(II) Hierarchical Methods

Hierarchical clustering method works by grouping data objects into a hierarchy or "tree" of clusters. This method is useful for data summarization & visualization that represent the data in compressed way

For eg:- organize the employees into major groups such as executives, managers & staff. Staff can be further divided into subgroups of senior officers, officers & trainees.

Hierarchical clustering methods :-

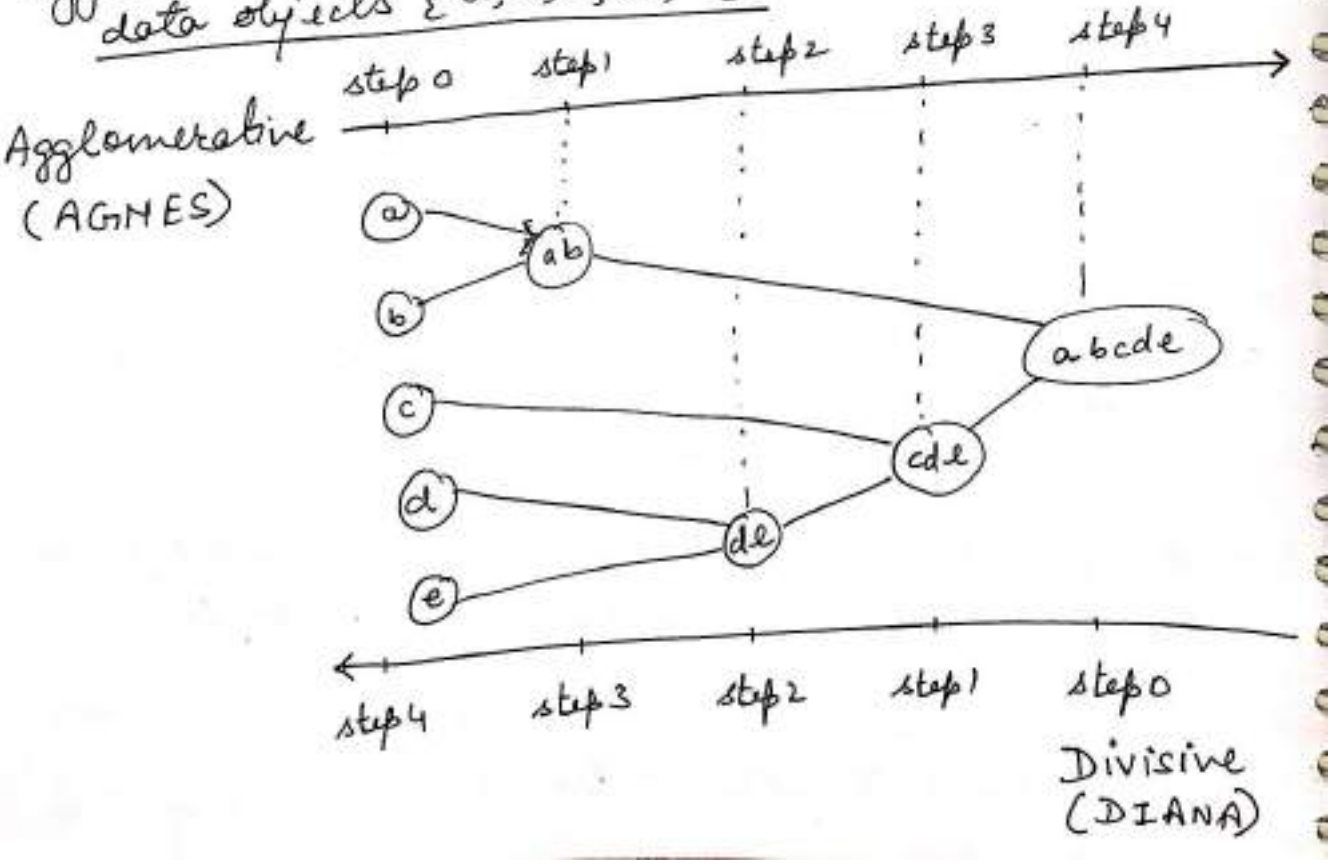
1) Agglomerative clustering method :-

- uses a bottom-up strategy.
- start with individual objects as clusters, which are iteratively merged to form larger clusters, until all objects are in a single cluster or certain termination conditions are satisfied.
- Single cluster becomes the hierarchy root.
- For the merging step, it finds two clusters that are closest to each other (acc. to some similarity measure) & combines the two to form one cluster.
- Because the two clusters are merged per iteration, where each cluster contain at least one object.
- The method requires at most n iterations.
- Application :- AGNES (AGglomerative NESting)

