Introduction to K-Means Clustering

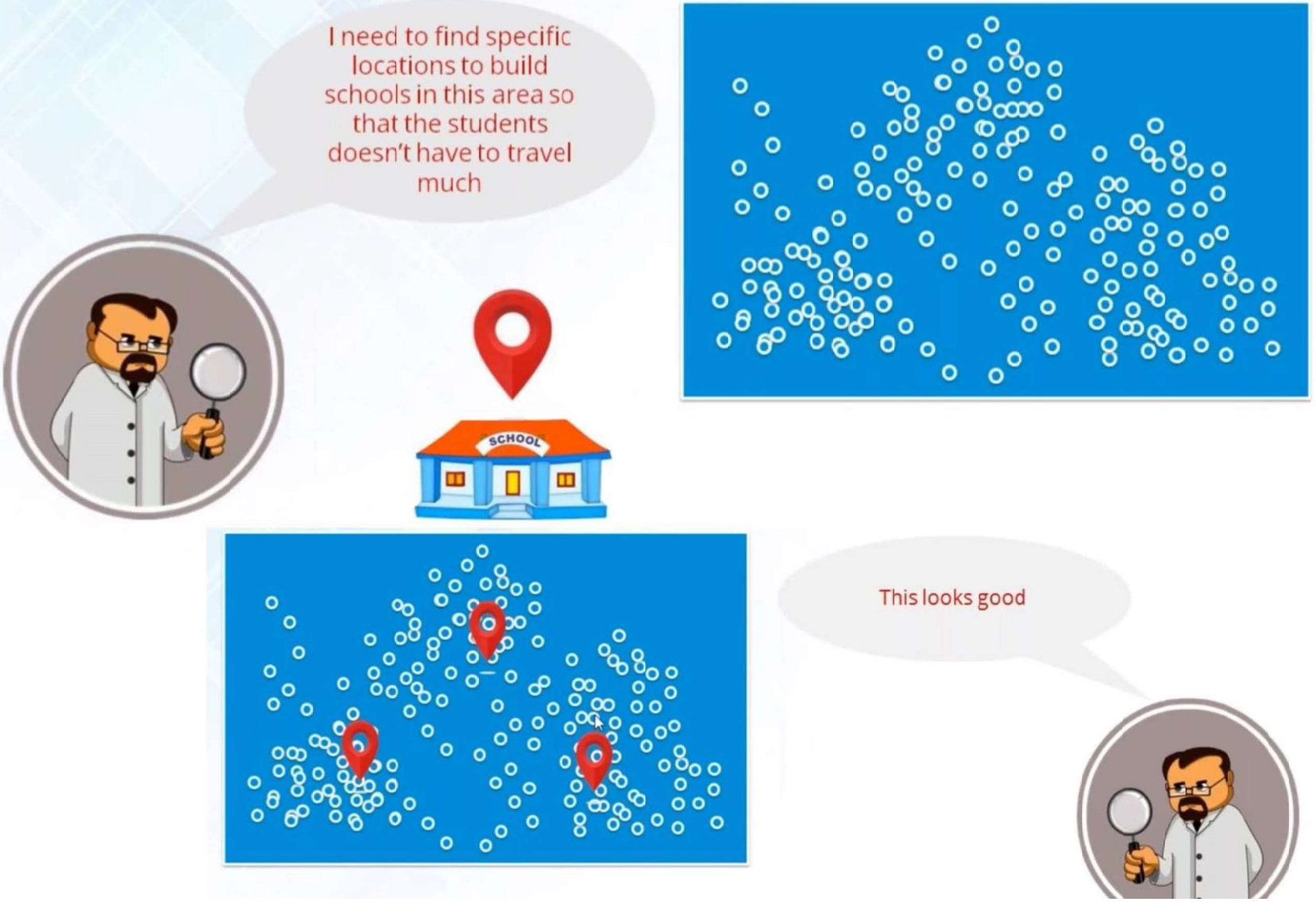
K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

> ***The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.***

K-Means Clustering

Example

The plot of students in an area is as given below,



Clustering and the K-means algorithm

Original Image

2 colors

4 colors

8 colors

Introduction to EM algorithm and K-Means Clustering

EM algorithm

- The goal of EM clustering is to estimate **the means and standard deviations** for each cluster, so as to maximize the likelihood of the observed data. (Distribution)

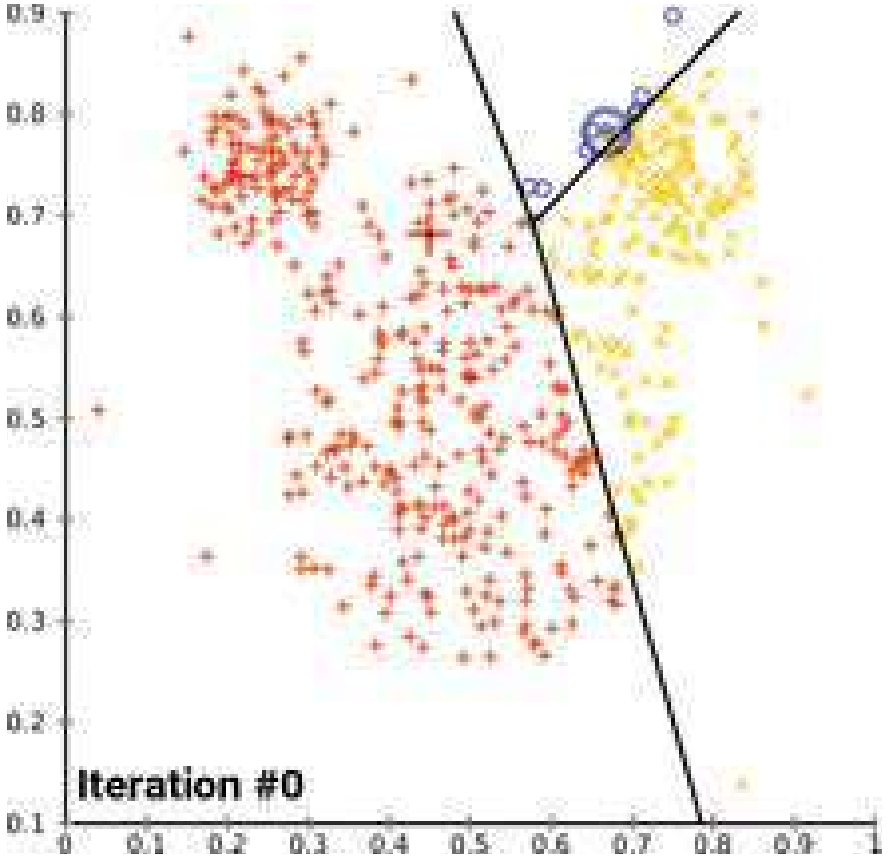
Likelihood: It is the probability of observing the given data given the parameters of the model.

The expectation-Maximization algorithm tries to use the existing data to determine the optimum values for these variables and then finds the model parameters.

