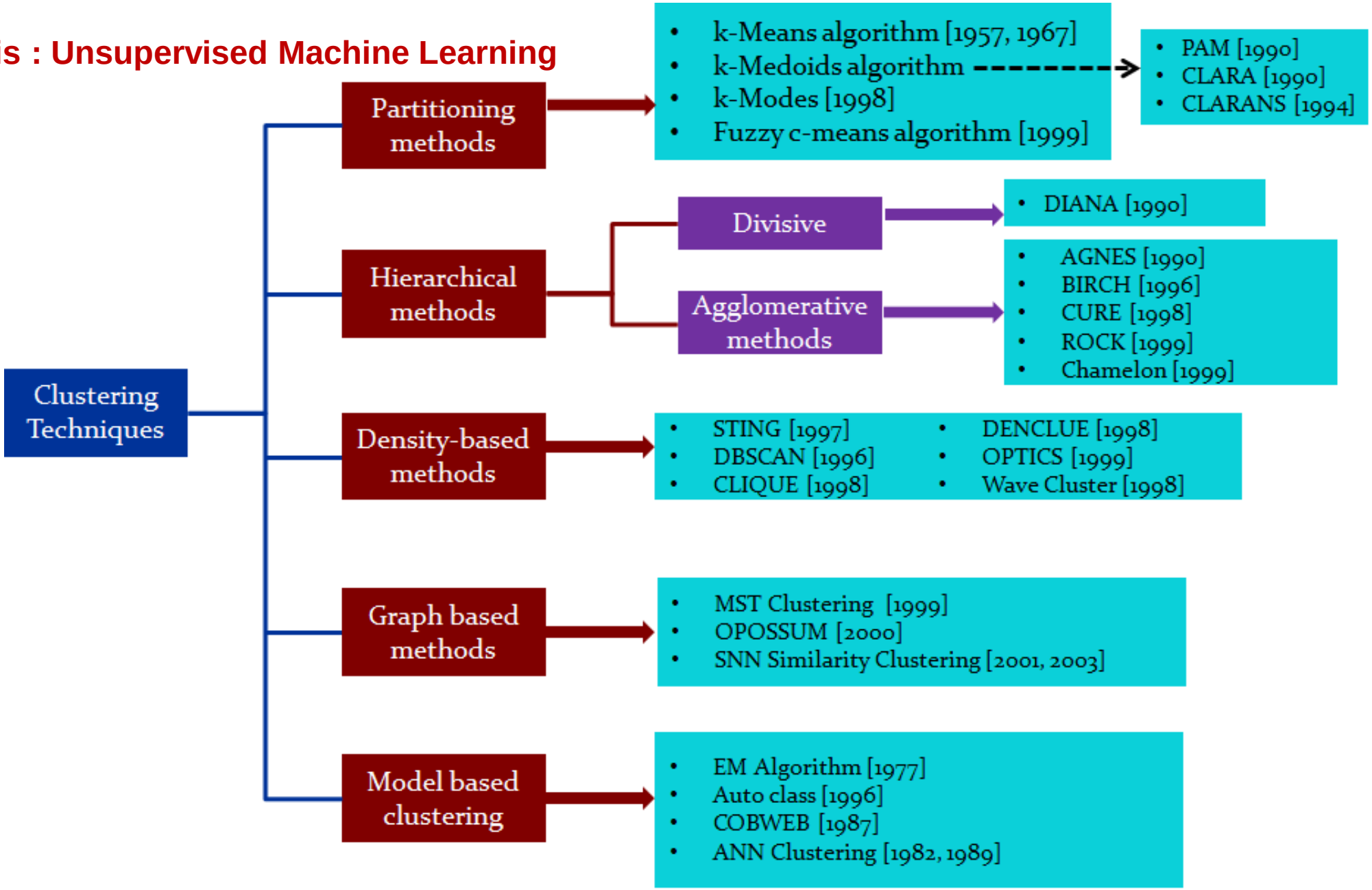
Multivariate Analysis : Unsupervised Machine Learning



Non – Hierarchical : Partitioning Clustering

- K-Means (Iterative)
- K-Mediods (PAM- Partition Around Mediods)
- CLARA – Clustering Large Applications

Hierarchical Clustering

- Agglomerative

Comparing Dendograms
Cluster Validation
Optimal Number of Clusters

What is clustering analysis?

Clustering analysis is a form of exploratory data analysis in which observations are divided into different groups that share common characteristics.

The purpose of cluster analysis is to construct groups (clusters)

while ensuring the following property:

- within a group the observations must be as similar as possible,
- while observations belonging to different groups must be as different as possible.

There are two main types of classification:

1. k-means clustering

2. Hierarchical clustering

The first is generally used when the number of clusters is fixed in advance, while the second is generally used for an unknown number of groups and helps to determine this optimal number.

Given a set of n distinct objects, the k-Means clustering algorithm partitions the objects into k number of clusters such that intracluster similarity is high but the intercluster similarity is low.

In this algorithm, user has to specify k, the number of clusters and consider the objects are defined with numeric attributes and thus using any one of the distance metric to demarcate the clusters.

K-Means Clustering Algorithm-

Step-01: Choose the number of clusters K.

Step-02: Randomly select any K data points as cluster centers.

Select cluster centers in such a way that they are as farther as possible from each other.

Step-03: Calculate the distance between each data point and each cluster center.

The distance may be calculated either by using given distance function or by using euclidean distance formula.

Step-04: Assign each data point to some cluster. A data point is assigned to that cluster whose center is nearest to that data point.

Step-05:Re-compute the center of newly formed clusters.

The center of a cluster is computed by taking mean of all the data points contained in that cluster.

Step-06:Keep repeating the procedure from Step-03 to Step-05 until any of the following stopping criteria is met- Center of newly formed clusters do not change

Data points remain present in the same cluster

Maximum number of iterations are reached

Advantages-

K-Means Clustering Algorithm offers the following advantages-

Point-01: It is relatively efficient with time complexity O(nkt) where-

n = number of instances
k = number of clusters
t = number of iterations

Point-02: It often terminates at local optimum.

Techniques such as Simulated Annealing or Genetic Algorithms may be used to find the global optimum.

Disadvantages-

K-Means Clustering Algorithm has the following disadvantages-

It requires to specify the number of clusters (k) in advance.
It can not handle noisy data and outliers.
It is not suitable to identify clusters with non-convex shapes.

Problem-01: 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)

Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7.

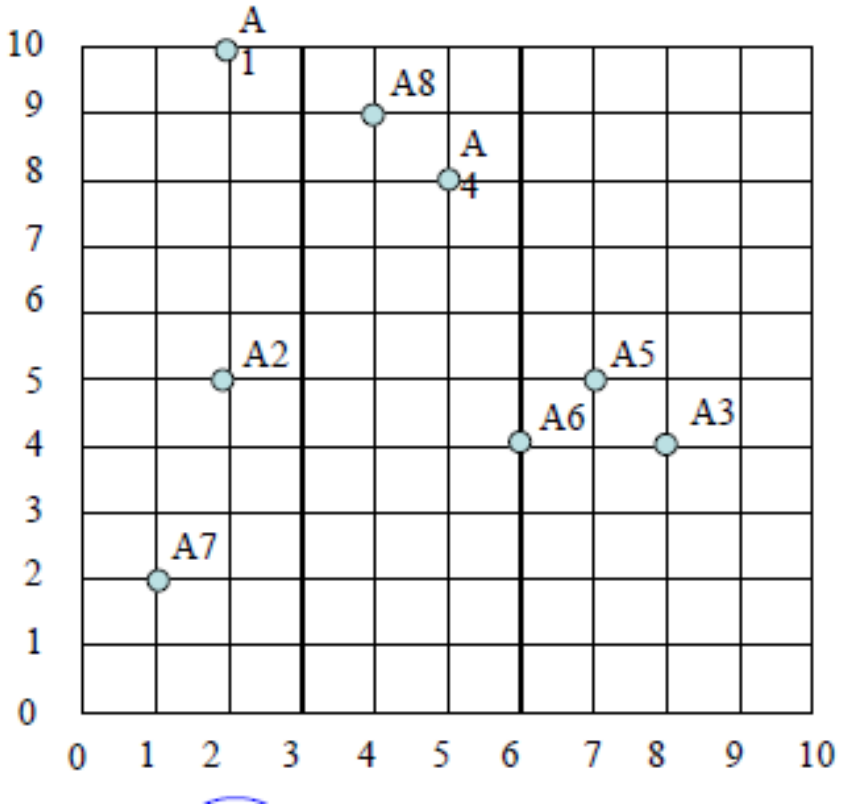
Run the k-means algorithm for 1 epoch only. At the end of this epoch show:

a) The new clusters (i.e. the examples belonging to each cluster)

b) The centers of the new clusters

The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as-

P(a, b) = |x2 – x1| + |y2 – y1| or **$d(a,b)=\sqrt{((x_b-x_a)^2+(y_b-y_a)^2)}$**



Iteration-01:

We calculate the distance of each point from each of the center of the three clusters.
The distance is calculated by using the given distance function.

