CSE515 Advanced Algorithms
Lecture 20
The k-center problem

Antoine Vigneron
antoine@unist.ac.kr

Ulsan National Institute of Science and Technology

May 6, 2021

1 Introduction

2 Clustering

3 Metric spaces

4 The k-center problem

5 A greedy algorithm

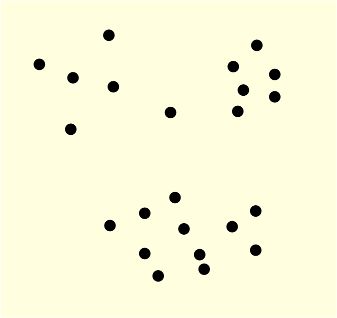
Introduction

- Assignment 3: deadline postponed to tomorrow due to a typo (the assignment paper mentioned May 7th as the deadline).
- Assignment 4 will be posted early next week.
- Today, I will present an algorithm for a clustering problem.
- Reference: Section 2.2 in The design of approximation algorithms by David P. Williamson and David B. Shmoys.

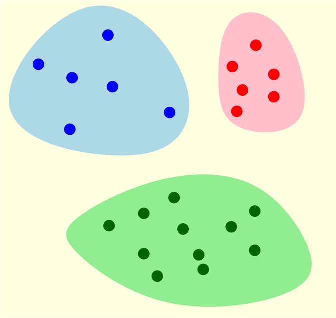
Clustering

Definition (clustering)

Clustering is the process of partitioning a set of objects into several subsets of similar objects, called *clusters*.



INPUT: A set of points



OUTPUT: Three clusters

Clustering

- Applications:
  - Data mining
  - Pattern recognition
  - Operations research (facility location)
  - ...
- Several approaches:
  - k-means
  - k-center
  - k-median
  - ...
- These problems are usually **NP**-hard.
- Here, we give an approximation algorithm for one such problem: the *k-center* problem.

Metric Spaces

Definition (Metric space)

Let M be a set and let d be a function $d : M \times M \to \mathbb{R}$. The pair (M, d) is a *metric space* if the following holds for all $x, y, z \in M$:

1. $d(x, y) = 0$ iff $x = y$
2. $d(x, y) = d(y, x)$ *(symmetry)*
3. $d(x, z) \leqslant d(x, y) + d(y, z)$ *(triangle inequality)*

- Remark: It follows from this definition that $d(x, y) \geqslant 0$ for all $x, y \in M$.

Examples of Metric Spaces

- The Euclidean plane $M = \mathbb{R}^2$ with the Euclidean distance

$$d(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}.$$

- An undirected graph with positive weights, and the shortest path distance

$$d(u, v) = \text{ weight of the shortest path from } u \text{ to } v.$$

- Any set M with the *discrete metric*

$$d(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

Metric Spaces

Definition (Ball)

The *open ball* of radius $r > 0$ about $x \in M$ is the set

$$\mathring{B}(x, r) = \{y \in M \mid d(x, y) < r\}.$$

The *closed ball* of radius $r > 0$ about $x \in M$ is the set

$$\bar{B}(x, r) = \{y \in M \mid d(x, y) \leqslant r\}.$$

- In this lecture, we only use closed balls, denoted $B(x, r)$.

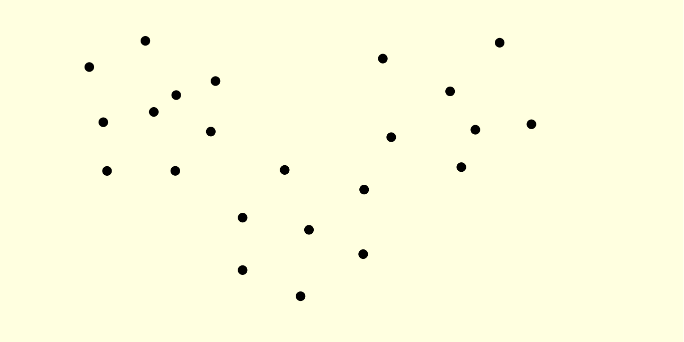
Definition (Distance to a subset)

The distance between $x \in M$ and a nonempty subset $S \subset M$ is

$$d(x, S) = \inf_{y \in S} \{d(x, y)\}.$$

The k-Center Problem

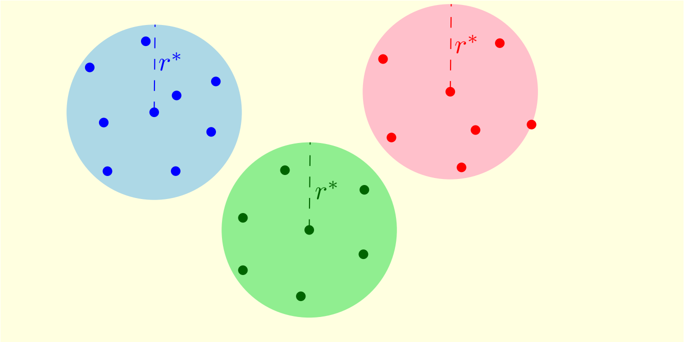
- The *k-center problem* is a clustering problem for metric spaces, where the k clusters are same radius balls. The goal is to minimize this radius.



example with $k = 3$

The k-Center Problem

- The *k-center problem* is a clustering problem for metric spaces, where the k clusters are same radius balls. The goal is to minimize this radius.



example with $k = 3$

The k-Center Problem

Input

A metric space (V, d), where $V = \{1, \dots, n\}$, and an integer k.

