k-center Clustering

TOP: Data Clustering 076/091

Instructor: Sayan Bandyapadhyay
Portland State University

Outline

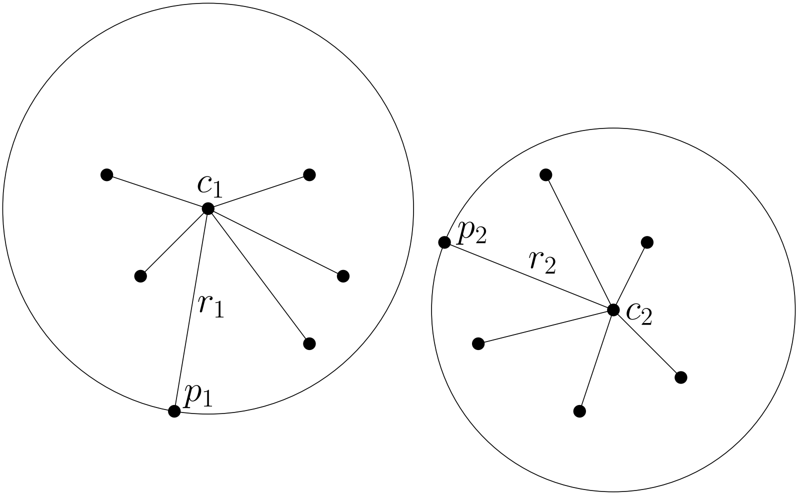
Discrete k-center Clustering

Given a set X of n points in the metric space (X, d)

- Find a set C of k points (cluster centers) in X that minimizes,

$$\begin{aligned} \mathrm{cost}(C) &= \max_{p \in X} d(p, NearestCenter(p)) \\ &= \max_{c \in C} \max_{p \in \mathrm{cluster}(c)} d(p, c) \end{aligned}$$

Each Cluster is a Ball



k-center Clustering

Given a set X of n points in the metric space (X, d)

- Find a set C of k points (cluster centers) in X that minimizes,

$$
\begin{aligned}
\text{cost}(C) &= \max_{p \in X} d(p, NearestCenter(p)) \\
&= \max_{c \in C} \max_{p \in \text{cluster}(c)} d(p, c) \\
&= \max_{c \in C} \text{radius}(c)
\end{aligned}
$$

Outline

Choices for Optimal Distances

$$\begin{aligned}\mathrm{cost}(C) &= \max_{c \in C} \mathrm{radius}(c) \\ &= \max_{c \in C} \max_{p \in \mathrm{cluster}(c)} d(p, c)\end{aligned}$$

- OPT is one of the $\binom{n}{2} \leq n^2$ distances: we can *guess* the optimal radius by trying out all choices
- We can assume we know the optimal radius

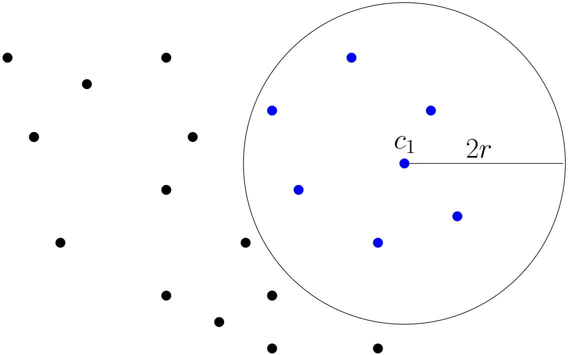
An Algorithm for k-center (Hochbaum-Shmoys '85)