The following illustration shows the calculation of distance between point A1(2, 10) and each of the center of the three clusters-

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1)
= |x2 – x1| + |y2 – y1|
= |2 – 2| + |10 – 10|
= 0

Calculating Distance Between A1(2, 10) and C2(5, 8)-

P(A1, C2)
= |x2 – x1| + |y2 – y1|
= |5 – 2| + |8 – 10|
= 3 + 2
= 5

Calculating Distance Between A1(2, 10) and C3(1, 2)-

P(A1, C3)
= |x2 – x1| + |y2 – y1|
= |1 – 2| + |2 – 10|
= 1 + 8
= 9

$P(a, b) = |x2 – x1| + |y2 – y1|$ or $d(a,b)=\sqrt{((x_b-x_a)^2+(y_b-y_a)^2)}$

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

From here, New clusters are-

Cluster-01: First cluster contains points- A1(2, 10)

Cluster-02: Second cluster contains points-
A3(8, 4)
A4(5, 8)
A5(7, 5)
A6(6, 4)
A8(4, 9)

Cluster-03: Third cluster contains points-
A2(2, 5)
A7(1, 2)

Now,
We re-compute the new cluster clusters.
The new cluster center is computed by taking mean of all the points contained in that cluster.

For Cluster-01:

We have only one point A1(2, 10) in Cluster-01.
So, cluster center remains the same.

For Cluster-02:

Center of Cluster-02
= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)
= (6, 6)

For Cluster-03:

Center of Cluster-03
= ((2 + 1)/2, (5 + 2)/2)
= (1.5, 3.5)

This is completion of Iteration-01.

Iteration -2	
Cluster	Center
1	(2, 10)
2	(6,6)
3	(1.5, 3.5)

Iteration-02:

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1)

= |x2 – x1| + |y2 – y1|

= |2 – 2| + |10 – 10|

= 0

Calculating Distance Between A1(2, 10) and C2(6, 6)-

P(A1, C2)

= |x2 – x1| + |y2 – y1|

= |6 – 2| + |6 – 10|

= 4 + 4

= 8

Calculating Distance Between A1(2, 10) and C3(1.5, 3.5)-

P(A1, C3)

= |x2 – x1| + |y2 – y1|

= |1.5 – 2| + |3.5 – 10|

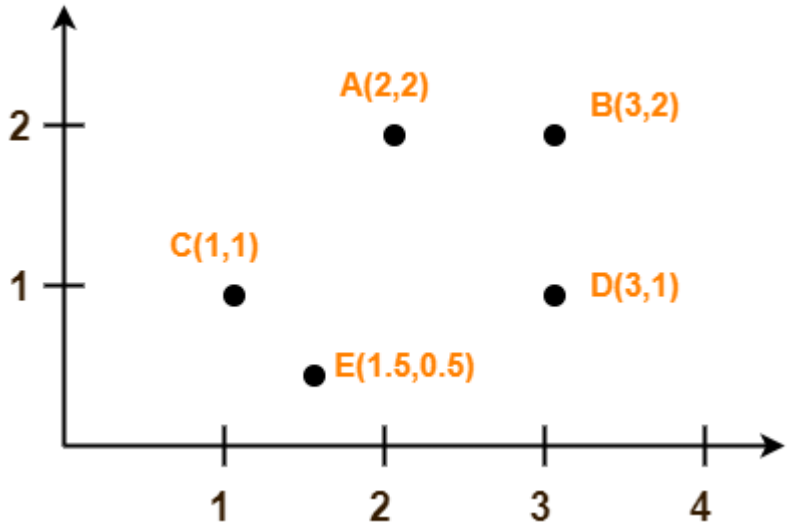
= 0.5 + 6.5

= 7

$P(a, b) = |x2 – x1| + |y2 – y1|$ or $\mathbf{d(a,b)=sqrt((x_b-x_a)^2+(y_b-y_a)^2))}$

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1



Solution-

We follow the above discussed K-Means Clustering Algorithm.
Assume A(2, 2) and C(1, 1) are centers of the two clusters.

Iteration-01:

We calculate the distance of each point from each of the center of the two clusters.
The distance is calculated by using the euclidean distance formula.

The following illustration shows the calculation of distance between point A(2, 2) and each of the center of the two clusters-

Calculating Distance Between A(2, 2) and C1(2, 2)-

P(A, C1)
$= \text{sqrt} [(x2 – x1)^2 + (y2 – y1)^2]$
$= \text{sqrt} [(2 – 2)^2 + (2 – 2)^2]$
$= \text{sqrt} [0 + 0]$
$= 0$

Calculating Distance Between A(2, 2) and C2(1, 1)-

P(A, C2)
$= \text{sqrt} [(x2 – x1)^2 + (y2 – y1)^2]$
$= \text{sqrt} [(1 – 2)^2 + (1 – 2)^2]$
$= \text{sqrt} [1 + 1]$
$= \text{sqrt} [2]$
$= 1.41$

In the similar manner, we calculate the distance of other points from each of the center of the two clusters.

Next,
We draw a table showing all the results.
Using the table, we decide which point belongs to which cluster.
The given point belongs to that cluster whose center is nearest to it.

Given Points	Distance from center (2, 2) of Cluster-01	Distance from center (1, 1) of Cluster-02	Point belongs to Cluster
A(2, 2)	0	1.41	C1
B(3, 2)	1	2.24	C1
C(1, 1)	1.41	0	C2
D(3, 1)	1.41	2	C1
E(1.5, 0.5)	1.58	0.71	C2

From here, New clusters are-

Cluster-01:

First cluster contains points-
A(2, 2)
B(3, 2)
D(3, 1)

Cluster-02:

Second cluster contains points-
C(1, 1)
E(1.5, 0.5)

Now,
We re-compute the new cluster clusters.
The new cluster center is computed by taking mean of all the points contained in that cluster.

For Cluster-01:

Center of Cluster-01
= ((2 + 3 + 3)/3, (2 + 2 + 1)/3)
= (2.67, 1.67)

For Cluster-02:

Center of Cluster-02
= ((1 + 1.5)/2, (1 + 0.5)/2)
= (1.25, 0.75)

This is completion of Iteration-01.
Next, we go to iteration-02, iteration-03 and so on until the centers do not change anymore.

Table :16 objects with two attributes A_1 and A_2.

A_1	A_2
6.8	12.6
0.8	9.8
1.2	11.6
2.8	9.6
3.8	9.9
4.4	6.5
4.8	1.1
6.0	19.9
6.2	18.5
7.6	17.4
7.8	12.2
6.6	7.7
8.2	4.5
8.4	6.9
9.0	3.4
9.6	11.1

Fig 1: Plotting data of Table 16.1

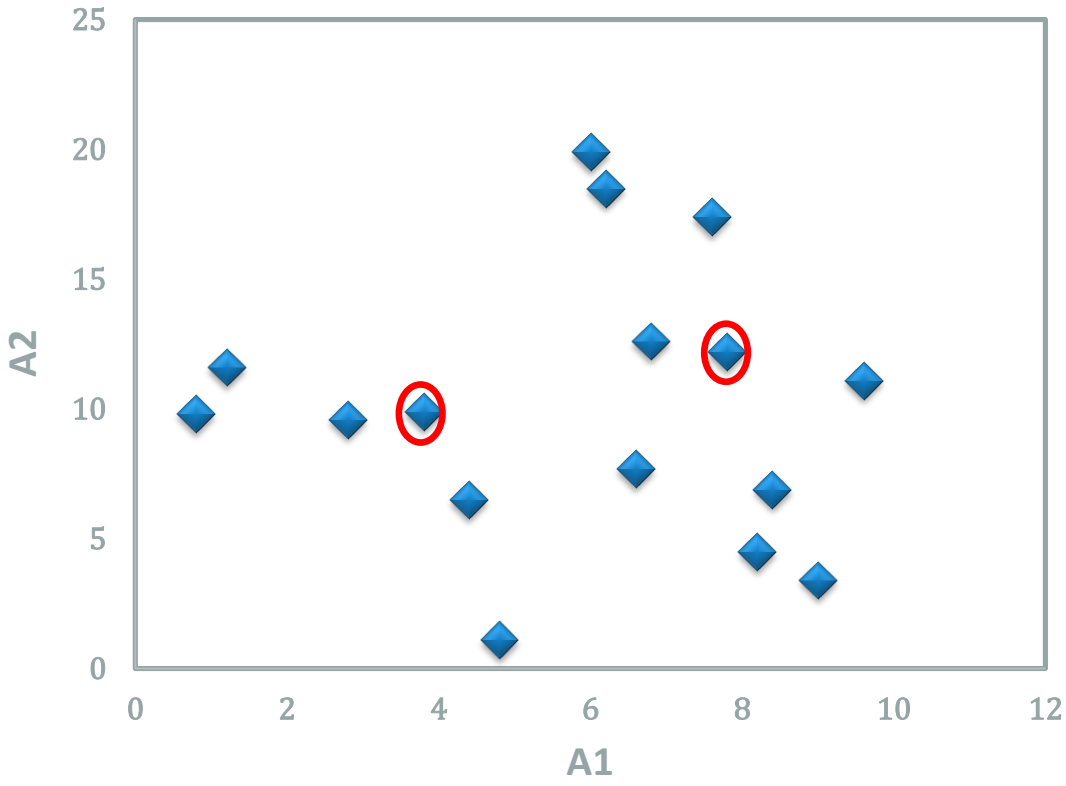


ILLUSTRATION OF K-MEANS CLUSTERING ALGORITHMS

- Suppose, k=3. Three objects are chosen at random shown as circled (see Fig 1). These three centroids are shown below.

