Heirarchical Clustering & DBSCAN

[Clustering]

→ 2 new ideas !!

→ Customer segmentation case study.

1st big idea:

↳ Use geometric intuition to design a metric that can quantify clusters.

→ Dunn, WCSS, Silloette ✓

↳ Use an optimisation setup to opt above metric ✗

2nd big idea

↳ Randomly initialise centers
→ assign clusters

→ update centers

→ repeat !

→ Kmeans ✓ Lloyd's algo.

Extention

↳ smart initialisation

→ Kmeans ++

→ code

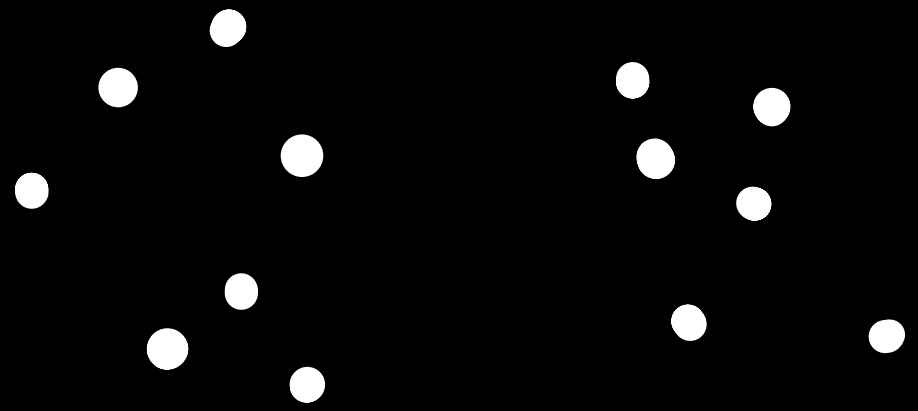
Next ideas :

