

3.1 k -Center

Definition 3.1.1 Given a metric space (V, d) and natural number k , the k -center problem is to select a subset $F \subseteq V$ with $|F| = k$ that minimizes $\max_{u \in V} d(u, F)$.

Note: In the above, $d(u, F)$ is taken to be $\min_{v \in F} d(u, v)$.

k -Center has applications in operations research and military planning, and admits several variants, including the following:

- k -Median: Input and feasible sets are as above, but uses objective function $\min_{F \subseteq V} \sum_{u \in V} d(u, F)$
- Facility Location: Feasible sets no longer carry the size restriction $|F| = k$, but each ‘center’ (element included in F) must be paid for, introducing a tradeoff.

Algorithm 1 A greedy algorithm for k -CENTER

Input: Metric space (V, d) , $k \in \mathbb{N}$.

Output: $F \subseteq V$, $|F| = k$, with minimum max distance to elements of V .

```
 $F \leftarrow \{u\}$ , for  $u \in V$  arbitrary  
while  $|F| < k$  do  
  Let  $u \in V \setminus F$  be the element maximizing  $d(u, F)$ .  
   $F \leftarrow F \cup \{u\}$   
end while  
return  $F$ 
```

Claim 3.1.2 F is feasible.

Proof: Clear; the algorithm increases $|F|$ by 1 on each iteration and ends when $|F| = k$. ■

Theorem 3.1.3 Algorithm 1 is a 2-approximation for the value of the optimal solution.

(Note: Intuitively, we might expect this result because we expect to be able to apply the triangle inequality when working with max distances, and the triangle inequality produces factors of 2.)

Proof of Theorem 3.1.3: Let F denote the output of the greedy algorithm, and let F^* denote the OPT solution. We will prove that for all $u \in V$, $d(u, F) \leq 2 \cdot \max_{v \in V} d(v, F^*) = 2 \cdot OPT$, from which it follows that $\max_{u \in V} d(u, F) \leq 2 \cdot \max_{u \in V} d(u, F^*) = 2 \cdot OPT$, and that we have a 2-approximation.

Definition 3.1.4 For each $v \in F^*$, let the *cluster* of v be given by $C(v) := \{u \in V : d(u, v) = d(u, F^*)\}$, where tie cases of the form $d(u, v_1) = d(u, v_2) = d(u, F^*)$ for $v_1, v_2 \in F^*$ are decided by placing u into one of the tied clusters arbitrarily.

Lemma 3.1.5 *Let $x, y \in C(v)$. Then $d(x, y) \leq 2 \cdot OPT$.*

Proof: By the triangle inequality, we have that $d(x, y) \leq d(x, v) + d(y, v)$; by the definition of $C(v)$ we have that this is equal to $d(x, F^*) + d(y, F^*) \leq 2 \cdot OPT$. ■

Returning to the proof of Theorem 3.1.3, we have two cases:

1. **Case 1:** For all $v \in F^*$, $C(v) \cap F \neq \emptyset$. Let $u \in V$, say with $u \in C(v)$ for $v \in F^*$. Then $F \cap C(v) \neq \emptyset$, so let $w \in C(v) \cap F$. Then $w \in F$, so $d(u, F) \leq d(u, w)$, and $u, w \in C(v)$ gives that $d(u, w) \leq 2 \cdot OPT$ by the lemma. Hence $d(u, F) \leq 2 \cdot OPT$. Note that this case does not use any properties specific to the greedy algorithm.
2. **Case 2:** There exists a $v \in F^*$ for which $C(v) \cap F = \emptyset$. By the pigeonhole principle (using that $|F| = |F^*| = k$), there exists $v' \in F^*$ s.t. $|C(v') \cap F| \geq 2$. So, suppose that $a, b \in C(v') \cap F$, and that a is added to F before b . Let F' give the set of elements added to F up to but not including b . Now let $u \in V$. Then we have the following series of inequalities:

$$\begin{aligned}
 d(u, F) &\leq d(u, F') && \text{(since } F' \subset F) \\
 &\leq d(b, F') \\
 &\leq d(b, a) && \text{(definition of } F', a) \\
 &\leq 2 \cdot OPT && \text{(Lemma 3.1.5)}
 \end{aligned}$$

The key inequality $d(u, F) \leq d(b, F')$ follows from the fact that if u is further from F' than b , the greedy algorithm would select u on the next iteration instead of b .

This exhausts all cases and completes the proof. ■

This leaves the question of whether the analysis above is tight, which may be answered via example:

Claim 3.1.6 *There are metric spaces for which the greedy algorithm returns a solution of value $2 \cdot OPT$*

Proof: Consider a set of 5 collinear vertices spaced at increments of 1 unit of distance, with $k = 2$. The optimal solution selects the second and fourth vertices, which have max distance 1 to all other vertices, but the greedy solution will always leave a vertex at distance 2 from F . ■

To conclude our analysis of k -Center, we answer the question of whether we can beat the constant factor of 2 incurred by the greedy algorithm with a hardness of approximation proof.

Theorem 3.1.7 *If there exists a c -approximation for k -Center for $c < 2$, $P = NP$.*

Proof of Theorem 3.1.7: By reduction from Vertex Cover (decision version): for the input $(G = (V, E), k)$, output ‘Yes’ if there exists a vertex cover of size at most k . Note that this is an NP-hard problem.