2) Divisive hierarchical clustering method :-

- It employs a Top down strategy.
- starts by placing all objects in one cluster, which is the hierarchy root. It then divides the root cluster into several smaller subclusters & recursively partitions those clusters into smaller ones.
- The partitioning process continues until each cluster at the lowest level is coherent enough - either containing only one object, or the objects within a cluster are sufficiently similar to each other.
- Application :- DIANA (DIvisive ANAlysis)

⇒ Agglomerative & Divisive hierarchical clustering on data objects {a, b, c, d, e}.



AGNES (Agglomerative method)

- Initially, AGNES, places each object into a cluster of its own.
- The clusters are then merged step-by-step acc. to some criterion.
- For eg:- clusters C_1 & C_2 may be merged if an object in C_1 & an object in C_2 form the minimum Euclidean distance between any two objects from different clusters.
- This is a single-linkage approach. In that each cluster is represented by all the objects in the cluster, & the similarity b/w the two clusters is measured by the similarity of the closest pair of data objects belonging to different clusters. Process continues until all the objects are merged to form one cluster.

DIANA (Divisive method)

- Proceeds in contrasting way.
- All the objects form one initial cluster.
- The cluster is split according to some principle such as maximum Euclidean distance b/w the closest neighbouring objects in the cluster.
- The cluster-splitting process repeats, until each new cluster contains only a single object.

hierarchical methods (categories).

- We can improve the clustering quality of hierarchical methods is to integrate hierarchical clustering with other clustering techniques, resulting in multiple-phase (or multiphase) clustering.

- Two such methods are :-

  (a). BIRCH (Balanced Iterative Reducing & clustering using hierarchies) :- first apply hierarchical clustering & then other methods on leaf & non-leaf.

  (b). Chameleon :- explores dynamic modeling in hierarchical structure.

- Hierarchical methods can also be categorized as :-

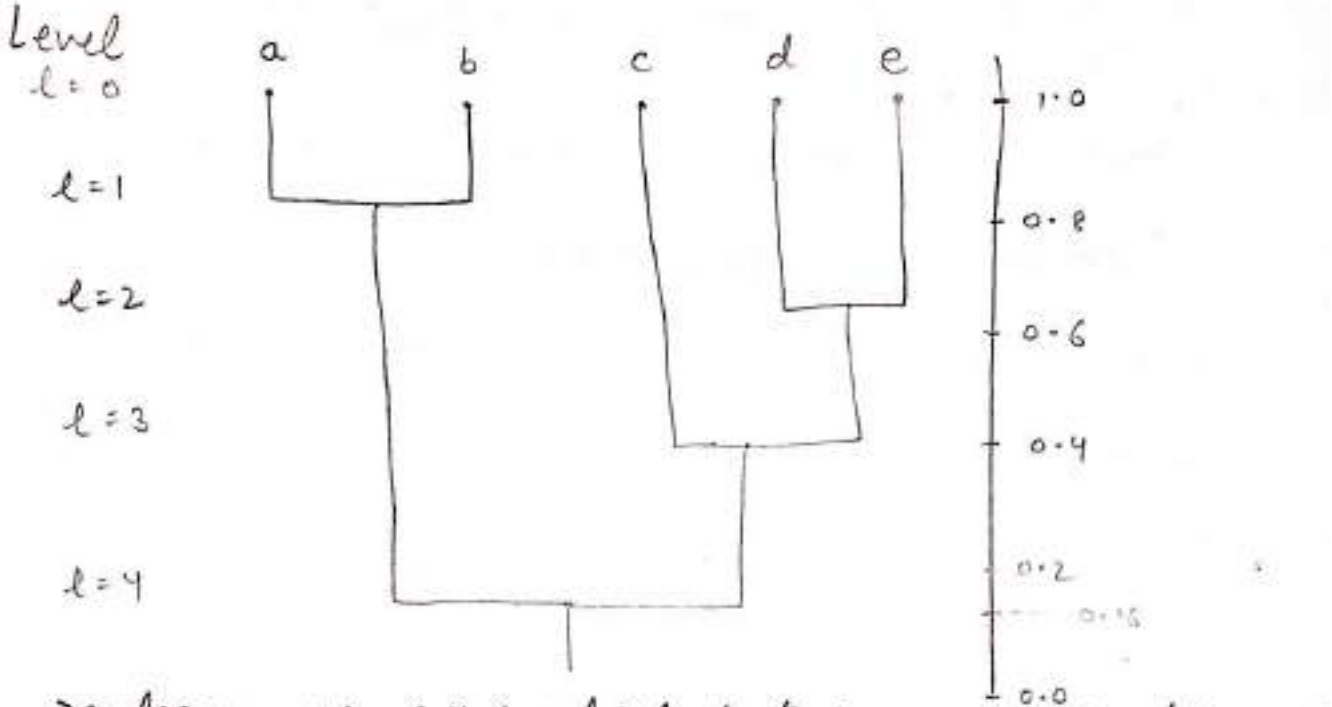
(a). Algorithmic methods (data objects are deterministic).
  - Agglomerative methods.
  - divisive methods.
  - multiphase methods.

