**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:

**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.

1. **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](https://www.javatpoint.com/clustering-in-machine-learning) algorithms such as [**K-Means Clustering**](https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning), 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.

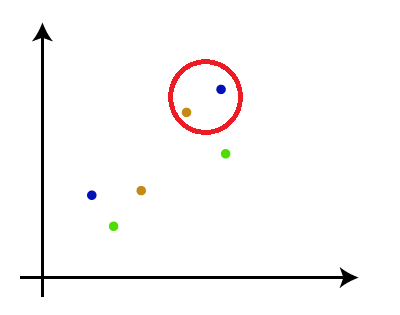
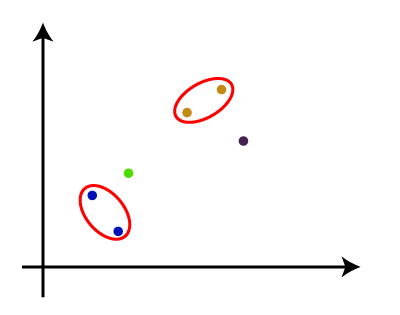
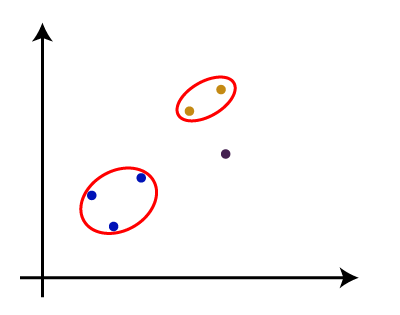
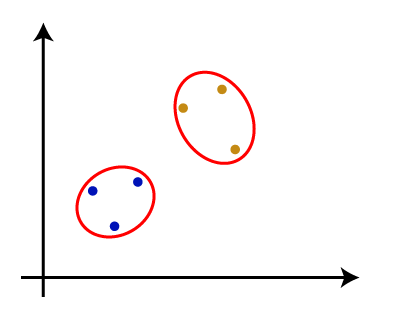
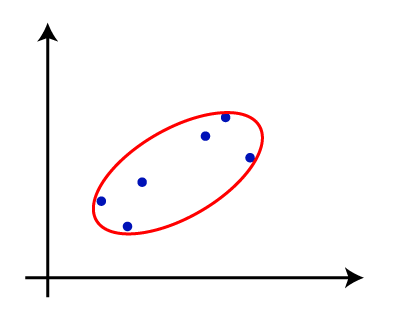
Agglomerative Hierarchical clustering

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?

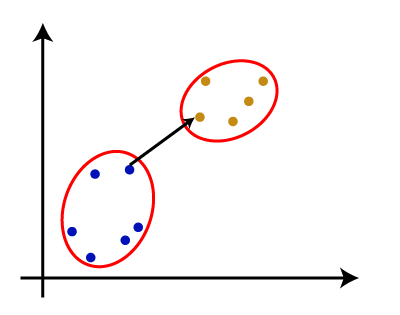
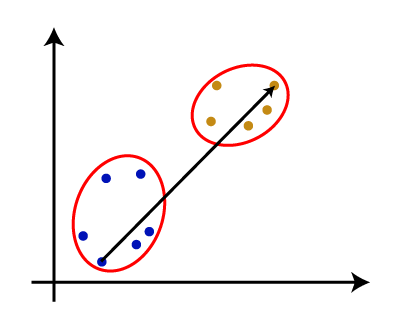
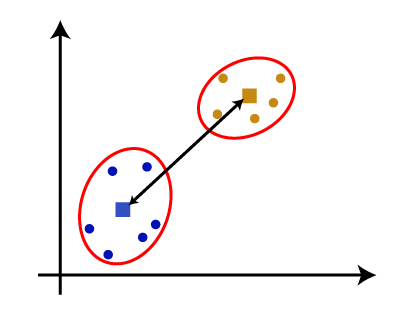
The working of the AHC algorithm can be explained using the below steps:

* **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.

Note: To better understand hierarchical clustering, it is advised to have a look on k-means clustering

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:

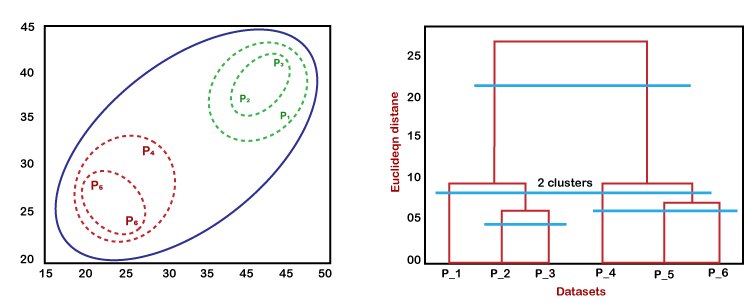
1. **Single Linkage:** It is the Shortest Distance between the closest points of the clusters. Consider the below image:  
   
2. **Complete Linkage:** It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than 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.

Divisive clustering **starts with one, all-inclusive cluster**. At each step, it **splits a cluster until each cluster contains a point** (or there are k clusters).

What is divisive dendrogram for hierarchical clustering?

Divisive hierarchical clustering: It's also known as DIANA (Divise Analysis) and it works in a top-down manner. The algorithm is an inverse order of AGNES. It begins with the root, in which all objects are included in a single cluster. At each step of iteration, the most heterogeneous cluster is divided into two.

## Divisive Clustering Example

The following is an example of Divisive Clustering.

| **Distance** | **a** | **b** | **c** | **d** | **e** |
| --- | --- | --- | --- | --- | --- |
| a | 0 | 2 | 6 | 10 | 9 |
| b | 2 | 0 | 5 | 9 | 8 |
| c | 6 | 5 | 0 | 4 | 5 |
| d | 10 | 9 | 4 | 0 | 3 |
| e | 9 | 8 | 5 | 3 | 0 |

**Step 1.** Split whole data into 2 clusters

1. Who hates other members the most? (in Average)
   * aa to others: mean(2,6,10,9)=6.75 →amean(2,6,10,9)=6.75 →a goes out! (Divide aa into a new cluster)
   * bb to others: mean(2,5,9,8)=6.0mean(2,5,9,8)=6.0
   * cc to others: mean(6,5,4,5)=5.0mean(6,5,4,5)=5.0
   * dd to others: mean(10,9,4,3)=6.5mean(10,9,4,3)=6.5
   * ee to others: mean(9,8,5,3)=6.25mean(9,8,5,3)=6.25
2. Everyone in the old party asks himself: “In average, do I hate others in old party more than hating the members in the new party?”
   * If the answer is “Yes”, then he will also go to the new party.

|  | α=α=**distance to the old party** | β=β=**distance to the new party** | α−βα−β |
| --- | --- | --- | --- |
| b | 5+9+83=7.335+9+83=7.33 | 2 | >0>0 (bb also goes out!) |
| c | 5+4+53=4.675+4+53=4.67 | 6 | <0<0 |
| d | 9+4+33=5.339+4+33=5.33 | 10 | <0<0 |
| e | 8+5+33=5.338+5+33=5.33 | 9 | <0<0 |

1. Everyone in the old party ask himself the same question as above again and again until everyone’s got the answer “No”.

|  | α=α=**distance to the old party** | β=β=**distance to the new party** | α−βα−β |
| --- | --- | --- | --- |
| c | … | … | <0<0 |
| d | … | … | <0<0 |
| e | … | … | <0<0 |

**Step 2.** Choose a current cluster and split it as in **Step 1.**

1. Choose a current cluster
   * If split the cluster with the largest number of members, then the cluster {c,d,ec,d,e} will be split.
   * If split the cluster with the largest diameter, then the cluster {c,d,ec,d,e} will be split.

| **cluster** | **diameter** |
| --- | --- |
| {a,b} | 2 |
| {c,d,e} | 5 |

1. Split the chosen cluster as in **Step 1.**

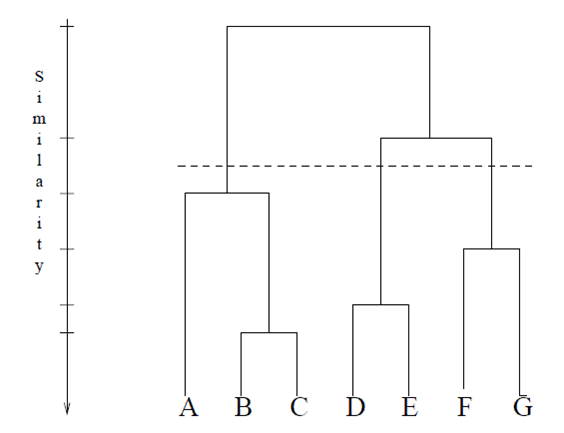
**Step 3.** Repeat **Step 2.** until each cluster contains a point (or there are kk clusters)

**Learn with an example : Hierarchical Clustering**

What is hierarchical clustering (agglomerative) ?

Clustering is a data mining technique to group a set of objects in a way such that objects in the same cluster are more similar to each other than to those in other clusters.

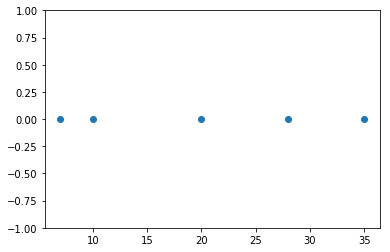
In hierarchical clustering, we assign each object (data point) to a separate cluster. Then compute the distance (similarity) between each of the clusters and join the two most similar clusters. Let’s understand further by solving an example.



