Hierarchical Clustering in Machine Learning

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

The hierarchical clustering technique has two approaches:

1. **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

Why hierarchical clustering?

As we already have other clustering algorithms such as **K-Means Clustering**, then why we need hierarchical clustering? So, as we have seen in the K-means clustering that there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size. To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters.

In this topic, we will discuss the Agglomerative Hierarchical clustering algorithm.

Agglomerative Hierarchical clustering

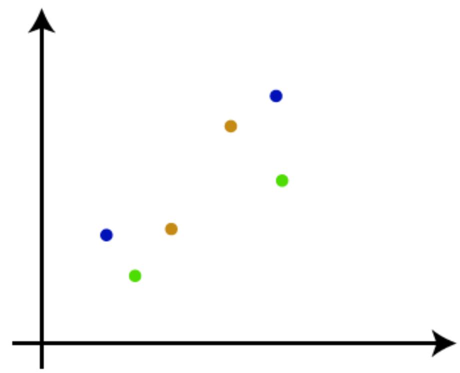
The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the **bottom-up approach**. It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets.

This hierarchy of clusters is represented in the form of the dendrogram.

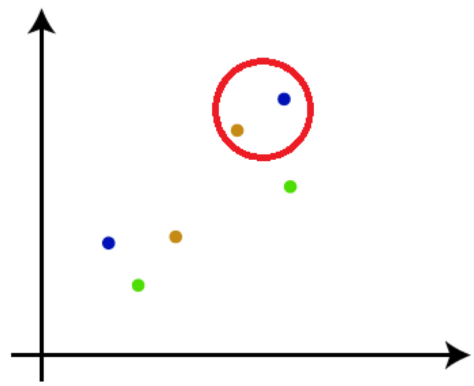
How the Agglomerative Hierarchical clustering Work?

The working of the AHC algorithm can be explained using the below steps:

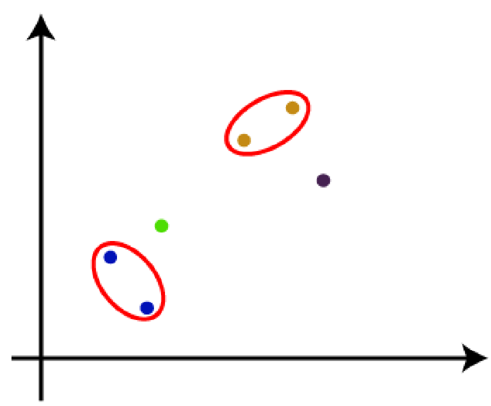
- **Step-1:** Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



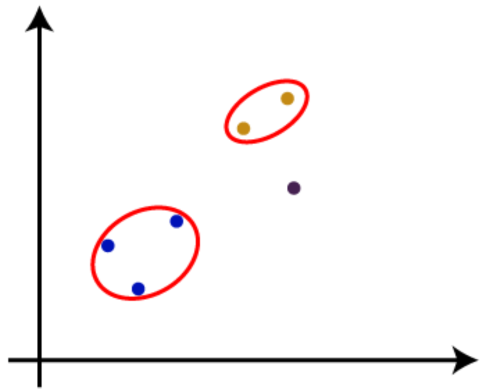
- **Step-2:** Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.

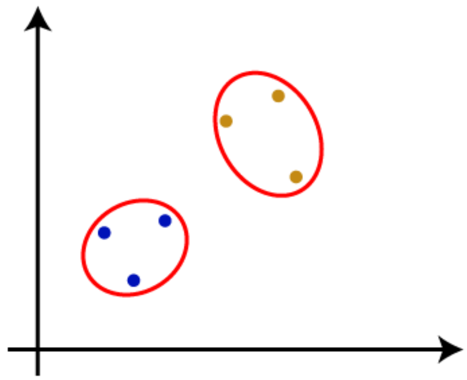


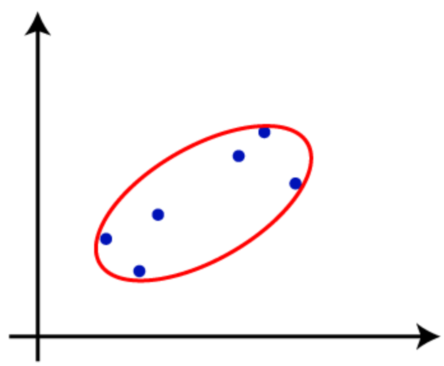
- **Step-3**: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



