Hierarchical Clustering

- It produces a hierarchy (dendogram) of nested clusters that can be visualized / analyzed
- It produces a set of possible solutions

Clustering	Partition
C_1	{A} {B} {C} {D} {E}
C_2	{A, B} {C} {D} {E}
C_3	{A, B} {C} {D, E}
C_4	{A, B} {C, D, E}
C_5	{A, B, C, D, E}

- Strength : No need to assume any particular # of clusters
- Two approaches: » Agglomerative » Divisive

1) Divisive Hierarchical Clustering:

- Start with one cluster containing all data points → at each step, split a cluster contains a point (or k clusters)

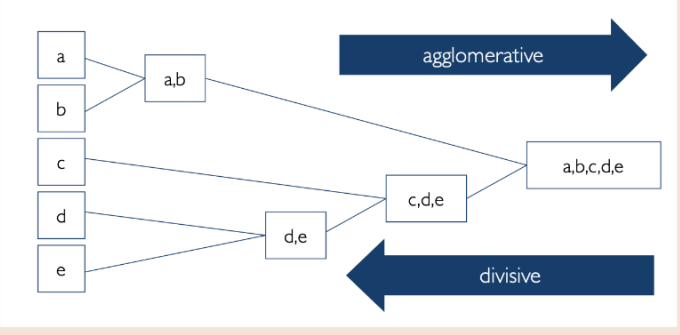
2) Agglomerative Hierarchical Clustering:

- Start with one cluster for each item → then each step merge similar clusters, until generate one (or k) cluster

Algorithm 14.1: Agglomerative Hierarchical Clustering Algorithm

AGGLOMERATIVECLUSTERING($\mathbf{D}, k$):

```
C ← {C_i = {x_i} | x_i ∈ D} // Each point in separate cluster
Δ ← {||x_i − x_j|| : x_i, x_j ∈ D} // Compute distance matrix
repeat
    Find the closest pair of clusters C_i, C_j ∈ C
    C_ij ← C_i ∪ C_j // Merge the clusters
    C ← (C \ {C_i, C_j}) ∪ {C_ij} // Update the clustering
    Update distance matrix Δ to reflect new clustering
until |C| = k
```



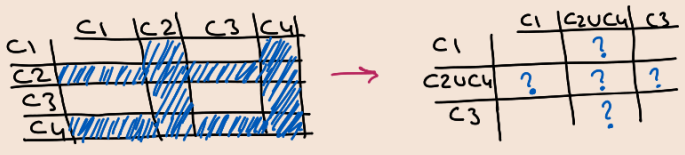
- Distance between clusters;

① Start with cluster of individual points and distance matrix

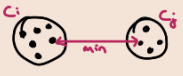
② After some merging step, we have some clusters

③ We decide to merge two closest clusters C2 and C4 → now need to update distance matrix

④ How to update the distance matrix?



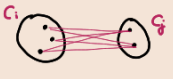
- Single Linkage: $\delta(C_i, C_j) = \min\{\|\mathbf{x}-\mathbf{y}\| \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$



- Complete Linkage: $\delta(C_i, C_j) = \max\{\|\mathbf{x}-\mathbf{y}\| \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$

- Mean / Centroid Linkage: $\delta(C_i, C_j) = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$

- Group Average: $\delta(C_i, C_j) = \frac{\sum_{\mathbf{x} \in C_i} \sum_{\mathbf{y} \in C_j} \|\mathbf{x}-\mathbf{y}\|}{n_i \cdot n_j}$



- Minimum Variance / Wards: $\delta(C_i, C_j) = \left(\frac{n_i n_j}{n_i + n_j}\right) \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2$ $= \nabla SSE_{ij} = SSE_{ij} - SSE_i - SSE_j$ ($\sum \|x - \mu_{ij}\|^2$)

- Lance-Williams Formula: $\delta(C_{ij}, C_r) = \alpha_i \cdot \delta(C_i, C_r) + \alpha_j \cdot \delta(C_j, C_r) + \beta \cdot \delta(C_i, C_j) + \gamma \cdot |\delta(C_i, C_r) - \delta(C_j, C_r)|$

↳ Whenever two clusters C_i and C_j merged into C_{ij}, update distance matrix by recomputing the distances C_{ij} to all other clusters C_r ($r \neq i, j$)

Measure	α_i	α_j	β	γ
Single link	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Mean distance	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i \cdot n_j}{(n_i+n_j)^2}$	0
Ward's measure	$\frac{n_i+n_r}{n_i+n_j+n_r}$	$\frac{n_j+n_r}{n_i+n_j+n_r}$	$\frac{-n_r}{n_i+n_j+n_r}$	0

- Ex:

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

a and b closer

	a,b	c	d	e
a,b	0			
c	5	0		
d	9	4	0	
e	8	5	3	0

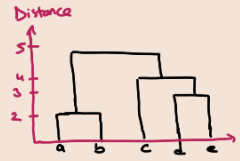
to update distances we used single-linkage between (a,b) and others

$d_{(1,2)3} = \min[d_{13}, d_{23}] = d_{23} = 5$

$d_{(1,2)4} = \min[d_{14}, d_{24}] = d_{24} = 9$

$d_{(1,2)5} = \min[d_{15}, d_{25}] = d_{25} = 8$

continue process until reaching one cluster



- Clustering Quality Measures

Internal Validation Measures

- Employ criteria that derived from data itself
- For instance; intracluster and intercluster distances to measure cohesion and separation

External Validation Measures

- Use prior or expert-specified info about the clusters.
- For ex: iris dataset; evaluate the clustering using known class labels
- Employ criteria that are not inherent to the dataset (do cosmic var clean)

Relative Validation Measures

- Aim to directly compare different solutions (obtained via different parameter setting for the same algorithm.)

