Clustering

Dr. Md. Saddam Hossain
Associate Professor
Dept. of CSE, UIU
Email: saddam@cse.uiu.ac.bd

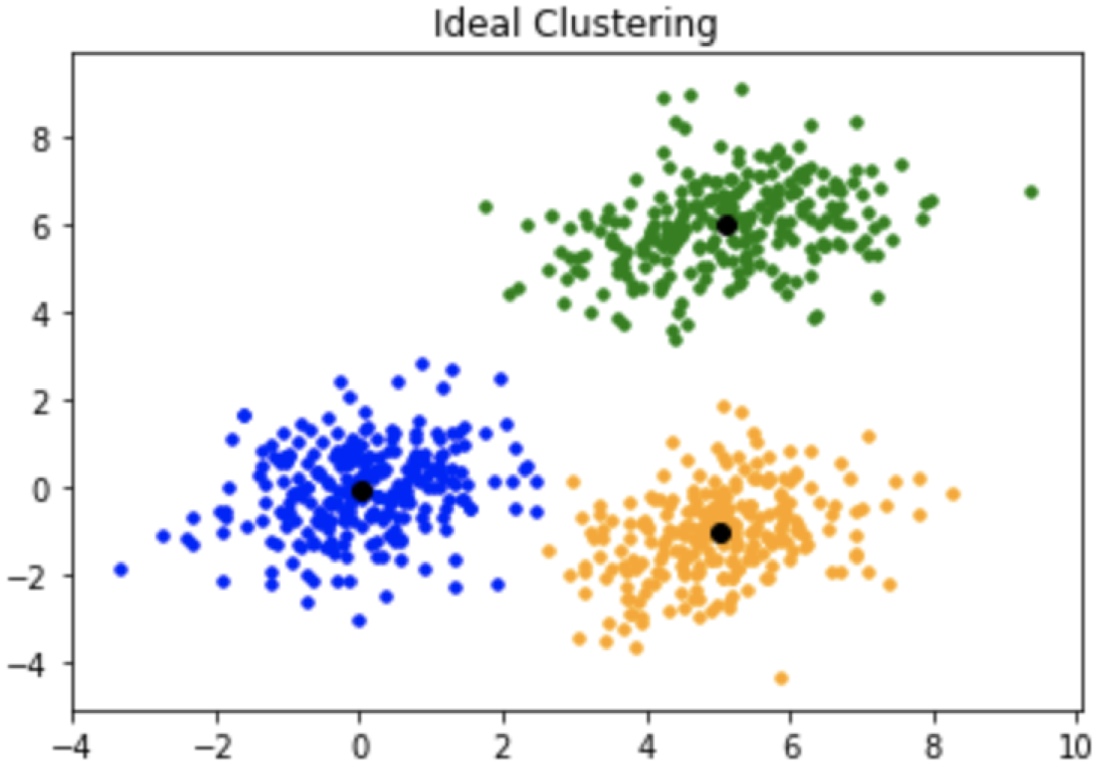
What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.
- The commonest form of *unsupervised learning*
    - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given
  - A common and important task that finds many applications in IR and other places

What is clustering?

- Clustering also called *data segmentation* in some applications because clustering partitions large datasets into groups according to their similarity
- A cluster is a collection of data objects that are similar to one another within the cluster and dissimilar to objects in other clusters. Clustering is sometimes called *automatic classification*
- Clustering can also be used for *outlier detection* where outliers may be more interesting than common cases
- Clustering is a form of *learning by observation* rather than learning by examples