Initial Centroids chosen randomly

Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

- Let us consider the Euclidean distance measure (L_2 Norm) as the distance measurement in our illustration.
- Let d_1, d_2 and d_3 denote the distance from an object to c_1, c_2 and c_3 respectively. The distance calculations are shown in Table 16.2.
- Assignment of each object to the respective centroid is shown in the right-most column and the clustering so obtained is shown in Fig 16.2.

ILLUSTRATION OF K-MEANS CLUSTERING ALGORITHMS

Table 16.2: Distance calculation

A_1	A_2	d_1	d_2	d_3	cluster
6.8	12.6	4.0	1.1	5.9	2
0.8	9.8	3.0	7.4	10.2	1
1.2	11.6	3.1	6.6	8.5	1
2.8	9.6	1.0	5.6	9.5	1
3.8	9.9	0.0	4.6	8.9	1
4.4	6.5	3.5	6.6	12.1	1
4.8	1.1	8.9	11.5	17.5	1
6.0	19.9	10.2	7.9	1.4	3
6.2	18.5	8.9	6.5	0.0	3
7.6	17.4	8.4	5.2	1.8	3
7.8	12.2	4.6	0.0	6.5	2
6.6	7.7	3.6	4.7	10.8	1
8.2	4.5	7.0	7.7	14.1	1
8.4	6.9	5.5	5.3	11.8	2
9.0	3.4	8.3	8.9	15.4	1
9.6	11.1	5.9	2.1	8.1	2

Fig 16.2: Initial cluster with respect to Table 16.2

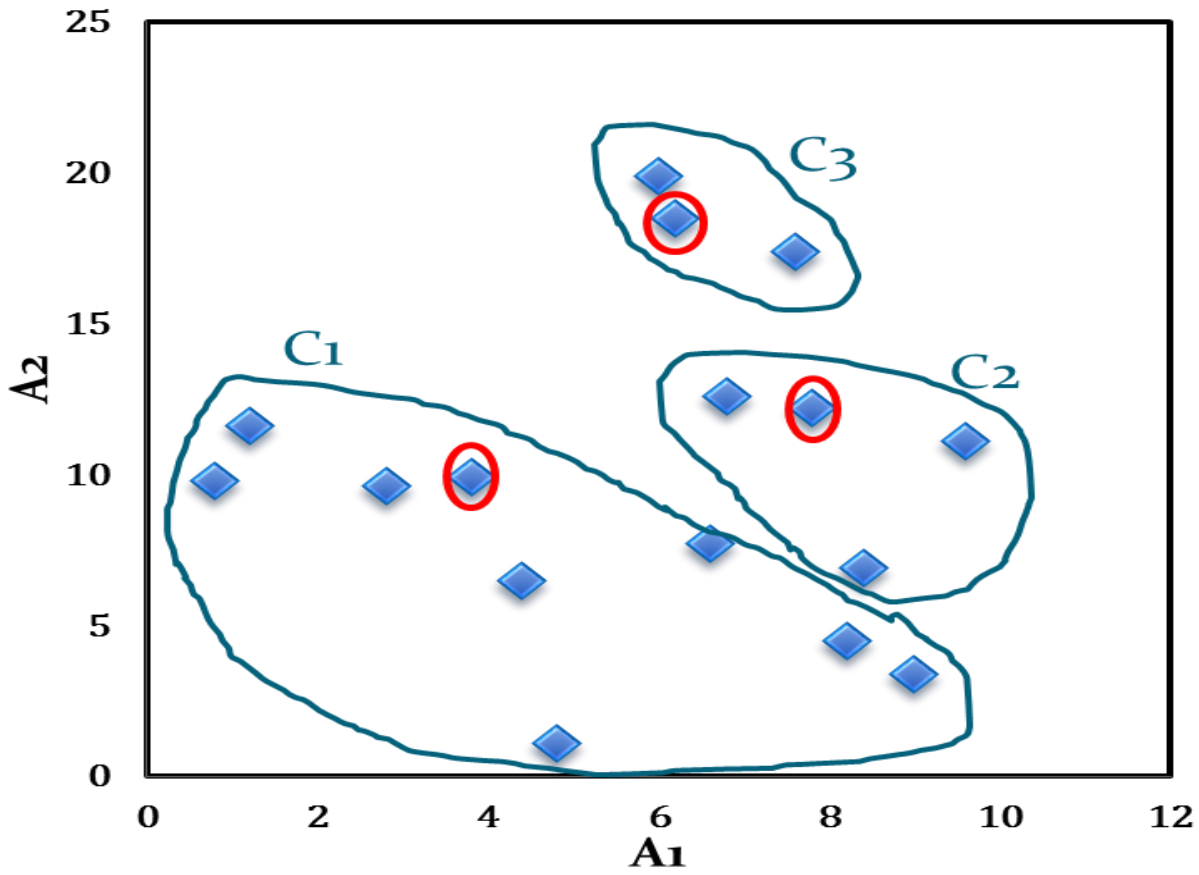


ILLUSTRATION OF K-MEANS CLUSTERING ALGORITHMS

The calculation new centroids of the three cluster using the mean of attribute values of A_1 and A_2 is shown in the Table below. The cluster with new centroids are shown in Fig 16.3.

Calculation of new centroids

New Centroid	Objects	
	A1	A2
c_1	4.6	7.1
c_2	8.2	10.7
c_3	6.6	18.6

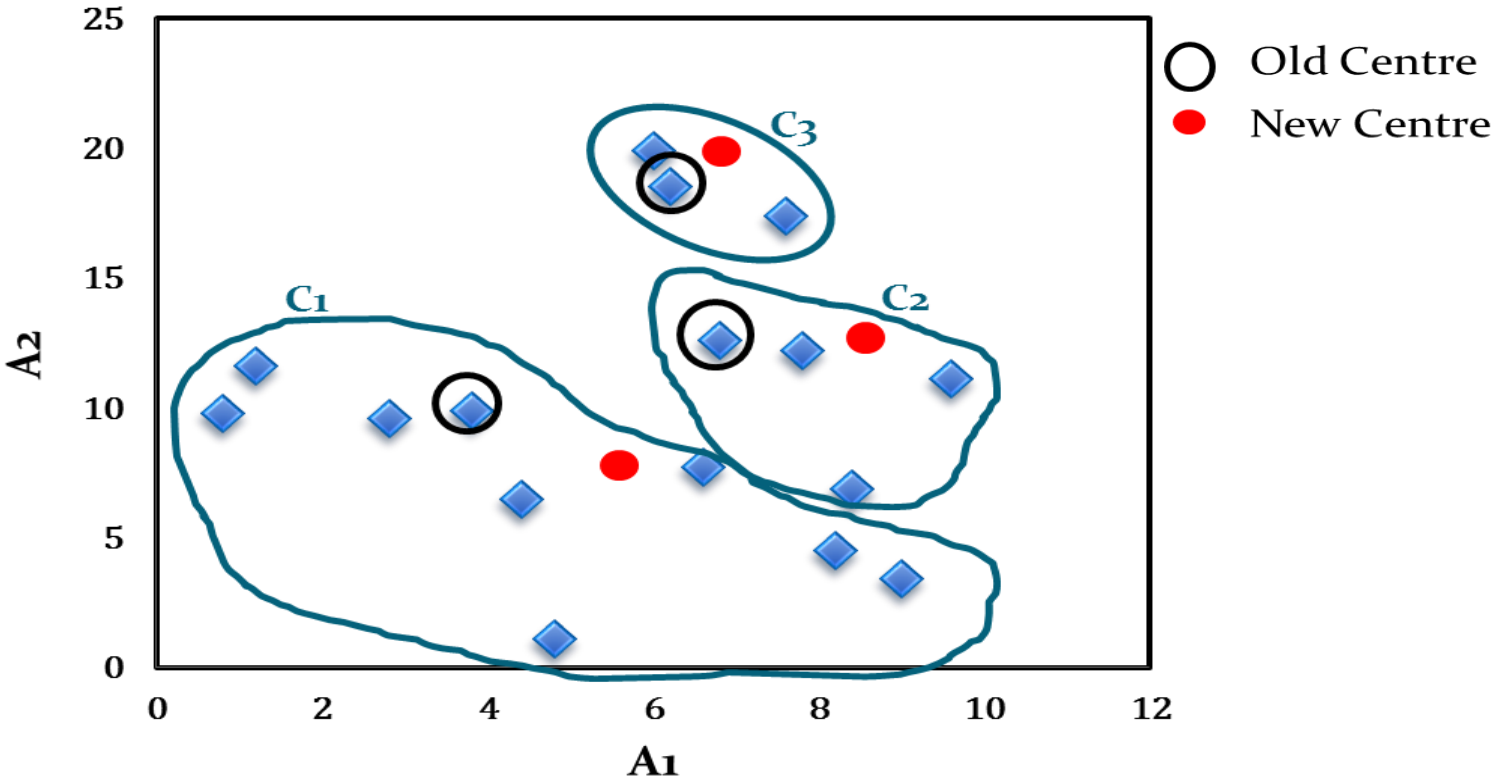


Fig 16.3: Initial cluster with new centroids

ILLUSTRATION OF K-MEANS CLUSTERING ALGORITHMS

We next reassign the 16 objects to three clusters by determining which centroid is closest to each one. This gives the revised set of clusters shown in Fig 16.4.

