In this lecture we will see some more examples of approximation algorithms based on LP relaxations. This time we will use randomness in rounding the fractional solutions to integral ones and show that the value of our solution is good in expectation. For some problems it will be useful to use some advanced tools from probability theory called concentration bounds. We will discuss these also.

11.1 Set Cover

Let $E = \{e_1, e_2, .., e_n\}$ be a set of n elements. Let $S_1, S_2, .., S_m$ be subsets of E with associated costs $c_1, .., c_m$. In the set cover problem, our goal is to pick a minimum cost collection of sets from among $S_1, \cdots, S_m$, such that the union of these sets is E; in other words, this collection "covers" E. The set cover problem is NP-hard. In fact, it is a generalization of vertex cover, because in the latter, we can think of the edges as the elements and the vertices as sets of elements containing the edges incident on them. We will now show an LP relaxation based approach to solving this problem with an $O(\log n)$ approximation.

11.1.1 ILP formulation of set cover

Let x_i be a random variable associated with each subset S_i. We intend for x_i to be 1, if S_i is in the solution, and 0 otherwise. The following ILP encodes set cover.

$$\begin{aligned} \text{Minimize } & \sum_{i=1}^{m} c_i x_i \text{ s.t.} & & \\ & \sum_{i:e\in S_i} x_i \geq 1 & \forall e \in E & \qquad (11.1.1) \\ & x_i \in \{0,1\} & \forall i & \end{aligned}$$

The constraints in 11.1.1 ensure that every element is present in at least one of the chosen subsets. It can be noted that this problem formulation is a generalization of the vertex cover. If we map every edge to an element in E, and every vertex to one of the the subsets of E, the vertex cover is the same as the set cover problem with the additional constraint that every element e appears in exactly two subsets (corresponding to the two vertices it is incident on).

We can relax the last integrality constraint to $x_i \in [0, 1]$, obtaining a linear program. Solving the linear program gives us a fractional solution with cost no more than the cost of the optimal set cover.

11.1.2 Deterministic rounding

First, let us see the approximation we get by using a deterministic rounding scheme analogous to the one we used in the vertex cover problem.

Theorem 11.1.1 *There exists a deterministic poly-time rounding scheme which gives us an F-approximation to the solution, where F is the maximum frequency of an element (i.e. the number of sets an element belongs to).*

Proof: Let F be the maximum frequency of an element, or the maximum number of subsets an element appears in. Let $x_1, .., x_m$ be the optimal solution to the LP. Our algorithm picks all sets S_i with $x_i \geq \frac{1}{F}$. We first note that this is a feasible solution—in every constraint in 11.1.1 we have atmost F variables on the LHS, and atleast one of them should be $\geq \frac{1}{F}$; So we will set at least one of those variables to 1 during the rounding, and every element will be covered. It can be noted here that the vertex cover has $F = 2$, and so we used a rounding threshold of $\frac{1}{2}$ during the previous lecture. Next consider the cost of our solution. Let $x'_i = 1$ if we pick the set S_i and 0 otherwise. Then, $x'_i \leq Fx_i$ for all i. Therefore, the algorithm's cost is $\sum_i x'_i c_i \leq F \sum_i x_i c_i$, where the latter is a lower bound on the optimal cost. ■

11.1.3 Randomized rounding

We will now study the following randomized rounding algorithm.

1. Solve the LP.
2. (Randomized rounding). $\forall$ i, pick S_i independently with probability x_i.
3. Repeat step 2 until all elements are covered.

The intution behind the algorithm is that higher the value of x_i in the LP solution, the higher the probability of S_i being picked during the random rounding.

In order to analyze the algorithm, we will first compute the expected cost of every iteration and then bound the number of iterations. Note that the number of iterations is not independent of the cost of an iteration. So we will instead consider the following version of the algorithm that decouples the two variables:

1. Solve the LP.
2. Repeat for $\log n + 2$ steps:
   $\forall$ i, pick S_i independently with probability x_i.
3. If the solution in step 2 does not cover all elements or has cost more than $(4 \log n + 8)$ times the cost of the LP solution, repeat step 2.

This time, in each iteration we perform randomized rounding $2 + \log n$ times. We will show that we only need less than two iterations in expectation to satisfy both the conditions in step 3.

We begin by analyzing the cost of a single rounding step.