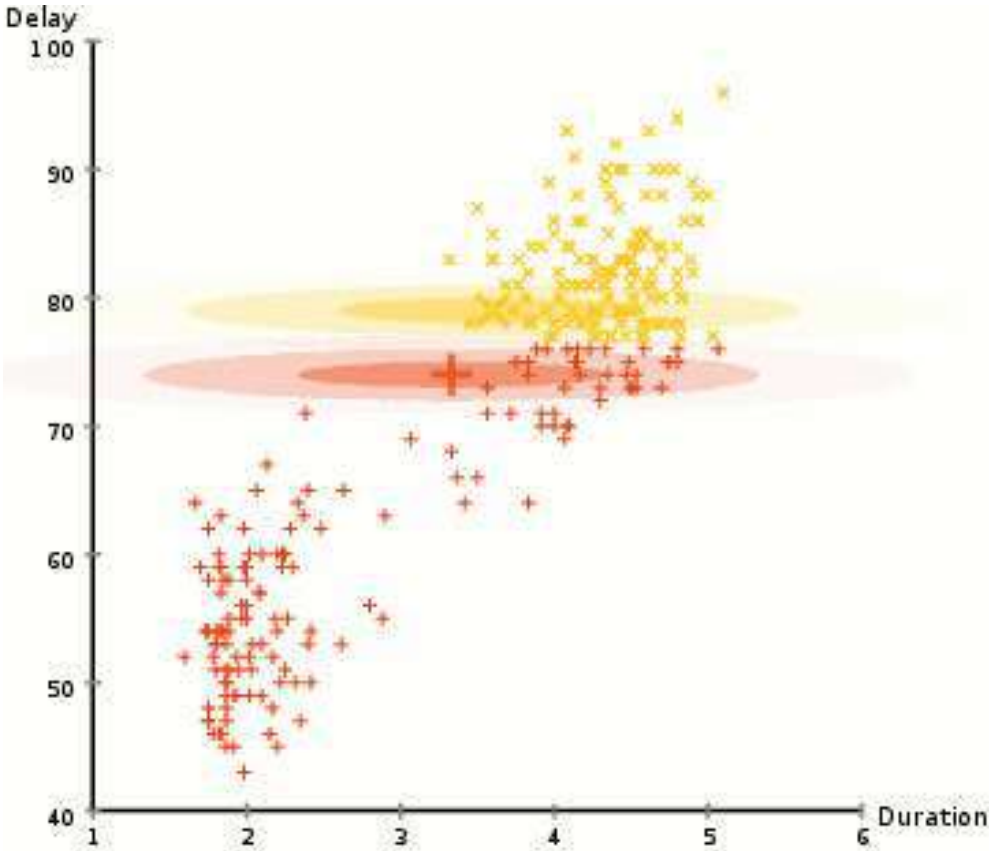
K-Means Clustering

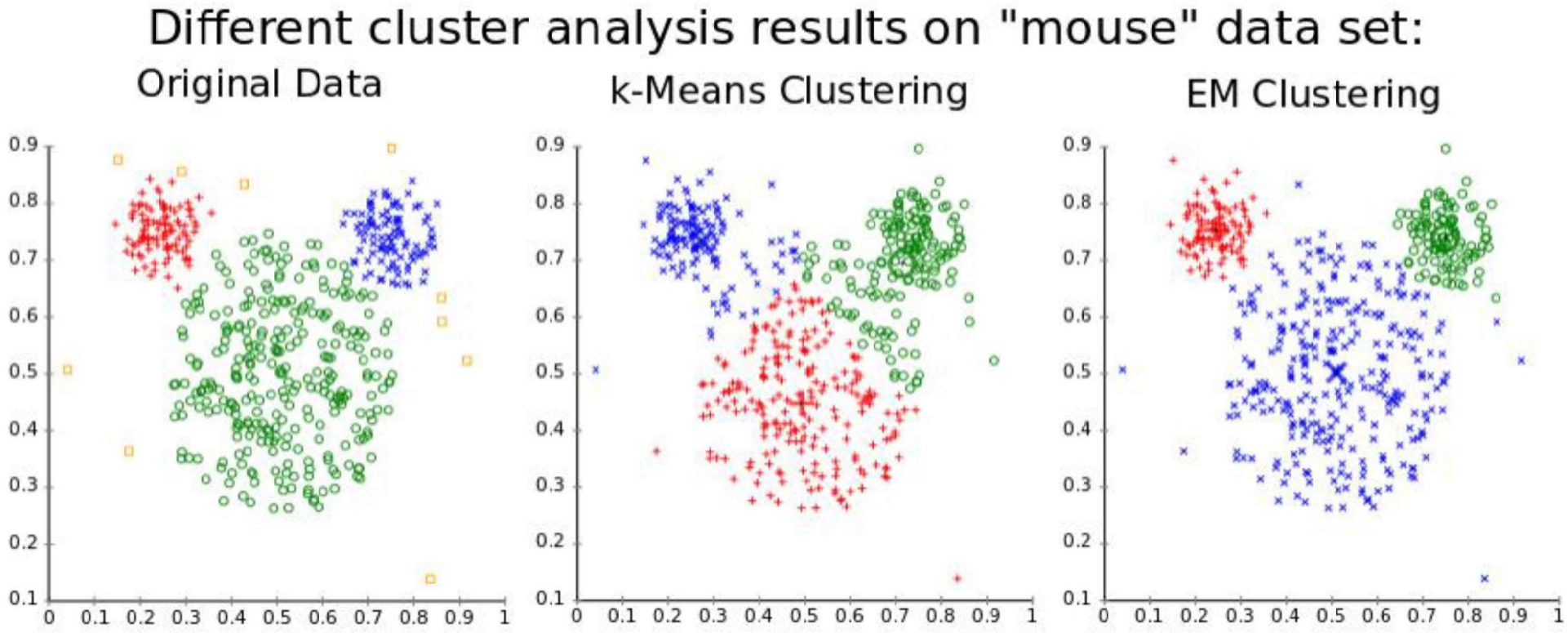
- The algorithm will categorize the items into ‘K’ groups of similarity. To calculate that similarity, we use **Euclidean distance as a measurement**.- Means.

K means Algorithm



EM algorithm

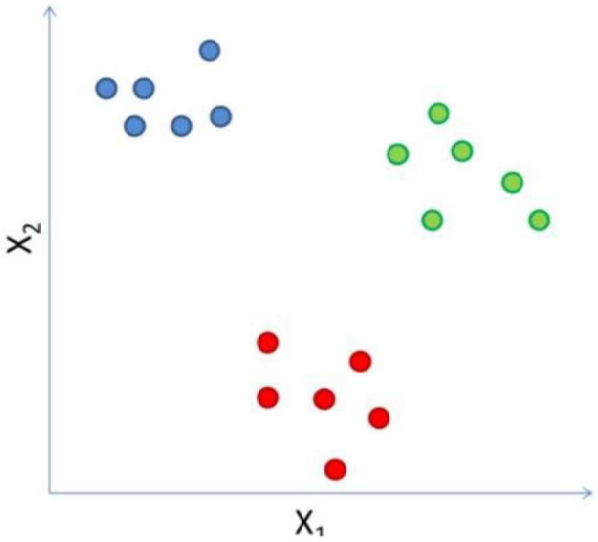




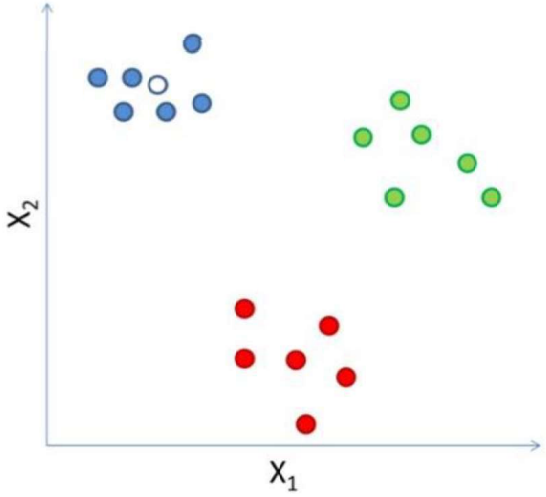
k-means clustering vs. EM clustering on an artificial dataset ("mouse"). The tendency of *k*-means to produce equal-sized clusters leads to bad results here, while EM benefits from the Gaussian distributions with different radius present in the data set.

Gaussian Mixture Models

GMM used to separate data into clusters.

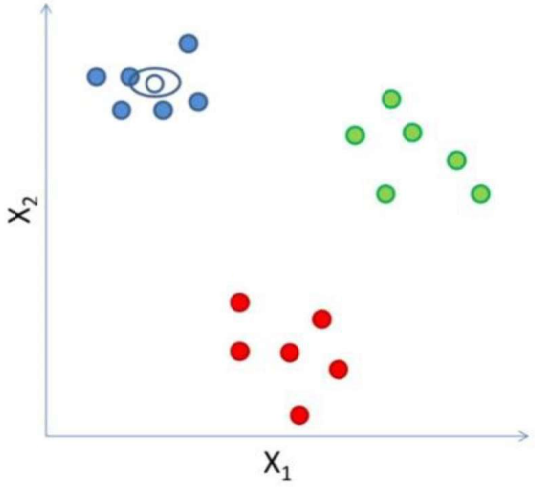


Like K-Means, GMM clusters have centers.



