Unsupervised Learning

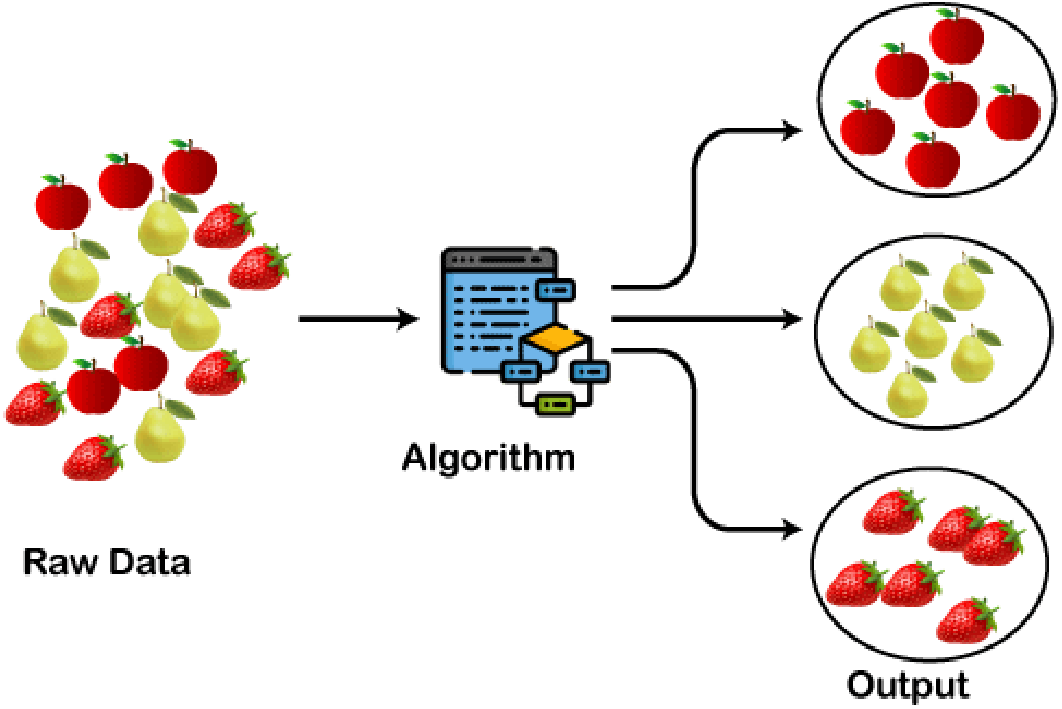
(Clustering)

Unsupervised Learning

- It uses machine learning algorithms to analyze and cluster unlabeled datasets
  - These algorithms discover hidden patterns or data groupings without the need for human intervention
- Its ability to discover similarities and differences in information make it the ideal solution for
  - exploratory data analysis
  - cross-selling strategies
  - customer segmentation
  - and image recognition

Unsupervised Learning contd.

- No labels are given to the learning algorithm, leaving it on its own to find structure in its input

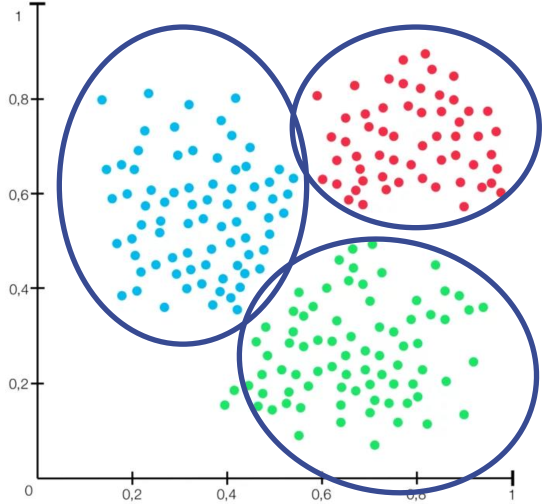


Types of Unsupervised Learning

- Clustering
  - The methods of finding the similarities between data items such as the same shape, size, color, price, etc. and grouping them to form a cluster is cluster analysis
- Dimensionality Reduction
  - Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset
  - More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality
- Association Rule Mining
  - Association rule mining is a procedure which aims to observe frequently occurring patterns, correlations, or associations from datasets

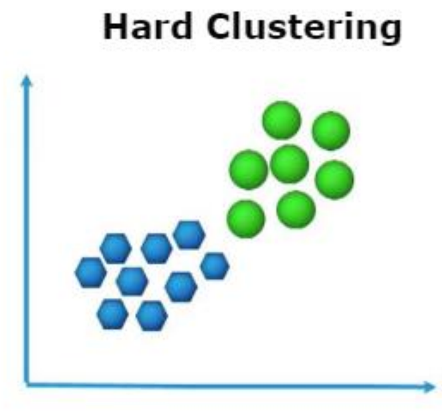
Clustering

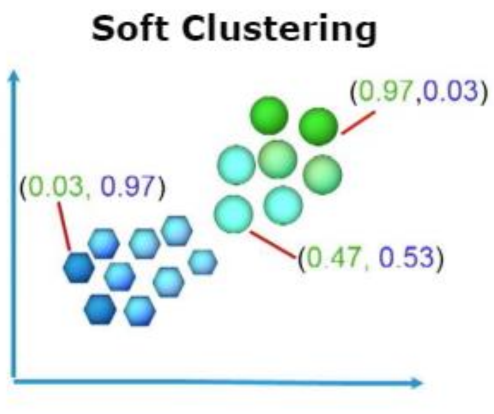
- Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups
    - In simple words, the aim is to segregate groups with similar traits and assign them into clusters



Types of Clustering

- Hard Clustering:
  - In hard clustering, each data point either belongs to a cluster completely or not
- Soft Clustering:
  - In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned





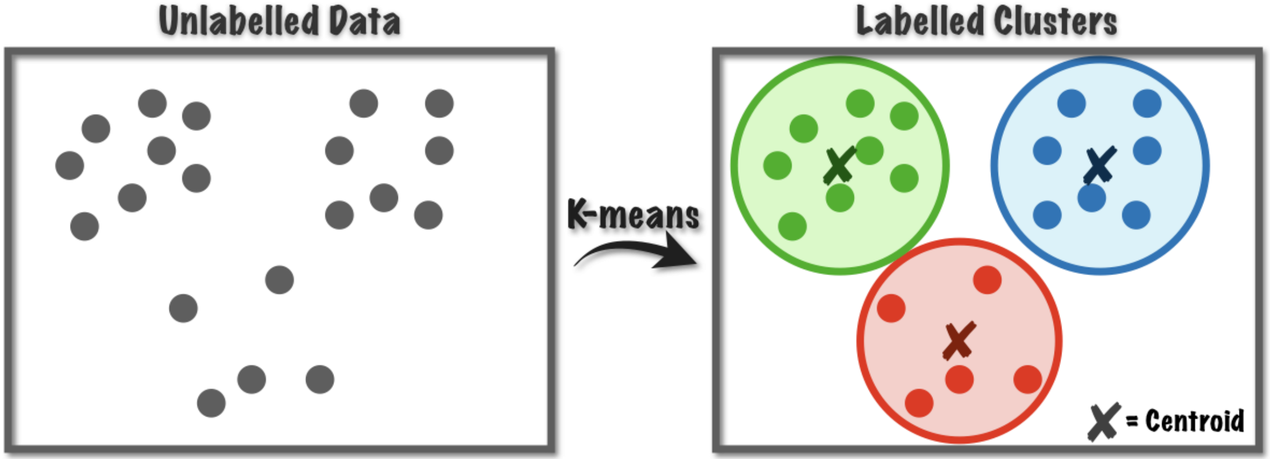
Types of clustering algorithms

- Connectivity models:
    - As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away
    - These models can follow two approaches
        - Starting with classifying all data points into separate clusters & then aggregating them as the distance decreases
        - All data points are classified as a single cluster and then partitioned as the distance increases
    - These models are very easy to interpret but lacks scalability for handling big datasets

Types of clustering algorithms contd.

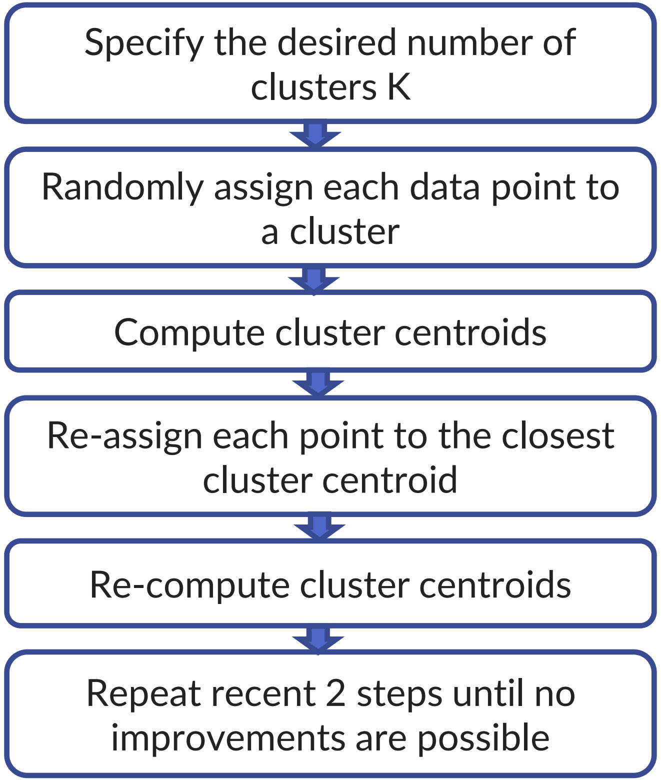
- Centroid models:
  - These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters
  - The no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset
- Distribution models:
  - These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution
  - These models often suffer from overfitting
- Density Models:
  - These models search the data space for areas of varied density of data points in the data space
  - It isolates various different density regions and assign the data points within these regions in the same cluster