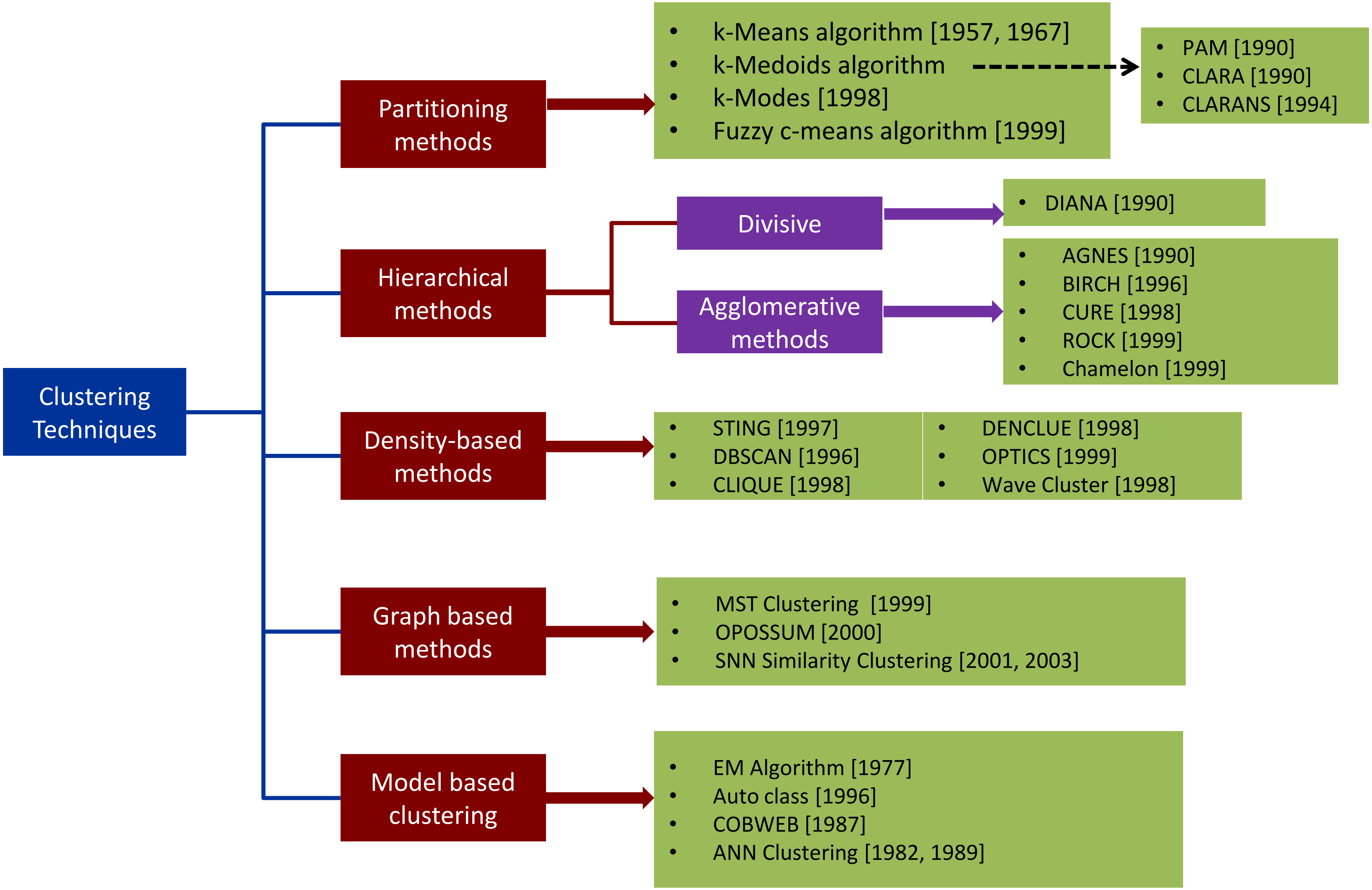
A data set with clear cluster structure



- How would you design an algorithm for finding the three clusters in this case?

Applications of clustering

- Marketing: segmentation of the customer based on behavior
- Banking: ATM Fraud detection (outlier detection)
- ATM classification: segmentation based on time series
- Gene analysis: Identifying gene responsible for a disease
- Image processing: identifying objects on an image (face detection)
- Insurance: identifying groups of car insurance policy holders with a high average claim cost
- Houses: identifying groups of houses according to their house type, value, and geographical location



Clustering Techniques

Partitioning methods

- k-Means algorithm [1957, 1967]
- k-Medoids algorithm
- k-Modes [1998]
- Fuzzy c-means algorithm [1999]

- PAM [1990]
- CLARA [1990]
- CLARANS [1994]

Hierarchical methods

Divisive

- DIANA [1990]

Agglomerative methods

- AGNES [1990]
- BIRCH [1996]
- CURE [1998]
- ROCK [1999]
- Chamelon [1999]

Density-based methods

- STING [1997]
- DBSCAN [1996]
- CLIQUE [1998]

- DENCLUE [1998]
- OPTICS [1999]
- Wave Cluster [1998]

Graph based methods

- MST Clustering [1999]
- OPOSSUM [2000]
- SNN Similarity Clustering [2001, 2003]

Model based clustering

- EM Algorithm [1977]
- Auto class [1996]
- COBWEB [1987]
- ANN Clustering [1982, 1989]

Issues for clustering

- Representation for clustering
  - Document representation
    - Vector space? Normalization?
      - Centroids aren’t length normalized
  - Need a notion of similarity/distance
- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters - too large or small
      - If a cluster's too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much.

Notion of similarity/distance

- Ideal: semantic similarity.
- Practical: term-statistical similarity
  - We will use cosine similarity.
  - Docs as vectors.
  - For many algorithms, easier to think in terms of a *distance* (rather than similarity) between docs.
  - We will mostly speak of Euclidean distance
    - But real implementations use cosine similarity

Partitioning Algorithms

- Partitioning method: Construct a partition of *n* documents into a set of *K* clusters
- Given: a set of documents and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms

K-Means Clustering

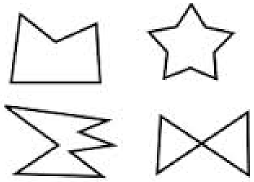
- Assumes documents are real-valued vectors.
- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, *c*:

$$\vec{\mu}(\mathrm{c}) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

- Reassignment of instances to clusters is based on distance to the current cluster centroids.
  - (Or one can equivalently phrase it in terms of similarities)

K-means

- Strengths
  - Simple iterative method
  - User provides “K”
- Weaknesses
  - Often too simple → bad results
  - Difficult to guess the correct “K”
  - Not good for non convex shape
  - Sensitive to noise and outliers



K-means Clustering

- Iterate:
  - Calculate distance from objects to cluster centroids.
  - Assign objects to closest cluster
  - Recalculate new centroids
- Stop based on convergence criteria
  - No change in clusters
  - Max iterations

k-Means Algorithm

The algorithm can be stated as follows.

- First it selects *k* number of objects at random from the set of n objects. These *k* objects are treated as the *centroids* or *center of gravities* of *k* clusters.

- For each of the remaining objects, it is assigned to one of the closest centroid. Thus, it forms a collection of objects assigned to each centroid and is called a *cluster*.

