CSE 632: Analysis of Algorithms II	**Fall 2017**

Lecture 3 (09/06/2017): Maximum Coverage, Set Cover

Lecturer: Shi Li *Scribe: Zhenggang Xue*

3.1 Definitions of Maximum Coverage and Set Cover

Let $[n]$ represents set $\{1, 2, 3, \cdots, n\}$.

Definition 3.1 (Maximum Coverage) *Given $S_1, S_2, \cdots, S_m \subseteq [n]$, $\bigcup_{i=1}^m S_i = [n]$ and an integer $k \geq 1$, the goal of maximum coverage is to choose a set $I \subseteq [m]$, so as to maximize $|\bigcup_{i \in I} S_i|$.*

That is, we select a sub-collection of S whose size is at most k such that the maximum number of elements are covered.

Definition 3.2 (Set Cover) *Given $S_1, S_2, \cdots, S_m \subseteq [n]$ and $\bigcup_{i=1}^m S_i = [n]$, the goal of set cover problem is to choose a minimum size $I \subseteq [m]$, subject to $\bigcup_{i \in I} = [n]$.*

That is, we select the smallest sub-collection of S whose union covers all elements.

3.2 Greedy Algorithms

Algorithm 1 Greedy Algorithms for Maximum Coverage

1: $I \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to k **do**
3: let $i^* \in [m]$, with maximum $|S_{i^*} \setminus \bigcup_{i \in I} S_i|$
4: $I \leftarrow I + \{i^*\}$
5: **return** I

At first, let's define some notations that will be used in the below analysis. Let *opt* be the value of optimum solution and y_j be the number of elements our algorithm covered after iteration j. So we have $y_j = |\bigcup_{i \in I_j} S_i|$, where I_j is the set after iteration j.

Claim 3.3

$$y_i - y_{i-1} \geq \frac{opt - y_{i-1}}{k}$$

Proof: let $X_{j-1} = \bigcup_{i \in I_{j-1}} S_i$ and I^* be the optimum solution. We will have $|\bigcup_{i \in I^*} S_i \setminus X_{j-1}|$ elements that are covered in optimum solution but not covered by our greedy algorithm solution after iteration $j - 1$. It is trivial that the number of those elements is at least $(opt - y_{j-1})$ and this lower bound happens when our greedy algorithm picks sets whose elements are all covered in optimum solution. So we have $|\bigcup_{i \in I^*} S_i \setminus X_{j-1}| \geq opt - y_{i-1}$. Since the optimum solution uses

k sets to cover opt elements, some set must cover at least $\frac{1}{k}$ fraction of the at least $(opt - y_{j-1})$ remaining uncovered elements. This means

$$\exists i \in I^*, s.t. |S_i \setminus X_{j-1}| \geq \frac{opt - y_{j-1}}{k}$$

At each iteration, our greedy algorithm always picks the set that covers the maximum number of uncovered elements. So for the i^* we select at iteration j,

$$|S_{i^*} \setminus X_{j-1}| \geq \frac{opt - y_{j-1}}{k}$$

so we get

$$y_i - y_{i-1} \geq \frac{opt - y_{i-1}}{k}$$

∎

This claim means that at each iteration, the new added uncovered elements by our algorithm will be at least $\frac{1}{k}$ of the elements which are covered by optimum solution but not in our set after last iteration.

Lemma 3.4

$$opt - y_j \leq (1 - \frac{1}{k})(opt - y_{j-1})$$

Proof: Consider

$$\begin{aligned} opt - y_j &= opt - y_{j-1} - (y_j - y_{j-1}) \\ &\leq opt - y_{j-1} - \frac{opt - y_{j-1}}{k} \\ &= (1 - \frac{1}{k})(opt - y_{j-1}) \end{aligned}$$

∎

This lemma ensures that each iteration will decrease the gap between optimum and greedy result by a factor of $(1 - \frac{1}{k})$.

Theorem 3.5 *Greedy algorithm is a $(1 - \frac{1}{e})$-approximation for Maximum Coverage.*

Proof:

$$\begin{aligned} opt - y_k &\leq (1 - \frac{1}{k})^k (opt - y_0) \\ &= (1 - \frac{1}{k})^k \cdot opt \\ &\leq \frac{1}{e} \cdot opt \end{aligned}$$

∎

Algorithm 2 Greedy Algorithms for Set Cover

1: $I \leftarrow \emptyset$, $V = [n]$
2: **while** $V \neq \emptyset$ **do**
3: let $i^* \in [m]$, with maximum $|S_{i^*} \setminus \bigcup_{i \in I} S_i|$
4: $I \leftarrow I + \{i^*\}$, $V \leftarrow V \setminus S_j$
5: **return** I

Theorem 3.6 (Feige) *: Unless $NP \subseteq DPTIME(n^{O(lgn)})$, then there is no $(1 - \frac{1}{e} + \varepsilon)$-approximation for maximum coverage for any $\varepsilon > 0$.*

Now we turn to Set Cover problem. The approximation algorithm for Set Cover is almost the same with the one for Maximum Coverage.

