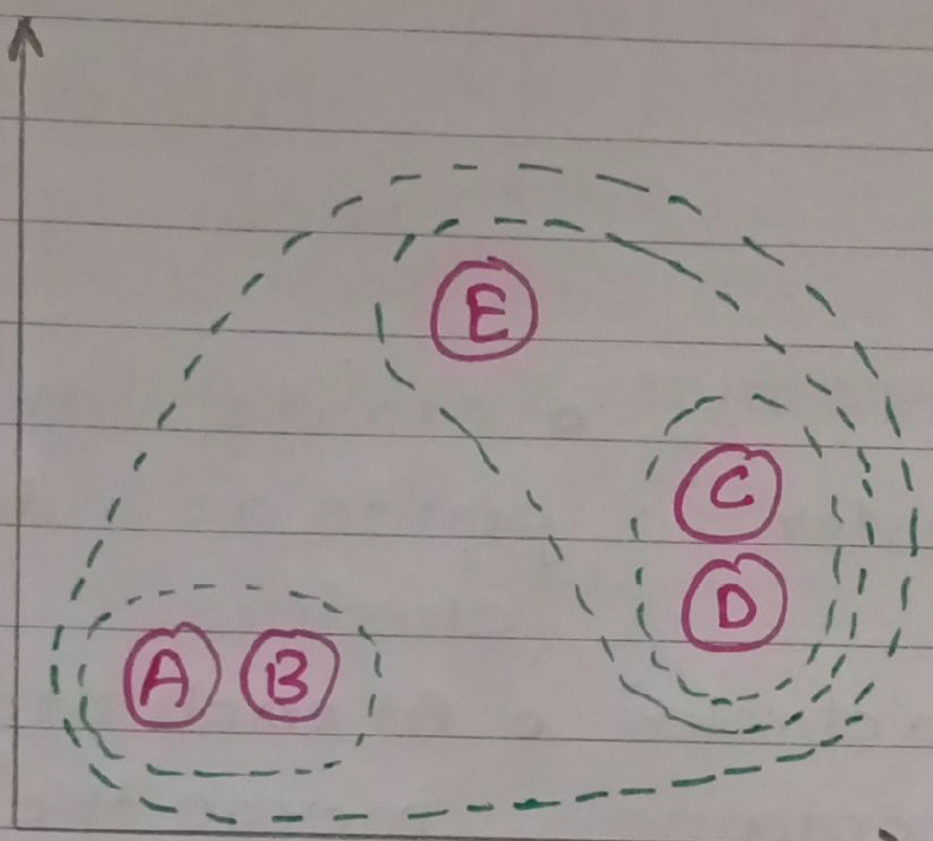
Hierarchical Clustering:

Suppose we have 5 points

(A) (B) (C) (D) (E)



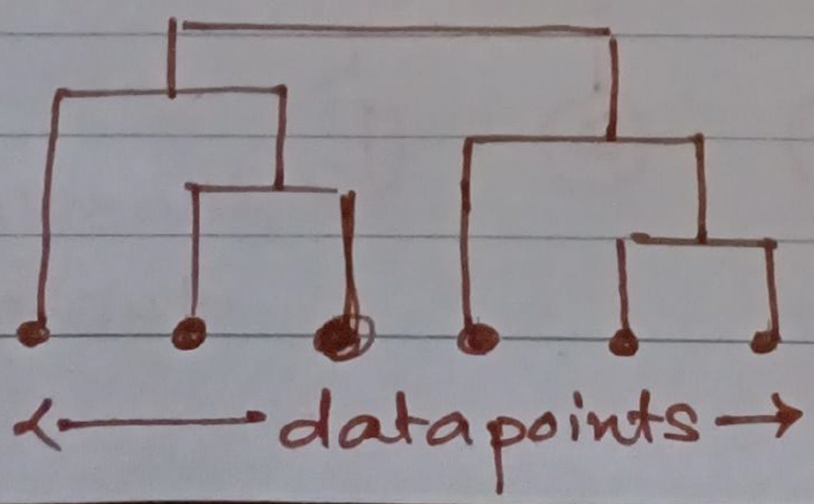
Hierarchical clustering is a popular method for grouping objects.

It creates groups so that objects within a group are similar to each other and different from objects in other groups.

Clusters are visually represented in a hierarchical tree called dendrogram.

The sole concept of Hierarchical clustering lies in just the construction and analysis of a dendrogram.

A dendrogram is a tree-like structure that explains the relationship between all the data points in the system.



Komal Diwe

Types of Hierarchical Clustering

Divisive clustering (top-down)

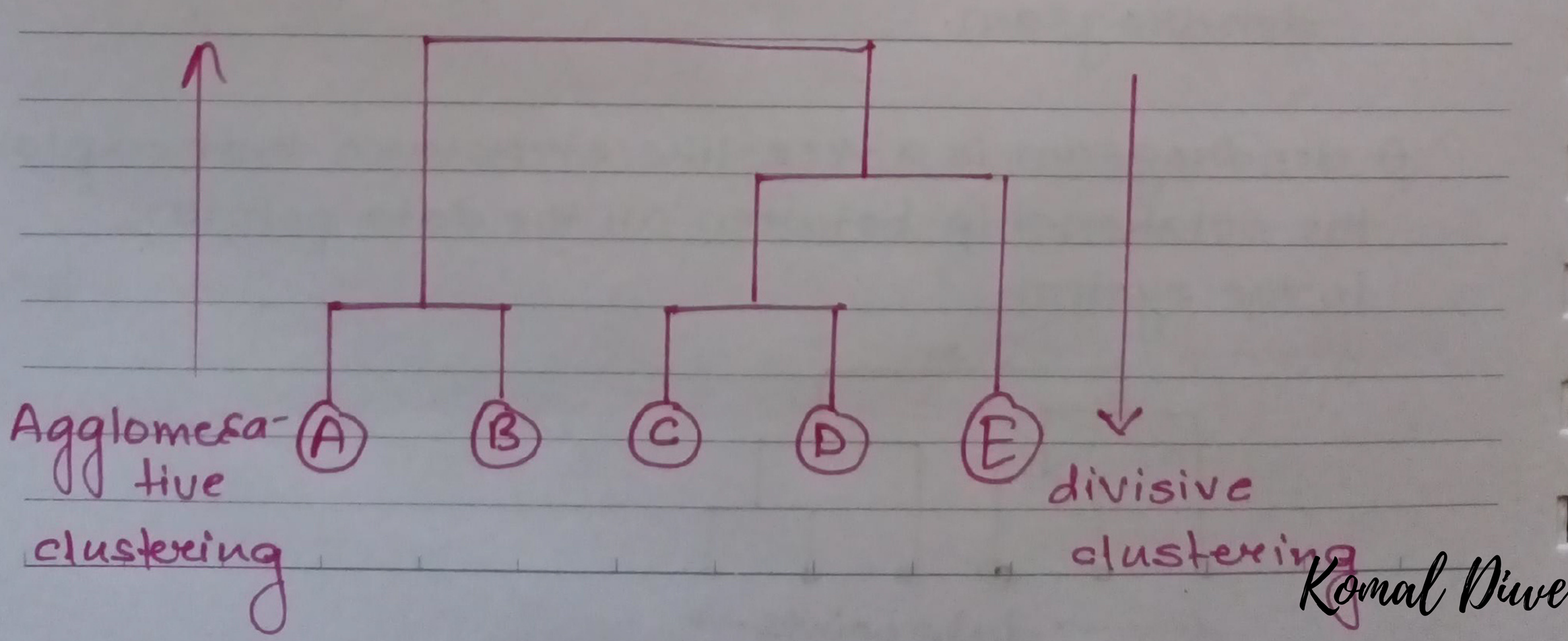
- start with one, all-inclusive cluster
- At each step, split a cluster until each cluster contains a point (or there are k cluster)

Agglomerative clustering (bottom-up)

- starts with the points as individual clusters.
- At each step, merge the closest pair of clusters until only one cluster left.

Traditional Hierarchical algorithms use a similarity or distance matrix.

→ merge or split one cluster at a time.



Komal Diwe

Strengths of Hierarchical Clustering.

