1. What is Clustering - Introduction
2. Intuition in Hierarchical clustering
3. Application of Hierarchical Clustering in the real world.
4. Why is the name Hierarchical Clustering?
5. Assumptions in Hierarchical Clustering
6. Strength and Weakness of the Hierarchical Clustering
7. Evaluation metrics in Hierarchical clustering
8. How Hierarchical Clustering works
9. Overfitting and Underfitting
10. Case Study

Introduction to Clustering

Clustering is one of the most used data analysis techniques to find traits in data. The goal is to find homogeneous subgroups such that data points in each cluster are as much identical as possible. In other words, it aims at grouping the data points having analogous properties and/or features, while data points in different groups should have highly eccentric properties and/or features.

In the nutshell, the aim is to integrate groups with similar traits and allocate them into clusters.

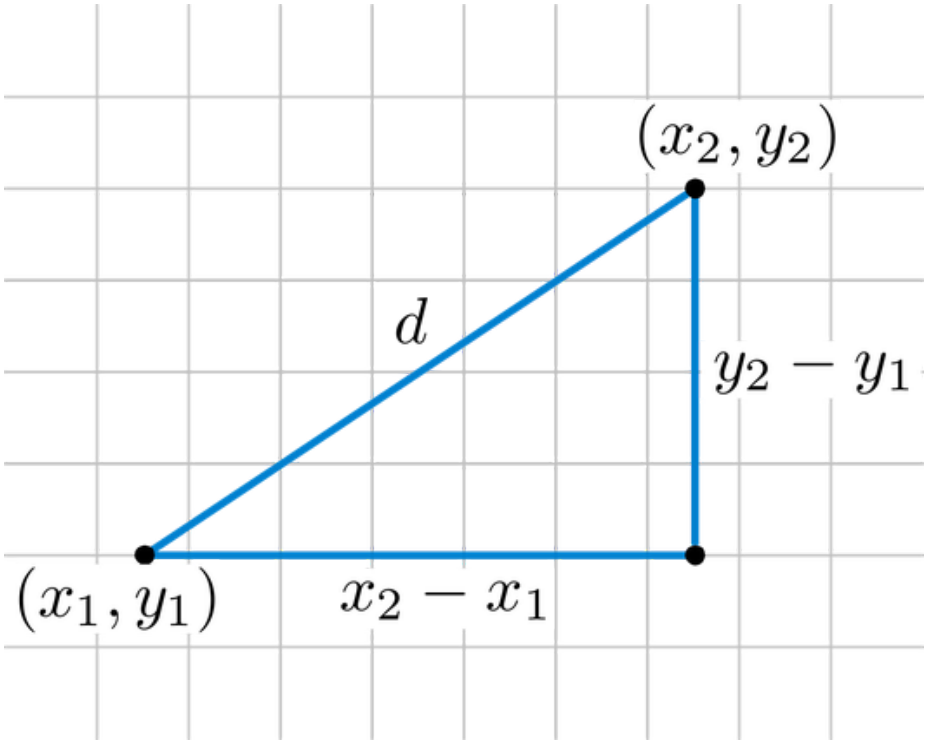
When we tend to cluster data points, we expect to group data with similar features. Since there isn't a response variable, we can classify clustering as an application of Unsupervised Learning, which means it seeks to find relationships between the n observations without being trained.