- Next, the centroid of each cluster is then updated (by calculating the mean values of attributes of each object).

- The assignment and update procedure is until it reaches some stopping criteria (such as, number of iteration, centroids remain unchanged or no assignment, etc.)

k-Means Algorithm

Input: D is a dataset containing *n* objects, *k* is the number of cluster

Output: A set of *k* clusters

Steps:

1. Randomly choose *k* objects from D as the initial cluster centroids.
2. **For** each of the objects in D **do**
    - Compute distance between the current objects and *k* cluster centroids
    - Assign the current object to that cluster to which it is closest.
3. Compute the “cluster centers” of each cluster. These become the new cluster centroids.
4. Repeat step 2-3 until the convergence criterion is satisfied
5. Stop

k-Means Algorithm

Note:

1) Objects are defined in terms of set of attributes. $A = \{A_1, A_2, \ldots\ldots, A_m\}$ where each A_i is continuous data type.

2) Distance computation: Any distance such as L_1, L_2, L_3 or cosine similarity.

3) Minimum distance is the measure of closeness between an object and centroid.

4) Mean Calculation: It is the mean value of each attribute values of all objects.

5) Convergence criteria: Any one of the following are termination condition of the algorithm.
   - Number of maximum iteration permissible.
   - No change of centroid values in any cluster.
   - Zero (or no significant) movement(s) of object from one cluster to another.

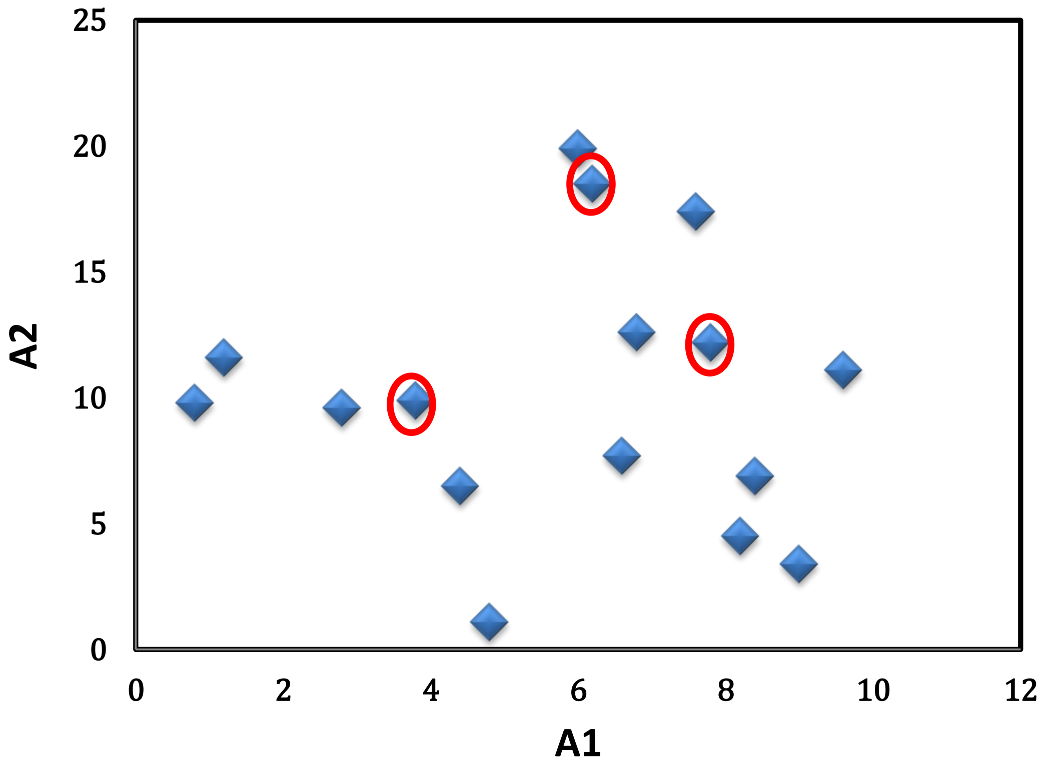
K-means Issues

- Distance measure is squared Euclidean
  - Scale should be similar in all dimensions
  - Not good for nominal data.
- Approach tries to minimize the within-cluster sum of squares error (WCSS)
  - Implicit assumption that SSE is similar for each group

Illustration of k-Means clustering algorithms

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

16 objects with two attributes A_1 and A_2.



Plotting data of Table

Illustration of k-Means clustering algorithms

- Suppose, k=3. Three objects are chosen at random shown as circled (see Fig 16.1). These three centroids are shown below.

