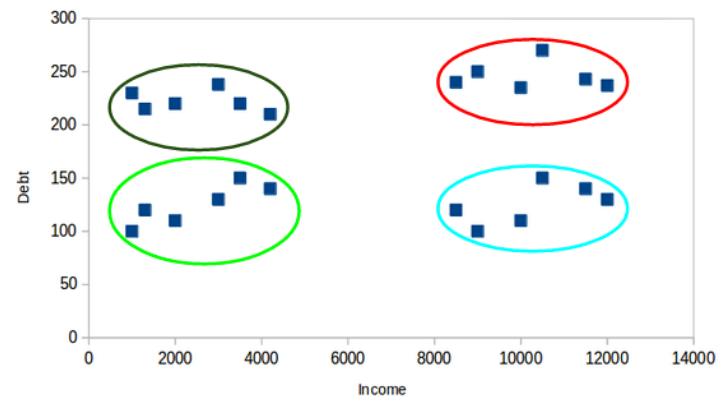
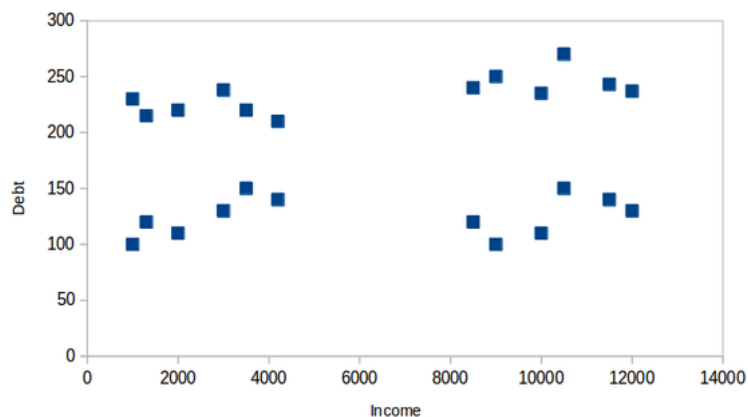


Clustering

Clustering

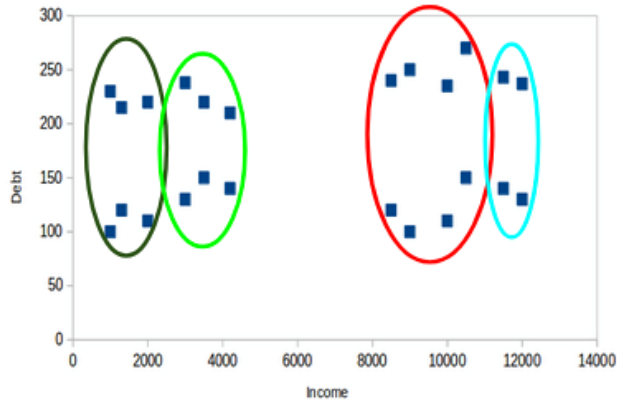
In clustering, we do not have a target to predict. We look at the data and then try to club similar observations and form different groups. Hence it is an unsupervised learning problem.



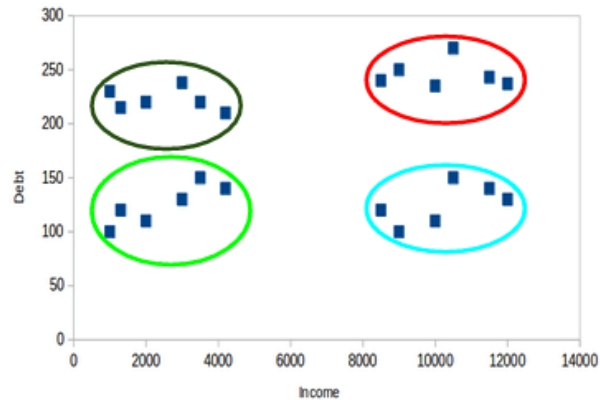
Properties of Clusters

All the data points in a cluster should be similar to each other.

The data points from different clusters should be as different as possible.



Case - I



Case - II

Applications of Clustering

Customer Segmentation

Customer segmentation is important for businesses including telecom, e-commerce, advertising, sales etc.

Document Clustering

Clustering helps us group these documents such that similar documents are in the same clusters.

Image Segmentation

Clustering to create clusters having similar pixels in the same group.

Recommendation Engines

Similar songs, videos, products can be clustered together and recommended to the viewers, customers etc.