Clustering Distance Measures

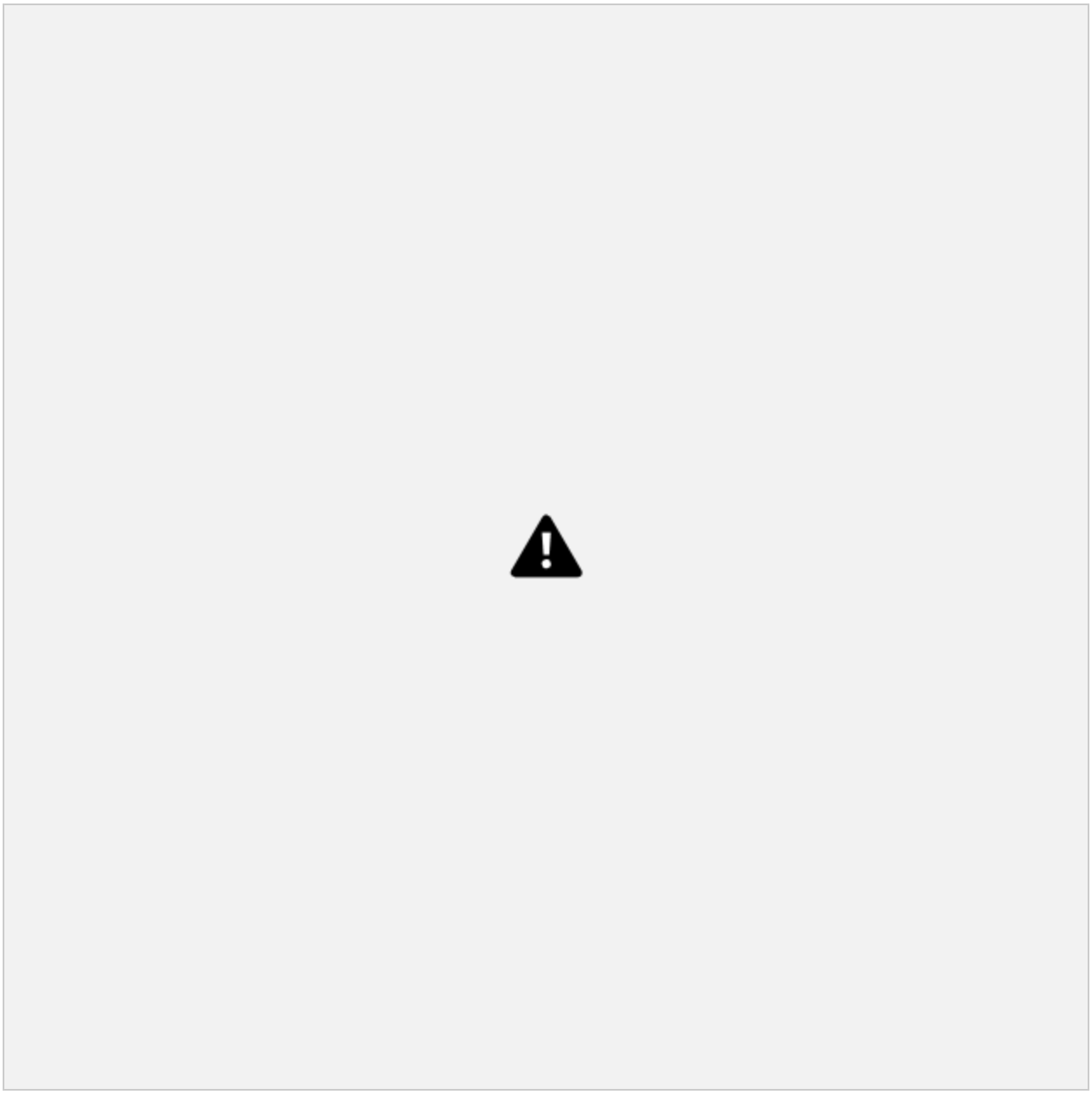
The classification often requires the computation of the distance between each pair of data points. The distance measure method helps us to understand the similarity between two data points and it is decisive in determining the shape of the cluster.

The traditional methods to calculate distance are:

- Euclidean Distance - It's defined as the square root of the sum of the squared differences between two points.



- Manhattan Distance - This is the distance between real vectors using the sum of their absolute difference.



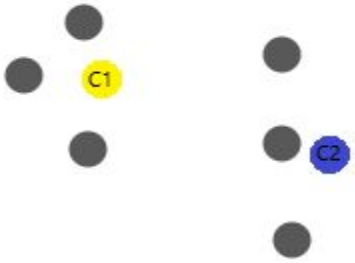
Intuition in Hierarchical Clustering

Before diving deep into Hierarchical Clustering, it’s important to brief about K-Means Clustering.

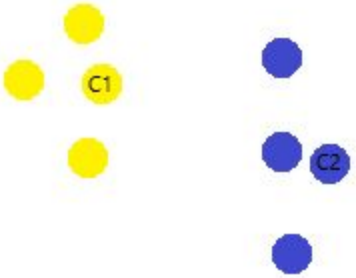
The steps involved in K-means clustering are:

Classify data among some “K” number of clusters.

1. Initialize K-points.
2. Categorize each item to its closest mean.



3. Update coordinates of mean that is average of items categorized in mean so far.



4. Repeat the above steps until our algorithm converges.

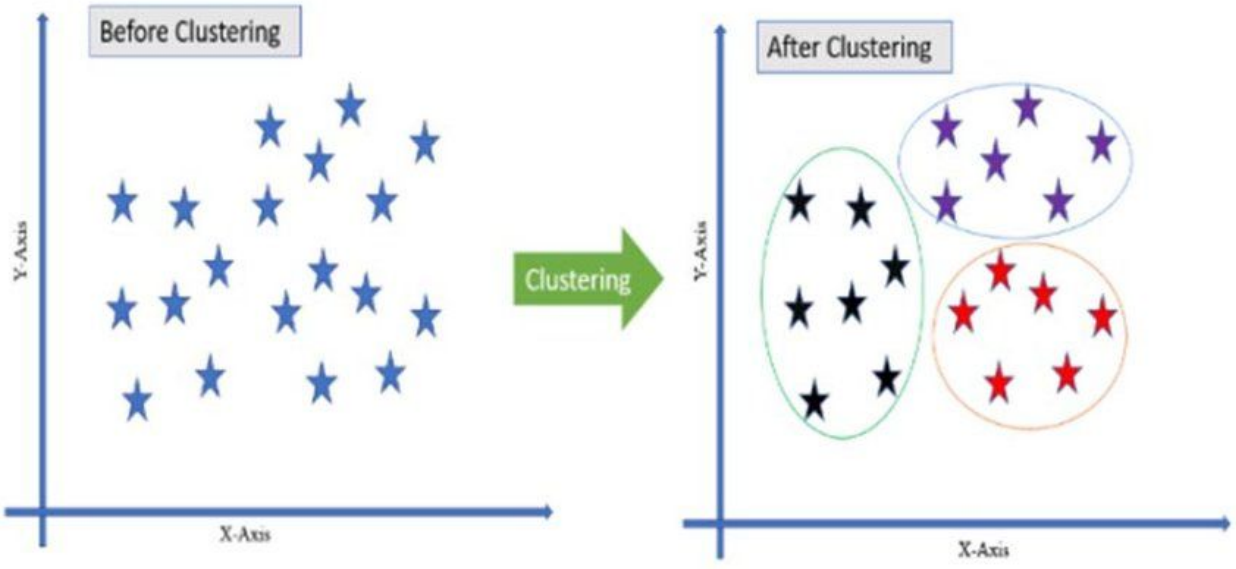
It is a repeating process. It will keep on running until the centroids of newly formed clusters do not amend.