Algorithm Greedy 1

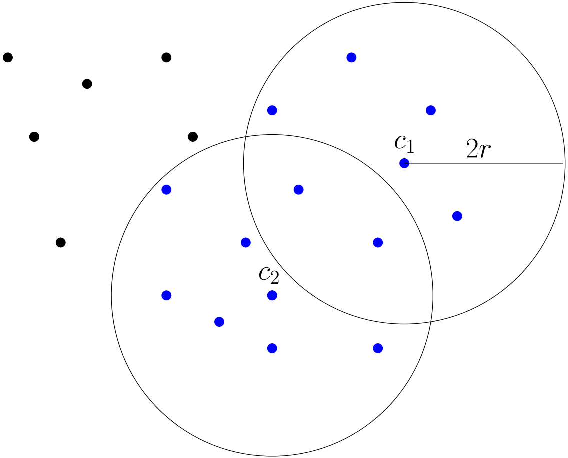
Require: Set of points X, radius r

$T \leftarrow X$
$C \leftarrow \emptyset$
while $T \neq \emptyset$ **do**
    Take any point x in T
    $C \leftarrow C \cup \{x\}$
    if $|C| = k + 1$ **then**
        return False
    end if
    for each $p \in T \setminus \{x\}$ **do**
        if $d(p, x) \leq 2 \cdot r$ **then**
            $T \leftarrow T \setminus \{p\}$
            Add p to cluster(x)
        end if
    end for
end while
return C

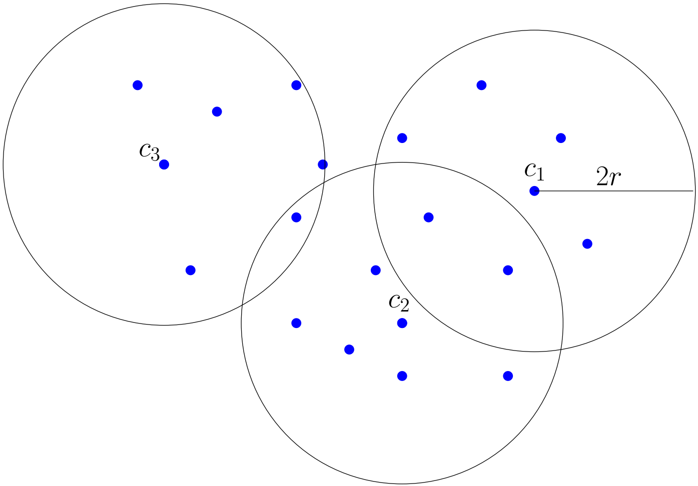
Demonstration of Greedy 1



Demonstration of Greedy 1



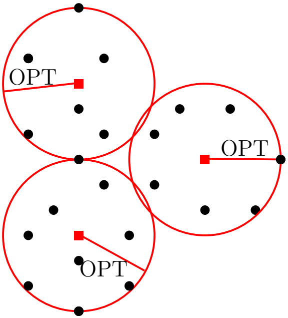
Demonstration of Greedy 1



Analysis of Greedy 1

- Suppose the given radius $r \geq OPT$
- Claim: The algorithm returns at most k centers/clusters
  - Optimal solution can cover all points with k balls of radius OPT

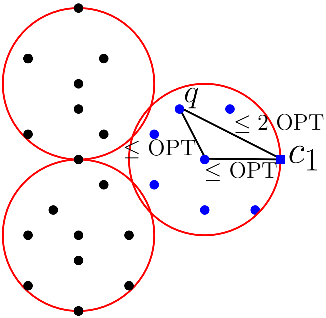
Covering by Optimal Solution



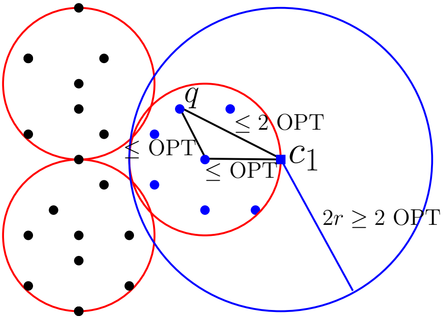
Analysis of Greedy 1

- Suppose the given radius $r \geq OPT$
- Claim: The algorithm returns at most k centers/clusters
  - Optimal solution can cover all points with k balls of radius OPT
  - Consider the center c_1 chosen; all the points in the optimal cluster containing c_1 are removed from T after 1st iteration

