Matters of Discussion

Non-hierarchical Clustering

K-means Algorithm

K-Medoids

Compiled By: Dr. Nilamadhab Mishra [(PhD- CSIE) Taiwan]

K-means Algorithm

- K-means algorithm is an iterative algorithm
  - that tries to partition the dataset into **K** pre-defined distinct non-overlapping subgroups (clusters)
  - where each data point belongs to only one group.
- The algorithm is used when you have unlabeled data.
- The goal is to find certain groups based on some kind of similarity in the data with K number of groups.

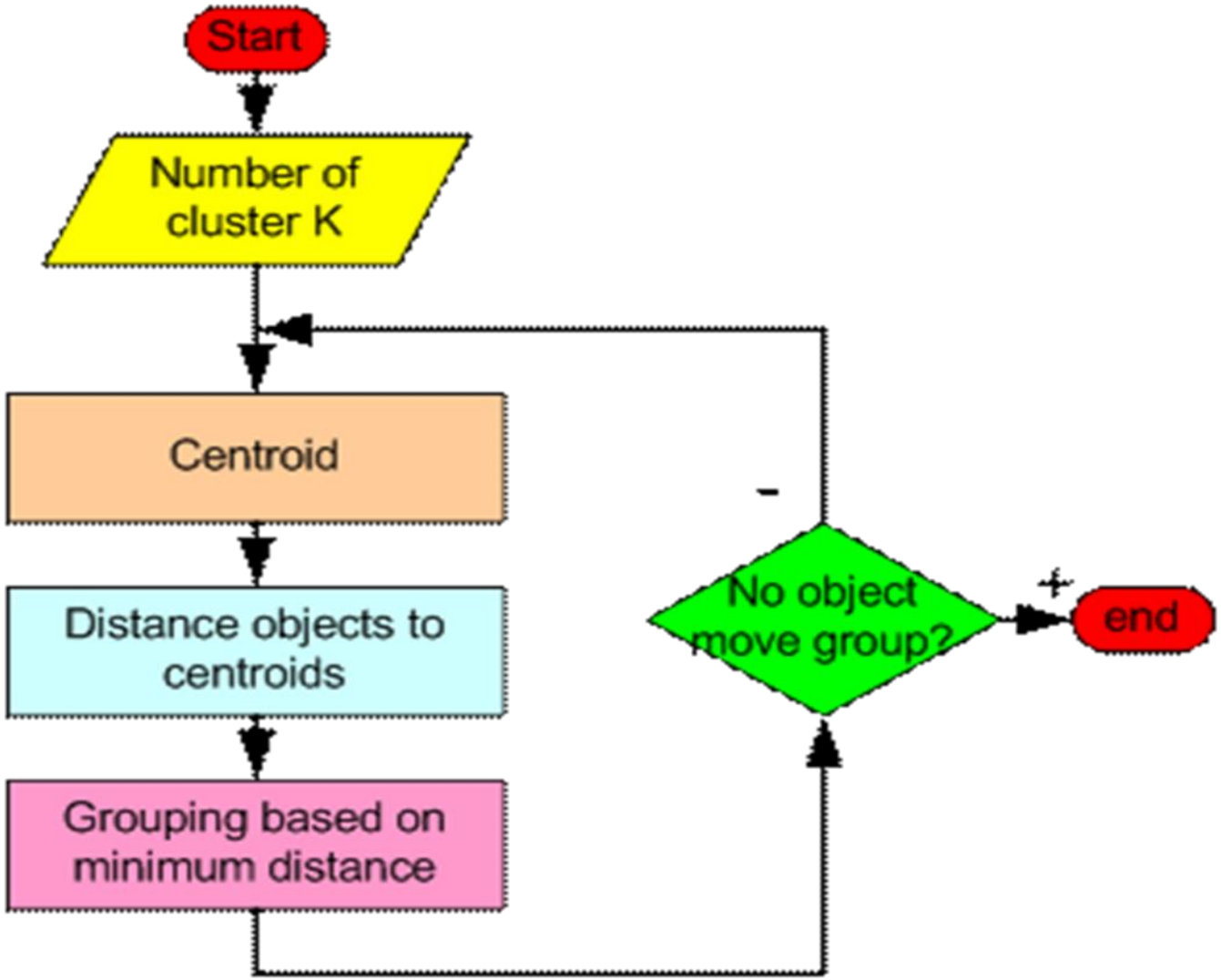
Cont.. Key point

- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum.
- cluster's centroid :- (arithmetic mean of all the data points that belong to that cluster).