- **Step-4:** Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:





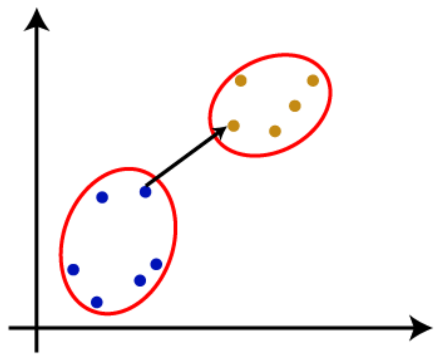


- **Step-5:** Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

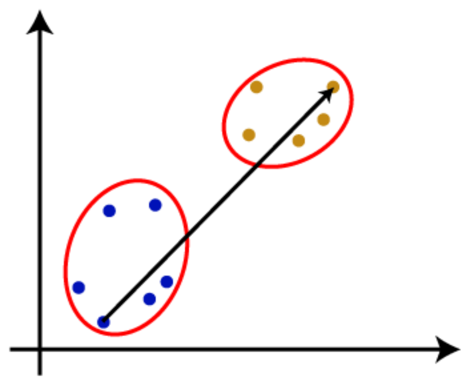
Measure for the distance between two clusters

As we have seen, the **closest distance** between the two clusters is crucial for the hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

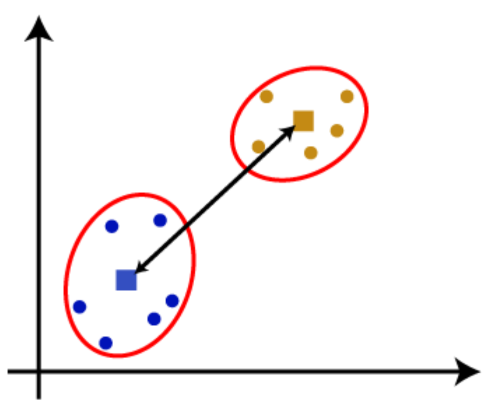
1. **Single Linkage:** It is the Shortest Distance between the closest points of the clusters. Consider the below image:



2. **Complete Linkage:** It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



3. **Average Linkage:** It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.
4. **Centroid Linkage:** It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:

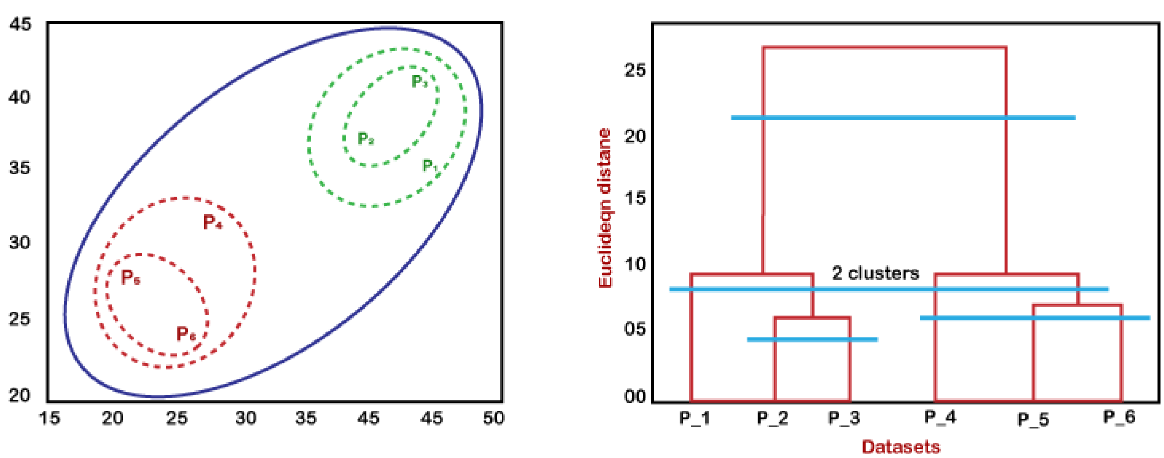


From the above-given approaches, we can apply any of them according to the type of problem or business requirement.

Woking of Dendrogram in Hierarchical clustering

The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs. In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.

The working of the dendrogram can be explained using the below diagram:



In the above diagram, the left part is showing how clusters are created in agglomerative clustering, and the right part is showing the corresponding dendrogram.