Points in a Cluster has Diameter 2·OPT



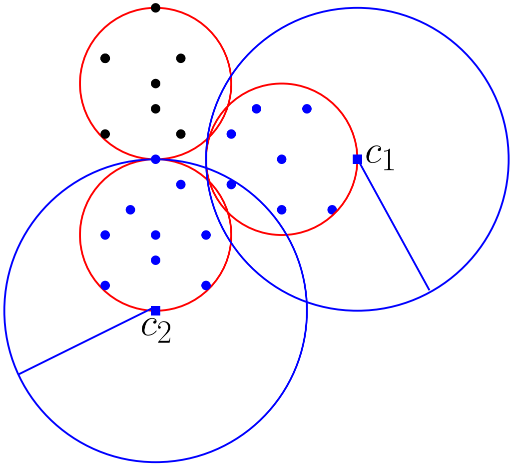
Removing All Points of the Optimal Cluster



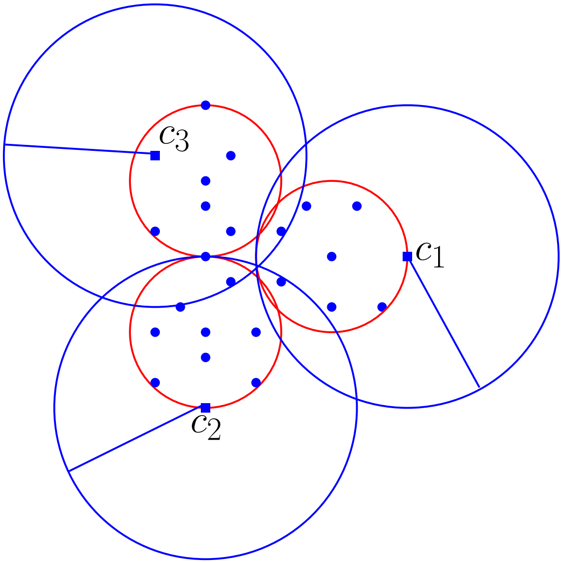
Analysis of Greedy 1

- Suppose the given radius $r \geq OPT$
- Claim: The algorithm returns at most k centers/clusters
    - Optimal solution can cover all points with k balls of radius OPT
    - Consider the center c_1 chosen; all the points in the optimal cluster containing c_1 are removed from T after 1st iteration; need a figure
    - In every iteration, points of at least one cluster gets removed

Removing All Points of the 2nd Optimal Cluster



Removing All Points of the 3rd Optimal Cluster



Analysis of Greedy 1

- Suppose the given radius $r \geq OPT$
- Claim: The algorithm returns at most k centers/clusters
  - Optimal solution can cover all points with k balls of radius OPT
  - Consider the center c_1 chosen; all the points in the optimal cluster containing c_1 are removed from T after 1st iteration; need a figure
  - In every iteration, points of at least one cluster gets removed
  - T is empty after at most k iterations
  - The algorithm returns a set of centers

Analysis of Greedy 1

- Suppose the given radius $r \geq OPT$
- Claim: The algorithm returns at most k centers/clusters
  - Optimal solution can cover all points with k balls of radius OPT
  - Consider the center c_1 chosen; all the points in the optimal cluster containing c_1 are removed from T after 1st iteration; need a figure
  - In every iteration, points of at least one cluster gets removed
  - T is empty after at most k iterations
  - The algorithm returns a set of centers

For $r = \text{OPT}$, we obtain a 2-approximation

Time Complexity

Algorithm Greedy 1

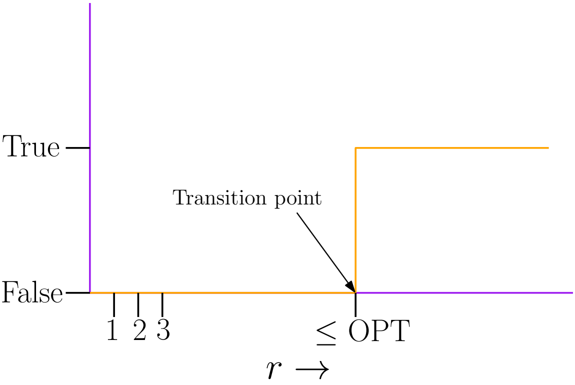
Require: Set of points X, radius r

$T \leftarrow X$
$C \leftarrow \emptyset$
while $T \neq \emptyset$ **do**
    Take any point x in T
    $C \leftarrow C \cup \{x\}$
    if $|C| = k + 1$ **then**
        return False
    end if
    for each $p \in T \setminus \{x\}$ **do**
        if $d(p, x) \leq 2 \cdot r$ **then**
            $T \leftarrow T \setminus \{p\}$
            Add p to cluster(x)
        end if
    end for
end while
return C

