Lecture 2

Wednesday, March 22, 2017 1:18 PM

Problems

① Set Cover

Input: • Universe of n elements $\{e_1, \ldots, e_n\} = \mathcal{U}$
• Sets $S_1, S_2, \ldots, S_m \subseteq \mathcal{U}$
• Costs $c(S_1), c(S_2), \ldots, c(S_m) \geq 0$

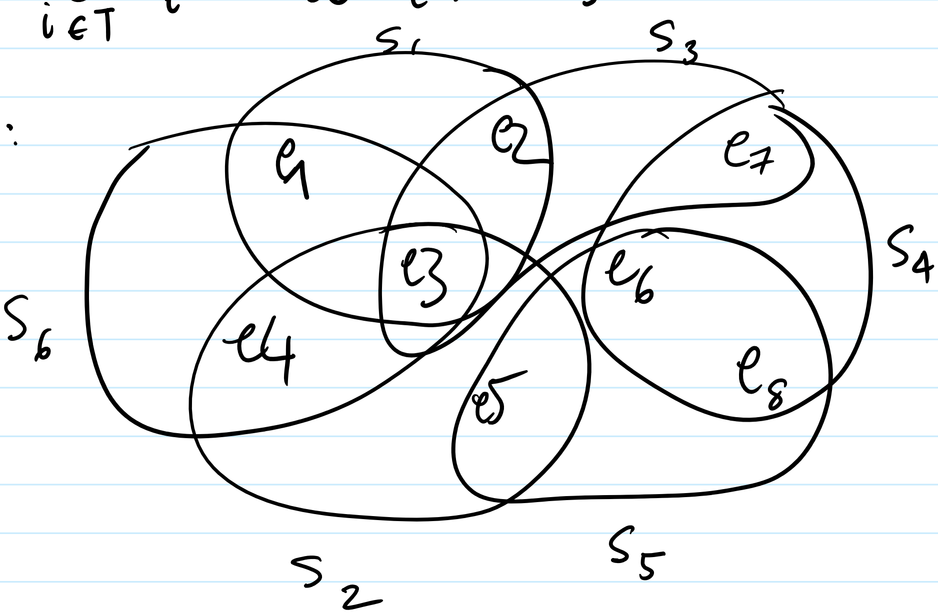
(set family)

Output: - A set cover.
- Subcollection of the input sets whose union cover all the elements.

- $T \subseteq \{1, 2, \ldots, m\}$ is a set-cover if

$$\bigcup_{i \in T} S_i = \mathcal{U} = \{1, \ldots, n\}$$

Example:



$\{1, 2, 4\}$ is a set cover

S_0 is $\{3, 5, 6\}$.

Objective: Output a set cover with

Objective: Output a set cover with minimum total cost, ie,

return $T \subseteq \{1, \dots, m\}$ minimizing $\sum_{i \in T} c(S_i)$

② Max k-Coverage

Input: · Set-family (Universe, Sets)

Output: $\le k$-sets indexed by $T \subseteq \{1, \dots, m\}$

$$|T| \le k$$

Goal: Maximize $\left| \bigcup_{i \in T} S_i \right|$

For this problem, no costs on sets.

Algorithm for Set Cover

- $A = \emptyset$ // index of sets covered by algo.
- $C = \emptyset$ // C is the set of covered elts initialized to empty set
- At any pt, $X \overset{(def)}{:=} U \setminus C$

// X : set of uncovered elts

- While $C \neq U$:
  - Pick set i which minimizes

"cost per new elt" $\left\{ \dfrac{c(S_i)}{|S_i \cap X|} \right.$ ← cost of set; ← # of new elts S_i covers

  - $A = A \cup i$
  - $C = C \cup S_i$, $X = X \setminus S_i$

Analysis

- Let's rename the sets s.t. $\{S_1, S_2, \ldots, S_r\}$ are the sets picked by the algorithm in that order.
- Moreover, let C_t be the set of covered elts after set S_t has been picked.

Note: $C_0 = \emptyset$, $C_1 = S_1$, $C_2 = S_1 \cup S_2, \ldots$

Similarly, we define $X_t = U \setminus C_t$.

- Finally, let the optimum set cover be indexed by $O \subseteq \{1, \ldots, m\}$.

Fix a "time" instant t just before S_t is picked. X_{t-1} is the set of uncovered elts.

$$\frac{c(S_t)}{|S_t \cap X_{t-1}|} \leq \frac{c(S_j)}{|S_j \cap X_{t-1}|} \quad \text{for any other set } S_j$$

In particular

$$\text{LHS} \leq \frac{c(S_j)}{|S_j \cap X_{t-1}|} \quad \forall j \in O$$

Fact: for any $a, b, c, d \geq 0$, we have

$$\min\left\{\frac{a}{b}, \frac{c}{d}\right\} \leq \frac{a+c}{b+d} \leq \max\left\{\frac{a}{b}, \frac{c}{d}\right\}$$

$$\therefore \frac{c(S_t)}{|S_t \cap X_{t-1}|} \leq \frac{\sum_{j \in O} c(S_j) \; \{= opt\}}{\sum_{j \in O} |S_j \cap X_{t-1}| \; \{\geq |X_{t-1}|\}}$$

