sum_{t=1}^{T} \vec{c^t}(i) + \epsilon$ for all $i \in \{0, 1, 2, \ldots, n\}$, and then we will see why we need a stronger assumption (that the additive error is as small as ϵ/n instead of ϵ).

Let $X = \frac{1}{T}\sum_{t=1}^{T} X^t, \forall i \in \{1, 2, 3, \ldots, n\}$. We have

$$\begin{aligned}
& ne_i \cdot e_i^T X - 1 \\
=& \frac{1}{T}\sum_{t=1}^{T}(ne_i \cdot e_i^T X^t - 1) \\
=& \frac{1}{T}\sum_{t=1}^{T} \vec{c^t}(i) \\
\geq& \frac{1}{T}\sum_{t=1}^{T} \vec{p^t}\vec{c^t} - \epsilon \\
=& \frac{1}{T}(\sum_{t=1}^{T} \vec{p^t}(0)(\frac{n}{4b}L \cdot X^t - 1) + (\sum_{i=1}^{n} n(e_i e_i^T)X^t - 1) \cdot \vec{p^t}(i)) - \epsilon \\
=& \frac{1}{T}(\sum_{t=1}^{T}(M^t \cdot X^t - 1) - \epsilon \\
\geq& -2\epsilon
\end{aligned}$$

Therefore for every $i = 1, 2, 3, \ldots, n$, we have

$$n(e_i e_i^T)X \geq 1 - 2\epsilon.$$

□

Similarly we can prove that

$$\frac{n}{4b} L \cdot X \geq b(1 - 2\epsilon).$$

However, such X is not necessarily a feasible solution to the SDP, since we need to make $n(e_i e_i^T)X \geq 1 - 2\epsilon$ to meet the stronger requirement that $n(e_i e_i^T)X \geq 1$ for all $i \in \{1, 2, 3, \ldots, n\}$. The following claim states that if we start with $\epsilon' = \frac{\epsilon}{2n}$, then we can transform the solution to a feasible solution while only losing a little in the objective function.

Claim: We could construct X' out of X s.t.

$$\begin{aligned} X' &\in K \\ \frac{n}{4m} L \cdot X' &\geq b(1 - 2\epsilon - n\epsilon) \\ n e_i e_i^T X' &\geq 1 \end{aligned}$$

2.4 value of ρ and T

To decide ρ, we need to look at $\vec{c}$. Note that $\vec{c^t}_i = \begin{cases} \frac{n}{4b} L \cdot X^t - 1 & i = 0 \\ n e_i \cdot e_i^T \cdot X^t - 1 & i \geq 1 \end{cases} \in [-n, n]$. Thus we pick $\rho = n$.

Then we have $T = O(\frac{n^2 \ln n}{\epsilon'^2}) = O(\frac{n^4 \ln n}{\epsilon^2})$. Thus the overall time is $O(\frac{n^4 \ln n}{\epsilon^2} \cdot \frac{mn \ln n}{\epsilon}) = O(\frac{n^5 \ln^2 nm}{\epsilon^3})$.

Using the similar idea, an improved algorithm [1] (with more careful analysis) uses $O(\frac{nm \text{poly} \log n}{\epsilon^3})$ time.

References

[1] Klein, Philip, and Hsueh-I. Lu. Efficient approximation algorithms for semidefinite programs arising from MAX CUT and COLORING. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing.* ACM, 1996. APA