Lecture 8: Randomized Rounding of LP and SDP

1 Congestion Minimization

1.1 Chernoff Bound

Recall some basic bounds in probability theory.

Theorem 1.1. *(Markov's inequality) For any random variable $X \geq 0$,*

$$\Pr[X > a] \leq \frac{\mathbb{E}[X]}{a}$$

Theorem 1.2. *(Chebyshev's bound) For any random variable $X \geq 0$,*

$$\Pr[|X - \mathbb{E}[X]| \geq a] \leq \frac{var(X)}{a^2}$$

These two bounds hold for any random variable X. Now we consider a more restricted form of random variable X. Suppose $X = \sum_{i=1}^{n} X_i$ such that $X_i \in \{0,1\}$ and $\mathbb{E}[X_i] = p_i$ for all $i \leq n$. Then notice that $\mu = \mathbb{E}[X] = \sum_i p_i$.

The following is a popular form of Chernoff bound that is probably easiest to use.

Theorem 1.3. *(Chernoff bound, small δ) For any $\delta \in (0,1)$,*

$$\Pr[X > (1+\delta)\mu] \leq e^{-\delta^2\mu/3}$$

And

$$\Pr[X < (1-\delta)\mu] \leq e^{-\delta^2\mu/2}$$

Theorem 1.4. *(Chernoff bound, large δ) For all positive value δ,*

$$\Pr[X > (1+\delta)\mu] \leq e^{-\delta^2\mu/(2+\delta)}$$

1.2 Discrepancy

We are given a set system $(X, \mathcal{S})$ where X is a ground set, and we have $S \subseteq X$ for each $S \in \mathcal{S}$. Let $\chi : X \rightarrow \{-1, 1\}$ be a coloring of X by two colors. Discrepancy of function χ is $disc(X, \mathcal{S}, \chi) = \max_{S \in \mathcal{S}} |\sum_{x \in S} \chi(x)|$. Now, the discrepancy of the set system $(X, \mathcal{S})$ is defined as

$$disc(X, \mathcal{S}) = \inf_{\chi} disc(X, \mathcal{S}, \chi)$$

We will use Chernoff bound to prove the following theorem.

Theorem 1.5. *For any set system $(X, \mathcal{S})$ with $|X| = n$ and $|\mathcal{S}| = m$, we have $disc(X, \mathcal{S}) = O(\sqrt{n \log m})$*

Proof. We show that a random coloring obtains this bound with high probability. For each $x \in X$, we pick a random color $\chi(x) \in \{-1, 1\}$ uniformly.

Lemma 1.1. *For any set $S \in \mathcal{S}$,*

$$\mathsf{Pr}\left[|\sum_{x\in S}\chi(x)| > 10\sqrt{n\log m}\right] \le 1/2m$$

Proof. Let $Y_x = (1+\chi(x))/2$ for each $x \in X$. Notice that $Y_x \in \{0,1\}$ and that $|\sum_x \chi(x)| \ge \lambda$ if and only if$|\sum_x Y_x - \mathbb{E}\left[\sum_x Y_x\right]| \ge \lambda/2$. So we only need to bound the probability that $|\sum_{x\in S} Y_x - \mathbb{E}\left[Y\right]| \ge 5\sqrt{n\log m}$. For any set S, the expectation $\mathbb{E}\left[\sum_{x\in S} Y_x\right] = |S|/2$. Now we can now apply Chernoff bound to say that:

$$\mathsf{Pr}\left[|\sum_{x\in S} Y_x - \mathbb{E}\left[\sum_{x\in S} Y_x\right]| \ge (10\sqrt{n\log m}/|S|)\mathbb{E}\left[\sum_{x\in S} Y_x\right]\right] \le e^{\frac{-100n\log m|S|}{6|S|^2}} \le 1/2m$$

$\square$

From this lemma, we can apply union bound over all sets $S \in \mathcal{S}$ to get the desired result. $\square$

1.3 Congestion Minimization

We start from another extreme of Edge Disjoint Paths, called Congestion Minimization. In this problem, we are given a graph $G = (V,E)$ and k terminal pairs $\{(s_i,t_i)\}_{i=1}^k$. For each $i \in [k]$, we want to find a path P_i connecting s_i to t_i. Given a collection of paths $\mathcal{P} = \{P_1,\ldots,P_k\}$, the congestion is defined as $\max_i cong(e)$ where $cong(e) = |\{i : e \in P_i\}|$. Our goal is to route all paths while minimizing the maximum congestion.