K Means Clustering



K-Means Clustering

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



Specify the desired number of clusters K

Randomly assign each data point to a cluster

Compute cluster centroids

Re-assign each point to the closest cluster centroid

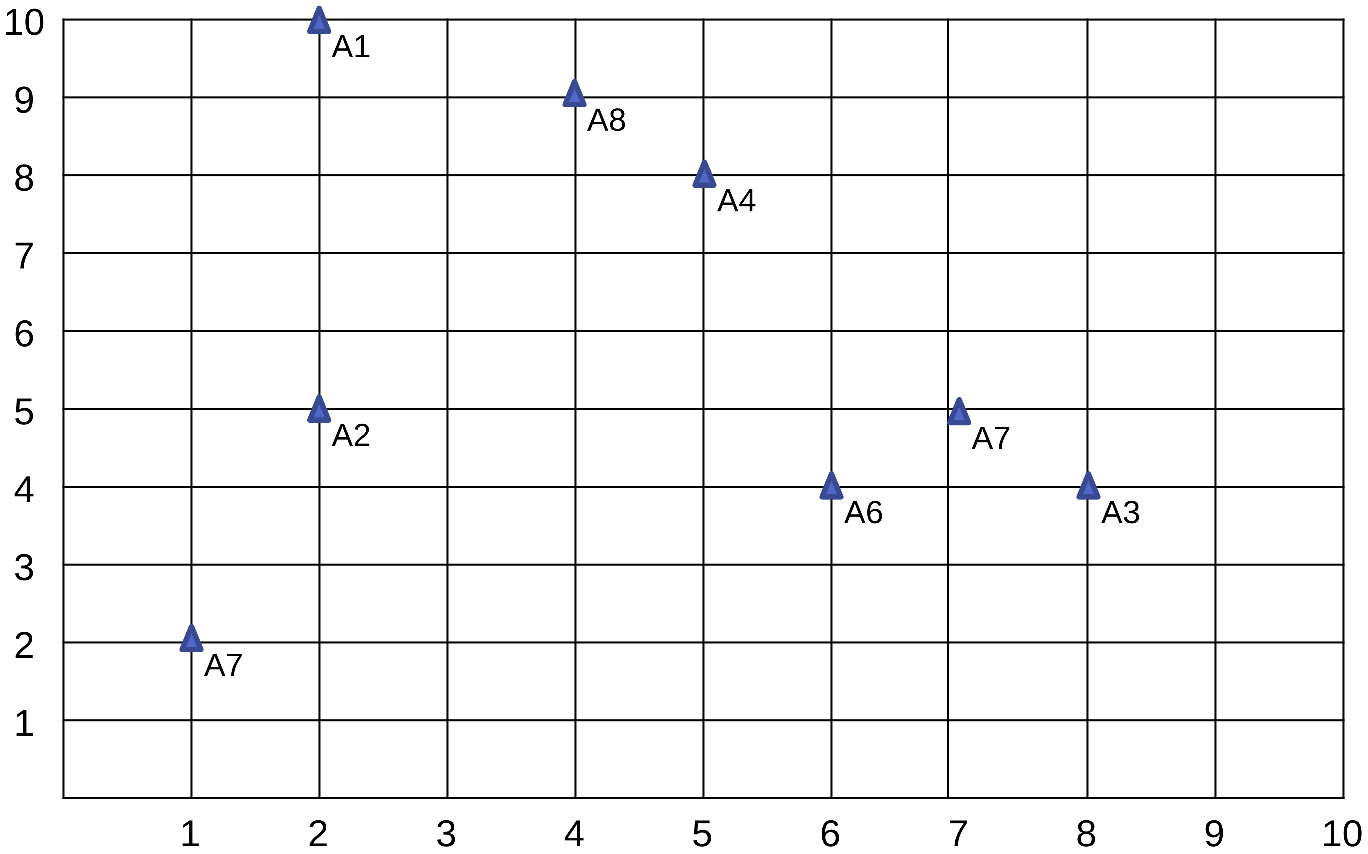
Re-compute cluster centroids

Repeat recent 2 steps until no improvements are possible

K-Means Example

Cluster the following eight points (with (x, y) representing locations) into three clusters:

A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)



K-Means Example ntd.

Iteration 1

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

Points	Distance from Center 1 (2,10)	Distance from Center 2 (5, 8)	Distance from Center 3 (1, 2)	Cluster Assignment
A1 (2,10)	0	3.61	8.06	C1
A2 (2,5)	5	4.24	3.16	C3
A3 (8,4)	8.49	5.00	7.28	C2
A4 (5,8)	3.61	0	7.21	C2
A5 (7,5)	7.07	3.61	6.71	C2
A6 (6,4)	7.21	4.12	5.39	C2
A7 (1,2)	8.06	7.21	0	C3
A8 (4,9)	2.24	1.41	7.62	C2

$$\boxed{d(A,C) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}$$

$$d(A1,C1) = \sqrt{(2-2)^2 + (10-10)^2}$$

$$d(A1,C1) = 0$$

$$d(A2,C1) = \sqrt{(2-2)^2 + (5-10)^2}$$

$$d(A2,C1) = \sqrt{25} = 5$$

$$d(A3,C1) = \sqrt{(8-2)^2 + (4-10)^2}$$

$$d(A3,C1) = \sqrt{36+36}$$

$$d(A3,C1) = 8.49$$

$$d(A4,C1) = \sqrt{(5-2)^2 + (8-10)^2}$$

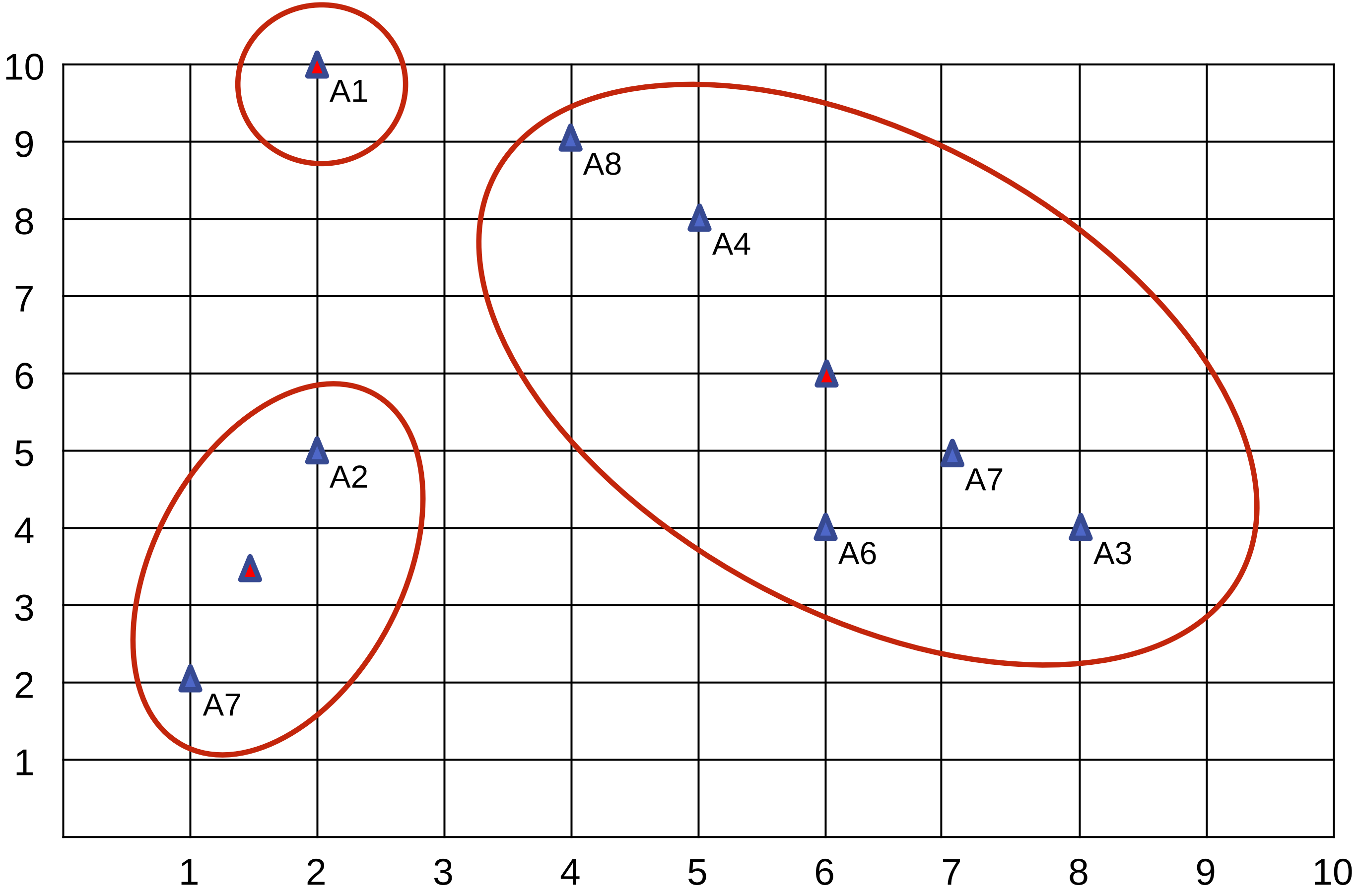
$$d(A4,C1) = \sqrt{9+4}$$

$$d(A4,C1) = 3.61$$

K-Means Example contd.

<table>
<tr><th>Cluster 1</th><th>Cluster 2</th><th>Cluster 3</th></tr>
<tr><td>A1 (2, 10)</td><td>A3 (8, 4)
A4 (5, 8)
A5 (7, 5)
A6 (6, 4)
A8 (4, 9)</td><td>A2 (2, 5)
A7 (1, 2)</td></tr>
<tr><td colspan="3">Updated Centroids (Averaging of Data Points)</td></tr>
<tr><td>(2, 10)</td><td>(6, 6)</td><td>(1.5, 3.5)</td></tr>
</table>

K-Means Example contd.



K-Means Example

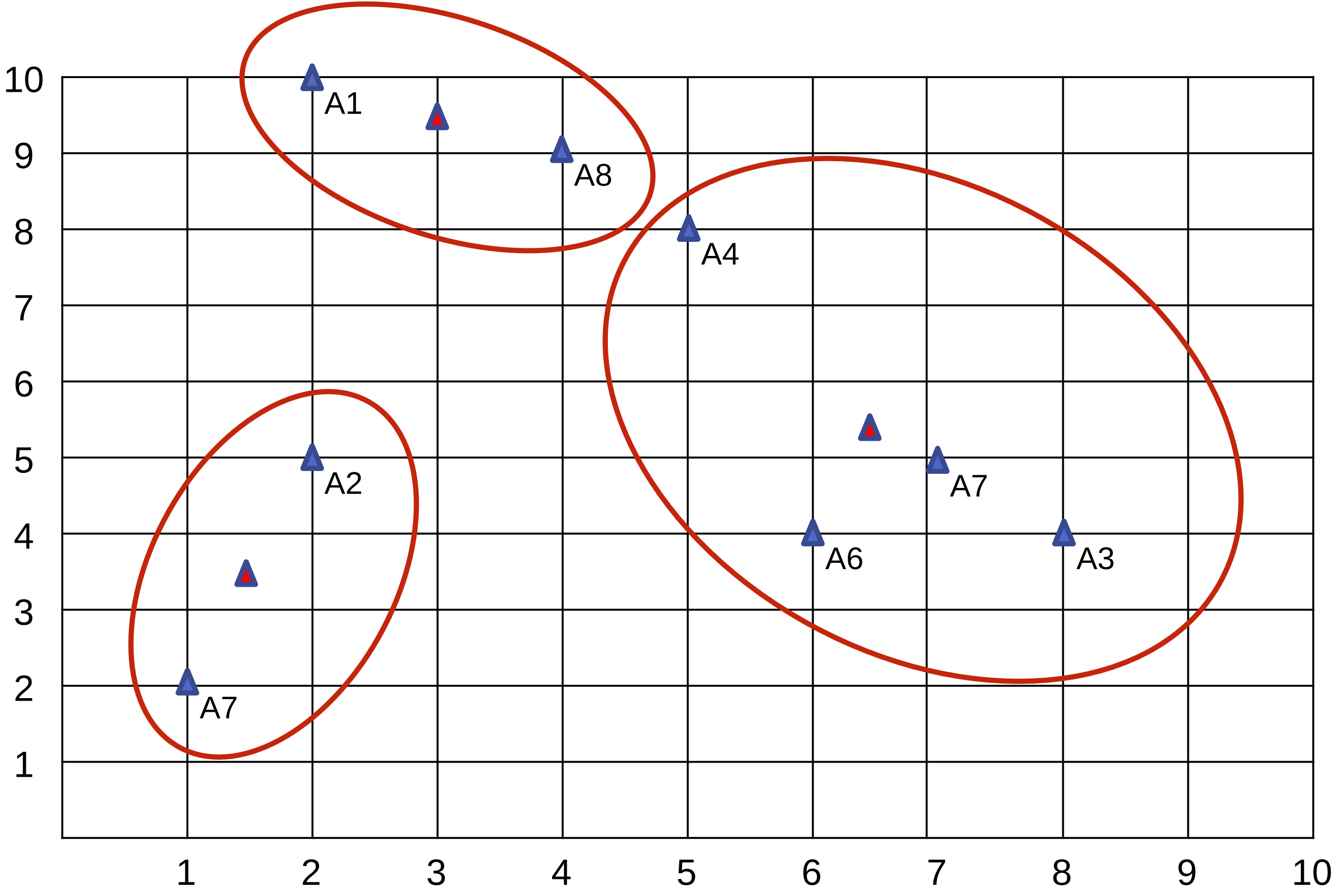
Iteration 2

Points	Distance from Center 1 (2,10)	Distance from Center 2 (6, 6)	Distance from Center 3 (1.5, 3.5)	Cluster Assignment
A1 (2,10)	0	5.66	6.52	C1
A2 (2,5)	5	4.12	1.58	C3
A3 (8,4)	8.49	2.83	6.52	C2
A4 (5,8)	3.61	2.24	5.70	C2
A5 (7,5)	7.07	1.41	5.70	C2
A6 (6,4)	7.21	2.00	4.53	C2
A7 (1,2)	8.06	6.40	1.58	C3
A8 (4,9)	2.24	3.61	6.04	C1

K-Means Example contd.