(b). Probabilistic methods (measures quality of clusters by fitting of models).

(c) Bayesian methods. (computes distribution of possible clusterings).

Example of tree structure - dendrogram

- It shows how objects are grouped together (in an agglomerative method) or partitioned (in a divisive method) step-by-step.



Dendogram rep. of Heirarchical clustering of data objects {a, b, c, d, e}

- l = 0 shows the five objects as singleton clusters at level 0.
- l = 1, objects 'a' & 'b' are grouped together to form the first cluster & they stay together at all subsequent levels.
- We use a vertical axis to show the similarity scale b/w clusters. For eg., when the similarity of two groups of objects {a, b} & {c, d, e} is roughly 0.16, they are merged together to form a single cluster.

Disadvantages of Divisive methods

- Challenge with Divisive method is how to partition large cluster into several smaller ones.
- There are $2^{n-1} - 1$ possible ways to partition a set of 'n' objects into two exclusive subsets.
- when 'n' is large, it is computationally prohibitive to examine all possibilities.
- Consequently, a divisive method typically uses heuristics in partitioning, which can lead to inaccurate results.
- for the sake of efficiency, divisive methods do not backtrack on partitioning decisions that have been made.

Thus agglomerative methods are preferred over divisive methods.

Distance Measures in Algorithmic Methods

- In Algorithmic methods, a core need is to measure the distance between two clusters, where each cluster is a set of objects.
- Four widely used measures for distance between clusters are as follows where
  - $|p - p'|$ is the distance b/w two objects or poin
  - m_i is mean for cluster C_i
  - n_i is the no. of objects in C_i
- Also known as *linkage measures*.

(1) minimum distance :-

$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\} \quad -(1)$$

(2) maximum distance :

$$dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\} \quad -(2)$$

(3) Mean distance :

$$dist_{mean}(C_i, C_j) = |m_i - m_j| \quad -(3)$$

(4) Average distance :

$$dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{\substack{p \in C_i, \\ p' \in C_j}} |p - p'| \quad -(4)$$

Minimum Distance :- If $dist_{min}(C_i, C_j)$ measure is used, then algorithm is also called nearest-neighbor clustering algorithm.

- If the clustering process is terminated when the distance between nearest clusters exceeds a user-defined threshold, it is called single linkage algorithm.

- If we view datapoint as nodes of a graph, with edges forming a path b/w the nodes in a cluster, then the merging of two clusters, C_i & C_j means adding an edge b/w the nearest pair of nodes in C_i & C_j.

- Because edges linking clusters always go between distinct clusters, the resulting graph will generally a tree.

- An Agglomerative hierarchical clustering algorithm that uses the minimum distance measure is also called a minimal spanning tree Algorithm.

Maximum distance

- when an Algorithm uses the maximum distance, $dist_{max}(C_i, C_j)$ to measure the distance between clusters, it is also called farthest-neighbor clustering Algorithm.
- If the clustering process is terminated when the maximum distance between nearest clusters exceeds a user-defined threshold, it is called a complete-linkage-algorithm.
- By viewing data points as a nodes of a graph, with edges linking nodes, we can think of each cluster as a complete subgraph, ie, with edges connecting all the nodes in the clusters.