Lemma 11.1.2 *The expected cost of each rounding step in step 2 is equal to the cost of the LP solution.*

Proof: Let Y_i be an indicator random variable with $Y_i = 1$ if S_i is picked, and 0 otherwise. Let $Y = \sum_{i=1}^{n} c_i Y_i$. Then $E[Y] = \sum_{i=1}^{n} c_i E[Y_i]$ by linearity of expectation, which is equal to $\sum_{i=1}^{n} c_i x_i$, the cost of the LP solution. ■

The following simple corollary follows from linearity of expectation and Markov's inequality.

Corollary 11.1.3 *The expected cost of the collection of sets picked in any iteration of step 2 is* $(\log n + 2)$ *times the LP cost. The probability that this cost is more than* $(4 \log n + 8)$ *times the LP cost is at most* $1/4$.

Next we consider the probability that all elements get covered in one iteration of step 2. In order to do so, fix some element $e \in E$ and consider the probability that e is not covered in any of the $c \log n$ rounding steps. Using the independence in picking each set we get

$$\begin{aligned}\mathbf{Pr}[e \text{ is not covered in any one execution of the rounding step}] &= \prod_{i:e\in S_i} \mathbf{Pr}[S_i \text{ is not picked}] \\ &= \prod_{i:e\in S_i} (1 - x_i)\end{aligned}$$

We know that $\sum_{i:e\in S_i} x_i$ is at least 1. To relate the above product to this sum, we use a simple but useful inequality: for all $x \in [0,1]$, $(1 - x) \leq e^{-x}$. So we get that the above product is

$$\leq \prod_{i:e\in S_i} e^{-x_i} = e^{-\sum_{i:e\in S_i} x_i} \leq \frac{1}{e}$$

where the last inequality follows from the fact that $\sum_{i:e\in S_i} x_i \geq 1$.

Now note that in each iteration of the algorithm we execute the rounding step $c \log n$ times. The probability that e is not covered in any of those rounding steps is at most

$$(1/e)^{\log n + 2} < \frac{1}{4n}$$

Now we can take the union bound over the above probabilities, one for each element $e \in E$ to get that the probability that there exists some element that is not covered in a single iteration is at most n times $1/4n$, that is $1/4$.

Finally, the probability that the collection picked in some iteration of step 2 of the algorithm fails to satisfy the two conditions in step 3 is at most the probability that either one of two events happen: (1) not all elements are covered, or (2), the cost of the solution exceeds $(4 \log n + 8)$ times the LP cost. Each of these events have a probability of at most $1/4$. So using the union bound, the probability that one of these events happens is at most 2. Now it follows that step 2 is executed at most twice in expectation.

Theorem 11.1.4 *The algorithm above is a* $O(\log n)$ *approximation to set cover.*

11.2 Concentration Bounds

In analyzing randomized techniques for rounding LP solutions, it is useful to determine how close to its expectation a random variable is going to turn out to be. This can be done using concentration bounds: We look at the probability that given a certain random variable X, the probability that X lies in a particular range of values (Say, the deviation from the expectation value). For instance, we want $\mathbf{Pr}[X \geq \lambda]$ for some value of λ. Note that if a random variable has small variance, or (as is often the case with randomized rounding algorithms) is a sum of many independent random variables, then we can give good bounds on the probability that it is much larger or much smaller than its mean.

11.2.1 Markov's Inequality

The simplest concentration bound is Markov's inequality. We proved this in a previous lecture. Given a random variable $X \geq 0$, we have,

$$\mathbf{Pr}[X \geq \lambda] \leq \frac{\mathbf{E}[X]}{\lambda} \tag{11.2.2}$$

This simple inequality extends to any arbitrary non-negative function applied to X: given some $f : X \to \Re^+ \cup \{0\}$,

$$\mathbf{Pr}[f(X) \geq f(\lambda)] \leq \frac{\mathbf{E}[f(X)]}{f(\lambda)} \tag{11.2.3}$$

If the function f is monotonically increasing then in addition to (11.2.3), the following also holds:

$$\mathbf{Pr}[X \geq \lambda] = \mathbf{Pr}[f(X) \geq f(\lambda)] \leq \frac{\mathbf{E}[f(X)]}{f(\lambda)} \tag{11.2.4}$$

11.2.2 Chebyshev's Inequality

Other concentration results can be obtained by using Markov's inequality with an appropriate choice for f. We first study a tighter bound called the Chebyshev's inequality. Let $f(X) = (X - \mathbf{E}[X])^2$. Note that f is an increasing function when $X > \mathbf{E}[X]$.

We have,

$$\mathbf{Pr}[|X - \mathbf{E}[X]| \leq \lambda] = \mathbf{Pr}\left[(X - \mathbf{E}[X])^2 \leq \lambda^2\right] \tag{11.2.5}$$

$$= \frac{\mathbf{E}\left[(X - \mathbf{E}[X])^2\right]}{\lambda^2} = \frac{\sigma^2(X)}{\lambda^2} \tag{11.2.6}$$

That is, the deviation of X from $\mathbf{E}[X]$ is a function of its variance($\sigma(X)$). If the variance is small, then we have a tight bound.

The bound also imples that with probability p, $X \in \mathbf{E}[X] \pm \sqrt{\frac{1}{p}}\sigma(X)$.

11.2.3 Chernoff-Hoeffding bounds

Next we present the Chernoff-Hoeffding bound which can sometimes be much tighter compared to Chebyshev's inequality. As an aside, note that the tighter the bound we want, the more assumptions we need on the random variables. For example, Markov's inequality holds for any non-negative random variable, whereas Chebyshev's inequality needs the variance to be bounded and small in order to be useful. Chernoff-Hoeffding will require much stronger assumptions.

Let random variables X_1, X_2, ..., X_n be independent and identically distributed random variables, where $X_i \in [0, 1]$. Note that the class of indicator random variables lies in this category.

Let $X = \sum_{i=1}^{n} X_i$. Also $\mathbf{E}[X] = \sum_{i=1}^{n} \mathbf{E}[X_i] =: \mu$.

Then for any $\delta > 0$,

$$\mathbf{Pr}[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \tag{11.2.7}$$

The following is sometimes a more useful form of the inequality for $\delta > 0$,

$$\mathbf{Pr}[X \geq (1+\delta)\mu] \leq e^{-\left(\frac{\delta^2}{2+\delta}\right)\mu} \tag{11.2.8}$$

Note that the above probability resembles a gaussian/bell curve. When $0 \leq \delta \leq 1$ we can get a simpler expression:

$$\mathbf{Pr}[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2}{3}\mu} \tag{11.2.9}$$

$$\mathbf{Pr}[X \leq (1-\delta)\mu] \leq e^{-\frac{\delta^2}{2}\mu} \tag{11.2.10}$$

Example: We will now use coin tossing as an example to show how the three bounds apply and compare. We toss n independent unbiased coins. We want to find a t such that: $\mathbf{Pr}$[We get at least t heads in n tosses] $\leq 1/n$.

Let us use indicator random variables to solve this: If the ith coin toss is heads, let $X_i = 1$, otherwise $X_i = 0$. Let $X = \sum_{i=1}^{n} X_i$. Here X is the total number of heads we get in n independent tosses. Note that $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} \mathbf{E}[X_i] = n/2$, since $\mathbf{E}[X_i] = 1/2$ (the probability of getting a heads).

To get bounds on $\mathbf{Pr}[X \geq t]$, we first apply Markov's inequality:

$$\mathbf{Pr}[X \geq t] \leq \frac{\mathbf{E}[X]}{t} = \frac{n/2}{t}$$

$$\mathbf{Pr}[X \geq t] = 1/n \Rightarrow \frac{n/2}{t} = 1/n \Rightarrow t = n^2/2$$

But, we don't do any more than n coin tosses, so this bound is not useful. Note that Markov's bound is very weak.

Applying Chebyshev's inequality:

$$\mathbf{Pr}[X \geq t] \leq \mathbf{Pr}[|X - \mu| \geq t - \mu] \leq \frac{\sigma^2(X)}{(t-\mu)^2}$$

Evaluating $\sigma(X)$ where $X_i \in \{0, 1\}$,

$$\mathbf{E}[X_i] = 1/2$$

$$\sigma^2(X_i) = \mathbf{E}\big[(X_i - \mathbf{E}[X_i])^2\big] = \frac{1}{2}(\frac{1}{4}) + \frac{1}{2}(\frac{1}{4}) = 1/4$$

Since this is a sum of independent random variables, the variances can be summed together:

$$\sigma^2(X) = \sum_{i=1}^{n} \sigma^2(X_i) = n/4$$

Evaluating t,

$$\mathbf{Pr}[X \geq t] \leq \frac{\sigma^2(X)}{(t-\mu)^2} = 1/n$$

$$\Rightarrow t = \mu + n/2 = n$$

Again this bound is quite weak.

