Hierarchical Clustering

TOP: Data Clustering 076/091

Instructor: Sayan Bandyapadhyay

Portland State University

Outline

Guessing the right value of *k*

- Clustering: Exploratory mechanism

Guessing the right value of k

- Clustering: Exploratory mechanism
- What is the ideal value of k?

Guessing the right value of k

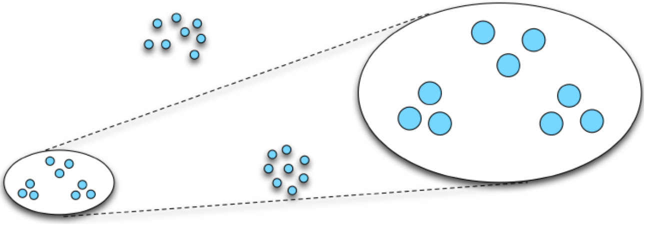
- Clustering: Exploratory mechanism
- What is the ideal value of k?
- Two ways
  - Try to actively guess k based on cost
  - A compact universal representation, encodes the clustering for all k

Guessing the right value of k

- Clustering: Exploratory mechanism
- What is the ideal value of k?
- Two ways
  - Try to actively guess k based on cost
  - A compact universal representation, encodes the clustering for all k

Hierarchical clustering is an example of the latter formulation

Hierarchical clustering



Views at different scales

Hierarchical clustering

- What is the right view?

Hierarchical clustering

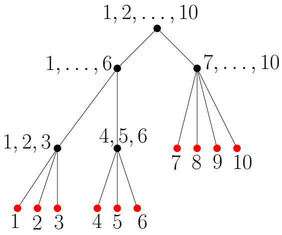
- What is the right view?
- Partition based clustering: depends on the cost

Hierarchical clustering

- What is the right view?
- Partition based clustering: depends on the cost
- Hierarchical: both are right
    - Multi-scale clustering
    - Effective in exploratory setting
    - One of the popular methods

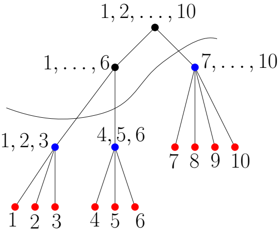
Representation

- A clustering is still a partition of X
- Multi-scale clustering is represented by a rooted tree
- Each node v corresponds to $S(v) \subseteq X$
  - Root r is associated with X; $S(r) = X$
  - If v is a child of u, $S(v) \subset S(u)$
  - If a node u has children $v_1, v_2, \ldots, v_m$, $\{S(v_1), S(v_2), \ldots, S(v_m)\}$ is a partition of $S(u)$



A Single Clustering

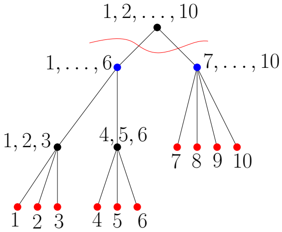
- A set of nodes $v_1, v_2, \ldots, v_k$ of the tree
  - no two v_i have an ancestor descendant relationship
  - any path from r to a leaf intersects exactly one v_i



$\{S(v_1), S(v_2), \ldots, S(v_k)\}$ is a partition of X

A Single Clustering

- A set of nodes $v_1, v_2, \dots, v_k$ of the tree
  - no two v_i have an ancestor descendant relationship
  - any path from r to a leaf intersects exactly one v_i



$\{S(v_1), S(v_2), \dots, S(v_k)\}$ is a partition of X

Outline

Two Approaches

- Top-down: Divisive
- Bottom-up: Agglomerative

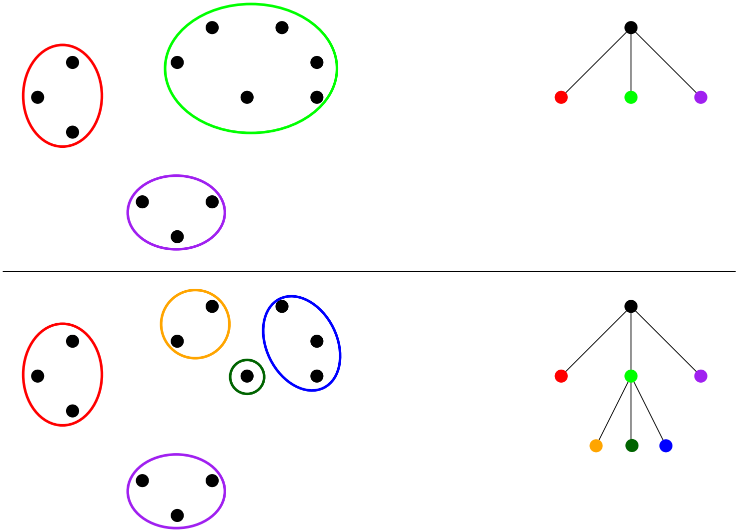
Divisive Hierarchical Clustering (DHC)

Algorithm DHC

Require: Set of points X, root node r, a constant c

1: $S(r) \leftarrow X$
2: **if** $|X| \leq c$ **then** return r
3: **end if**
4: Partition X into c pieces $X_1, \ldots, X_c$ using any c-clustering algorithm
5: Create nodes $v_1, \ldots, v_c$. $S(v_i) \leftarrow X_i$ for $1 \leq i \leq c$. Set r as the parent of each v_i
6: Recursively call DHC on each (X_i, v_i, c)
7: Return the tree rooted at r, and all $S(v)$

An Example with $c = 3$



Hierarchical Agglomerative Clustering (HAC)