Cluster 1	Cluster 2	Cluster 3
A1 (2, 10) A8 (4, 9)	A3 (8, 4) A4 (5, 8) A5 (7, 5) A6 (6, 4)	A2 (2, 5) A7 (1, 2)
Updated Centroids (Averaging of Data Points)		
(3, 9.5)	(6.5, 5.25)	(1.5, 3.5)

K-Means Example contd.



K-Means Example

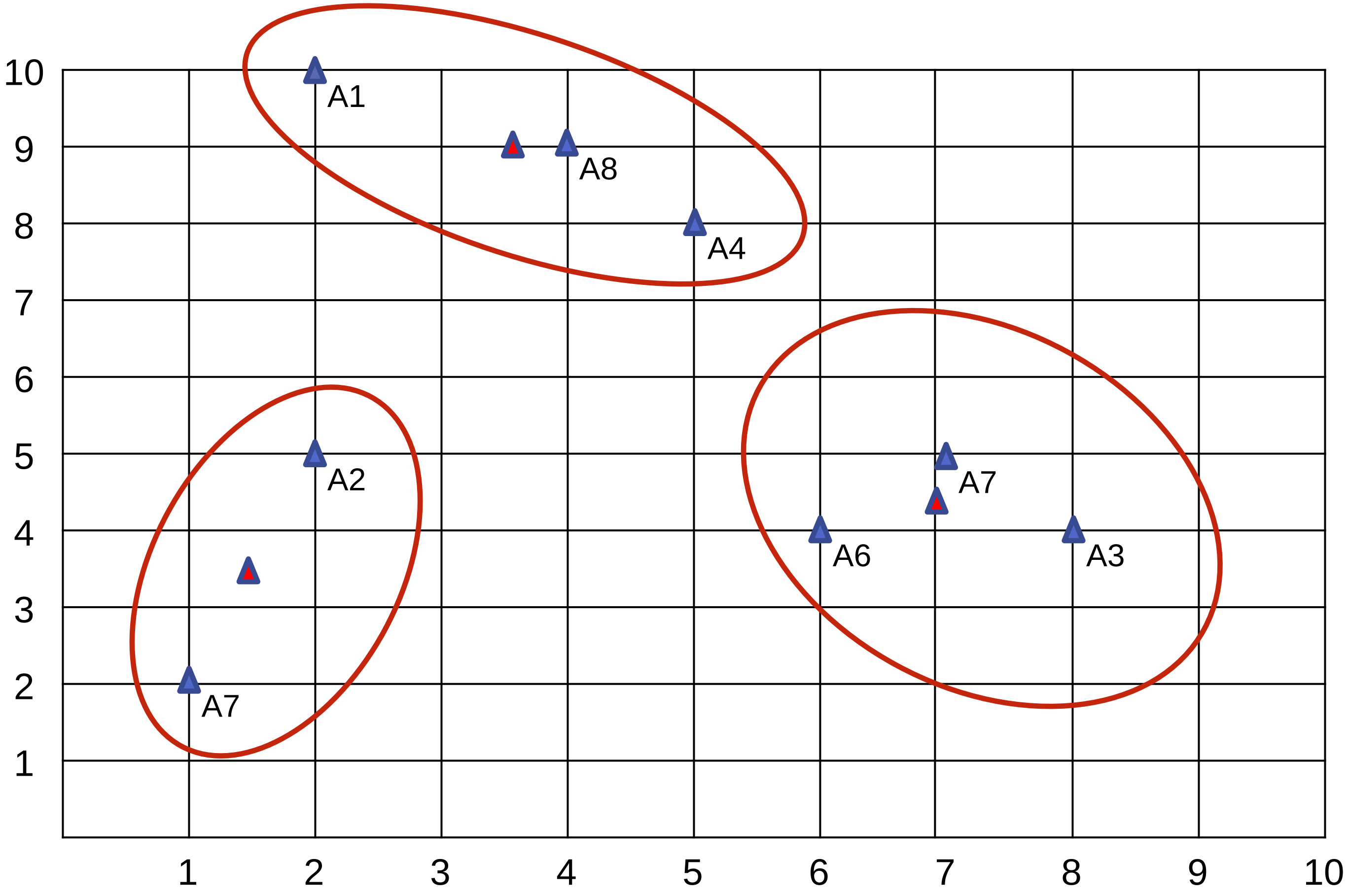
Iteration 3

Points	Distance from Center 1 (3, 9.5)	Distance from Center 2 (6.5, 5.25)	Distance from Center 3 (1.5, 3.5)	Cluster Assignment
A1 (2,10)	1.12	6.54	6.52	C1
A2 (2,5)	4.61	4.51	1.58	C3
A3 (8,4)	7.43	1.95	6.52	C2
A4 (5,8)	2.50	3.13	5.70	C1
A5 (7,5)	6.02	0.56	5.70	C2
A6 (6,4)	6.26	1.35	4.53	C2
A7 (1,2)	7.76	6.39	1.58	C3
A8 (4,9)	1.12	4.51	6.04	C1

K-Means Example contd.

Cluster 1	Cluster 2	Cluster 3
A1 (2, 10) A4 (5, 8) A8 (4, 9)	A3 (8, 4) A5 (7, 5) A6 (6, 4)	A2 (2, 5) A7 (1, 2)
Updated Centroids (Averaging of Data Points)		
(3.6, 9)	(7, 4.3)	(1.5, 3.5)

K-Means Example contd.



K-Means Example

Iteration 4

Points	Distance from Center 1 (3.6, 9)	Distance from Center 2 (7, 4.3)	Distance from Center 3 (1.5, 3.5)	Cluster Assignment
A1 (2,10)	1.89	7.58	6.52	C1
A2 (2,5)	4.31	5.05	1.58	C3
A3 (8,4)	6.66	1.04	6.52	C2
A4 (5,8)	1.72	4.21	5.70	C1
A5 (7,5)	5.25	0.70	5.70	C2
A6 (6,4)	5.55	1.04	4.53	C2
A7 (1,2)	7.47	6.43	1.58	C3
A8 (4,9)	0.40	5.58	6.04	C1

K-Means Example contd.

Cluster 1	Cluster 2	Cluster 3
A1 (2, 10) A4 (5, 8) A8 (4, 9)	A3 (8, 4) A5 (7, 5) A6 (6, 4)	A2 (2, 5) A7 (1, 2)
Updated Centroids (Averaging of Data Points)		
No need to update, as the clusters remain the same		

Exercise 1

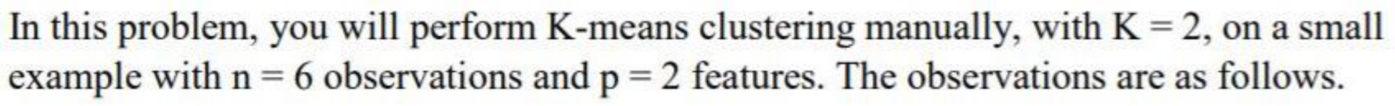
Use the **K-means** algorithm and **Euclidean distance** to cluster the following 10 examples into 3 clusters:

Pt	X1	X2
A	3	3
B	8	5
C	4	4
D	2	4
E	7	7
F	5	8
G	3	5
H	4	8
I	6	9
J	9	6

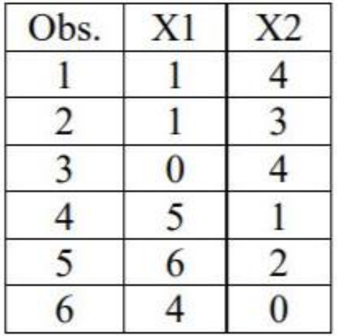
a. Perform K-Means clustering and show all the calculations performed at each iteration.
Assume that the initial clusters are A, E and H.

b. Draw a 10 by 10 space with all the 10 points and show the clusters and the new centroids after each iteration.

Exercise 2



In this problem, you will perform K-means clustering manually, with K = 2, on a small example with n = 6 observations and p = 2 features. The observations are as follows.



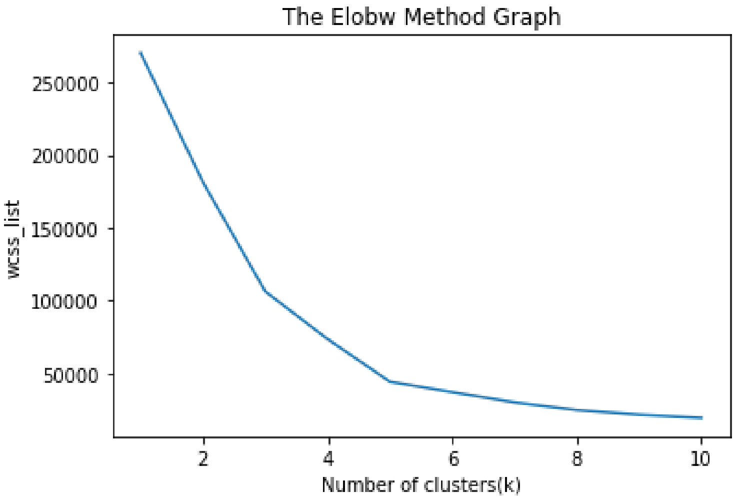
Obs.	X1	X2
1	1	4
2	1	3
3	0	4
4	5	1
5	6	2
6	4	0

Choose any two random points to be initial cluster centroids.

How to determine K value?

- The elbow method for this purpose
  - The elbow method uses the WCSS concept to draw the plot by plotting WCSS values on the Y-axis and the number of clusters on the X-axis
  - WCSS stands for within-cluster sum of squares
  - The location of bend in the plot is generally considered an indicator of the approximate number of clusters

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$



Hierarchical Clustering

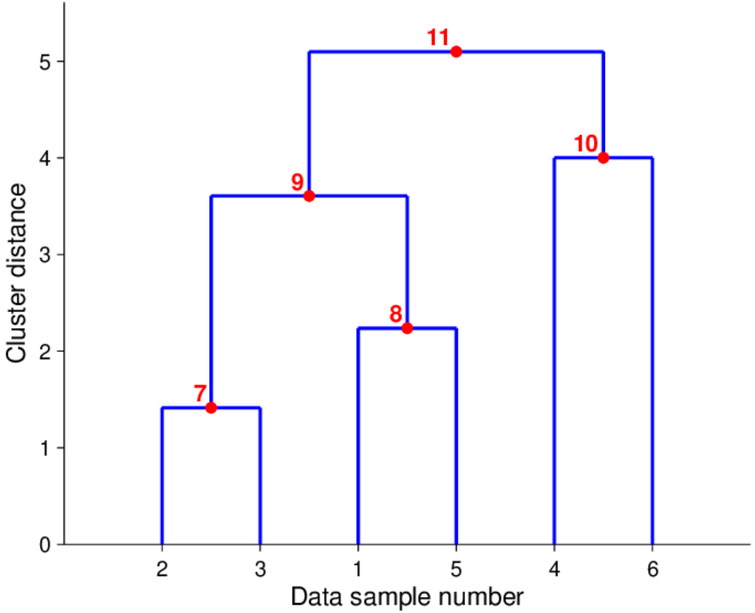
- Another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA
  - It works via grouping data into a tree of clusters
  - It begins by treating every data points as a separate cluster
  - Then, it repeatedly executes the subsequent steps:
    1. Identify the 2 clusters which can be closest together, and
    2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together

Why Hierarchical Clustering?

- K-Mean Clustering
  - Requires us to predetermine the number of clusters (the K value)
  - and It always tries to create the clusters of the same size

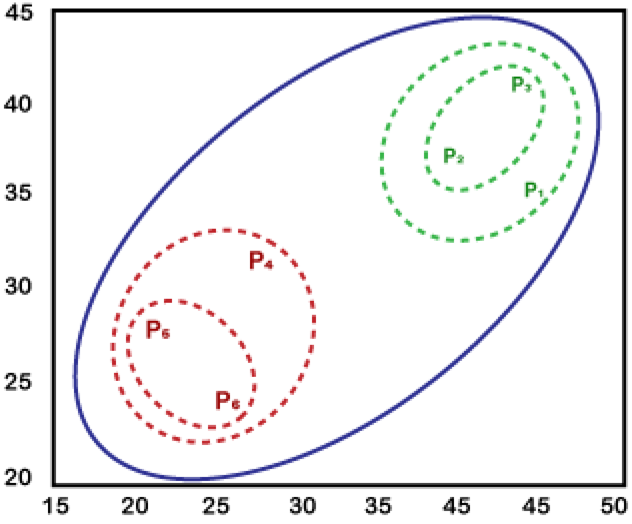
Dendrogram

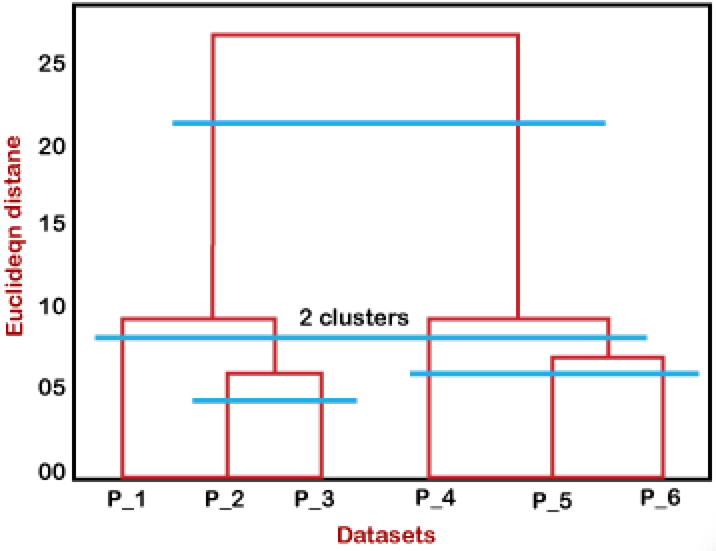
- A diagram called Dendrogram that graphically represents the hierarchy and is an inverted tree that describes the order in which data points are merged (bottom-up view) or cluster are broken up (top-down view) is used in hierarchical clustering



Working of Dendrogram

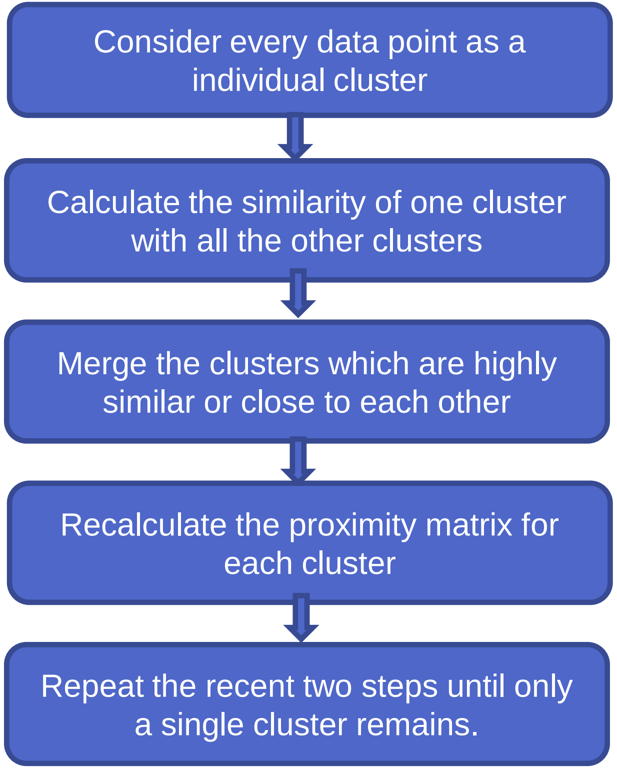
- In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset





Types of Hierarchical Clustering - Agglomerative

- It initially considers every data point as an individual Cluster and at every step, merge the nearest pairs of the cluster



Consider every data point as a individual cluster

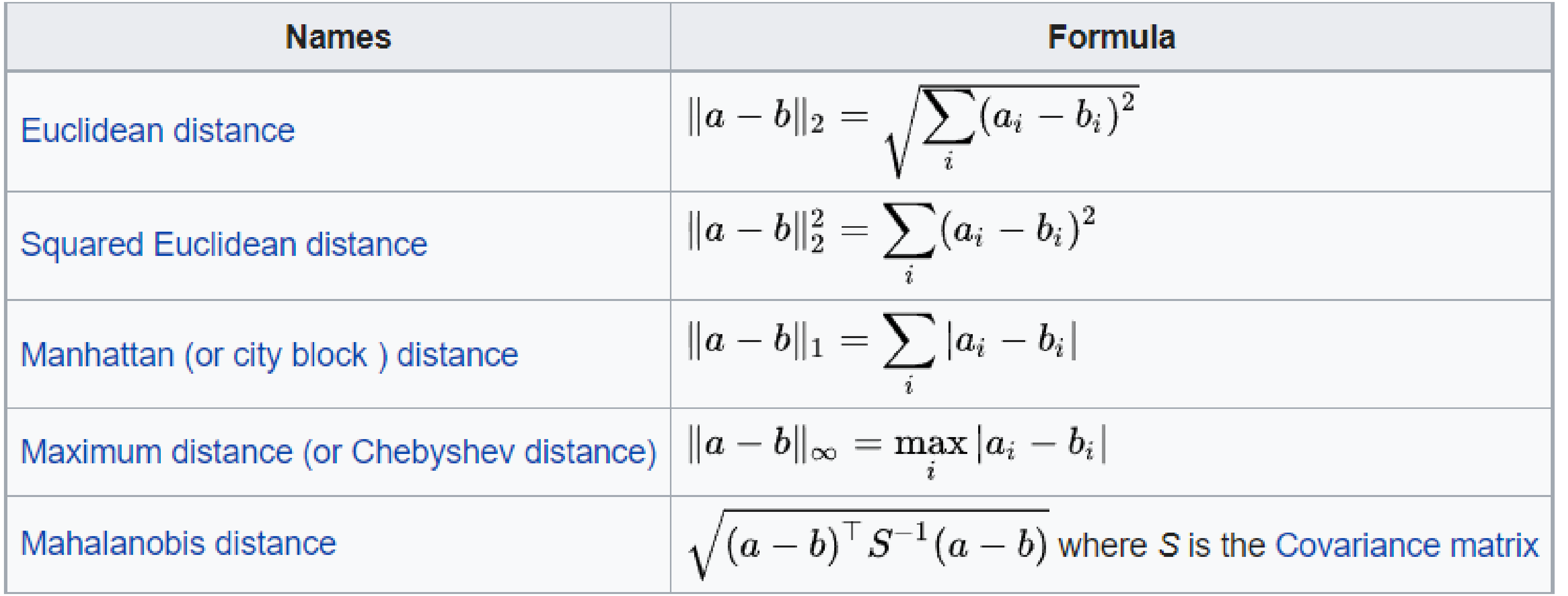
Calculate the similarity of one cluster with all the other clusters

Merge the clusters which are highly similar or close to each other

Recalculate the proximity matrix for each cluster

Repeat the recent two steps until only a single cluster remains.

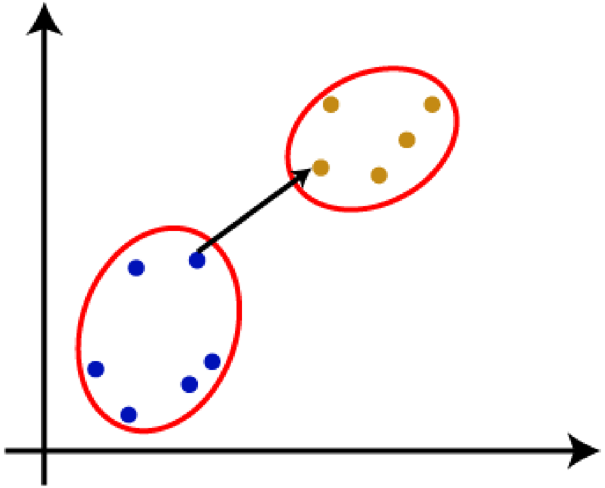
Agglomerative Clustering - Distance Metrics



Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan (or city block) distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maximum distance (or Chebyshev distance)	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where *S* is the Covariance matrix

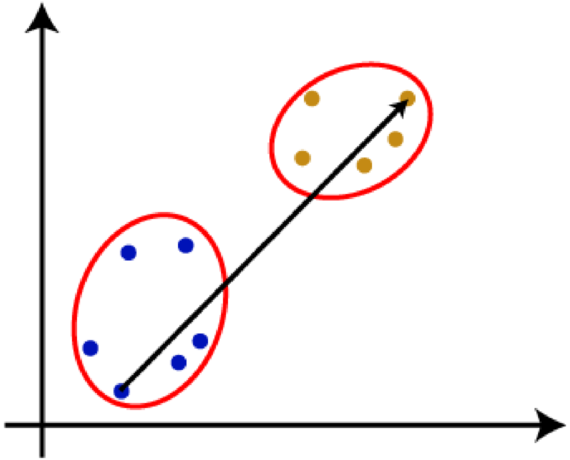
Agglomerative Clustering

- Measuring the distance between tow clusters (Linkage Methods)
    - Single Linkage: It is the Shortest Distance between the closest points of the clusters



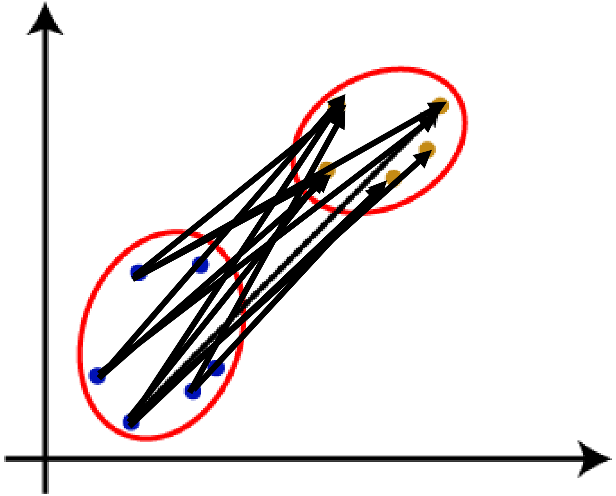
Agglomerative Clustering contd.

