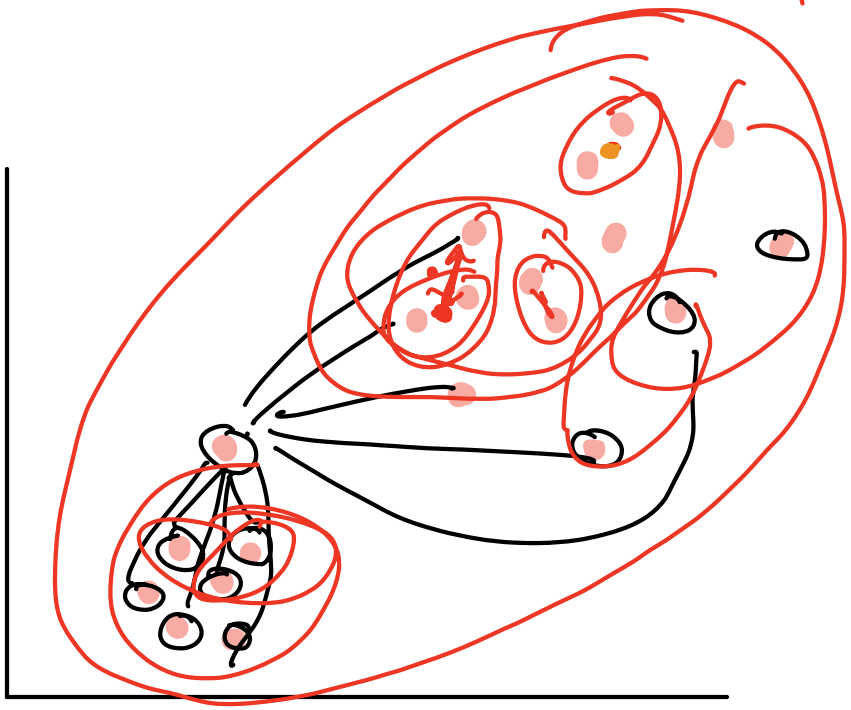
- Recap
- Hierarchical Clustering
- Proximity Matrix
- Implementation Scipy / Sklearn
- Advantage Disadvantange

Hierarchical Clustering:

1. Agglomerative Clustering
2. Divisive Clustering

Agglomerative Clustering

- Bottom up approach



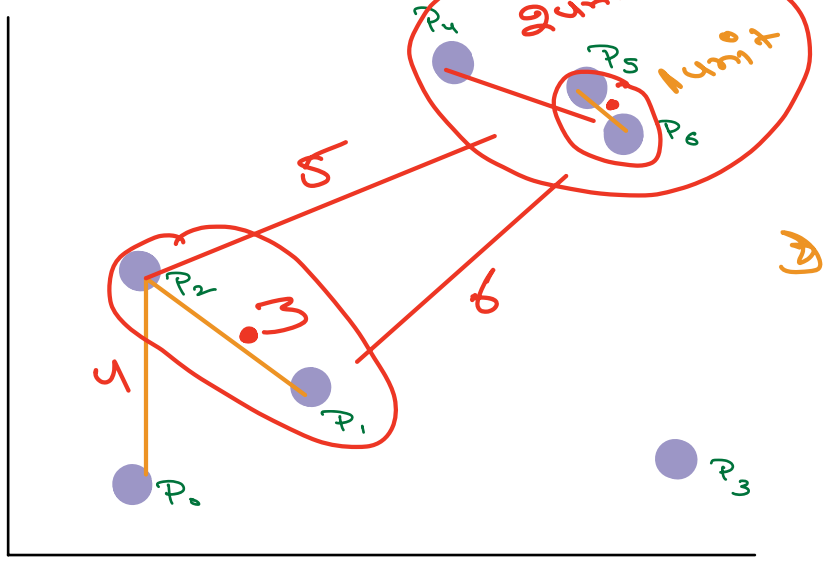
$n \geq 500$

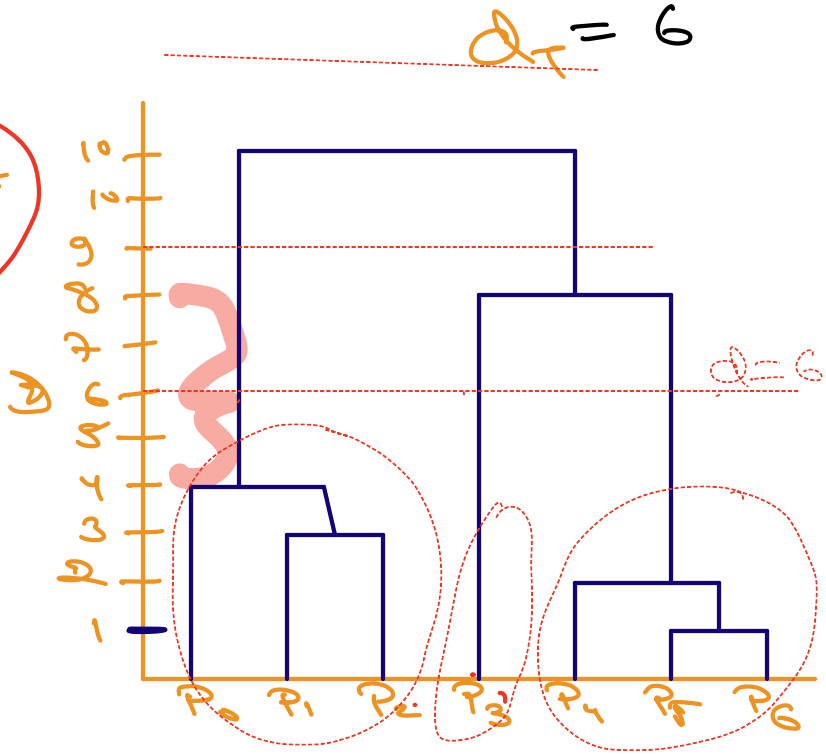
1. $n \rightarrow n\_clusters_0$
2. $n\_clusters_1 < n\_clusters_0$
   [we merge the clusters which are most similar]
3. Repeat till only 1 cluster is left

Step 1: Assume Every Single point is a Cluster

Step 2: Calculate proximity Matrix P(n x n)

Step 3: Using proximity Matrix
a) Merge the closest points
b) Update proximity matrix

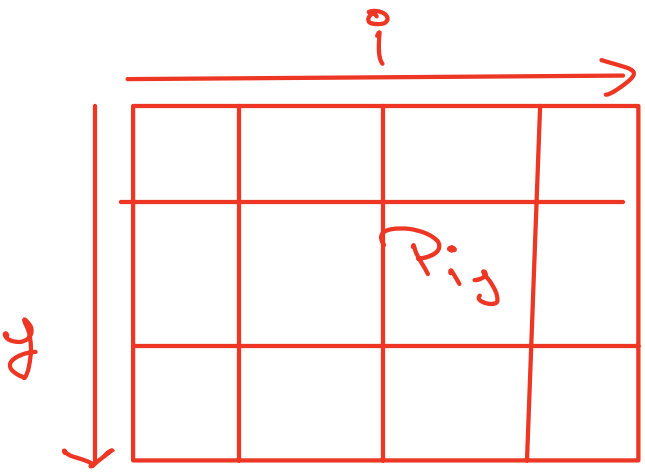




Dendogram

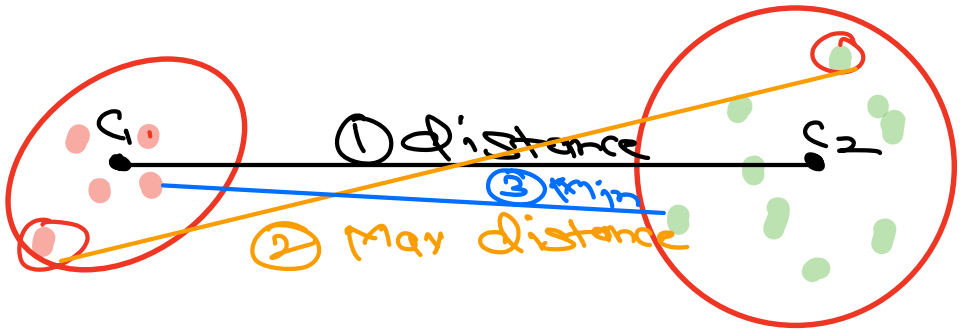
Proximity Matrix

- A matrix containing Similarity Values
- $P_{n \times n}$ - P_{ij} - $Sim(C_i, C_j)$



Methods to Calculate Similarity for Proximity Matrix

1. Distance (Euclidean) between centroids of Cluster
2. Maximum Distance between 2 Cluster
3. Minimum Distance b/w 2 Cluster
4. Average Distance $\frac{\sum_{x_i \in C_i} \sum_{x_j \in C_j} dist(x_i, x_j)}{|C_i||C_j|}$
5. Ward's Distance $\frac{\sum_{x_i \in C_i} \sum_{x_j \in C_j} dist(x_i, x_j)^2}{|C_i||C_j|}$



Iteration ①

	1	2	3	4	5	6
1						
2						
3						0.11
4						
5						
6						

↓

Iteration ②

	1	2	3	4	5
1			0		
2					
C_{36}					
4					
5					

Continue iteration till $P_{nxn} \Rightarrow P_{1x1}$

Limitation of Agglomerative

① Complexity : large Dataset

Space Complexity → $O(n^2)$

Time Complexity

→ Proximity Matrix Calculation $(n \times n) \times n \times d$

W.C → $O(n^3)$

K-means → S.C → $O(k)$ → k-clusters

T.C → $O(n \times k \times d)$ → k dimensions, k-clusters

Kmeans++ → $O(n \times k \times d \times i)$ → init_cent