Initial Centroids chosen randomly

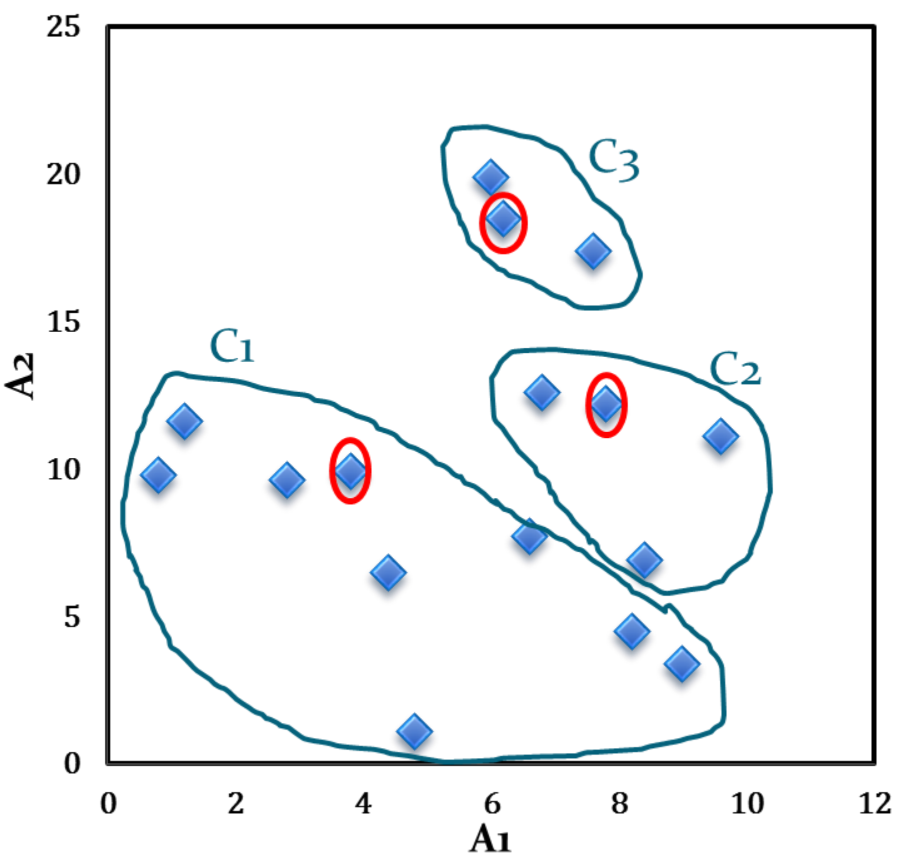
Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

- 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.
- Assignment of each object to the respective centroid is shown in the right-most column and the clustering so obtained is shown next.

Illustration of k-Means clustering algorithms

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

Distance calculation



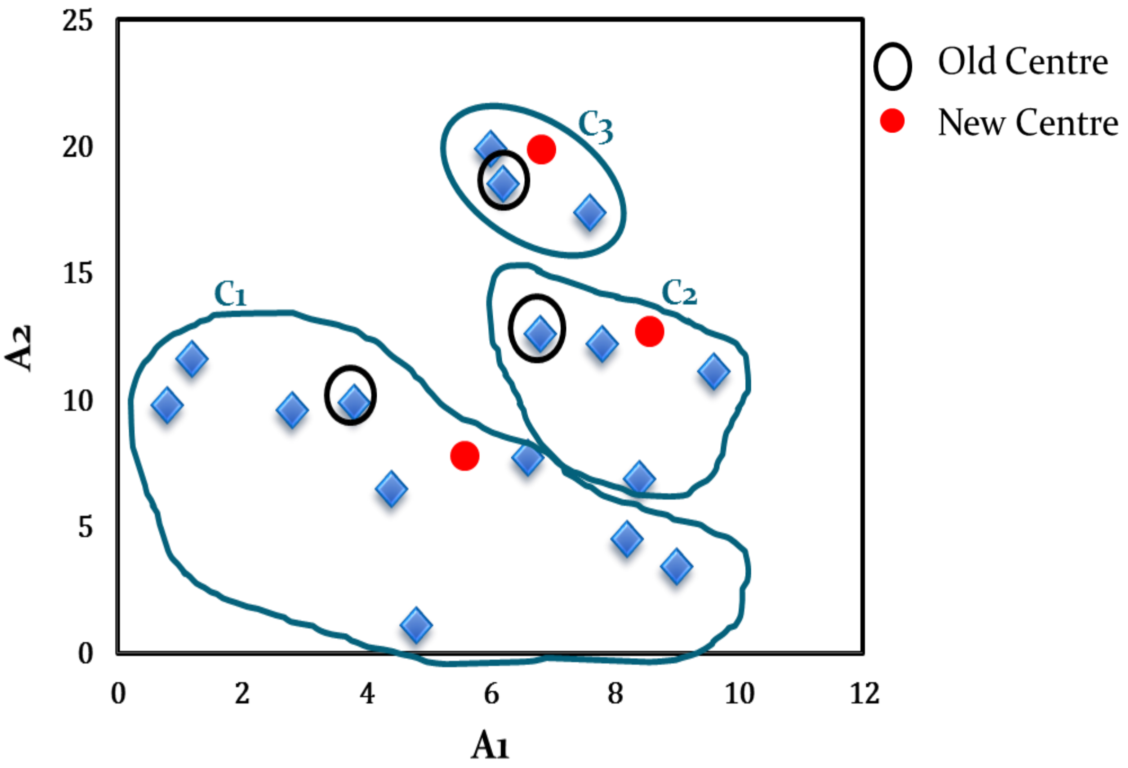
Initial cluster with respect to Table 16.2

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 Figure.

