The K-means optimisation step makes sense:

(Step 1+2) Given clustering $C = \{C_\ell\}_{\ell=1}^{K}$

$$W = \frac{1}{2} \sum_{\ell=1}^{K} \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \| \vec{x}^{(i)} - \vec{x}^{(j)} \|^2$$

with centroids $\vec{m}_\ell = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} \vec{x}^{(i)}$ $\quad \ell = 1, \dots, K$

$$W = \frac{1}{2} \sum_{\ell=1}^{K} \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \| (\vec{x}^{(i)} - \vec{m}_\ell) - (\vec{x}^{(j)} - \vec{m}_\ell) \|^2$$

$$= \sum_{\ell=1}^{K} \frac{1}{2} \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \Big[\| \vec{x}^{(i)} - \vec{m}_\ell \|^2 + \| \vec{x}^{(j)} - \vec{m}_\ell \|^2 - 2 (\vec{x}^{(i)} - \vec{m}_\ell) \cdot (\vec{x}^{(j)} - \vec{m}_\ell) \Big]$$

$$= \sum_{\ell=1}^{K} \sum_{i \in C_\ell} \Big[\| \vec{x}^{(i)} - \vec{m}_\ell \|^2 - (\vec{x}^{(i)} - \vec{m}_\ell) \cancel{\underbrace{\frac{1}{|C_\ell|} \sum_{j \in C_\ell} (\vec{x}^{(j)} - \vec{m}_\ell)}_{= 0}} \Big]$$

In summary:

$$W = \frac{1}{2} \sum_{\ell=1}^{K} \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \| \vec{x}^{(i)} - \vec{x}^{(j)} \|^2$$

$$= \sum_{\ell=1}^{K} \sum_{i \in C_\ell} \| \vec{x}^{(i)} - \vec{m}_\ell \|^2$$

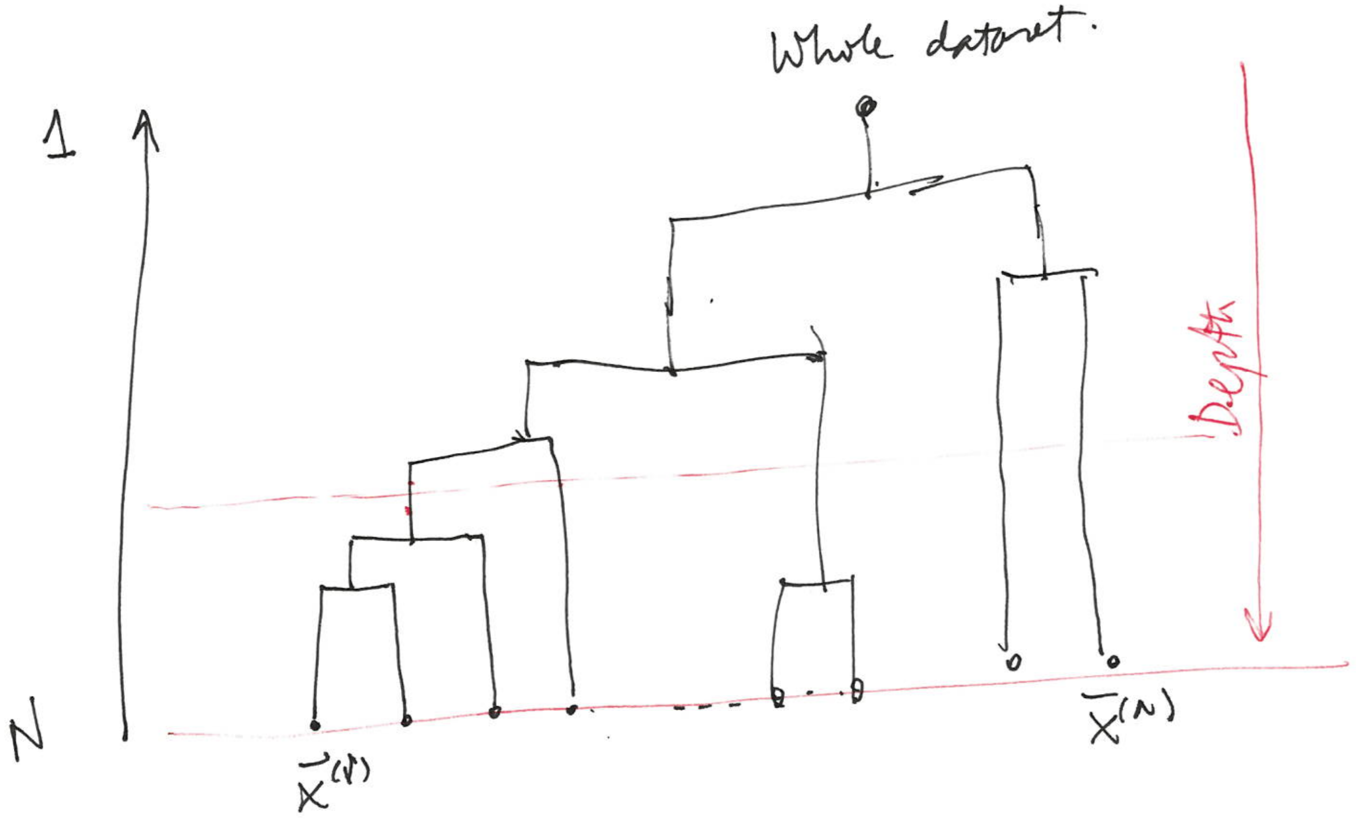
So trying to minimize the distance to the centroids is our objective.

(2) Hierarchical clustering

$$\{\vec{x}^{(i)}\} \qquad i=1,\dots,N$$

$$D_{N\times N} = (D_{ij}) \qquad D_{ij} = d(\vec{x}^{(i)}, \vec{x}^{(j)})$$

Represents the whole data set as a binary tree ('dendrogram') where leaves are samples and the root is the whole dataset.



- Hierarchical structure imposed on the data because of the requirement of binary splits.

- Monotonicity relating ~~depth~~ height and dissimilarity within cluster.

Agglomerative schemes:

Reduce the number of clusters 1 by 1 from N to 1.

(1) Simple linkage (SL):

Two clusters G, H : $$d_{SL}(G,H) = \min_{\substack{i \in G \\ j \in H}} D_{ij}$$

Nearest neighbour

(2) Complete linkage (CL):

Furthest neighbour: $$d_{CL}(G,H) = \max_{\substack{i \in G \\ j \in H}} D_{ij}$$

(3) Group average:

$$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{\substack{i \in G \\ j \in H}} D_{ij}$$

Divisive:

Recursive K-means with $\underline{K=2}$ gives a binary tree also <u>but</u>:

(1) monotonicity is not preserved

(2) it depends on the initialisation at every split.

So it can be quite variable.

Better schemes exist but, in general, agglomerative algorithms are used more extensively.