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by "cutting" the dendogram at the proper level.
- They may correspond to meaningful taxonomies
  - Example in biological sciences

Agglomerative Clustering

- more popular hierarchical clustering technique.
- key operation is the computation of the proximity of two clusters.
  - different approaches to defining the distance between clusters distinguish the different algorithms.

Komal Diwe

- Basic algorithm is straightforward

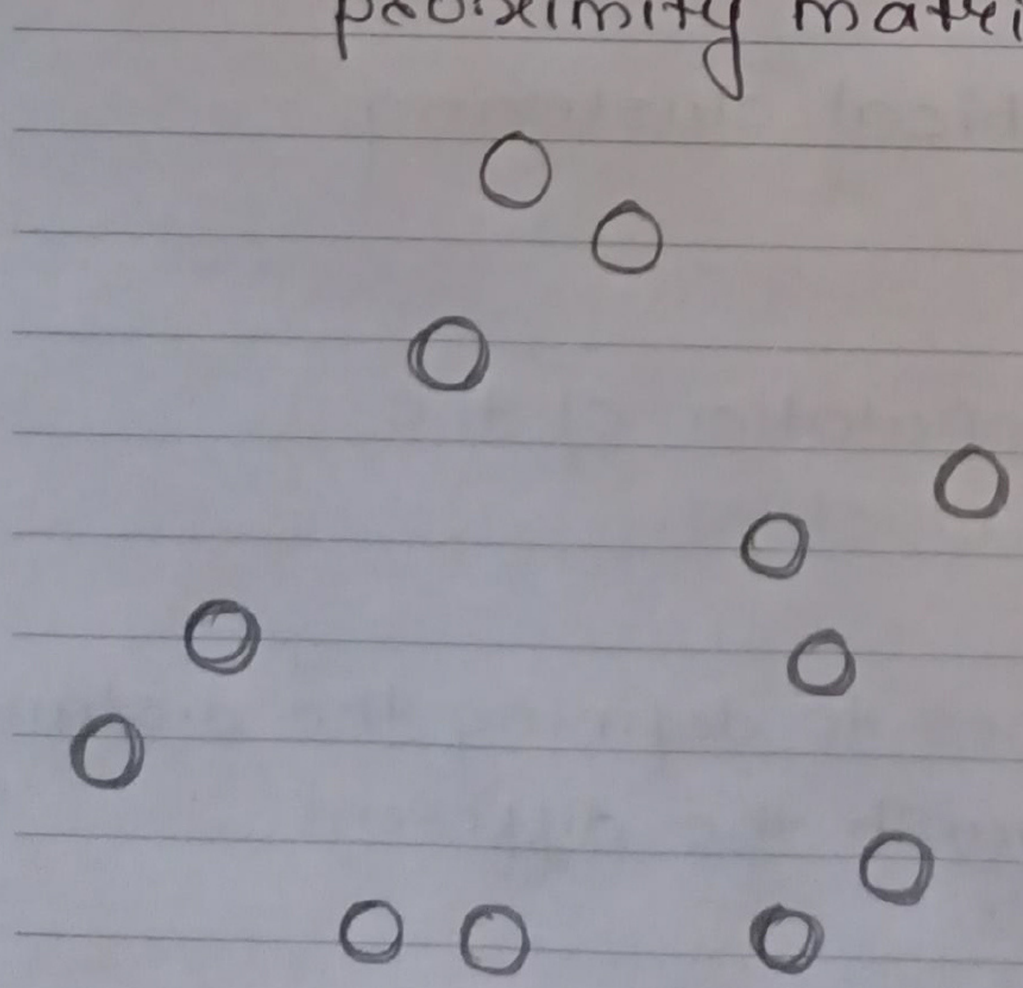
1. compute the proximity matrix
2. let each data point be a cluster

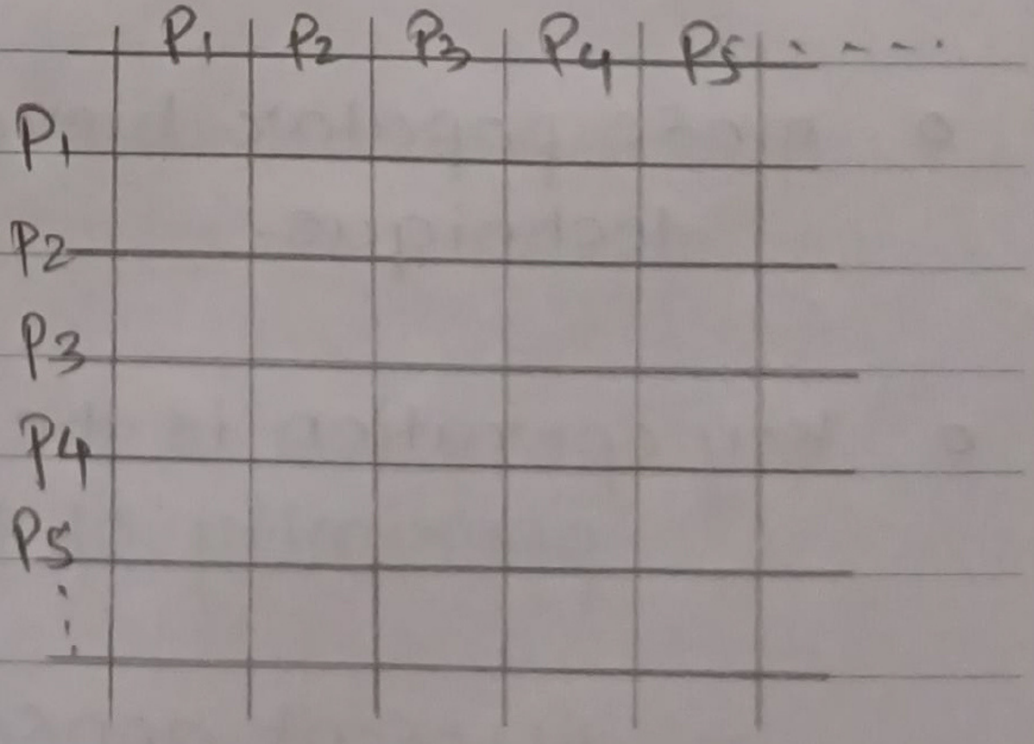
3. Repeat
4. merge the two closest clusters
5. update the proximity matrix

6. until only a single cluster remains

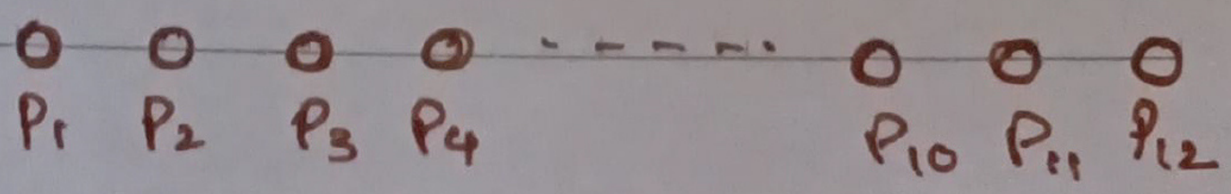
Starting Situation:

- Start with clusters of individual points and a proximity matrix.





