22C:253 Lecture 2

Scribe: Chris Choi Chang-Hui

September 4, 2002

In the last lecture we studied a factor-2 approximation algorithm for Cardinality Vertex Cover (CVC). There are three questions one can ask about that algorithm and its proof.

1. Is our analysis tight? In other words, is it the case that our algorithm, for some instance, produces a vertex cover whose size is twice the size of the optimal.

   The answer is yes. Consider the complete bipartite graph $K_{n,n}$. The algorithm produces a vertex cover of size $2n$, while $OPT = n$.

2. Is there some other algorithm that uses the same lower bound (maximal matching), but produces a factor-f approximation for some $f < 2$?

   The answer is no. Consider the complete graph K_n, for odd n. In this case, any maximal matching has size $(n-1)/2$ and $OPT = n - 1$.

3. Is there some other algorithm that produces a better than 2 approximation factor?

   No one knows. This is one of the most tantalizing open questions in this area.

Set Cover. Our next example is a problem for which we will present a factor-$O(\log n)$ greedy approximation algorithm.
SET COVER (SC)
Input: Given a universe U of n elements and a collection $C = \{S_1, S_2, ..., S_k\}$ of subsets of U, and an assignment $c : C \to Q^+$ of costs to the subsets in C.
Output: A subcollection C' with the minimum cost that covers the universe.

A subcollection C' *covers* U if $\bigcup_{S \in C'} = U$. The *cost* of a subcollection is simply the sum of the costs of the subsets in it.

Example. Suppose $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $C = \{\{1, 4\}, \{2, 4, 7\}, \{3, 7, 8\}, \{5, 6, 8\}, \{357\}\}$, and the costs of the sets in the collection are $3, 4, 2, 1, 7$ respectively. A subcollection C' that covers U is $\{\{1, 4\}, \{2, 4, 7\}, \{5, 6, 8\}, \{3, 5, 7\}\}$ and it has cost $3 + 4 + 1 + 7 = 15$.

There is a simple "greedy algorithm" for the set cover problem. At each step pick a subset that has the smallest cost per new element that it covers. This is also the same as saying "pick a subset that covers the most number of new elements per unit cost."

1. $A = \emptyset$
2. **while** $(U \neq \emptyset)$ **do**
3. Pick a set S_i that minimizes $\frac{\text{Cost}(S_i)}{|S_i \cup U|}$

4. $A = A \cup \{Si\}$
5. $U = U - S_i$
6. output A

How do we show that this algorithm achieves a factor-$O(\log n)$ approximation? Let $e_1, e_2, \ldots, e_n$ be the order in which elements are covered by the above algorithm, with ties broken arbitrarily. Each element $e_k \in U$ is first covered by some set S_i in Step 3. Define the *price* of e_k, denoted $price(e_k)$ as $\frac{\text{Cost}(S_i)}{|S_i \cup U|}$. We can prove the following lemma.

Lemma 1 *For each k, $1 \leq k \leq n$,*

$$price(e_k) \leq \frac{OPT}{n-k+1}.$$

Lemma 2

$$cost(A) \leq H_n \cdot OPT,$$

where H_n is the Harmonic number $1 + 1/2 + 1/3 + \cdots + 1/n$.

Proof:

$$\text{cost}(A) = \sum_{k=1}^{n} price(e_k) \leq OPT \sum_{k=1}^{n} 1/(n-k+1) = H_n \cdot OPT.$$

□

Proof: (**Lemma 1**) Let

$$C = \{S_{i_1}, S_{i_2}, S_{i_3}, \ldots, S_{i_t}\}$$

be a minimal subcollection of OPT that covers $e_k, e_{k+1}, \ldots, e_n$. Let $E = \{e_1, e_2, \ldots, e_{k-1}\}$ and let $E' = \{e_k, e_{k+1}, \ldots, e_n\}$. For $e_\ell \in E'$ let S_{i_j} be the first set that contains e_ℓ. Define

$$f(e_\ell) = \frac{\text{Cost}(S_{i_j})}{|S_{i_j} - E - S_{i_1} - S_{i_2} - \cdots - S_{i_{j-1}}|}.$$

Note that

$$\sum_{\ell=k}^{n} f(e_\ell) = \text{cost}(C) \leq OPT$$

and so there is some $e_\ell \in E'$ such that $f(e_\ell) \leq OPT/(n-k+1)$.

Just before the greedy algorithm covers e_k, none of the sets $S_{i_1}, S_{i_2}, \ldots, S_{i_t}$ have been picked by the algorithm, because elements in E' are still uncovered. If at this stage S_{i_j} were chosen, it would assign to element e_ℓ the price

$$price(e_\ell) = \frac{\text{Cost}(S_{i_j})}{|S_{i_j} - E|} \leq f(e_\ell) \leq OPT/(n-k+1).$$

Since the greedy algorithms choses to pick a set that minimizes "price" and e_k is the next element covered, $price(e_k) \leq price(e_\ell) \leq \frac{OPT}{/}(n-k+1)$. □

Question : Is this analysis tight? Yes. Consider the following example. Let $U = \{1, 2, \ldots, n\}$. Let $C = \{S_1, S_2, \ldots, S_n, S_{n+1}\}$ be a given collection of subsets of U such that $S_i = \{i\}$ for $1 \leq i \leq n$ and $S_{n+1} = \{1, 2, \ldots, n\}$. Let $\text{cost}(S_i) = 1/i$ for $1 \leq i \leq n$ and let $\text{cost}(S_{n+1}) = 1 + \epsilon$. The greedy algorithm picks $\{S_1, S_2, \ldots, S_n\}$ for a cost of H_n, while $OPT = (1 + \epsilon)$.