K-Means Clustering Technique

K-Means Clustering Technique

It is an unsupervised learning algorithm.

It groups the unlabeled dataset into different clusters.

Here, K defines the number of predefined clusters that need to be created in the process, as if $K=2$, there will be two clusters, and for $K=3$, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

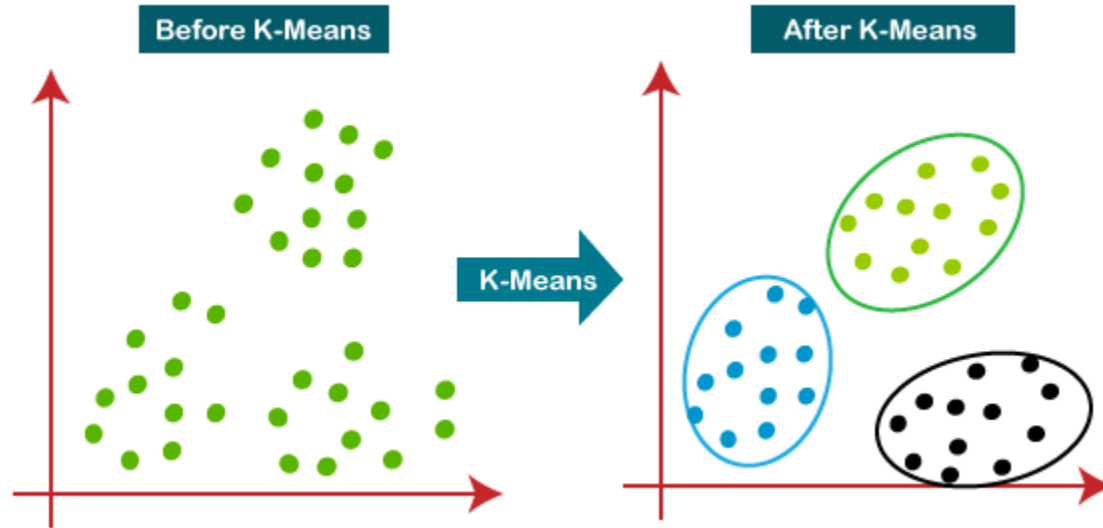
K-Means Clustering Technique

It is a centroid-based algorithm, where each cluster is associated with a centroid.

The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding centroids.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

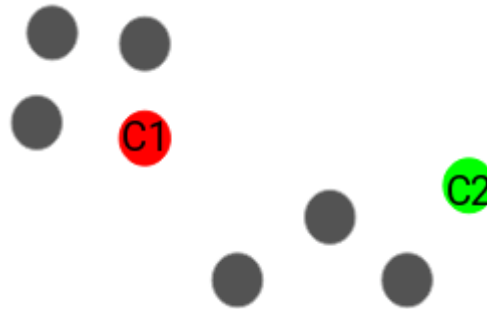
K-Means Clustering Technique



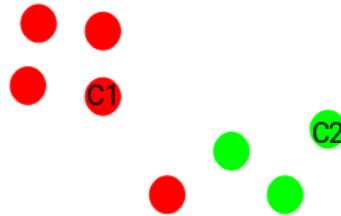
How K-Means Clustering works?

Step 1: Choose the number of clusters k .

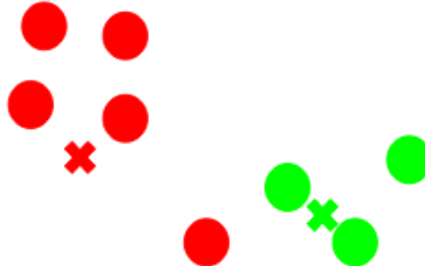
Step 2: Select k random points from the data as centroids. These points can be different from the ones present in the data.



Step 3: Assign all the points to the closest cluster centroid.



Step 4: Recompute the centroids of newly formed clusters



Step 5: Repeat steps 3 and 4 until convergence.

Stopping Criteria for K-Means Clustering

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations are reached

Mathematical Formulation

Suppose we have a data set $\{x_1, x_2, \dots, x_n\}$ consisting of N observations of a random D -dimensional Euclidean variable x . The goal of K-means is to partition that data set into some number K of clusters.

Suppose, a set of D -dimensional vectors μ_k as representing centres of k clusters.