- As we have discussed above, firstly, the datapoints P2 and P3 combine together and form a cluster, correspondingly a dendrogram is created, which connects P2 and P3 with a rectangular shape. The hight is decided according to the Euclidean distance between the data points.
- In the next step, P5 and P6 form a cluster, and the corresponding dendrogram is created. It is higher than of previous, as the Euclidean distance between P5 and P6 is a little bit greater than the P2 and P3.
- Again, two new dendrograms are created that combine P1, P2, and P3 in one dendrogram, and P4, P5, and P6, in another dendrogram.
- At last, the final dendrogram is created that combines all the data points together.

We can cut the dendrogram tree structure at any level as per our requirement.

Example:

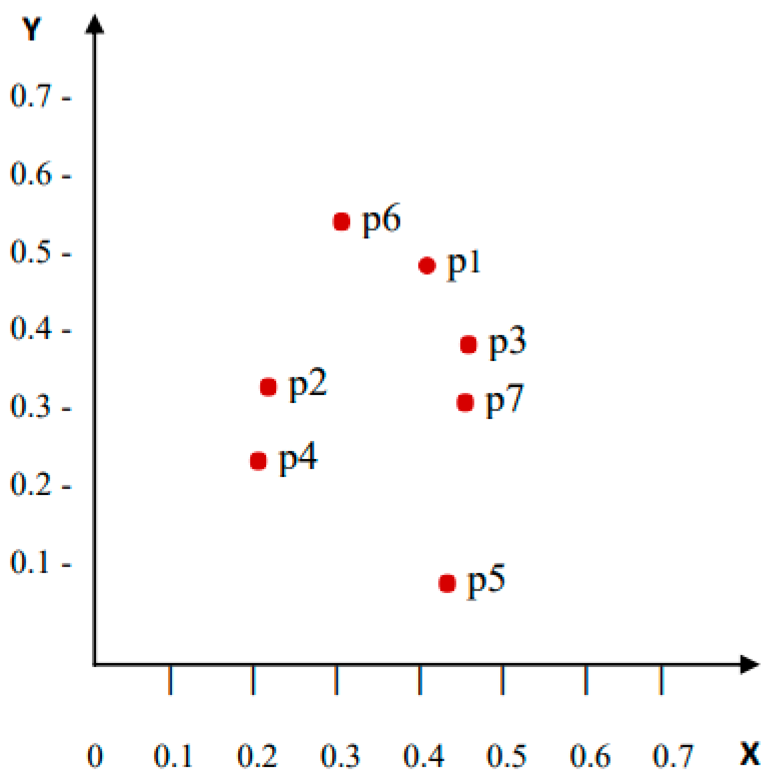
Single linkage clustering prepare clusters by calculating minimum distance between data points. Here we have taken 7 data points on the 2 dimensional space and prepare clusters. These data points are p1,p2,p3..p7 are shown in table 1. We have calculated all distances using MATLAB. The complete procedure for the single linkage clustering has describe below.

Table 1. Dataset with 7 objects

Object	X	Y
p1	0.40	0.50
p2	0.22	0.35
p3	0.45	0.40
p4	0.20	0.25
p5	0.42	0.10
p6	0.30	0.55
P7	0.45	0.32

4.1 Plotting Objects in 2D space

Here we plot attributes x and y in n dimensional space. In above dataset p1, p2, p3..p7 are 7 objects and x and y shows 2 dimensional space.



Calculating Distance Matrix In this step we calculate a distance matrix. This distance matrix is prepared by calculating the distance from each object to all other using Euclidean distance measure. Distance between two points i and j is

$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j1}|^2 + \ldots + |x_{in} - x_{jn}|^2}$$