Note that point p moves from cluster C_2 to cluster C_1.

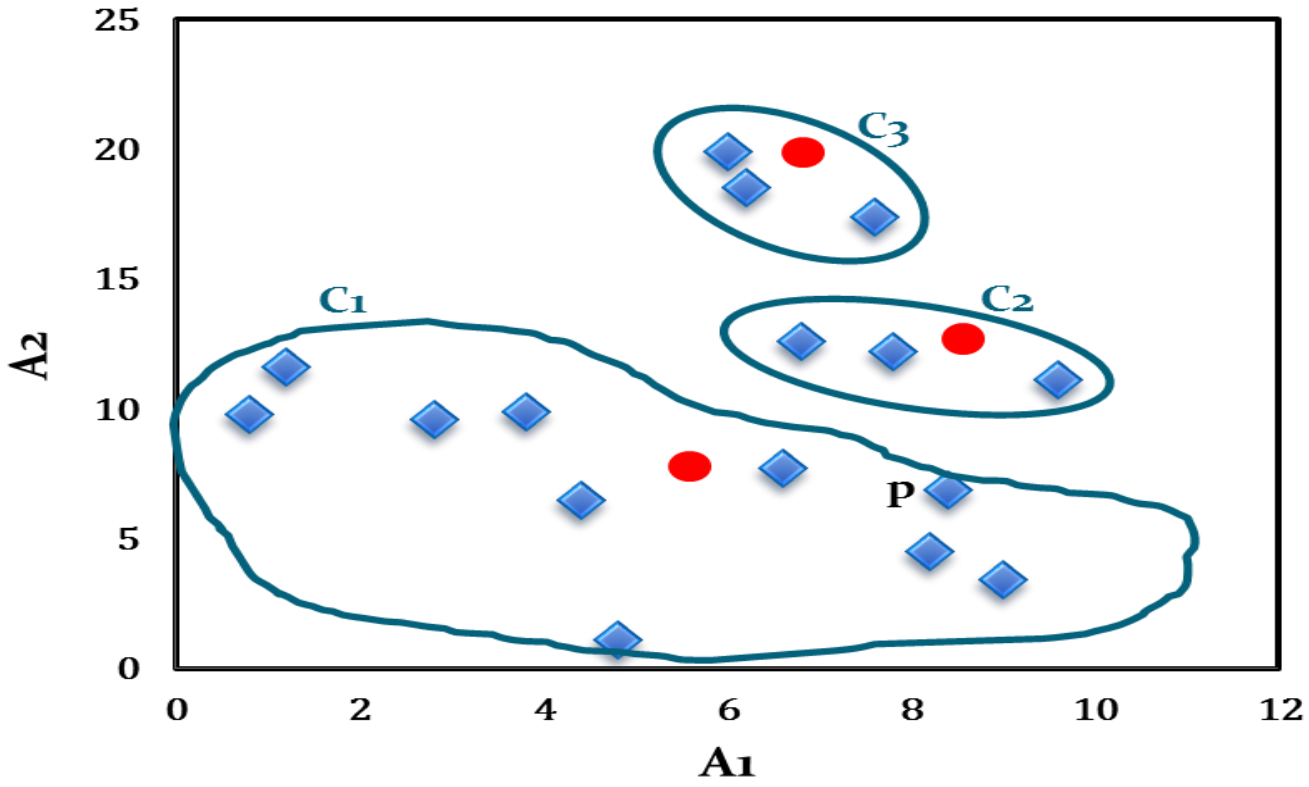


Fig 16.4: Cluster after first iteration

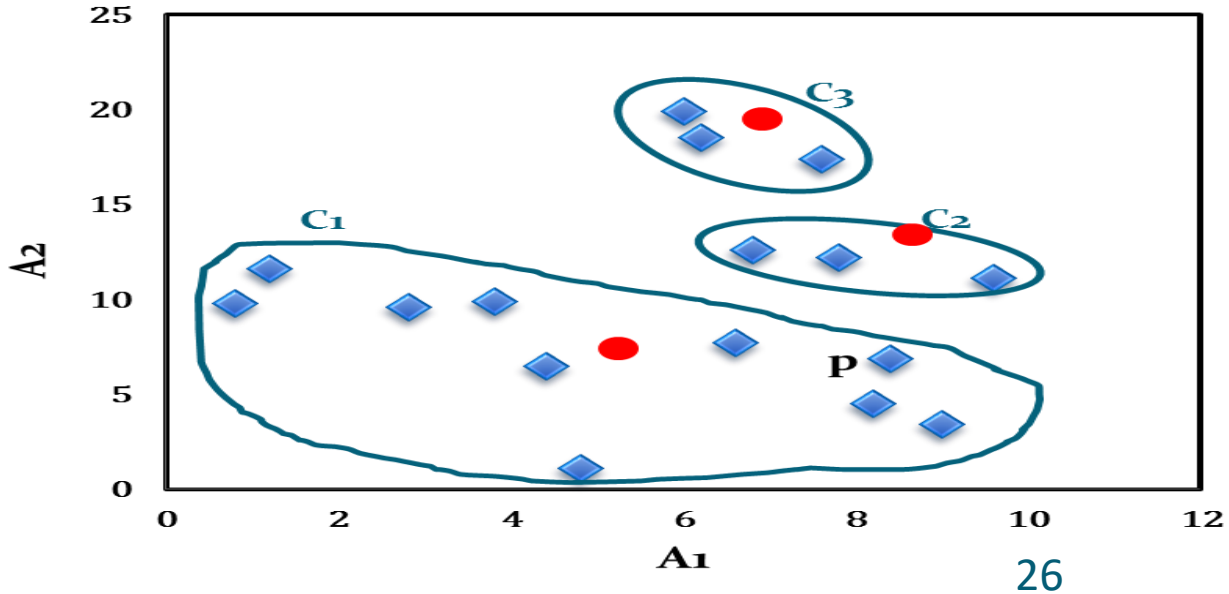
ILLUSTRATION OF K-MEANS CLUSTERING ALGORITHMS

- The newly obtained centroids after second iteration are given in the table below. Note that the centroid c_3 remains unchanged, where c_2 and c_1 changed a little.
- With respect to newly obtained cluster centres, 16 points are reassigned again. These are the same clusters as before. Hence, their centroids also remain unchanged.
- Considering this as the termination criteria, the k-means algorithm stops here. Hence, the final cluster in Fig 16.5 is same as Fig 16.4.

Cluster centres after second iteration

Centroid	Revised Centroids	
	A1	A2
c_1	5.0	7.1
c_2	8.1	12.0
c_3	6.6	18.6

Fig 16.5: Cluster after Second iteration



COMMENTS ON K-MEANS ALGORITHM

Let us analyse the k-Means algorithm and discuss the pros and cons of the algorithm.

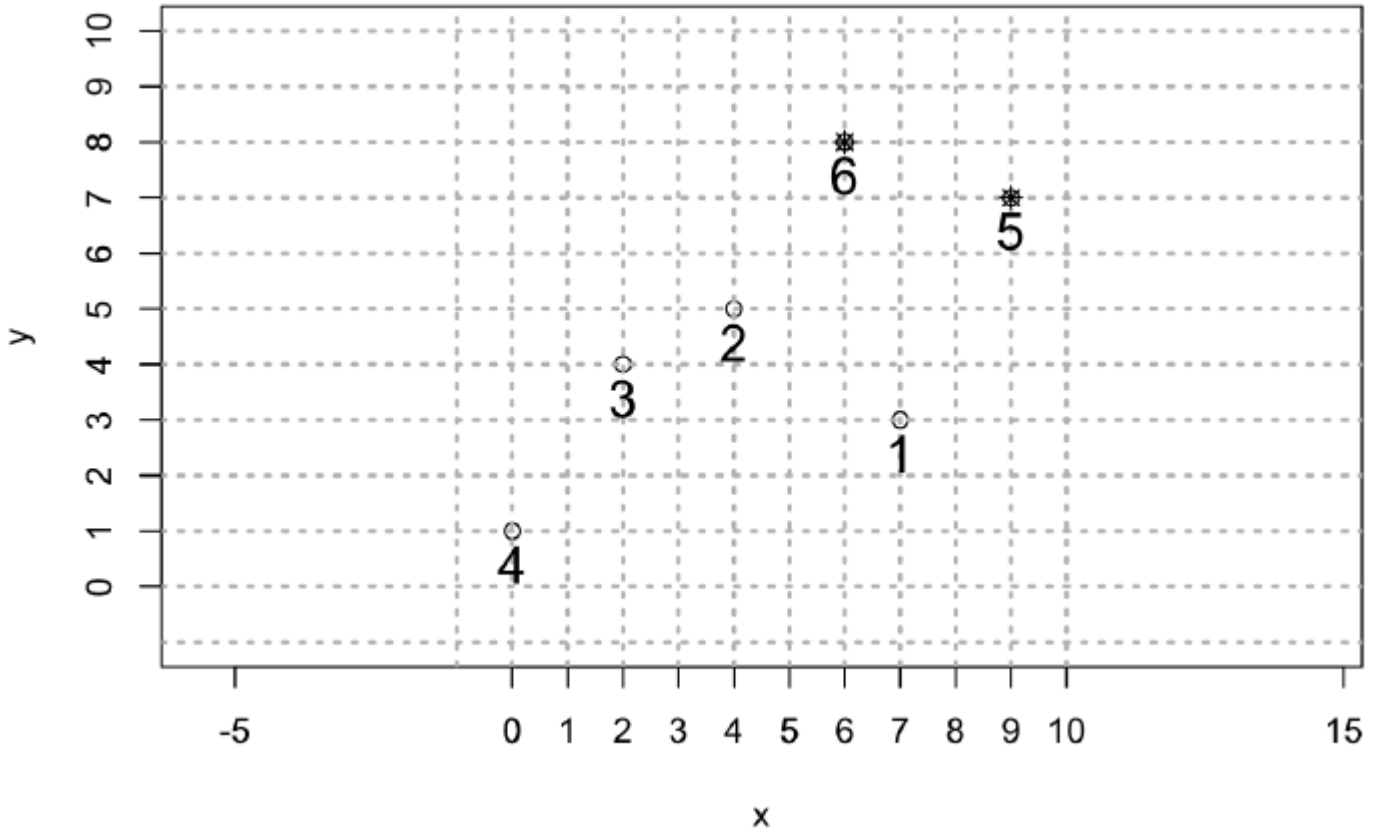
We shall refer to the following notations in our discussion.

