Randomized Algorithms, Summer 2016	Lecture 11 (4 pages)

Randomized Rounding

Instructor: Thomas Kesselheim

One of the key ideas in randomized algorithms is that good outcomes are likely enough to happen if we flip coins. This allows us to avoid distinguishing many cases and treating each separately. In this lecture, we will study one particular such technique, which allows us to design approximation algorithms. In all of our algorithms, we will rely on linear programs (LPs). The examples are selected in a way that it is not necessary to have much background in LP theory. As a matter of fact, we will only need that LPs of polynomial size can be solved in polynomial time.

1 Set Cover

As a first example, we study the set cover problem in its weighted variant. You are given a universe of elements $U = \{1, \ldots, m\}$ and a family of subsets of U called $\mathcal{S} \subseteq 2^U$. For each $S \in \mathcal{S}$, there is a weight w_S. Your task is to find a *cover* $\mathcal{C} \subseteq \mathcal{S}$ of minimum weight $\sum_{S \in \mathcal{C}} w_S$. A set $\mathcal{C}$ is a cover if for each $i \in U$ there is an $S \in \mathcal{C}$ such that $i \in S$. Alternatively, you could say $\bigcup_{S \in \mathcal{C}} S = U$.

We assume that each element of U is included in at least one $S \in \mathcal{S}$. So in other words $\mathcal{S}$ is a feasible cover. Otherwise, there might not be a feasible solution.

This problem is NP hard. We will devise an algorithm that computes an approximate solution in polynomial time. As a matter of fact, the basic algorithm description only runs in expected polynomial time.

We can state it as an integer program as follows

$$
\begin{aligned}
\text{minimize} \sum_{S \in \mathcal{S}} w_S x_S & & & \text{(minimize the overall weight)} \\
\text{subject to} \sum_{S \colon i \in S} x_S \geq 1 & & \text{for all } i \in \mathcal{U} & \quad \text{(cover every element at least once)} \\
x_S \in \{0, 1\} & & \text{for all } S \in \mathcal{S} & \quad \text{(every set is either in the set cover or not)}
\end{aligned}
$$

We can relax the problem by exchanging the constraints $x_S \in \{0, 1\}$ by $0 \leq x_S \leq 1$. (These are the only constraints requiring integrality of the solution.) We get the following LP relaxation

$$
\begin{aligned}
\text{minimize} \sum_{S \in \mathcal{S}} w_S x_S & & \\
\text{subject to} \sum_{S \colon i \in S} x_S \geq 1 & & \text{for all } i \in \mathcal{U} \\
0 \leq x_S \leq 1 & & \text{for all } S \in \mathcal{S}
\end{aligned}
$$

This LP can be solved in polynomial time. Every set cover solution $\mathcal{C}$ corresponds to a feasible solution of this LP with the objective-function value being exactly the weight of the cover. However, feasible solutions of the LP are generally fractional and might have a smaller value than the best set cover

Example 11.1. *Consider $U = \{1, 2, 3\}$, $\mathcal{S} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, $w_S = 1$ for all $S \in \mathcal{S}$. The optimal set cover solution has weight 2 because we need to take two sets. However, setting $x_S = \frac{1}{2}$ for all $S \in \mathcal{S}$ is a feasible solution to the LP relaxation.*

How can be turn a fractional solution of the LP into an integral one? This step is generally called *rounding*. One of the easiest approaches is to do it in a randomized way. Let us consider the following algorithm to derive a set $\mathcal{R}$, which is usually not a cover.

- Let x^* be an optimal solution to the LP relaxation
- Add $S \in \mathcal{S}$ to $\mathcal{R}$ with probability x_S^* independently

We have to ask two questions: What is the weight of the set $\mathcal{R}$? How likely is it that elements are covered?

Lemma 11.2. *The expected weight of $\mathcal{R}$ is $\mathbf{E}\left[\sum_{S \in \mathcal{R}} w_S\right] = \sum_{S \in \mathcal{S}} w_S x_S^*$. In particular, it is bounded by the weight of an optimal set cover solution.*

Proof. For $S \in \mathcal{S}$, let X_S be a 0/1 random variable that is 1 is $S \in \mathcal{R}$ and 0 otherwise. We now have by linearity of expectation

$$\mathbf{E}\left[\sum_{S \in \mathcal{R}} w_S\right] = \mathbf{E}\left[\sum_{S \in \mathcal{S}} w_S X_S\right] = \sum_{S \in \mathcal{S}} w_S \mathbf{E}\left[X_S\right] = \sum_{S \in \mathcal{S}} w_S x_S^* \ . \qquad \square$$

Lemma 11.3. *Each element $i \in U$ is covered by $\mathcal{R}$ with probability at least $1 - \frac{1}{e}$.*

Proof. Fix an element $i \in U$. Let $\mathcal{T}$ be the subset of $\mathcal{S}$ of sets S that contain i.

