Chapter 4 : Part 2

**Hierarchical Clustering**

**4.2.1 Hierarchical Clustering**

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.

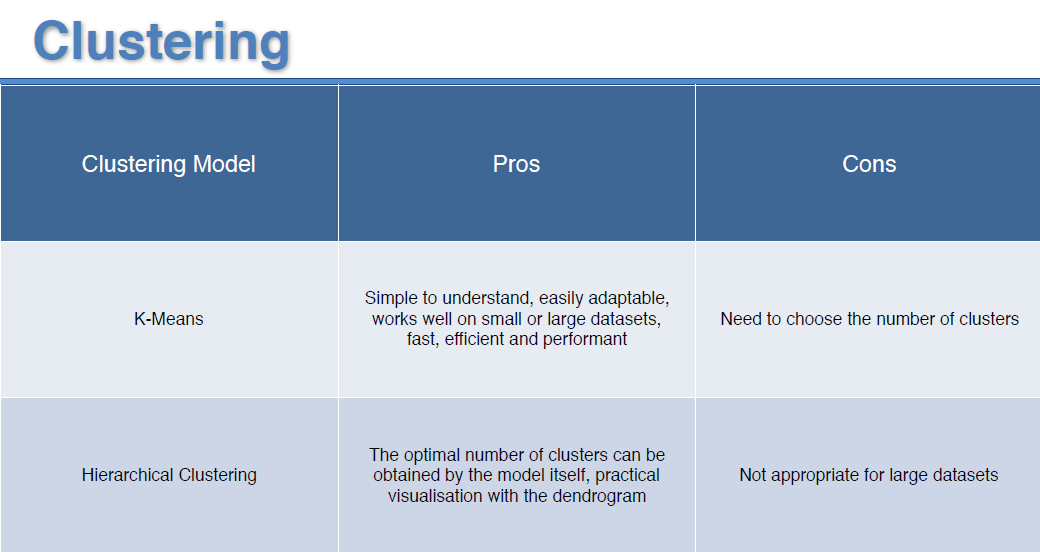
* Dendrogram: In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.
* Sometimes the results of ***K-Means Clustering*** and ***Hierarchical Clustering*** may look similar, but they both differ depending on how they work.
* There is no requirement to predetermine the ***number of clusters*** as we did in the K-Means algorithm.
* Agglomerative and Divisive clustering: 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***.
* *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?
* As we have seen in the ***K-means clustering*** that we need 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.

**4.2.2 Agglomerative Hierarchical clustering**

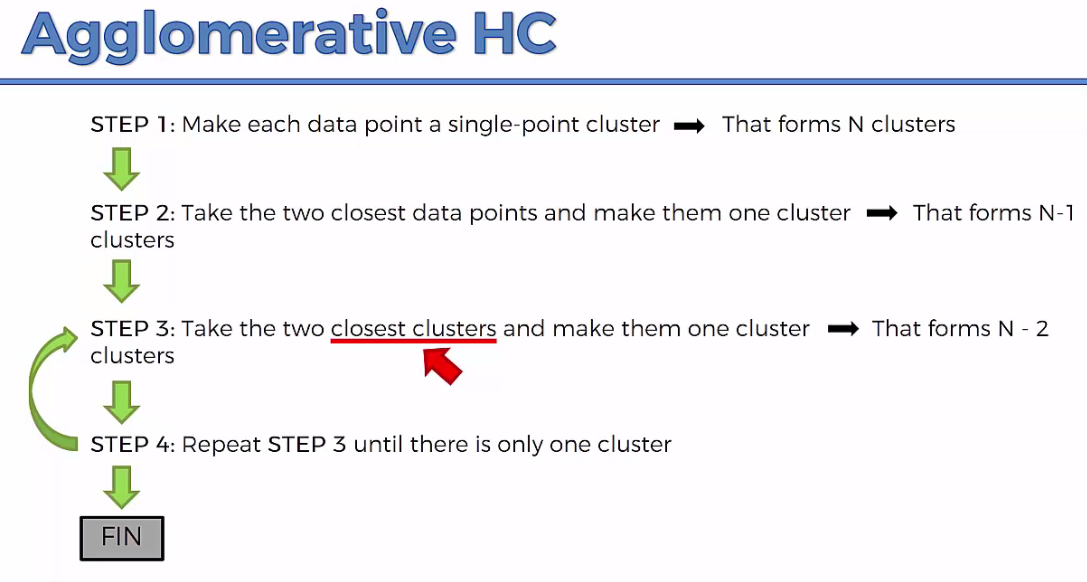
The Agglomerative Hierarchical Clustering algorithm is a popular example of ***HCA***. To group the datasets into clusters, it follows the Bottom-Up approach. It means, this algorithm considers ***each dat-point as a single cluster*** at the beginning, and then start COMBINING the *closest pair* of clusters 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.