A natural approach to attack this problem is to start from LP relaxation. For each pair (s_i,t_i), let $\mathcal{P}_i$ denote the set of all paths connecting s_i to t_i. For each path $P \in \mathcal{P}_i$, we have variable x_P which takes the value of 1 if this path is chosen in the solution.

$$\begin{aligned}
&\text{(LP)} \\
&\min \quad W \\
&\text{s.t.} \quad \sum_{i=1}^{k}\sum_{P\in\mathcal{P}_i : e\in P} x_P \le W \text{ for all } e \in E \\
&\qquad \sum_{P\in\mathcal{P}_i} x_P = 1 \text{ for all } i \in [k]
\end{aligned}$$

Notice that this LP has exponential number of variables. We cannot even represent it, but there is an equivalent LP with polynomial number of variables. We will not discuss that point here. Let us assume that we can solve this LP and see how this implies an approximation algorithm.

Algorithm: Now we can treat $\{x_P\}_{P\in\mathcal{P}_i}$ as a probability distribution. For each $i \in \mathcal{P}_i$, we choose a path P_i according to this probability (so each path P is chosen with probability exactly x_P). Ntice that, for each edge e, $\mathbb{E}\left[cong(e)\right] \le W$. Now we can apply Chernoff bound with $\delta = 100\log n$, so we have $\mathsf{Pr}\left[cong(e) > 100\log nW\right] \le e^{-25W\log n} \le n^{-25}$. By applying union bound, the probability that there is an edge $e \in E$ with congestion more than $100\log nW$ is very low.

If we use another formulation of Chernoff bound, this approximation factor can be made $O(\log n/\log\log n)$. This remains the best known bound.

1.4 Lovasz Local Lemma (LLL)

We have seen the use of union bounds many times, where we usually write some bad event $\mathcal{E}$ as $\mathcal{E} = \bigcup_{j\in[k]} \mathcal{E}_j$, and then upper bound the bad event by union bound $\Pr[\mathcal{E}] \le k\Pr[\mathcal{E}_j]$. If k is relatively low, the union bound is generally enough, but what if k is large, i.e. much larger than $1/\Pr[\mathcal{E}_j]$?

Theorem 1.6. *(LLL, symmetric version) Let A_i, $i = 1, \dots, m$ be a collection of (bad) events such that $\Pr[A_i] \le p$ for all i. Moreover, each event A_i only depends on d other events with $ep(d+1) \le 1$. Then $\Pr[\bigcup_{i=1}^n A_i] < 1$.*

Notice that, we only need $p \le 1/e(d+1)$ in order to be able to say that all bad events are avoided.

2 Limits of LP based approach: Integrality Gap

We can look at LP as a model of computation in which algorithms try to compute a feasible solution by rounding linear programs. Then we can study *limits of computation* in this restricted model. Given a minimization problem Π and a linear programming relaxation (LP) for Π, we know that, for any instance $J \in \Pi$, $\mathsf{OPT}_{LP}(J) \le \mathsf{OPT}(J)$. If an algorithm were to produce an α approximation by rounding this LP, it will always produce a solution whose cost is at most $\alpha\mathsf{OPT}_{LP}(J)$ for any instance J. So this will end up bounding $\mathsf{OPT}(J)/\mathsf{OPT}_{LP}(J)$ by a factor of α for any instance J of Π. By this reasoning, if there is any instance J such that $\mathsf{OPT}(J)/\mathsf{OPT}_{LP}(J) > \alpha$, this would imply that obtaining α approximation by rounding (LP) is not possible.

