CS 170 Spring 2017 − Discussion 6

Raymond Chan

Set Cover

In the set cover problem, we have a set B of n elements and subsets $S_1, \ldots, S_m \subseteq B$. The goal is to find a selection fo S_i whose union is B while minimizing the number of sets picked. Since finding the optimal solution takes a very inefficient running time, we want to have a polynomial time algorithm that approximates the optimal solution. Let's consider the following optimal algorithm.

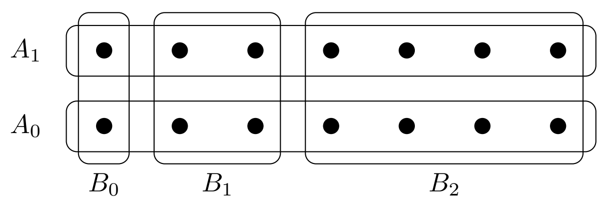
procedure GREEDY-SET-COVER($B, S_1, \ldots, S_m$)
  while there are still uncovered elements **do**
    Pick S_i with the largest number of uncovered elements.

If we have n elements and k optimal sets to cover all elements, we want to show that the number of sets from GREEDY is at most $k \ln n$.
Let $U(S_t)$ be the number of uncovered elements from set S_t at the t-th iteration. We know that

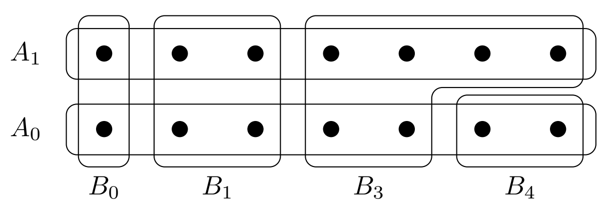
$$U(S_1) \geq \frac{n}{k}$$

By GREEDY, $U(S_i) \leq U(S_1)$, $i \in \{2 \ldots\}$
If $U(S_1) = \frac{n}{k}$, then all k optimal sets have $\frac{n}{k}$ uncovered elements.
Suppose for the sake of contradiction that $U(S_1) < \frac{n}{k}$, then for $i \in \{2 \ldots k\}$, $U(S_i) < U(S_1) < \frac{n}{k}$. If so, we cannot possibly cover all n elements with k sets. If we can, then there must be a set whose number of uncovered elements is $> \frac{n}{k}$ and we would have picked this instead of S_1.

Let's look at the following example.



We see that the optimal solution is to pick A_0 and A_1 as they both cover 7 elements each. However, GREEDY will choose B_2, B_1, B_0 in that order. At the beginning, we have 14 elements and B_2 has 7 uncovered elements whereas A_0 and A_1 only have 7 uncovered elements. At the second iteration, B_1 has 4 uncovered elements whereas A_0 and A_1 have 3 each. Finally B_0 have 2 uncovered elements whereas A_0 and A_1 have 1 each.



Now suppose we split B_3 into B_4. $U(B_3) = 6 < \frac{14}{2} = 7$. But we wouldn't have picked it as both A_0 and A_1 have 7 uncovered elements and we would have pick one of those in the first iteration.

Currently we have $U(S_1) \geq \frac{n}{k}$. Let's define n_t as the number of remaining elements after t iterations. $n_0 = n$.

$$n_1 = n_0 - U(S_1) \leq n_0 - \frac{n_0}{k} = n_0\big(1 - \frac{1}{k}\big)$$

With n_1 remaining elements, we could have $k-1$ optimal elements that cover them all. Or all of them need still be covered by k optimal elemenWith n_1 remaining elements, we could have $k-1$ optimal elements that cover them all. Or all of them need still be covered by k optimal elements.

$$\begin{aligned} n_2 = n_1 - U(S_2) \leq n_1 - \frac{n_1}{k-1} \leq n_1 - \frac{n_1}{k} = n_1\big(1 - \frac{1}{k}\big) \\ \leq n_0\big(1 - \frac{1}{k}\big)\big(1 - \frac{1}{k}\big) = n_0\big(1 - \frac{1}{k}\big)^2 \end{aligned}$$

So after some t iterations, we have

$$n_{t+1} \leq n_t - \frac{n_t}{k} \leq n_0\big(1 - \frac{1}{k}\big)^{t+1}$$

Generalizing this, we have

$$n_t \leq n\big(1 - \frac{1}{k}\big)^t$$

Now we want to find how many iterations or sets does it take for us to reach less than 1 remaining number of elements.

$$n_t \leq n\big(1 - \frac{1}{k}\big)^t < 1$$

We want to make use of the following inequality

$$1 - x \leq e^{-x} \quad \forall x \quad \text{equalty iff } x = 1$$

$$\begin{aligned} ne^{-\frac{t}{k}} &< 1 \\ e^{-\frac{t}{k}} &< \frac{1}{n} \\ \ln e^{-\frac{t}{k}} &< \ln 1 - \ln n \\ -\frac{t}{k} &< -\ln n \\ \frac{t}{k} &> \ln n \\ t &> k \ln n \end{aligned}$$

Since is takes more than $k \ln n$ iterations, lets find out by how much more. Set $t = k \ln n$

$$ne^{-\frac{t}{k}} = ne^{-\frac{k \ln n}{k}} = ne^{-\ln n} = nn^{-\ln e} = n\frac{1}{n} = 1$$

Thus we need 1 more iteration to get no more remaining elements.
GREEDY will need $k \ln n + 1$ sets, which is $O(k \ln n)$.