The distance b/w two cluster is determined by the most distant nodes in the two clusters.

distance b/w two clusters can also be the distance b/w centroids.

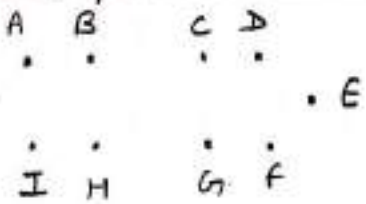
Mean & Average distance

The use of mean or average distance is a compromise b/w the minimum & maximum distance & overcomes the outlier sensitivity problem.

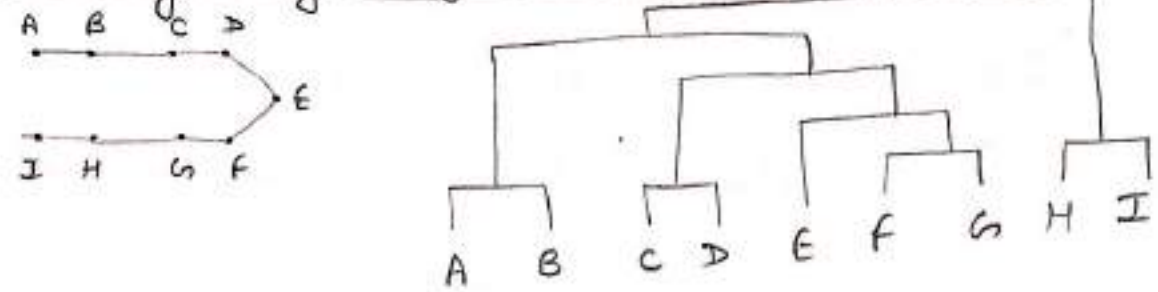
- mean distance is simplest to compute. (cannot compute mean for categoric data.)
- Average distance can handle categoric as well as numeric data.

Single versus complete linkages

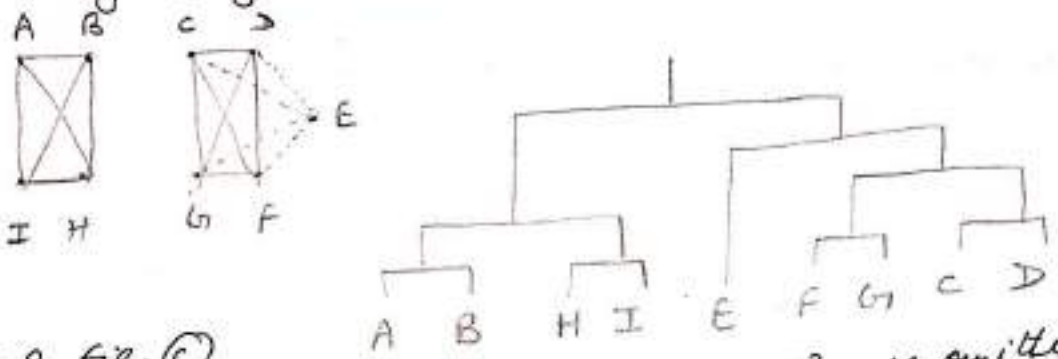
(a) Dataset



(b) clustering using single linkage



(c) clustering using complete linkage.



[In fig-(c), # Edges b/w {A, B, I, H} & {C, D, G, F, E} are omitted for ease of presentation.]

using single linkages we can find hierarchical clusters defined by local proximity, whereas complete linkage tends to find clusters opting for global closeness.

ch-14. Hierarchical clustering

- Create sequence of nested partitions which can be visualize as a tree or hierarchy of clusters, also called cluster dendrogram.
- from fine grained (lowest level) to coarse grained (highest level)
- meaningful clustering at intermediate levels.
- Two approaches
  - → Agglomerative (Bottom up).
  - → Divisive (Top down manner).