Proximity matrix

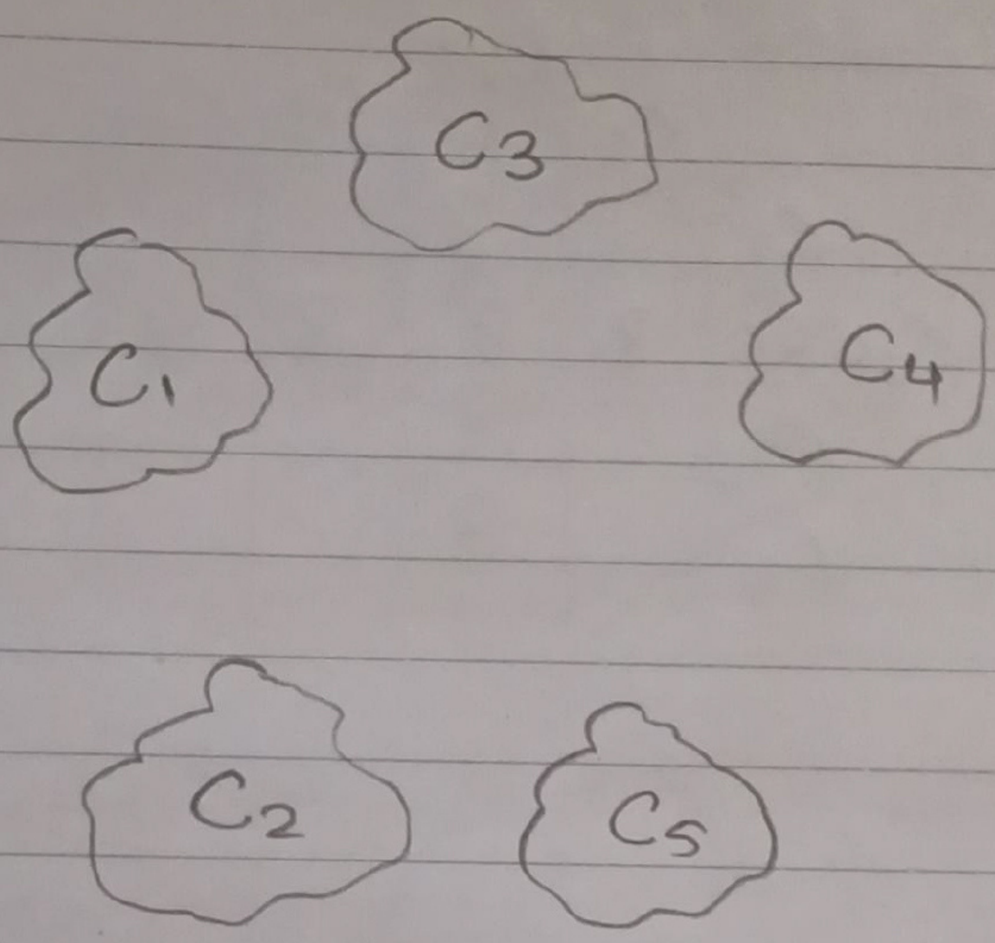


Komal Dive

__/__/__

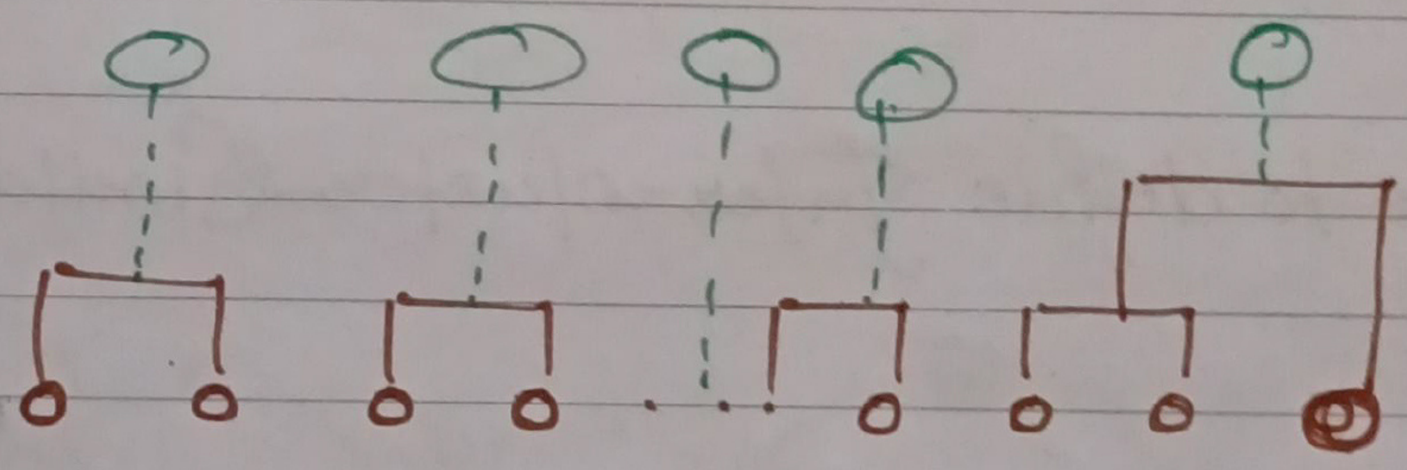
Intermediate Situation:

- after some merging steps, we have some clusters.

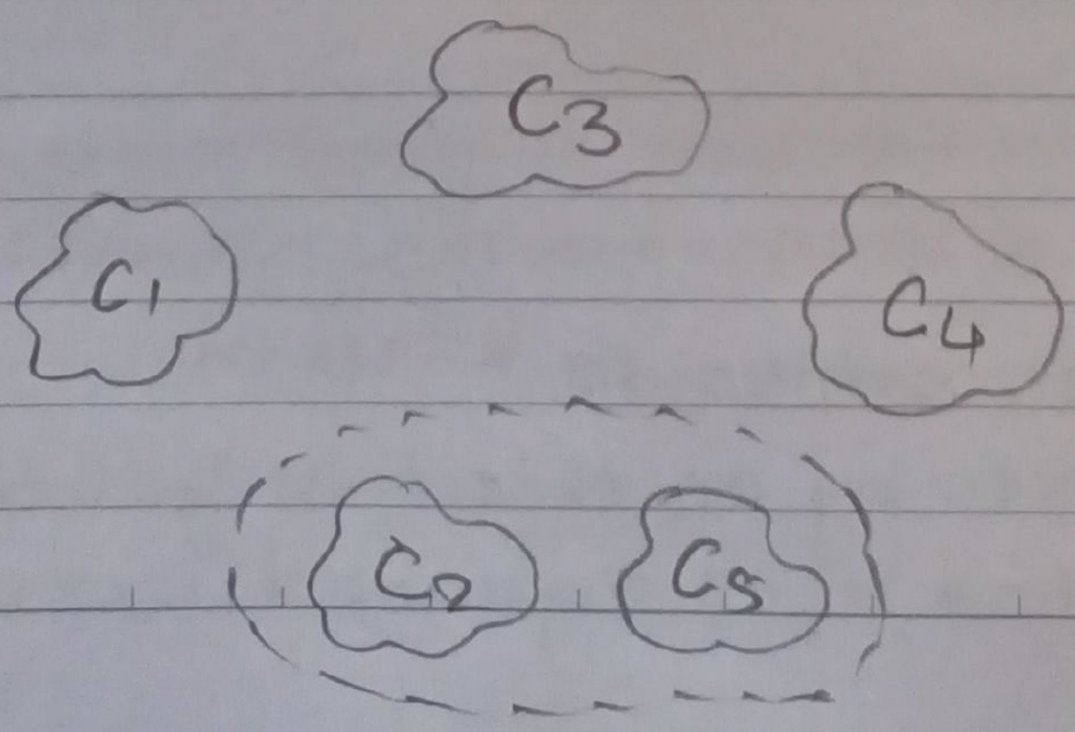


	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

proximity matrix



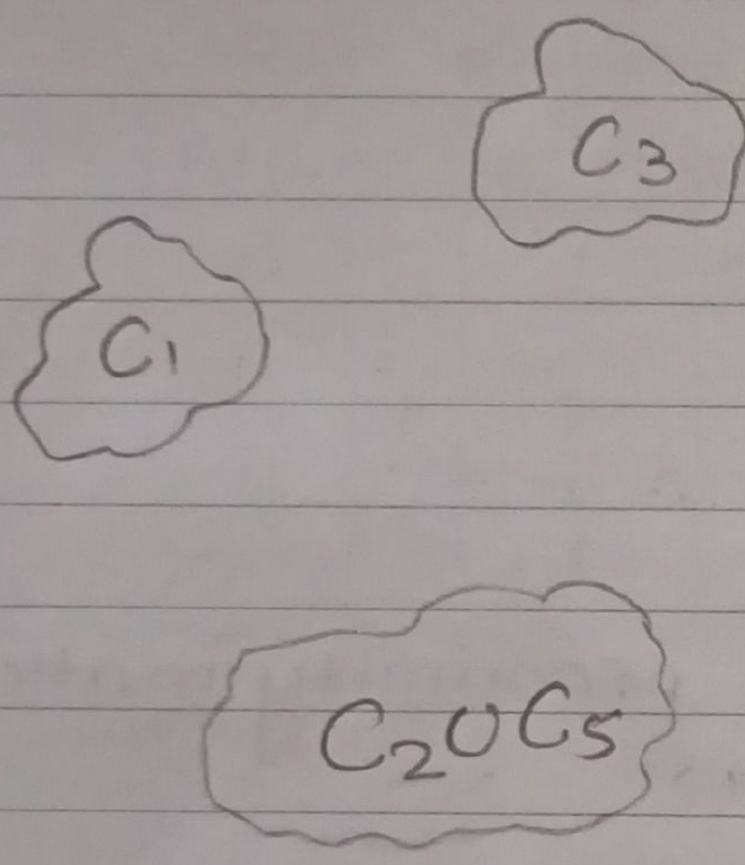
- we want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