$$d(p1, p2) = \sqrt{|x_{p1} - x_{p2}|^2 + |y_{p1} - y_{p2}|^2}$$

$$= \sqrt{|0.40 - 0.22|^2 + |0.50 - 0.35|^2}$$

$$= \sqrt{|0.18|^2 + |0.15|^2}$$

$$= \sqrt{0.0324 + 0.0225} \quad = \sqrt{0.0549}$$

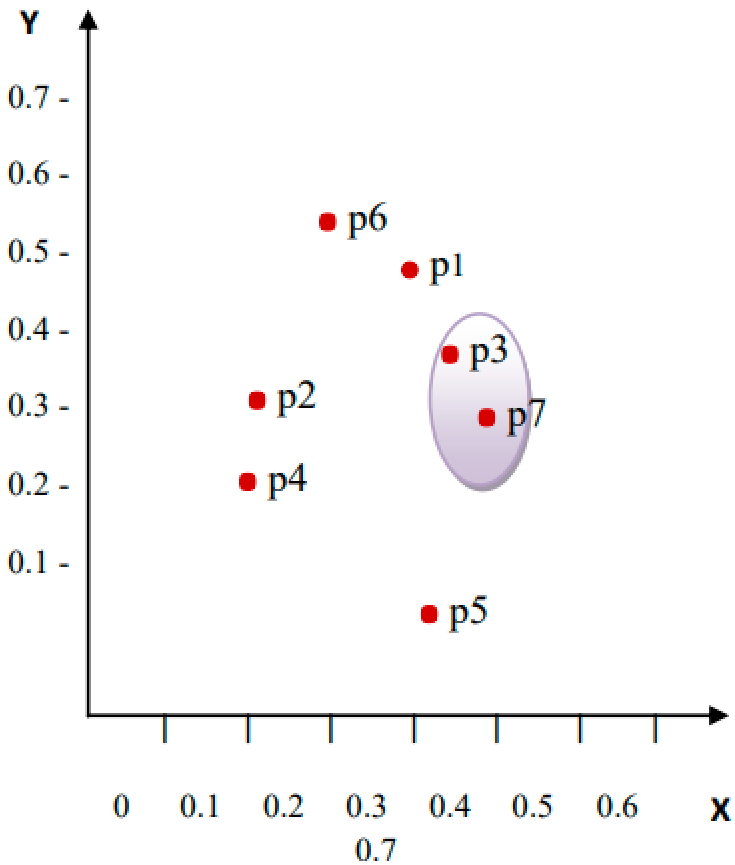
$$= \mathbf{0.2343}$$

Similarly we calculate the distance between all points.

Table 2

	p1	p2	p3	p4	p5	p6	p7
p1	0						
p2	0.2343	0					
p3	0.1118	0.2354	0				
p4	0.3202	0.1020	0.2915	0			
p5	0.4005	0.3202	0.3015	0.2663	0		
p6	0.1118	0.2154	0.2121	0.3162	0.4657	0	
p7	0.1868	0.2319	**0.0800**	0.2596	0.2220	0.2746	0

Assigning Objects into cluster Now we select two objects with the shortest distance in the matrix and then merge them together.



In Table 2 shortest distance points are p3 and p7. So that we merge these two points in one cluster by combining (p3, p7) together in a single cluster it became one entry. We again recalculate the distance from each point to new cluster (p3, p7). In single link method the proximity of two cluster is defined as the minimum distance between two clusters therefore the distance of (p3,p7) and p1 will calculated as