Then, the goal is to find an assignment of data points to clusters, as well as a set of vectors $\{\mu_k\}$, such that the sum of squares of the distances of each data point to its closest vector μ_k , is a minimum.

For each instance, let there be a corresponding set of binary indicator variables

$r_{nk} \in \{0,1\}$, where $k=1,\dots,K$. If instance x_n is assigned to the cluster k then $r_{nk}=1$, and $r_{nj} = 0$ for $j \neq k$.

The objective function (a.k.a distortion measure) is given as:

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} ||x_n - \mu_k||^2$$

which represents the sum of the squares of the distance of each instance to its cluster centre represented by the vector μ_k .

Goal is find values for the $\{r_{nk}\}$ and the $\{\mu_k\}$ so as to minimise J .

Iterative Process for Minimising the Objective Function

The objective function can be minimised using an iterative process in which iteration involves two successive steps.

Step 1: Choose some initial values for μ_k . Now, minimise J with respect to r_{nk} keeping μ_k fixed.

Optimise J for each instance by choosing r_{nk} to be 1 for whichever value of k gives minimum value of $\|x_n - \mu_k\|^2$.

$$r_{nk} = \left\{ 1 \quad \text{if } k = \arg \min \|x_n - \mu_k\|^2; 0 \text{ otherwise} \right\}$$

Step 2: Minimise J with respect to μ_k keeping r_{nk} fixed.

The objective function is a quadratic function of μ_k , and it can be minimised by setting its derivative with respect to μ_k equal to 0, that gives:

$$2 \sum_{n=1}^N r_{nk} (x_n - \mu_k) = 0$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

The two steps are repeated until convergence.

Hierarchical Clustering

Hierarchical Clustering

It is an unsupervised learning technique.

It is an algorithm that groups similar objects into groups called clusters.

The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

Hierarchical Clustering is divided into two types:

- Agglomerative Hierarchical Clustering
- Divisive Hierarchical Clustering

Agglomerative Hierarchical Clustering

Agglomerative Hierarchical clustering is a bottom-up approach.

It starts by treating each observation as a separate cluster.

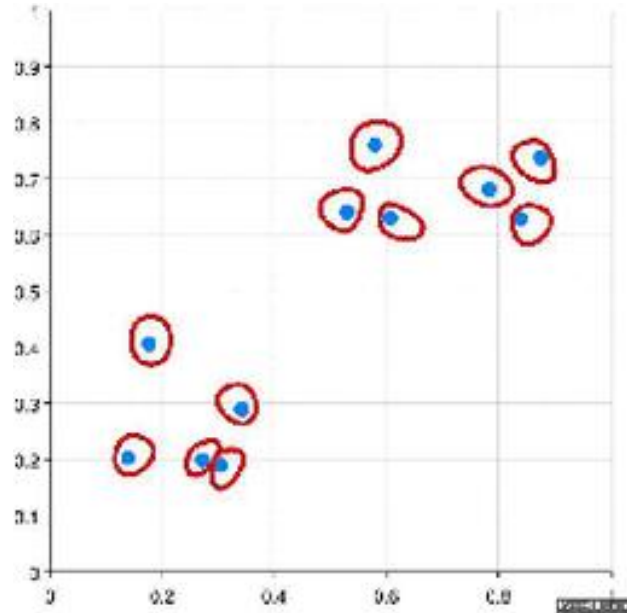
Then, it repeatedly executes the following two steps:

- (1) identify the two clusters that are closest together, and
- (2) merge the two most similar clusters.

This iterative process continues until all the clusters are merged together.

