

Lecture 8: Set Cover

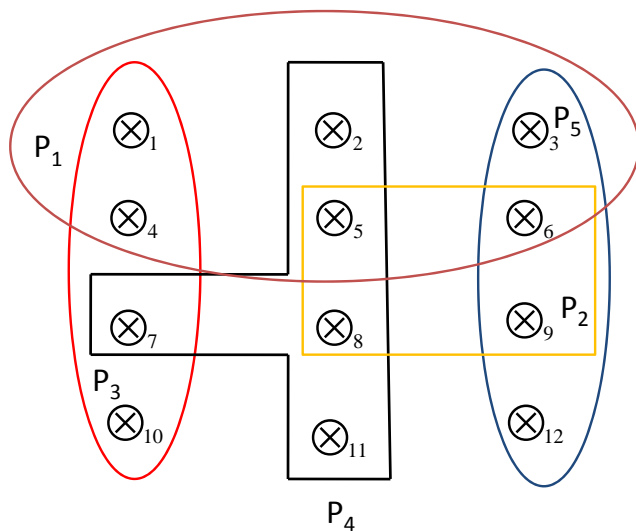
1 Ordinary Formulation

$$X = \{X_1 \dots X_n\} \text{ is a set of skills} \tag{1}$$

$$P = \{P_1 \dots P_r\} \text{ is a set of people} \tag{2}$$

$X_i \in P_j$ if person j possesses skill i . Find the (minimum set of people) smallest subset $R \subseteq P$ such that R covers all the skills i.e.:

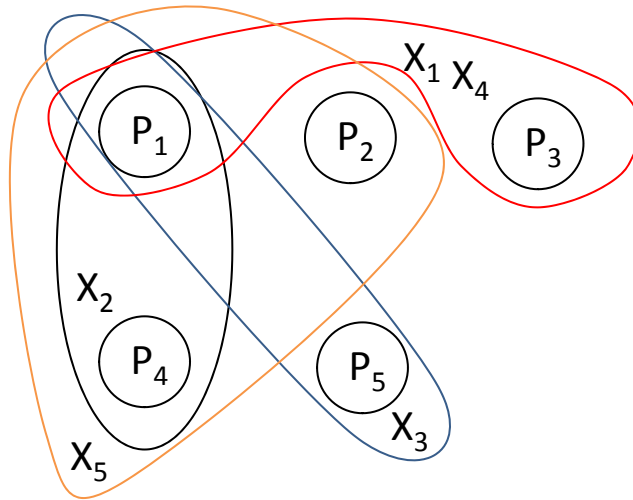
$$\cup_{P \in R} P = X \tag{3}$$



2 Hitting Set Formulation

$P = \{P_1 \dots P_r\}$ is the set of people and $X = \{X_1 \dots X_n\}$ is the set of skills
 $X_i \subseteq P$

Goal: Find a subset $H \subseteq P$ of minimum size such that $|H \cap X_i| \neq 0, \forall i$.



Set Cover is NP-hard. Present: $\mathcal{O}(\log n)$ -approximation algorithm for set cover.

Fact: can't get a better approximation factor unless $P = NP$. See Hochbaum, Chapter 10.

Use a simple greedy algorithm: at each stage, pick a set P (person) that covers the most of the elements that are still uncovered. Add P to the cover.

Theorem 2.0.1 : *The above greedy algorithm produces a solution that is at most $H(\max\{|S| : S \in P\})$ times optimal where*

$$H(n) = \sum_{i=1}^n \frac{1}{i} \leq \ln n + 1. \quad (4)$$

Proof: *Let C^* denote the size of an optimal set cover. Let S_i denote the i^{th} subset (person) that the greedy algorithm adds. Spread the cost ' L '*

of person i among all elements newly covered by S_i . Denote by C_x the cost charged to element x (note that x gets charged exactly once for the first person added who has skill x).

Suppose x is covered for the first time by S_i , then:

$$C_x = \frac{1}{|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})|}. \quad (5)$$

The algorithm finds a set of C of total cost $|C|$:

$$|C| = \sum_{x \in X} C_x \leq \sum_{S \in C^*} \sum_{x \in S} C_x \quad (6)$$

where inequality \leq follows because we know C_x is counted at least once for every X because C^* is a set cover.