dis {(p3,p7),p1}=Min{distance(p3,p1),distance(p7,p1)}

=Min {0.1118, 0.1868}

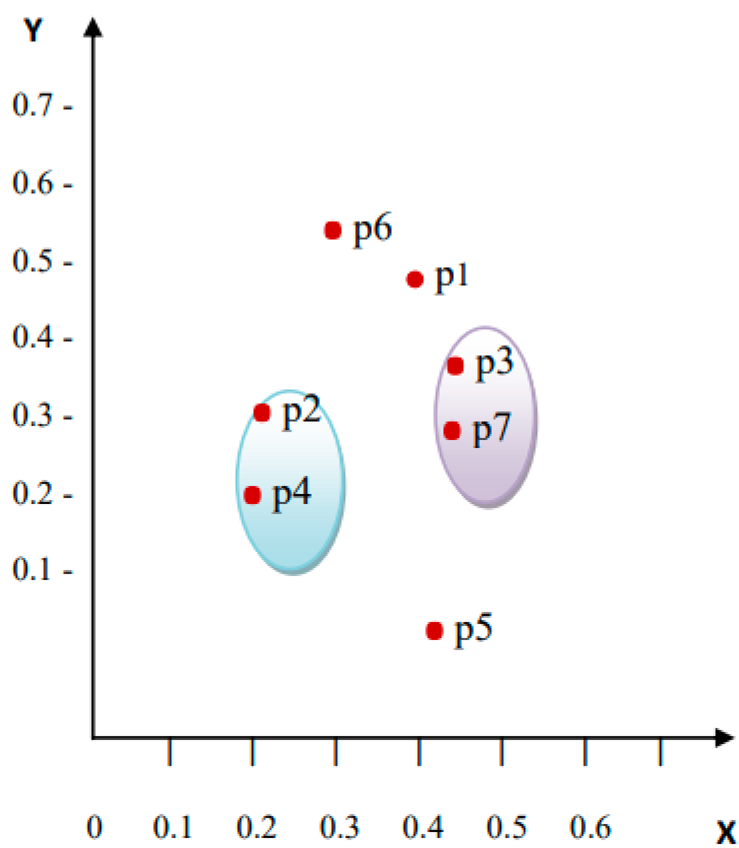
=**0.1118**

Similarly we calculate the distance between (p3, p7) to p2, p4, p5 and p6.

Table 3. Distance matrix for (p3, p7) to all points

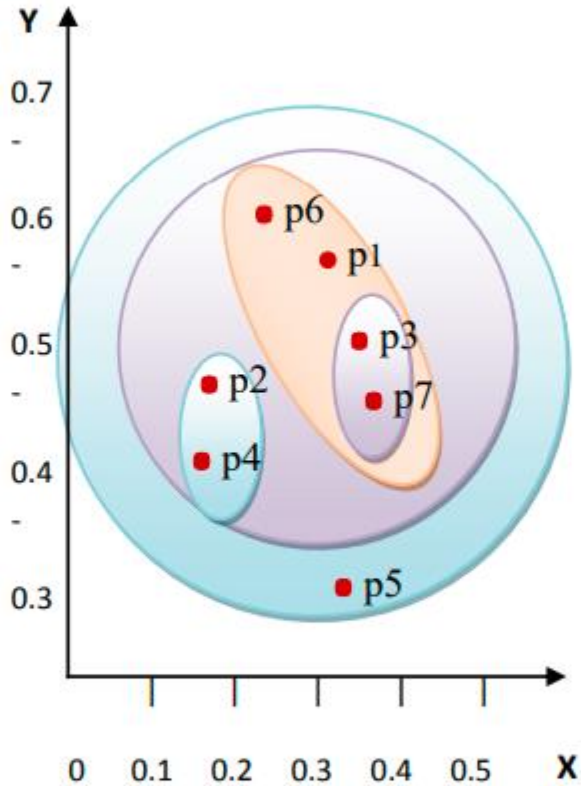
	p1	p2	(p3,p7)	p4	p5	p6
p1	0					
p2	0.2343	0				
(p3,p7)	0.1118	0.2319	0			
p4	0.3202	**0.1020**	0.2596	0		
p5	0.4005	0.3202	0.2220	0.2663	0	
p6	0.1118	0.2154	0.2121	0.3162	0.4657	0

Similarly we repeat above steps until all the objects are not assigned into appropriate clusters. In above matrix the smallest distance is 0.1020 between p2 and p4, so these points are merging together.

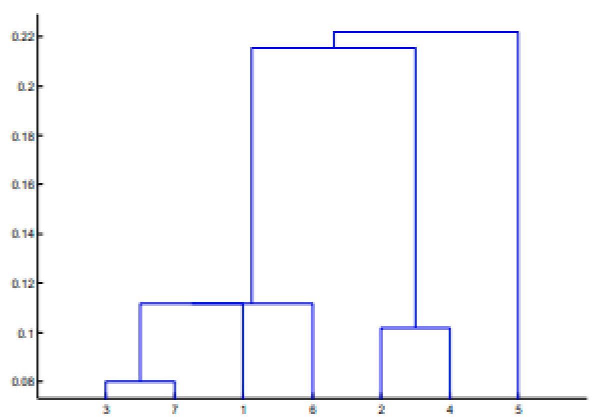


Similarly the points p2 and p4 are merged together and prepare a cluster which is shown in Fig. 9. This whole process of recalculating distance matrix and combining objects into

clusters are repeated. Thus we find final clusters



The various clusters prepared by above single linkage clustering method can also be displayed as a dendogram



Here we would like the data partitioned into several clusters for unsupervised learning. Therefore the process required to stop clustering at some point – either the user will specify the number of clusters he would like to have, or the process has to make a decision on its own. In above example if we stop at the threshold (distance) 0.12 then we have {(p3, p7, p1, p6), (p2, p4), (p5)} that means 3 clusters. But if we stop at threshold 0.1, we have only cluster (p3, p7) and (p2, p4).