But there are certain obstacles with K-means. It always tries to create The hierarchical same sized clusters. Also, deciding the number of clusters even before beginning the algorithm is quite tedious.

The hierarchical clustering is used to overcome the drawbacks of K-Means clustering. It bridges the gap one needs for almost perfect clustering techniques.

Let’s say, you’re an active Spotify user, suppose you logged in one day and found the archive to be cluttered and vague. How tedious it would be to find a song similar to your instant mood.

That’s where a technique like clustering comes into play, it’ll help to cluster out catalog according to some features/traits thus providing a seamless and customized experience to users.



Application of Hierarchical Clustering in the real world.

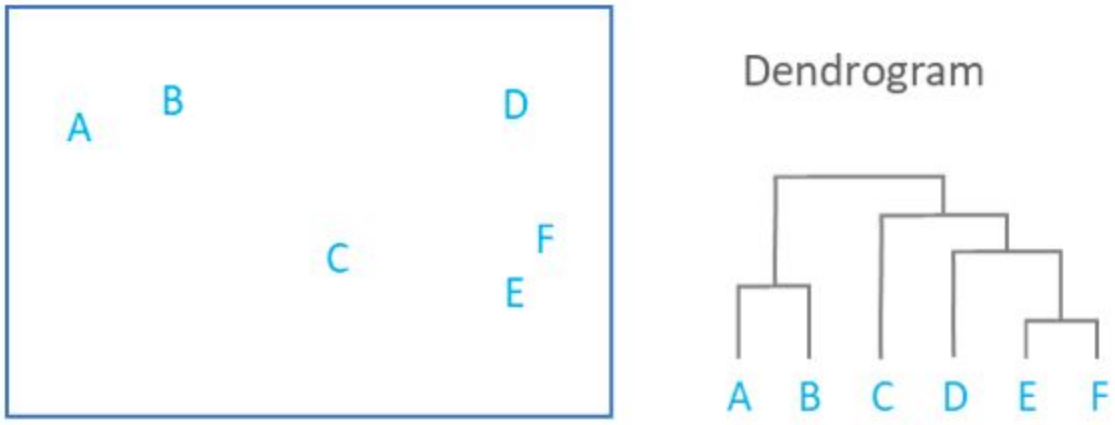
Clustering is often employed in realms of:

- *wireless sensor networks* - Clustering helps to find cluster heads which collect all data in its respective group.
- *customer segmentation* - Based on several factors such as customer income, customer spending one can easily determine the cluster of customers who will be interested in buying the particular goods.
- *recommendation systems* - Clustering helps to find data with similar traits thus creating personalized content for users.
- *search engines* - Clustering plays an important role in search engines. When a search is made, the search results are expected to be grouped and we often use clustering.
- *diagnostic systems* - It is used in creating smarter medical decision support systems based on similar features.
- *Grouping Text documents*- The technique can help us to group text having similar traits.
- *Outlier Detection* - Hierarchical clustering is also used for outlier detection.

Why is the name Hierarchical Clustering?

The hierarchical clustering approach is achieved via grouping data into a tree of clusters. In the beginning, every data point is treated as separate clusters.

Due to the ranked and stratified nature of this algorithm, it is known as hierarchical clustering.

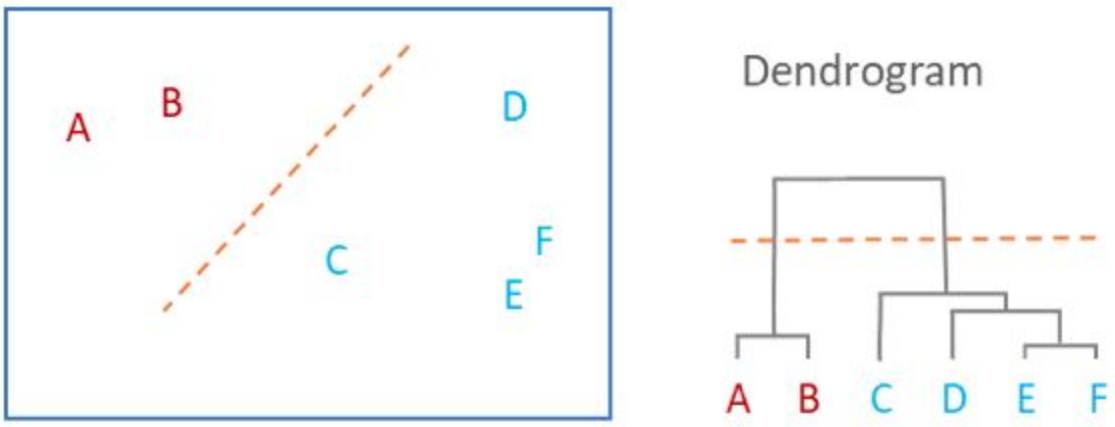


In Hierarchical Clustering, the goal is to produce a hierarchical series of nested clusters. A tree-like diagram that represents the sequence of merges or split known as Dendrogram is employed, to check similarities in clusters.