Komal Diwe

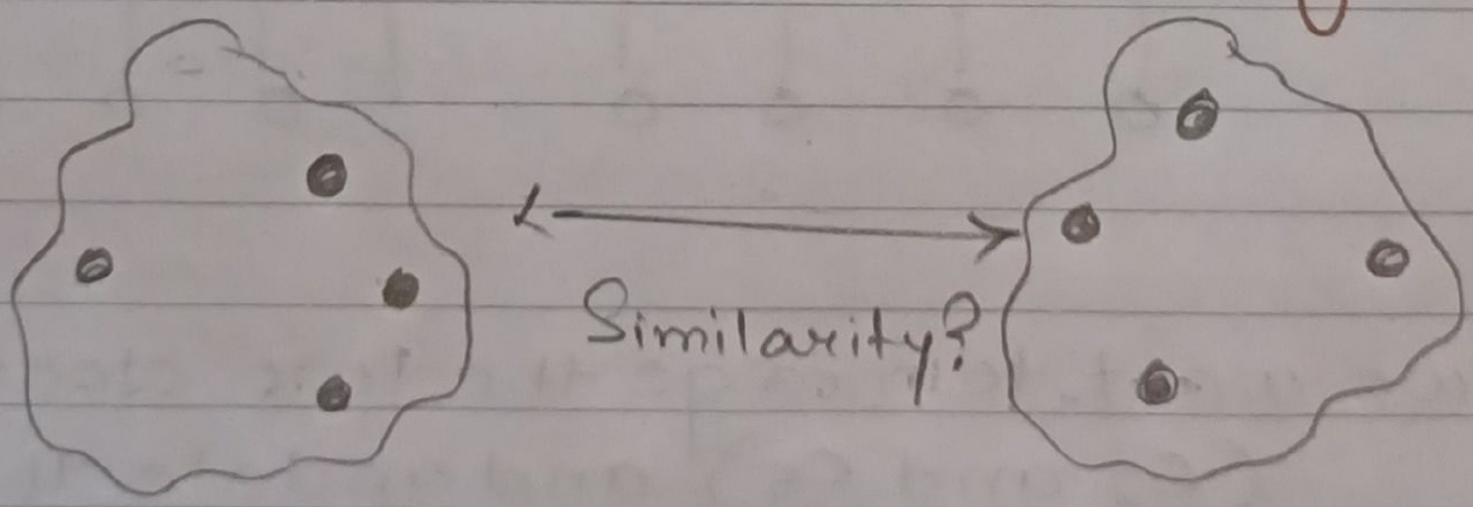
After merging:

- the question is "How do we update the proximity matrix?"



	C1	C2UC5	C3	C4
C1		?		
C2UC5	?	?	?	?
C3		?		
C4		?		

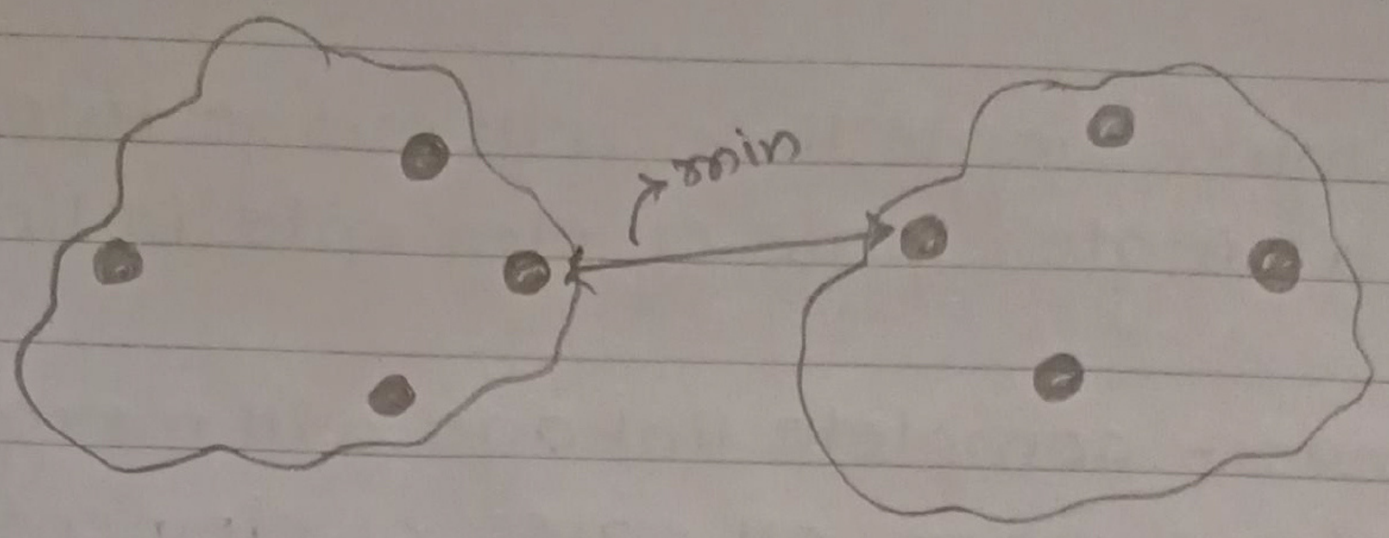
How to define Inter-cluster Similarity?



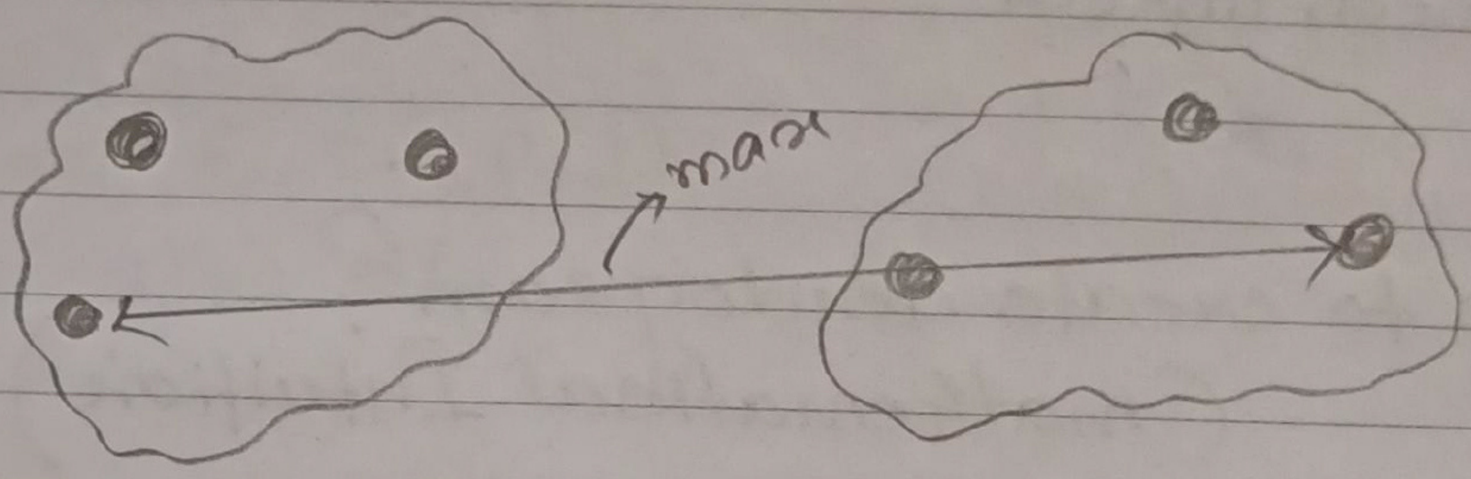
- min
- max
- group average
- distance between centroids
- other methods driven by an objective function
  - Ward's method uses squared errors.