Finally, applying Chernoff-Hoeffding bounds we get:

$$\mathbf{Pr}[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2}{3}\mu} = e^{-\frac{\delta^2}{6}n}$$

Setting $e^{-\frac{\delta^2}{6}n} = 1/n$ we get,

$$\delta = \sqrt{\frac{6 \log n}{n}}$$

$$\implies t = (1+\delta)\mu = n/2 + \sqrt{\frac{3n \log n}{2}}$$

■

11.3 Routing to minimize congestion

We will now see another example of randomized rounding as well as an application of Chernoff bounds to the analysis.

We are given a graph G = (V, E) and k "commodities". Each commodity i has a source s_i and a destination t_i. The goal is to find an (s_i, t_i) path in the graph G for every commodity i, while minimizing the maximum congestion along any edge. The latter can be written formally as Minimize congestion $C = \max_{e \in E} |\{i : P_i \ni e\}|$, where P_i is an (s_i, t_i) path for commodity i.

It can be noted that this problem is similar to a network flow problem. But it is not a network flow problem because for each commodity i, we need to have a single path between s_i and t_i. We cannot have the flow splitting along multiple branches.

This problem is an NP-hard problem. So we will solve it by formulating it as an ILP problem, relaxing it to an LP problem, solving the LP, and rounding the solution to an ILP solution.

11.3.1 ILP formulation with exponential number of variables and constraints

Let $i = 1, \dots, k$ be the k commodities. Let $\mathcal{P}_i$ be the set of all paths from s_i to t_i. We have a variable x_P for every $P \in \cup_i \mathcal{P}_i$.

$$
\begin{aligned}
&\text{Minimize } t \ s.t. && && \text{(Path-ILP)}\\
&\sum_{P \in \mathcal{P}_i} x_P = 1 && \forall i &&\\
\sum_i &\sum_{P \in \mathcal{P}_i, P \ni e} x_P \le t && \forall e \in E && (11.3.1)\\
&t \ge 0 &&&&\\
&x_P \in \{0,1\} && \forall P &&
\end{aligned}
$$

Note that the actual objective function which is in a min-max form is not linear. So, we employ the trick of introducing a new variable t. We introduce the constraint (11.3.1) that congestion on any edge is $\le t$, and so all we need to do now is to minimize t, which is equivalent to minimizing the maximum congestion.

One problem with this ILP formulation is that we can have an exponential number of paths between s_i and t_i (for example, in a completely connected graph). So even if we relax it to an LP problem, we will still have an exponential number of variables and constraints. We would like a formulation with a polynomial (polynominal in $|V|$, $|E|$, and k) number of variables and constraints. Next we consider an alternative ILP formulation.

11.3.2 ILP formulation with polynomial number of variables and constraints

In order to obtain a polynomial size LP, instead of encoding in the program whether a commodity uses a particular path, we encode whether a particular commodity uses a particular edge in the graph. We have variables $x_{e,i}$, $\forall e \in E, i = 1, \dots, k$. $x_{e,i} = 1$, if $e \in P_i$, and 0 otherwise, where P_i is the chosen (s_i, t_i) path for commodity i. The new ILP formulation is as follows.

$$
\begin{aligned}
&\text{Minimize } t \ s.t. && && \text{(Flow-ILP)}\\
&\sum_{e \in \delta^+(t_i)} x_{e,i} = \sum_{e \in \delta^-(s_i)} x_{e,i} = 1 && \forall i && (11.3.2)\\
&\sum_{e \in \delta^+(v)} x_{e,i} = \sum_{e \in \delta^-(v)} x_{e,i} && \forall i, \forall v \ne s_i, t_i && (11.3.3)\\
&\sum_i x_{e,i} \le t && \forall e \in E && (11.3.4)\\
&x_{e,i} \in \{0,1\} && \forall e \in E, i \in [k] &&
\end{aligned}
$$

In the above formulation, δ^+ indicates the flow coming into a vertex, and δ^- indicates the flow going out of a vertex. Constraint (11.3.2) makes sure that we have a unit flow being routed out of every s_i, and a unit flow being routed into every t_i. Constraint (11.3.3) is like flow conservation at every other vertex. Congestion along an edge is just the number of commodities being routed along that edge, and constraint (11.3.4) ensures that $t \geq$ congestion along any edge. The objective function is to minimze t.