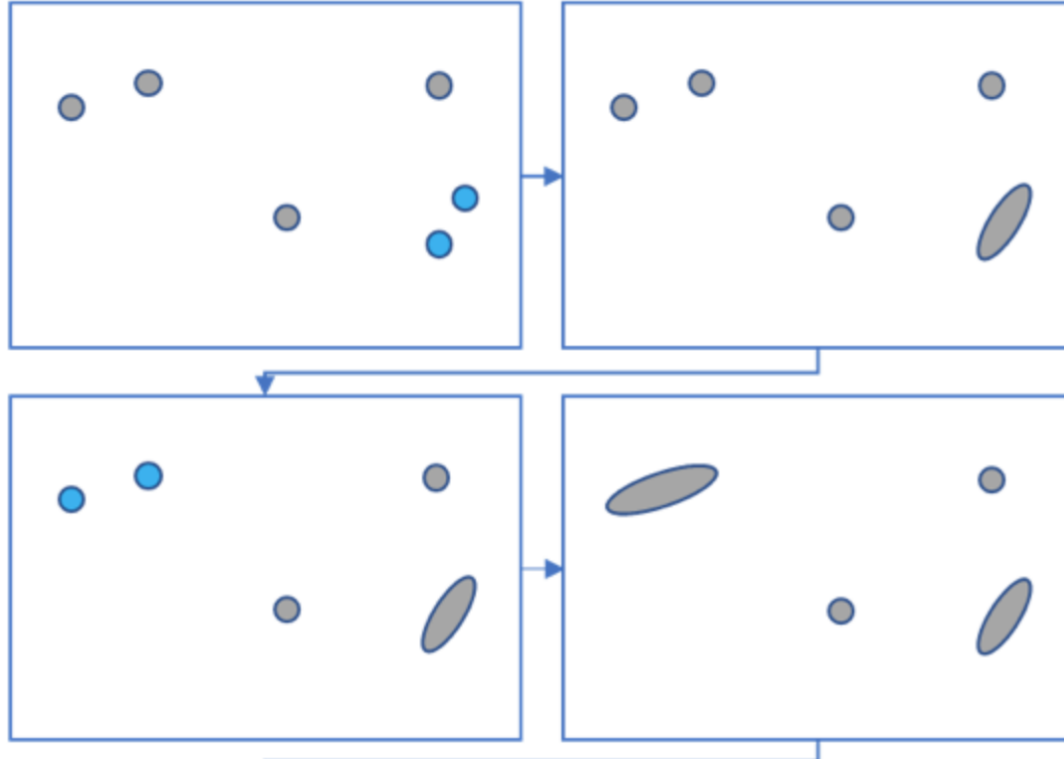
At each iteration the algorithm constructs a new partition of the data by merging the two nearest clusters together.