Dendogram

Objective : For the one dimensional data set {7,10,20,28,35}, perform hierarchical clustering and plot the dendogram to visualize it.

Solution : First, let’s the visualize the data.

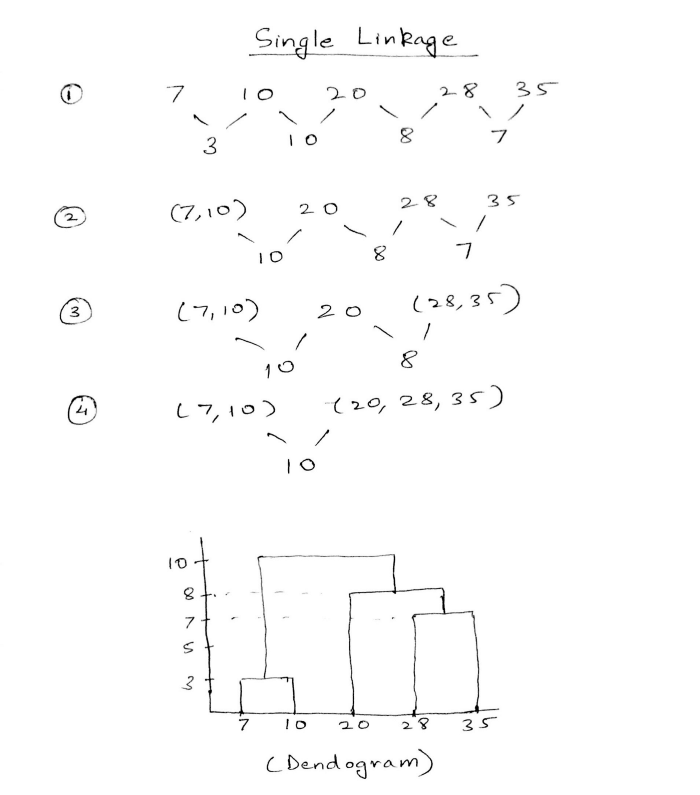


Observing the plot above, we can intuitively conclude that:

1. The first two points (7 and 10) are close to each other and should be in the same cluster
2. Also, the last two points (28 and 35) are close to each other and should be in the same cluster
3. Cluster of the center point (20) is not easy to conclude

Let’s solve the problem by hand using both the types of agglomerative hierarchical clustering :

1. Single Linkage : In single link hierarchical clustering, we merge in each step the two clusters, whose two closest members have the smallest distance.

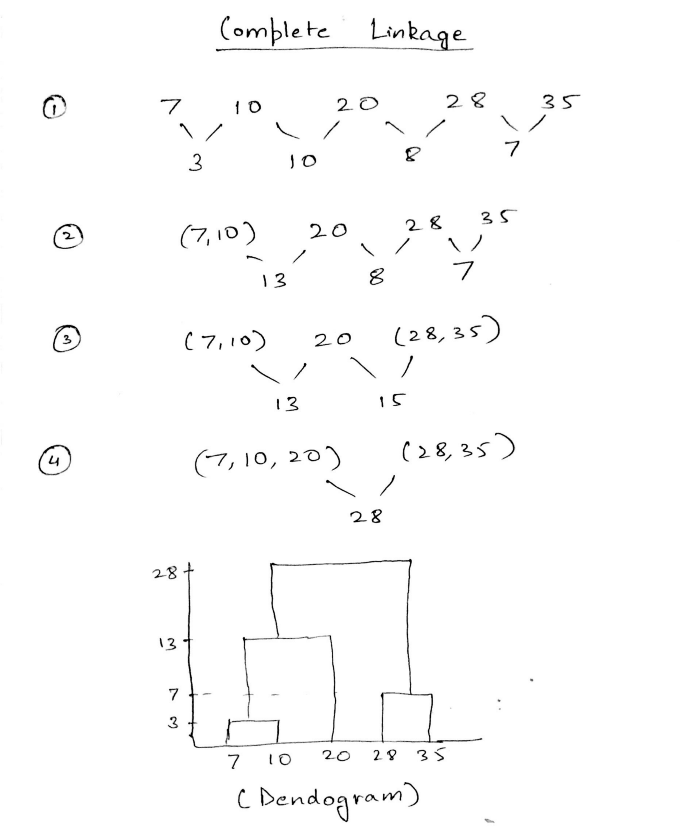


Using single linkage two clusters are formed :

Cluster 1 : (7,10)

Cluster 2 : (20,28,35)

2. Complete Linkage : In complete link hierarchical clustering, we merge in the members of the clusters in each step, which provide the smallest maximum pairwise distance.



Using complete linkage two clusters are formed :

Cluster 1 : (7,10,20)

Cluster 2 : (28,35)

Conclusion : Hierarchical clustering is mostly used when the application requires a hierarchy, e.g creation of a taxonomy. However, they are expensive in terms of their computational and storage requirements.