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

Calculation of new centroids

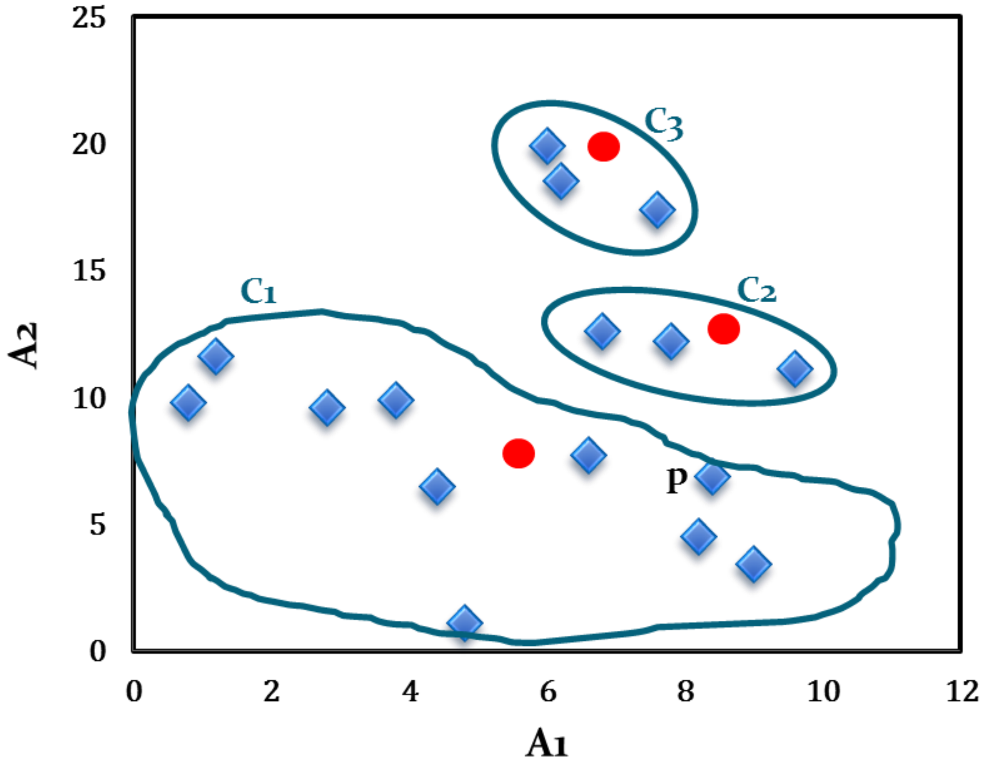


Initial cluster with new centroids

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.

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



Cluster after first iteration

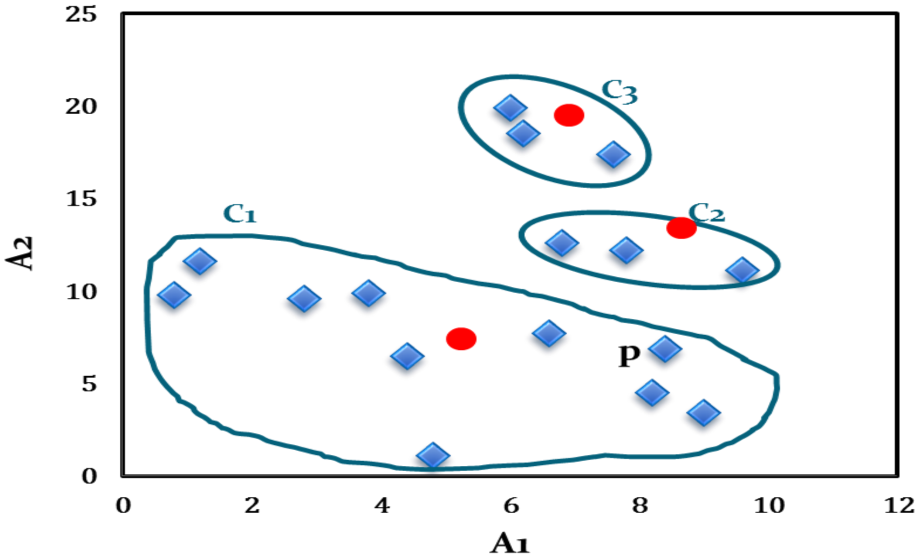
Illustration of k-Means clustering algorithms

- 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.

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

Cluster after Second iteration



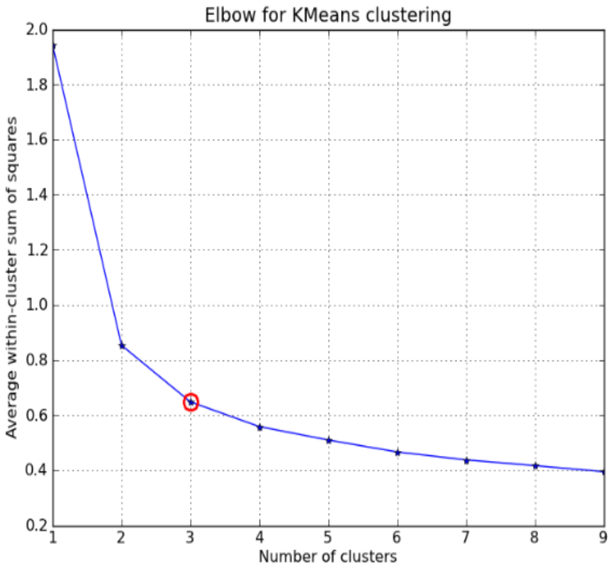
Comments on k-Means algorithm

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.
   - 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.
   - Normally $k \ll n$ and there is heuristic to follow $k \approx \sqrt{n}$.

Comments on k-Means algorithm (Elbow Method)

2. **Value of k:**

- The Elbow method involves finding a metric to evaluate how good a clustering outcome is for various values of K and finding the elbow point.
- Initially the quality of clustering improves rapidly when changing value of K, but eventually stabilizes.
- The elbow point is the point where the relative improvement is not very high any more.