Time Complexity of Greedy 1

- While loop runs at most $k + 1$ times and the for loop runs at most $n - 1$ times. Time complexity is $O(nk)$
- If we have to run for all possible guesses, then complexity is at most $n^2 \cdot O(nk) = O(n^3 k)$
- Our algorithm works for $r \geq OPT$

Our algorithm works for $r \geq OPT$



Time Complexity of Greedy 1

- While loop runs at most $k + 1$ times and the for loop runs at most $n - 1$ times. Time complexity is $O(nk)$
- If we have to run for all possible guesses, then complexity is at most $n^2 \cdot O(nk) = O(n^3 k)$
- Our algorithm works for $r \geq OPT$
- Do a binary search on the range $[\min, \max]$
- Time complexity is $n^2 + (\log n^2 \cdot O(nk)) = O(n^2 + nk \log n)$

Outline

Another Algorithm for k-center

- We don't want to guess the optimal radius

Another Algorithm for k-center

- We don't want to guess the optimal radius
- Diameter $\Delta(X)$ is the maximum interpoint distance

Another Algorithm for k-center

- We don't want to guess the optimal radius
- Diameter $\Delta(X)$ is the maximum interpoint distance
- For $k = 2$, if the two maximum-distance points are in the same ball, radius is $\geq \Delta(X)/2$

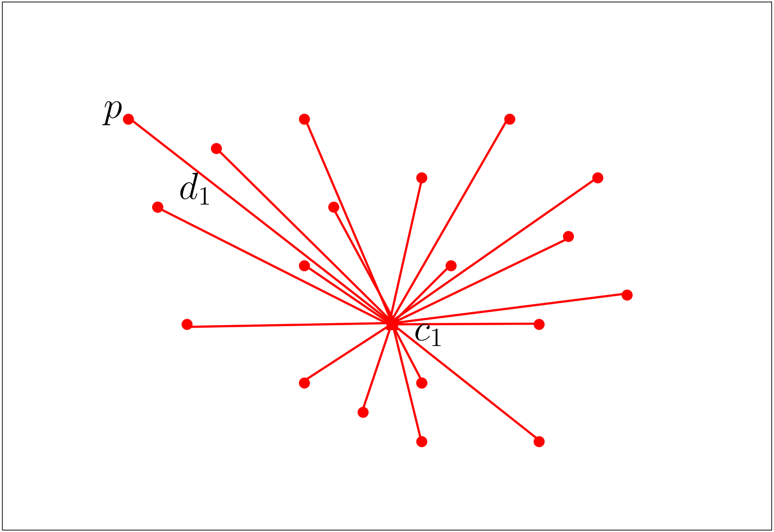
Another Algorithm for k-center

- We don't want to guess the optimal radius
- Diameter $\Delta(X)$ is the maximum interpoint distance
- For $k = 2$, if the two maximum-distance points are in the same ball, radius is $\geq \Delta(X)/2$
- This is as good as not splitting the points into 2 groups