- Partition dataset $D = \{x_1, x_2, \ldots x_n\}$ where $x_i \in R^d$, into k clusters $C = \{C_1, C_2, \ldots, C_k\}$ where $C_i \in D$ such that $C_i \cap C_j = \phi \ (i \neq j)$ & $\bigcup_{i=1}^{k} C_i = D$

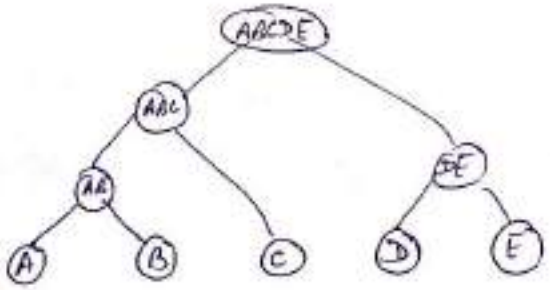
- A clustering $A = \{A_1, A_2, \ldots, A_r\}$ is said to be nested in another clustering $B = \{B_1, B_2, \ldots B_s\}$ iff $r > s$. $\{A_i \in A, B_j \in B\}$ for each cluster A_i there exist a cluster B_j such that $A_i \subseteq B_j$

$D = \{ABCDE\}$

$C = \{ABC, DE\}$

$B = \{AB, C, DE\}$

$A = \{A, B, C, D, E\}$
or
$A = \{A_1, A_2, A_3, A_4, A_5\}$



Here A is nested in B because $5 > 3$.

For each cluster A_i there there exist a cluster AB such that $A \subseteq AB$.

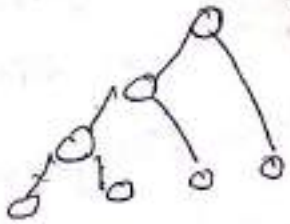
Number of Hierarchical clustering

means of different binary rooted trees or dendrograms with n leaves having distinct labels.

A rooted tree with 'n' leaves has

- m-1 internal nodes
- m + m-1 total no. of nodes. (2m-1)
- 2m-1 - 1 = 2m-2 no. of edges.