Once points in the dendrogram are employed, we can further merge them to create an inverted tree-like structure.

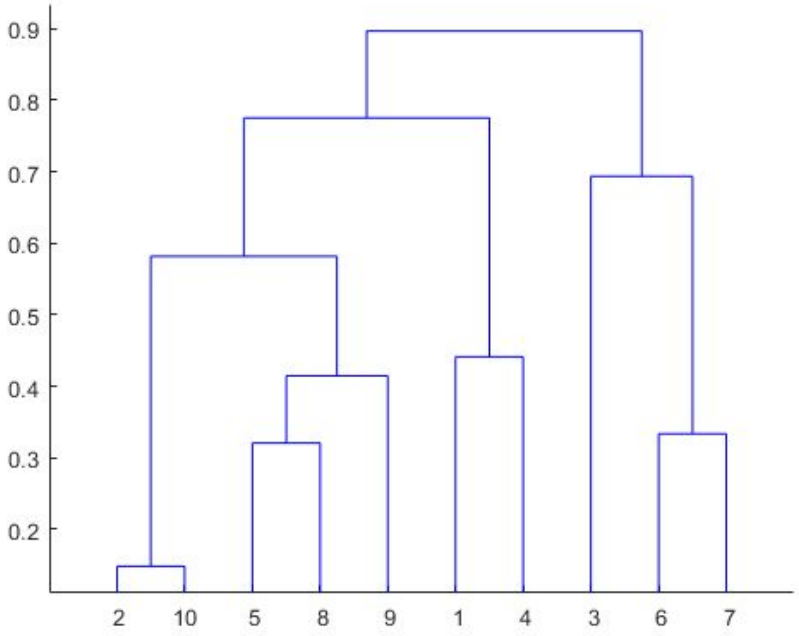
The dendrogram is employed to allocate objects to the clusters.

A dendrogram mainly focuses on the height at which any two objects are merged. The height of the dendrogram indicates the order in which clusters were merged. The trait of an informative dendrogram can be where heights reflect the measure between clusters.



A dendrogram can be analyzed by allocating a horizontal line between the points. In the illustration, A and B belong to cluster 1 and the rest to cluster 2.

The ordering of the leaves is supposed to be arbitrary, as is their horizontal position. The heights of the internal nodes may be arbitrary or may be related to the metric information used to form the clustering.



Dendrograms never assures how many clusters are made using algorithms, the dendrogram is true only when data satisfies Ultrametric Tree Inequality i.e.

dist(a,c) ≤ max(dist(a,b), dist(b,c)) where a,b,c are three data points.

Though they never assure the number of clusters, they do assure the relationship between clusters. A distance matrix is maintained by the dendrogram, where height of each block demonstrated the distance between clusters. The distance between data points represents dissimilarities between them, more far data points, fewer similarities they share.

Assumptions in Hierarchical Clustering

It is always preferred to understand the assumption behind the methods to get an idea about the strength and weakness of each method, that could further help to decide when to use each method under what conditions.

Scale/Standardize the data: It is always preferred to standardize the data before inputting it into a model as it helps in developing a uniform distribution and steeps down the probability of errors.

Spherical Clusters: While implementing Hierarchical Clustering it is assumed that the clusters are spherical (where the radius is equal to the distance between the centroid and farthest data point).

Bigger Clusters: While adding data points to any cluster, Hierarchical clustering gives more weight to the bigger clusters.

Therefore, it is advised to keep the number of clusters as optimal as possible.

Space and Time complexity for Hierarchical Clustering

Space complexity: We need to store the proximity matrix in RAM thus its space complexity is very high.

Space complexity = $O(n^2)$ where n is the number of data points.

Time complexity: We need to perform n iterations and with each iteration, we are supposed to update the proximity matrix, thus the time complexity is high. The time complexity is the order of the cube of n.

Time complexity = $O(n^3)$ where n is the number of data points.

Strength and Weakness of the Hierarchical Clustering

Pros

- **No hyperparameters are needed.**
- **Easy to implement.**
- **Better representability**: Can be represented pictorially using Dendrograms.

- **Not dependent on Initial Values:** Always generates the same clusters unlike that in K-Means where clustering depends on how the centroids are initiated.

Cons

- **Computation speed:** It is a slower algorithm compared to k-means. Hierarchical clustering takes a long time to run, especially for large data sets**.**
- There is no mathematical objective for Hierarchical clustering.
- **Complexity:** It caters to high space and time complexity.
- **Huge Data**: Hierarchical Clustering can't be employed for large data.