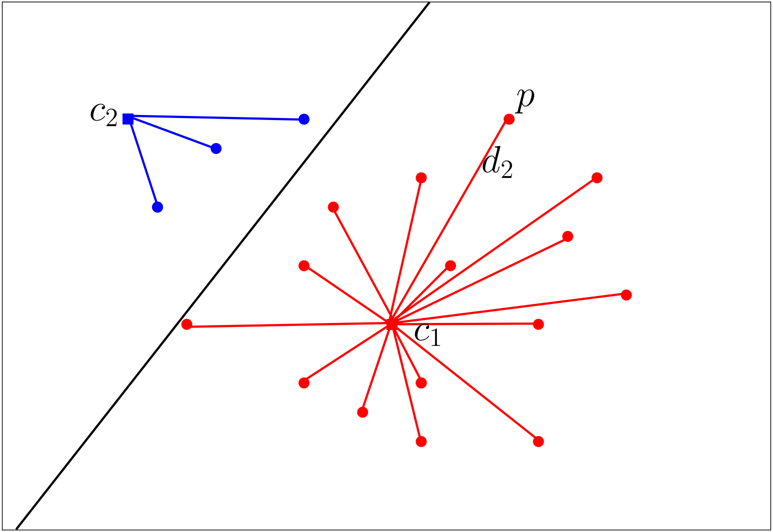
Another Algorithm for k-center

- We don't want to guess the optimal radius
- Diameter $\Delta(X)$ is the maximum interpoint distance
- For $k = 2$, if the two maximum-distance points are in the same ball, radius is $\geq \Delta(X)/2$
- This is as good as not splitting the points into 2 groups
- So, the two maximum-distance points should be in different groups
- This is the intuition behind our algorithm

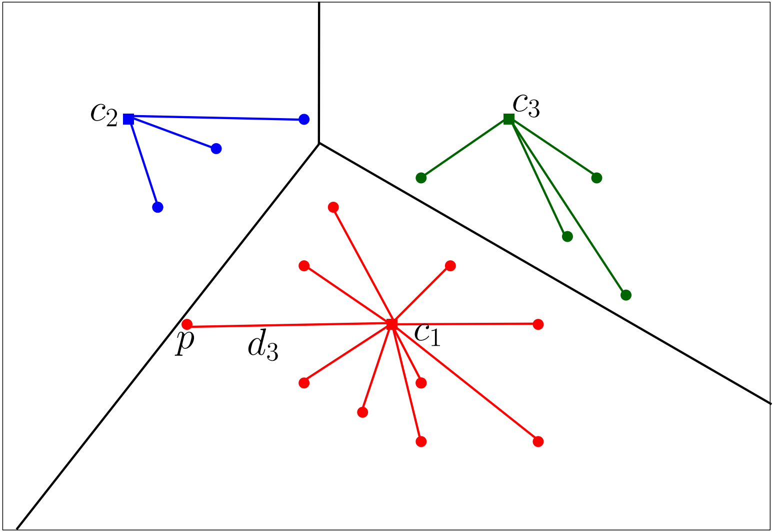
Picking the 1st center



Picking the 2nd center



Picking the 3rd center



Another Algorithm for k-center (Gonzalez, Dyer, Frieze '85)

Algorithm Greedy 2

Require: Set of points X

1: Select an arbitrary center c_1
2: $C \leftarrow \{c_1\}$
3: **for** $i = 2$ to k **do**
4: $\quad c_i \leftarrow \arg\max_{p \in X} d(p, C) \quad \triangleright\ d(p, C) = \min_{c \in C} d(p, c)$
5: $\quad C \leftarrow C \cup \{c_i\}$
6: **end for**
7: **return** C

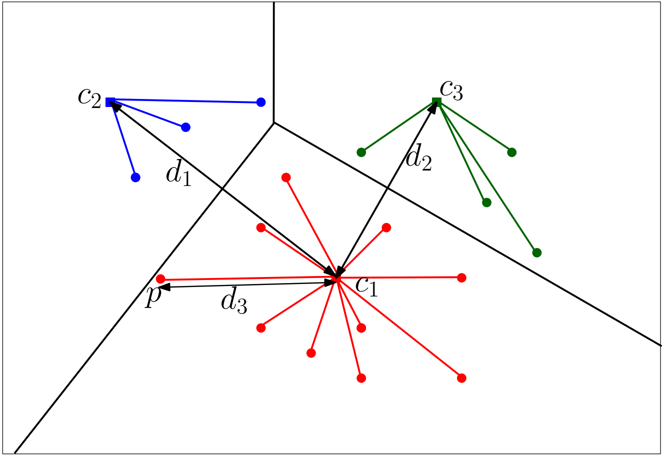
Time Complexity of Greedy 2

- The for loop runs $k - 1$ times
- Inside the for loop, we do $n * k$ distance computations. Time complexity is $O(nk^2)$

Time Complexity of Greedy 2

- The for loop runs $k - 1$ times
- Inside the for loop, we do $n * k$ distance computations. Time complexity is $O(nk^2)$
- We can keep track of the nearest center for each point in an array
- Inside the for loop, when we add a new center, we need to update at most n entries
- Time complexity is $O(nk)$

Analysis of Greedy 2



$d_1 \geq d_2 \geq \ldots \geq d_k$; our cost is d_k

What is a lower bound on the optimal cost?