- Notations:
  - x : an object under clustering
  - n : number of objects under clustering
  - $\boldsymbol{C}_i$: the *i-th* cluster
  - c_i : the centroid of cluster $\boldsymbol{C}_i$
  - n_i : number of objects in the cluster $\boldsymbol{C}_i$
  - c : denotes the centroid of all objects
  - k : number of clusters

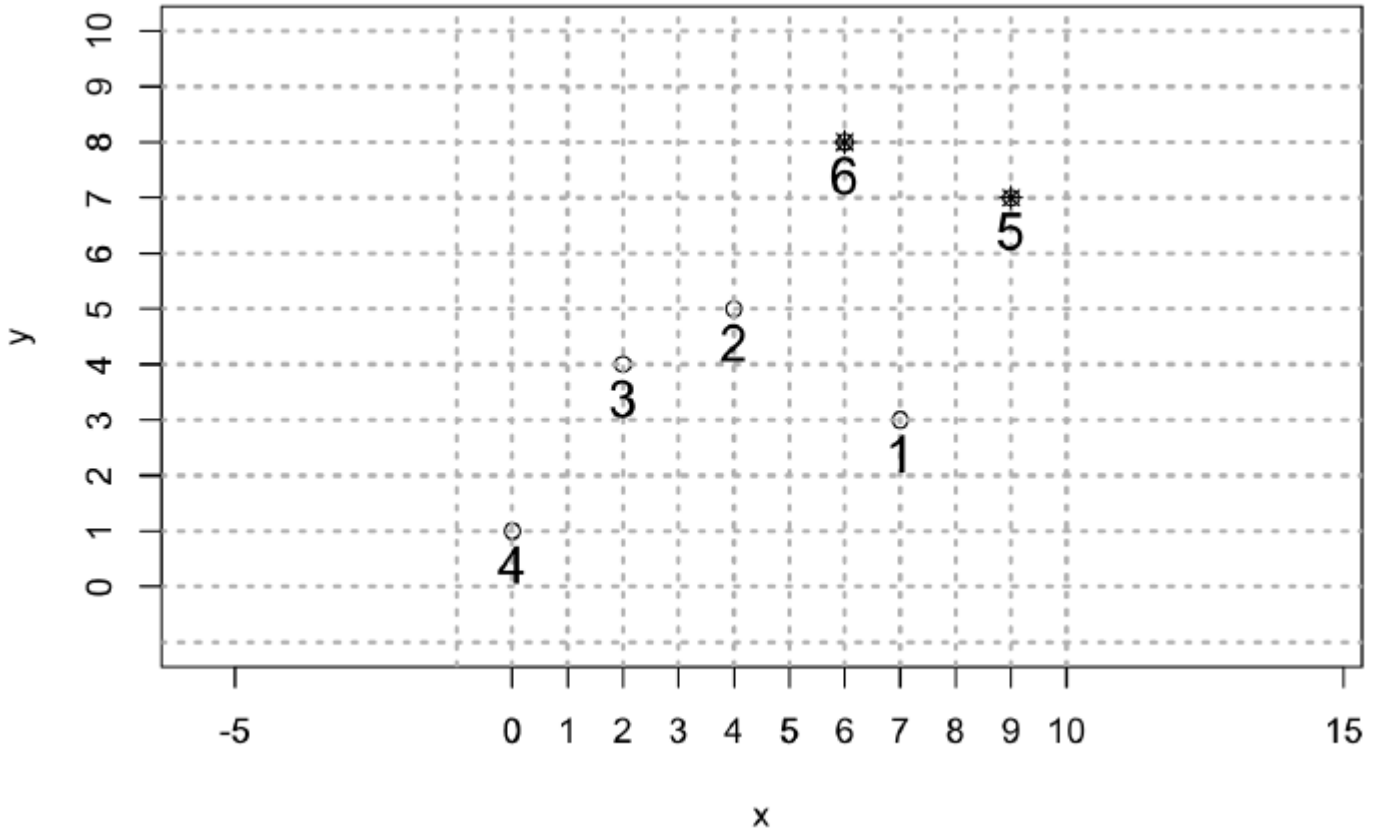
COMMENTS ON K-MEANS ALGORITHM

1. Value of k:

- The k-means algorithm produces only one set of clusters, for which, user must specify the desired number, k of clusters.
- In fact, k should be the best guess on the number of clusters present in the given data. Choosing the best value of k for a given dataset is, therefore, an issue.
- We may not have an idea about the possible number of clusters for high dimensional data, and for data that are not scatter-plotted.
- Further, possible number of clusters is hidden or ambiguous in image, audio, video and multimedia clustering applications etc.
- There is no principled way to know what the value of k ought to be. We may try with successive value of k starting with 2.
- The process is stopped when two consecutive k values produce more-or-less identical results (with respect to some cluster quality estimation).
- Normally $k \ll n$ and there is heuristic to follow $k \approx \sqrt{n}$.



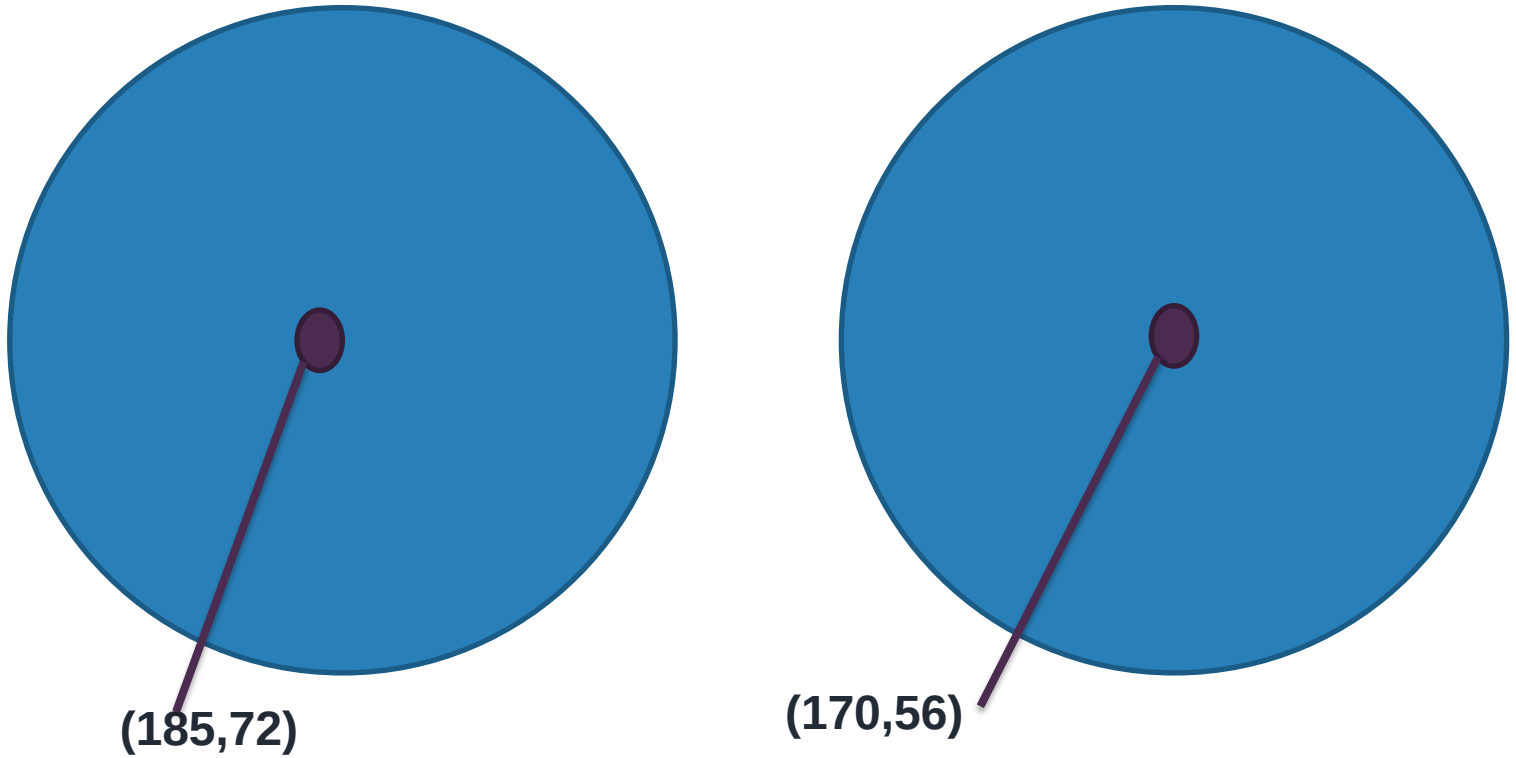
The initial clusters are
Group 1:(9,7)
Group 2:(6,8)