How Hierarchical Clustering works

Hierarchical clustering can be achieved following these steps:

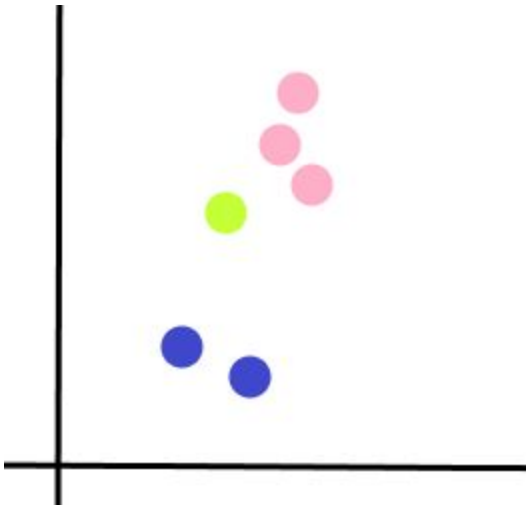
1. Identify the 2 clusters which can be closest together.
2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged.

Proximity Matrix: The proximity matrix contains the distance between each point. When we create clusters, we need to update our proximity matrix. The distance between each point can be calculated either via Euclidean distance or Manhattan distance or cosine distance.

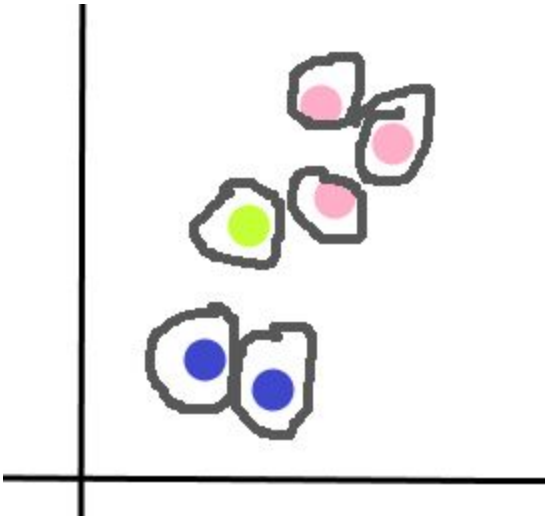
There are further two approaches Hierarchical Clustering employs:

- **Agglomerative Clustering:** It follows the “Bottom-up” approach, it is also known as AGNES(Agglomerative Nesting). In this, each data point starts its clusters and merges as one moves up the hierarchy.

