COL870: Clustering Algorithms

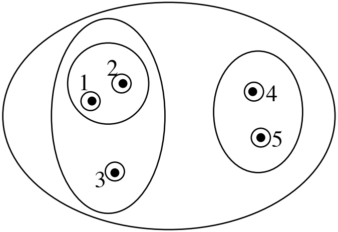
Ragesh Jaiswal, CSE, IIT Delhi

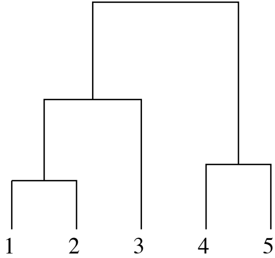
Hierarchical Clustering

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.

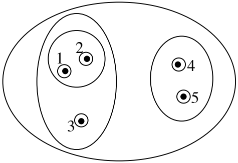
- When clustering data by solving the k-means/median/center problem, how do we know the value of k (i.e., the number of clusters)?
  - Idea 1: Run algorithms for various values of k.
  - Idea 2: Hierarchical Clustering: Do a *recursive* partitioning of the dataset. This gives clustering for all values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

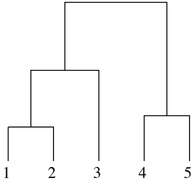




Hierarchical Clustering

- 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.

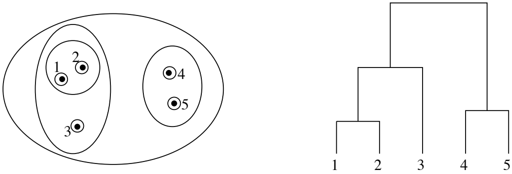




- Is it possible to do a hierarchical clustering of a given dataset such that for all k, the clustering corresponding to k is optimal?

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?

- 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 $1, 2, ..., n$ in the order that they are picked.
- Let $R_i = \min_{t \in \{1,...,i-1\}} D(i, t)$.
- 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 $\pi : \{1, ..., n\} \to \{1, ..., n\}$ such that $\forall i, \pi(i) \in \{1, ..., i-1\}$.

- 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 $1, 2, ..., n$ in the order that they are picked.
- Let $R_i = \min_{t \in \{1,...,i-1\}} D(i, t)$.
- 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 $\pi : \{1, ..., n\} \rightarrow \{1, ..., n\}$ such that $\forall i, \pi(i) \in \{1, ..., i-1\}$. 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 $2 \rightarrow \pi(2), 3 \rightarrow \pi(3), ..., k \rightarrow \pi(k)$ are deleted. This gives a Hierarchical Clustering.

- Let $R_i = \min_{t \in \{1,...,i-1\}} D(i,t)$.
- 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 $\pi : \{1,...,n\} \rightarrow \{1,...,n\}$ such that $\forall i, \pi(i) \in \{1,...,i-1\}$. 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 $2 \rightarrow \pi(2), 3 \rightarrow \pi(3), ..., k \rightarrow \pi(k)$ are deleted. This gives a Hierarchical Clustering.

Algorithm

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

Algorithm

- Number the data points by the **Pick-farthest** algorithm
- For $i = 2, 3, ..., n$: $R_i \leftarrow \min_{t \in \{1,...,i-1\}} D(i, t)$
- $R \leftarrow R_2$; $L_0 \leftarrow \{1\}$
- For $j > 1$: $L_j \leftarrow \{i : R/2^j < R_i \leq R/2^{j-1}\}$
- For $i = 1, ..., n$: 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, $D(i, \pi(i)) \leq \frac{R}{2^{level(i)-1}}$, where $level(i)$ denotes index j such that $i \in L_j$.
- Claim 2: For any k, the induced k-clustering has cost $\leq 4R_{k+1} \leq 8OPT_k$.

Hierarchical Clustering

- Hierarchical Clustering Algorithms fall into the following two broad categories:
  - Divisive Algorithms: This is a top-down approach where one starts with all points being in a single cluster and then recursively dividing a cluster into more than one clusters.
  - Agglomerative Algorithms: This is a bottom-up approach where one starts with n cluster (one each for every data point) and then merge small clusters to construct larger clusters.
- Agglomerative algorithms follow the following general template:
  - Start with n clusters, each a single data point and then repeatedly merge the two "closest" clusters.

Hierarchical Clustering

- Agglomerative algorithms follow the following general template:
  - Start with n clusters, each a single data point and then repeatedly merge the two "closest" clusters.
- Following are the typical notions of closeness of two clusters that are used in practice:
  - Single linkage: $\min_{i,j}\{D(i,j) | i \in C, j \in C'\}$.
  - Complete linkage: $\max_{i,j}\{D(i,j) | i \in C, j \in C'\}$.
  - Average linkage:
    - $mean\{D(i,j) | i \in C, j \in C'\}$.
    - $||mean(C) - mean(C')||^2$.
    - $\frac{|C| \cdot |C'|}{|C| + |C''|} \cdot ||mean(C) - mean(C')||^2$ (Ward's Method)

Theorem

For any $C, C' \subset \mathbb{R}^d$, we have

$$\Delta_1(C \cup C') = \Delta_1(C) + \Delta_1(C') + \frac{|C| \cdot |C'|}{|C| + |C'|} \cdot ||mean(C) - mean(C')||^2.$$

Here $\Delta_1(.)$ denotes the optimal 1-means cost.

End