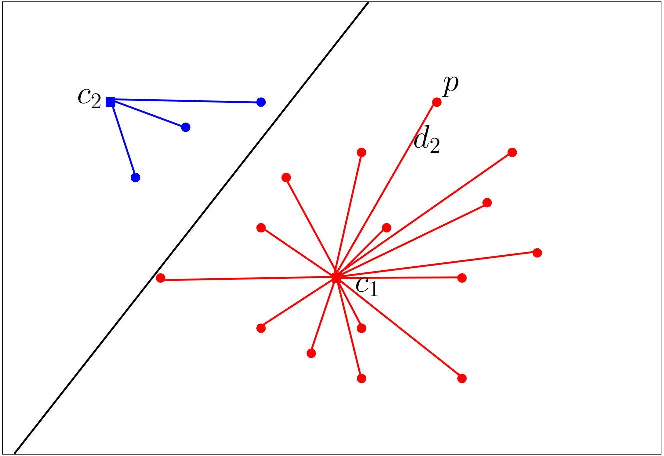
Plan: To show: $\boxed{\text{OPT} \geq d_k/\alpha} \Longrightarrow$ our cost $d_k \leq \alpha \cdot$ OPT

What is a lower bound on the optimal cost?

Plan: To show: $\boxed{\text{OPT} \geq d_k/\alpha} \Rightarrow$ our cost $d_k \leq \alpha \cdot$ OPT

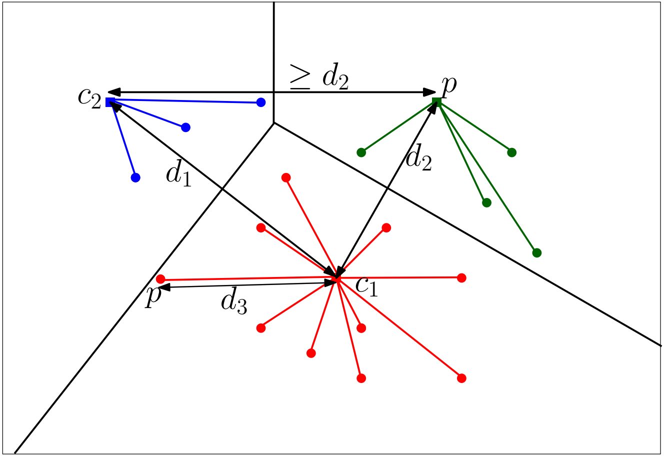
- c_2 is d_1 away from c_1
- c_3 is d_2 away from c_1 and c_2

Analysis of Greedy 2



p is d_2 away from c_1

Analysis of Greedy 2



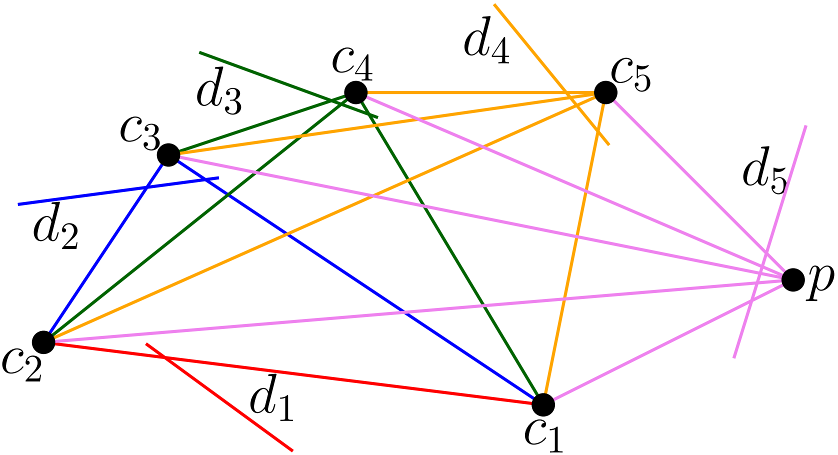
p is d_2 away from c_1 and c_2

What is a lower bound on the optimal cost?