Comments on k-Means algorithm

Choosing initial centroids:

- Another requirement in the k-Means algorithm to choose initial cluster centroid for each *k* would be clusters.
- It is observed that the k-Means algorithm terminate whatever be the initial choice of the cluster centroids.
- It is also observed that initial choice influences the ultimate cluster quality. In other words, the result may be trapped into local optima, if initial centroids are not chosen properly.
- One technique that is usually followed to avoid the above problem is to choose initial centroids in multiple runs, each with a different set of randomly chosen initial centroids, and then select the best cluster (with respect to some quality measurement criterion, e.g. SSE).
- However, this strategy suffers from the combinational explosion problem due to the number of all possible solutions.

Comments on k-Means algorithm

Distance Measurement:

Thus, in the context of different measures, the sum-of-squared error (i.e., objective function/convergence criteria) of a clustering can be stated as under.

Data in Euclidean space (L_2 norm):

$$SSE = \sum_{i=1}^{k} \sum_{x \in \boldsymbol{C}_i} (c_i - x)^2$$

Data in Euclidean space (L_1 norm):

The Manhattan distance (L_1 norm) is used as a proximity measure, where the objective is to minimize the sum-of-absolute error denoted as SAE and defined as

$$SAE = \sum_{i=1}^{k} \sum_{x \in \boldsymbol{C}_i} |c_i - x|$$

Comments on k-Means algorithm

Distance with document objects

Suppose a set of *n* document objects is defined as *d* document term matrix (DTM) (a typical look is shown in the below form).

Document	Term			
	t_1	t_2	… …	t_n
D_1	f_{11}	f_{12}		f_{1n}
D_2	f_{21}	f_{22}		f_{2n}
⋮				
D_n	f_{n1}	f_{n2}		f_{nn}

$$\cos(x, c_i) = \frac{x \cdot c_i}{\|x\|\|c_i\|}$$

$$x \cdot c_i = \textstyle\sum_j x_j \, c_{ij} \qquad \text{and} \qquad \|x\| = \sqrt{\textstyle\sum_j^p {x_j}^2}$$

$$\hat{x} = \textstyle\sum_{j=1}^p \hat{x}_j \qquad \hat{c}_i = \textstyle\sum_{j=1}^p \hat{c}_{ij} \qquad \|\|c_{ij}\|\| = \sqrt{\textstyle\sum_j^p {c_{ij}}^2}$$

Clustering Quality

Minkowski Distance

The minkowski distance of order *p* (where p is an integer) between two points *X=(x1, x2,…, xn)* and *Y=(y1, y2,…, yn)*

is defined as:

$$D\left(X,Y\right)=\left(\sum_{i=1}^{n}\left|x_i-y_i\right|^p\right)^{\frac{1}{p}}$$

Minkowski Distance

Two important special cases of the Minkowski metric are p=1 and p=2 of the Minkowski distance:

Manhattan distance or city block distance or L1 norm (when p=1):

$$d(x, y) = L1 = \sum_{i=1}^{n} |xi - yi|$$

Minkowski Distance

Two important special cases of the Minkowski metric are p=1 and p=2 of the Minkowski distance:

Euclidean distance or L2 norm (when p=2):

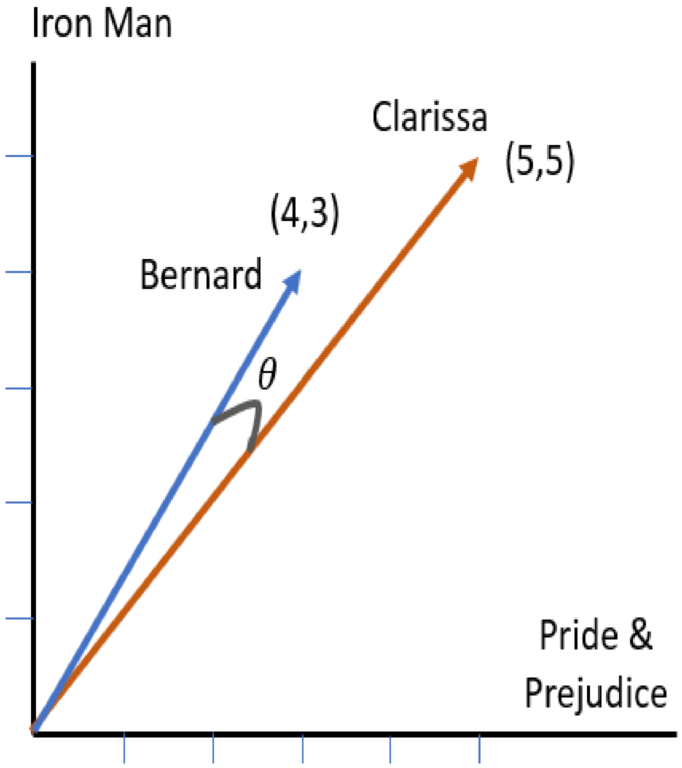
$$d(x, y) = L2 = \sqrt{\sum_{i=1}^{n} (xi - yi)^2}$$

Cosine Similarity

Cos(x,y) measures the similarity of the two objects x and y by calculating the cosine of the angle between the feature vectors. The degree of similarity range from -1 (highest degree of dissimilar) to 1 (highest degree of similar with 0 degree)

$$sim(x, y) = cos(x, y) = \frac{\sum_{i=1}^{n} x_i * yi}{\sqrt{\sum_{i=1}^{n} {x_i}^2} * \sqrt{\sum_{i=1}^{n} {y_i}^2}}$$