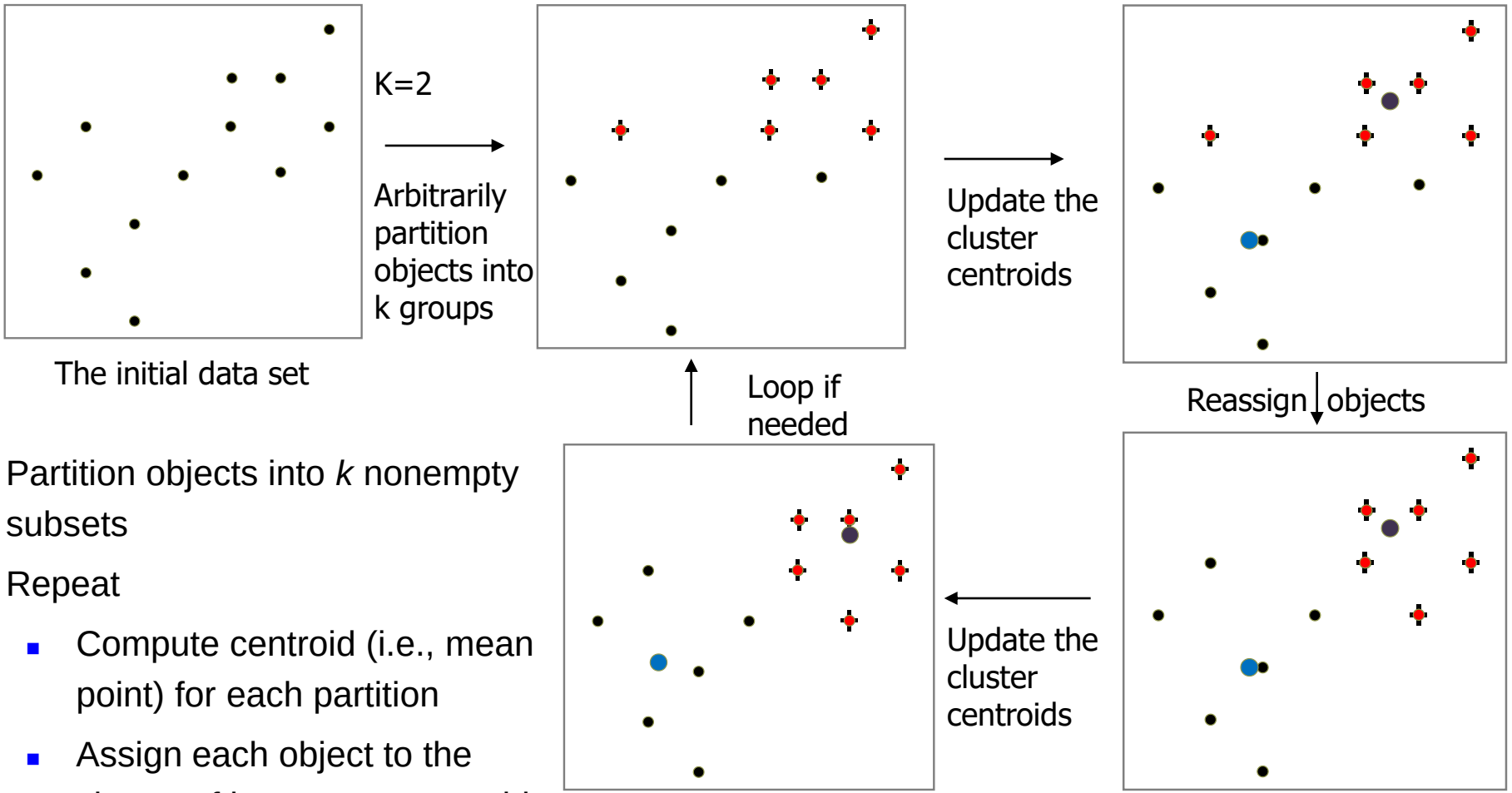
K-Means Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  3. Assign each object to the cluster with the nearest seed point
  4. Go back to Step 2, stop when the assignment does not change

Cont..


