Lecture 6: Randomized Rounding

6th Feb, 2015

In this lecture, we look at randomized approximation algorithms. Since these algorithms are randomized, the solutions of these algorithms will be random variables.

Definition 1. *For a minimization problem, an α-approximate randomized algorithm returns a feasible solution S of* **expected** *cost* $\mathbf{Exp}[c(S)] \leq \alpha OPT$. *For a maximization problem, an α-approximate randomized algorithm returns a feasible solution S of* **expected** *value* $\mathbf{Exp}[v(S)] \geq OPT/\alpha$.

Often, a slightly weaker definition is used.

Definition 2. *For a minimization problem, an α-approximate randomized algorithm returns a solution S such that $\mathbf{Pr}[S \text{ is feasible} \wedge c(S) \leq \alpha OPT] > \delta$, where δ could be as small as $\frac{1}{\mathtt{poly}(n)}$.*

For a minimization problem, an α-approximate randomized algorithm returns a solution S such that $\mathbf{Pr}[S \text{ is feasible} \wedge c(S) \geq OPT/\alpha] > \delta$, where δ could be as small as $\frac{1}{\mathtt{poly}(n)}$.

Note that if an algorithm with parameter δ existed, then one can boost it to get an algorithm for parameter $(1-\varepsilon)$ as well. We will explore this connection more in the exercises.

Randomized rounding for Set cover. In randomized rounding, one interprets the fractional value returned by the LP as probabilities with which the elements should be picked in the solution. Then one has to take care of various consistency issues. Let us illustrate with the set cover example.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} c(S_i)x_i & \\
\text{subject to} \quad & \sum_{i:j\in S_i} x_i \geq 1 & \forall j \in U \\
& 0 \leq x_i \leq 1 & \forall i = 1 \ldots m
\end{aligned}
$$

Here is the randomized algorithm.

1. Sample each set i **independently** with probability $p_i = \min(1, \ln n \cdot x_i)$.
2. For each element j uncovered in step 1, pick the set $S(j) = \arg\min_{S_i:j\in S_i} c(S)$.

Clearly, the algorithm returns a feasible solution. We now argue that the expected cost of the solution is small. First an easy observation

Claim 1. *For any element j, we have $c(S(j)) \le LP$.*

Proof. Since $\sum_{i:j\in S_i} x_i \le 1$, we have

$$LP \ge \sum_{i:j\in S_i} c(S_i)x_i \ge c(S(j)) \sum_{i:j\in S_i} x_j \ge c(S(j)).$$

$\square$

The algorithms cost $ALG = ALG_1 + ALG_2$ where ALG_i is the cost of the sets picked in step i. Note that both of these are random variables and can be expressed thus.

$$ALG_1 = \sum_{i=1}^{m} c(S_i)X_i$$

where $X_i \in \{0,1\}$ is the **indicator variable** of whether set S_i is sampled in Step 1 or not. We know that $\mathbf{Exp}[X_i] = \mathbf{Pr}[X_i = 1] = p_i = \min(1, \ln x_i)$. And so,

$$\mathbf{Exp}[ALG_1] \le \ln n LP.$$

Now observe that

$$ALG_2 = \sum_{j\in U} c(S(j))Y_j$$

where Y_j is the indicator variable of whether j is left uncovered or not in the first step. $\mathbf{Exp}[Y_j] = \mathbf{Pr}[Y_j = 1] = \prod_{i:j\in S_i}(1 - p_i)$. Now if $p_i = 1$ for any i with $j \in S_i$, then $\mathbf{Exp}[Y_j] = 0$. Therefore, we get

$$\mathbf{Exp}[Y_j] \le \exp(-\sum_{i:j\in S_i} p_i) = \exp(-\ln n \sum_{i:j\in S_i} x_i) \le 1/n.$$

Therefore, we get (using the claim above)

$$\mathbf{Exp}[ALG_2] \le LP$$

Together, we get the theorem

Theorem 1. *The algorithm described above is a $(1 + \ln n)$-approximate randomized algorithm.*

Maximum Independent Set (MIS) problem. In a graph, an independent set is a subset S of vertices such that there is no edge (u, v) with both end points in S. The MIS problem asks us to pick an independent set of the largest size.

