Hierarchical clustering

The building block of this type of clustering is the similarity between data points.

Q: How to define the similarity between data points?

A: This can be done via using one of the distance metrics.

Q: what are the distance metrics?

A: There are many functions, such as:-

- Euclidean distance
- Hamming distance
- Manhattan distance
- etc.

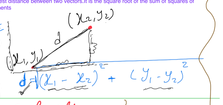
https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/

a nice description for many of the distance functions

Euclidean distance function:

Euclidean Distance represents the shortest distance between two vectors.It is the square root of the sum of squares of differences between corresponding elements

$$D_e = \left(\sum_{i=1}^{n} (P_i - q_i)^2\right)^{1/2}$$



$$d = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

Manhattan distance function:

$$D_m = \sum_{i=1}^{n} |P_i - q_i|$$

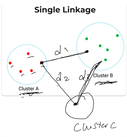
more are here

Q: How would we use these functions to do hierarchical clustering?

A: we use the concept of linkage.

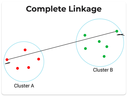
- **Single linkage**

This represents the shortest distance between points of pair of clusters.



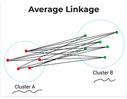
- **Complete linkage**

This represents the farthest distance between observations across pairs of clusters



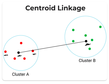
- **Average linkage**

This represents the average distance between all pairs of the points across clusters



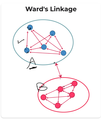
- **Centroid linkage**

1- Find the centroids of the two clusters
2- Compute the distance between centroids



- **Ward linkage**

This represents the distance that minimises variance within a cluster, and maximises variance across clusters

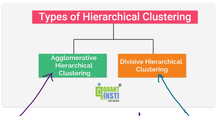


- **Hierarchical clustering**

we use a linkage method to split/merge data points (or clusters), until convergence.

- **Steps:-**

1- Compute distance matrix between points (or clusters)
2- Merge closest clusters
3- update distance matrix
4- Repeat 1,2,3 until convergence



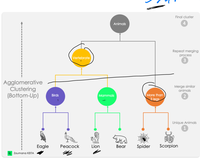
- Handle each point as a separate cluster
- Merge using the steps above

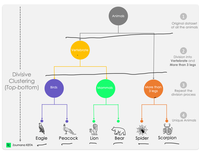
- Handle all data points as one cluster
- Split until convergence

Q: How to determine the number of clusters?

A: Using the concept of Dendrogram.

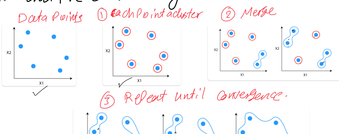
- Dendrogram; is a tree-like diagram that encodes the implicit relationship between data points.



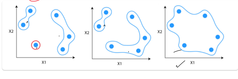


- **Illustrative Example**

please note the steps of applying the agglomerative clustering as follows:



③ Repeat until convergence.



Pros	Cons
• Encode inherent structures & similarities across entities.	• Computationally expensive
• Scalable to multiple domains.	• Sensitive to noise
• You don't need to determine # clusters	

- References:

1- https://blog.quantinsti.com/hierarchical-clustering-python
2- https://www.datacamp.com/tutorial/introduction-hierarchical-clustering-python