Unsupervised learning → understand just Patterns of data ("x")

DATE

Clustering (~~Silhouette~~)
↳ Explain Data
↳ Filling Missing Values

"unlabelled" data into "clusters" of Similar inputs

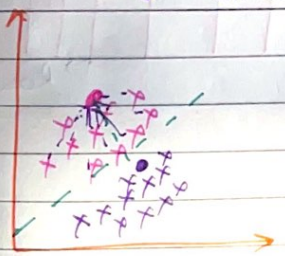
→ Describe data by Discrete "Groups"
Group data by visual Similarity

1. K-Means clustering

→ Find Similarity between points
→ make them cluster

⇒ ALGORITHM Random (Prototypes/Means) {centroids} K=2 (2 clusters)

1. Select "K" Value
2. Initialize centroids to randomly, closest to cluster K
3. Find the distance between the points
4. Select the Group and find the mean
5. Update the centroids, till no more movements



⇒ Why K = 2 / Find distance between Points

1. Euclidean distance (Pythagoras theorem)

$P_1 (x_1, y_1)$ = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

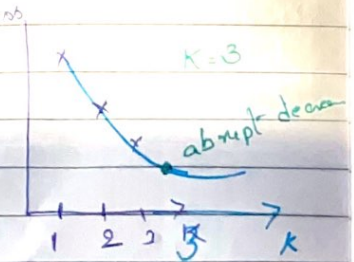
$P_2 (x_2, y_2)$

Select 'K' Value

⇒ Elbow method

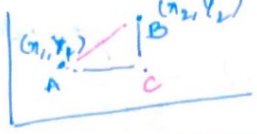
K = 1 to 20

$w_{css} = \sum_{i=1}^{m} ((c_i + x_i)^2$



vision

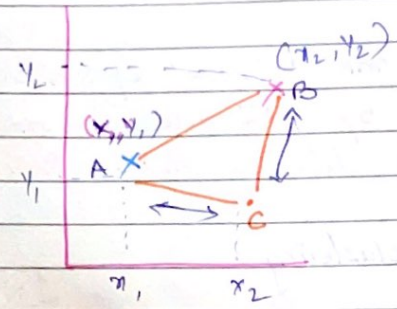
PAGE



$AB^2 = AC^2 + BC^2$

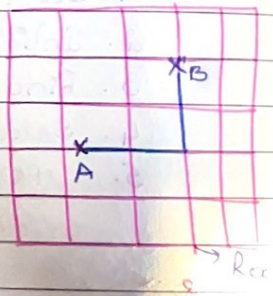
DATE

Manhattan distance

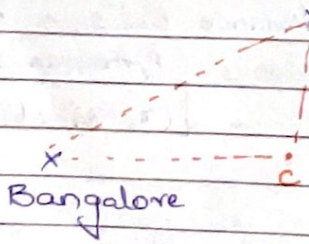


$$= |(x_2 - x_1)| + |(y_2 - y_1)|$$

Example :- Manhattan →



Euclidean



VISION

PAGE

DATE

=> Agglomerative Hierarchical clustering

→ recursive Kmeans (Top down)

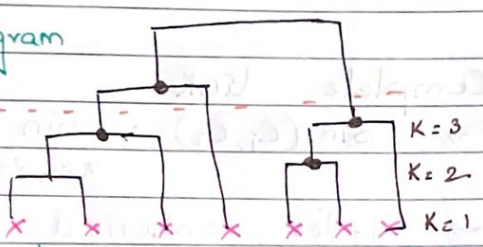
→ Build the Bottom level

→ 'm' object, every object in cluster

→ find the closest cluster and then

m-1

→ Dendogram



step 1, step 4, step 3, step 6 step 2, step 5

=> Nodes = Subsets

Characteristic = Root (whole i/p set)

leaves = individual elements

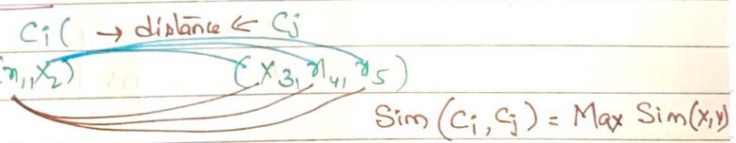
internal nodes = union of children.

Each level = different no. of clusters

=> Distance = closest Pair

P_1 P_2

=> 1) Single link



$Sim(C_i, C_j) = \max_{x \in C_i, y \in C_j} Sim(x, y)$

=> Most Similar Pair

2) Complete link :-

Least Similar Pair

3) Average link :-

average of Similarity

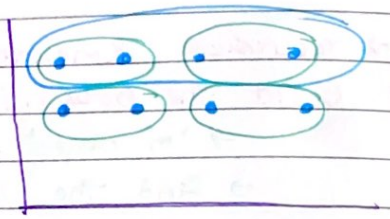
4) Centroid

Centers of Gravity

DATE

⇒ Single link Example (Thin, long)

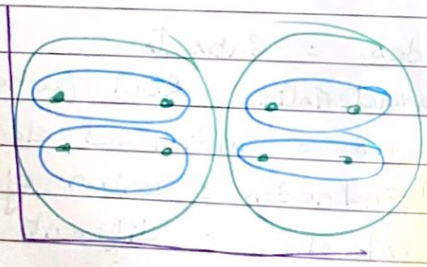
→ one Pair of link



⇒ Complete link

→ $Sim(C_i, C_j) = \min_{x \in C_i, y \in C_j} Sim(x, y)$

→ Tighter, Spherical



⇒ Computational complexity

HAC → N objects

N^2 Similarity $O(N^2)$

N-1 steps

⇒ Average link

→ All ordered pairs

→ All pairs between clusters

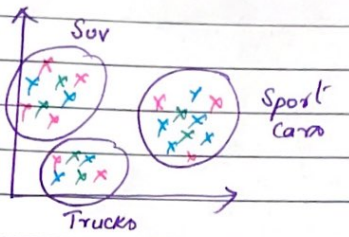
$$Sim(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\vec{x} \in C_i} \sum_{\vec{y} \in C_j} Sim(\vec{x}, \vec{y})$$

vision

PAGE

DATE

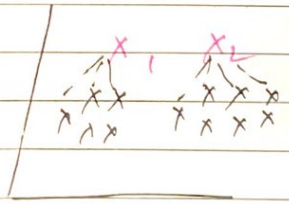
Ex:- Cars between Horsepower and Speed.



→ Are clusters well seprated?
Are clusters linearly Separable?
(How difficult to seprate stuff)

Challenges:-
1. overlapp / complicated shapes.
2. Algorith — No. of clusters
— don't need no. of clusters

⇒ K-Means algorithm
⇒ How many means in K clusters?
⇒ applied on numbers



⇒ Good Centroids → distance should be minimum.