Lecture Notes CS:5350 A greedy 2-approximation algorithm for the k-CENTER problem
Lecture 21: Nov 5, 2019
Scribe: Tarun Roy

1 Greedy Heuristics in Approximation Algorithm

Some of the well known greedy algorithms are —

1. Prim's and Kruskal's Algorithm (finds minimum spanning tree)
2. Dijkstra's Shortest Path First(SPF) algorithm (finds the shortest paths between nodes in a graph)

There are some situations where greedy algorithms don't work well either as exact algorithms or as approximation algorithms. To understand whether a greedy algorithm provides good or bad approximation, two problems are discussed in class:

- k-CENTER
- Set Cover

We will see that for the above mentioned problems the greedy algorithms does provides the best possible approximation.

2 k-CENTER Problem

Given n cities with specified distances, one wants to build k warehouses in different cities and minimize the maximum distance of a city to a warehouse. In graph theory this means finding a set of k vertices for which the shortest path distance of any vertex to its closest vertex in the size-k set is minimum.

2.1 Problem formulation

Let P be a set of n points. $D = P \times P \rightarrow \mathbf{R}_{>0}$ is a metric where,

$$D(p, S) = \min_{p' \in P} D(P, p') \quad for\ p \in P, S \subseteq P$$

2.2 k-CENTER

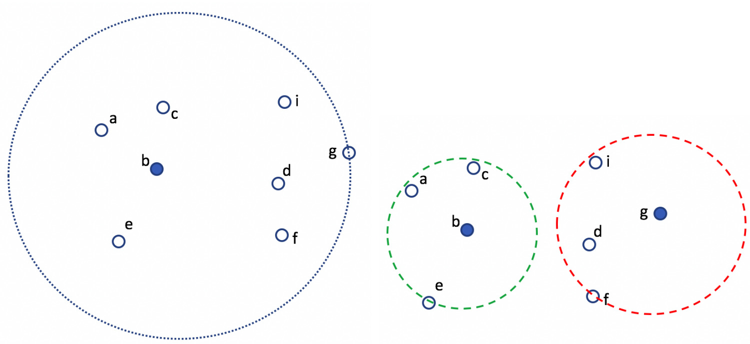
Input: D, k (positive integer)
Output: $S \subseteq P$ such that $|S| = k$ and, $\max_{p \in P} D(p, S)$ is minimized

2.3 Greedy Algorithm for k-CENTER

2.3.1 Idea:

First, we will pick one center arbitrarily as shown in Figure 1(a), point b is the center 1. Cluster is defined by the furthest point from b that is g, this distance will cover the entire set. This is a partial solution to the k-CENTER problem. To make another cluster we will chose this furthest point g as our center 2 for the new cluster.(see Figure 1(b))

Assignment of the centers are just for visualization. After two clusters we will take the next furthest point as the third center (f) and now we will have 3 clusters.



(a) 1^{st} iteration (b) 2^{nd} iteration

Figure 1: Greedy Choice

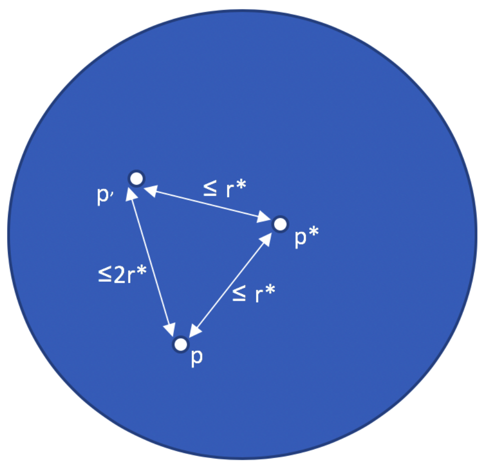


Figure 2: An arbitrary point B_i

Pseudo-code

$S \leftarrow \phi$
Add an arbitrary point to S
for $i \leftarrow 0$ *to* $k-1$ **do**
  Pick $p \in (P \backslash S)$ that maximizes $D(p, S)$
  Add p to S
end for
return S

Theorem 1. *The greedy algorithm for k-*CENTER *is a 2-approximation algorithm*

Proof. Let S^* be the optimal set of k centers and r^* be the maximum distance between a point in p and S^*. Therefore $r^* = \max_{p \in P} D(p, S^*)$ Let $B_1, B_2, ..., B_k$ denote the k balls centered at points in S^*. Note that for every $i, 1 \leq i \leq k$ every point in B_i is at distance at most r^* from the center of B_i

Case 1: Suppose there is exactly one point of S in each B_i, $1 \leq i \leq k$. S is the output of the greedy algorithm.
Then we can produce the following assignment of points in P to centers in S. Points in B_i get assigned center from S that is in B_i.
Let p be the center in S that is in B_1. Let p^* be the center in S^* that is in B_1 (see Figure 3). Let p' be an arbitrary point in B_1.

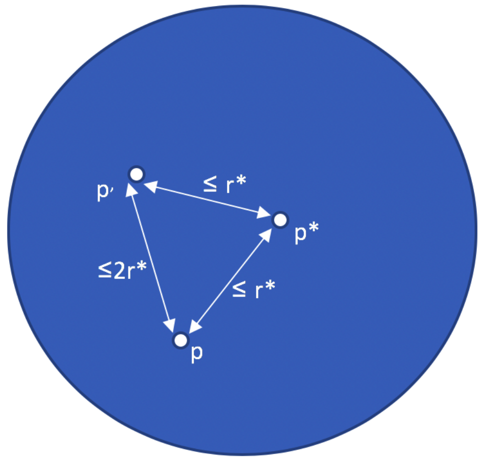


Figure 3: An arbitrary point B_1

$$D(p', p) \leq D(p, p^*) + D(p^*, p')$$
$$\Rightarrow D(p', p) \leq r^* + r^*.$$
$$\Rightarrow D(p', p) \leq 2r^*.$$

Hence, every ball centered at a point in S has radius $\leq 2r^*$.

Case 2: There exist a ball B_i that contains at least 2 centers from S. Let p^* be the center from S^* in B_i. p_1, p_2 denote two center in S that are in B_i (see Figure 2) and assume that p_2 is picked by greedy algorithm after p_1. Let S' be the centers picked by the greedy algorithm before p_2 is picked. When p_2 is picked,

$$D(p_2, S') \leq D(p_2, p_1) \leq 2r^*$$

Since, p_2 is greedy choice,

$$D(p, S') \leq 2r^*; \qquad \text{for any point } p \in P$$

Therefore S satisfies the property that

$$\max_{p \in P} D(p, S) \leq 2r^*$$

□

3 Hardness of Approximating k-CENTER

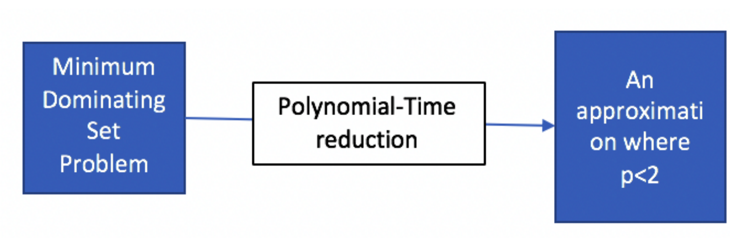


Figure 4: Polynomial-Time reduction

Theorem 2. *For any $\rho < 2$ if there is a ρ approximation algorithm for K-CENTER then $P = NP$*

Proof. We will reduce the Minimum Dominating Set(MDS) problem in polynomial time to K-CENTER.(see Figure 4) □