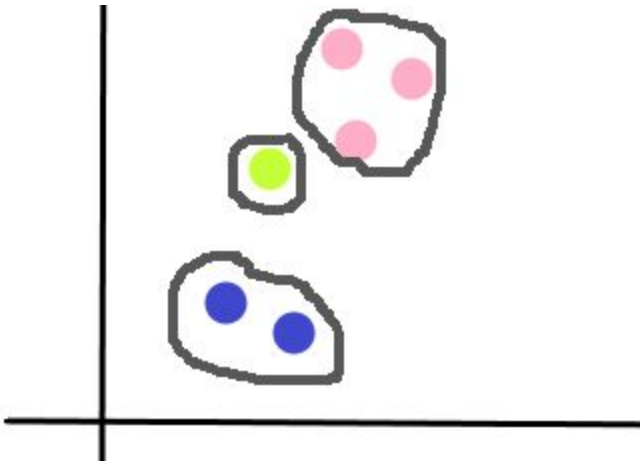
Algorithm



1. Compute the proximity matrix
2. Represent each data point a single-point cluster → Total clusters= N

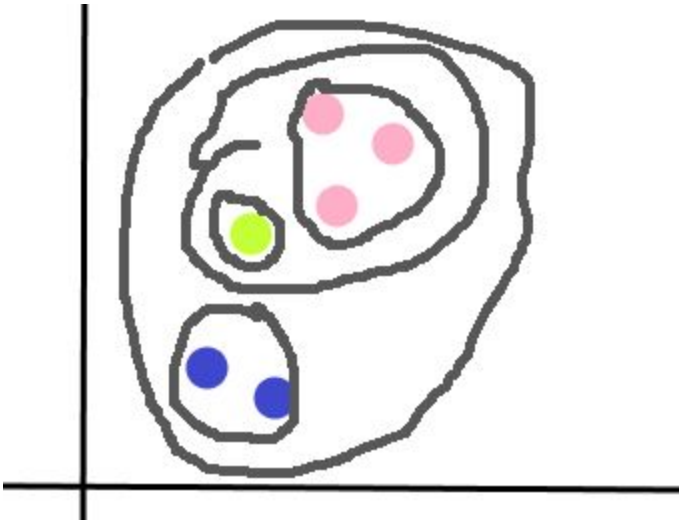


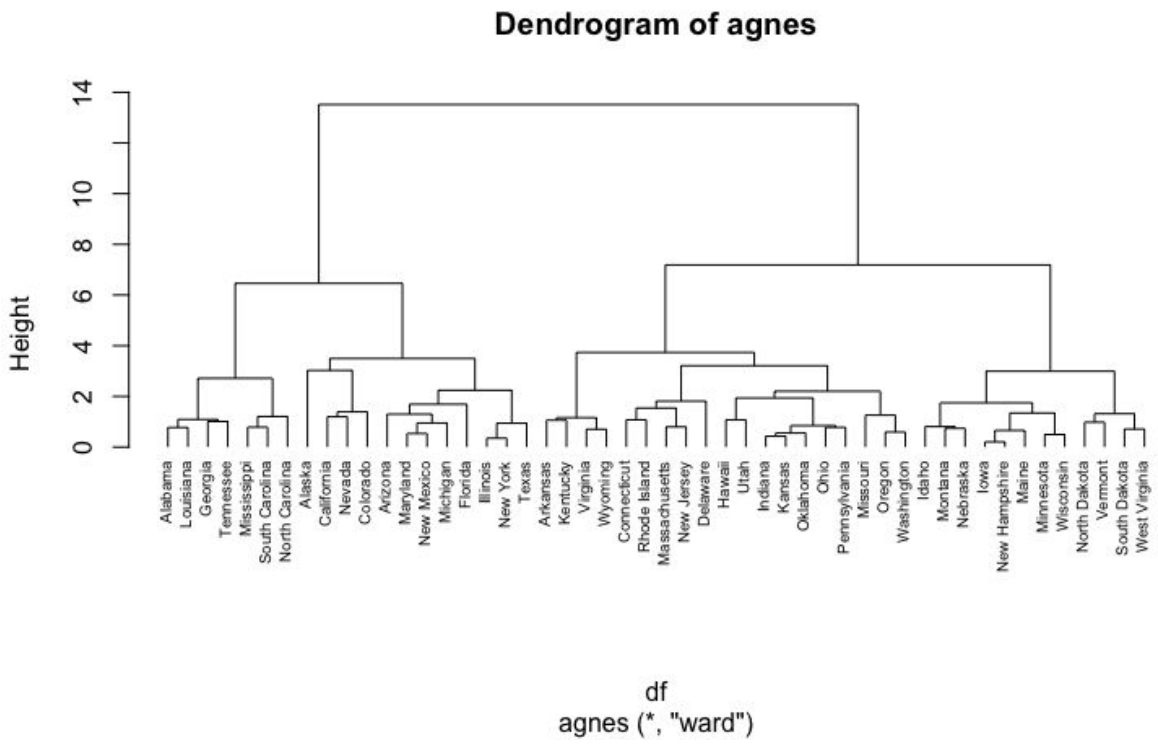
3. Merge two closest data points and make them one cluster →
   Total clusters = N-1, compute proximity matrix.



4. Take the two closest clusters and make them one cluster →
   Total clusters = N-2, compute proximity matrix.

5. Repeat step-3 until only one cluster is left.



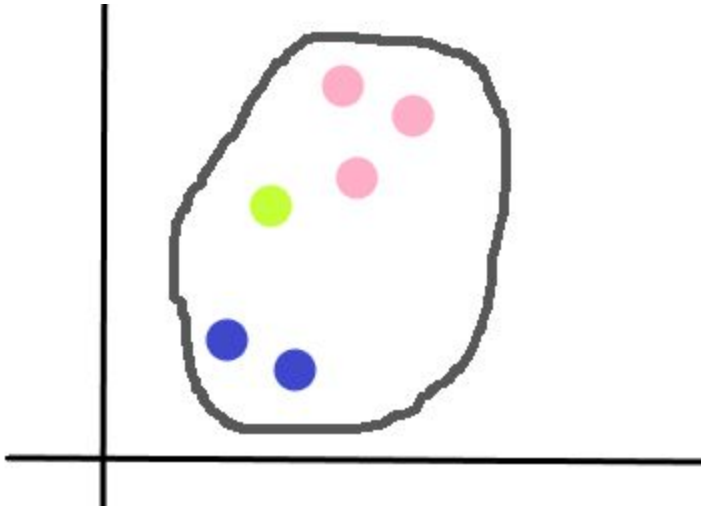


- **Divisive Clustering:** Divisive Clustering is exactly opposite to AGNES, it follows the “Top-Down” approach. In this, we consider all the data points as a single cluster and in each iteration, we separate the data points from the cluster with less or no similarity. Each point that is separated is considered as an individual cluster. In the end, we’ll be left with n clusters.

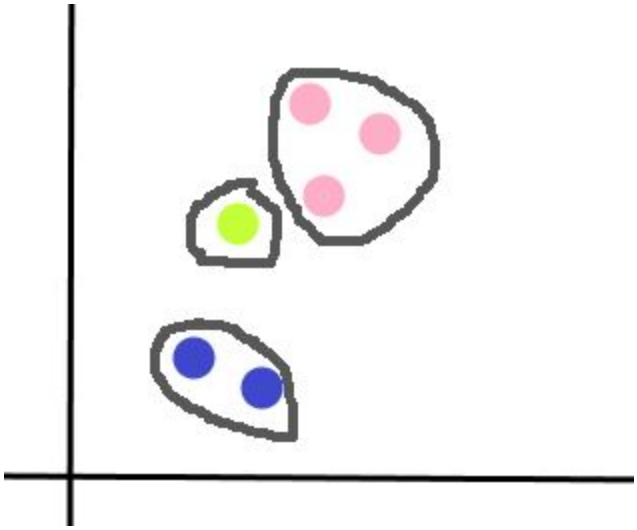
  It follows a divisive approach thus it is named as Divisive Hierarchical clustering.