ID	X	Y
1	7	3
2	4	5
3	2	4
4	0	1
5	9	7
6	6	8

	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76

Divide into two clusters



Exercise 3. Hierarchical clustering

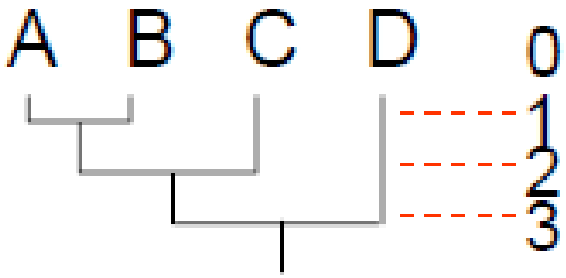
Use single and complete link agglomerative clustering to group the data described by the following distance matrix. Show the dendrograms.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Hierarchical clustering

Use single and complete link agglomerative clustering to group the data described by the following distance matrix. Show the dendrograms.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0



Solution:

Agglomerative initially every point is a cluster of its own and we merge cluster until we end-up with
one unique cluster containing all points.

a) single link: distance between two clusters is the shortest distance between a pair of elements from the
two clusters.

d k K Comments

0 4 {A}, {B}, {C}, {D} We start with each point = cluster

1 3 {A, B}, {C}, {D} Merge {A} and {B} since A & B are the closest: d(A, B)=1

2 2 {A, B, C}, {D} Merge {A, B} and {C} since B & C are the closest: d(B, C)=2

3 1 {A, B, C, D} Merge D

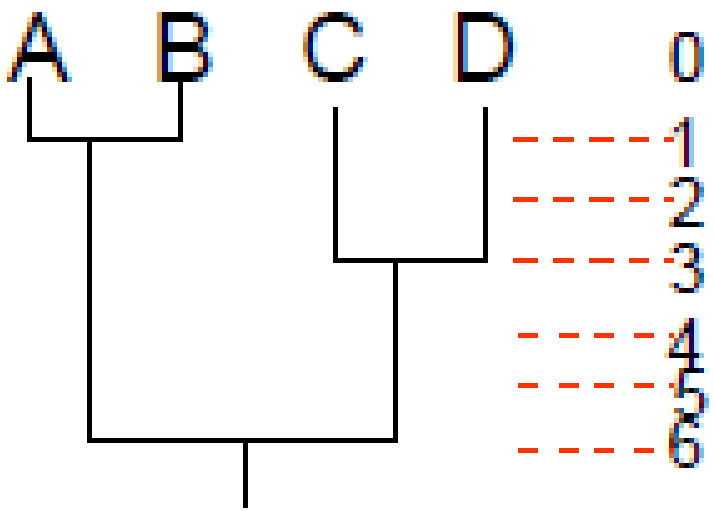
Complete linkage

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

d	k	K	Comments
0	4	{A},{B},{C},{D}	Start with each point
1	3	{A,B},{C},{D}	d(A,B) =1 , as <=1, Merge A and B
2	3	{A,B},{C},{D}	Cannot merge as distance value between points are greater than 2
3	2	{A,B},{C,D}	d(C,D) =3, as <=3, Merge C and D
4	2	{A,B},{C,D}	Cannot merge
5	2	{A,B},{C,D}	Cannot merge
6	1	{A,B,C,D}	Merge {C,D} with {A,B}

Complete Link

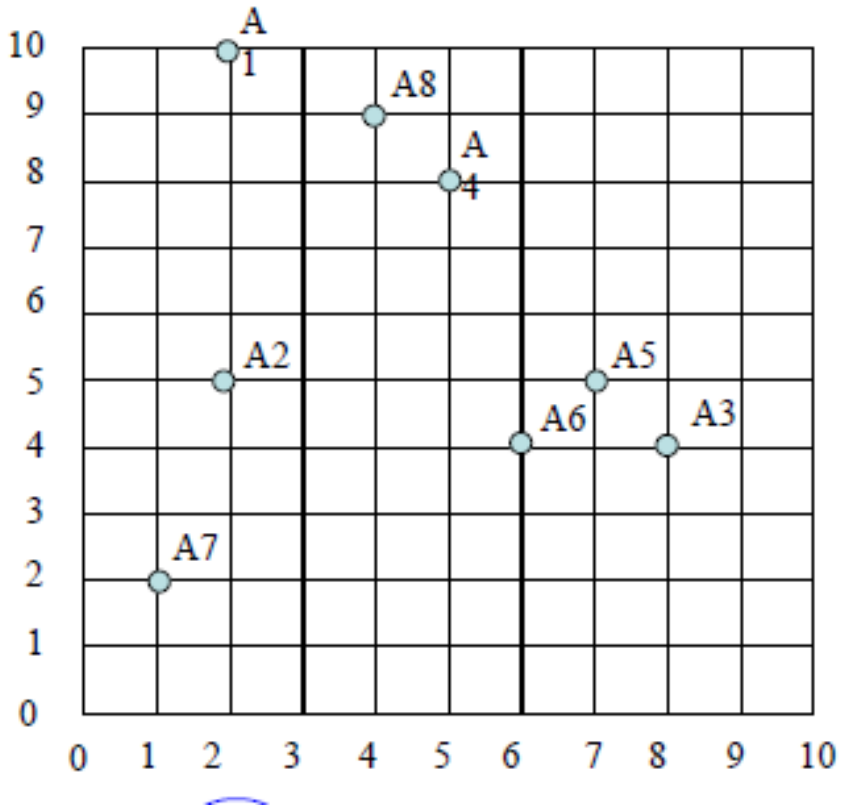
d	k	K	Comments
0	4	{A}, {B}, {C}, {D}	We start with each point = cluster
1	3	{A, B}, {C}, {D}	d(A,B)=1<=1 ➔ merge {A} and {B}
2	3	{A, B}, {C}, {D}	d(A,C)=4>2 so we can't merge C with {A,B} d(A,D)=5>2 and d(B,D)=6>2 so we can't merge D with {A, B} d(C,D)=3>2 so we can't merge C and D
3	2	{A, B}, {C, D}	- d(A,C)=4>3 so we can't merge C with {A,B} - d(A,D)=5>3 and d(B,D)=6>3 so we can't merge D with {A, B} - d(C,D)=3 <=3 so merge C and D
4	2	{A, B}, {C, D}	{C,D} cannot be merged with {A, B} as d(A,D)= 5 >4 (and also d(B,D)= 6 >4) although d(A,C)= 4 <= 4, d(B,C)= 2<=4)
5	2	{A, B}, {C, D}	{C,D} cannot be merged with {A, B} as d(B,D)= 6 > 5
6	1	{A, B, C, D}	{C, D} can be merged with {A, B} since d(B,D)= 6 <= 6, d(A,D)= 5 <= 6, d(A,C)= 4 <= 6, d(B,C)= 2 <= 6



Hierarchical clustering

Use single-link, complete-link, to cluster the following 8 examples:
A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0							
A2	5	0						
A3	12	7	0					
A4	5	6	7	0				
A5	10	5	2	5	0			
A6	10	5	2	5	2	0		
A7	9	4	9	10	9	7	0	
A8	3	6	9	2	7	7	10	0



Solution:

Single Link:

d k K

0 8 {A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}

1 8 {A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}

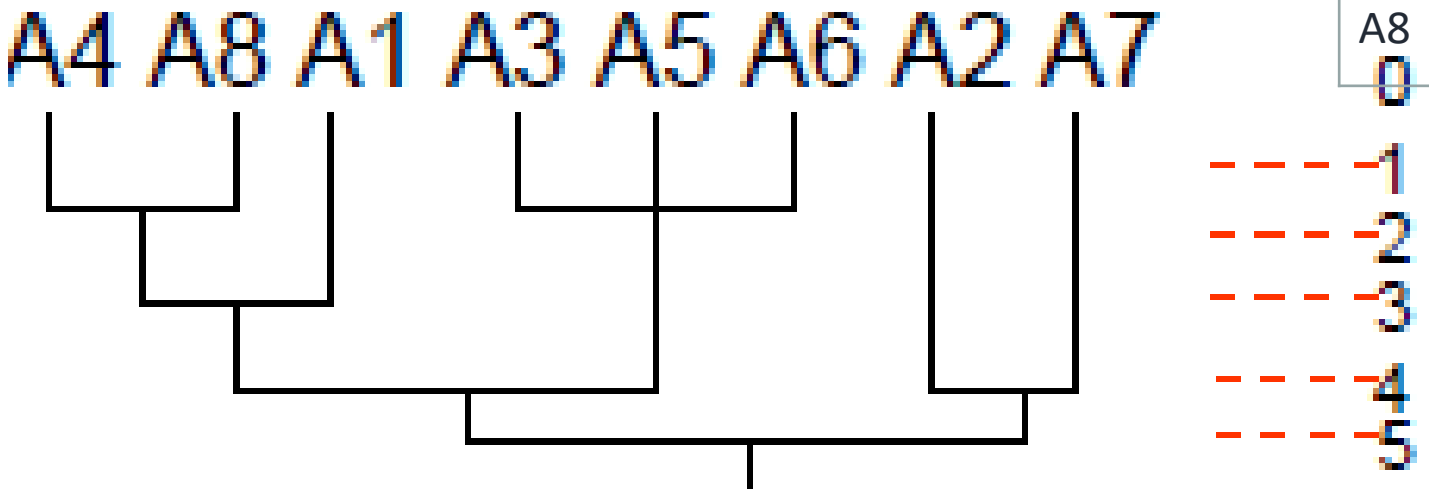
2 5 {A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}

3 4 {A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}

4 3 {A4, A8, A1}, {A3, A5, A6}, {A2, A7}

5 1 {A1, A3, A4, A5, A6, A8, A2, A7}

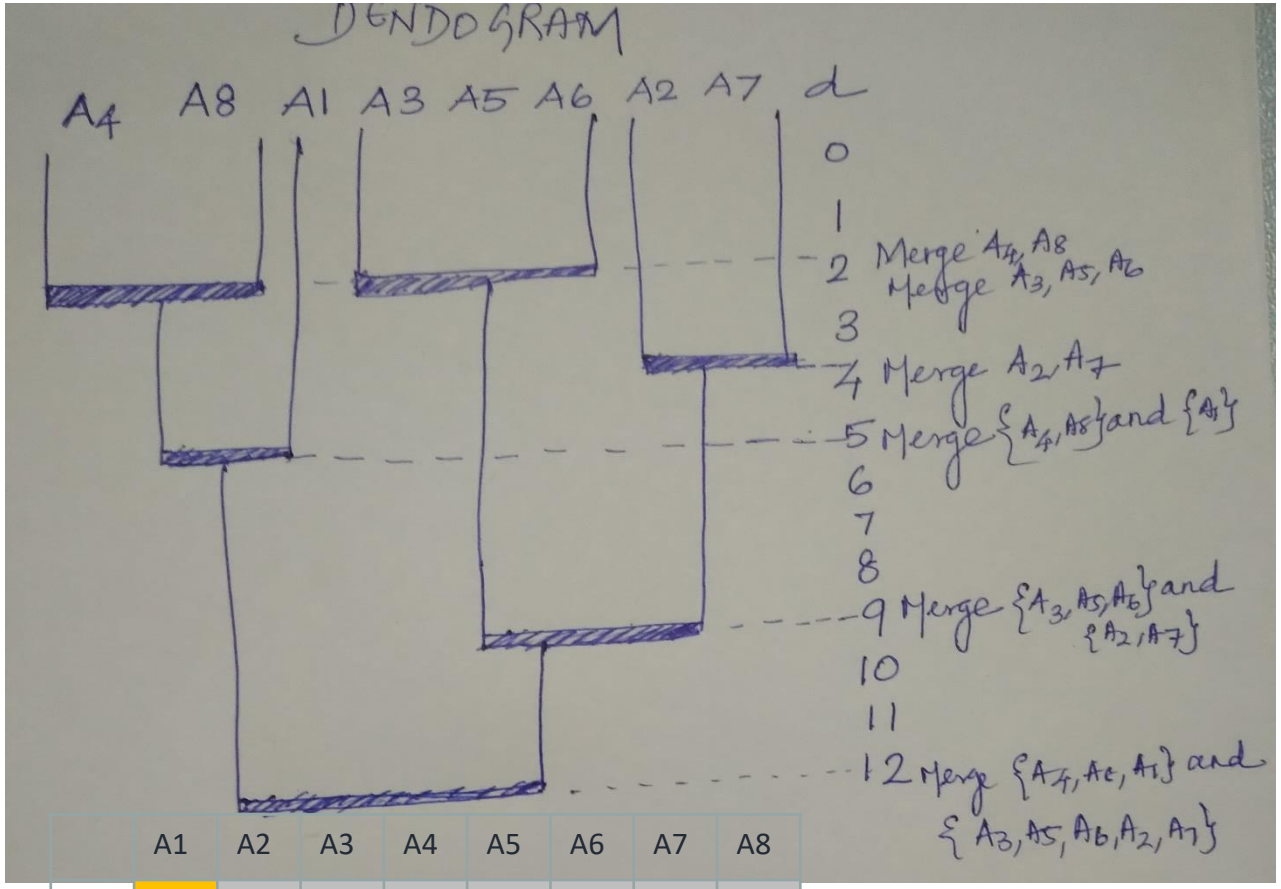
	A1	A2	A3	A4	A5	A6	A7	A8
A1	0							
A2	5	0						
A3	12	7	0					
A4	5	6	7	0				
A5	10	5	2	5	0			
A6	10	5	2	5	2	0		
A7	9	4	9	10	9	7	0	
A8	3	6	9	2	7	7	10	0



Single linkage is Nearest Neighbor Method, The dendogram is shown for Euclidean Distance measure

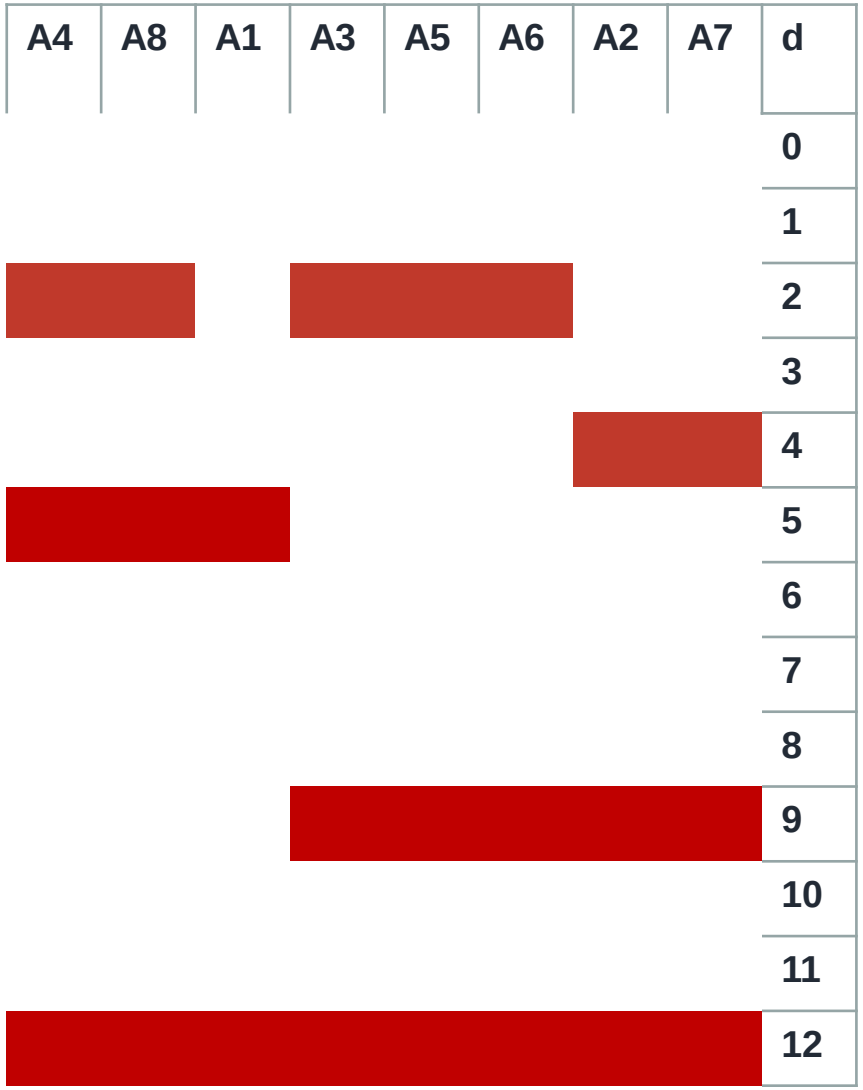
Complete Linkage is Farthest Neighbor Method

d	k	K	Comments
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}	
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}	
2	5	{A4,A8}, {A3,A5,A6}, {A1}, {A2},{A7}	
3	5	{A4,A8}, {A3,A5,A6}, {A1}, {A2},{A7}	
4	4	{A2, A7}, {A4,A8}, {A1}, {A3,A5,A6}	
5	3	{A4,A8,A1}, {A3,A5,A6},{A2,A7}	
6	3	{A4,A8,A1}, {A3,A5,A6},{A2,A7}	
7	3	{A4,A8,A1}, {A3,A5,A6},{A2,A7}	
8	3	{A4,A8,A1}, {A3,A5,A6},{A2,A7}	
9	2	{A2,A3,,A5,A6,A7}, {A4,A8,A1}	
10	2	{A2,A3,,A5,A6,A7}, {A4,A8,A1}	
11	2	{A2,A3,,A5,A6,A7}, {A4,A8,A1}	
12	1	{A1,A2,A3,A4,A5,A6,7,A8}	