The problem can be relaxed to an LP problem by letting $x_{e,i} \in [0,1]$. It can be noted that this formulation (let's call it the edge formulation) is equivalent to the previous exponential-size formulation (let's call that the path formulation). While the edge-formulation is easier to solve, being polynomial in size, the path formulation is easier to round. So, given a solution to the edge formulation $\{x_{e,i}, t\}$, we can obtain a solution to the path formulation $\{x_P, t\}$ with the following properties: t is the same as before; the solution is feasible for Path-ILP; for any edge e and commodity i, $\sum_{P \in \mathcal{P}_i, P \ni e} x_P = x_{e,i}$. (That this can be done in polynomial time is left as an exercise to the reader). Now, we have a fractional solution $\{x_P\}$ for each variable x_P. We also know that t^*, the solution to the LP, is no more than C^*, the optimal solution to the ILP. Now, we need to round the LP solution to an ILP solution such that we achieve a reasonable approximation to t^*.

11.3.3 Randomized rounding

Our rounding approach is similar to that for set cover. For every commodity i, consider the probability distribution on $\mathcal{P}_i$ given by $\{x_P\}_{P \in \mathcal{P}_i}$, and pick a path in $\mathcal{P}_i$ according to this probability distribution. Call this path P_i. For example, say we have three paths with values 0.5, 0.3, and 0.2. Get a random number between 0 and 1. If the number is between 0 and 0.5, pick the first path. If the number is between 0.5 and 0.8, pick the second path, and if the number is between 0.8 and 1, pick the third path. This gives us a feasible solution and now we will analyze the cost of this solution.

The good thing about this rounding is that the expected load on an edge is exactly the same as the load on that edge in the LP. In particular, for any edge e, $\mathbf{Pr}[\text{commodity i is routed along e}] = x_{e,i}$. To compute the load on edge e, we will use indicator random variables $X_{e,i}$ denoting whether commodity i uses the edge e. Let $X_e = \sum_i X_{e,i}$ denote the number of commodities i with $e \in P_i$. In other words, X_e is a random variable that denotes the congestion on edge e.

Now, $X_{e,i}$ is 1 if any of the paths $P \in \mathcal{P}_i$ with $e \in P$ is chosen as the final path P_i. Since these are all disjoint events with probabilities x_P respectively, the probability that $X_{e,i}$ is equal to 1 is exactly $\sum_{P \in \mathcal{P}_i, P \ni e} x_P = x_{e,i}$. Therefore we get

$$\mathbf{E}[X_e] = \sum_i \mathbf{E}[X_{e,i}] = \sum_i x_{e,i} \leq t^*$$

So the expected congestion on any one edge is at most t^*. However, since we have many edges in the graph, some of these may end up with congestion much larger than the expectation. We would like to show that there exists some number λ such that for every edge, $\mathbf{Pr}[X_e \geq \lambda t^*] \leq \frac{1}{n^3}$. Then by the union bound, $\mathbf{Pr}[\exists \text{ edge } e \text{ s.t. } X_e \geq \lambda t^*] \leq \frac{|E|}{n^3} \leq \frac{1}{n}$, and we would get a λ approximation with a high probability.

What value of λ satisfies this property? Note that X_e is a sum of a number of independent $[0,1]$ random variables, so we can apply Chernoff's bound. Let μ denote the expected value of X_e. We will also use the fact that $C^* \geq \max\{t^*, 1\}$, so $\mu \leq C^*$.

Suppose first that $\mu \geq 1$. So we get using Chernoff bounds:

$$\begin{aligned} \mathbf{Pr}[X_e > \lambda C^*] &\leq \mathbf{Pr}[X_e > \lambda\mu] \\ &\leq \left(\frac{e^{\lambda-1}}{\lambda^\lambda}\right)^\mu \\ &\leq (\lambda/e)^{-\lambda\mu} \leq (\lambda/e)^{-\lambda} \end{aligned}$$

Setting $\lambda^\lambda \approx n$, we get

$$\lambda = O\left(\frac{\log n}{\log\log n}\right)$$

When $\mu < 1$, a similar analysis goes through. Specifically,

$$\begin{aligned} \mathbf{Pr}[X_e > \lambda C^*] \leq \mathbf{Pr}[X_e > \lambda] &\leq \mathbf{Pr}\left[X_e > \frac{\lambda}{\mu}\mu\right] \\ &\leq (\lambda/e\mu)^{-\mu\times\lambda/\mu} \leq (\lambda/e)^{-\lambda} \end{aligned}$$

Once again, setting $\lambda^\lambda \approx n$, gives $\lambda = O\left(\frac{\log n}{\log\log n}\right)$.

Therefore, with a high probability our congestion is no more than $O(\log n/\log\log n)$ times the optimal.