  Algorithm:

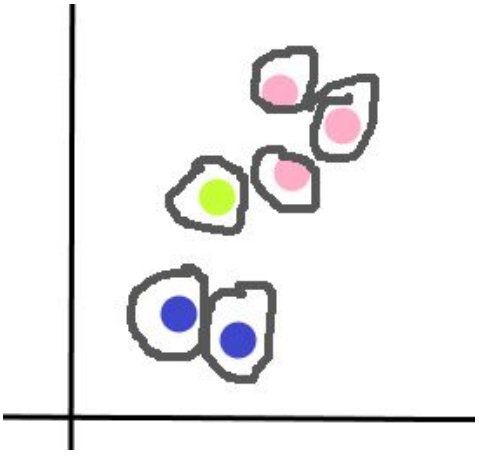
  1. Take all data points as a single cluster.



2. Based on similarities between points, cluster them together.

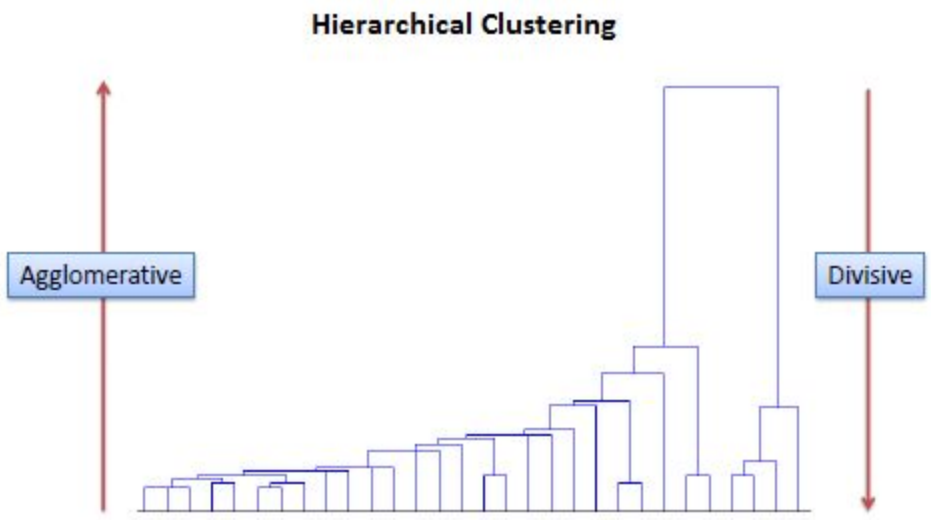


3. Based on similarities between clusters, split them.



4. Repeat until there are N clusters.

Dendrogram for Divisive Hierarchical clustering:

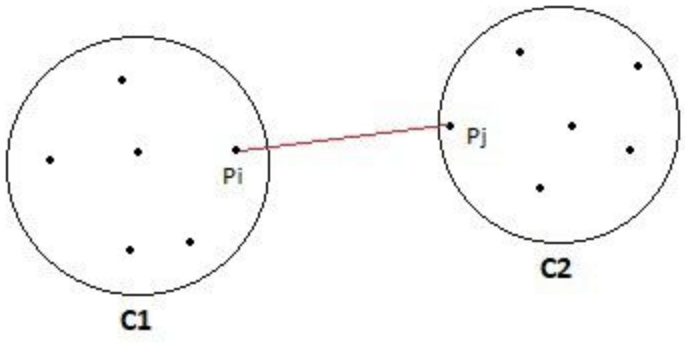


Evaluation metrics in Hierarchical clustering

To create quality clusters, it is important to calculate the similarity between clusters carefully. Approaches to carefully calculate similarities are:

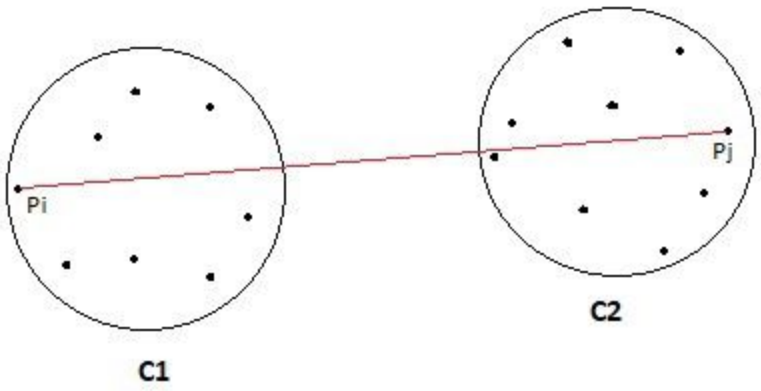
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Ward’s Method

MIN: It is also referred to as a single-linkage algorithm. It is defined as the similarity of two clusters C1 and C2 that is equal to the minimum of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.



