Further Discussion on Set Cover and Hitting Set

Yufei Tao's Teaching Team

Department of Computer Science and Engineering
Chinese University of Hong Kong

Set Cover

Let U be a finite set called the **universe**.

We are given a family $\mathcal{S}$ where

- each member of $\mathcal{S}$ is a set $S \subseteq U$;
- $\bigcup_{S \in \mathcal{S}} S = U$.

A sub-family $\mathcal{C} \subseteq \mathcal{S}$ is a **universe cover** if every element of U appears in at least one set in $\mathcal{C}$.

- Define the **cost** of $\mathcal{C}$ as $|\mathcal{C}|$.

The set cover problem:
Find a universe cover with the smallest cost.

Example: $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $\mathcal{S} = \{S_1, S_2, ..., S_5\}$ where

$$\begin{aligned} S_1 &= \{1, 2, 3, 4\} \\ S_2 &= \{2, 5, 7\} \\ S_3 &= \{6, 7\} \\ S_4 &= \{1, 8\} \\ S_5 &= \{1, 2, 3, 8\}. \end{aligned}$$

An optimal solution is $\mathcal{C} = \{S_1, S_2, S_3, S_4\}$.

Our Approximation Algorithm

1. $\mathcal{C} = \emptyset$
2. **while** U still has elements not covered by any set in $\mathcal{C}$
3. $\quad F \leftarrow$ the set of elements in U not covered by any set in $\mathcal{C}$
   /* for each set $S \in \mathcal{S}$, define its **benefit** to be $|S \cap F|$ */
4. $\quad$ add to $\mathcal{C}$ a set in $\mathcal{S}$ with the largest benefit
5. **return** $\mathcal{C}$

We proved in the lecture that the algorithm is $(1 + \ln |U|)$-approximate.

Next, we will prove that the algorithm is also h-approximate, where $h = \max_{S \in \mathcal{S}} |S|$.

Example: $\mathcal{S} = \{S_1, S_2, ..., S_5\}$ where

$$\begin{aligned} S_1 &= \{1, 2, 3, 4\} \\ S_2 &= \{2, 5, 7\} \\ S_3 &= \{6, 7\} \\ S_4 &= \{1, 8\} \\ S_5 &= \{1, 2, 3, 8\}. \end{aligned}$$

Then, $h = 4$.

Theorem: The algorithm returns a universe cover with cost at most $h \cdot OPT_{\mathcal{S}}$.

Proof. Suppose that our algorithm picks t sets. Every time the algorithm picks a set, at least one **new** element is covered. For each $i \in [1, t]$, denote by e_i an arbitrary element that is **newly** covered when the i-th set is picked.

Let $\mathcal{C}^*$ be an optimal universe cover. Because each e_i exists in at least one set of $\mathcal{C}^*$, we have:

$$
\begin{aligned}
t = \sum_{i=1}^{t} 1 \quad & \leq \quad \sum_{i=1}^{t} \#\text{ sets in } \mathcal{C}^* \text{ containing } e_i \\
& \leq \quad \sum_{e \in U} \#\text{ sets in } \mathcal{C}^* \text{ containing } e \\
& = \quad \sum_{S \in \mathcal{C}^*} |S| \leq |\mathcal{C}^*| \cdot h.
\end{aligned}
$$

Corollary: If $h = O(1)$, then our algorithm achieves a constant approximation ratio.

Remark: With a more careful analysis, we can actually prove that our algorithm has an approximation ratio of $1 + \ln h$.

- Not required in this course.

Our set cover algorithm can be used to solve many problems with approximation guarantees. Next, we will see two examples.

Vertex Cover

$G = (V, E)$ is an undirected graph. We want to find a small subset $V' \subseteq V$ such that every edge of E is incident to at least one vertex in V'. The optimization goal is to minimize $|V'|$.

Convert the problem to set cover:

- For every $v \in V$, define S_v = the set of edges incident on v.
- Apply our algorithm on the set-cover instance: $\mathcal{S} = \{S_v \mid v \in V\}$.

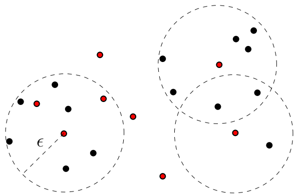
This gives an $\min\{O(\ln |V|), h\}$-approximate solution, where $h = \max_{v \in V} |S_v|$.