- Linkage Methods
  - 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



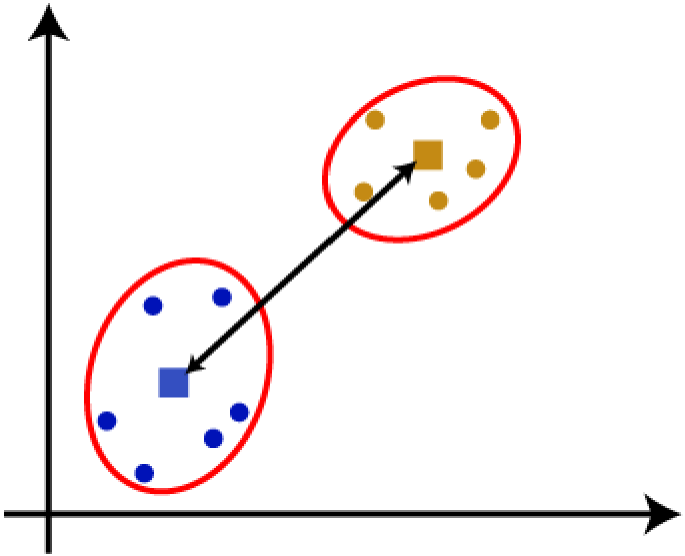
Agglomerative Clustering contd.

- Linkage Methods
  - 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



Agglomerative Clustering contd.

- Linkage Methods
  - Centroid Linkage: It is the linkage method in which the distance between the centroid of the clusters is calculated



Agglomerative Clustering - Example

- Data points = (18, 22, 43, 42, 27, 25)
  - Rewriting them to two dimensional space

Name	Data Point
P1	(18, 0)
P2	(22, 0)
P3	(43, 0)
P4	(42, 0)
P5	(27, 0)
P6	(25, 0)

Distance Metric:

$$d(P1, P2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Linkage Method:

Single Linkage

Agglomerative Clustering – Example contd.

- Distance Matrix

	P1 (18, 0)	P2 (22, 0)	P3 (43, 0)	P4 (42, 0)	P5 (27, 0)	P6 (25, 0)
P1 (18, 0)	0					
P2 (22, 0)	4	0				
P3 (43, 0)	25	21				
P4 (42, 0)	24	20	1	0		
P5 (27, 0)	9	5	16	15	0	
P6 (25, 0)	7	3	18	17	2	0

Shortest Distance

$$d(P1, P2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$d(P1, P2) = \sqrt{(18 - 22)^2 + (0 - 0)^2}$$

$$d(P1, P2) = 4$$

$$d(P1, P3) = \sqrt{(18 - 43)^2 + (0 - 0)^2}$$

$$d(P1, P3) = 25$$

$$d(P1, P4) = \sqrt{(18 - 42)^2 + (0 - 0)^2}$$

$$d(P1, P4) = 24$$

$$d(P1, P5) = \sqrt{(18 - 27)^2 + (0 - 0)^2}$$

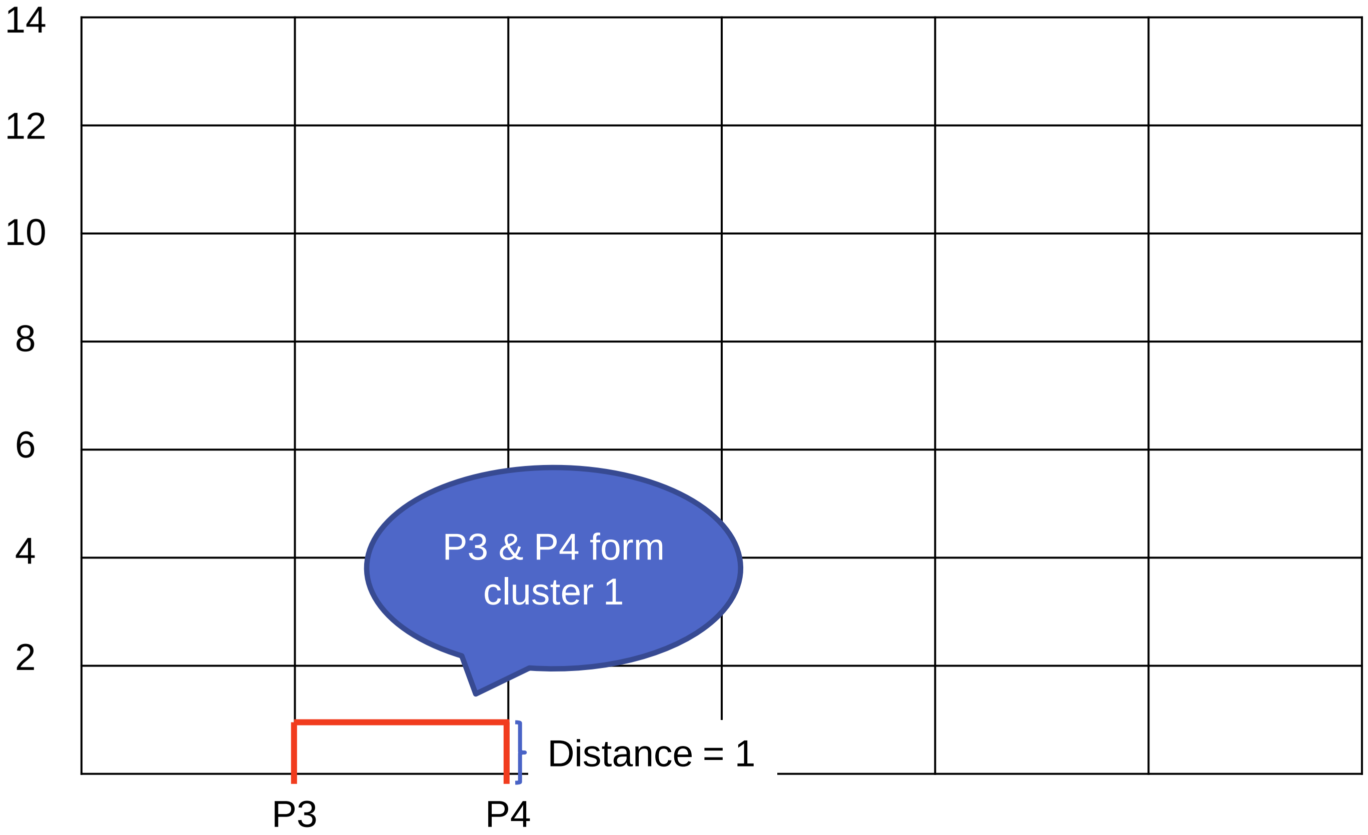
$$d(P1, P5) = 9$$

$$d(P1, P5) = \sqrt{(18 - 25)^2 + (0 - 0)^2}$$

$$d(P1, P6) = 7$$

Agglomerative Clustering – Example contd.

- Dendrogram



Agglomerative Clustering – Example contd.

- Distance Matrix

	P1 (18, 0)	P2 (22, 0)	P3 & P4	P5 (27, 0)	P6 (25, 0)
P1 (18, 0)	0				
P2 (22, 0)	4	0			
P3 & P4	24	20	0		
P5 (27, 0)	9	5	15		
P6 (25, 0)	7	3	17	2	0

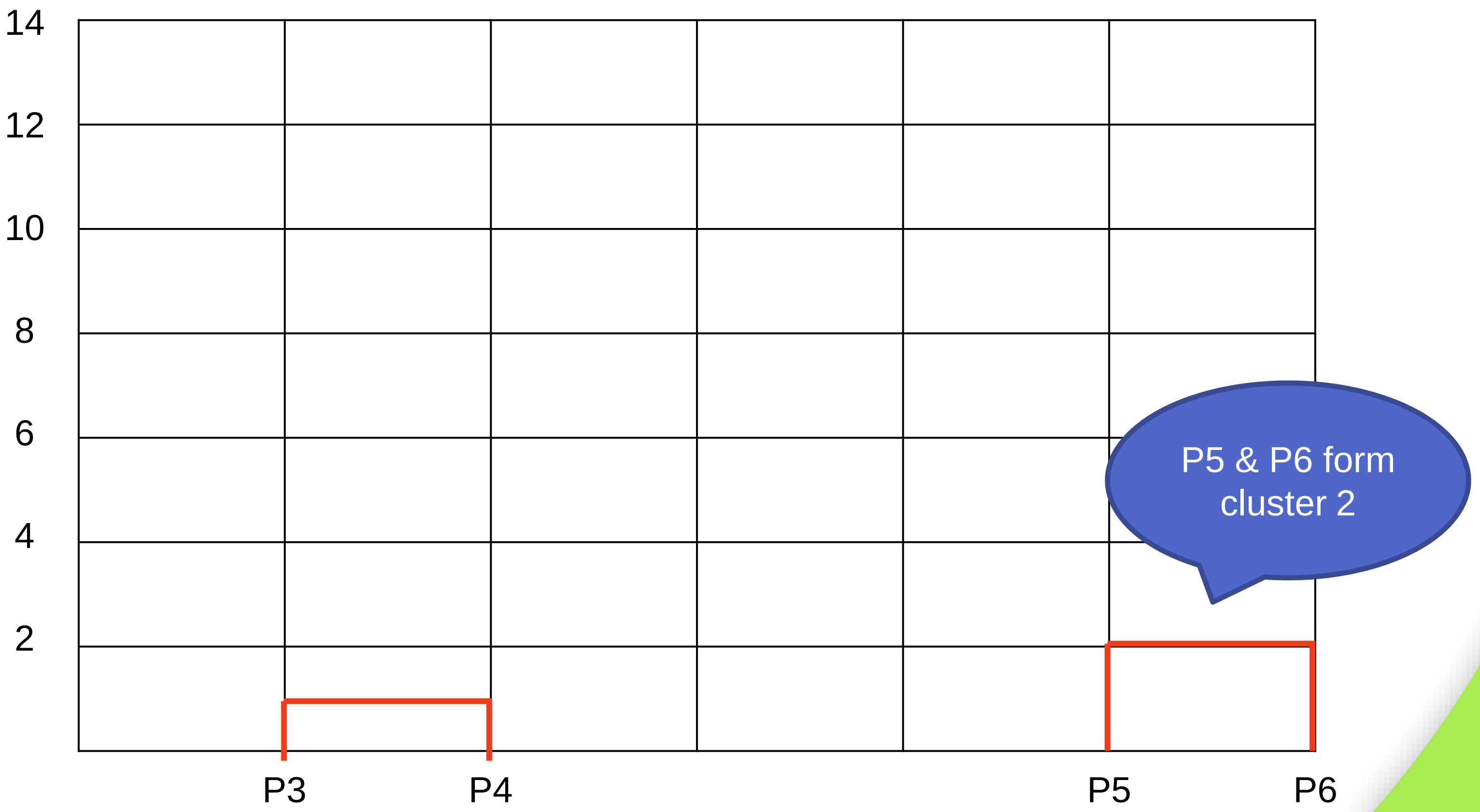
Shortest Distance

P1 to P3P4 = min(P1->P3, P1->P4)
= min(25,24) = 24

P2 to P3P4 = min(P2->P3, P2->P4)
= min(21,20) = 20

Agglomerative Clustering – Example contd.