Let $[G = (V, E), k]$ be a VC instance, and let $V' := \{v_e \mid e \in E\}$. We reduce to k -Center on the set $V \cup V'$, with k as provided in the instance, and the metric $d(\cdot, \cdot)$ with distances:

- $d(u, v) = 1$ if $u, v \in V$ and $\{u, v\} \in E$.
- $d(u, v_e) = 1$ if $e = \{u, w\}$ for some w .

- $d(u, \cdot) = 2$ otherwise

To see why the above is a metric, notice in particular that every distance is either 1 or 2, triangle inequality cannot be violated.

Lemma 3.1.8 *G has a vertex cover of size k iff $(V \cup V', d)$ has a k -Center solution of value 1.*

Proof: $[\Rightarrow]$ Let S be a VC of G , $|S| = k$; we would like to show that S is a solution to k -Center of value 1. To see this, let $u \in V$. Note that in solving Vertex Cover we need never consider isolated vertices, so we can suppose wlog that there exists $v \in V$ such that $\{u, v\} \in E$. Because S is a VC, either u or v must be covered by S . If $u \in S$, $d(u, S) = d(u, u) = 0$. Else if $v \in S$, $d(u, S) \leq d(u, v) = 1$. Now let $v_e \in V'$, with $e = \{u, v\} \in E$. Then again either u or v is in S , and thus $d(v_e, S) \leq \min\{d(v_e, u), d(v_e, v)\} = 1$. So every vertex of $V \cup V'$ is within distance 1 from a node in S .

$[\Leftarrow]$ Let S be a k -Center solution of value 1. If there exists $v_e \in S \cap V'$, replace it in S by one of its endpoints (i.e. if v_e has $e = \{u, v\}$, add u or v to S and remove v_e), forming a new set $S' \subseteq V$. We would now like to show that S' is a VC with $|S'| \leq k$. To see this, let $e = \{u, v\}$ be an edge. Then $d(v_e, S) \leq 1$, because S was a k -Center solution of value 1. It follows that either $v_e \in S$, $u \in S$, or $v \in S$. In all cases, the replacement process above ensures that either u or v is in S' , and S' is a VC of G . ■

With this lemma in hand, suppose A is an algorithm which c -approximates k -Center, for $c < 2$. Then an algorithm for VC is given by reducing to k -Center by the steps described above, running A on that instance, and returning ‘Yes’ if A has value less than 2, and returning ‘No’ otherwise. If the starting Vertex Cover instance is a YES instance (there is a vertex cover of size at most k), then Lemma 3.1.8 implies that there is a k -Center solution of cost 1, and thus A must return a solution of cost at most $c \cdot 1 = c < 2$ so we will correctly return Yes. On the other hand, if the starting Vertex Cover instance is a NO instance then Lemma 3.1.8 implies that every k -Center solution has cost larger than 1 (and thus equal to 2 since all distances in the instance are either 1 or 2). Since A must return a feasible solution, it returns a value at least 2, so we will correctly answer No.

This completes the proof of the theorem. ■

3.2 Set Cover

The next problem we considered, Set Cover, generalizes the familiar Vertex Cover problem.

Definition 3.2.1 *The input in the Set Cover problem is a universe U , with $|U| = n$, and a family of sets S_1, S_2, \dots, S_m with $S_i \subseteq U$ for each i . Feasible solutions are index sets $I \subseteq [m]$ such that $\bigcup_{i \in I} S_i = U$, and the objective is to minimize $|I|$.*

Like k -Center, the Set Cover problem admits a straightforward greedy algorithm.

Theorem 3.2.2 *If OPT uses k sets to cover U , then Algorithm 2 uses at most $k(\log \frac{n}{k} + 1)$ sets.*

Note that if we can prove this Theorem, then we will obtain as corollaries that:

1. The greedy algorithm is an $O(\log n)$ -approximation.

Algorithm 2 A greedy algorithm for SET COVER

Input: Universe U of n elements, family $\{S_i\}_{i=1}^m$ of subsets of U .

Output: A minimum-size index set $I \subseteq [m]$ satisfying $\bigcup_{i \in I} S_i = U$.

$I \leftarrow \emptyset, X \leftarrow U$

while $X \neq \emptyset$ **do**

 Let i be the index maximizing $|X \cap S_i|$

$I \leftarrow i, X \leftarrow X \setminus S_i$

end while

return I

2. If for every i we have $|S_i| \leq B$, $B \in \mathbb{N}$, then the greedy algorithm is an $O(\log B)$ approximation, for then we must have $k \geq \frac{n}{B}$ if the OPT solution is to cover.

Proof of Theorem 3.2.2: Let I_t contain the indices added to I by the greedy algorithm after t iterations, and let n_t be the number of uncovered elements of U after t iterations (s.t. $I_0 = \emptyset$, and $n_0 = n$). We would like to show that the greedy algorithm makes ‘sufficient’ progress on each iteration; in particular we claim that $n_t \leq (1 - \frac{1}{k})n_{t-1}$. To see this, note first that the average set in OPT must cover at least $\frac{n_{t-1}}{k}$ elements, so there exists some single set in OPT covering at least $\frac{n_{t-1}}{k}$ elements. The greedy algorithm will therefore pick a set covering at least $\frac{n_{t-1}}{k}$ new elements, such that $n_t \leq n_{t-1} - \frac{n_{t-1}}{k} = (1 - \frac{1}{k})n_{t-1}$. ■

The proof of Theorem 3.2.2 was continued in Lecture 4 on 2/5/15.