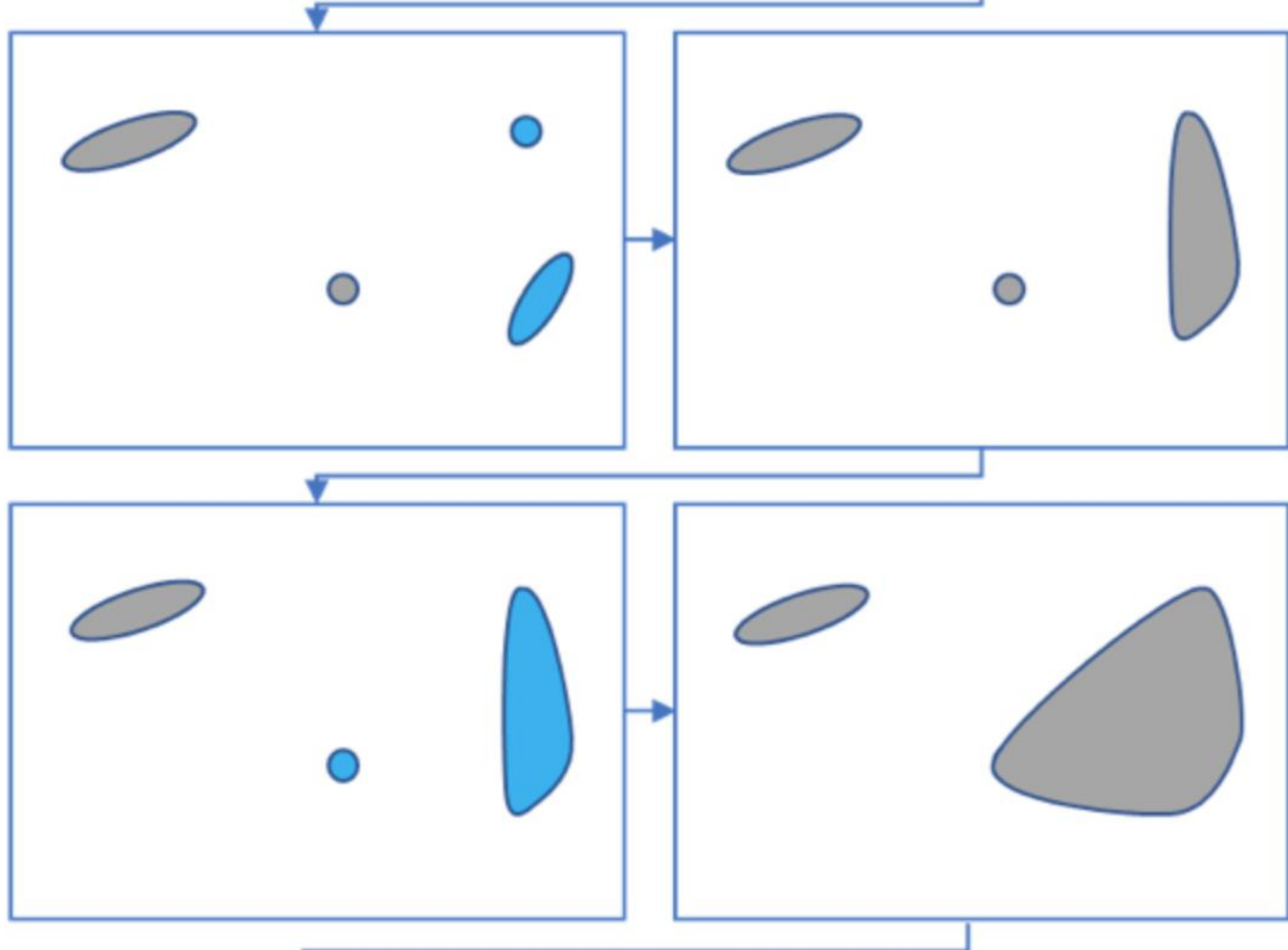
Agglomerative Hierarchical Clustering

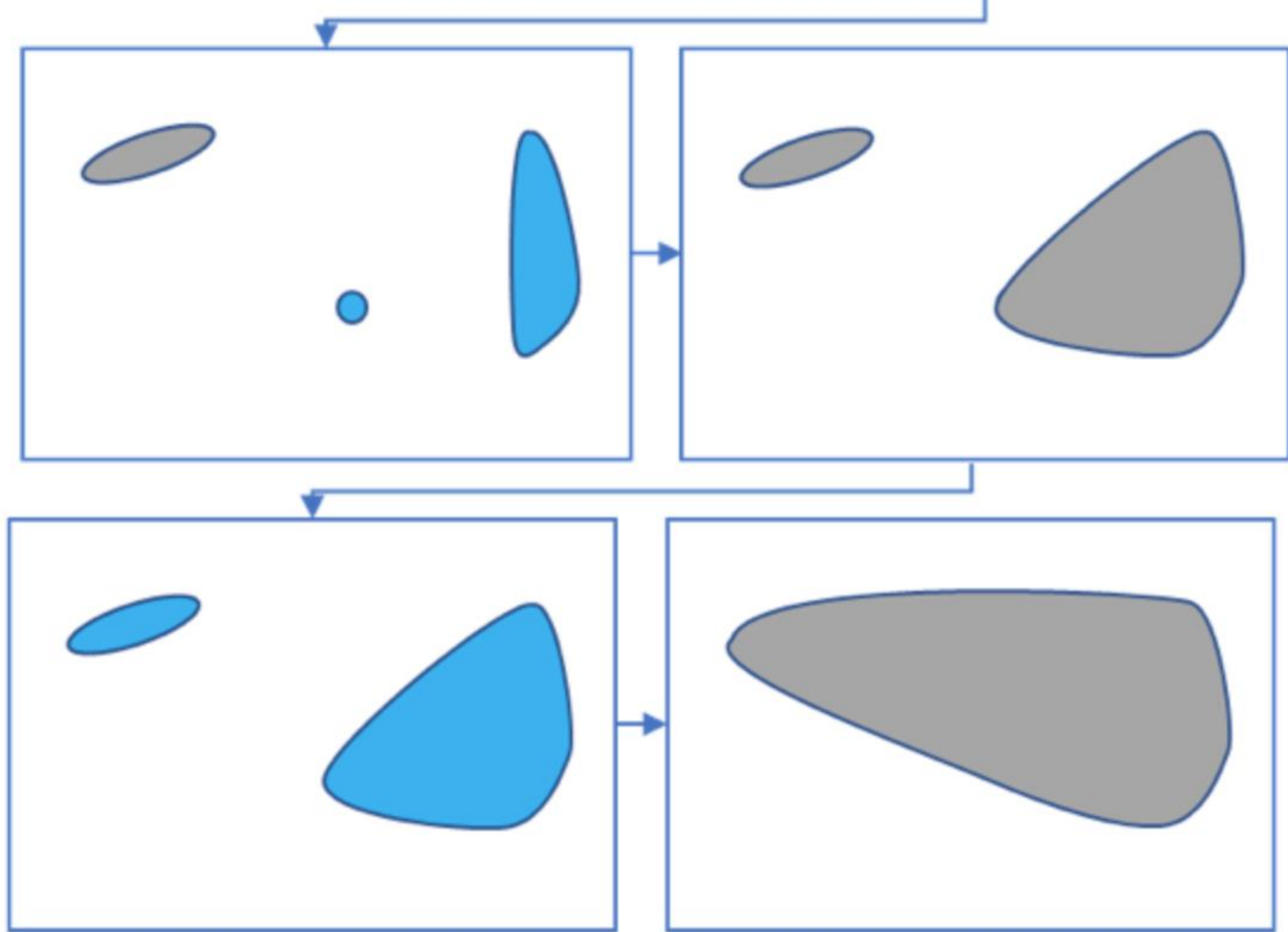


Identify the two clusters that are **closest** together

Merge the two most similar clusters







Algorithm 8.4: $\text{HAC}(D, L)$ – Hierarchical agglomerative clustering.

Input : data $D \subseteq \mathcal{X}$; linkage function $L : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \rightarrow \mathbb{R}$ defined in terms of distance metric.

Output : a dendrogram representing a descriptive clustering of D .

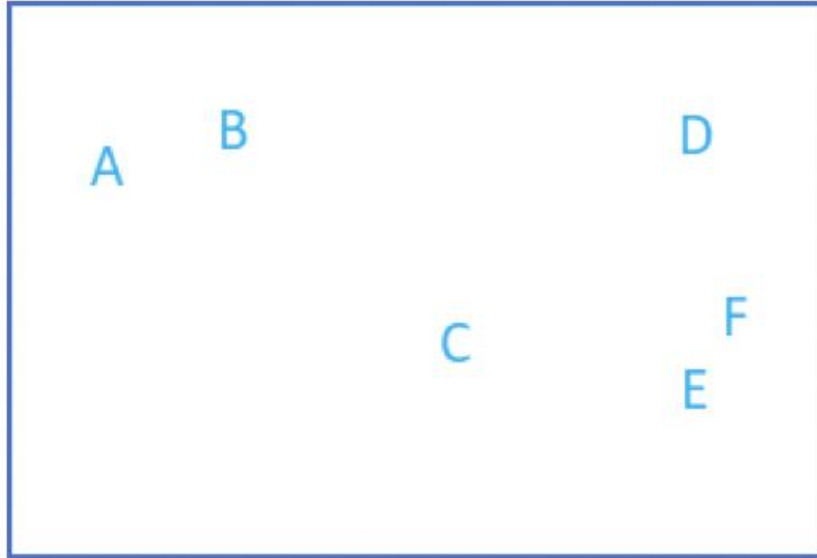
- 1 initialise clusters to singleton data points;
 - 2 create a leaf at level 0 for every singleton cluster;
 - 3 **repeat**
 - 4 | find the pair of clusters X, Y with lowest linkage l , and merge;
 - 5 | create a parent of X, Y at level l ;
 - 6 **until** all data points are in one cluster;
 - 7 **return** the constructed binary tree with linkage levels;
-

Dendrogram

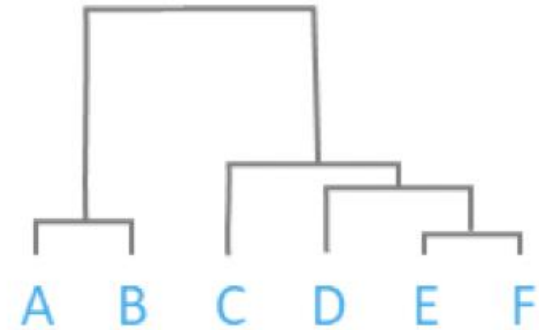
The main output of Hierarchical Clustering is a ***dendrogram***, which shows the hierarchical relationship between the clusters.

