__/__/____

(35) Clustering Methods

Clustering is the process of grouping similar datapoints together based on some notion of similarity (often distance)

Given data points $X = \{x_1, x_2 \dots x_n\}$ we aim to partition them into K groups (clusters) such that:

- Points within a cluster are more similar to each other.
- Points across clusters are more dissimilar.

→ Partitional Clustering (K-Means) [Nonlinear dim-reduction]

- The goal is to divide data into k clusters that directly optimize a criterion such as distance / variance

We aim to reduce the within-cluster sum of squares (wcss)

Objective function

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

C_i : cluster i , μ_i : mean (centroid) of cluster i

x_j : datapoints assigned to that cluster.

We're trying to find :

$\{C_1 \dots C_k\}$, $\{\mu_1, \dots \mu_k\}$ that minimize J

Steps:

1. Initialize : Choose k random points as initial centroids
2. Assignment: Assign each point to the cluster with the nearest centeroid:

$$C_i = \{x_j : \|x_j - \mu_i\|^2 \leq \|x_j - \mu_k\|^2, \forall k\}$$

3. Update step: Recompute centroids as the mean of assigned points :

$$\mu_i = 1/|C_i| \sum_{x_j \in C_i} x_j$$

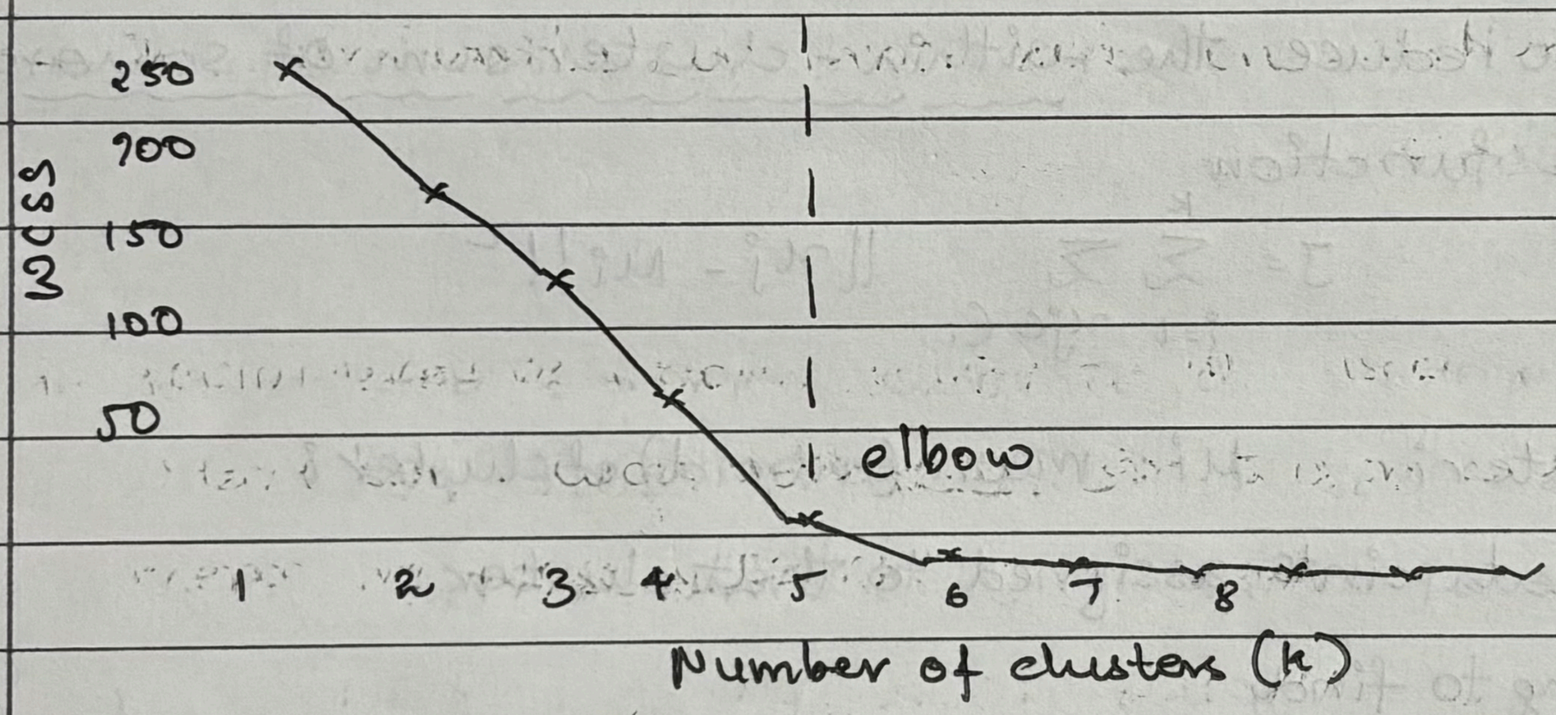
4. Repeat until centroids stop changing (convergence)

__/__/____

→ Deciding the ideal k is not straight forward.
The Elbow Method helps by plotting the WCSS against increasing k values and looking for a point where the improvement slows down - elbow.