→ Heirarchical clustering

→ DBSCAN

→ GMM

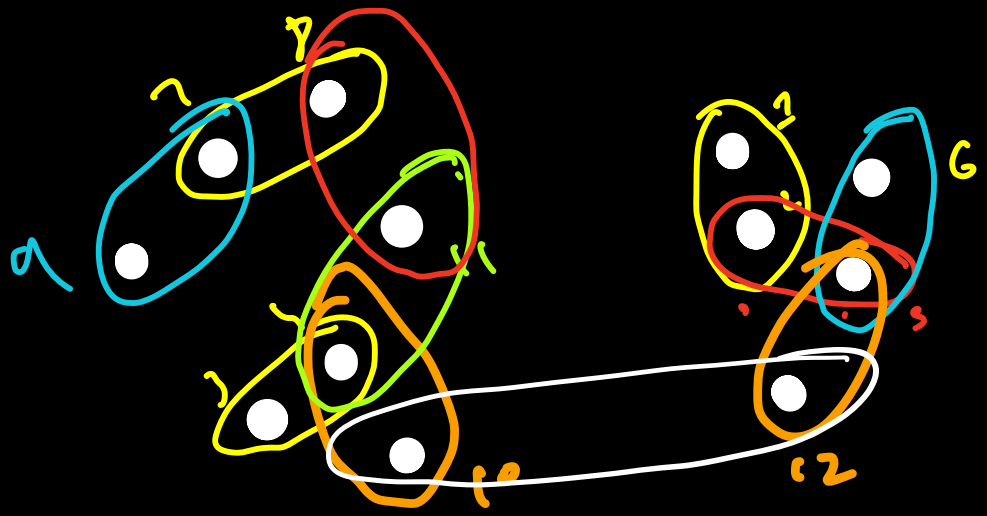
Heirarchical Clustering

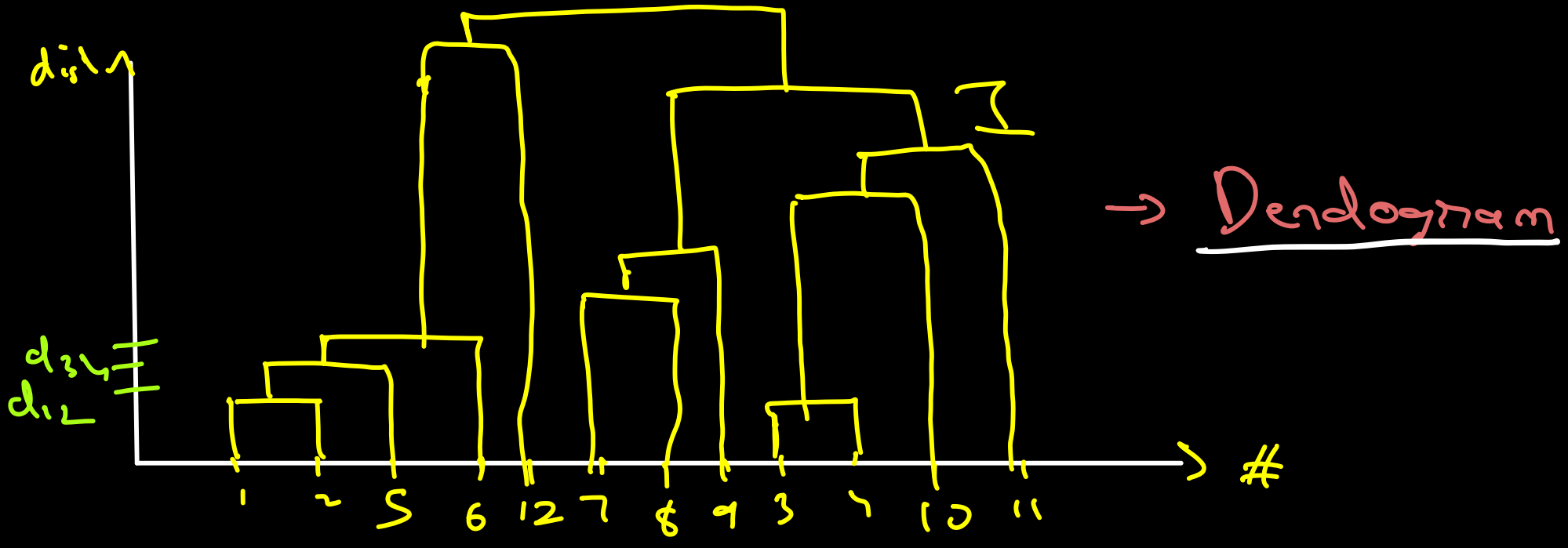


→ Compute some form of distance b/w all points.