Here is an LP relaxation for the problem.

$$\begin{aligned} \max \quad & \sum_{v\in V} x_v & \\ \text{subject to} \quad & x_u + x_v \le 1 & \forall (u, v) \in E \\ & 0 \le x_v \le 1 & \forall v \in V \end{aligned}$$

We now describe an algorithm which is a $O(\sqrt{m})$-approximation, where m is the number of edges.

1. If $LP < 2\sqrt{m}$, return any singleton vertex and exit.
2. Sample independently vertex v with probability $p_v = x_v/\sqrt{m}$ to get a set S.
3. For each *violated* edge $e = (u, v)$ with both end points in S, **delete** both the points from S. (It would've sufficed to delete any one, but as we show this overzealousness doesn't hurt.)

Once again, it is clear that the solution returned is an independent set. Also note that if $LP < 2\sqrt{m}$, a singleton vertex is a $2\sqrt{m}$-approximate solution. So we may assume otherwise. Let ALG be the number of vertices in this returned solution. As before, note

$$\mathbf{Exp}[ALG] = \sum_{v \in V} X_v - \sum_{(u,v)\in E} 2Z_{uv}$$

where X_v is the indicator variable for the event that v is picked in S and Z_{uv} is the indicator variable that both $u, v \in S$. Now, $\mathbf{Exp}[X_v] = x_v/\sqrt{m}$ and so the first term is precisely $LP/\sqrt{m}$. Also note

$$\mathbf{Pr}[Z_{uv} = 1] = p_u p_v = \frac{x_u x_v}{m} \le \frac{x_u^2 + x_v^2}{2m} \le \frac{x_u + x_v}{2m} \le 1/2m.$$

The first inequality is AM-GM, the second follows since $x_u \le 1$ for all u, and the last is the LP constraint. Therefore, the second term in the expression for $\mathbf{Exp}[ALG]$ is at most 1. Now since $LP \ge 2\sqrt{m}$, we get $\mathbf{Exp}[ALG] \ge \frac{LP}{2\sqrt{m}}$.

Theorem 2. *The above theorem is a $2\sqrt{m}$-approximation algorithm.*

Minimizing congestion in routing The input is a directed graph with k pairs of vertices denoted as $(s_1, t_1), \ldots, (s_k, t_k)$. The goal is to pick a collection of k paths from s_i to t_i for each i, such that the **congestion** of the solution is minimized. The congestion of a collection of paths is the maximum over all edges, the number of paths it appears in.

The LP is the following

$$\begin{aligned}
\min \quad & c & \\
\text{subject to} \quad & \sum_{p \in P : e \in p} f_p \le c & \forall e \in E \\
& \sum_{p \in P_i} f_p = 1 & \forall i = 1 \ldots m \\
& f_p \ge 0 &
\end{aligned}$$

P_i is the set of (s_i, t_i)-paths.

The above LP has exponentially many vertices. Observe, however, a bfs would have only polynomially many f_p's > 0. In the next lecture, we will discuss a way to obtain such a sparse solution in polynomial time. In the exercises, we will discuss an alternate route to obtain a solution to the above LP. For the time being, assume we have such a solution. Throwing out all p such that $f_p = 0$, we assume $|P_i|$ is polynomially bounded. The randomized rounding algorithm is the following.

- Independently, for each i, pick **one** path $p \in P_i$ with probability f_p by rolling a $|P_i|$-sided dice (recall, $\sum_{p \in P_i} f_p = 1$.)

Obviously, it is a feasible solution. We need to argue about congestion. Fix an edge e. For each i, there is at most one path $p \in P_i$ which contains it. For an i for which such a path exists, let X_i be the rv indicating whether that path $p \in P_i$ is chosen. The congestion on edge e is precisely $\sum_i X_i$. The expected congestion on edge e is $\mu_e = \mathbf{Exp}[\sum_i X_i] = \sum_{p:e \in P} f_p \leq c$. Thus in expectation, each edge has congestion what the LP suggests.

To argue about high probability, we need to apply the Chernoff bound (Fact5) which is stated later. Therefore, for any edge e, the probability that the total congestion exceeds $(1+\delta)c$ is at most

$$\exp\left(-c\left((1+\delta)\ln(1+\delta) - \delta\right)\right)$$

Now if $(1+\delta) = 6\ln n / \ln\ln n$, using the fact that $c \geq 1$ one can verify that this expression is at most $1/n^3$. Since the number of edges is at most n^2, we get the desired result via an union bound.

GAP revisited

Recall the GAP problem from two classes ago. We have m machines and n jobs; each job j has a profit of p_{ij} if allocated on machine i and puts a load w_{ij} on it. Each machine has an upper bound of B_i on the total load assigned on it. The goal is to find an assignment which maximizes profit. We saw a 2-approximation via a certain natural LP, and in the exercises we ask to prove that the integrality gap of the LP is also at most 1/2. Now we show a different LP, called the **configuration LP** in the literature and use it to give a better approximation algorithm.

Configuration LP This LP will have *lots* of variables, but not too many constraints. In this class, we cannot show how to solve it in polynomial time and this will have to wait till next lecture. So we will just believe this LP can be solved.

For every machine i, let $\mathcal{F}_i$ denote the collection of subsets of jobs S such that $\sum_{j \in S} w_{ij} \leq B_i$. That is, $\mathcal{F}_i$ is the collection of plausible sets of jobs which can be feasibly assigned to machine i. These feasible sets are called *configurations*. In the configuration LP, there is a variable $x_{i,S}$ for every machine i and every set $S \in \mathcal{F}_i$. There are two types of constraints: one which states that every machine gets exactly one configuration, and the second which states that each job participates in at most one configuration. Below $p_i(S)$ is the shorthand for $\sum_{j \in S} p_{ij}$. Mathematically, the LP is the following.

$$\begin{array}{lll}
\max & \displaystyle\sum_{i=1}^{m}\sum_{S \in \mathcal{F}_i} p_i(S) x_{i,S} & \\
\text{subject to} & \displaystyle\sum_{S \in \mathcal{F}_i} x_{i,S} = 1 & \forall i \in \{1,\ldots,m\} \\
& \displaystyle\sum_{i=1}^{m}\sum_{S \in \mathcal{F}_i : j \in S} x_{i,S} \leq 1 & \forall j \in \{1,\ldots,n\} \\
& 0 \leq x_{i,S} & \forall i = 1\ldots m, \forall S \in \mathcal{F}_i
\end{array}$$

As mentioned above, the number of variables in the LP are exponentially many. However, note that the number of non-trivial constraints are only $(n+m)$. If you recall last time's lecture, this implies that in any basic feasible solution, and in particular in the optimum bfs, the number of strictly positive $x_{i,S}$ is at most $(n+m)$. It is probably not clear how to find such a sparse solution – we will come to this in the next lecture. For the time being assume we can get an optimum solution with $x_{i,S} > 0$ for a certain polynomially many values. We now use this to give a $e/e-1$ approximation.

The algorithm is the following.

- Each agent i, **independently**, is tentatively assigned jobs from one set $S \in \mathcal{F}_i$ with probability $x_{i,S}$ (recall, $\sum_{S\in\mathcal{F}_i} x_{i,S} = 1$).
- A job j may be tentatively assigned to many machines. It is assigned to the machine i with the highest p_{ij} among the machines which it tentatively is assigned to.

Theorem 3. *The above algorithm is an $\frac{e}{e-1}$-approximation algorithm.*

Proof. For every job j and machine i, define $y_{ij} := \sum_{S\in\mathcal{F}_i:j\in S} x_{i,S}$. The probability job j is tentatively assigned to machine i is $\sum_{S\in\mathcal{F}_i:j\in S} x_{i,S} = y_{ij}$. The constraints of the LP gives us $\sum_i y_{ij} \leq 1$ for all jobs j. Finally, the objective value of the LP is

$$\sum_i \sum_{S\in\mathcal{F}_i} \left(x_{i,S} \sum_{j\in S} p_{ij} \right) = \sum_j \sum_i p_{ij} \sum_{S\in\mathcal{F}_i:j\in S} x_{i,S} = \sum_{i,j} p_{ij} y_{ij}$$

Fix a job j. Rename the machines such that $p_{1j} \geq p_{2j} \geq \cdots \geq p_{mj}$. Let us calculate the probability that machine i gets assigned job j. For this to happen, none of the machines 1 to $(i-1)$ should tentatively ask for job j and machine i should ask for job j. The probability that this occurs is precisely $y_{ij}(1-y_{1j})(1-y_{2j})\cdots(1-y_{i-1,j})$. Therefore, the total profit that is obtained from job j is precisely

$$p_{1j}y_{1j} + p_{2j}y_{2j}(1-y_{1j}) + \cdots + p_{nj}y_{nj}\prod_{i<n}(1-y_{ij}) \tag{1}$$