$\because O$ is a set cover

$$\frac{c(S_t)}{|X_{t-1}| - |X_t|} \leq \frac{OPT}{|X_{t-1}|}$$

$$\Rightarrow c(S_t) \leq OPT \cdot \left(\frac{|X_{t-1}| - |X_t|}{|X_{t-1}|}\right)$$

$$\Rightarrow \quad c(S_t) \le OPT \cdot \left(\frac{|X_{t-1}| - |X_t|}{|X_{t-1}|} \right)$$

for all t

$$\therefore \; ALG = \sum_{t=1}^{r} c(S_t)$$

$$\le OPT \left(\frac{|X_0| - |X_1|}{\underbrace{|X_0|}_{=n}} + \frac{|X_1| - |X_2|}{|X_1|} + \cdots + \frac{|X_{r-1}| - \overbrace{|X_r|}^{=0}}{|X_{r-1}|} \right)$$

$$\le OPT \left(\underbrace{\frac{1}{n} + \frac{1}{n-1} + \cdots + 1}_{\text{is called } H(n) \text{ or simply } H_n \text{ and is } \approx \ln n} \right)$$

Theorem: The algorithm described above is a H_n-factor approximation algorithm for the Set-Cover Problem.

A Better Analysis (Charging Argument)

- When we pick set S_t, for every element $e_j \in S_t \cap X_{t-1}$, ie, all the new elts that S_t covers, we put a "charge"

$$\alpha_j := \frac{c(S_t)}{|S_t \cap X_{t-1}|} \quad \text{on } e_j$$

At the end, every element gets a charge & $\sum_{j=1}^{n} \alpha_j = ALG.$

- Now, fix a set S_i in the optimal set cover, ie, $i \in O$.

Order the elements of S_i in the order they are covered by the algorithm, ie, in the order these elements obtain "charges":

- Say $|S_i| = k$, and its elements are ordered to be $\{e_1, \cdots, e_k\}$

Fix elt e_j. and consider the time it gets its charge. Just before this time $\{e_j, e_{j+1}, \cdots, e_k\}$ are uncovered by defn. Since S_i was a choice avail. to the algo,

$$\alpha_j \leq \frac{c(S_i)}{k-j+1}$$

- $$ALG = \sum_{j=1}^{n} \alpha_j \leq \sum_{i \in O} \sum_{j \in S_i} \alpha_j$$

$\because$ O is a valid set cover

$$= \sum_{i \in O} \sum_{j=1}^{|S_i|} \frac{c(S_i)}{|S_i|-j+1}$$

$$= \sum_{i \in O} c(S_i) \cdot H(|S_i|)$$

$\leq \quad OPT \cdot H_\Delta$

where $\Delta \equiv \max_{i=1}^{m} |S_i|$

note, $\Delta \leq n$, and can be much smaller.

Theorem: The algorithm described above is a H_Δ-approximation algorithm.

Algorithm for Max k-coverage

For $t = 1, 2, \ldots, k$

- Pick S_t which contains the most number of uncovered elements.

Analysis

- Rename the sets s.t. $S_1, \ldots, S_k$ are the sets picked by the algorithm.
- Let $A_i := \bigcup_{j \leq i} S_i$ be the elements covered by the first i sets.
- $ALG = |A_k|$, the number of elts covered

by our algorithm.

- Let $OPT = |A^*|$ be obtained by the sets indexed by O; $|O| = k$

What do we know?

$$|A_1| \geqslant \frac{OPT}{k}$$

since one of the set S_i, $i \in O$ must be $|S_i| \geqslant \frac{OPT}{k}$.

$$|A_2 \setminus A_1| \geqslant \frac{|A^* \setminus A_1|}{k} \quad \text{(same reason)}$$

$$|A_2 \setminus A_1| = |A_2| - |A_1|$$

$$\geqslant \frac{OPT - |A_1|}{k}$$

$$\therefore \quad |A_2| \geqslant \frac{OPT}{k} + \left(1 - \frac{1}{k}\right) \cdot |A_1|$$

More generally, for $i \geqslant 1$

$$|A_i| \geqslant \frac{OPT}{k} + \left(1 - \frac{1}{k}\right)|A_{i-1}|$$

$$\geq \frac{OPT}{k} + \left(1-\frac{1}{k}\right)\frac{OPT}{k} + \left(1-\frac{1}{k}\right)^2 |A_{i-2}|$$

$$\vdots$$

$$\geq \frac{OPT}{k}\left(1 + \left(1-\frac{1}{k}\right) + \cdots + \left(1-\frac{1}{k}\right)^{i-1}\right)$$

$$= OPT \cdot \left(1 - \left(1-\frac{1}{k}\right)^{i}\right)$$

$$\Rightarrow ALG \geq OPT \cdot \left(1 - \left(1-\frac{1}{k}\right)^{k}\right)$$

$$\geq OPT\left(1 - \frac{1}{e}\right)$$

Fact: $\left(1-\frac{1}{k}\right)^k \leq \frac{1}{e}$

plot & see, & then prove.

More generally, $e^x \geq 1 + x$ for all x and the above is obtained by setting $x = -\frac{1}{k}$.

Theorem: The Greedy algorithm is a

<u>Theorem</u>: The Greedy algorithm is a $\left(1 - \frac{1}{e}\right)$-factor appx. algorithm for the Max k-Coverage problem.