More formally, we define the integrality gap of problem Π with respect to (LP) as:

$$\sup_{J\in\Pi} \frac{\mathsf{OPT}(J)}{\mathsf{OPT}_{LP}(J)}$$

To show a lower bound, it is enough to exhibit one example $J \in \Pi$ such that $\mathsf{OPT}(J)/\mathsf{OPT}_{LP}(J) > \alpha$. To show an upper bound, we need an approximation algorithm.

2.1 Set Cover

We show an example of instance where an LP solution only costs 2, while any integral solution must cost at least $\Omega(\log n)$. We start from sets $S_1, \dots, S_m$. For each $I \subseteq [m] : |I| = m/2$, we have an element $e(I)$ such that $e(I) \in S_j$ if and only if $j \in I$. The number of elements is $n = \binom{m}{m/2} = O(2^m)$. Recall that we have the following LP relaxation:

$$\text{(LP)} \qquad \begin{aligned} \min \quad & \sum_i x_i \\ \text{s.t.} \quad & \sum_{i: e\in S_i} x_i \ge 1 \text{ for all } e \in E \end{aligned}$$

Fractional solution: Notice that each element $e(I)$ is contained in $m/2$ sets, so a fractional solution can assign $2/m$ to each set to ensure that each element is covered, i.e. $x_i = 2/m$ for all $i \in [m]$.

Integral solution: Now we argue that any integral feasible solution must choose at least $m/2$ sets. Suppose that $J \subseteq [m]$ be indices of the sets such that $|J| < m/2$, so we must have $J' \subseteq [m] \setminus J : |J'| = m/2$ such that $e(J')$ is not contained in any sets chosen.

2.2 Machine Minimization

Given a set of jobs $J = \{1, \ldots, n\}$ where each job is equiped with a collection $\mathcal{J}_j$ of intervals on the line. The goal is to choose, for each job $j \in J$, an interval $I_j \in \mathcal{J}_j$, such that the maximum congestion on the line formed by intervals $I_1, \ldots, I_n$ is minimized.

Now we can write an relaxation with variable $x(j, I)$ for each job $j \in [n]$ and interval $I \in \mathcal{J}_j$. We have a constraint $\sum_{I \in \mathcal{J}_j} x(j, I) = 1$ for each job j, and a constraint $\sum_j \sum_{I: p \in I} x(j, I) \leq W$ for each point p on the line. Using the same randomized rounding algorithm, we can get $O(\log n / \log\log n)$ approximation algorithm. Now we show that the integrality gap of this LP relaxation is also $\Omega(\log n / \log\log n)$.

Let M be a parameter. Our construction is recursive and will produce collections of jobs $J_1 \cup \ldots, \cup J_M$ where J_i is a collection of jobs at level i. Job at level 1 is simply $j(1)$ with interval set $\mathcal{J}_{j(1)} = [0, 1]$. Assume that jobs in J_i have been defined such that intervals of these jobs are disjoint, and we will define J_{i+1}. Let I be an interval at level i. We will have a job $j(i+1, I)$ at level $i+1$ where interval I is divided into M equal-length intervals $I_1, \ldots, I_M$ and $\mathcal{J}_{j(i+1,I)} = \{I_1, \ldots, I_M\}$. This construction guarantees that the intervals at level $i+1$ would also be disjoint and that $|J_{i+1}| = M^i$. The LP solution assigns $x(j, I) = 1/M$ to every interval in the instance. Since intervals in J_i are disjoint, there can be at most M intervals sharing a point, and this implies that the maximum congestion is at most 1 for the LP solution.

Lemma 2.1. *Any integral solution must have congestion at least M.*

This creates the integrality gap of at least M, while the size of the instance is $M^{O(M)}$. In other words, $M = \Omega(\log n / \log\log n)$.