Komal Diwe

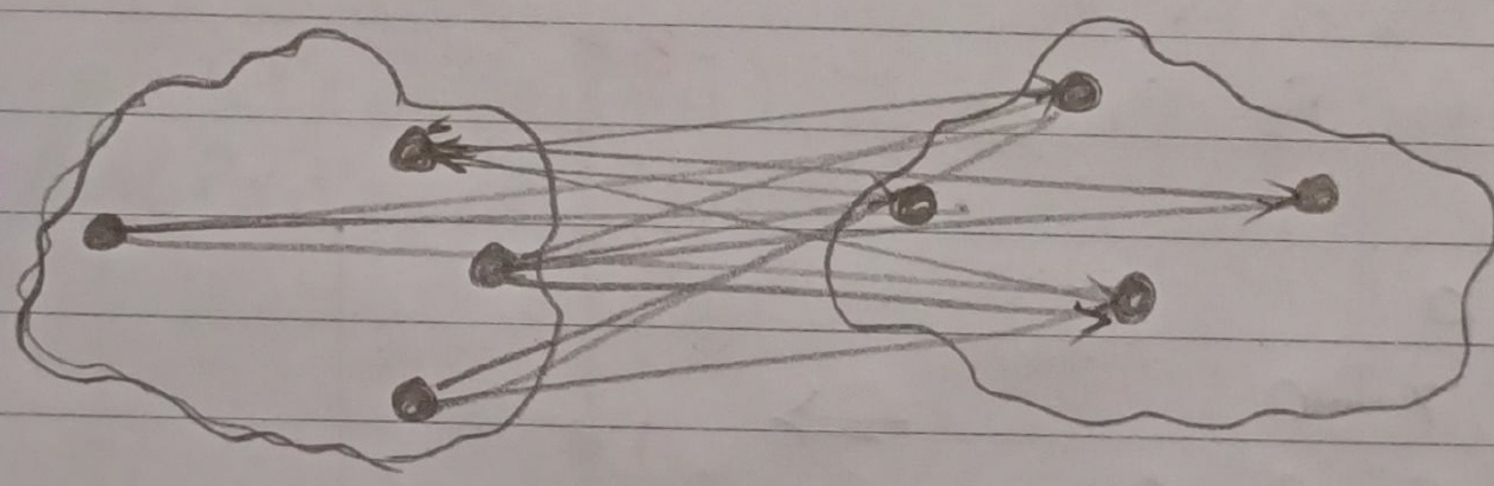
min



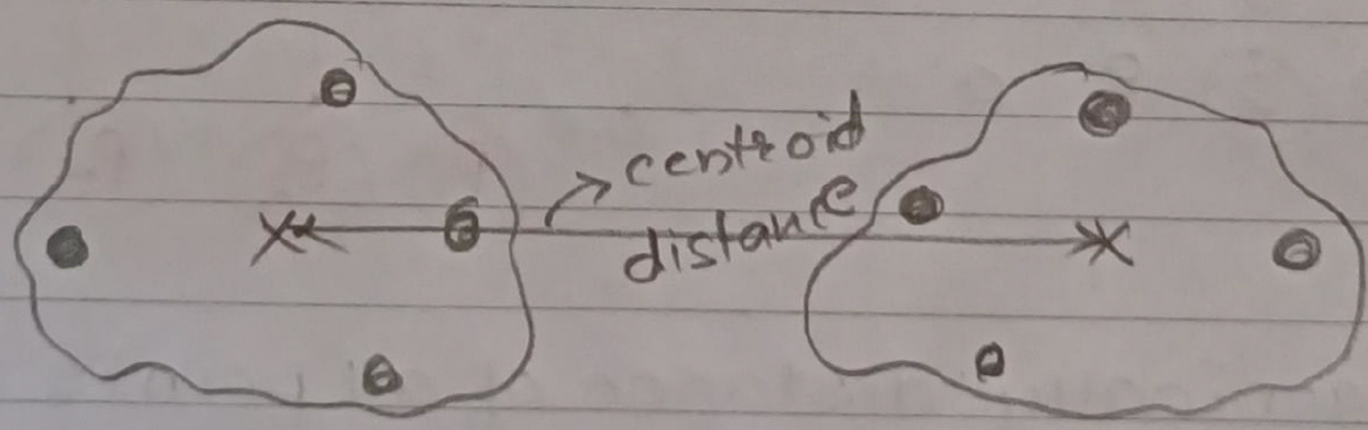
max



group average



distance between centroids



Single Link- complete link:

- Another way to view the processing of the hierarchical algorithm is that we create links between their elements in order of increasing distance.

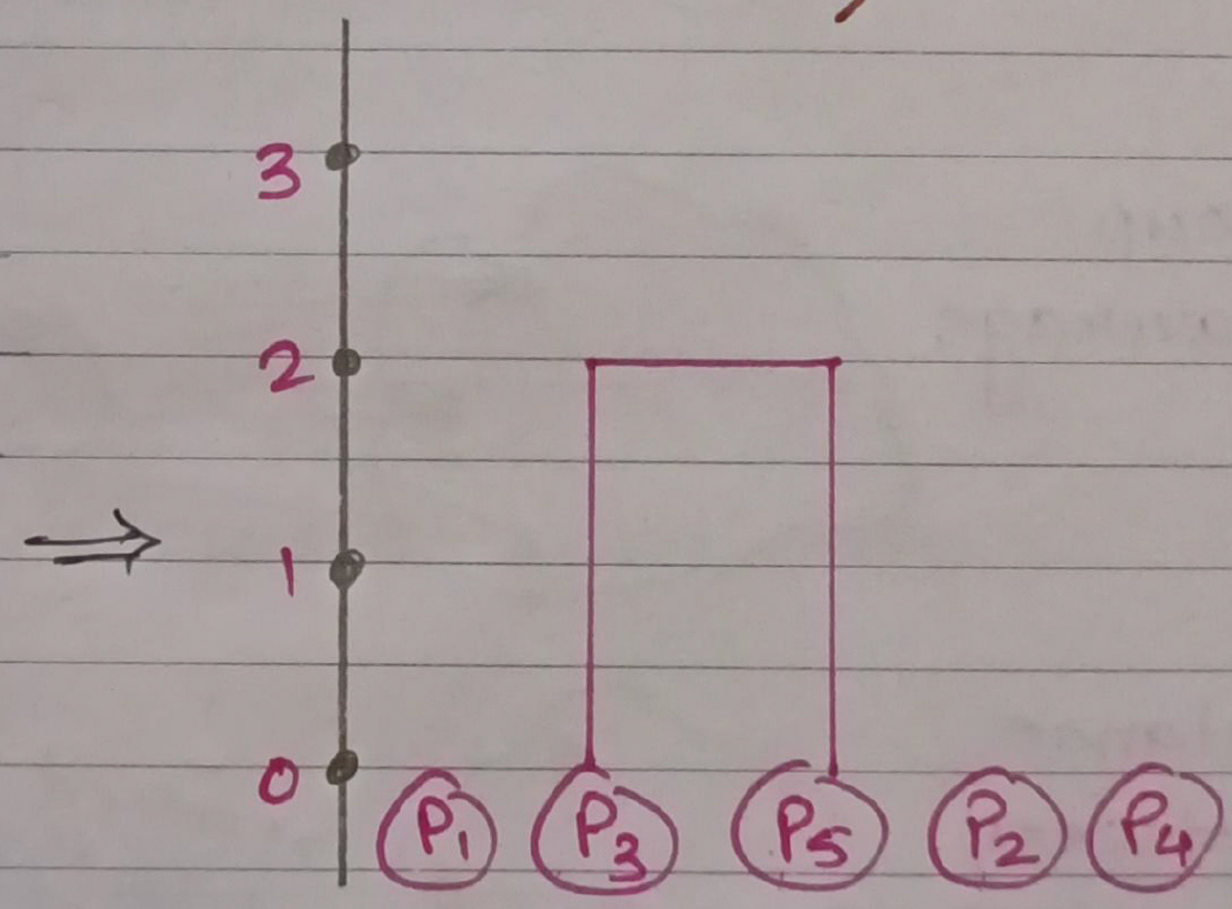
Komal Diwe

- The min - single link, will merge two clusters when a single pair of elements is linked.