Definition: *Given a data set D , a dendrogram is a binary tree with the elements of D at its leaves. An internal node of the tree represents the subset of elements in the leaves of the subtree rooted at that node. The level of a node is the distance between the two clusters represented by the children of the node. Leaves have level 0.*

Dendrogram



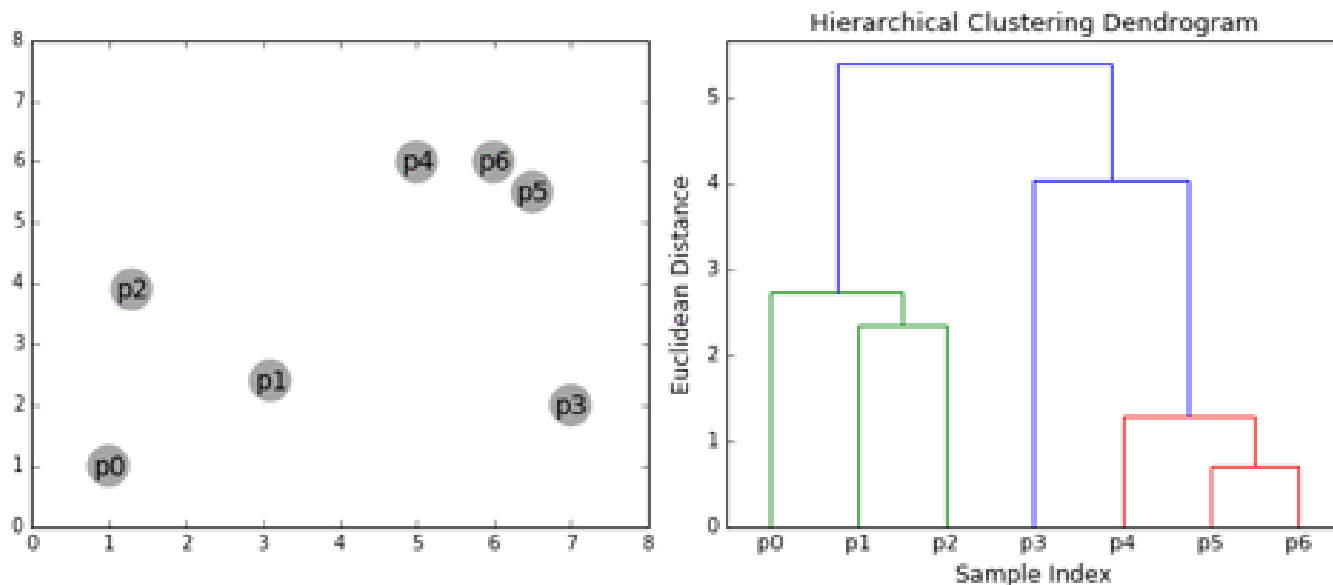
Dendrogram



Dendrogram


Distance between data points represents dissimilarities.

Height of the blocks represents the distance between clusters.



Linkage Function

There are several ways to measure the distance between clusters in order to decide the rules for clustering, and they are often called Linkage Functions or Linkage Methods.

Definition 8.5 (Linkage function). A linkage function $L : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \rightarrow \mathbb{R}$ calculates the distance between arbitrary subsets of the instance space, given a distance metric $\text{Dis} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. 

In general, the HAC algorithm gives different results when different linkage functions are used.

Common Linkage Functions

- Single Linkage
- Complete Linkage
- Average Linkage
- Centroid Linkage

Single Linkage

Single Linkage defines the distance between two clusters as the smallest pairwise distance between elements from each cluster.

$$L_{\text{single}}(A, B) = \min_{x \in A, y \in B} \text{Dis}(x, y)$$

Hierarchical clustering using single linkage can essentially be done by calculating and sorting all pairwise distances between data points, which requires $O(n^2)$ time for n points.

Complete Linkage

Complete Linkage defines the distance between two clusters as the largest pointwise distance.

$$L_{\text{complete}}(A, B) = \max_{x \in A, y \in B} \text{Dis}(x, y)$$

Average Linkage

Average Linkage defines the cluster distance as the average pointwise distance.

$$L_{\text{average}}(A, B) = \frac{\sum_{x \in A, y \in B} \text{Dis}(x, y)}{|A| \cdot |B|}$$

Centroid Linkage

Centroid Linkage defines the cluster distance as the point distance between the cluster means.

$$L_{\text{centroid}}(A, B) = \text{Dis} \left(\frac{\sum_{x \in A} x}{|A|}, \frac{\sum_{y \in B} y}{|B|} \right)$$

Properties of Linkage Functions

All linkage functions coincide for singleton clusters: $L(\{x\}, \{y\}) = \text{Dis}(x, y)$.