Algorithm Template for HAC

Require: Set of points X

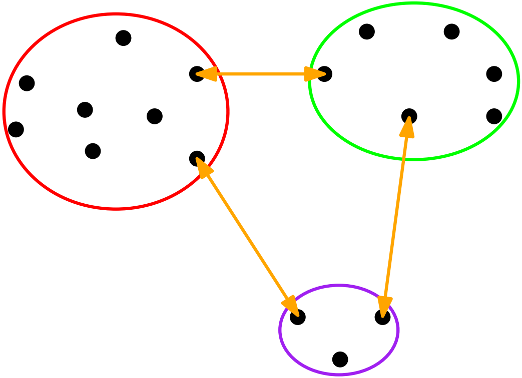
$\mathcal{S} \leftarrow \{(x, \{x\}) \mid x \in X\}$
$\mathcal{C} \leftarrow \mathcal{S}$
while $|\mathcal{C}| > 1$ **do**
  Find (v, C) and (v', C') in $\mathcal{C}$ that are the closest
  Create node r and cluster $\hat{C} = C \cup C'$. Set $S(r) = \hat{C}$
  Assign r as the parents of v and v'
  Insert $(r, \hat{C})$ and remove (v, C) and (v', C') from $\mathcal{C}$
  Insert $(r, \hat{C})$ into $\mathcal{S}$
end while

Distance between Two Clusters

- Single-Linkage
- Complete-Linkage
- Average-Linkage

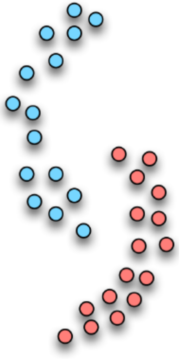
Single-Linkage

Distance between C and C' is $\min_{x \in C, x' \in C'} d(x, x')$



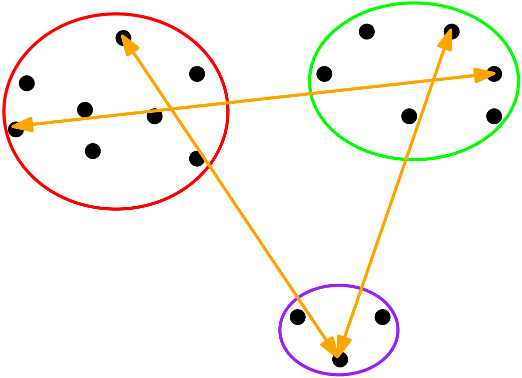
Single-Linkage

Retrieves clusters that are not convex



Complete-Linkage

Distance between C and C' is $\max_{x \in C, x' \in C'} d(x, x')$



Average-Linkage

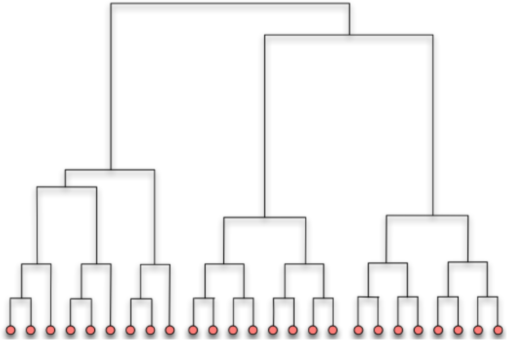
- A compromise between Single-Linkage and Complete-Linkage

Average-Linkage

- A compromise between Single-Linkage and Complete-Linkage

- Distance between C and C' is $\frac{1}{|C||C'|} \sum_{x \in C, x' \in C'} d(x, x')$

Dendograms



- A tree/hierarchical structure to represent HACs
- Vertical edge lengths represent the distance between two clusters

Outline

Quality of Hierarchical Clustering

- $\mathcal{C}_k$: merge until k clusters

Quality of Hierarchical Clustering

- $\mathcal{C}_k$: merge until k clusters
- Compare with k-clustering, e.g., k-center

Quality of Hierarchical Clustering

- $\mathcal{C}_k$: merge until k clusters
- Compare with k-clustering, e.g., k-center
- Single-Linkage $\mathcal{C}_k$ is k factor worse than optimal k-center
- Complete-Linkage and Average-Linkage $\mathcal{C}_k$ is $\log k$ factor worse than optimal k-center

A Gap Example for Single-Linkage

- Distance between points x_j and x_{j+1} is $1 - j\epsilon$

A Gap Example for Single-Linkage

- Distance between points x_j and x_{j+1} is $1 - j\epsilon$
- Single-Linkage merges clusters right to left:
  $\mathcal{C}_k = \{\{x_1\}, \{x_2\}, \ldots, \{x_{k-1}\}, \{x_k, \ldots, x_n\}\}$

A Gap Example for Single-Linkage

- Distance between points x_j and x_{j+1} is $1 - j\epsilon$
- Single-Linkage merges clusters right to left:
  $\mathcal{C}_k = \{\{x_1\}, \{x_2\}, \ldots, \{x_{k-1}\}, \{x_k, \ldots, x_n\}\}$
- Cost (or radius) of the supercluster:
  $\sum_{j=k}^{n-1}(1 - j\epsilon)/2 \geq (n - k - \epsilon\binom{n}{2})/2 \approx (n - k - 1)/2$

A Gap Example for Single-Linkage

- Distance between points x_j and x_{j+1} is $1 - j\epsilon$
- Single-Linkage merges clusters right to left:
  $\mathcal{C}_k = \{\{x_1\}, \{x_2\}, \ldots, \{x_{k-1}\}, \{x_k, \ldots, x_n\}\}$
- Cost (or radius) of the supercluster:
  $\sum_{j=k}^{n-1}(1 - j\epsilon)/2 \geq (n - k - \epsilon\binom{n}{2})/2 \approx (n - k - 1)/2$
- k-center OPT: $\approx n/2k$