	A1	A2	A3	A4	A5	A6	A7	A8
A1	0							
A2	5	0						
A3	12	7	0					
A4	5	6	7	0				
A5	10	5	2	5	0			
A6	10	5	2	5	2	0		
A7	9	4	9	10	9	7	0	
A8	3	6	9	2	7	7	10	0

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0							
A2	5	0						
A3	12	7	0					
A4	5	6	7	0				
A5	10	5	2	5	0			
A6	10	5	2	5	2	0		
A7	9	4	9	10	9	7	0	
A8	3	6	9	2	7	7	10	0



Complete Link

d k K

0 8 {A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}

1 8 {A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}

2 5 {A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}

3 5 {A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}

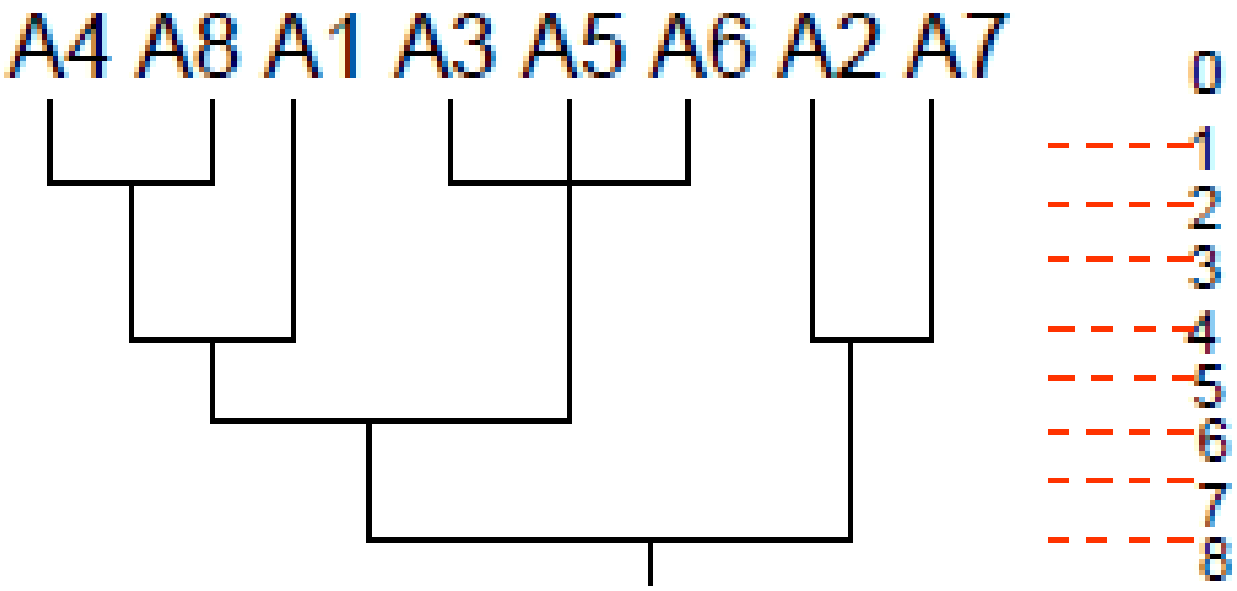
4 3 {A4, A8, A1}, {A3, A5, A6}, {A2, A7}

5 3 {A4, A8, A1}, {A3, A5, A6}, {A2, A7}

6 2 {A4, A8, A1, A3, A5, A6}, {A2, A7}

7 2 {A4, A8, A1, A3, A5, A6}, {A2, A7}

8 1 {A4, A8, A1, A3, A5, A6, A2, A7}



Complete linkage is Farthest Neighbor Method, The dendogram is shown for Euclidean Distance measure

2. Proximity Measure for Binary Attributes

- A contingency table for binary data

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

- Distance measure for symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

- Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables)

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

	1	2	3	4
1	0			
2	d(2,1)	0		
3	d(3,1)	d(3,2)	0	
4	d(4,1)	d(4,2)	d(4,3)	0

	1	2	3	4
1	0			
2	0	0		
3	1	1	0	
4	1	1	0	0

Object *i*

CustomerID	Gender	Age
1	Male	19
2	Male	21
3	Female	20
4	Female	23

Object *j*

	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

d(2,1)	ID2			
ID1		1(M)	0(F)	sum_{row}
	1 (M)	1	0	1
	0 (F)	0	0	0
	sum_{col}	1	0	1

d(3,1)	ID3			
ID1		1(M)	0(F)	sum_{row}
	1 (M)	0	1	
	0 (F)	0	0	
	sum_{col}			

d(4,1)	ID4			
ID1		1(M)	0(F)	sum_{row}
	1 (M)	0	1	
	0 (F)	0	0	
	sum_{col}			

	1	2	3	4
1	0			
2	d(2,1)	0		
3	d(3,1)	d(3,2)	0	
4	d(4,1)	d(4,2)	d(4,3)	0

	1	2	3	4
1	0			
2	0.5	0		
3	0.25	0.25	0	
4	1	0.5	0.75	

Object *i*

CustomerID	Gender	Age
1	Male	19
2	Male	21
3	Female	20
4	Female	23

Object *j*

	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

Proximity measure for Symmetric Binary

	1	2	3	4
1	0			
2	0	0		
3	1	1	0	
4	1	1	0	0

Proximity measure for Mixed Type

	1	2	3	4
1	0			
2	0.25	0		
3	0.625	0.625	0	
4	1	0.75	0.375	0

Proximity measure for Numeric

	1	2	3	4
1	0			
2	0.5	0		
3	0.25	0.25	0	
4	1	0.5	0.75	

CustomerID	Gender	Age
1	Male	19
2	Male	21
3	Female	20
4	Female	23

HOW TO FIND OPTIMAL NUMBER OF CLUSTERS ?

1. Elbow Method
2. Silhoutte Coefficient
3. Gap Statistics (2001)

Elbow Method

The optimal number of clusters can be defined as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.

2. For each k, calculate the total within-cluster sum of square (wss).

3. Plot the curve of wss according to the number of clusters k.

4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

- **Within Cluster Sums of Squares :** $$\mathrm{WSS} = \sum_{i=1} \sum_{x \in C_i} d(\mathbf{x}, \bar{\mathbf{x}}_{\mathbf{C_i}})^2$$
- **Between Cluster Sums of Squares:** $$\mathrm{BSS} = \sum_{i=1}^{N_C} |C_i| \cdot d(\bar{\mathbf{x}}_{\mathbf{C_i}}, \bar{\mathbf{x}})^2$$

C_i = Cluster, N_c = # clusters, $\bar{x}_{c_i}$ = Cluster centroid, $\bar{x}$ = Sample Mean

The silhouette analysis measures how well an observation is clustered and it estimates the **average distance between clusters**. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

For each observation i, the silhouette width s_i is calculated as follows:

1. For each observation i, calculate the average dissimilarity a_i between i and all other points of the cluster to which i belongs.

2. For all other clusters C, to which i does not belong, calculate the average dissimilarity $d(i, C)$ of i to all observations of C. The smallest of these $d(i, C)$ is defined as $b_i = \min_C d(i, C)$. The value of b_i can be seen as the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does not belong.

3. Finally the silhouette width of the observation i is defined by the formula: $S_i = (b_i - a_i)/max(a_i, b_i)$.

Silhouette width can be interpreted as follow:

Interpretation:

- Observations with a large S_i (almost 1) are very well clustered.

- A small S_i (around 0) means that the observation lies between two clusters.

- Observations with a negative S_i are probably placed in the wrong cluster.

Gap Statistics

The gap statistic has been published by R. Tibshirani, G. Walther, and T. Hastie (Standford University, 2001).

- The approach can be applied to any clustering method.
- The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data.
- The estimate of the optimal clusters will be value that maximize the gap statistic (i.e, that yields the largest gap statistic).
- This means that the clustering structure is far away from the random uniform distribution of points.

Gap Statistics algorithm:

1. Cluster the observed data, varying the number of clusters from k = 1, ..., k_{max}, and compute the corresponding total within intra-cluster variation W_k.

2. Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, ..., k_{max}, and compute the corresponding total within intra-cluster variation W_{kb}.

3. Compute the estimated gap statistic as the deviation of the observed W_k value from its expected value W_{kb} under the null hypothesis: $Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W_{kb}^{*}) - log(W_k)$. Compute also the standard deviation of the statistics.

4. Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1: $Gap(k) \geq Gap(k+1) - s_{k+1}$.