Cohesion: How similar are the points in same cluster?

$$WSS(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, \mu_i)^2$$

(within-cluster sum of squares; μ_i: centroid of C_i (in Euclidean space))

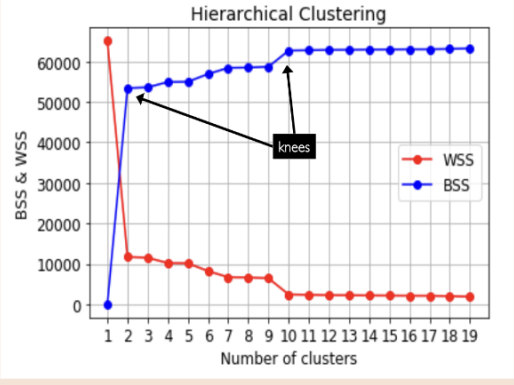
Separation: How far apart are points in different clusters?

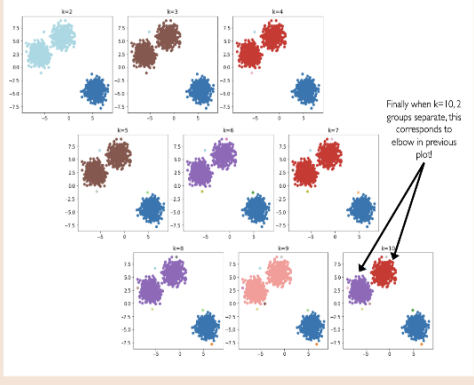
$$BSS(C) = \sum_{i=1}^{k} |C_i| \, d(\mu, \mu_i)^2$$

(between-cluster sum of squares; μ: centroid of whole dataset)

» Evaluation of Hierarchical Clustering using Knee/Elbow Analysis

- Plot BSS and WSS for every clustering
- Look for a knee/elbow in the plot that show a significant variation of the evaluation metrics





We should use plots as guides but also evaluate multiple solutions around knees/elbows because; it is limited, there is no single best criteria to select # of clusters

Apply more methods, look for an aggreement

↳ if methods disagree and do not provide clear indications none of result is good

»Clustering Evaluation Metrics:

- Purity
- Variation of information
- Pairwise Measures
- Dunn Index
- Silhouette Coefficient

Algorithm 17.1: Clustering Stability Algorithm for Choosing k

CLUSTERINGSTABILITY $(A, t, k^{max}, \mathbf{D})$:

$n \leftarrow |\mathbf{D}|$

// Generate t samples — generate t bootstrap samples

for $i = 1, 2, \ldots, t$ **do**

  $\mathbf{D}_i \leftarrow$ sample n points from $\mathbf{D}$ with replacement

// Generate clusterings for different values of k — Apply clustering algorithm for different values of k

for $i = 1, 2, \ldots, t$ **do**

  for $k = 2, 3, \ldots, k^{max}$ **do**

    $\mathcal{C}_k(\mathbf{D}_i) \leftarrow$ cluster $\mathbf{D}_i$ into k clusters using algorithm A

// Compute mean difference between clusterings for each k — Computes distance between clusterings for each pair of datasets D_i and D_j for each k

foreach *pair* $\mathbf{D}_i, \mathbf{D}_j$ *with* $j > i$ **do**

  $\mathbf{D}_{ij} \leftarrow \mathbf{D}_i \cap \mathbf{D}_j$ // create common dataset using Eq. (17.52)

  for $k = 2, 3, \ldots, k^{max}$ **do**

    $d_{ij}(k) \leftarrow d\big(\mathcal{C}_k(\mathbf{D}_i), \mathcal{C}_k(\mathbf{D}_j), \mathbf{D}_{ij}\big)$ // distance between clusterings

for $k = 2, 3, \ldots, k^{max}$ **do** — Computes expected pairwise distance $\mu_d(k)$ are select k corresponding to minimum $\mu_d(k)$

  $\mu_d(k) \leftarrow \frac{2}{t(t-1)} \sum_{i=1}^{t} \sum_{j>i} d_{ij}(k)$ // expected pairwise distance

// Choose best k

$k^* \leftarrow \arg\min_k \{\mu_d(k)\}$

- Cluster Stability: Clustering obtained from datasets sampled from same distribution should be "stable"

It can be used to find good parameters for given clustering algorithm