The element i is covered by $\mathcal{R}$ if and only if $\sum_{S \in \mathcal{T}} X_S \geq 1$.

As we add each set S to $\mathcal{R}$ independently, we have

$$\mathbf{Pr}\left[i \text{ is not covered by } \mathcal{R}\right] = \mathbf{Pr}\left[\bigwedge_{S \in \mathcal{T}} S \notin \mathcal{R}\right] = \prod_{S \in \mathcal{T}} \mathbf{Pr}\left[S \notin \mathcal{R}\right]$$

Now, we plug in the definition of the probabilities. We have $\mathbf{Pr}\left[S \notin \mathcal{R}\right] = 1 - x_S^*$. Furthermore, $\sum_{S \in \mathcal{T}} x_S^* \geq 1$ by the LP constraints. This gives us

$$\mathbf{Pr}\left[i \text{ is not covered by } \mathcal{R}\right] = \prod_{S \in \mathcal{T}} (1 - x_S^*) \overset{(*)}{\leq} \prod_{S \in \mathcal{T}} e^{-x_S^*} = e^{-\sum_{S \in \mathcal{T}} x_S^*} \leq \frac{1}{e} \ ,$$

where $(*)$ follows from $e^y \geq 1 + y$ for all $y \in \mathbb{R}$. $\square$

Observe that it is highly unlikely that the set $\mathcal{R}$ covers all our elements. However, each single element is covered with decent probability. Therefore, if we repeatedly compute such as set $\mathcal{R}$, we will end up with a cover quickly. This is the idea of Algorithm 1.

Algorithm 1: Set Cover via Randomized Rounding

```
let x* be an optimal solution to the LP relaxation
repeat
    for t = 1, ..., T do
        add S ∈ 𝒮 to ℛ_t with probability x*_S independently
    let 𝒞 = ⋃_{t=1}^{T} ℛ_t
until 𝒞 is a cover and ∑_{S∈𝒞} w_S ≤ 4T ∑_{S∈𝒮} w_S x*_S
```

Theorem 11.4. *For $T \geq \ln(4m)$, Algorithm 1 completes after one iteration of the* repeat*-loop with probability at least $\frac{1}{2}$. Consequently, the expected number of* repeat*-iterations is at most* 2.

Proof. Let us first bound the expected weight of $\mathcal{C}$ using Lemma 11.2

$$\mathbf{E}\left[\sum_{S \in \mathcal{C}} w_S\right] \leq \mathbf{E}\left[\sum_{t=1}^{T} \sum_{S \in \mathcal{R}_t} w_S\right] = T \sum_{S \in \mathcal{S}} w_S x_S^* \ .$$

Therefore, by Markov's inequality

$$\mathbf{Pr}\left[\sum_{S\in\mathcal{C}} w_S > 4T\sum_{S\in\mathcal{S}} w_S x_S^*\right] \leq \frac{1}{4} \enspace .$$

Next, we bound the probability that $\mathcal{C}$ is a cover. Consider an arbitrary element $i \in U$. By Lemma 11.3, the probability that it is not covered by the set $\mathcal{C}$ is at least

$$\mathbf{Pr}\left[i \text{ is not covered by } \mathcal{C}\right] = \prod_{t=1}^{T} \mathbf{Pr}\left[i \text{ is not covered by } \mathcal{R}_t\right] \leq \frac{1}{e^T} = \frac{1}{4m} \enspace .$$

By a union bound, we get

$$\mathbf{Pr}\left[\mathcal{C} \text{ is not a cover}\right] \leq \sum_{i\in U} \mathbf{Pr}\left[i \text{ is not covered by } \mathcal{C}\right] \leq m\frac{1}{4m} = \frac{1}{4} \enspace .$$

By a union bound, the *repeat* loop does not stop after one iteration is at most $\frac{1}{2}$. □

Overall, we find a feasible set cover solution that is at most an $O(\log m)$-factor worse than the optimal fractional solution in expected polynomial time.

2 Integer Multi-Commodity Flow

Next, we consider the following problem in an undirected graph $G = (V, E)$ with $m = |E|$ edges. We are given k pairs of vertices (s_i, t_i). We have to connect as many of them as possible using a path. Our selection is constrained by the fact that every edge has only capacity C, i.e., for every $e \in E$ only C paths may include e. Letting $\mathcal{P}_i$ denote the set of paths between s_i and t_i, we get the following integer program.

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{k} \sum_{P\in\mathcal{P}_i} x_{i,P} \\
\text{subject to} \quad & \sum_{i=1}^{k} \sum_{\substack{P\in\mathcal{P}_i \\ e\in P}} x_{i,P} \leq C && \text{for all } e \in E \\
& \sum_{P\in\mathcal{P}_i} x_{i,P} \leq 1 && \text{for all } i \in [k] \\
& x_{i,P} \in \{0,1\} && \text{for all } i \in [k],\ P \in \mathcal{P}_i
\end{aligned}$$

Again, we can get an LP by exchanging $x_{i,P} \in \{0,1\}$ by $0 \leq x_{i,P} \leq 1$ in the last constraint. This LP generally has a size that is exponential in the graph because the number of paths can be huge. However, it is still pretty easy to solve in polynomial time because it is a multi-commodity flow problem.