- Dendrogram



Agglomerative Clustering – Example contd.

- Distance Matrix

	P1 (18, 0)	P2 (22, 0)	P3 & P4	P5 & P6
P1 (18, 0)	0			
P2 (22, 0)	4	0		
P3 & P4	24	20		
P5 & P6	7	3	15	0

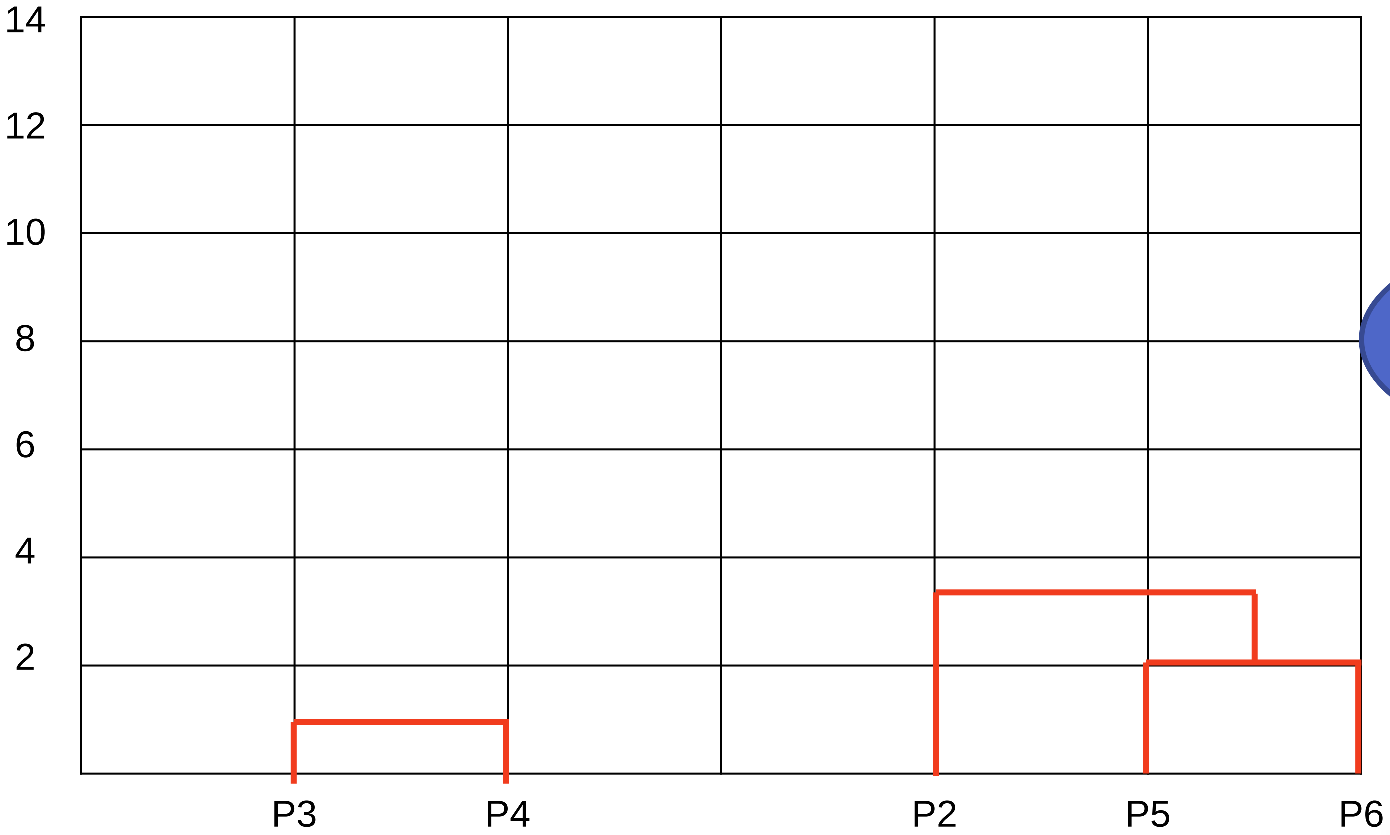
Shortest Distance

P1 to P5P6 = min(P1->P5, P1->P6)
= min(9, 7) = 7

P3P4 to P5P6 = min(P3P4->P5, P3P4->P6)
= min(15, 17) = 15

Agglomerative Clustering – Example contd.

- Dendrogram



P2 & P5P6 form cluster 3

Agglomerative Clustering – Example contd.

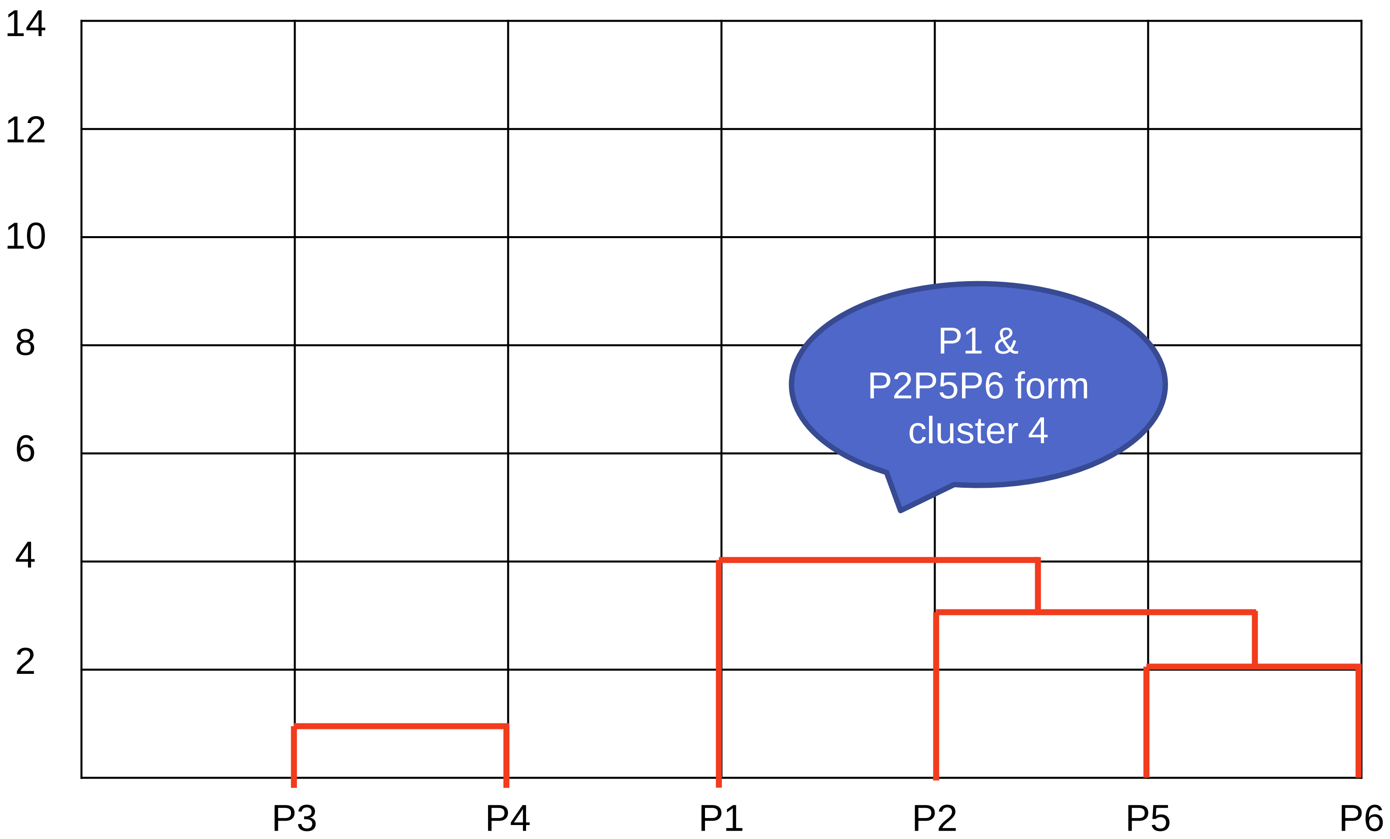
- Distance Matrix

	P1 (18, 0)	P3 & P4	P2, P5 & P6
P1 (18, 0)	0		
P3 & P4	24		
P2, P5 & P6	4	15	0

Shortest Distance

Agglomerative Clustering – Example contd.

- Dendrogram



Agglomerative Clustering – Example contd.