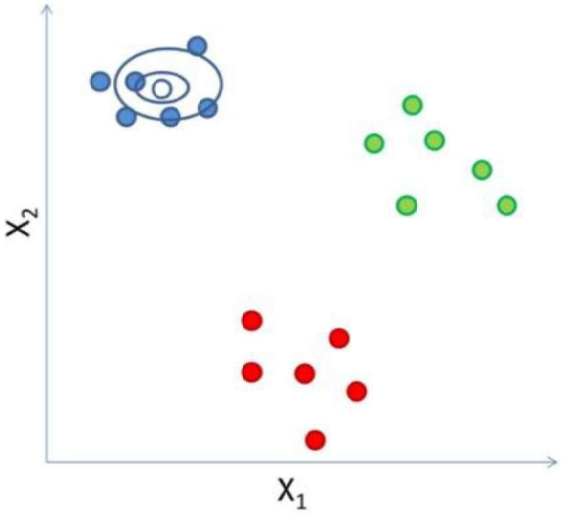
Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.



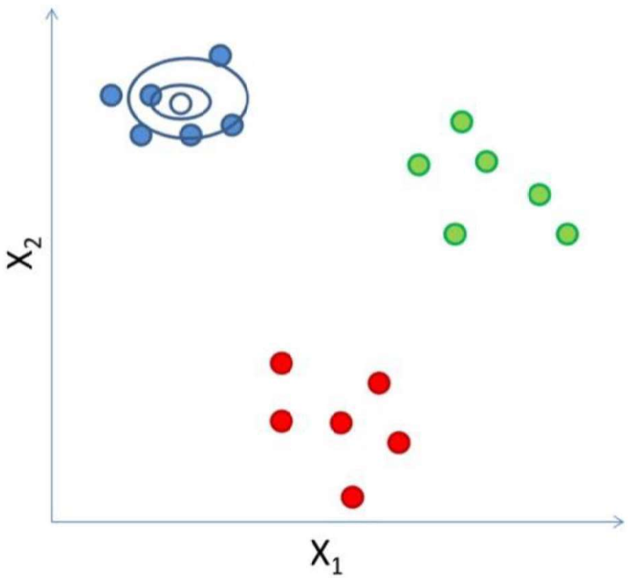
Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.



Probability distribution gives the probabilities of occurrence of different possible outcomes for an experiment.

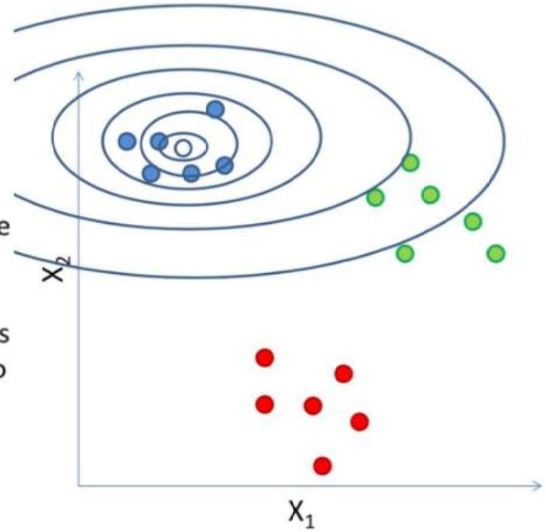
Gaussian Mixture Models



Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.

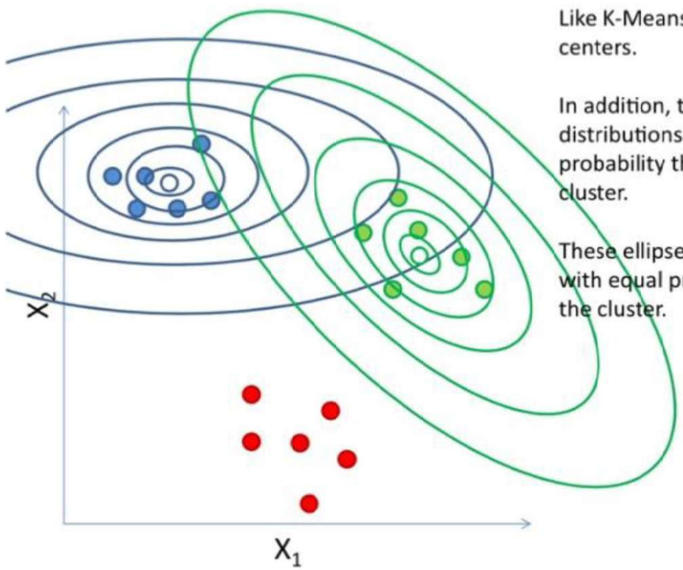
These ellipses show "level sets": lines with equal probability of belonging to the cluster.



Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.

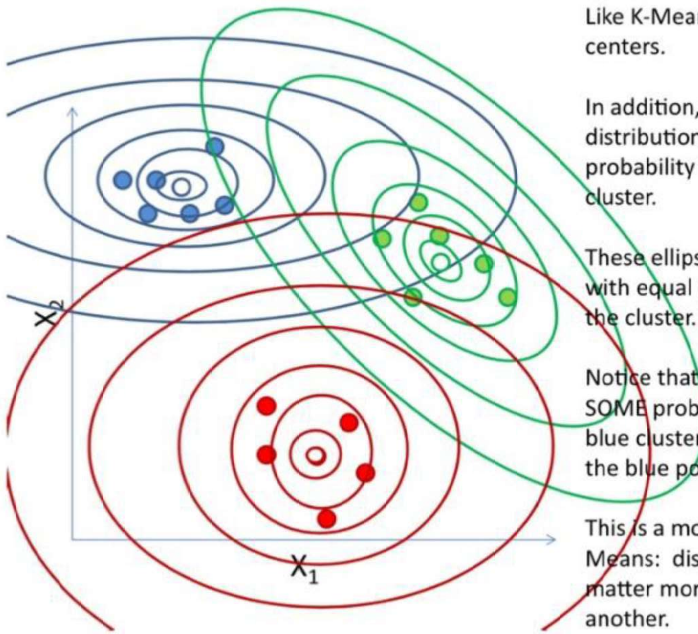
These ellipses show "level sets": lines with equal probability of belonging to the cluster.



Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.

These ellipses show "level sets": lines with equal probability of belonging to the cluster.



Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.

These ellipses show "level sets": lines with equal probability of belonging to the cluster.

Notice that green points still have SOME probability of belonging to the blue cluster, but it's much lower than the blue points.

This is a more complex model than K-Means: distance from the center can matter more in one direction than another.

K-Means Clustering

Algorithm *k-means*

1. Randomly choose K data items from X as initial centroids.

2. Repeat

   - Assign each data point to the cluster which has the closest centroid.

   - Calculate new cluster centroids.

   Until the convergence criteria is met.

Algorithm

Algorithmic steps for k-means clustering

Let $X = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \ldots\ldots, v_c\}$ be the set of centers.

1) Randomly select 'c' cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

where, 'c_i' represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

Detail: K-Means Clustering

Let's now take an example to understand how K-Means actually works:

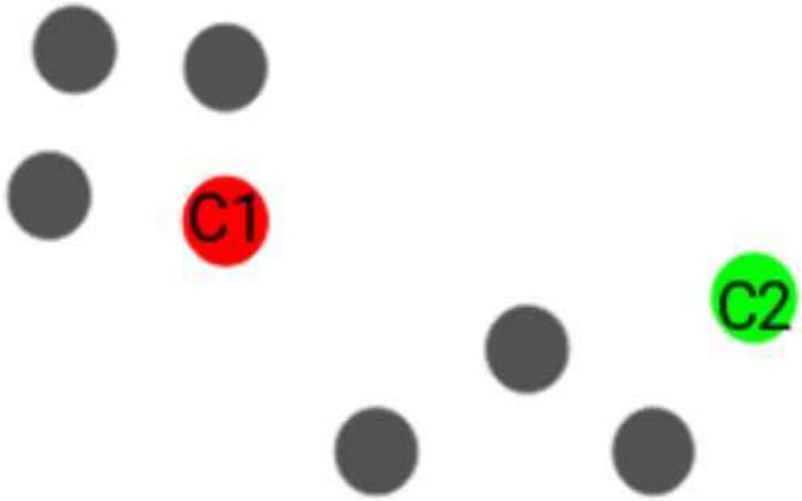
We have these 8 points and we want to apply k-means to create clusters for these points. Here's how we can do it.

Step 1: Choose the number of clusters *k*

The first step in k-means is to pick the number of clusters, k.

Step 2: Select k random points from the data as centroids

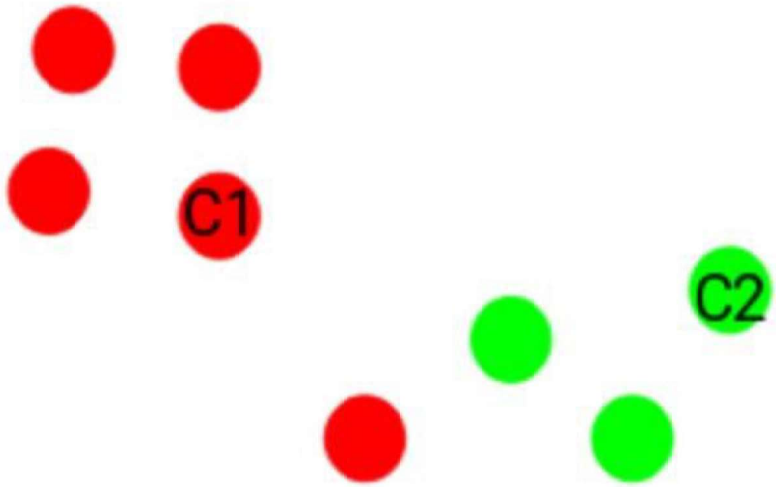
Next, we randomly select the centroid for each cluster. Let's say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid:



Here, the red and green circles represent the centroid for these clusters.

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

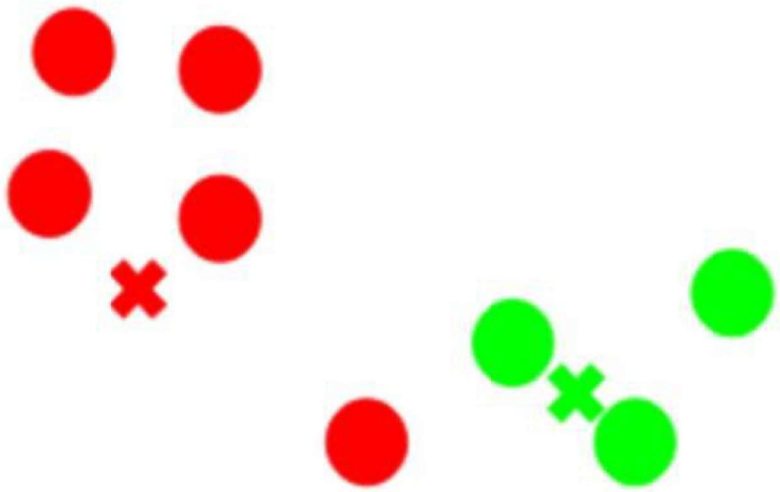
Once we have initialized the centroids, we assign each point to the closest cluster centroid:



Here you can see that the points which are closer to the red point are assigned to the red cluster whereas the points which are closer to the green point are assigned to the green cluster.

Step 4: Recompute the centroids of newly formed clusters

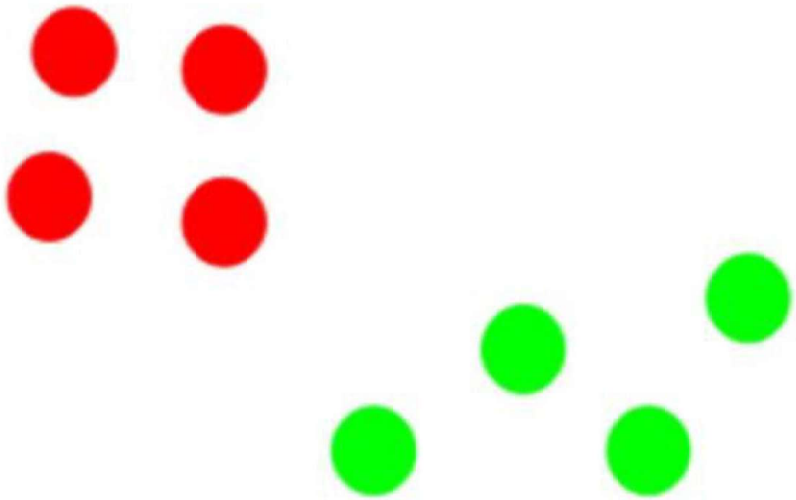
Now, once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters:



Here, the red and green crosses are the new centroids.

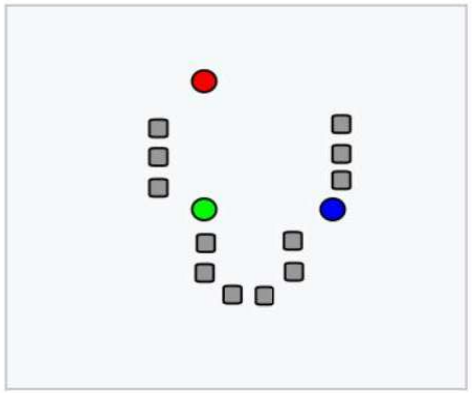
Step 5: Repeat steps 3 and 4

We then repeat steps 3 and 4:

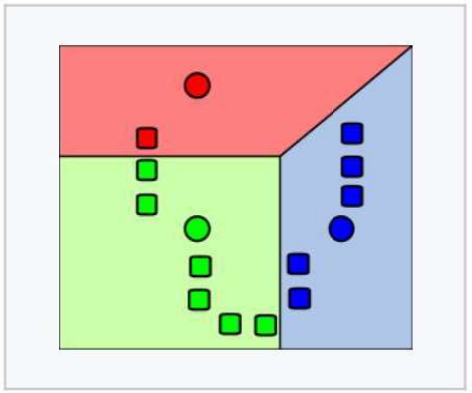


The step of computing the centroid and assigning all the points to the cluster based on their distance from the centroid is a single iteration. But wait – when should we stop this process? It can't run till eternity, right?

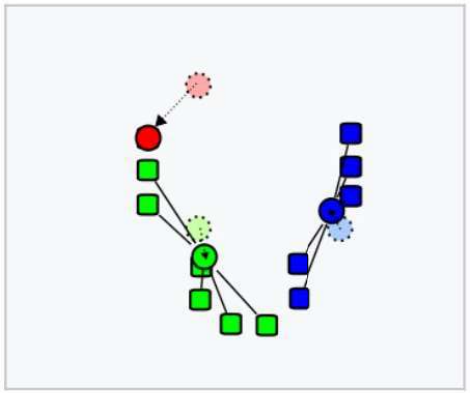
Demonstration of the standard algorithm



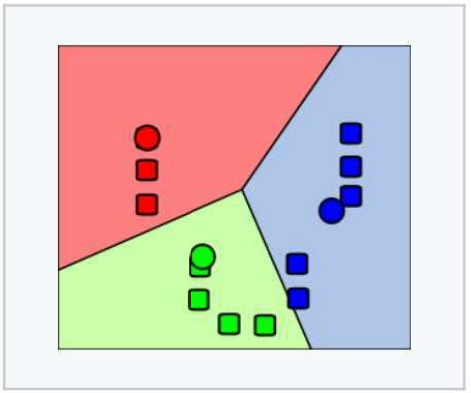
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the *k* clusters becomes the new mean.