Lemma: For all sets S belonging to P ,

$$\sum_{x \in S} C_x \leq H(|S|) \quad (7)$$

Assuming the Lemma we have:

$$|C| = \sum_{x \in X} C_x \leq \sum_{S \in C^*} \sum_{x \in S} C_x \leq \sum_{S \in C^*} H(|S|) \leq \quad (8)$$

$$\leq |C^*| \max_{|S|} H(S) \leq |C^*| H(n) = |C^*| (\lg n + 1) \quad (9)$$

Proof of Lemma: Fix $S \in P$ for all $i = 1, \dots, |C|$, let $u_i = |S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})|$ be the number of elements in S remaining uncovered after S_1, \dots, S_i have been selected by the algorithm $u_0 = |S|$.

Clearly $u_{i-1} \geq u_i$ and $u_{i-1} - u_i$ elements are covered for the first time by S_i :

$$\sum_{x \in S} C_x = \sum_{i=1}^k (u_{i-1} - u_i) \frac{1}{|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})|} \quad (10)$$

but we have

$$|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})| \geq |S - (S_1 \cup S_2 \cup \dots \cup S_{i-1})| = U_{i-1}. \quad (11)$$

$$\sum_{x \in S} C_x \leq \sum_{i=1}^k \frac{u_{i-1} - u_i}{u_{i-1}} \quad (12)$$

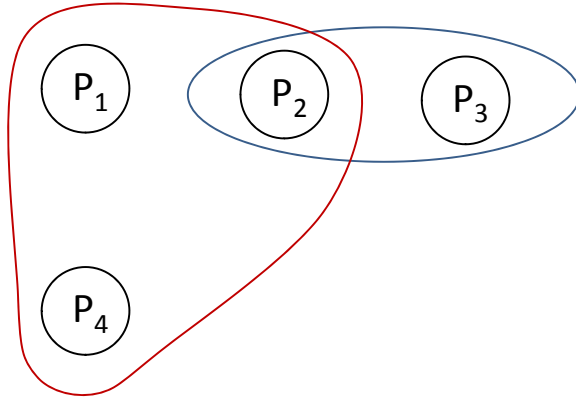
Lemma 2: If $a < b$ are integers then:

$$H(b) - H(a) = \sum_{i=a+1}^b \frac{1}{i} \geq \frac{b-a}{b} \quad (13)$$

By Lemma 2

$$\sum_{i=1}^k \frac{u_{i-1} - u_i}{u_{i-1}} \leq \sum_{i=1}^k H(u_{i-1}) - H(u_i) = H(u_0) - H(u_k) \quad (14)$$

$$(15)$$



P_1	70k
P_2	65k
P_3	30k
P_4	100k

Hiring P_i costs W_i (salary).

Find $H \subseteq P$ s.t. $H \cap X_i \neq \emptyset \forall x_i \in X$ and $\sum_{i \in H} W_i$ is minimized.

- people $\{P_1, \dots, P_n\}$

- salary $\{W_1, \dots, W_n\}$
- skills $\{X_1, \dots, X_n\}$

3 Integer Program Formulation

For $1 \leq i \leq n$, set:

$$V_i = \begin{cases} 1 & \text{if } P_i \in H \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Goal: minimize

$$\sum_{i=1}^n W_i \cdot V_i \quad (17)$$

subject to

$$\sum_{i: P_i \in x_j} V_i \geq 1, \forall x_j \in X \quad (18)$$

LP - relaxation ($0 \leq V_i \leq 1$) corresponds to allowing fractional part-time people. *LP* relaxation can be solved in poly time.

3.1 Weighted set cover randomized approximation algorithm

- Solve the *LP*
- Let \hat{V}_i be the value assigned to V in an optimal *LP* solution. Then we apply “randomized” rounding.

•

$$V_i = \begin{cases} 1 & \text{with probability } \hat{V}_i \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

The expected weight (cost) of the resulting partial cover is:

$$E\left(\sum_{i=1}^n W_i \cdot \hat{V}_i\right) = \sum_{i=1}^n W_i \cdot E[V_i] \quad (20)$$

$$\begin{aligned}
&= \sum_{i=1}^k nW_i \cdot \hat{V}_i \\
&= \text{OPT cost of LP} \leq \text{OPT cost of IP}
\end{aligned}$$

Now calculate the probability that X_i is covered. Suppose X_i contains P_1, \dots, P_k . We know that:

$$\sum_{j=1}^k \hat{V}_j \geq 1 \tag{21}$$

$$\begin{aligned}
\Pr[X_i \text{ is covered}] &= 1 - \text{no } P_j \text{ is chosen} & (22) \\
&= \Pr[P_1 \text{ isn't chosen } P_2 \text{ isn't chosen, } \dots \\
&\quad \dots P_k \text{ isn't chosen}] \\
&\quad (1 - \hat{V}_1) \cdot \dots \cdot (1 - \hat{V}_k) \\
&= 1 - \prod_{j=1}^k (1 - \hat{V}_j) \\
&\geq 1 - (1 - 1/k)^k \geq 1 - \frac{1}{e} > \frac{1}{2}
\end{aligned}$$

Repeat this game independently with new coin flips where m is the # of skills.

Probability that skill i is not covered after $\log m$ rounds:

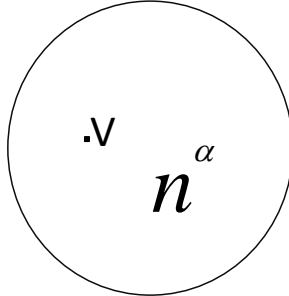
$$1 - \left(\frac{1}{2}\right)^{2 \lg m} = 1 - \frac{1}{2^m}. \tag{23}$$

Therefore, the probability that any skill S is not covered after $2 \log m$ rounds

$$\Pr[\text{any } S \text{ not covered}] \leq \sum_{S \in \mathcal{S}} \Pr[S \text{ is not covered}] \leq \sum_{S \in \mathcal{S}} \frac{1}{2^m} = m \cdot \frac{1}{2^m} = \frac{1}{2} \tag{24}$$

try it C times. The probability one of them doesn't cover all the skills is $< 1 - (1/2)^C$ cover.

Each round has weight $\leq OPT$. $2 \log m$ rounds have weight $\leq OPT 2 \log m$



4 Graph Spanners

Suppose instead of shortest paths, we only want short paths, that is, if $d(u, v)$ is the shortest path distance between u and v , and $r(u, v)$ is the shortest path distance between u and v in a graph containing only a subset of the edges. We want to guarantee:

$$r(u, v) \leq 3d(u, v), \forall u, v \in G \quad (25)$$

For every vertex v define set $=n^\alpha$ closest things (neighbors).

We will set $\alpha = 1/2$.

Will choose a set of landmarks with 2 conditions:

- shouldn't be too big
- every vertex has at least one landmark in its local neighborhood (i.e among its \sqrt{n} closest neighbors (that's the hitting set problem)).

Choose every vertex with prob $1/\sqrt{n}$ and do $\log n$ rounds.

Set of landmarks of size $\mathcal{O}(\sqrt{n} \lg n)$.

Now define a new graph H to be the following subset of the edges of G : it contains shortest single source path trees from every vertex in L , also contains truncated single source shortest path trees from every vertex to all vertices in its local neighborhood.

For a total of $\rightarrow n \cdot \sqrt{n} + \sqrt{n} \lg n \cdot n$ store $\mathcal{O}(n^{3/2} \lg n)$ edges.

If u is in v 's local neighborhood

- route directly

else

- route to closest landmark and then to v

Claim: the resulting route is at most 3 times the length of the shortest path. If u is in v 's local neighborhood, then we take exactly the shortest path in the local neighborhood. Otherwise, we have $d(L, v) < d(u, v)$, where L is v 's closest landmark, and so the route $d(u, L) + d(L, v) < d(u, v) + d(v, L) + d(L, v) < 3d(u, v)$ where the first inequality follows from the triangle inequality.