Remark: This algorithm is not as competitive as the 2-approximate vertex-cover algorithm we discussed in the lecture. But the point here is to demonstrate the usefulness of set cover, rather than improving the approximation ratio.

Facility Location

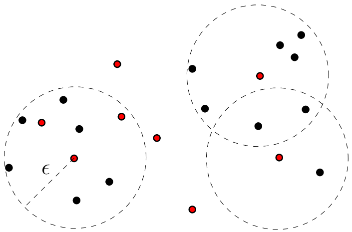
R = a set of n 2D red points, each called a **facility**
B = a set of n 2D black points, each called a **customer**
ϵ = a positive integer.

A subset $S \subseteq R$ is a **feasible facility set** if, for every black point $b \in B$, there is at least one point $r \in S$ with $dist(r, b) \leq \epsilon$.



OPT = the smallest size of all feasible facility sets.
Goal: Return a feasible facility set with size $OPT \cdot O(\log n)$ (assuming the existence of at least one feasible facility set).

Facility Location



Convert the problem to set cover:

- For every $r \in R$, define S_r = the set of black points b satisfying $dist(r, b) \leq \epsilon$.
- Apply our algorithm on the set-cover instance: $\mathcal{S} = \{S_r \mid r \in R\}$.

This gives an $O(\log n)$-approximate solution.

Next, we will turn our attention to the hitting set problem.

Hitting Set

Let U be a finite set called the **universe**.

We are given a family $\mathcal{S}$ where

- each member of $\mathcal{S}$ is a set $S \subseteq U$;
- $\bigcup_{S \in \mathcal{S}} S = U$.

A subset $H \subseteq U$ **hits** a set $S \in \mathcal{S}$ if $H \cap S \neq \emptyset$.
A subset $H \subseteq U$ is a **hitting set** if it hits all the sets in $\mathcal{S}$.

The hitting set problem:
Find a hitting set H of the minimize size.

Example: $U = \{1, 2, 3, 4, 5\}$ and $\mathcal{S} = \{S_1, S_2, ..., S_8\}$ where

$$\begin{aligned}
S_1 &= \{1, 4, 5\} \\
S_2 &= \{1, 2, 5\} \\
S_3 &= \{1, 5\} \\
S_4 &= \{1\} \\
S_5 &= \{2\} \\
S_6 &= \{3\} \\
S_7 &= \{2, 3\} \\
S_8 &= \{4, 5\}
\end{aligned}$$

An optimal solution is $H = \{1, 2, 3, 4\}$.

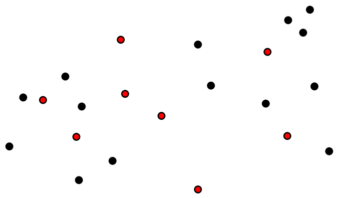
We can obtain a $(1 + \ln |\mathcal{S}|)$-approximate solution by resorting to a set-cover algorithm.

Set cover and hitting set are essentially the same problem.

Facility Location (Revisited)

R = a set of n 2D red points, each called a **facility**
B = a set of n 2D black points, each called a **customer**
ϵ = a positive integer.

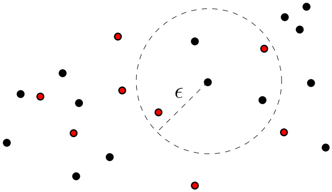
A subset $S \subseteq R$ is a **feasible facility set** if, for every black point $b \in B$, there is at least one point $r \in S$ with $dist(r, b) \leq \epsilon$.



OPT = the smallest size of all feasible facility sets.

How to cast the problem as an instance of the hitting set problem?

Facility Location (Revisited)



Convert the problem to hitting set:

- For every $b \in B$, define S_b = the set of red points r satisfying $dist(r, b) \leq \epsilon$.
- Solve the hitting set instance: $\mathcal{S} = \{S_b \mid b \in B\}$.

Why both set cover and hitting set?

Sometimes, one perspective is easier to perceive than the other.

Scheduling

We have t events: 1, 2, ..., t.
Set S_i contains the dates on which event i can be scheduled to take place.

Goal: Find the smallest number of dates to schedule all events.

Is this a hitting set or set cover problem in your eyes?

Earlier, for set cover, we proved that our algorithm taught in the class has an approximation ratio h, where h is the size of the largest set in the input collection.

As set cover is equivalent to hitting set, that result should also imply a new approximation ratio for hitting set. What is the ratio?