m = 4

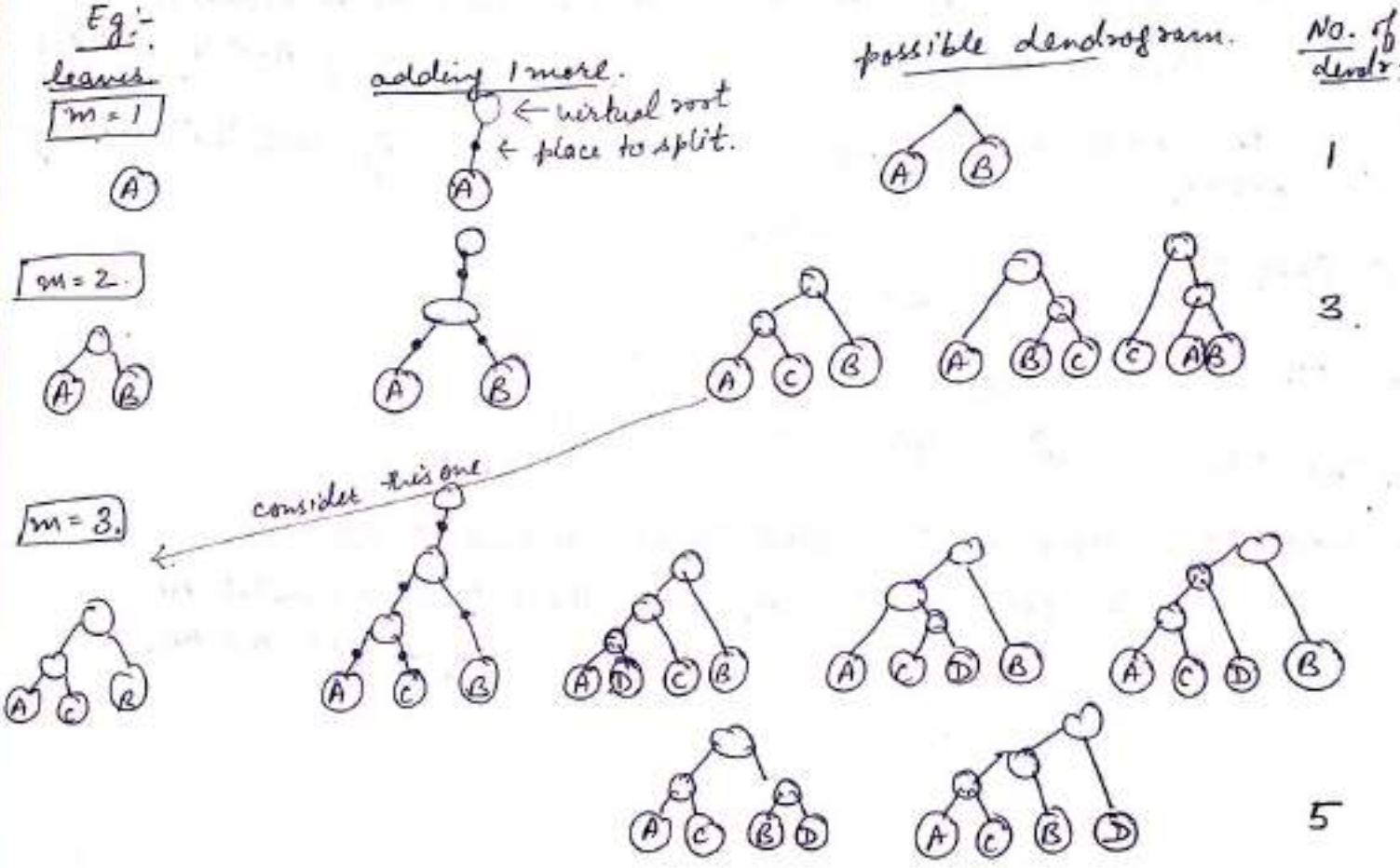


- leaves = 4(m)
- Internal nodes = 3. (m-1)
- 4 + 3 = 7 (total nod
- edges = 2.4 - 2 = 6.

To count no. of possible dendrograms, extend leaves from 'm' to 'm+1'.

- We can add the extra leaf by splitting / branching any of 2m-2 edges.

Eg:-



similarly for other two ∴ Total dendrograms possible are 5+5+5 = 15 by adding 1 leaf to existing 3 leaves

* we can also add new leaf as a child to the new root it
$2m-2+1 = 2m-1$ new dendrograms with $m+1$ leaves.

The total no. of different dendrograms with n leaves is given by:

$$\prod_{m=1}^{n-1} (2m-1) = 1 \times 3 \times 5 \times \ldots \times (2n-3) = (2n-3)!!$$

AGGLOMERATIVE hierarchical clustering

- Begin with each of 'n' points in a separate cluster.
- Repeatedly merge two closest clusters until all points are members of the same cluster.
- given set of clusters $\mathcal{C} = \{C_1, \ldots C_m\}$
  find closest pair of clusters C_i & C_j,
  merge them into a new cluster $C_{ij} = C_i \cup C_j$,
  update the set of clusters by removing C_i & C_j & adding C_{ij} as-

$$\mathcal{C} = (\mathcal{C} \setminus \{C_i, C_j\}) \cup \{C_{ij}\}$$

- Repeat it until $\mathcal{C}$ has only one cluster.

- Agglomerative Clustering (D, k):

1. $\mathcal{C} \leftarrow \{C_i = \{x_i\} \mid x_i \in D\}$ // each point in separate cluster.
2. $\Delta \leftarrow \{\delta(x_i, x_j) : x_i, x_j \in D\}$ // compute distance matrix
3. repeat
4. find closest pair of clusters $C_i, C_j \in \mathcal{C}$.
5. $C_{ij} \leftarrow C_i \cup C_j$ // merge the clusters.
6. $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{C_i, C_j\}) \cup \{C_{ij}\}$ // update the clustering
7. update distance matrix Δ to reflect new clustering.
8. until $|\mathcal{C}| = k$.

Distance between clusters

- Determine the closest pair of clusters.
- Compute distance b/w two points using Euc distance / L_2-norm.

$$\delta(x,y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

- Distance measures

(1) Single Link :- minimum distance b/w a point in C_i & a point in C_j.

$$\delta(C_i, C_j) = \min\{\delta(x,y) \mid x \in C_i, y \in C_j\}$$

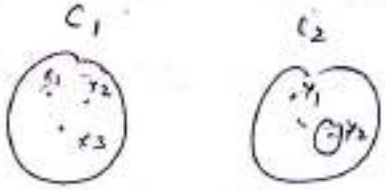
(2) Complete link :- maximum distance b/w a point in C_i & a point in C_j.

$$\delta(C_i, C_j) = \max\{\delta(x,y) \mid x \in C_i, y \in C_j\}$$

(3) Group Average :- average pairwise distance b/w points in C_i & C_j.