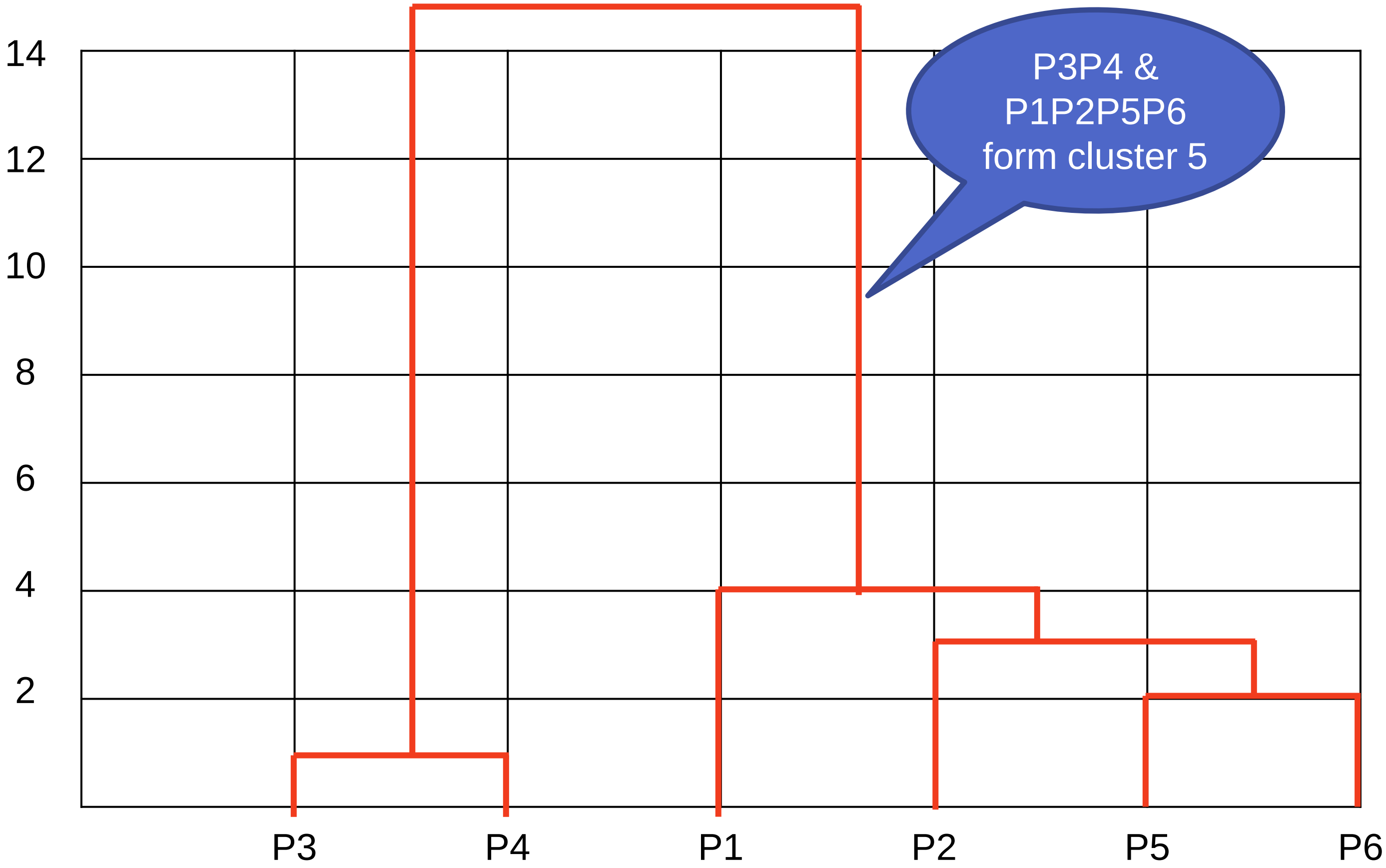
- Distance Matrix

	P3 & P4	P1, P2, P5 & P6
P3 & P4	0	
P1, P2, P5 & P6	15	0

The only Distance Left

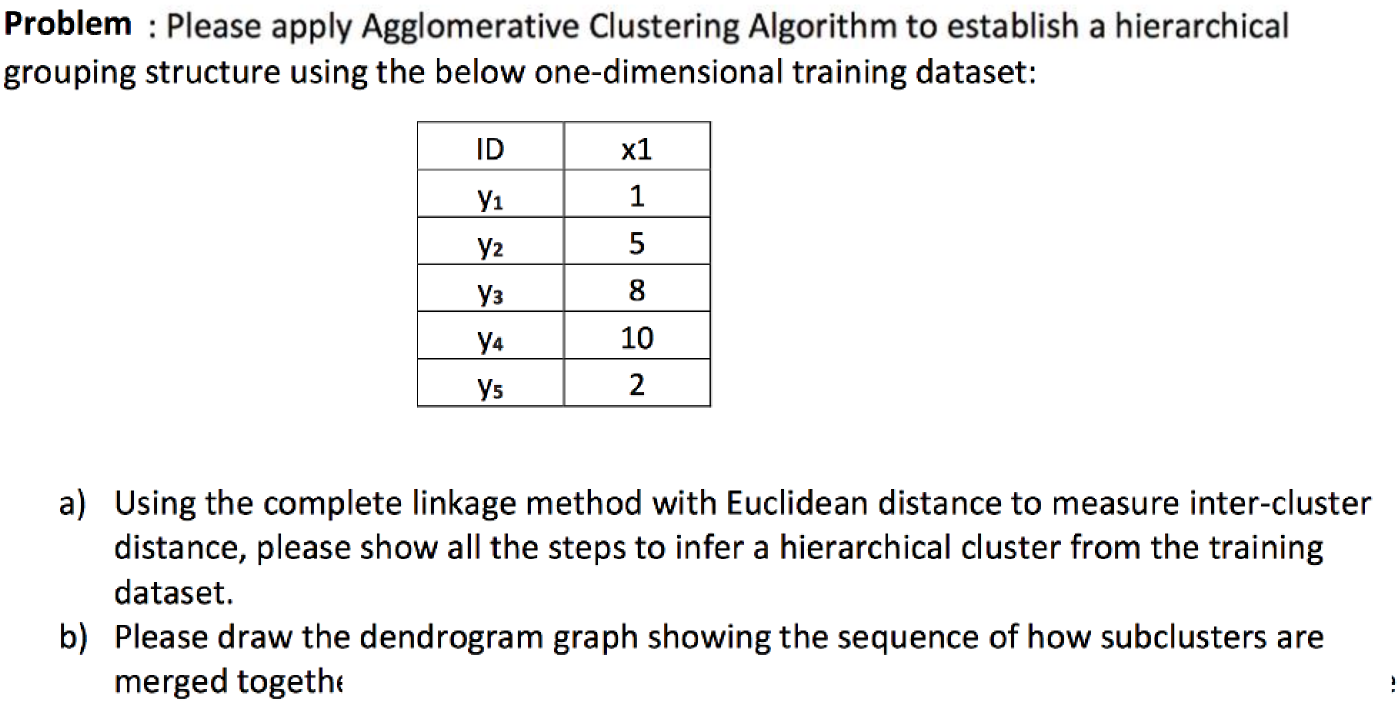
Agglomerative Clustering – Example contd.

- Dendrogram



P3P4 &
P1P2P5P6
form cluster 5

Exercise



Problem : Please apply Agglomerative Clustering Algorithm to establish a hierarchical grouping structure using the below one-dimensional training dataset:

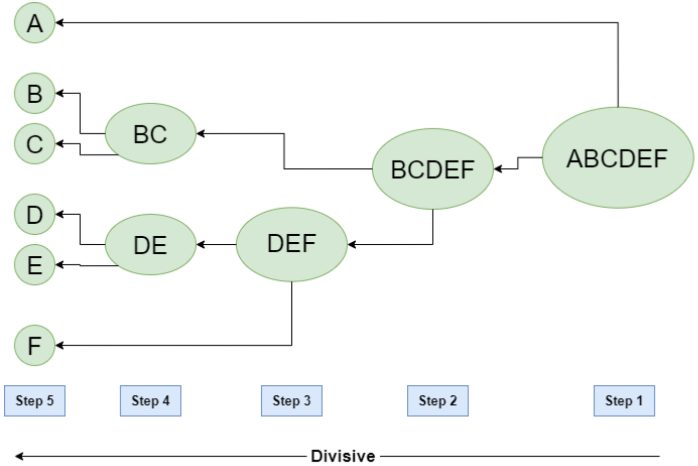
ID	x1
y_1	1
y_2	5
y_3	8
y_4	10
y_5	2

a) Using the complete linkage method with Euclidean distance to measure inter-cluster distance, please show all the steps to infer a hierarchical cluster from the training dataset.

b) Please draw the dendrogram graph showing the sequence of how subclusters are merged togethe

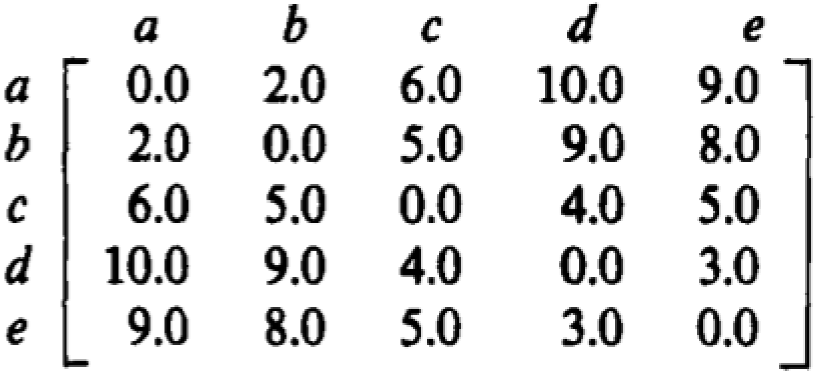
Types of Hierarchical Clustering - Divisive

- The Divisive Hierarchical clustering is precisely the opposite of the Agglomerative Hierarchical clustering
- In Divisive Hierarchical clustering,
  - We take into account all of the data points as a single cluster and in every iteration
  - We separate the data points from the clusters which aren’t comparable
  - In the end, we are left with N clusters



Divisive Clustering Example

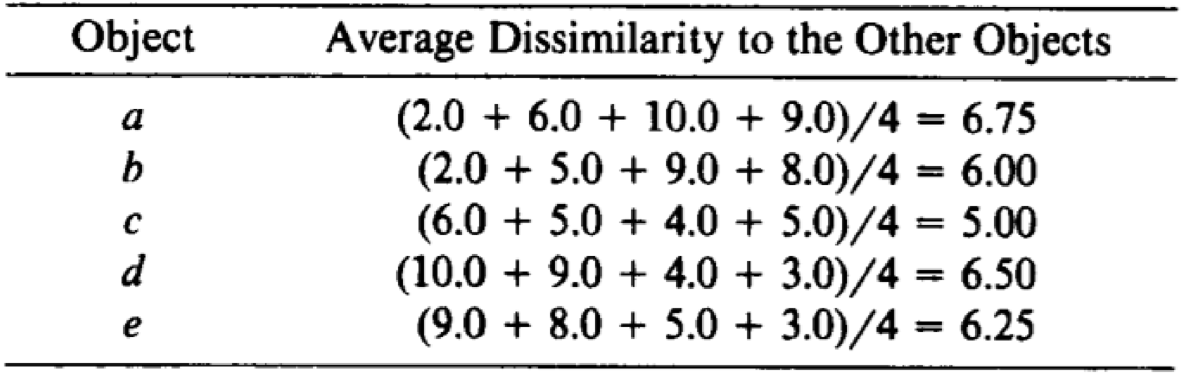
- At each step, a divisive method splits up a cluster into two smaller ones, until finally all clusters contain only a single element
  - This means that the hierarchy is again built in n - 1 steps when the data set contains n objects
  - The input data consist of a matrix of dissimilarities



$$\begin{array}{c|ccccc|}
 & a & b & c & d & e \\
a & 0.0 & 2.0 & 6.0 & 10.0 & 9.0 \\
b & 2.0 & 0.0 & 5.0 & 9.0 & 8.0 \\
c & 6.0 & 5.0 & 0.0 & 4.0 & 5.0 \\
d & 10.0 & 9.0 & 4.0 & 0.0 & 3.0 \\
e & 9.0 & 8.0 & 5.0 & 3.0 & 0.0
\end{array}$$

Divisive Clustering – Example contd.

- In the first step, the algorithm has to split up the data set into two clusters
  - To make this precise, we have to define the dissimilarity between an object and a group of objects
  - We use the average dissimilarity for this purpose, so we look for the object for which the average dissimilarity to all other objects is largest

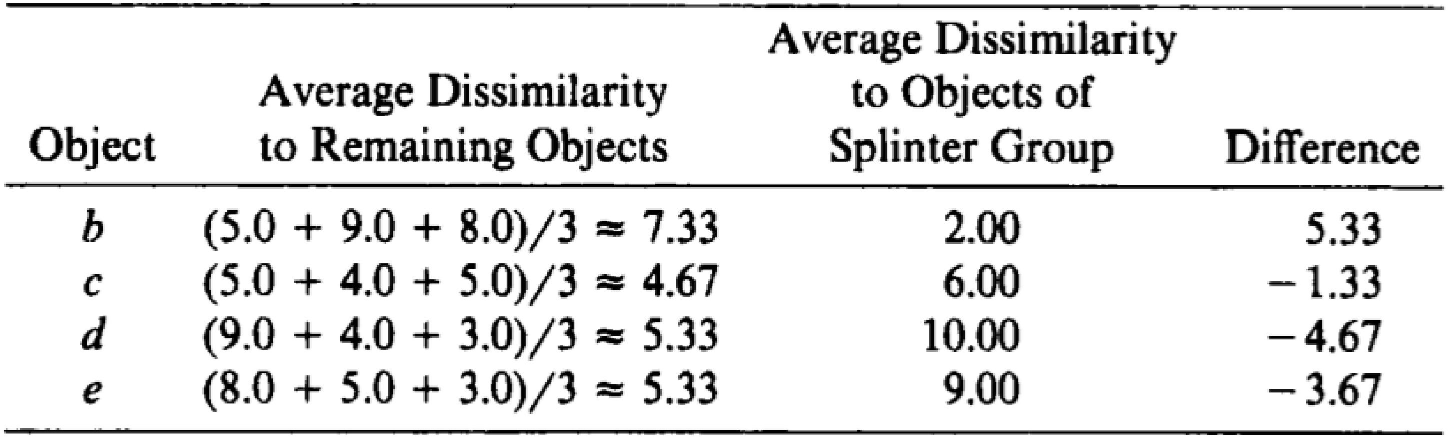


Object	Average Dissimilarity to the Other Objects
a	(2.0 + 6.0 + 10.0 + 9.0)/4 = 6.75
b	(2.0 + 5.0 + 9.0 + 8.0)/4 = 6.00
c	(6.0 + 5.0 + 4.0 + 5.0)/4 = 5.00
d	(10.0 + 9.0 + 4.0 + 3.0)/4 = 6.50
e	(9.0 + 8.0 + 5.0 + 3.0)/4 = 6.25

So object a is chosen to initiate the so-called splinter group. At this stage we have the groups {a} and { b, c, d, e}

Divisive Clustering – Example contd.