Working of elbow

1. We iterate over a range of k values
2. Foreach k, we calculate a distance measure called WCSS (Within Cluster Sum of Squares)
WCSS → measures how well the data points are clustered around their respective centroids.
3. We try different k values (no of clusters)
For each k we run k means & calculate the WCSS.
4. Plot a graph with k on x-axis and WCSS on Y-axis



5. There comes a point where adding more clusters results in only a marginal decrease in WCSS. This is the elbow.

To evaluate, the quality of data points grouped together

1. Distortion: Measures the average squared distance b/w each data point and its assigned cluster center.
A measure of how well the clusters represent the data.
A lower distortion value indicates better clustering

Distortion: $\frac{1}{n}\sum_{i=1}^{n} \min_{c \in clusters} \|x_i - c\|^2$

__/__/____

2. Inertia : sum of squared distances of each data point to its closest cluster center. Its essentially the total squared error of the clustering. Lower inertia → better clustering

Inertia = $\sum_{i=1}^{n}$ distance $(x_i, c_j^*)^2$

Disadvantages of K-Means:
1. Fails on non-convex shapes
2. Outliers can drag centroids
3. fixed k

→ K-Mediods: Minimizes distance to actual datapoints
→ It uses mediods (actual points) → robust to outliers

→ K-Means++ : A smarter initialization step that tends to select centroids that are distant from one another, it makes K-Means algorithm much less likely to converge to a suboptimal solution.

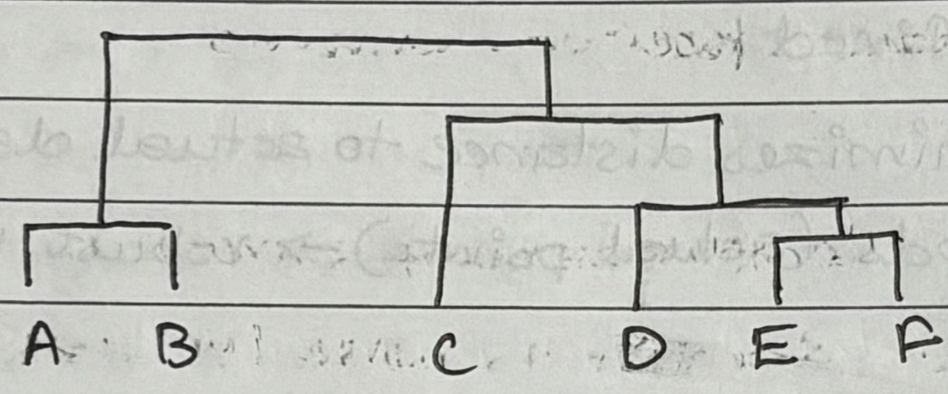
* Active Learning: To continue improving your model and your training set, the next step could be to do a few rounds of active learning, which is when a human expert interacts with the learning algorithm, providing labels for specific instances when the algorithm requests them.

__/__/____

(36) Hierarchical Clustering

Builds a tree like structure (dendogram) that represents data grouped at different levels of similarity.

- You can cut the tree at any level to get your desired no of clusters
- A dendogram is a tree that shows how clusters are merged step-by-step:



- We keep merging the two most similar clusters step by step untill all points belong to one big cluster
- The similarity / distance between clusters depend on the 'linkage method'.

Types:

1. Agglomerative (Bottom-up) - Most used
   Start with each data point as its own cluster ; iteratively merge the closest pairs.
2. Divisive (Top-Down) - Rarely Used
   Start with one big cluster; recursively split it into smaller clusters.

Agglomerative Algorithm Steps:

1. Treat each point as its own cluster → n clusters
2. Compute Distance Matrix : Pairwise distances b/w all clusters.
3. Merge Closest Clusters : Combine two with minimum distance
4. Update Distance Matrix : Update distances b/w new cluster & remaining
5. Repeat until only one cluster remains.

__/__/____

→ "Linkage Method": How distances between clusters are computed

1. Simple linkage: $\min_{x_i \in A, x_j \in B} d(x_i, x_j)$
   - Closest pair between clusters
2. Complete linkage: $\max_{x_i \in A, x_j \in B} d(x_i, x_j)$
   - Farthest Pair ~ compact equal sized clusters
3. Ward's Method: Minimize increase in total variance.
   - Like K-means uses variance minimization.
   - Cluster shape: Spherical Clusters.