Cosine Similarity



$Calculating$:

$$b.c = \sum_{i=1}^{n} b_i c_i = (4 \times\ 5) + (3 \times 5) = 35$$

$$\|b\| = \sqrt{4^2 + 3^2} = 5$$

$$\|c\| = \sqrt{5^2 + 5^2} = 5\sqrt{2}$$

$$similarity = \frac{35}{5 \times 5\sqrt{2}} \sim 0.989$$

Jaccard Coefficient

The *Jaccard coefficient* or *Tanimoto coefficient* is as follows:

$$sim(x,y) = jaccard(x,y) = \frac{|X \ \cap Y|}{|X\ U\ Y|}$$

Measuring Clustering Quality

- *How good is the clustering generated by a method, and how can we compare the clusterings generated by different methods?*
- *Intrinsic methods* evaluate a clustering by examining how well the clusters are separated and how compact the clusters are

Silhouette coefficient

- For a data set, *D*, of *n* objects, suppose *D* is partitioned into *k* clusters, C_1, … ,C_k.

- For each object ***o*** ϵ *D*, we calculate *a(**o**)* as the average distance between ***o*** and all other objects in the cluster to which ***o*** belongs.

$$\text{a(0)}= \frac{\sum_{o' \in Ci,\, o \neq o'}^{n} dist\ (o,o')}{|Ci|}$$

Silhouette coefficient

- Similarly, *b*(***o***) is the minimum average distance from ***o*** to all clusters to which ***o*** does not belong.

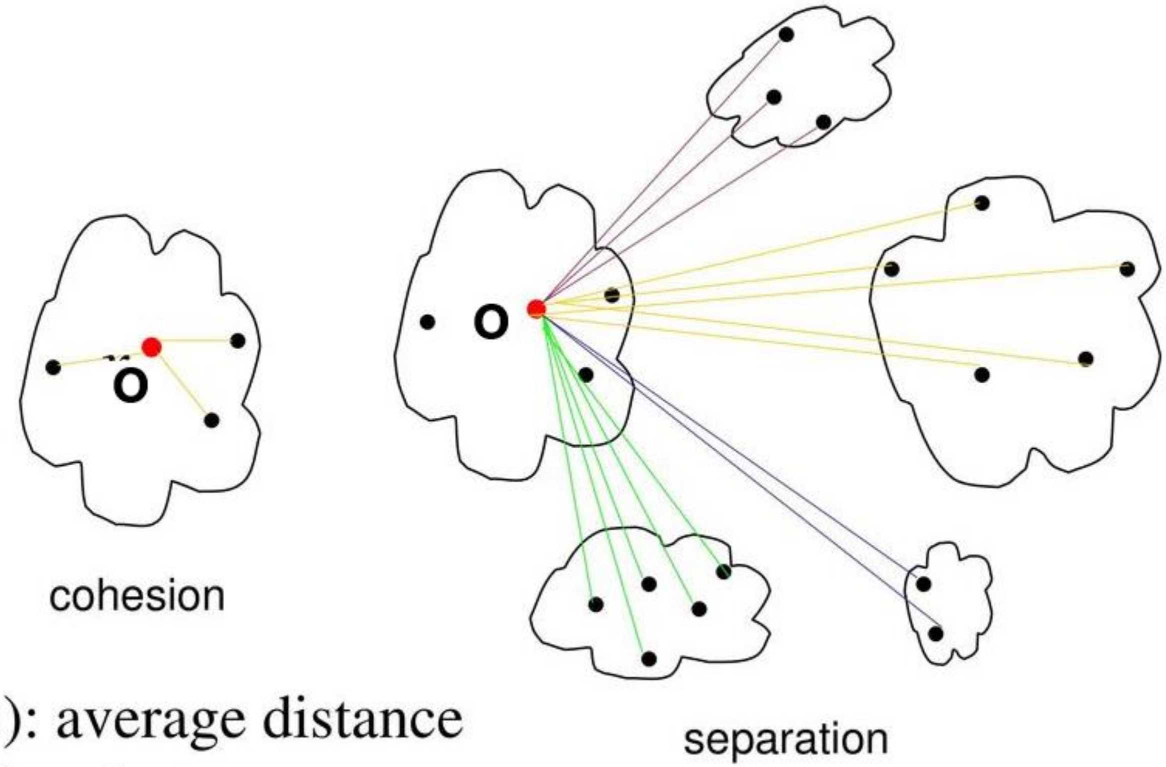
$$b(\boldsymbol{o}) = \min_{C_j:1\leq j\leq k, j\neq i} \left\{ \frac{\sum_{\boldsymbol{o}' \in C_j} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_j|} \right\}$$

Silhouette coefficient

- The **silhouette coefficient** of ***o*** is then defined as

$$s(\boldsymbol{o}) = \frac{b(\boldsymbol{o}) - a(\boldsymbol{o})}{\max\{a(\boldsymbol{o}), b(\boldsymbol{o})\}}$$

Silhouette coefficient



a(o): average distance in the cluster

b(o): average distances to others clusters, find minimal

Silhouette coefficient

- The value of the silhouette coefficient is between -1 and 1. The value of ***a***(***o***) reflects the compactness of the cluster to which ***o*** belongs. The smaller the value, the more compact the cluster.
- The value of *b*(***o***) captures the degree to which ***o*** is separated from other clusters. The larger *b*(***o***) is, the more separated ***o*** is from other clusters.