4. Steps 2 and 3 are repeated until convergence has been reached.

Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

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

We can stop the algorithm if the centroids of newly formed clusters are not changing. Even after multiple iterations, if we are getting the same centroids for all the clusters, we can say that the algorithm is not learning any new pattern and it is a sign to stop the training.

Another clear sign that we should stop the training process if the points remain in the same cluster even after training the algorithm for multiple iterations.

Finally, we can stop the training if the maximum number of iterations is reached. Suppose if we have set the number of iterations as 100. The process will repeat for 100 iterations before stopping.

K-Means Clustering Algorithm – Solved Example

- Use K Means clustering to cluster the following data into two groups.
- Data Points: { 2, 4, 10, 12, 3, 20, 30, 11, 25 }
- The distance function used is Euclidean distance.
- Initial cluster centroid are M1 = 4 and M2 = 11.

K-Means Clustering Algorithm –Example

Initial Centroids:
M1: 4
M2: 11

Data Points	Distance to M1	Distance to M2	Cluster	New Cluster
2	2			
4	0			
10	6			
12	8			
3	1			
20	16			
30	26			
11	7			
25	21			

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

$$= \sqrt{(4-2)^2}$$

$$= 2$$

K-Means Clustering Algorithm –Example

Initial Centroids:
M1: 4
M2: 11

Data Points	Distance to M1	Distance to M2	Cluster	New Cluster
2	2	9		
4	0	7		
10	6	1		
12	8	1		
3	1	8		
20	16	9		
30	26	19		
11	7	0		
25	21	14		

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

$$= \sqrt{(11-4)^2}$$

Initial Centroids:
M1: 4
M2: 11

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	2	9	C1	
4	0	7	C1	
10	6	1	C2	
12	8	1	C2	
3	1	8	C1	
20	16	9	C2	
30	26	19	C2	
11	7	0	C2	
25	21	14	C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Initial Centroids:
M1: 4
M2: 11

Therefore
C1= {2, 4, 3}
C2= {10, 12, 20, 30, 11, 25}

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	2	9	C1	
4	0	7	C1	
10	6	1	C2	
12	8	1	C2	
3	1	8	C1	
20	16	9	C2	
30	26	19	C2	
11	7	0	C2	
25	21	14	C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Initial Centroids:

M1: 4

M2: 11

Therefore

C1= {2, 4, 3}

C2= {10, 12, 20, 30, 11, 25}

New Centroids:

M1: 3

M2: 18

$$\frac{2+4+3}{3}$$

$$\frac{10+12+20+30+11+25}{6}$$

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	2	9	C1	
4	0	7	C1	
10	6	1	C2	
12	8	1	C2	
3	1	8	C1	
20	16	9	C2	
30	26	19	C2	
11	7	0	C2	
25	21	14	C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 3
M2: 18

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2			C1	
4			C1	
10			C2	
12			C2	
3			C1	
20			C2	
30			C2	
11			C2	
25			C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 3
M2: 18

Data Points	Distance to M1	Distance to M2	Cluster	New Cluster
2	1	16	C1	
4	1	14	C1	
10	7	8	C2	
12	9	6	C2	
3	0	15	C1	
20	17	2	C2	
30	27	12	C2	
11	8	7	C2	
25	22	7	C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 3
M2: 18

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	1	16	C1	C1
4	1	14	C1	C1
10	7	8	C2	C1
12	9	6	C2	C2
3	0	15	C1	C1
20	17	2	C2	C2
30	27	12	C2	C2
11	8	7	C2	C2
25	22	7	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 3
M2: 18

Therefore
C1= {2, 4, 10, 3}
C2= {12, 20, 30, 11, 25}

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	1	16	C1	C1
4	1	14	C1	C1
10	7	8	C2	C1
12	9	6	C2	C2
3	0	15	C1	C1
20	17	2	C2	C2
30	27	12	C2	C2
11	8	7	C2	C2
25	22	7	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 3
M2: 18

Therefore
C1= {2, 4, 10, 3}
C2= {12, 20, 30, 11, 25}

New Centroids:
M1: 4.75
M2: 19.6

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	1	16	C1	C1
4	1	14	C1	C1
10	7	8	C2	C1
12	9	6	C2	C2
3	0	15	C1	C1
20	17	2	C2	C2
30	27	12	C2	C2
11	8	7	C2	C2
25	22	7	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 4.75
M2: 19.6

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2			C1	
4			C1	
10			C1	
12			C2	
3			C1	
20			C2	
30			C2	
11			C2	
25			C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Here, assign a new cluster column to the cluster column

K-Means Clustering Algorithm – Solved Example

Current Centroids:
M1: 4.75
M2: 19.6

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	2.75	17.6	C1	
4	0.75	15.6	C1	
10	5.25	9.6	C1	
12	7.25	7.6	C2	
3	1.75	16.6	C1	
20	15.25	0.4	C2	
30	25.25	10.4	C2	
11	6.25	8.6	C2	
25	20.25	5.4	C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 4.75
M2: 19.6

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	2.75	17.6	C1	C1
4	0.75	15.6	C1	C1
10	5.25	9.6	C1	C1
12	7.25	7.6	C2	C1
3	1.75	16.6	C1	C1
20	15.25	0.4	C2	C2
30	25.25	10.4	C2	C2
11	6.25	8.6	C2	C1
25	20.25	5.4	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 4.75
M2: 19.6

Therefore
C1= {2, 4, 10, 11, 12, 3}
C2= {20, 30, 25}

New Centroids:
M1: 7
M2: 25

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	2.75	17.6	C1	C1
4	0.75	15.6	C1	C1
10	5.25	9.6	C1	C1
12	7.25	7.6	C2	C1
3	1.75	16.6	C1	C1
20	15.25	0.4	C2	C2
30	25.25	10.4	C2	C2
11	6.25	8.6	C2	C1
25	20.25	5.4	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 7
M2: 25

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2			C1	
4			C1	
10			C1	
12			C1	
3			C1	
20			C2	
30			C2	
11			C1	
25			C2	

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 7
M2: 25

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	5	23	C1	C1
4	3	21	C1	C1
10	3	15	C1	C1
12	5	13	C1	C1
3	4	22	C1	C1
20	13	5	C2	C2
30	23	5	C2	C2
11	4	14	C1	C1
25	18	0	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Current Centroids:
M1: 7
M2: 25

Final Cluster are:
C1= {2, 4, 10, 11, 12, 3}
C2= {20, 30, 25}

Data Points	Distance to		Cluster	New Cluster
	M1	M2		
2	5	23	C1	C1
4	3	21	C1	C1
10	3	15	C1	C1
12	5	13	C1	C1
3	4	22	C1	C1
20	13	5	C2	C2
30	23	5	C2	C2
11	4	14	C1	C1
25	18	0	C2	C2

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

Here, the cluster (column) and the new cluster (column) are the same. So reached convergence.

K-Means Example

- Use the k-means algorithm and Euclidean distance to cluster the following 5 instances into 2 clusters.

