HIERARCHICAL CLUSTERING

→ In K-means, selecting K is a question of granularity
- ↳ How coarse / fine-grained is the structure in your data?

→ Instead of picking K, we find a hierarchy of structures
- ↳ Topmost cluster: contains every pt. in the dataset
- ↳ More, smaller clusters as you go down the hierarchy
- ↳ Bottom level: set of n singleton clusters

→ Two strategies:

1) Top-down approach [DIVISIVE CLUSTERING]

↳ i) Run K-means algo. on the original data $x_1 \dots x_n$

ii) For each of the resulting clusters C_i, $i = 1..K$, recursively run K-means on points in C_i

(+): Fast bc. recursive calls operate on a slice → $O(Knd \log_K n)$

(−): Greedy: Nearby pts. may end up in diff. clusters bc. they can't cross boundaries imposed by top levels

2) Bottom-up approach [AGGLOMERATIVE CLUSTERING]

↳ The idea is to ensure nearby pts. end up in the same cluster

i.) Start w/ a collection C of n singleton clusters

ii.) Repeat until only 1 cluster is left:

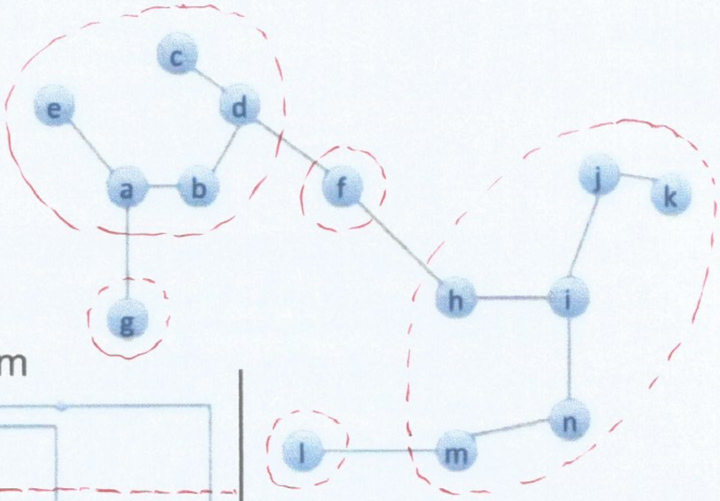
need to define a distance metric over clusters, not individual points

- Find a pair of clusters that is closest → $\min_{i,j} D(c_i, c_j)$
- Merge the clusters c_i and c_j into a new cluster c_{i+j}
- Remove c_i, c_j from the collection C and add c_{i+j}

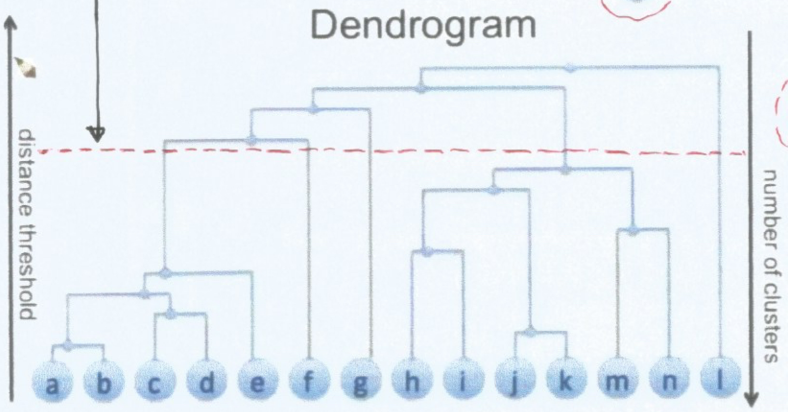
↳ (−): Slow: $O(n^2 d + n^3)$

↳ Produces a dendrogram – hierarchical tree of clusters

AGGLOMERATIVE CLUSTERING e.g.



If you want to turn this into flat clustering, find a distance threshold at which you're going to cut the Dendrogram.



↳ Can use scree plot trick to see how many clusters we should use

CLUSTER DISTANCE MEASURES

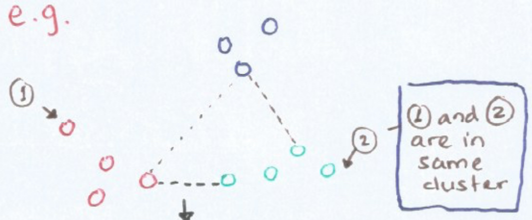
1) SINGLE LINK

$$\boxed{D(c_1, c_2) = \min_{x_1 \in c_1,\, x_2 \in c_2} D(x_1, x_2)}$$

↳ Distance between closest elements in clusters

↳ Merge clusters w/ shortest distance

e.g.