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

An Example of *K-Means* Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Key Analysis

Finally, this algorithm aims at minimizing an objective function as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

$x_i^{(j)}$ = data point

c_j = cluster center

n = Number of data points

k = Number of cluster

$||x_i^{(j)} - c_j||^2$ = distance between a data point $x_i^{(j)}$ and cluster center c_j

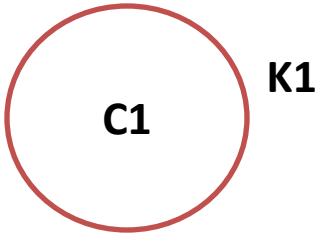
K-means: Method-01

Using K-means clustering, cluster the following data into two clusters and show each step.

{2, 4, 10, 12, 3, 20, 30, 11, 25}

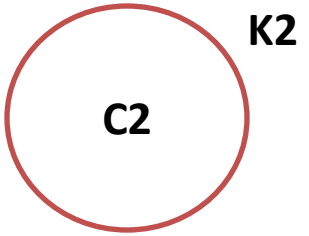
Solution:

Given: {2, 4, 10, 12, 3, 20, 30, 11, 25}



Step 1: Assign alternate value to each cluster randomly.

Step 2: k_1= {2, 10, 3, 30, 25} Mean value= 14
k_2= {4, 12, 20, 11} Mean value = 11.75



Step 3: Again assign the values,
k_1 = {20, 30, 25} Mean value = 25
k_2 = {2, 4, 10, 12, 3, 11} Mean value = 7

Step 4: Again assign the values,
k_1 = {20, 30, 25} Mean value = 25
k_2 = {2, 4, 10, 12, 3, 11} Mean value = 7

Step-3& 4: No change

Step-2: K1 cluster having cluster centroid C1 = 14
K2 cluster having Cluster centroid C2 = 11.5

Method-1 Analysis

- **Computation to move from Step-2 to Step-3.**
- In step-2, The clusters centroid values are as follows:
  - C1= 14 of K1 cluster
  - C2 = 11.5 of K2 cluster
- **Now Consider each data point from K1 and k2 clusters**
- **compute the distance from C1 and C2,**
- **consider the minimum distance,**
- **and assign the respective data point to the cluster k1 or k2.**

K1/C1 K2/C2

- **E.g. for data point '2' : Min(|2-14 |, |2-11.5 |)= 9.5 so, data point '2' assigns to cluster K2 having centroid C2[K2/C2].**

K-means: Method-02

{2, 4, 10, 12, 3, 20, 30, 11, 25}

Step 1: Randomly assign the means: $m_1 = 3$, $m_2 = 4$

Step 2: Group the numbers close to mean $m_1 = 3$ are grouped into cluster k_1 and $m_2 = 4$ are grouped into cluster k_2

Step 3: $k_1 = \{2, 3\}$, $k_2 = \{4, 10, 12, 20, 30, 11, 25\}$, $m_1 = 2.5$, $m_2 = 16$

Step 4: $k_1 = \{2, 3, 4\}$, $k_2 = \{10, 12, 20, 30, 11, 25\}$, $m_1 = 3$ $m_2 = 18$

Step 5: $k_1 = \{2, 3, 4, 10\}$, $k_2 = \{12, 20, 30, 11, 25\}$, $m_1 = 4.75$, $m_2 = 19.6$

Step 6: $k_1 = \{2, 3, 4, 10, 11, 12\}$, $k_2 = \{20, 30, 25\}$, $m_1 = 7$, $m_2 = 25$

Step 7: $k_1 = \{2, 3, 4, 10, 11, 12\}$, $k_2 = \{20, 30, 25\}$, $m_1 = 7$, $m_2 = 25$

Step 8: Stop. The clusters in step 6 and 7 are same.

Final answer: $k_1 = \{2, 3, 4, 10, 11, 12\}$ and $k_2 = \{20, 30, 25\}$

Applications of K-Means Clustering:

k-means can be applied to data that has a smaller number of dimensions, is numeric, and is continuous. such as

document clustering, identifying crime-prone areas, customer segmentation, insurance fraud detection, public transport data analysis, clustering of IT alerts…etc.