We will assume that $C \geq 12 \ln m$ and we will get within a constant factor of the LP optimum. Generally, one gets even better guarantees for larger C and it is also possible to get results for small C. In all cases, the algorithm and its analysis follow the same pattern devised here.

Algorithm 2: Integer Multi-Commodity Flow via Randomized Rounding

let x^* be an optimal solution to the LP relaxation
foreach $i \in [k]$ **do**
  choose a single $P \in \mathcal{P}_i$ by setting set $Y_{i,P} = 1$ with probability $\frac{1}{2}x^*_{i,P}$, no path is selected with probability $1 - \frac{1}{2}\sum_{P\in\mathcal{P}_i} x^*_{i,P}$
if *all* $Y_{i,P}$ *define a feasible selection of paths* **then** $Z_{i,P} = Y_{i,P}$ for all i, P
else $Z_{i,P} = 0$ for all i, P
output the path selection by $Z_{i,P}$

Theorem 11.5. *If $C \geq 12 \ln m$, the expected number of pairs that are connected by Algorithm 2 is at least* $\frac{1}{4} \sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i} x^*_{i,P}$.

Proof. Let consider a fixed $j \in [k]$ and let us condition on $Y_{j,\tilde{P}} = 1$ for some $\tilde{P} \in \mathcal{P}_i$. As all other pairs are independent, we have for each edge $e \in E$ that

$$\mathbf{E}\left[\sum_{i \neq j} \sum_{\substack{P \in \mathcal{P}_i \\ e \in P}} Y_{i,P} \;\middle|\; Y_{j,\tilde{P}} = 1 \right] = \sum_{i \neq j} \sum_{\substack{P \in \mathcal{P}_i \\ e \in P}} \frac{1}{2} x^*_{i,P} \leq \frac{1}{2} C \enspace .$$

By a Chernoff bound using $\delta = 1$ and $\mu = \frac{1}{2}C$, it follows that

$$\mathbf{Pr}\left[\sum_{i \neq j} \sum_{\substack{P \in \mathcal{P}_i \\ e \in P}} Y_{i,P} \geq C \;\middle|\; Y_{j,\tilde{P}} = 1 \right] \leq \mathbf{Pr}\left[\sum_{i \neq j} \sum_{\substack{P \in \mathcal{P}_i \\ e \in P}} Y_{i,P} \geq 2\mu \;\middle|\; Y_{j,\tilde{P}} = 1 \right]$$
$$\leq \exp\left(-\frac{\mu}{3}\right) = \exp\left(-\frac{C}{6}\right) \leq \exp\left(-\frac{12 \ln m}{6}\right) = \frac{1}{m^2} \enspace .$$

The next step is to see that one of the edge constraints is violated only if after taking out one path the remaining number of paths crossing this edge is still at least C. Therefore, we get

$$\mathbf{Pr}\left[\text{edge } e \text{ is overloaded} \;\middle|\; Y_{j,\tilde{P}} = 1\right] = \mathbf{Pr}\left[\sum_{i=1}^{m} \sum_{\substack{P \in \mathcal{P}_i \\ e \in P}} Y_{i,P} > C \;\middle|\; Y_{j,\tilde{P}} = 1 \right] \leq \frac{1}{m^2} \enspace .$$

By a union bound, it follows that

$$\mathbf{Pr}\left[Z_{j,\tilde{P}} = 0 \;\middle|\; Y_{j,\tilde{P}} = 1\right] = \mathbf{Pr}\left[\text{the path selection is not feasible} \;\middle|\; Y_{j,\tilde{P}} = 1\right] \leq \frac{1}{m} \leq \frac{1}{2} \enspace .$$

So

$$\mathbf{Pr}\left[Z_{j,\tilde{P}} = 1\right] \geq \frac{1}{2} \mathbf{Pr}\left[Y_{j,\tilde{P}} = 1\right]$$

Finally, because $\mathbf{E}\left[Y_{i,P}\right] = \frac{1}{2} x^*_{i,P}$, we have

$$\mathbf{E}\left[\sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i} Z_{i,P} \right] \geq \frac{1}{2} \sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i} \mathbf{E}\left[Y_{i,P}\right] = \frac{1}{4} \sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i} x^*_{i,P}$$

□

3 Further Reading

- Chapter 14 in Vazirani, Approximation Algorithms
- Chapters 5 and 12 in Williamson/Shmoys, The Design of Approximation Algorithms, `http://www.designofapproxalgs.com`