$$\delta(C_i, C_j) = \frac{\sum_{x \in C_i} \sum_{y \in C_j} \delta(x,y)}{n_i \cdot n_j}$$

eg :-



$$\delta(C_1, C_2) = \frac{\sum_{x \in C_i} \left(\delta(x,y_1) + \delta(x,y_2)\right)}{3 \cdot 2}$$

$$\delta(C_1, C_2) = \delta(x_1,y_1) + \delta(x_2,y_1) + \delta(x_3,y_1) + \delta(x_1,y_2) + \delta(x_2,y_2) + \delta(x_3,y_2)$$

$$= \frac{6}{[3*2]}.$$

where $n_i = |C_i|$

no. of points in cluster C_i.

Mean Distance :- Distance b/w the means or centroids of two clusters.

$$\delta(C_i, C_j) = \delta(\mu_i, \mu_j) \quad \text{where } \mu_i = \frac{1}{n_i} \sum_{x \in C_i} x.$$

(5) **Minimum Variance** :- (Ward's method) :-

Distance b/w two clusters is defined as increase in the SSE when two clusters are merged.

SSE for a given cluster C_i is given as -

$$SSE_i = \sum_{x \in C_i} \|x - \mu_i\|^2$$

$$= \sum_{x \in C_i} \left(x^T.x - 2x^T.\mu_i + \mu_i^T.\mu_i\right)$$

$$= \sum_{x \in C_i} x^T.x - 2n_i\mu_i \frac{\sum_{x \in C_i} x^T}{n_i} + \mu_i^T \sum_{x \in C_i} .1$$

$$= \sum_{x \in C_i} x^T.x - 2n_i\mu_i.\mu^T + \mu_i^T.\mu_i.n_i$$

$$= \sum_{x \in C_i} x^T.x - n_i\mu_i^T.\mu_i \quad \text{[This is for single cluster 'i']} \quad \text{---(1)}$$

The SSE for a clustering $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ is given as

$$SSE = \sum_{i=1}^{m} SSE_i = \sum_{i=1}^{m} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Ward's measure defines the distance b/w two clusters C_i & C_j as the net change in SSE value b/w SSE_{ij} & $(SSE_i$ & $SSE_j)$.

$$\therefore \delta(C_i, C_j) = \Delta SSE_{ij} = SSE_{ij} - SSE_i - SSE_j. \quad \text{---(2)}$$

Also, $C_{ij} = C_i \cup C_j$

$C_i \cap C_j \neq \phi$,

$\therefore |C_{ij}| = n_{ij} = |n_i| + |n_j|$.

updating Distance Matrix

- merge C_i & $C_j \longrightarrow C_{ij}$
- update distance of C_{ij} to all other clusters C_γ ($\gamma \neq i, \gamma \neq j$).
- Lance William eq. to recompute the distances for all of the cluster proximity measures is given as :-

$$\delta(C_{ij}, C_\gamma) = \alpha_i \delta(C_i, C_\gamma) + \alpha_j \delta(C_j, C_\gamma) + \beta \delta(C_i, C_j) + \gamma |\delta(C_i, C_\gamma) - \delta(C_j, C_\gamma)|$$

Measure	α_i	α_j	β	γ
single link	$1/2$	$1/2$	0	$-1/2$
complete link	$1/2$	$1/2$	0	$1/2$
group Avg.	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
mean distance	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i n_j}{(n_i+n_j)^2}$	0
ward's measure	$\frac{n_i+n_\gamma}{n_i+n_j+n_\gamma}$	$\frac{n_j+n_\gamma}{n_i+n_j+n_\gamma}$	$\frac{-n_\gamma}{n_i+n_j+n_\gamma}$	0

Complexity.

- $O(n^2)$ time to create pairwise distance matrix (unless specified as I/p)
- At merge step, distance from merged cluster to all other clusters is recomputed. ie at step 't', $O(n-t)$ distances are computed.
- To find closest pair in matrix, keep n^2 distances in min heaps. Thus $O(1)$ time to find closest pair.
- Creation of heap takes $O(n^2)$.
- Deleting / updating distances from heap takes $O(\log n)$ for each operation for a total time across all merge steps of $O(n^2 \log n)$.
- Computational complexity of hierarchical clustering is $O(n^2 \log n)$.