Comments on the K-Means Method

Strength:

Efficient: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.

Weakness:

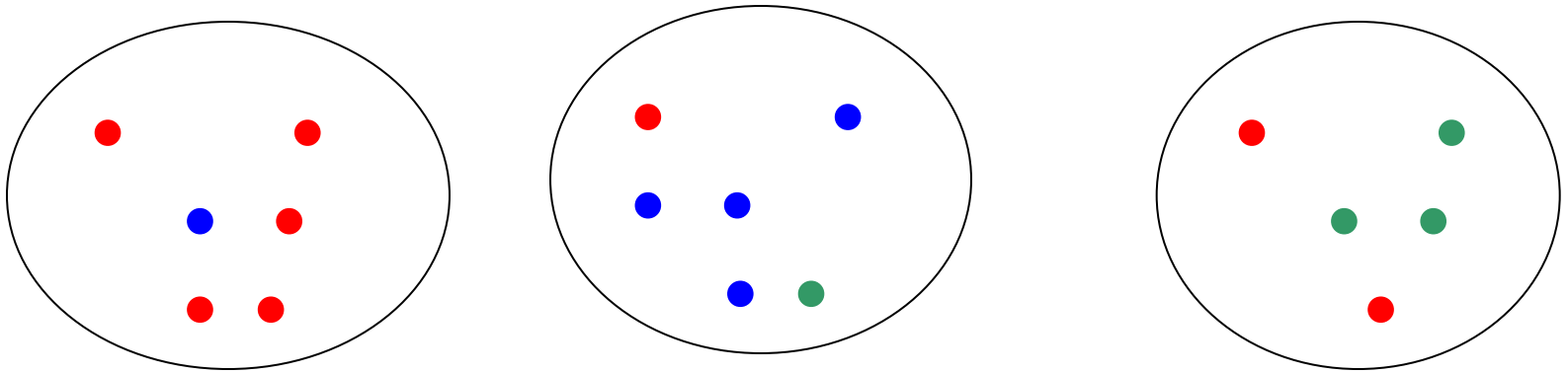
- Need to specify *k,* the *number* of clusters, in advance
- Sensitive to noisy data and *outliers*

Evaluation of Cluster Quality using Purity

- Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
- **Assesses a clustering with respect to ground truth … requires labeled data**
- Assume documents with C gold standard classes, while our clustering algorithms produce K clusters, ω1, ω2, …, ωK with ni members
- **Simple measure**: purity, the ratio between the dominant class in the cluster πi and the size of cluster ωi

$$Purity(\omega_i) = \frac{1}{n_i} \max_j (n_{ij}) \quad j \in C$$

Purity in Clusters: example



Cluster I **Cluster II** **Cluster III**

- Assume that we cluster three category of data items (those colored with red, blue and green) into three clusters as shown in the above figures. Calculate purity to measure the quality of each cluster.

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6 = 83%

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6 = 67%

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5 = 60%

R-Implements

- R base has a function to run the k mean algorithm. The basic function of k mean is:

kmeans(df, k)

arguments:

-df: dataset used to run the algorithm

-k: Number of clusters

- K-means clustering can handle larger datasets than hierarchical cluster approaches.
- Unlike hierarchical clustering, K-means clustering requires that the number of clusters to extract be specified in advance.

Package	Objective	Function	Argument
base	Train k-mean	kmeans()	df, k
	Access cluster	kmeans()$cluster	
	Cluster centers	kmeans()$centers	
	Size cluster	kmeans()$size	

EXAMPLE: mtcars DATASET IN R

- Let's look at the sort of output that the kmeans function provides for the mtcars dataset. We'll specify the number of clusters as 5.
- We can view the clusters that result from this.

```
> kmeans(data.matrix(mtcars[,1:4]), 5)$cluster
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive
                  1                   1                   4                   3
  Hornet Sportabout             Valiant          Duster 360           Merc 240D
                  5                   1                   5                   1
           Merc 230            Merc 280           Merc 280C          Merc 450SE
                  1                   1                   1                   3
          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental
                  3                   3                   5                   5
  Chrysler Imperial            Fiat 128          Honda Civic       Toyota Corolla
                  5                   4                   4                   4
       Toyota Corona     Dodge Challenger           AMC Javelin          Camaro Z28
                  4                   3                   3                   5
    Pontiac Firebird            Fiat X1-9       Porsche 914-2        Lotus Europa
                  5                   4                   4                   4
     Ford Pantera L         Ferrari Dino       Maserati Bora          Volvo 142E
                  5                   2                   5                   4
```