- The max - complete linkage will merge two clusters when all pairs of elements have been linked.

How to create dendogram? (mathematical Intuition)

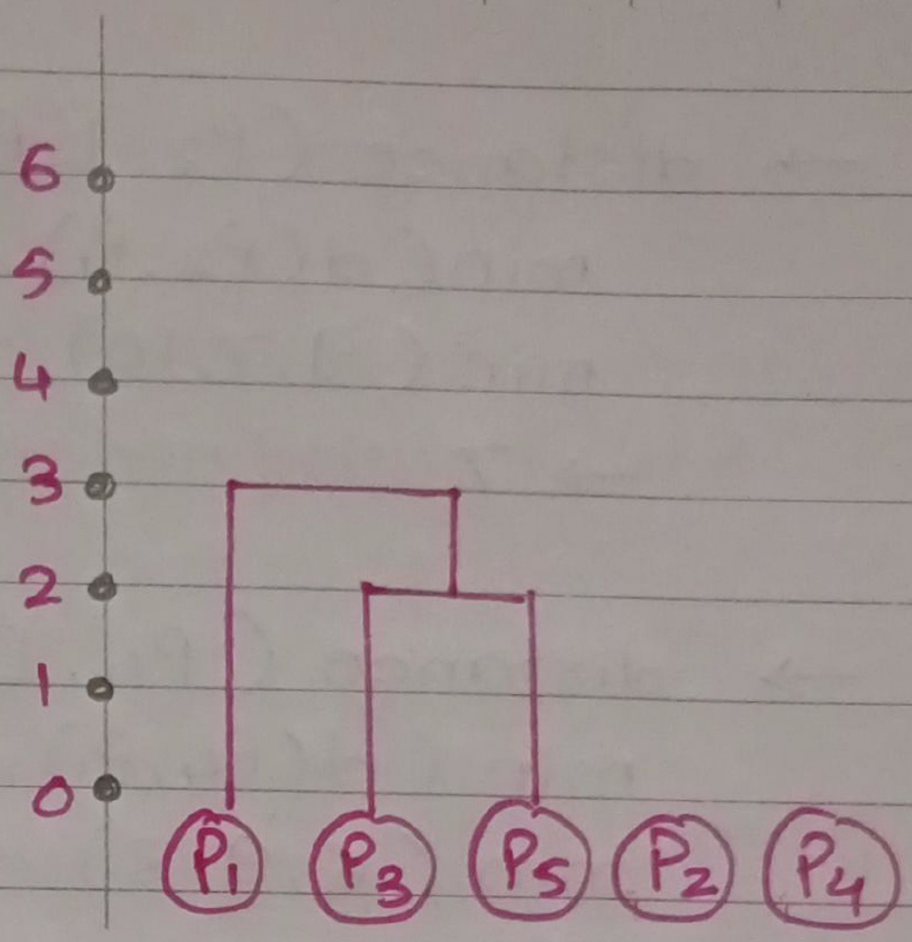
	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2	8	0



Diagonically distance of all points will be zero.

Komal Diwe

	P_1	P_2	$P_3 \cup P_5$	P_4
P_1	0			
P_2	9	0		
$P_3 \cup P_5$	3	7	0	
P_4	6	5	8	0



→ distance (P_1, [P_3, P_5])

min (d(P_1, P_3), d(P_1, P_5))

min (3, 11)

→ 3.

→ distance (P_2, [P_3, P_5])

min (d(P_2, P_3), d(P_2, P_5))

min (7, 10)

→ 7

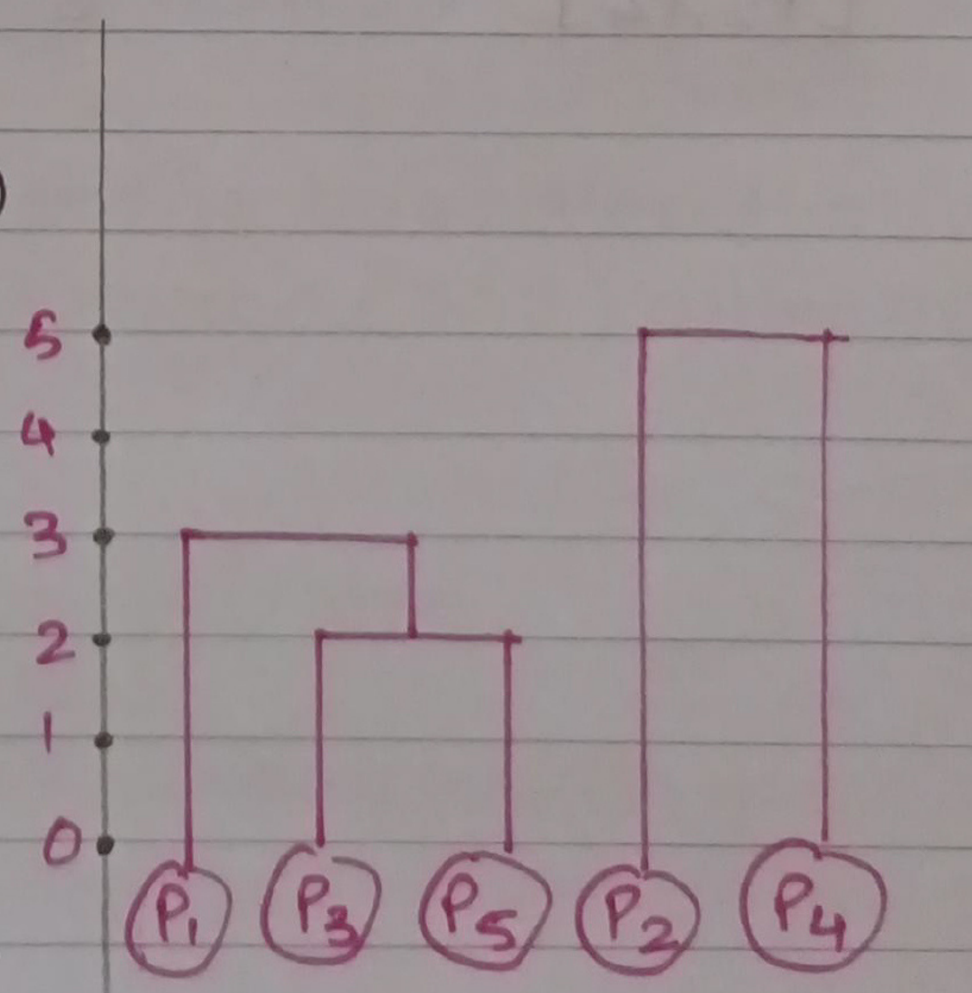
→ distance ([P_3, P_5], P_4)

min (d(P_3, P_4), d(P_5, P_4))

min (9, 8)

→ 8.