- For each object of the larger group we compute the average dissimilarity with the remaining objects, and compare it to the average dissimilarity with the objects of the splinter group:



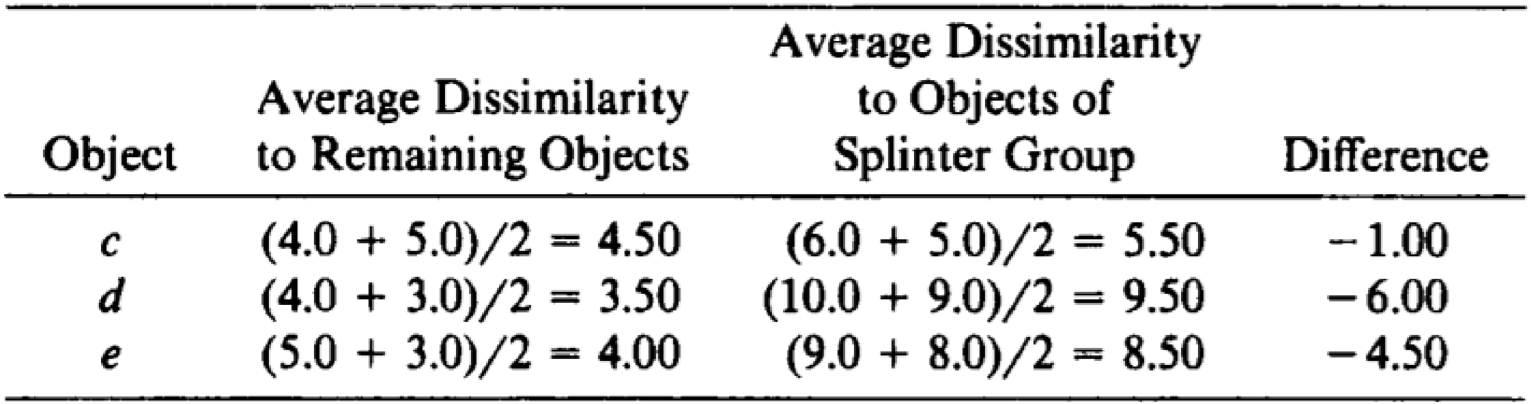
Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
b	$(5.0 + 9.0 + 8.0)/3 \approx 7.33$	2.00	5.33
c	$(5.0 + 4.0 + 5.0)/3 \approx 4.67$	6.00	−1.33
d	$(9.0 + 4.0 + 3.0)/3 \approx 5.33$	10.00	−4.67
e	$(8.0 + 5.0 + 3.0)/3 \approx 5.33$	9.00	−3.67

The difference is largest for object b, which lies much further from the remaining objects than from the splinter group

Object b changes sides, so the new splinter group is { a, b} and the remaining group becomes { c, d, e 1

Divisive Clustering – Example contd.

- When we repeat the computations we find

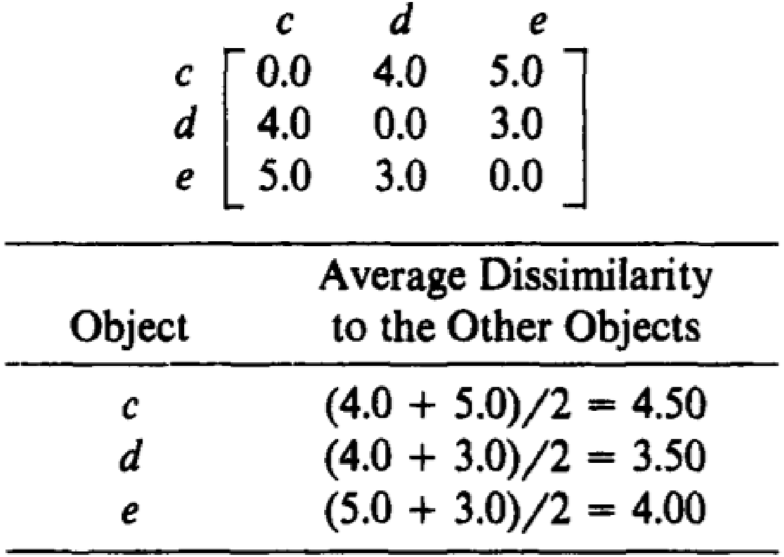


Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
c	(4.0 + 5.0)/2 = 4.50	(6.0 + 5.0)/2 = 5.50	−1.00
d	(4.0 + 3.0)/2 = 3.50	(10.0 + 9.0)/2 = 9.50	−6.00
e	(5.0 + 3.0)/2 = 4.00	(9.0 + 8.0)/2 = 8.50	−4.50

  - The remaining objects have more quarrels with the splinter group than with each other
  - Therefore, no further moves are made
  - The process stops and we have completed the first divisive step, which splits the data into the clusters {a, b} and { c, d, e}

Divisive Clustering – Example contd.

- In the next step, we divide the biggest cluster that is, the cluster with the largest diameter
  - The diameter of a cluster is just the largest dissimilarity between two of its objects
  - Applying the previous procedure to {c, d, e}

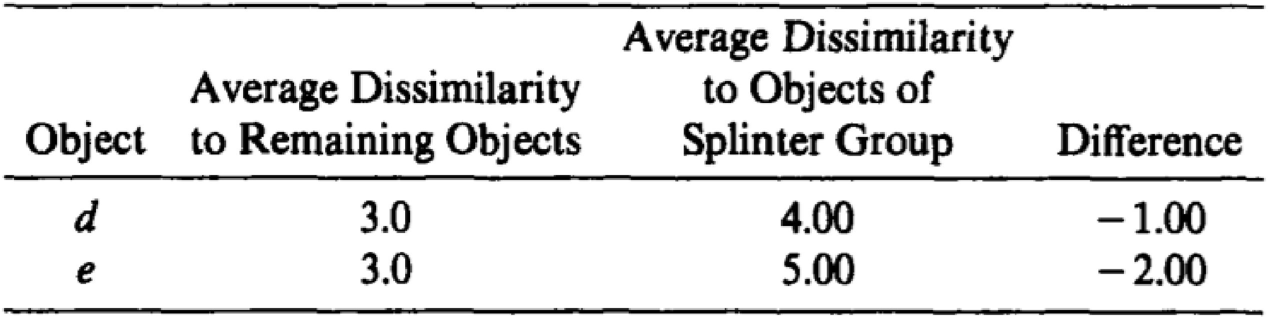


$$\begin{array}{c} \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccc} \quad c & \quad d & \quad e \end{array} \\ \begin{bmatrix} 0.0 & 4.0 & 5.0 \\ 4.0 & 0.0 & 3.0 \\ 5.0 & 3.0 & 0.0 \end{bmatrix} \end{array}$$

Object	Average Dissimilarity to the Other Objects
c	$(4.0 + 5.0)/2 = 4.50$
d	$(4.0 + 3.0)/2 = 3.50$
e	$(5.0 + 3.0)/2 = 4.00$

Divisive Clustering – Example contd.

- We obtain object c. Afterward, we find

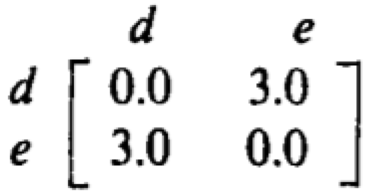


Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
d	3.0	4.00	−1.00
e	3.0	5.00	−2.00

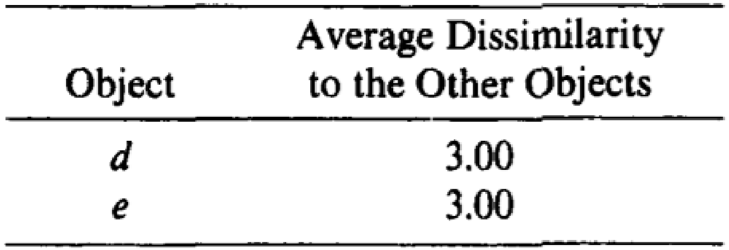
  - The process stops because all differences are negative
  - Therefore, our second step divides { c, d, e} into { c} and { d, e}, so we are left with the clusters {a, b), { c}, and { d, e}
  - The cluster { c} is called a singleton because it contains only one object

Divisive Clustering – Example contd.

- The cluster {a, b} has diameter 2 and that of { d, e) is 3
    - Therefore, we have to divide the cluster { d, e} with dissimilarity matrix



$$\begin{array}{c|cc} & d & e \\ \hline d & 0.0 & 3.0 \\ e & 3.0 & 0.0 \end{array}$$

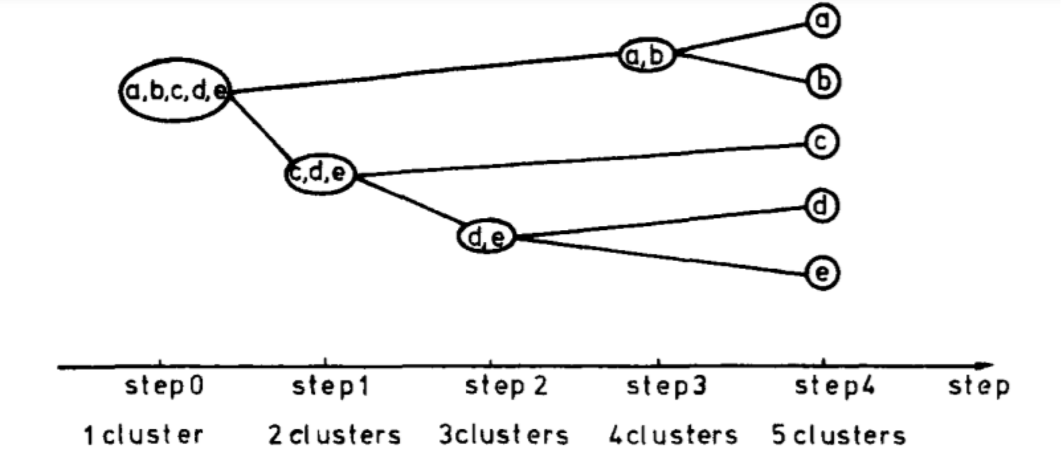


Object	Average Dissimilarity to the Other Objects
d	3.00
e	3.00

    - We may choo-e either object to begin the splinter group with as both have same dissimilarity
    - Let us choose object d, so we obtain { d } and { e}.

Divisive Clustering – Example contd.

- Step 3 leaves us with four clusters: (a, b), {c), {d}, and {e}
- In the fourth step we have to split up the cluster {a, b), because all the others contain only a single object
    - After this fourth step we only have singleton clusters {a}, { b}, { c}, { d }, and { e), so the algorithm stops



Thank you