- We can also view the cluster centre points.

```
> kmeans(data.matrix(mtcars[,1:4]), 5)$centers
       mpg      cyl     disp        hp
1 31.00000 4.000000  76.1250  62.25000
2 14.64444 8.000000 388.2222 232.11111
3 24.18571 4.000000 121.7143  94.28571
4 16.83333 7.666667 284.5667 158.33333
5 19.46667 6.000000 170.8667 124.33333
```

Comments on the K-Means Method

Limitation

- ✓ Applicable only when *mean* is defined, then what about categorical data?
- ✓ Need to specify *k*, the *number* of clusters, in advance
- ✓ Unable to handle noisy data and *outliers*

- The k-means algorithm is sensitive to outliers !
  - ✓ Since an object with an extremely large value may substantially distort the distribution of the data.

The K-Medoids Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids)-

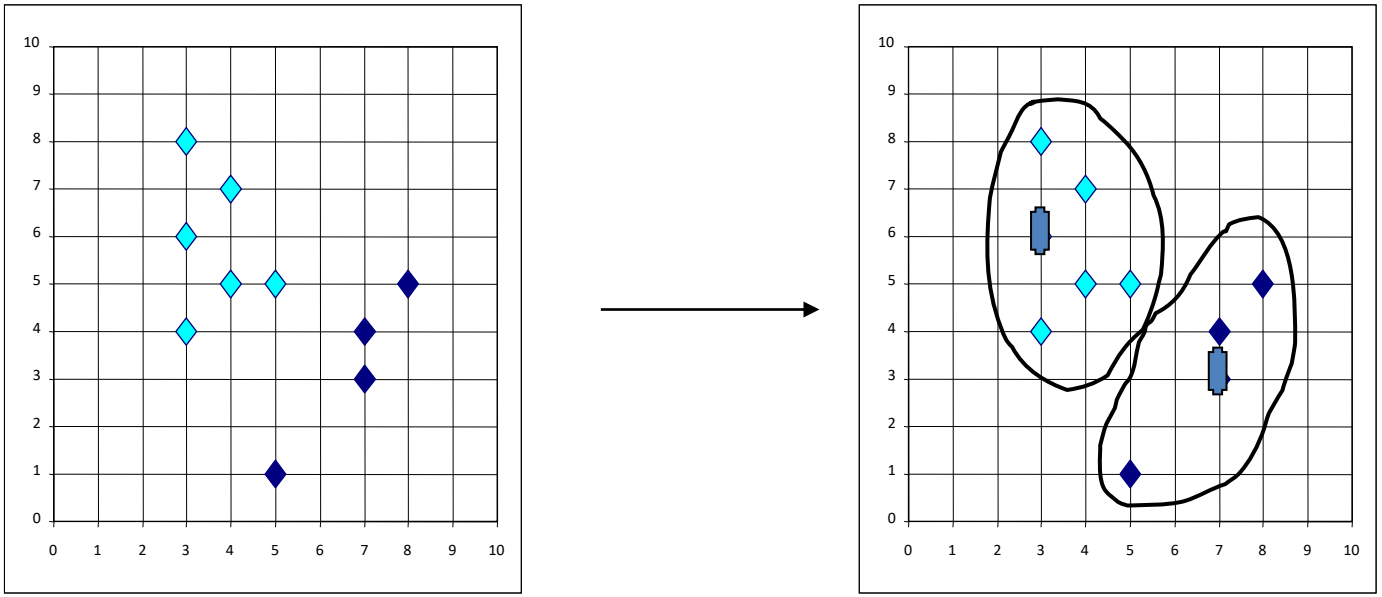
✓ starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering.