	P_1, P_3, P_5	P_2	P_4
[P_1, P_3, P_5]	0		
P_2	7	0	
P_4	6	5	0



Komal Diwe

→ distance $(P_2, [P_1, P_3, P_5])$

$\min(d(P_2, P_1), d(P_2, P_3), d(P_2, P_5))$

$\min(9, 7, 10)$

→ 7

→ distance $(P_4, [P_1, P_3, P_5])$

$\min(d(P_4, P_1), d(P_4, P_3), d(P_4, P_5))$

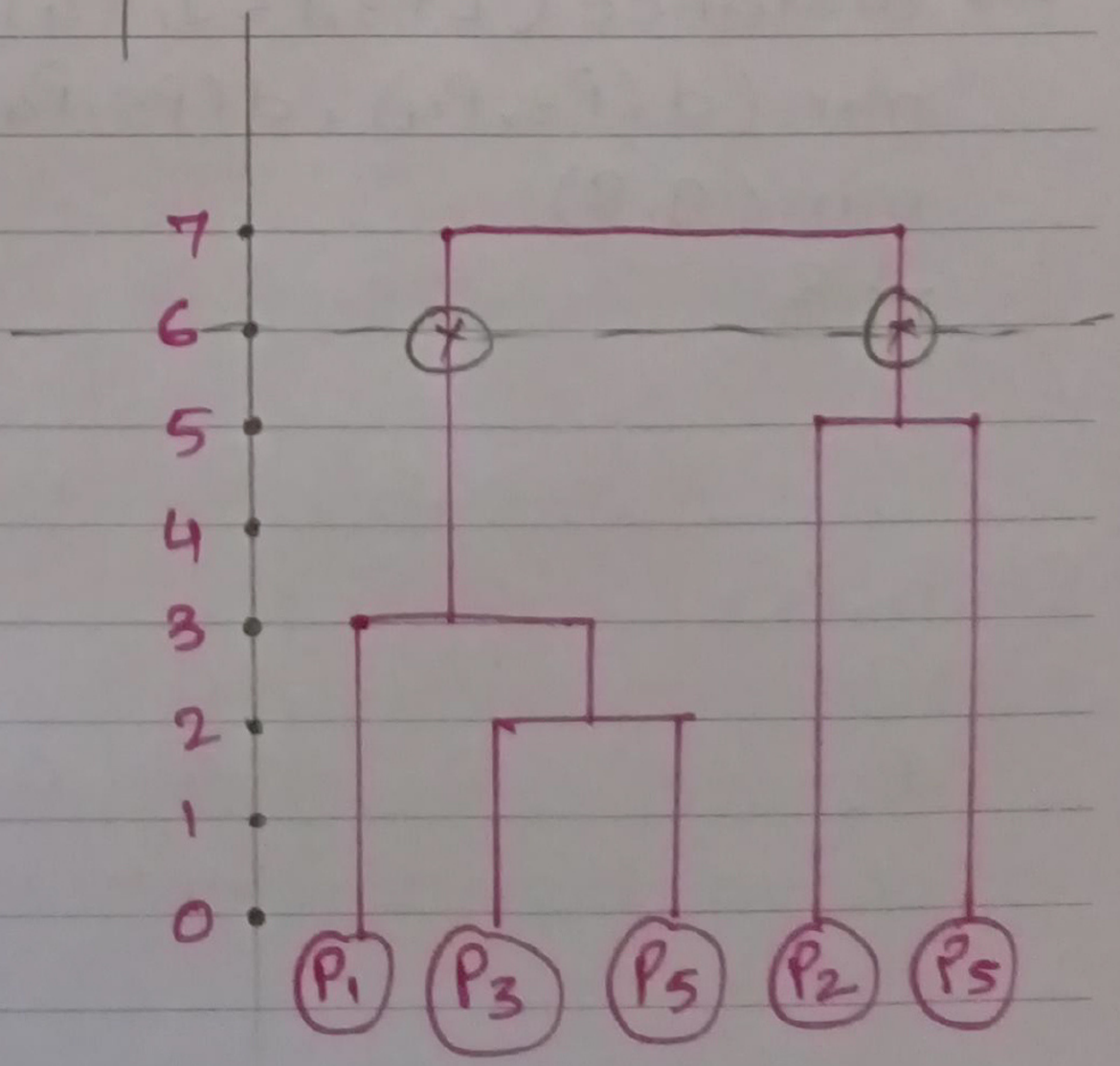
$\min(6, 9, 8)$

→ 6.

Now the new cluster will be

$[P_1, P_3, P_5]$ and $[P_2, P_4]$

	$[P_1, P_3, P_5]$	$[P_2, P_4]$
$[P_1, P_3, P_5]$	0	
$[P_2, P_4]$	6	0



Komal Diwe

Cluster similarity : Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$proximity(cluster_i, cluster_j) = \frac{\sum_{p_i \in c_i, p_j \in c_j} proximity(p_i, p_j)}{|cluster_i| \times |cluster_j|}$$

$p_i \in cluster_i$

$p_j \in cluster_j$

- limitations: biased towards globular clusters.

Cluster similarity : Ward's method

- similarity of two clusters is based on the increase in squared error (SSE) when two clusters are merged.

- Biased towards globular clusters

Komal Diwe

Hierarchical clustering: time and space requirements

$O(N^2)$ space since it uses the proximity matrix.
- N is the number of points

$O(N^3)$ time in many cases.
- There are N steps and at each step the size, N^2, proximity matrix must be updated and searched.

- complexity can be reduced to $O(N^2 \log(N))$ time for some approaches.

Komal Diwe

Density-Based Clustering

- (1) Density-based spatial clustering of applications with noise (DBSCAN)

- DBSCAN is a Density-Based Clustering algorithm.

The DBSCAN algorithm uses two parameters:

- minPts: The minimum number of points (a threshold) clustered together for a region to be considered dense.

- eps (ε): A distance measure that will be used to locate the points in the neighborhood of any point.

These parameters can be understood if we explore two concepts called Density Reachability and Density Connectivity.

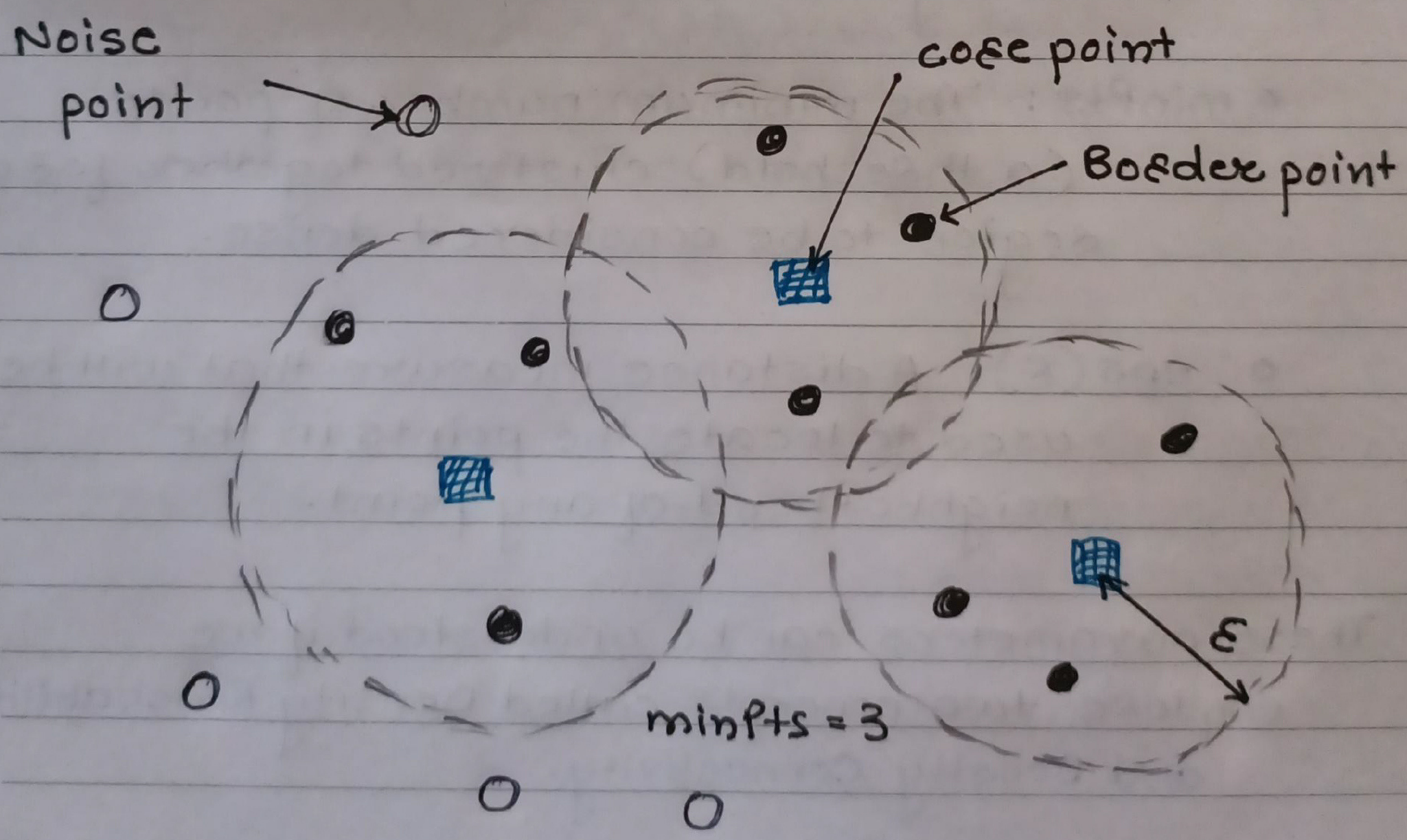
Komal Diwe

__/__/__

Reachability: in terms of density establishes a point to be reachable from another if it lies within a particulare distance (eps) from it.

connectivity: involves a transitivity based chaining-approach to determine whether points are located in a particular cluster.