**Clustering Pros & Cons**



* How Agglomerative HCA Work: The working of the ***AHC*** algorithm can be explained using the below steps:



* The way the hierarchical clustering algorithm works is that it *maintains a memory* of how we went through following process and that *memory* is *stored* in a DENDROGRAM.

|  |  |
| --- | --- |
| * 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***. | Hierarchical Clustering in Machine Learning |
| * Step-2: Take *two* closest *data-points* or *clusters* and *merge* *them* to form one cluster. So, there will now be N-1 clusters. | Hierarchical Clustering in Machine Learning |
| * 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: | Hierarchical Clustering in Machine Learning |
| Hierarchical Clustering in Machine Learning Hierarchical Clustering in Machine Learning Hierarchical Clustering in Machine Learning | |
| * Step-5: Once all the clusters are combined into one big cluster, develop the ***Dendrogram*** to divide the clusters as per the problem. | |

**4.2.3 Distance between two clusters**

* Linkage Methods: 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:

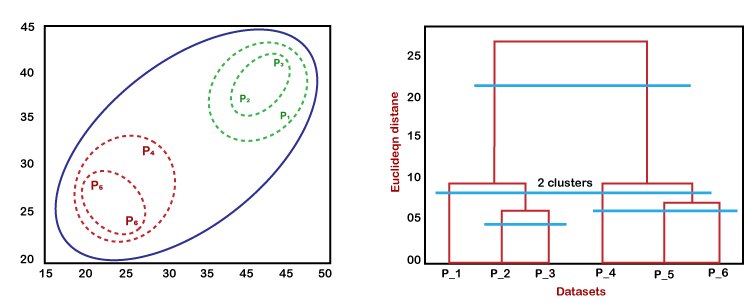
|  |  |
| --- | --- |
| 1. Single Linkage: It is the *Shortest Distance* between the *closest* *points* of the *clusters*. Consider the beside image: | Hierarchical Clustering in Machine Learning |
| 1. Complete Linkage: It is the *farthest* *distance* between the *two* *points* of two different *clusters*. It is one of the popular linkage methods as it forms tighter clusters than single-linkage. | Hierarchical Clustering in Machine Learning |
| 1. Average Linkage: It is the linkage method in which the *distance between each pair* of ***data-points*** is added up and then divided by the ***total*** ***number*** of ***data-points*** to calculate the average distance between two clusters. It is also one of the most popular linkage methods. | |
| 1. Centroid Linkage: It is the linkage method in which the ***distance*** ***between*** the ***centroid*** of the ***clusters*** is calculated. Consider the beside image: | Hierarchical Clustering in Machine Learning |