Instances	X	Y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Instances	X	Y	A(1,1)	C(0,2)	cluster
A	1	1	0	1.4	1
B	1	0	1	2.2	1
C	0	2	1.4	0	2
D	2	4	3.2	2.8	2
E	3	5	4.5	4.2	2

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

Centroid((2/2),(1/2))=(1,0.5)

and(5/3,11/3)=(1.7,3.7)

Instances	X	Y	C1(1,0.5)	C2(1.7,3.7)	cluster
A	1	1	0.5	2.7	1
B	1	0	0.5	3.7	1
C	0	2	1.8	2.4	1
D	2	4	3.6	0.5	2
E	3	5	4.9	1.9	2

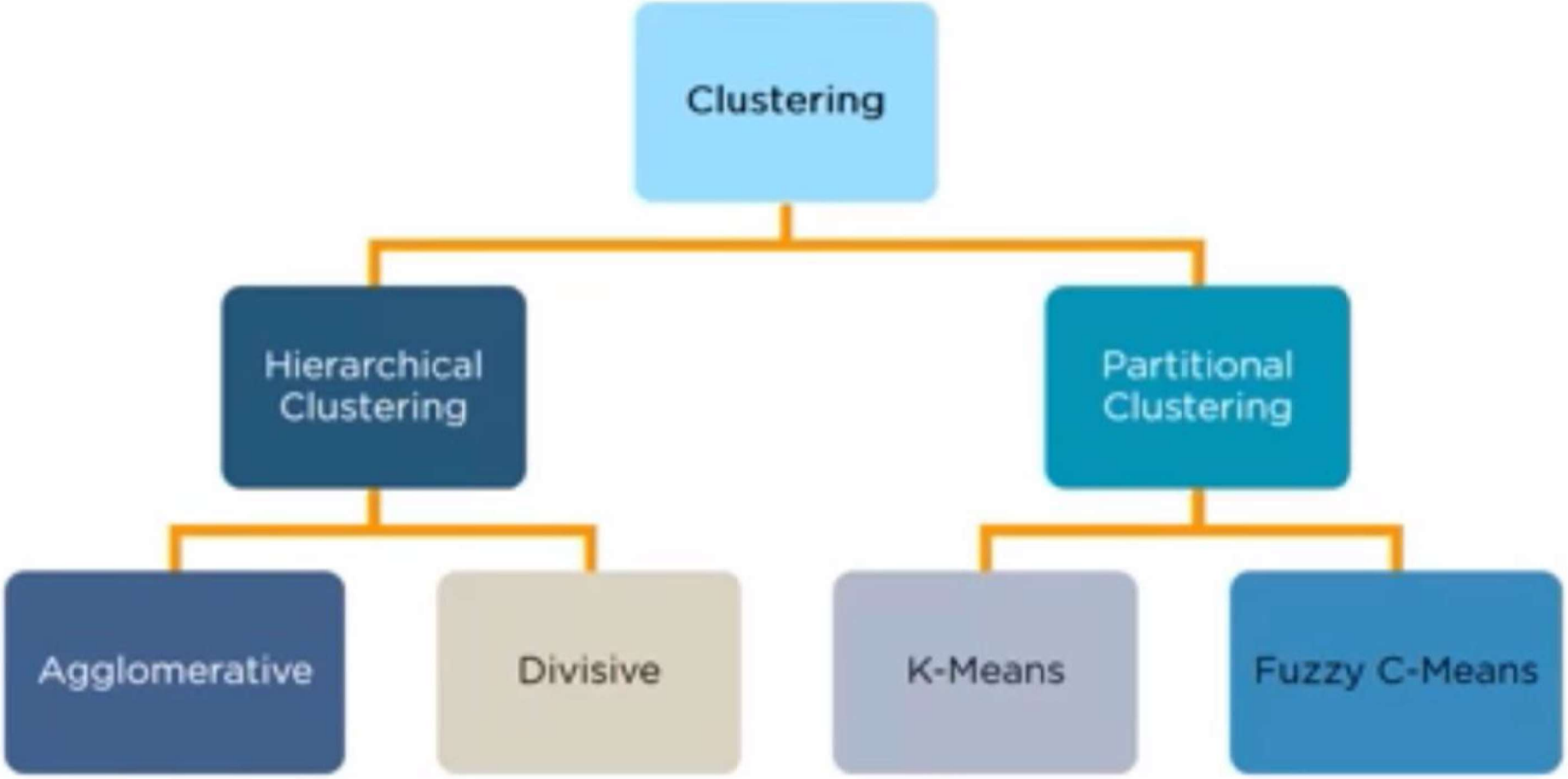
- Centroid(2/3,3/3)=(0.7,1) and (5/2,9/2)=(2.5,4.5)

Instances	X	Y	C1(0.7,1)	C2(2.5,4.5)	cluster
A	1	1	0.3	3.8	1
B	1	0	1.04	1.70	1
C	0	2	1.7	2.75	1
D	2	4	10.3	0.75	2
E	3	5	4.6	0.75	2

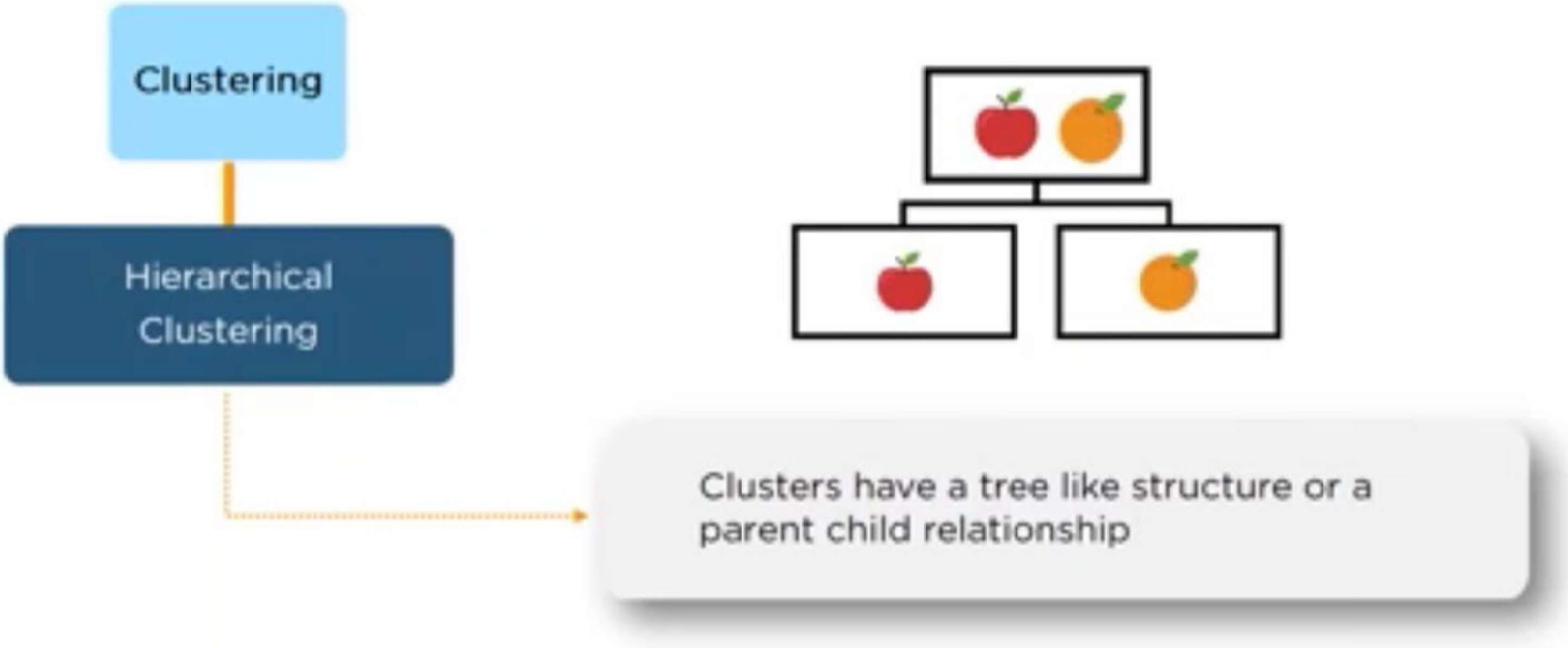
Use the k-means algorithm and Euclidean distance to cluster the following 8 instances into 3 clusters.