These are three types of points after the DBSCAN clustering is complete:



Komal Diwe

core: this is a point that has at least m points within distance n from itself.

- a point is a core point if it has more than a specified number of points (minPts) within Eps

- These points belong in a dense region and are at the interior of a cluster.

Border: this is a point that has at least one core point at a distance 'n'.

- a border point has fewer than minPts within Eps, but is in the neighborhood of a core point.

Noise: this is a point that is neither a core nor a border and it has less than m points within distance n from itself.

Komal Diwe

core :

a spe
Eps

DBSCAN:

- **Density at point p:** number of points within a circle of radius Eps.
- **Dense Region:** A circle of radius Eps that contains at least MinPts points.

- These points belong in a dense region and are at the interior of a cluster.

Border: this is a point that has at least one core point at a distance 'n'.

- a border point has fewer than minPts within Eps, but is in the neighborhood of a core point.

Noise: this is a point that is neither a core nor a border and it has less than m points within distance n from itself.

Komal Diwe

Algos

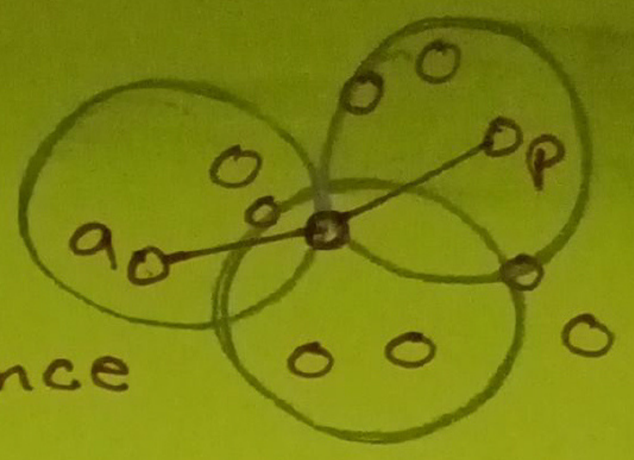
- The

pick

(

Density edge:

- we place an edge betn two core points q and p if they are within distance Eps.



Density-connected:

- a point p is density-connected to a pt q if there is a path of edges from p to q.

- If there are at least 'minPoints' points within a radius of 'ε' to the point then we consider all these points to be part of the same cluster.

- The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point.

- label points as core, border and noise
- eliminate noise points

- For every core point p that has not been assigned to a cluster

  - create a new cluster with the point p and all the points that are density-connected to p.
- Assign border points to the cluster of the closest core point.

Komal Diwe

Algorithmic steps :⇒

- The algorithmic proceeds by arbitrarily picking up a point in the dataset
  (untill all points have been visited)

- If there are at least 'minPoints' points within a radius of 'ε' to the point then we consider all these points to be part of the same cluster.

- The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point.

. label points as core, border and noise
. eliminate noise points

. For every core point p that has not been assigned to a cluster

  - create a new cluster with the point p and all the points that are density-connected to p.
. Assign border points to the cluster of the closest core point.

Komal Diwe

__/__/__

Parameter Estimation :- - For DBSCAN, the parameters ε and minPts are needed.

- **minPts :** as a rule of thumb, a minimum minPts can be derived from the number of dimensions D in the data set,

  $minPts \geq D+1$

$minPts \simeq 1 \longrightarrow$ doesn't make any sense

$minPts \leq 2 \longrightarrow$ the result will be the same as of hierarchical clustering with the single link metric, with the dendrogram cut at height ε.

∴ minPts must be choosen at least 3.

- **ε :** The value for ε can then be chosen by using a k-distance graph, plotting the distance to the $k = minPts - 1$ nearest neighbor ordered from the largest to the smallest value.

Complexity :- $O(n \log n) \longrightarrow$ Best case

$O(n^2) \longrightarrow$ worst case

Komal Dive

Advantages of DBSCAN:

- Is great at separating clusters of high density versus clusters of low density within a given dataset.

- Is great with handling outliers within the dataset.

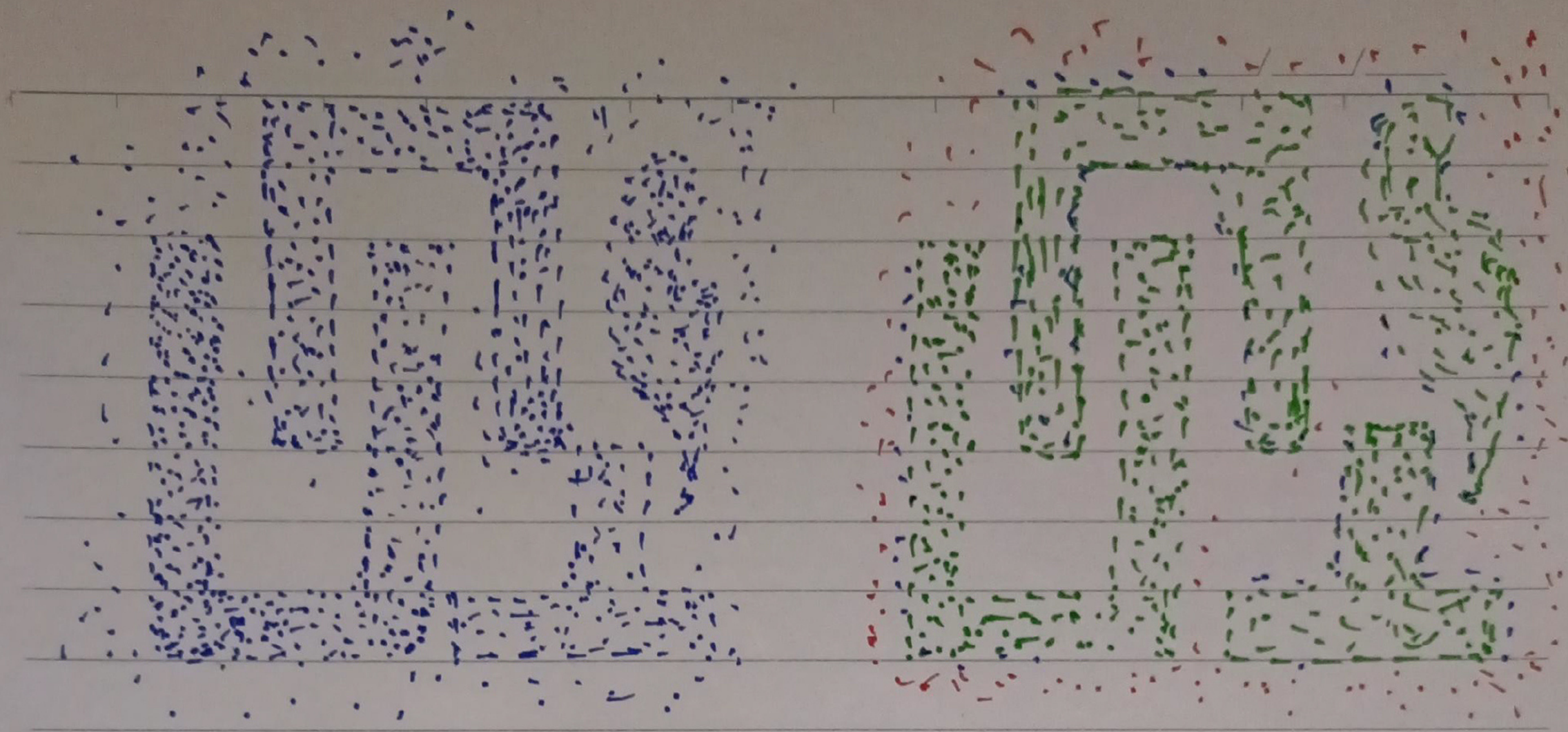
Disadvantages of DBSCAN:

- does not work well when dealing with clusters of ~~var~~ different densities.

while DBSCAN is great at separating high density clusters from low density clusters, DBSCAN struggles with clusters of similar density.

- struggle with high dimensionality data.

Komal Diwe



original points.

points types:
core, border and
noise

Eps = 10, minPts = 4

Komal Diwe