Here we assume that optimum solution of Set Cover can cover all the n elements using k set. So the question will be: for what iteration j can we guarantee $n - y_j = 0$?

Since n and y_j are both integers and $n > y_j$, so the condition $n - y_j = 0$ can be replaced with $n - y_j < 1$. For Set Cover problem, *opt* will be n. Via lemma 3.1, we can get $n - y_j \leq (1 - \frac{1}{k})^j \cdot n$. To guarantee that at j iteration we have $n - y_j < 1$, there is $(1 - \frac{1}{k})^j \cdot n < 1 \Rightarrow$

$$j > lg_{1-\frac{1}{k}} \frac{1}{n} = \frac{\ln n}{\ln \frac{1}{1-\frac{1}{k}}} = \Theta(k \cdot \lg n)$$

Theorem 3.7 *Greedy algorithm is a $(c \cdot \lg n)$-approximation for Set Cover.*

3.3 Sub-modular function

A sub-modular function is a set function whose value has the property that the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases.

Let 2^Ω be all subsets of Ω.

Definition 3.8 (Sub-modular function) *$f : 2^\Omega \to \mathbb{R}_{\geq 0}$ is a sub-modular function if $\forall X \subseteq Y \subseteq \Omega$ and $u \in \Omega \setminus Y$, we have $f(Y \bigcup \{u\}) - f(Y) \leq f(X \bigcup \{i\}) - f(X)$.*

Given $S_1, S_2, \cdots, S_m \subseteq [n]$. $\forall I \subseteq [m]$, define $g[I] = |\bigcup_{i \in I} S_i|$, and we will have below lemma.

Lemma 3.9 *g is a sub-modular on $2^{[m]}$.*

Next, we introduce a general problem based on sub-modular function called max-sub-modular-function problem.

Definition 3.10 (max-sub-modular-function) *Given a monotone sub-modular function $f : 2^{[m]} \to \mathbb{R}_{\geq 0}$, and an integer $k \geq 1$, find a set $I \subseteq [m]$, $|I| = k$, so as to maximize $f[I]$.*

Algorithm 3 Greedy Algorithms for Max-Sub-Modular-Function

1: $I \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to k **do**
3: let $u \in \Omega$ be the element that maximizes $f(I \bigcup \{u\}$
4: $I \leftarrow I + \{u\}$
5: **return** I

Many problems, such as our Maximum Coverage problem, can be cast as special cases of this general maximization problem under suitable constraints.

Similarly, we also have greedy algorithm for max-sub-modular-function problem.

Let I^* be the optimum solution, $|I^*| = k$, and $f(I^*) = opt$.

Lemma 3.11 $f(I \bigcup I^*) - f(I) \leq \sum_{u \in I^* \setminus I} (f(I \bigcup \{u\}) - f(I))$.

Proof: Suppose $I^* \setminus I = \{u_1, u_2, \cdots, u_p\}$, we have

$$\begin{aligned} f(I \bigcup I^*) - f(I) &= f(I \bigcup \{u_1\}) - f(I) \\ &+ f(I \bigcup \{u_1, u_2\}) - f(I \bigcup \{u_1\}) \\ &\dots \\ &\dots \\ &+ f(I \bigcup \{u_1, u_2, \cdots, u_p\}) - f(I \bigcup \{u_1, u_2, \cdots, u_{p-1}\}) \\ \\ &\leq f(I \bigcup \{u_1\}) - f(I) + f(I \bigcup \{u_2\}) - f(I) + \cdots + f(I \bigcup \{u_p\}) - f(I) \\ &= \sum_{u \in I^* \setminus I} (f(I \bigcup \{u\}) - f(I)) \end{aligned}$$

■

From this lemma, we can derive that there exists a $u \in I^* \setminus I$, so that $f(I \bigcup \{u\}) - f(I) \geq \frac{1}{k}(f(I^*) - f(I)) \geq \frac{1}{k}(opt - f(I))$. That equals to

$$\begin{aligned} opt - f(I \bigcup \{u\} &\leq opt - f(I) - \frac{1}{k}(opt - f(I)) \\ &= (1 - \frac{1}{k})(opt - f(I)) \end{aligned}$$

With same proof, for max-sub-modular-function, the greedy algorithm gives $(1 - \frac{1}{e})$-approximation.