Instances	X	Y
A1	2	10
A2	2	5
A3	8	4
A4	5	8
A5	7	5
A6	6	4
A6	1	2
A8	4	9

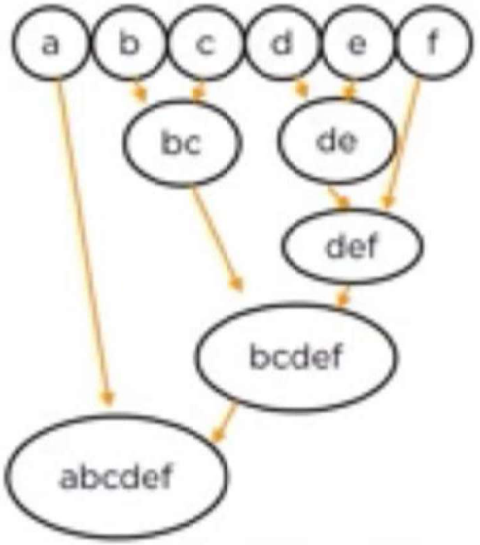
Types of Clustering



Types of Clustering

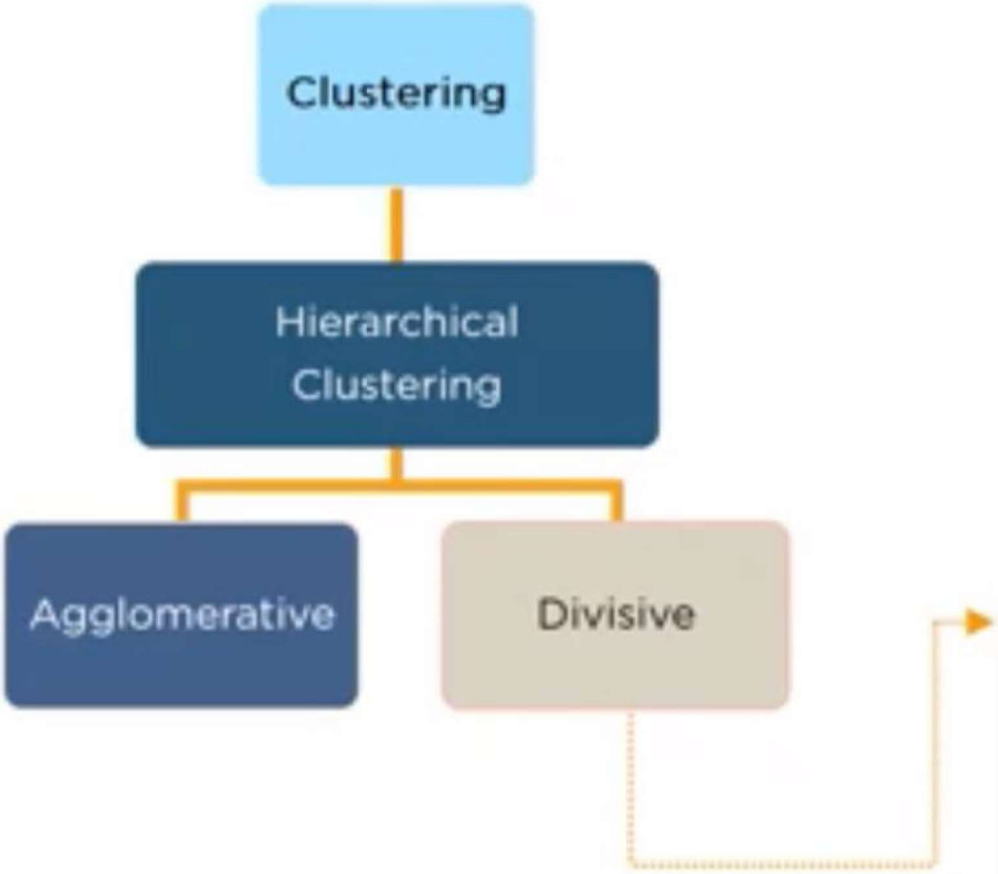


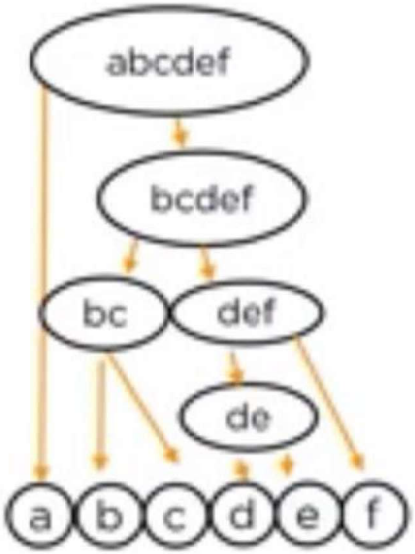
Types of Clustering



"Bottom up" approach: Begin with each element as a separate cluster and merge them into successively larger clusters

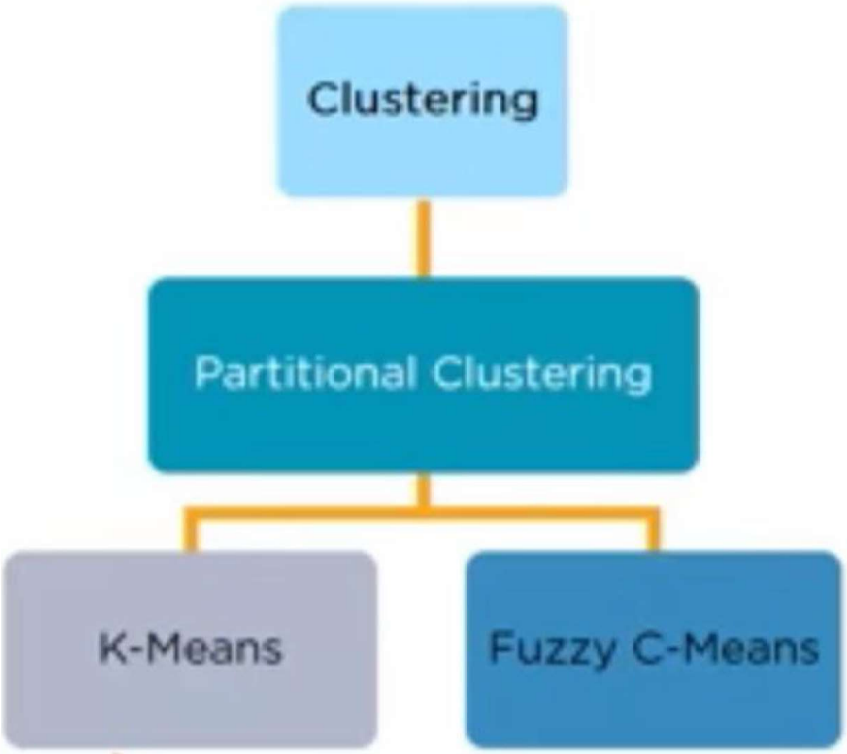
Types of Clustering

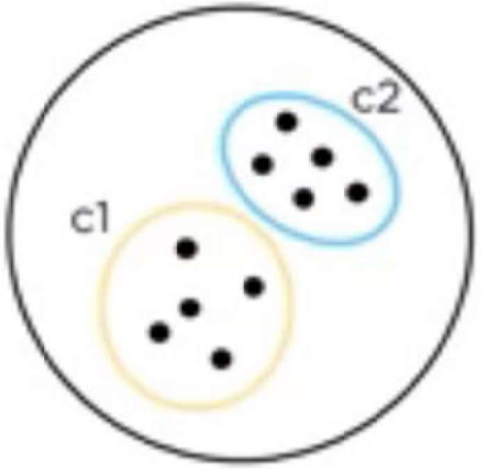




"Top down" approach begin with the whole set and proceed to divide it into successively smaller clusters.