Output

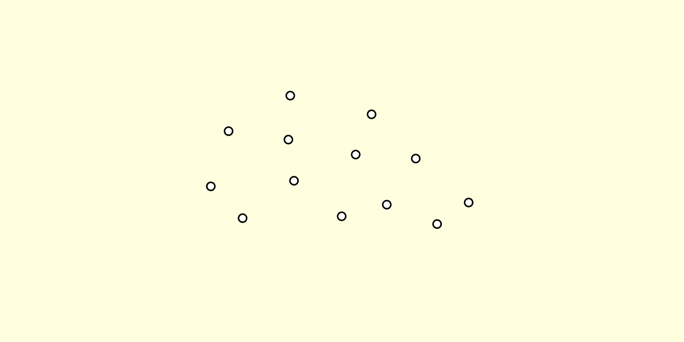
A subset $S^* \subseteq V$ of k vertices that minimizes the maximum distance between any point of V and the closest point in S^*:

$$S^* = \arg \min_{\substack{S: S \subset V \\ |S| = k}} \left(\max_{i \in V} d(i, S) \right).$$

- S^* is a set of *cluster centers*.
- The corresponding radius is

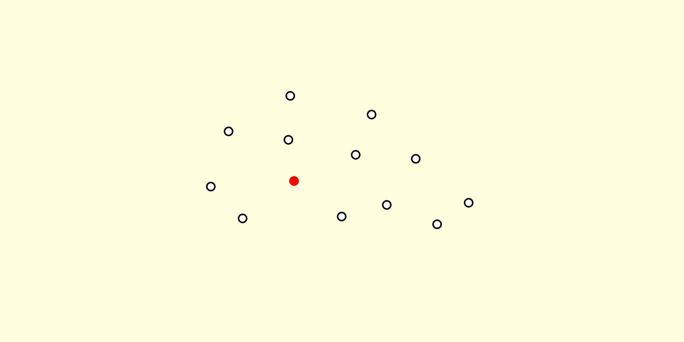
$$r^* = \min_{\substack{S: S \subset V \\ |S| = k}} \left(\max_{i \in V} d(i, S) \right).$$

Greedy Algorithm



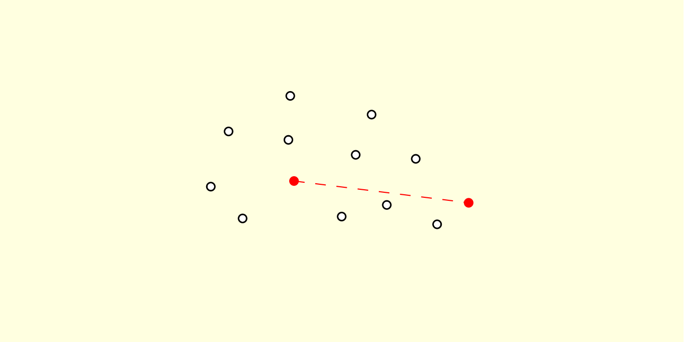
Input point set

Greedy Algorithm



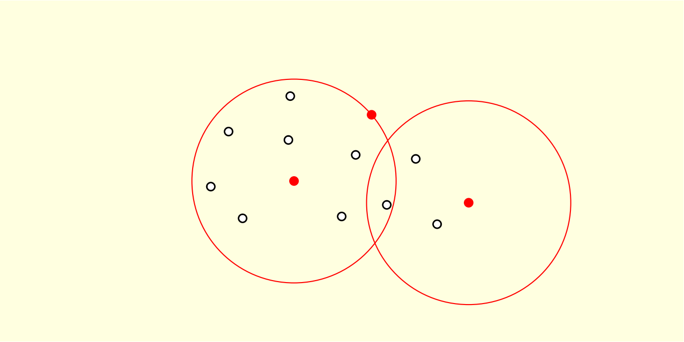
Pick a center arbitrarily.

Greedy Algorithm



Pick the furthest point from this center.

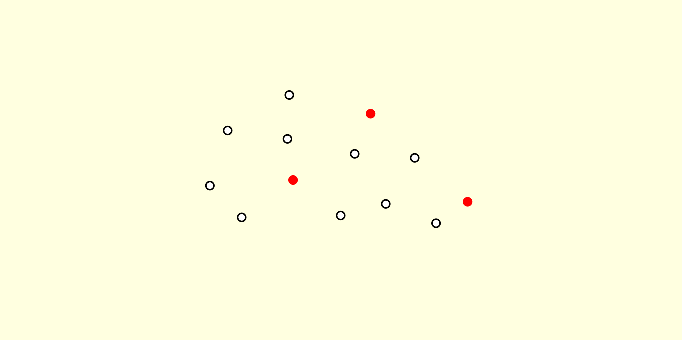
Greedy Algorithm



Pick the point that is farthest from the previous two centers.

i.e. this point maximizes the distance to the closest of the first two centers.

Greedy Algorithm



The 3 red points form an approximate 3-center.

Pseudocode

Greedy algorithm for k-center

pick arbitrary $i \in V$
$S \leftarrow \{i\}$
while $|S| < k$ **do**
    $j \leftarrow \arg\max_{j \in V} d(j, S)$
    $S \leftarrow S \cup \{j\}$
return S

Approximation factor

Theorem

The greedy algorithm is a 2-approximation algorithm for the k-center problem.

Proof.

Done in class. See textbook page 39. □

Hardness

Definition (Dominating set)

Given a graph $G = (V, E)$ and an integer k, the dominating set problem is to decide whether there exists a k-elements subset $S \subset V$ such that each vertex in V is in S, or is adjacent to a vertex in S.

- The dominating set problem is **NP**-hard. It follows that: (textbook p. 39)

Theorem

For any $\alpha < 2$, there is no α-approximation algorithm for the k-center problem unless **P**=**NP**.