✓ *PAM* works effectively for small data sets, but does not scale well for large data sets

✓ *CLARA* (**CL**ustering LARge Applications)

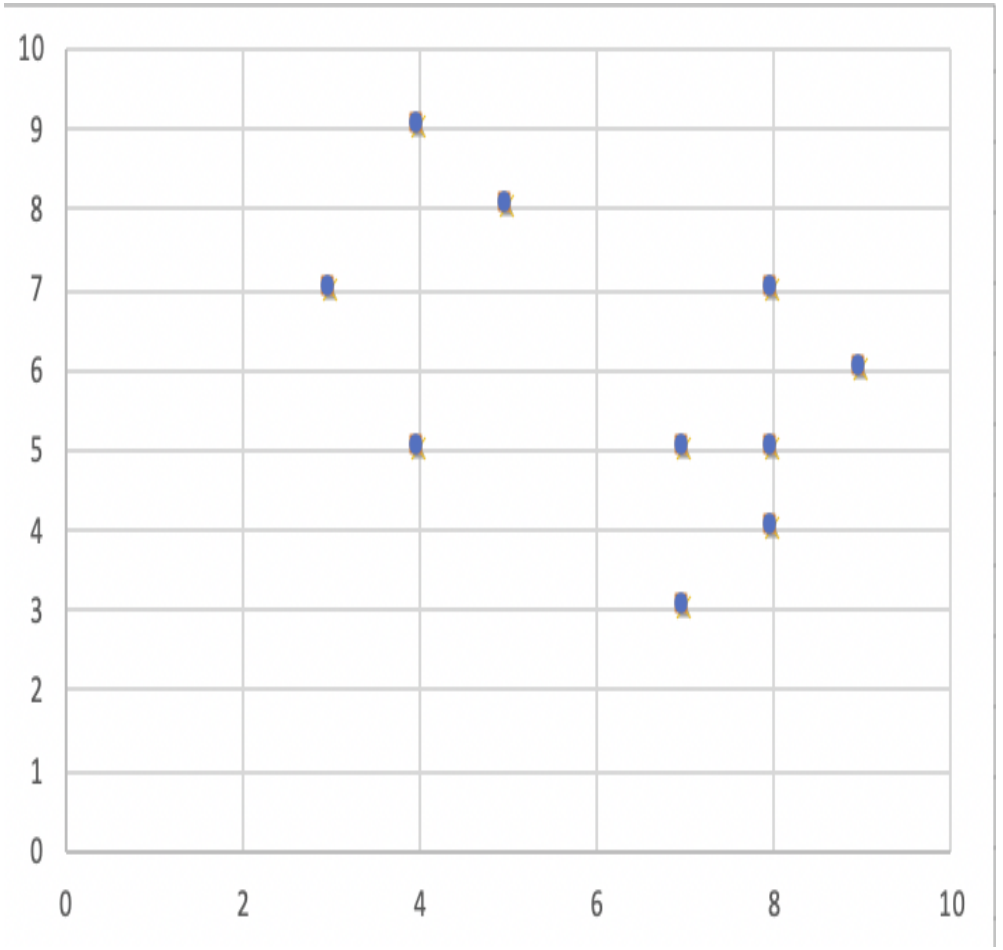
✓ *CLARANS* (**CL**ustering **LA**ge Applications based upon **RAN**domized **S**earch)

- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the *most centrally* located object in a cluster.



Cluster Computation using K-Medoids through a dataset scenario

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



Graph is drawn using the above data points

Step 1: Let the randomly selected 2 medoids, so select k = 2 and let **C1 = (4, 5)** and **C2 =(8, 5)** are the two medoids.

Step-2: Calculating cost. The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

- Each point is assigned to the cluster of that medoid whose dissimilarity is less.
- points 1, 2, 5 go to cluster C1
- 0, 3, 6, 7, 8 go to cluster C2. The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20
- Step 3: randomly select one non-medoid point and recalculate the cost.

Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	**8**	**4**	-	-
8	7	5	3	2
9	**4**	**5**	-	-

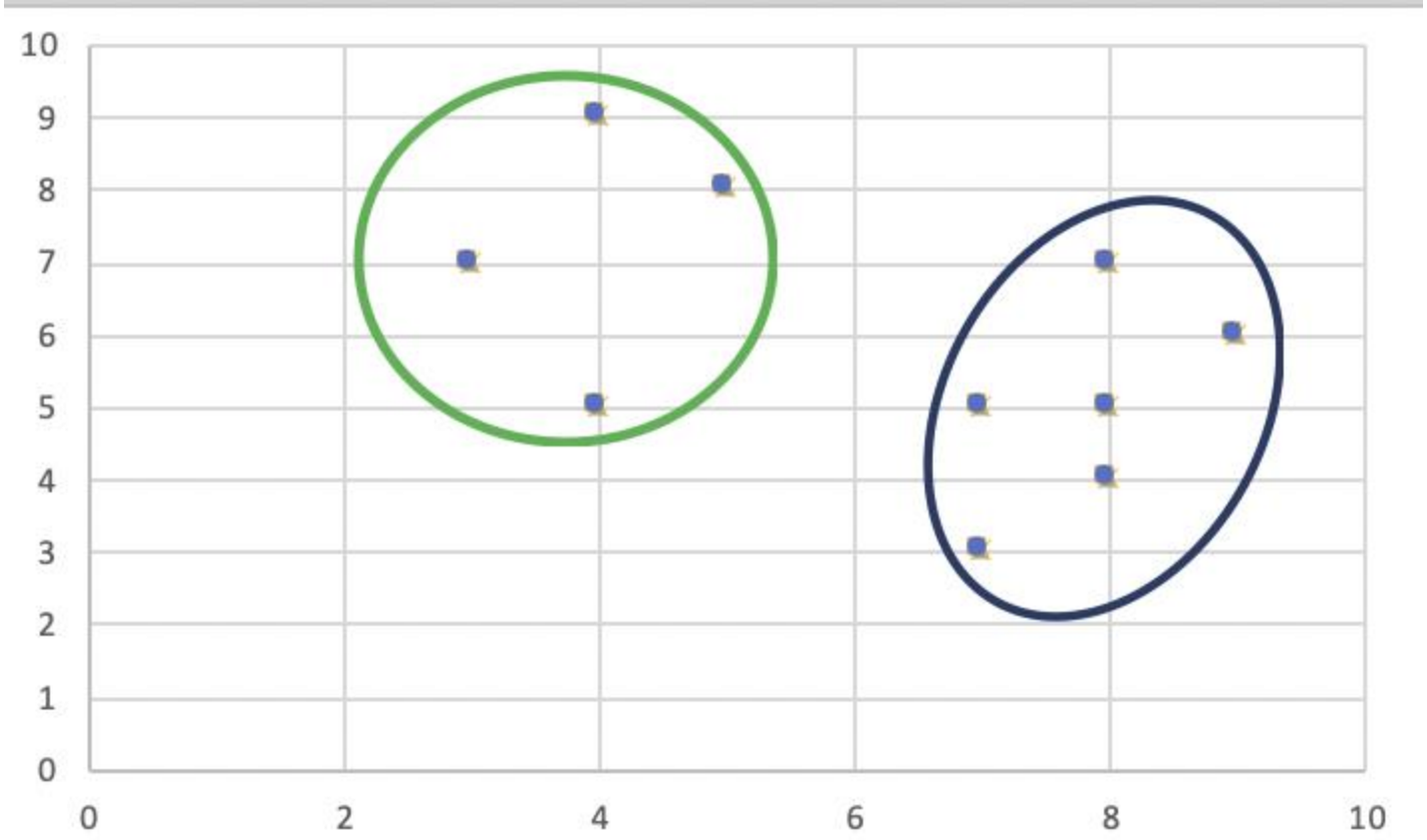
Each point is assigned to that cluster whose dissimilarity is less.