**4.2.4 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.

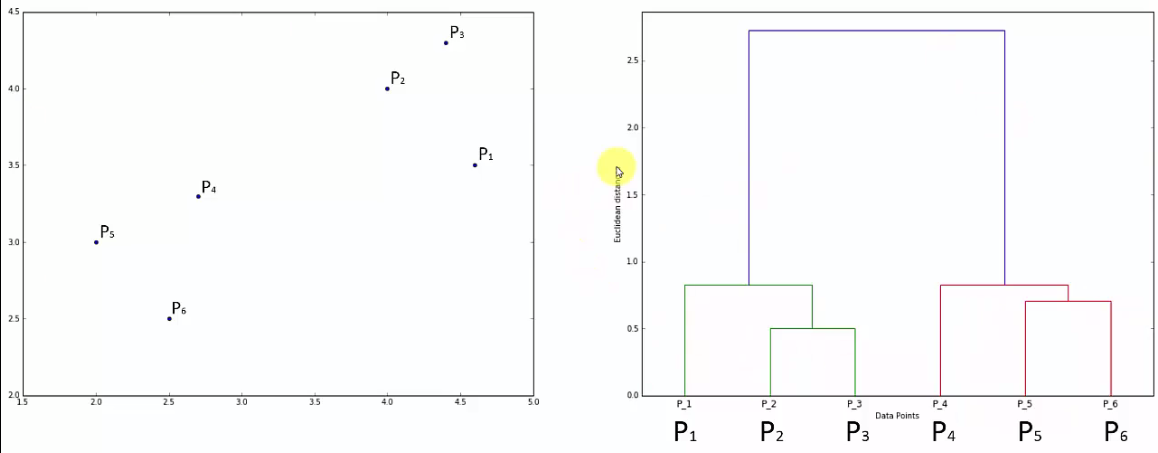
The working of the dendrogram can be explained using the below diagram:



* 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***.

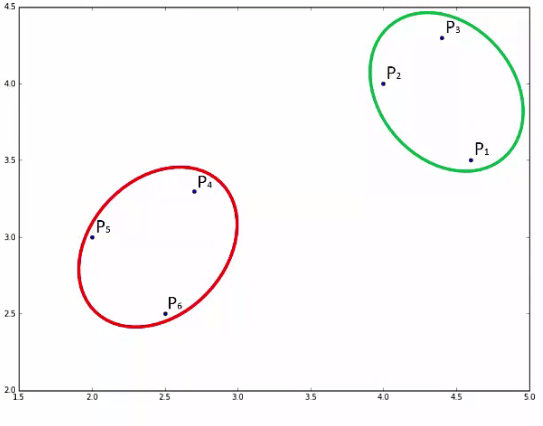
1. Firstly, the datapoints ***P2*** and ***P3*** combine together and form a *cluster*, correspondingly a *dendrogram* is *created*, which connects ***P2*** and ***P3*** with a ***rectangular*** ***shape***. The height is decided according to the *Euclidean* *distance* between the *data* *points*.
2. 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***.
3. Again, two new dendrograms are created that combine ***P1***, ***P2***, and ***P3*** in one dendrogram, and ***P4***, ***P5***, and ***P6***, in another dendrogram.
4. 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.
* Dendrogram contains the *memory* of the *hierarchical* *clustering* *algorithm*. You can understand in which order these classes were formed. Just by looking at the Dendrogram. Here I've got an example, this is the actual example generated by computer using HCA.