3 Semidefinite Program

A natural LP relaxation for Maximum Cut cannot has integrality gap lower bound of 2, so one cannot expect better algorithms from it. Here, we show a stronger relaxation, called Semidefinite Programming (SDP) which can be applied to Maximum Cut to get an improved approximation ratio.

3.1 Basic

We first recall some basic of linear algebra. A matrix $(X)_{n\times n}$ is *positive semidefinite* (PSD) if and only if for all vectors $v \in \mathbb{R}^n$, we have $v^T X v \geq 0$.

Proposition 1. *The following are equivalent:*

- *X is psd.*
- *X has non-negative eigenvalues*
- *$X = V^T V$ for some matrix $V \in \mathbb{R}^{m\times n}$ where $m \leq n$*
- *$X = \sum_{i=1}^n \lambda_i w_i w_i^T$ for some $\lambda \geq 0$ and vectors $w_i \in \mathbb{R}^n$ such that w_i are othogonal.*

Notice that the third statement implies that we have a collection of vectors $v_1, \ldots, v_n$ such that $v_i \cdot v_j = X_{ij}$. In semidefinite program, we have variables x_{ij} for $i, j = 1, \ldots, n$, and any linear constraints on them such that $x_{ij} = x_{ji}$. Finally, we have the constraint $X \succeq 0$. Equivalently, we may replace X_{ij} by $v_i \cdot v_j$, and this is called vector program.

3.2 SDP Relaxation

We are given a graph $G = (V, E)$ where each edge $ij \in E$ has weight $w_{ij} \geq 0$. Consider the following formulation of the problem.

$$\begin{array}{ll} \max & \frac{1}{2} \sum_{ij \in E} w_{ij}(1 - y_i y_j) \\ \text{s.t.} & y_i \in \{-1, 1\} \end{array}$$

We first argue that this is the correct formulation. For any solution $S \subseteq V$, define $y_i = -1$ for all $i \in S$ and $y_i = 1$ otherwise. If $ij \in E$ crosses the cut, the term $1 - y_i y_j$ evaluates to 2, so we get the contribution of w_{ij} in the objective; otherwise, the contribution is zero for all ij that is not cut. The converse can be argued similarly.

Now we can relax the constraints $y_i \in \{-1, 1\}$ to $v_i \in \mathbb{R}^n$ instead. This is a relaxation because the solution y of the previous program can be turned into $v_i = (y_i, 0, \ldots, 0)$ without changing the result. Also, we can add the constraint $v_i \cdot v_i = 1$.

3.3 Randomized Rounding

Think of each vector $v_i \in \mathbf{R}^n$ as a vector on unit sphere. Notice that the temr $1 - v_i \cdot v_j = 1 - \cos\theta_{ij}$ where θ_{ij} is the angle between v_i and v_j. This term has large value when the angle is large, so we can think of the SDP as trying to "embed" the vertices onto the sphere, trying to ensure that two endpoints of edges are separated.

Algorithm: We pick a random vector $\mathbf{r} \in \mathbb{R}^n$ and define $S = \{i : \mathbf{r} \cdot v_i > 0\}$. Now the probability that each edge $ij \in E$ is cut is exactly $\theta_{ij}/\pi = \arccos(v_i \cdot v_j)/\pi$. This implies that the expected value of the solution is $\sum_{ij \in E} w_{ij} \frac{1}{\pi} \arccos(v_i \cdot v_j)$. In the SDP, the contribution of $ij \in E$ is $\frac{1}{2}(1 - v_i \cdot v_j)$, so it is enough to show the following.

Lemma 3.1. *For any unit vectors v_i, v_j,*

$$\frac{\frac{1}{\pi}\arccos(v_i \cdot v_j)}{\frac{1}{2}(1 - v_i \cdot v_j)} = \frac{\frac{1}{\pi}\theta_{ij}}{\frac{1}{2}(1 - \cos\theta_{ij})} \geq 0.878$$

This can be verified by computers :)