So, the points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

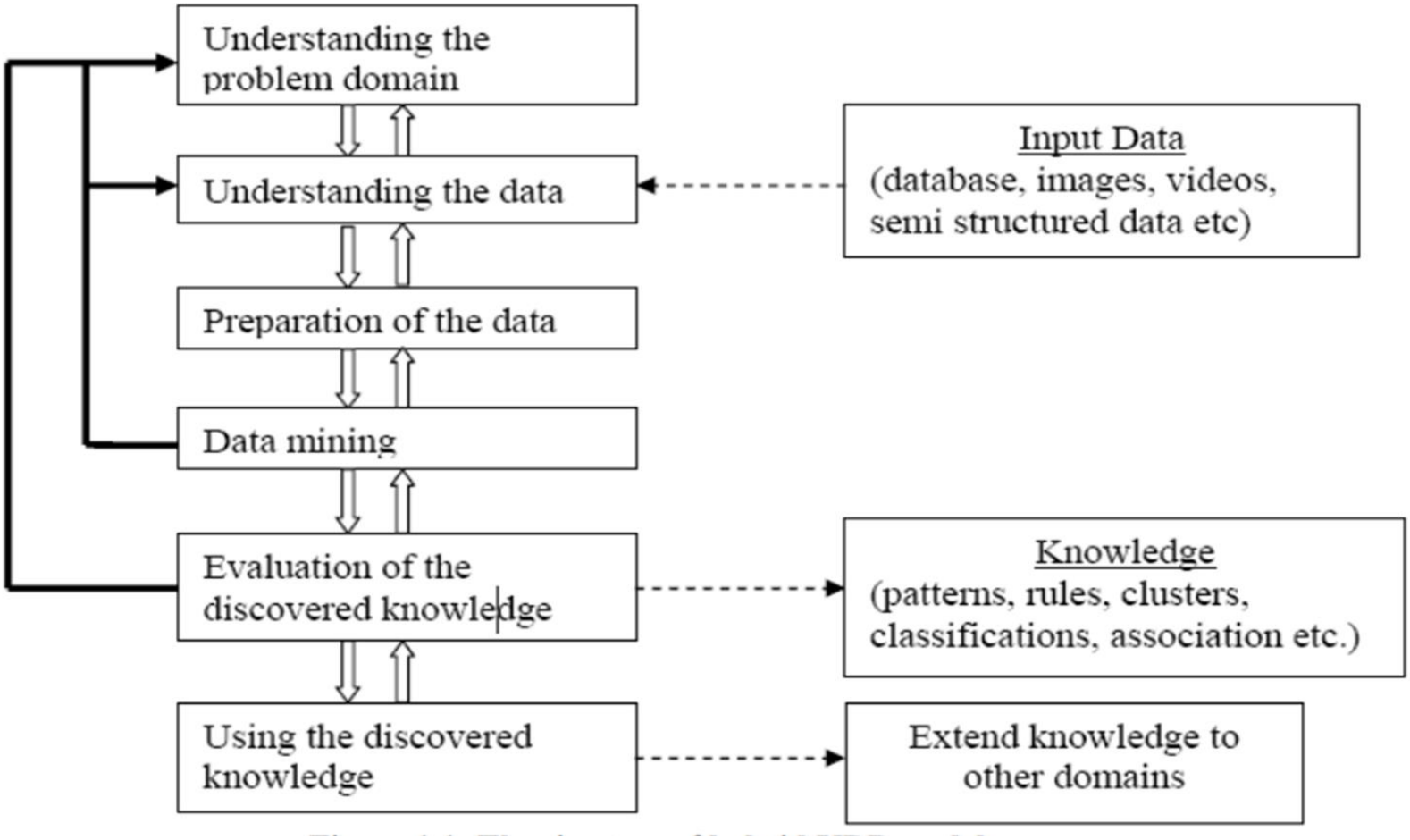
The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22

Swap Cost = New Cost – Previous Cost = 22 – 20 and 2 >0

❖As the swap cost is not less than zero, we undo the swap. Hence (4, 5) and (8, 5) are the final medoids. The clustering would be in the following way



Research Methodology



[A-16]:-LAB-08

Consider a dataset without any class level and implement the K-means Algorithm.

Try to analyze and understand the final results. Also, interpret the final result to estimate the accuracy.

TASK FOR YOU [A17]

1. Using K-means clustering, cluster the following data into two clusters and show each step.

{3, 5, 10, 13, 4, 21, 31, 12, 26}.

Give your step by step computational analysis.

2. Formulate any four cluster scenarios to Calculate purity to measure the quality of each cluster.

3. Investigate the computational processes of K-medoid algorithm with a suitable scenario.

Cheers For the Great Patience!

Query Please?

Compiled By: Dr. Nilamadhab Mishra [(PhD- CSIE) Taiwan]