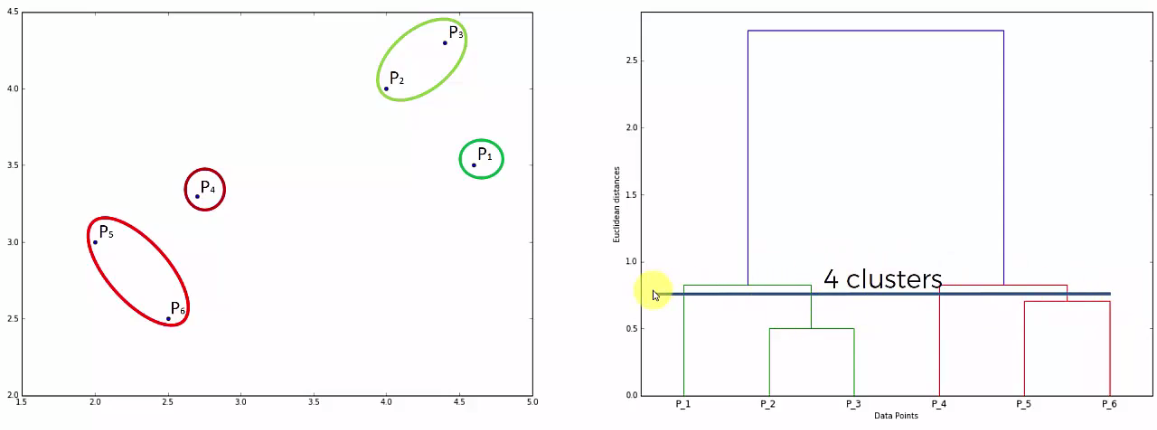


|  |  |
| --- | --- |
| * How do we get to that right number of clusters: Now we have to look at the horizontal levels and set thresholds so we can set heights thresholds or distance actually distance thresholds are also called dissimilarity thresholds because this vertical axis measures the Euclidean distance between points which also represents the *dissimilarity* between *them* or *points* or *clusters*. * So what we can do is *set a threshold* for all *dissimilarity* and we can say that we don't want dissimilarity to be greater than a certain level. |  |

* We were setting the dissimilarity threshold by saying that anything if we come across clusters that are above this *certain threshold* (i.e. above certain Euclidian distance) so we ***don't want within a cluster*** to have dissimilarity above this threshold. Then we end up with two clusters. The clysters are shown in the following figure:

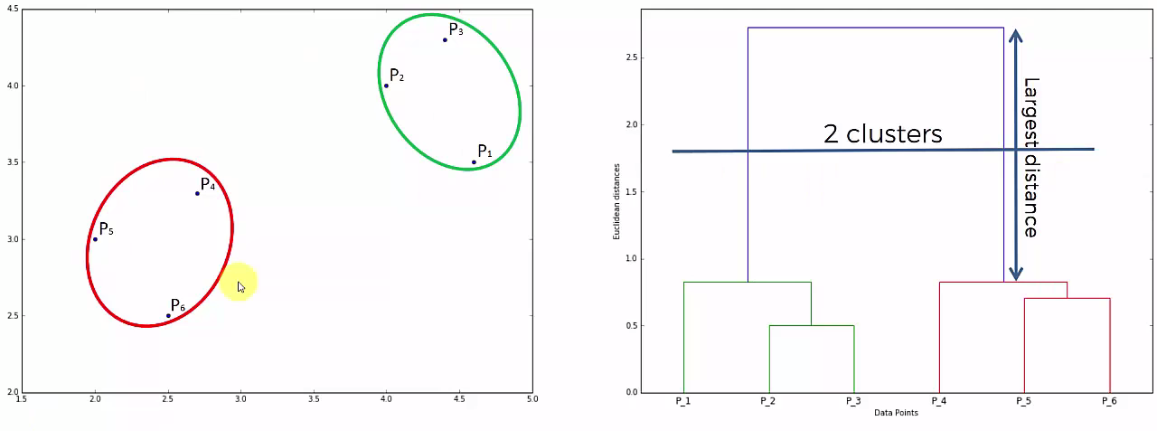


* What this is doing is it is not allowing any ***clusters*** that would have ***dissimilarity*** of ***greater*** than the *threshold value* *(1.7 in the figure)* within them.
* Calculating clusters numbers: You can quickly tell how many clusters you will have *at a certain threshold* by just looking at how many vertical lines this horizontal threshold actually crosses.
* So in above Dendrogram figure, the horizontal threshold line ***crosses two vertical line***, hence we get two clusters.
* In following case we have 4-clusters:



* Finding Optimal number of Clusters: Large vertical line means "Long distance/gap/void between the clusters". So we need to find a ***"threshold" that crosses maximum number "large vertical lines"***.
* We cannot consider the vertical lines that crosses "hypothetical extended horizontal line that forms dendrogram".

|  |  |
| --- | --- |
| * One of the standard approaches is just to look for the highest vertical distance that you can find in dendrogram. Any line that will not cross any horizontal lines. * For example: line at ***P2*** and ***P3*** can be considered but ***P1*** cannot be considered because it crosses ***hypothetical horizontal line (extended)*** that joins ***P2*** and ***P3***. Because it does not represents the distance between ***{P2, P3} cluster***, it actually distance between the single ***points P2***, ***P3***; and the actual distance between ***{P2, P3}*** and ***P1*** is the short vertical line (top on P2, P3). * So we have to find the ***largest vertical line that does not cross any of the existed horizontal lines that forms the dendrogram***. |  |



* One More Example:

