

AAIL100: Introduction to Data Science

Ragesh Jaiswal, IITD

The k -center Problem

k -center on a Metric Space

Let (X, D) denote a metric space. Given a set of points $S \subset X$, and an integer k , output a set $T \subset X$ of points (called *centers*) such that $|T| = k$ and the following cost function is minimised:

$$\Phi(S, T) = \max_{x \in S} \min_{z \in T} D(x, z).$$

Approximation algorithm for k -center (Pick-farthest)

- Pick any $z \in S$ and set $T = \{z\}$
- While $|T| < k$
 - $z = \arg \max_{x \in S} \min_{t \in T} D(x, t)$
 - $T \leftarrow T \cup \{z\}$

- Claim 1: The above algorithm gives a factor-2 approximation.
- Claim 2: Getting approximation factor of $2 - \epsilon$ for any $\epsilon > 0$ is NP-hard.

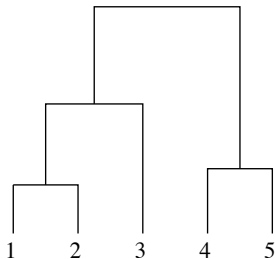
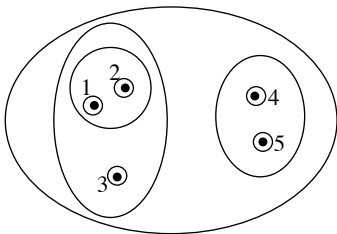
Hierarchical Clustering

Hierarchical Clustering

- When clustering data by solving the k -means/median/center problem, how do we know the value of k (i.e., the number of clusters)?
 - Idea 1: Run algorithms for various values of k .
 - Idea 2: **Hierarchical Clustering**: Do a *recursive* partitioning of the dataset. This gives clustering for all values of k .

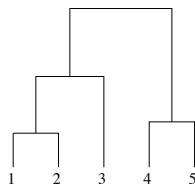
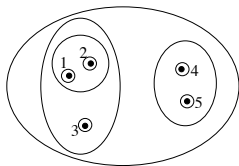
Hierarchical Clustering

- When clustering data by solving the k -means/median/center problem, how do we know the value of k (i.e., the number of clusters)?
 - Idea 1: Run algorithms for various values of k .
 - Idea 2: **Hierarchical Clustering**: Do a *recursive* partitioning of the dataset. This gives clustering for all values of k .



Hierarchical Clustering

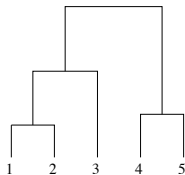
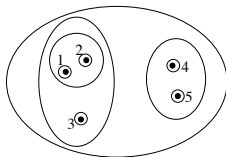
- When clustering data by solving the k -means/median/center problem, how do we know the value of k (i.e., the number of clusters)?
 - Idea 1: Run algorithms for various values of k .
 - Idea 2: **Hierarchical Clustering**: Do a *recursive* partitioning of the dataset. This gives clustering for all values of k .



- Is it possible to do a hierarchical clustering of a given dataset such that for all k , the clustering corresponding to k is optimal?

Hierarchical Clustering

- When clustering data by solving the k -means/median/center problem, how do we know the value of k (i.e., the number of clusters)?
 - Idea 1: Run algorithms for various values of k .
 - Idea 2: **Hierarchical Clustering**: Do a *recursive* partitioning of the dataset. This gives clustering for all values of k .



- Is it possible to do a hierarchical clustering of a given dataset such that for all k , the clustering corresponding to k is optimal?
- Is it possible to design approximation algorithms?

Hierarchical Clustering

Approximation algorithm for k -center

- Is it possible to design approximation algorithms?
 - We will see an 8-approximation algorithm for the k -center problem.

k -center problem

Let (X, D) denote any metric space. Given $S \subset X$ and an integer k , find a set of k points $T \subset X$ such that the following objective function is minimized $\Phi_T(S) = \max_{x \in S} \{\min_{t \in T} D(x, t)\}$.

- Recall, the **Pick-farthest** algorithm that we showed gives a 2-approximation.
- Suppose we run the algorithm for n steps. That is, picking all the n points. Let us number the points $1, 2, \dots, n$ in the order that they are picked.
- Let $R_i = \min_{t \in \{1, \dots, i-1\}} D(i, t)$.
- Claim 1: For all k , $R_{k+1} \leq 2 \cdot OPT_k$, where OPT_k denotes the optimal k -center cost.
- We will define a function $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ such that $\forall i, \pi(i) \in \{1, \dots, i-1\}$.

Hierarchical Clustering

Approximation algorithm for k -center

- Recall, the **Pick-farthest** algorithm that we showed gives a 2-approximation.
- Suppose we run the algorithm for n steps. That is, picking all the n points. Let us number the points $1, 2, \dots, n$ in the order that they are picked.
- Let $R_i = \min_{t \in \{1, \dots, i-1\}} D(i, t)$.
- Claim 1: For all k , $R_{k+1} \leq 2 \cdot OPT_k$, where OPT_k denotes the optimal k -center cost.
- We will define a function $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ such that $\forall i, \pi(i) \in \{1, \dots, i-1\}$. Such a function exhibit the following nice properties in the current context:
 - π defines a tree T^π .
 - Consider the k clustering given by the connected component of T^π when the edges $2 \rightarrow \pi(2), 3 \rightarrow \pi(3), \dots, k \rightarrow \pi(k)$ are deleted. This gives a Hierarchical Clustering.

Hierarchical Clustering

Approximation algorithm for k -center

- Let $R_i = \min_{t \in \{1, \dots, i-1\}} D(i, t)$.
- Claim 1: For all k , $R_{k+1} \leq 2 \cdot OPT_k$, where OPT_k denotes the optimal k -center cost.
- We will define a function $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ such that $\forall i, \pi(i) \in \{1, \dots, i-1\}$. Such a function exhibit the following nice properties in the current context:
 - π defines a tree T^π .
 - Consider the k clustering given by the connected component of T^π when the edges $2 \rightarrow \pi(2), 3 \rightarrow \pi(3), \dots, k \rightarrow \pi(k)$ are deleted. This gives a Hierarchical Clustering.

Algorithm

- Number the data points by the **Pick-farthest** algorithm
- For $i = 2, 3, \dots, n$: $R_i \leftarrow \min_{t \in \{1, \dots, i-1\}} D(i, t)$
- $R \leftarrow R_2$; $L_0 \leftarrow \{1\}$
- For $j > 1$: $L_j \leftarrow \{i : R/2^j < R_i \leq R/2^{j-1}\}$
- For $i = 1, \dots, n$: Let $\pi(i) \leftarrow$ closest point at a lower level.
- Output T^π

Hierarchical Clustering

Approximation algorithm for k -center

Algorithm

- Number the data points by the **Pick-farthest** algorithm
- For $i = 2, 3, \dots, n$: $R_i \leftarrow \min_{t \in \{1, \dots, i-1\}} D(i, t)$
- $R \leftarrow R_2$; $L_0 \leftarrow \{1\}$
- For $j > 1$: $L_j \leftarrow \{i : R/2^j < R_i \leq R/2^{j-1}\}$
- For $i = 1, \dots, n$: Let $\pi(i) \leftarrow$ closest point at a lower level.
- Output T^π

- The 8 approximation follows from the next two claims.
- Claim 1: For any i , $D(i, \pi(i)) \leq \frac{R}{2^{\text{level}(i)-1}}$, where $\text{level}(i)$ denotes index j such that $i \in L_j$.
- Claim 2: For any k , the induced k -clustering has cost $\leq 4R_{k+1} \leq 8OPT_k$.

End