Claim 2. *The expression in (1) is at least $(1-1/e)\cdot\sum_{i=1}^n p_{ij}y_{ij}$.*

Proof. In class, we did the verification for $p_{ij} = 1$ case which is nothing but the equality Note that

$$\begin{aligned} y_{1j} + y_{2j}(1-y_{1j}) + \cdots + y_{nj}\prod_{i<n}(1-y_{ij}) &= 1 - \prod_{i\leq n}(1-y_{ij}) && (2)\\ &\geq 1 - \exp(-\sum_{i=1}^n y_{ij}) && (3)\\ &\geq (1-1/e)\sum_{i=1}^n y_{ij} && (4) \end{aligned}$$

(3) follows since $1-x \leq e^{-x}$, and (4) follows since $1-e^{-t} \geq (1-1/e)t$ for all $0 \leq t \leq 1$. To see the latter, consider $f(t) = 1-e^{-t}-(1-1/e)t$ and note that $f'(t) = e^{-t} - 1 + 1/e$ and so f is increasing upto some point and then decreases from there on. So it suffices to check the inequality at the extremes $0, 1$ and one can check it is true.

Now note that we didn't use n anywhere in the above inequality, that is, in general the following is true.

$$y_{1j} + y_{2j}(1 - y_{1j}) + \cdots + y_{nj} \prod_{i<k} (1 - y_{ij}) \geq (1 - 1/e) \sum_{i=1}^{k} y_{ij} \tag{5}$$

Now multiply the kth inequality by $(p_{kj} - p_{k+1,j})$ (with $p_{n+1,j}$ defined to be 0), and add them all up. This implies the claim. To see this, note that the coefficient of y_{ij} in the RHS is precisely $(p_{ij} - p_{i+1,j}) + (p_{i+1,j} - p_{i+2,j}) + \cdots + (p_{n-1,j} - p_{nj}) + p_{nj}$ which telescopes to p_{ij}. The coefficient of $y_{ij} \prod_{t<i}(1 - y_{tj})$ is also, for the same reason, p_{ij}. □

The claim immediately implies the theorem. □

Some Probabilistic Facts.

Fact 1 (Linearity of Expectation). *Let* $X_1, \ldots, X_n$ *be random variables and let* $X := \sum_{i=1}^{n} X_i$. *Then,* $\mathbf{Exp}[X] = \sum_{i=1}^{n} \mathbf{Exp}[X_i]$.

Fact 2 (The Union Bound). *Given* n *events* $\mathcal{E}_1, \ldots, \mathcal{E}_n$*, the probability that one of them occurs is at most the sum of their probabilities.*

$$\mathbf{Pr}[\mathcal{E}_1 \vee \mathcal{E}_2 \vee \cdots \vee \mathcal{E}_n] \leq \sum_{i=1}^{n} \mathbf{Pr}[\mathcal{E}_i]$$

Fact 3 (Chebyshev's Inequality). *For any random variable* X*,*

$$\mathbf{Pr}[|X - \mathbf{Exp}[X]| \geq t] \leq \frac{\mathbf{Var}[X]}{t^2}$$

Fact 4 (Markov's Inequality). *Let* X *be a* non-negative *random variable. Then*

$$\mathbf{Pr}[X \geq t] \leq \frac{\mathbf{Exp}[X]}{t}$$

Fact 5 (Chernoff Bound). $X_1, \cdots, X_n$ *be* n independent $\{0,1\}$ *random variables with* $\mathbf{Pr}[X_i = 1] = p_i$. *Let* $X := \sum_{i=1}^{n} X_i$ *and let* $\mu = \sum_{i=1}^{n} p_i = \mathbf{Exp}[X]$. *Then for any* $L \leq \mu \leq U$*,*

$$\mathbf{Pr}[X > (1+\delta)U] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{U}$$

$$\mathbf{Pr}[X < (1-\delta)L] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{L}$$

Useful versions of the above:

- *For any* $\delta > 0$*,*

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

- *For* $t > 0$*,*

$$\mathbf{Pr}[|X - \mu| > t] \leq 2\exp(-2t^2/n)$$

- *For* $t > 4\mu$*,*

$$\mathbf{Pr}[X > t] \leq 2^{-t}$$