Shortest distance; Merge red and green clusters

(−): Produces long chains as clusters (instead of 'spherical clusters) [See brown arrow in e.g.]

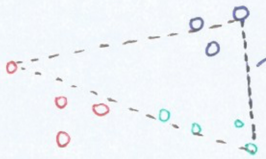
2) COMPLETE LINK

$$\boxed{D(c_1, c_2) = \max_{x_1 \in c_1,\, x_2 \in c_2} D(x_1, x_2)}$$

↳ Distance between farthest elements in clusters

↳ Pretends to merge the two clusters and ask the qn 'What is the resulting diameter?'

↳ Merge clusters w/ smallest resulting diameter



Smallest resulting diameter

Merge blue & green clusters

(+): Forces 'spherical' clusters

3) AVERAGE LINK

$$\boxed{D(C_1, C_2) = \frac{1}{|C_1|} \frac{1}{|C_2|} \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} D(x_1, x_2)}$$

↳ Avg. of all pairwise distances

↳ Less affected by outliers

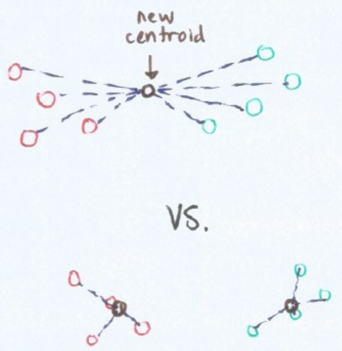
4) CENTROIDS

$$\boxed{D(C_1, C_2) = D\left(\left(\frac{1}{|C_1|} \sum_{x \in C_1} \vec{x}\right), \left(\frac{1}{|C_2|} \sum_{x \in C_2} \vec{x}\right)\right)}$$

↳ Distance between centroids of 2 clusters

5) WARD'S METHOD

$$\boxed{TD_{C_1 \cup C_2} = \sum_{x \in C_1 \cup C_2} D(x, \mu_{C_1 \cup C_2})^2}$$



↳ Pretends to merge two clusters, estimate a centroid for the resulting cluster and look into the sum of the squared deviations of all the points from the new centroid

↳ Pick the merge that results in smallest deviation

LANCE-WILLIAMS ALGO.

→ Regardless of the distance measure you use, it can be implemented as the same algorithm w/ same complexity

```
D = { D_{i,j} : distance between x_i and x_j for i,j = 1..N }
For N iterations :
    i, j = argmin D_{i,j}        ↗ Look for pair of closest clusters
    add cluster i+j
    delete clusters i, j
    For each remaining clusters k :
        D_{k, i+j} = α_i D_{k,i} + α_j D_{k,j} + β D_{i,j} + γ |D_{k,i} − D_{k,j}|
```

$$D_{k,\, i+j} = \alpha_i D_{k,i} + \alpha_j D_{k,j} + \beta D_{i,j} + \gamma \,|D_{k,i} - D_{k,j}|$$

For diff. settings of constants, we are able to implement any distance measures

Method	α_i	α_i	β	γ
Single linkage	0.5	0.5	0	-0.5
Complete linkage	0.5	0.5	0	0.5
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted group average	0.5	0.5	0	0
Centroid	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i \cdot n_j}{(n_i+n_j)^2}$	0
Ward	$\frac{n_i+n_k}{(n_i+n_j+n_k)}$	$\frac{n_j+n_k}{(n_i+n_j+n_k)}$	$\frac{-n_k}{(n_i+n_j+n_k)}$	0

Look at single linkage method:

$$D_{k,\, i+j} = \frac{1}{2}\left(D_{k,i} + D_{k,j} - |D_{k,i} - D_{k,j}| \right)$$

$$= \min \{ D_{k,i},\ D_{k,j} \}$$

↳ In general,

$$\min_{a,b} = \max_{a,b} - |a-b|$$

Say $D_{k,i} > D_{k,j}$, $|D_{k,i} - D_{k,j}|$ is the difference between the big no. & small no. Notice that

$$D_{k,i} - |D_{k,i} - D_{k,j}| = D_{k,j}$$

Hence, $\frac{1}{2}(2D_{k,j}) = \underline{D_{k,j}}$ ✓ The smaller number