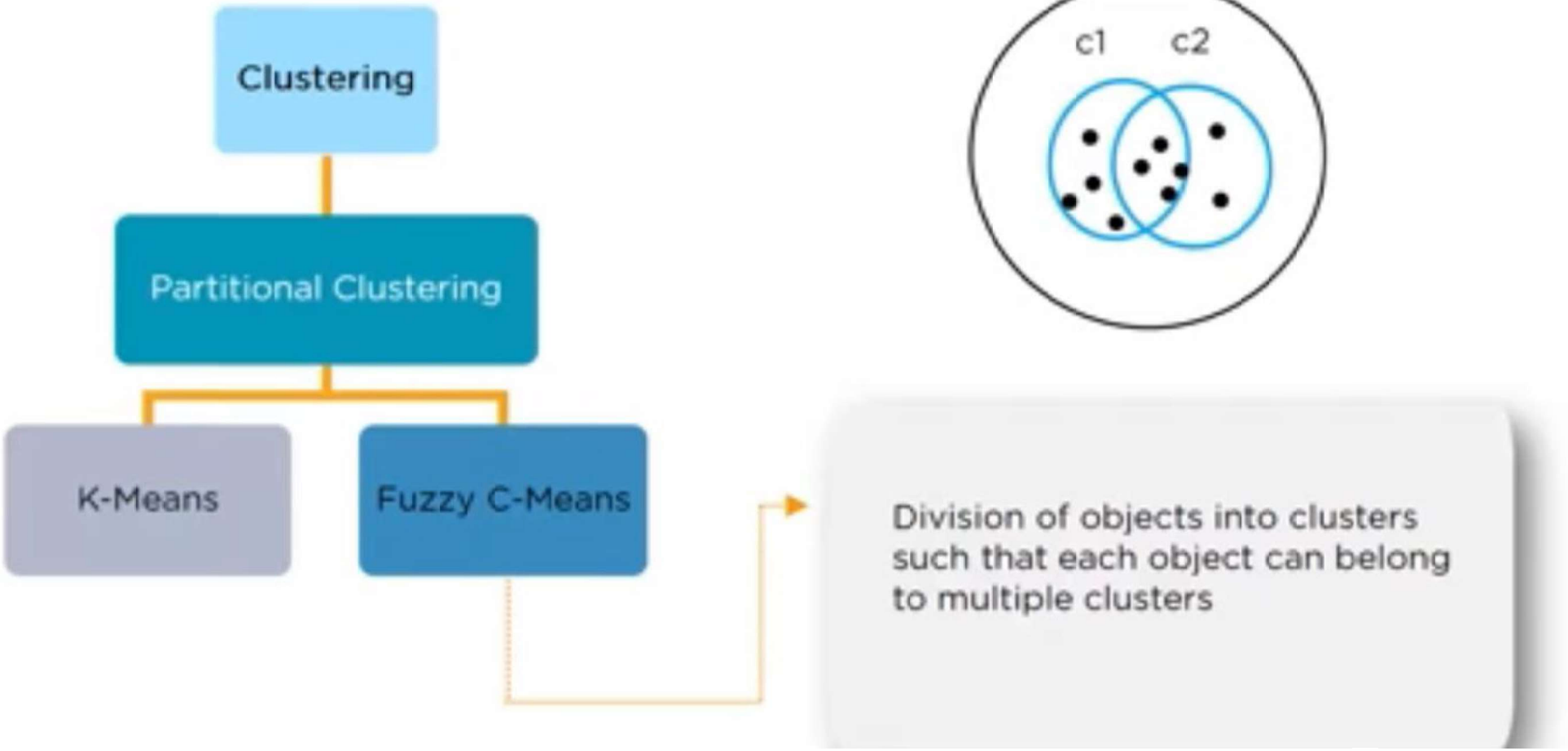
Types of Clustering



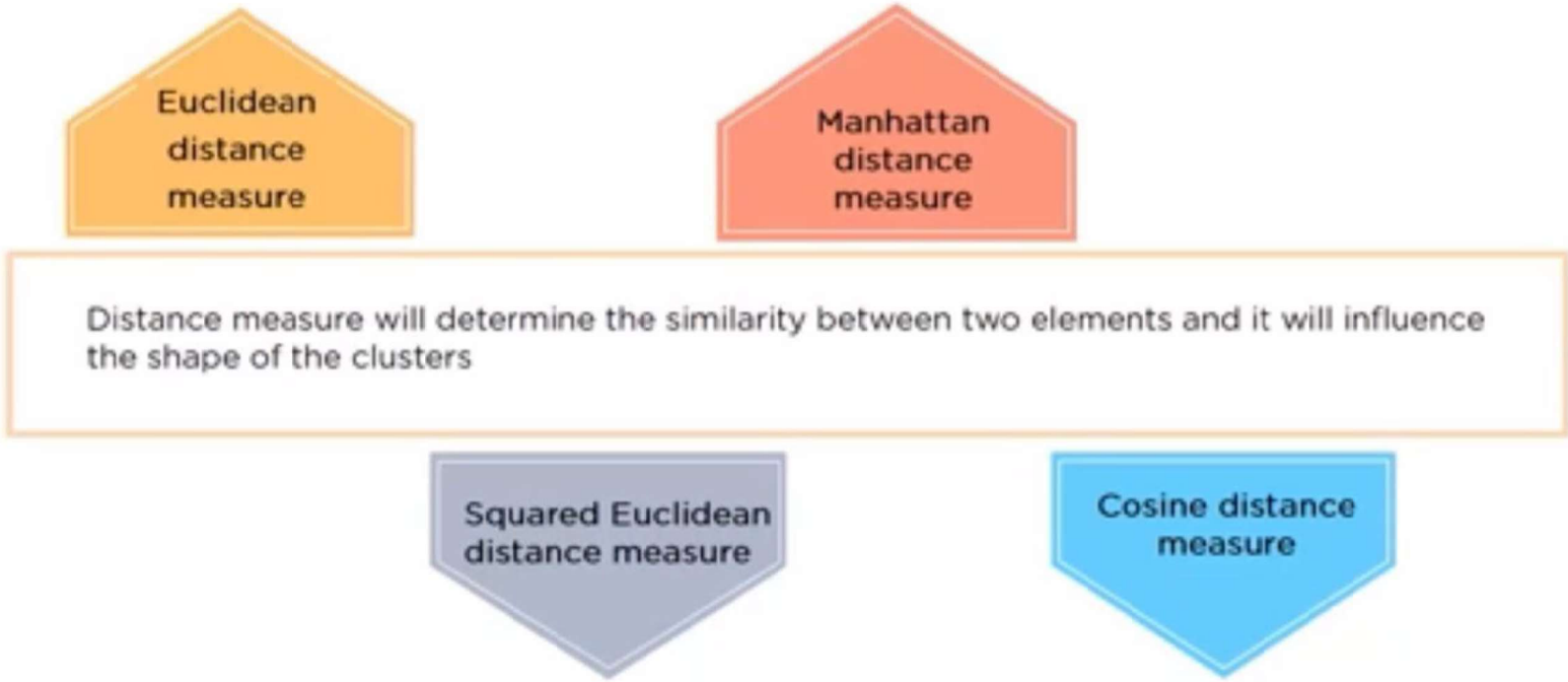


Division of objects into clusters such that each object is in exactly one cluster, not several

Types of Clustering



Distance Measure

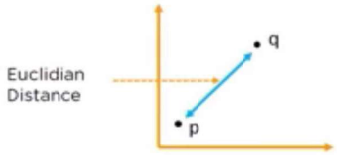


Euclidean Distance Measure

01 Euclidean distance measure
02 Squared euclidean distance measure
03 Manhattan distance measure
04 Cosine distance measure

- The Euclidean distance is the "ordinary" straight line
- It is the distance between two points in Euclidean space

$$d=\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$$



Squared Euclidean Distance Measure

01 Euclidean distance measure
02 Squared euclidean distance measure
03 Manhattan distance measure
04 Cosine distance measure

The **Euclidean squared distance** metric uses the same equation as the **Euclidean distance** metric, but does not take the square root.

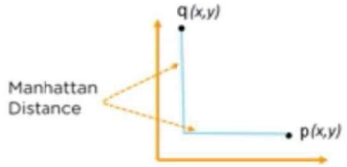
$$d=\sum_{i=1}^{n}(q_i-p_i)^2$$

Manhattan Distance Measure

01 Euclidean distance measure
02 Squared euclidean distance measure
03 Manhattan distance measure
04 Cosine distance measure

The Manhattan distance is the simple sum of the horizontal and vertical components or the distance between two points measured along axes at right angles

$$d=\sum_{i=1}^{n}|q_x-p_x|+|q_y-p_y|$$



Cosine Distance Measure

01 Euclidean distance measure
02 Squared euclidean distance measure
03 Manhattan distance measure
04 Cosine distance measure

The cosine distance similarity measures the angle between the two vectors

$$d=\frac{\sum_{i=0}^{n-1}q_i-p_x}{\sum_{i=0}^{n-1}(q_i)^2\times\sum_{i=0}^{n-1}(p_i)^2}$$