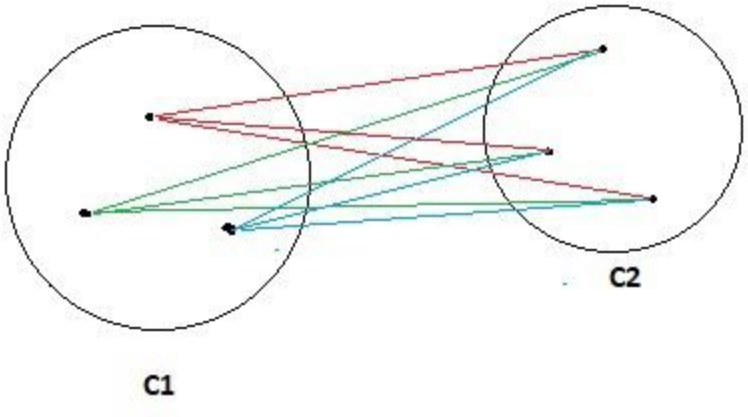
Sim(C1,C2) = Min Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2

MAX: It is also referred to as a complete linkage algorithm. It is exactly opposite to the MIN approach, it is defined as the similarity of two clusters C1 and C2 is equal to the maximum of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.



Sim(C1,C2) = Max Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2

Group Average: It is defined by taking all the pairs of points and computing their similarities and average of the similarities.



$sim(C1,C2) = \sum sim(Pi, Pj)/|C1|*|C2|$, where, $Pi \in C1$ & $Pj \in C2$

Ward's Method: This is defined as calculating the similarity between two clusters is the same as Group Average except that Ward's method calculates the sum of the square of the distances Pi and PJ.

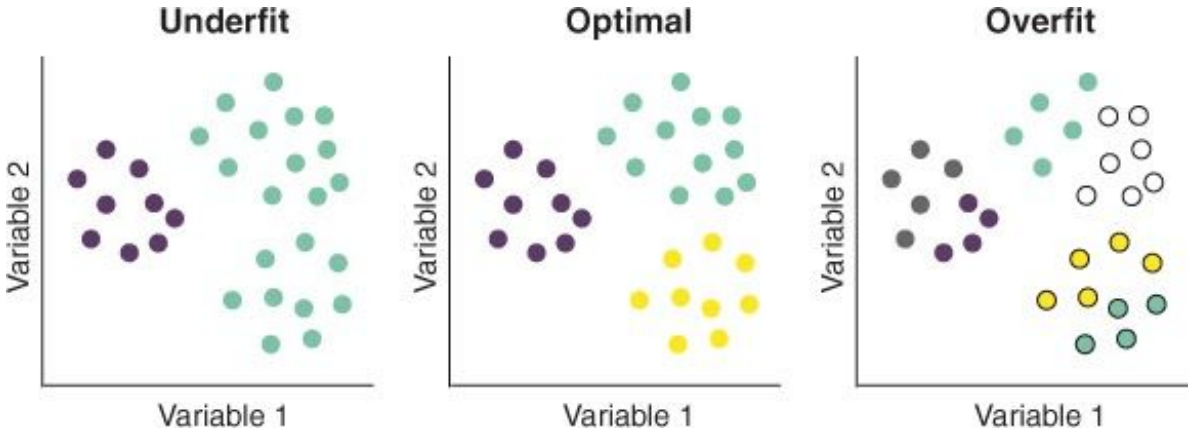
$sim(C1,C2) = \sum (dist(Pi, Pj))^2/|C1|*|C2|$

Overfitting and Underfitting

Overfitting is a condition where the model cramps even the minutiae detail of training data which impacts the performance of the model on training data.

But, in unsupervised learning, since no labeled data is provided, overfitting is often considered as a condition where clusters you are finding only exist in your dataset and can't be seen in new data.

Underfitting exists when the model learns nothing and isn’t generalized well for training data.



Avoiding overfitting and underfitting isn’t an easy task. The only way to create an optimal model is to use Evaluation metrics as in unsupervised learning there is no labeled set and the model itself has to find the same traits in data.

Computing Hierarchical Clustering in Python

class `sklearn.cluster.`**`AgglomerativeClustering`**(*n_clusters=2*, ***, *affinity='euclidean'*, *memory=None*, *connectivity=None*, *compute_full_tree='auto'*, *linkage='ward'*, *distance_threshold=None*)

Parameters accepted:

- **n_clusters** - The number of clusters to form as well as the number of centroids to generate.
- **Affinitystr-** Metric used to compute the linkage. Can be “euclidean”, “l1”, “l2”, “manhattan”, “cosine”, or “precomputed”.
- **linkage{“ward”, “complete”, “average”, “single”}:** Evaluation Metrics

Conclusion

In this post, we have learned the concepts of Hierarchical Clustering as well as the maths behind it. Also, we have applied the algorithm to real-life datasets. So I hope you are now able to relate the concepts that we have learned with the case study.

Thanks for reading! If you liked the blog then let me know your thoughts in the comment section.

Case Study:
https://github.com/dakshtrehan/Machine-Learning-Bootcamp/tree/master/Case-studies/Hierarchical%20Clustering