Plan: To show: $\boxed{\text{OPT} \geq d_k/\alpha} \Rightarrow$ our cost $d_k \leq \alpha \cdot$ OPT

- c_2 is d_1 away from c_1
- c_3 is d_2 away from c_1 and c_2
- c_k is d_{k-1} away from $c_1, c_2, \ldots, c_{k-1}$
- The maximum-distance point p is d_k away from $c_1, c_2, \ldots, c_k$

$k+1$ points in X pairwise d_k away



$d_1 \geq d_2 \geq \ldots \geq d_k$

What is a lower bound on the optimal cost?

Plan: To show: $\boxed{\mathrm{OPT} \geq d_k/\alpha}$ $\Rightarrow$ our cost $d_k \leq \alpha \cdot \mathrm{OPT}$

- c_2 is d_1 away from c_1
- c_3 is d_2 away from c_1 and c_2
- c_k is d_{k-1} away from $c_1, c_2, \ldots, c_{k-1}$
- The maximum-distance point p is d_k away from $c_1, c_2, \ldots, c_k$
- So, among the $k+1$ points $\{c_1, c_2, \ldots, c_k, p\}$, 2 points must be in a single optimal cluster

What is a lower bound on the optimal cost?

Plan: To show: $\boxed{\mathrm{OPT} \geq d_k/\alpha} \Rightarrow$ our cost $d_k \leq \alpha \cdot \mathrm{OPT}$

- c_2 is d_1 away from c_1
- c_3 is d_2 away from c_1 and c_2
- c_k is d_{k-1} away from $c_1, c_2, \ldots, c_{k-1}$
- The maximum-distance point p is d_k away from $c_1, c_2, \ldots, c_k$
- So, among the $k+1$ points $\{c_1, c_2, \ldots, c_k, p\}$, 2 points must be in a single optimal cluster
- The diameter of this optimal cluster is $\geq d_k$, so radius is $\geq d_k/2$

What is a lower bound on the optimal cost?

Plan: To show: $\boxed{\mathrm{OPT} \geq d_k/\alpha} \Rightarrow$ our cost $d_k \leq \alpha \cdot \mathrm{OPT}$

- c_2 is d_1 away from c_1
- c_3 is d_2 away from c_1 and c_2
- c_k is d_{k-1} away from $c_1, c_2, \ldots, c_{k-1}$
- The maximum-distance point p is d_k away from $c_1, c_2, \ldots, c_k$
- So, among the $k+1$ points $\{c_1, c_2, \ldots, c_k, p\}$, 2 points must be in a single optimal cluster
- The diameter of this optimal cluster is $\geq d_k$, so radius is $\geq d_k/2$
- We obtain a 2-approximation

Outline

Drawbacks of the k-center model

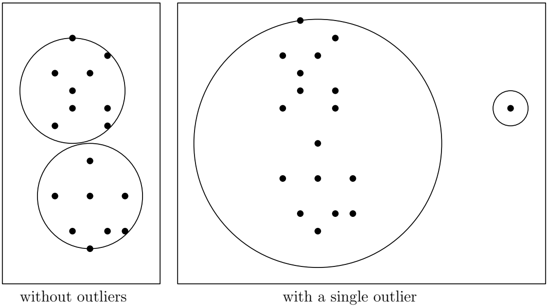
- Does not work if the clusters are not uniform

Drawbacks of the k-center model

- Does not work if the clusters are not uniform
- A large cluster/ball can get split: Dissection effect

Drawbacks of the k-center model

- Does not work if the clusters are not uniform
- A large cluster/ball can get split: Dissection effect
- Extremely sensitive to outliers




Thank You