Silhouette coefficient

1: Means clusters are well apart from each other and clearly distinguished.

0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.

-1: Means clusters are assigned in the wrong way.

Silhouette coefficient

Given K=3 where C1={2,4}, C2={6,7,8,9}, and C3={10,11,13}

Center of C1=3

a(O)= sqrt((3-2)^{2} +(3-4)^{2})/|2|=1.41/1=1.41

b(O)$_{C2}$= sqrt((3-6)^{2}+(3-7)^{2}+(3-8)^{2}+(3-9)^{2})/|C2|=2.31

b(O)$_{C3}$= sqrt((3-10)^{2}+(3-11)^{2}+(3-13)^{2})/|C3|=4.86

b(O)=min{2.31 , 4.86}=2.31

S(O)=2.31-1.41/2.31=0.9/2.31=0.38

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

Cluster Variation/SSE

- Consider six points in 1-D space having the values *1, 2, 3, 8, 9, 10*, and *25*, respectively
- The quality of cluster *Ci* can be measured by the ***within cluster variation*** **or** ***sum of squared error (SSE)***

$$SSE = \sum_{i=1}^{k} \sum_{x \in \boldsymbol{C}_i} (c_i - x)^2$$

SSE

- If we apply *k*-means using *k* =2 and applying SSE, the partitioning {{1, 2,3}, and {8, 9, 10,25}} has the within-cluster variation. Then mean of {1, 2,3} is 2 and mean of {8, 9, 10,25} is 13. Thus,

$$SSE= (1-2)^2+(2-2)^2+(3-2)^2+(8-13)^2+(9-13)^2+(10-13)^2+(25-13)^2$$

$$=1+0+1+25+16+9+144= 196$$

SSE

- If we reorganize the partitions {{1, 2, 3, 8}, and {9, 10,25}} where *k* =2 and applying SSE, the partitioning has the within-cluster variation. Then mean of {1, 2,3, 8} is 3.5 and mean of {9, 10, 25} is 14.67. Thus,

$$\text{SSE}= (1-3.5)^2+(2-3.5)^2+(3-3.5)^2+(8-3.5)^2+(9-14.67)^2+(10-14.67)^2+(25-14.67)^2$$

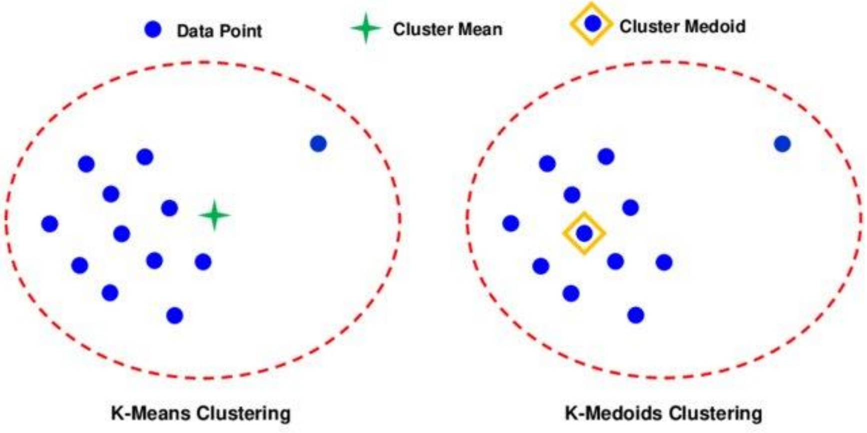
$$=6.25+2.25+0.25+20.25+32.15+21.80+106.71= 189.67$$

Observation

- The latter partitioning has the lowest within-cluster variation
- The *k*-means method assigns the value 8 to a cluster different from that containing 9 and 10 due to the outlier point 25
- The center of the second cluster, 14.67, is substantially far from all the members in the cluster

More on KMeans

- K-Means algorithm in terms of the way it selects the clusters' centers which may not actual points
- K-Medoids is a clustering algorithm picks the actual data points from the clusters as their centres



Kmedoids Algorithm

- Randomly choose ‘k’ points from the input data (‘k’ is the number of clusters to be formed).
- Each data point gets assigned to the cluster to which its nearest medoid belongs.
- For each data point of cluster i, its distance from all other data points is computed and added. The point of ith cluster for which the computed sum of distances from other points is minimal is assigned as the medoid for that cluster.
- Steps (2) and (3) are repeated until convergence is reached i.e. the medoids stop moving.

Example

First mediods:-

med1 (3, 4)   med2(6, 3)

Second mediods:

med1 (5, 2) , med2(7, 1)

First set of Mediods has low cost, thus previous cluster was better

pts	x	y	d_{med_1}	d_{med_2}	min-cost	d_{med_1}	d_{med_2}	min-cost
P1	1	3	2+1=3	5+0=5	3(m1)	4+1=5	6+2=8	5(m1)
P_2	2	1	1+3=4	4+2=6	4(m1)	3+1=4	5:	4(m1)
med1 = P_3	3	4				2+2=4	4+3=7	4(m1)
P_4	5	2	2+2=4	1+1=2	2(m2)	1+1=2		
med2 = P_5	6	3	3+1=4			1+1=2	1+2=3	2(m1)
P_6	7	1	4+3=7	1+2=3	3(m2)			
					Total cost = 12			Total cost = 15