22C:253 Lecture 8

Scribe: Ganesh Venkataraman

October 2, 2002

Randomized Rounding

The main idea behind randomized rounding is the following

1 Solve the LP-relaxation in polynomial time

2 Interpret the solution obtained as a probability vector

This idea was proposed in the 1980s by Raghavan and Thompson to solve a problem in VLSI Design Automation. We present two examples of randomized rounding: Set Cover and MAX-SAT.

Set Cover

We now discuss a randomized algorithm that gives a *O(log n)* approximation algorithm for the set cover problem. Recall that the LP-relaxation for the set cover problem can be stated as follows:

$$\min \sum_{i}^{k} cost(S_i) * x_i$$

subject to the constraints

$$\begin{aligned} \sum_{i:j \epsilon S_i} x_i &\geq 1 \text{ for each } j \in U \\ x_i &\geq 0 \text{ for each } i = 1, 2, \ldots, k \end{aligned}$$

Now carry out the following two steps
Step 1: Solve the LP-relaxation and let $x = (x_1, x_2, \ldots, x_k)$ represent the solution.
Step 2: Interpret each x_i as a probability and perform the following:

for $i = 1, 2, \ldots, k$ include S_i in the solution with probability x_i

Another way of stating the above is

for $i = 1, 2, \ldots, k$ do set $y_i = 1$ with probability x_i

Here $y_i's$ represent the integral solutions for the set cover ILP.
Some remarks about the above chosen method:

- The elements in S_i are chosen independently

- The resulting solution need not be feasible

Now we proceed to analyze steps 1 and 2. Let $Y = \sum_i^k cost(S_i) * y_i$. Note that the $y_i's$ are all binary random variables and

$$\text{Prob}[y_i = 1] = x_i \qquad \text{and} \qquad \text{Prob}[y_i = 0] = 1 - x_i.$$

Therefore $E[y_i] = x_i$. Furthermore,

$$E[Y] = E[\sum_i^k cost(S_i) * y_i] = \sum_i^k cost(S_i) * E[y_i] = \sum_i^k cost(S_i) * x_i$$

The RHS represents the optimal solution of the LP-relaxation and so $E[Y] \leq OPT$. where OPT represents the optimal solution of the ILP. The probability that $j \in U$ is NOT covered is $\prod_{i:j \epsilon S_i}(1 - x_i)$. Therefore the probability P_j that an element $j \in U$ is covered is given by:

$$P_j = 1 - \prod_{i:j \epsilon S_i} (1 - x_i)$$

It is easy to see that the quantity in the RHS is minimized when $x_i = 1/f_j$ where f_j is the frequency of j for all $i : j \in S_i$. Hence,

$$P_j \geq 1 - \prod_{i:j \epsilon S_i} (1 - 1/f_j) = 1 - (1 - 1/f_j)^{f_j}$$

Recall that for all real x, $e^x \geq (1 + x)$ and hence

$$e^{-1/f_j} \geq (1 - 1/f_j) \Rightarrow e^{-1} \geq (1 - 1/f_j)^{f_j}.$$

Hence,

$$P_j \geq (1 - 1/e).$$

Therefore the probability that each element is covered is at least a constant. So the expected number of elements covered is at least $(1 - 1/e)^n$. This fact motivates step 3 as follows:
Step 3: Repeat step 2 $c \log n$ times for some positive c to be fixed later.
Note that each repetition is independent of all other repetitions. Let C' be the collection of subsets thus obtained. The probability of $j \in U$ is not covered by a subset in C' is atmost $(1/e)^{clogn}$. Now pick c such that:

$$(1/e)^{clogn} \geq 1/4n \Rightarrow e^c \leq 4$$

Therefore the probability a $j \in U$ is not covered by a subset in C' is atmost $1/4n$. Let $P_j^{C'}$ represent the probability that there exists at least one $j \in U$ that is not covered.

$$P_j^{C'} \leq 1/4 \tag{1}$$

By using the union bound:

$$\text{Prob}[x_1 \vee x_2 \vee \cdots \vee x_n] \leq \sum_i^n \text{Prob}[x_i] \tag{2}$$

We get

$$E[cost(C^{'})] \leq clogn * OPT \tag{3}$$

Notice that (2) is not dependent on the fact that $x_i^{\prime}s$ are all independent. Recall Markov's inequality:

$$\text{Prob}[X \geq k] \leq \frac{E[X]}{k} \tag{4}$$

By using (3) and (4) we get

$$\text{Prob}[cost(C^{'}) \geq 4clogn * OPT] \leq 1/4 \tag{5}$$

We need two things:

- The cost($C^{'}$) should be less than 4clogn*OPT
- $C^{'}$ should be feasible

Let the probability that the above two things happen be $P_{desired}$. Then from (1) and (5) it is easy to see that:

$$P_{desired} \geq 1/2 \tag{6}$$

Now simply run steps 2 and 3 until the above conditions are met. By (6) we expect to repeat steps 2 and 3 not more than 2 times.
Note: OPT in (5) actually refers to the optimal solution of the ILP which is incalculable. However we could use the optimal solution of the LP-relaxation for practical purposes.

MAX-SAT

Input: A boolean formula f in CNF defined on boolean variables $x_1, x_2, \ldots, x_n$ and associated with each clause c of f a weight w_c .
Output: A truth assignment to $x_1, x_2, \ldots, x_n$ that maximizes the weight of satisfied clauses.
We now discuss a simple randomized algorithm that give a 1/2 factor approximation (We shall later use this to produce a 3/4 factor approximation scheme)
for each $i = 1, 2, \ldots, n$ do

Set $x_i = TRUE$ with probability 1/2 (independently)

Let W be the sum of the weights of satisfied edges:

$$E[W] = E[\sum_{c \epsilon C} W_c]$$

where

$$W_c = \begin{cases} 0 & \text{if c is not satisfied} \\ w_c & \text{otherwise} \end{cases}$$

$$E[W_c] = w_c(1 - 1/2^k)$$

where k represents the number of literals in clause c. Since $k \geq 1$, $(1 - 1/2^k) \geq 1/2$. Hence

$$E[W_c] \geq w_c/2$$

Therefore

$$E[W] \geq \sum_{c \in C} \frac{w_c}{2} \geq \frac{OPT}{2}$$

since

$$OPT \leq \sum_{c \in C} w_c$$