(Most used) →

Disadvantages:
1. Computationally expensive
2. Sensitive to noise / outliers
3. No reassignments, once merged

(37) Density Based Clustering Methods

K-Means and Hierarchical clustering assume
- Clusters are convex (spherical)
- Every point belongs to a cluster.

But real data may have
- Non Convex shapes
- Noise / outliers
- Varying densities

Density based clustering solves this by defining clusters as dense regions of points seperated by low-density areas.

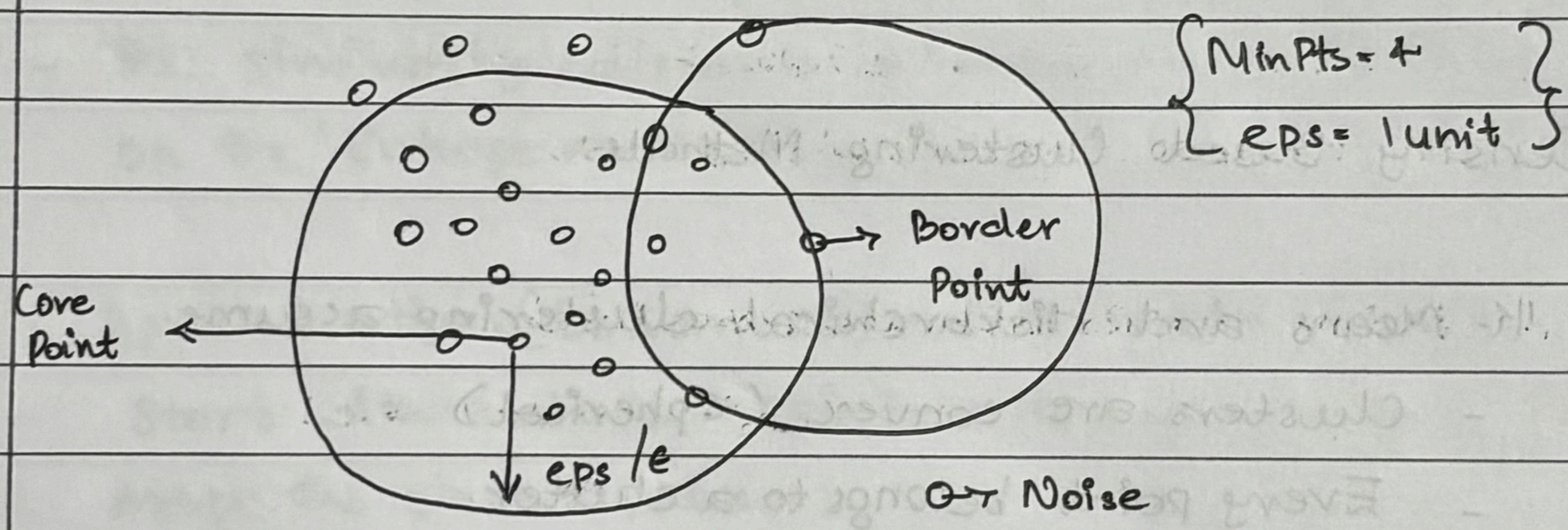
A cluster = region of high data density, seperated by regions of low density. If many points lie close together → dense region (cluster)

__/__/____

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Groups data points that are closely packed together and marks outliers as noise based on their density in the feature space.
It identifies clusters as dense regions in the data space seperated by areas of lower density.

Key Params:
- ϵ (epsilon): Radius of neighbourhood around a point.
  If eps is too small → most points will be classified as noise
  If eps is too large → clusters may merge & algo might fail to distinguish them.
- minPts: Minimum no of points required to form a dense region
  set minPts >= Dimensions in dataset + 1.



Steps:
1. For a given point p (neighbours)
   - Core Point: Has >= minPts points within ϵ
   - Border Point: Not a core point, but is within ϵ of a core poin
   - Outlier: Neither core nor border.
2. Algorithm: a. Pick a point p not yet visited.
   b. Find all points within ϵ → Neighborrhood(P)
   c. If |Neighborhood(P)| < minPts: mark as Noise
      Else: Mark p as a core point
      form a new cluster, expand recursively by adding all poR
   d. Continue until all points are assigned.

__/__/____

3. Output: cluster labels for dense regions.

Advantages: No need to mention / specify no of clusters
Can find arbitrarily shaped clusters.
Handles outliers naturally.

Dis Advantages: Sensitive to parameters (ϵ, minPts)
Struggles with varying densities.
ϵ hard to set in high dimensions.
Slower compared to k-Means

fix

OPTICS (Optical / Ordering Points To Identify the Clustering Structure)
- Handles varying densities
- Doesn't require single ϵ
- Produces reachability plot

HDBSCAN (Hierarchical DBSCAN)
- Handles variable density clusters better.