→ Combine 2 pts with min distance

→ treat the sub-clusters as pts and repeat till only one cluster remains

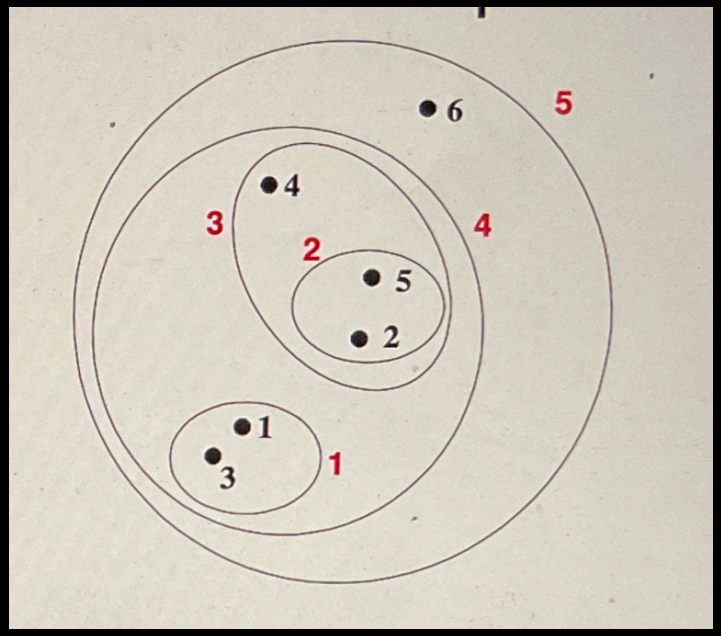


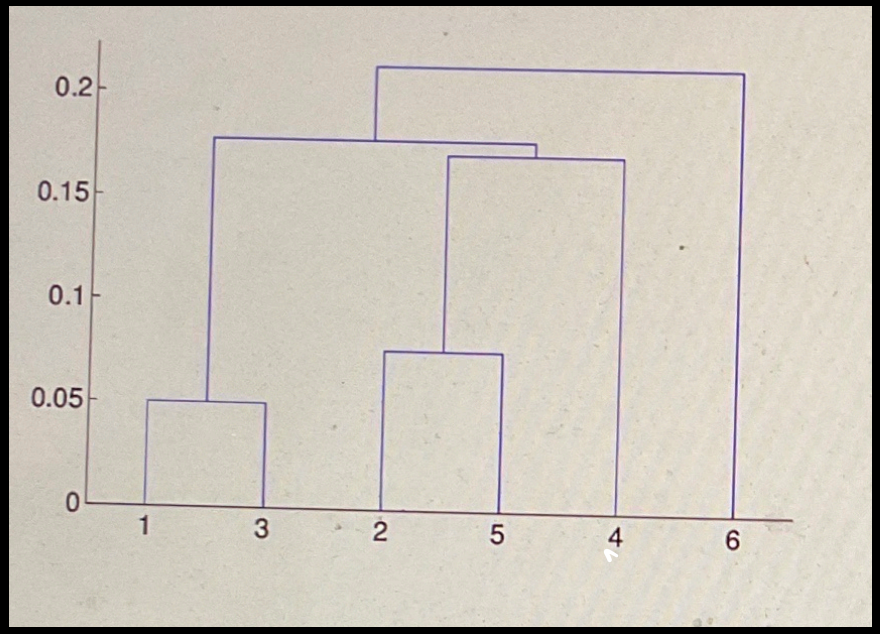


→ Dendogram

(another name)

→ Agglomerative clustering:





points

dendogram

distance b/w clusters:

→ min

→ animation

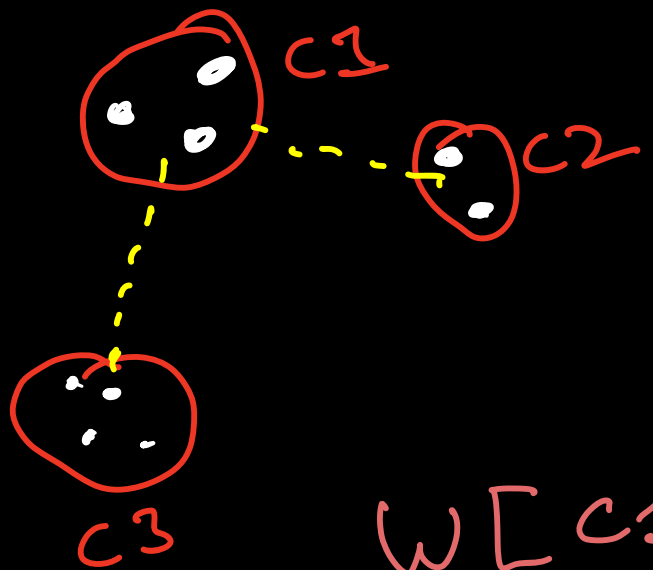
→ max
→ avg

} Pros and cons in post read.

→ any other func^, eg: squared, cosine etc..

→ wards method

Ward's distance / Ward's linkage



Using avg dist

$c_1 + c_2$, c_3

Ward's distance

$$W[c_1, c_2] =$$

$$WCSS(c_1 + c_2) - WCSS(c_1) + WCSS(c_2)$$

How much will ESS / WCSS increase

↓ error ↓ within cluster

when I merge 2 clusters.

→ We will merge clusters where there is min increase in ESS / WCSS

→ Very effective metric in practice.

Pros:

→ no need to pre-decide 'K'

→ hierarchy may actually be present in real world

Cons:

→ Sensitive to choice of linkage func.

→ Offline only. ← Can only be used for analysis

Term:

Proximity matrix:

→ code

distance (ci, cj)