However, for larger clusters they start to diverge.

For example, suppose $\text{Dis}(x, y) < \text{Dis}(x, z)$, then the linkage between $\{x\}$ and $\{y, z\}$ is different in all four cases:

$$L_{\text{single}}(\{x\}, \{y, z\}) = \text{Dis}(x, y)$$

$$L_{\text{complete}}(\{x\}, \{y, z\}) = \text{Dis}(x, z)$$

$$L_{\text{average}}(\{x\}, \{y, z\}) = (\text{Dis}(x, y) + \text{Dis}(x, z)) / 2$$

$$L_{\text{centroid}}(\{x\}, \{y, z\}) = \text{Dis}(x, (y + z) / 2)$$

Numerical Example

Construct the dendrogram using agglomerative hierarchical clustering for the five observations whose distances from each other are given in the table below. Use complete linkage to calculate distance between clusters.

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

Numerical Example contd.

In the given table, the smallest distance between point 3 and 5 i.e. 2. Thus, merge these two points into one cluster. Let's name this new cluster as '35'.

Now, obtain the new distance matrix. Points 3 and 5 are removed and replaced by a new cluster '35'. Use complete linkage (i.e. maximum pair-wise distance) to calculate the distance between new cluster '35' and every other point.

For instance, $D(1,3) = 3$ and $D(1,5) = 11$, thus $D(1,35) = 11$. The new distance matrix will be:

	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

Numerical Example contd.

Next, the smallest distance in the table is between point 2 and 4 i.e. 5. So 2 and 4 are merged together and new distance matrix will be:

$$D(35,24) = \max(D(3,2), D(3,4), D(5,2), D(5,4))$$

$$D(35,24) = \max(7, 9, 10, 8) = 10$$

$$D(1,35) = \max(D(1,3), D(1,5))$$

$$D(1,35) = \max(3, 11) = 11$$

$$D(1,24) = \max(D(1,2), D(1,4)) = \max(9, 6) = 9$$

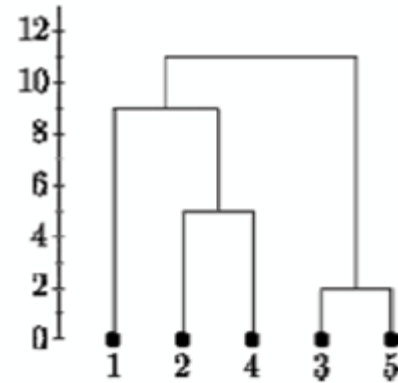
	35	24	1
35	0		
24	10	0	
1	11	9	0

Numerical Example contd.

Now, the smallest distance in the table is between the point 1 and cluster 24. Thus, 1 is merged into cluster 24.

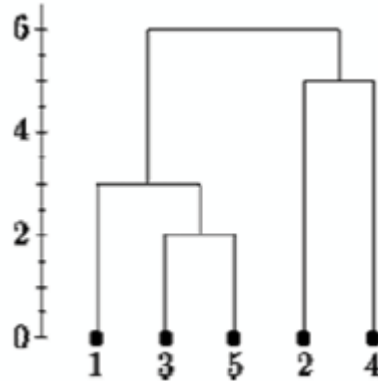
Next, only two clusters are left i.e. 124 and 35, which will be merged together into one.

The dendrogram obtained for this case is as shown:



Do It Yourself Exercise

For the same previous example, construct the dendrogram using single linkage criteria. Verify if you obtain the same dendrogram as given below.



Pros & Cons of Hierarchical Clustering

Dendrograms – like other tree models – have high variance in that small changes in the data points can lead to large changes in the dendrogram.

Hierarchical clustering methods have the distinct advantage that the number of clusters does not need to be fixed in advance.

Hierarchical Clustering has high computational cost.

HCA requires choice of distance metric and linkage function.

References

<https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>

<https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning>

<https://www.displayr.com/what-is-hierarchical-clustering/#:~:text=Hierarchical%20clustering%2C%20also%20known%20as,broadly%20similar%20to%20each%20other.>

<https://www.kdnuggets.com/2019/09/hierarchical-clustering.html>

Numerical Example from <https://online.stat.psu.edu/stat555/node/86/>