Learning

BY
DR. ANUPAM GHOSH

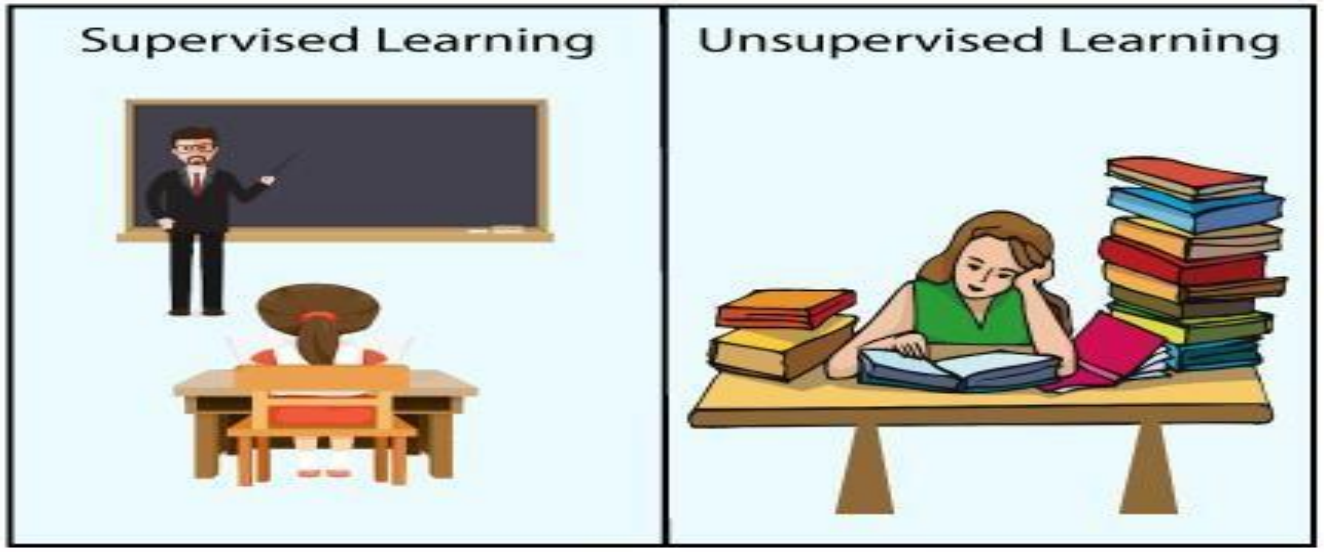
DATE: .9TH SEPT, 2024

Supervised vs. Unsupervised

fruit	length	width	weight	label
fruit 1	165	38	172	Banana
fruit 2	218	39	230	Banana
fruit 3	76	80	145	Orange
fruit 4	145	35	150	Banana
fruit 5	90	88	160	Orange
...				
fruit n	...	...	...	...

Unsupervised learning:
Learning a model from unlabeled data.

Supervised learning:
Learning a model from labeled data.



Styles of Learning

Supervised	Unsupervised	Semi-Supervised	Reinforcement
• Data has **known labels** or output	• Labels or output unknown • Focus on **finding patterns and gaining insight** from the data	• Labels or output known for **a subset of** data • A blend of supervised and unsupervised learning	• Focus on **making decisions** based on previous experience • Policy-making with feedback
• Insurance underwriting • Fraud detection	• Customer clustering • Association rule mining	• Medical predictions (where tests and expert diagnoses are expensive, and only part of the population receives them)	• Game AI • Complex decision problems • Reward systems

Copyright © SAS Institute Inc. All rights reserved.

sas

Clustering

- Clustering: Intuitively, finding clusters of points in the given data such that similar points lie in the same cluster
- Can be formalized using distance metrics in several ways
  - Group points into *k* sets (for a given *k*) such that the average distance of points from the centroid of their assigned group is minimized
    - Centroid: point defined by taking average of coordinates in each dimension.
  - Another metric: minimize average distance between every pair of points in a cluster
- Has been studied extensively in statistics, but on small data sets
  - Data mining systems aim at clustering techniques that can handle very large data sets
  - E.g., the Birch clustering algorithm (more shortly)

Clustering

- What is clustering?

Clustering

- Definition
  - Assignment of a set of observations into subsets so that observations in the same subset are similar in some sense
- Hard vs. Soft
  - Hard: same object can only belong to single cluster
  - Soft: same object can belong to different clusters
- Flat vs. Hierarchical
  - Flat: clusters are flat
  - Hierarchical: clusters form a tree
    - Agglomerative
    - Divisive

Similarity measures

- How to determine similarity between data points
  - using various distance metrics
- Let **x** = $(x_1,\ldots,x_n)$ and **y** = $(y_1,\ldots y_n)$ be n-dimensional vectors of data points of objects g_1 and g_2
  - g_1, g_2 can be two different genes in microarray data
  - n can be the number of samples

Summary of similarity measures

- Using different measures for clustering can yield different clusters
- Euclidean distance and correlation distance are the most common choices of similarity measure for microarray data
- Euclidean vs Correlation Example
  - g1 = (1,2,3,4,5)
  - g2 = (100,200,300,400,500)
  - g3 = (5,4,3,2,1)
  - Which genes are similar according to the two different measures?

K-MEANS CLUSTERING

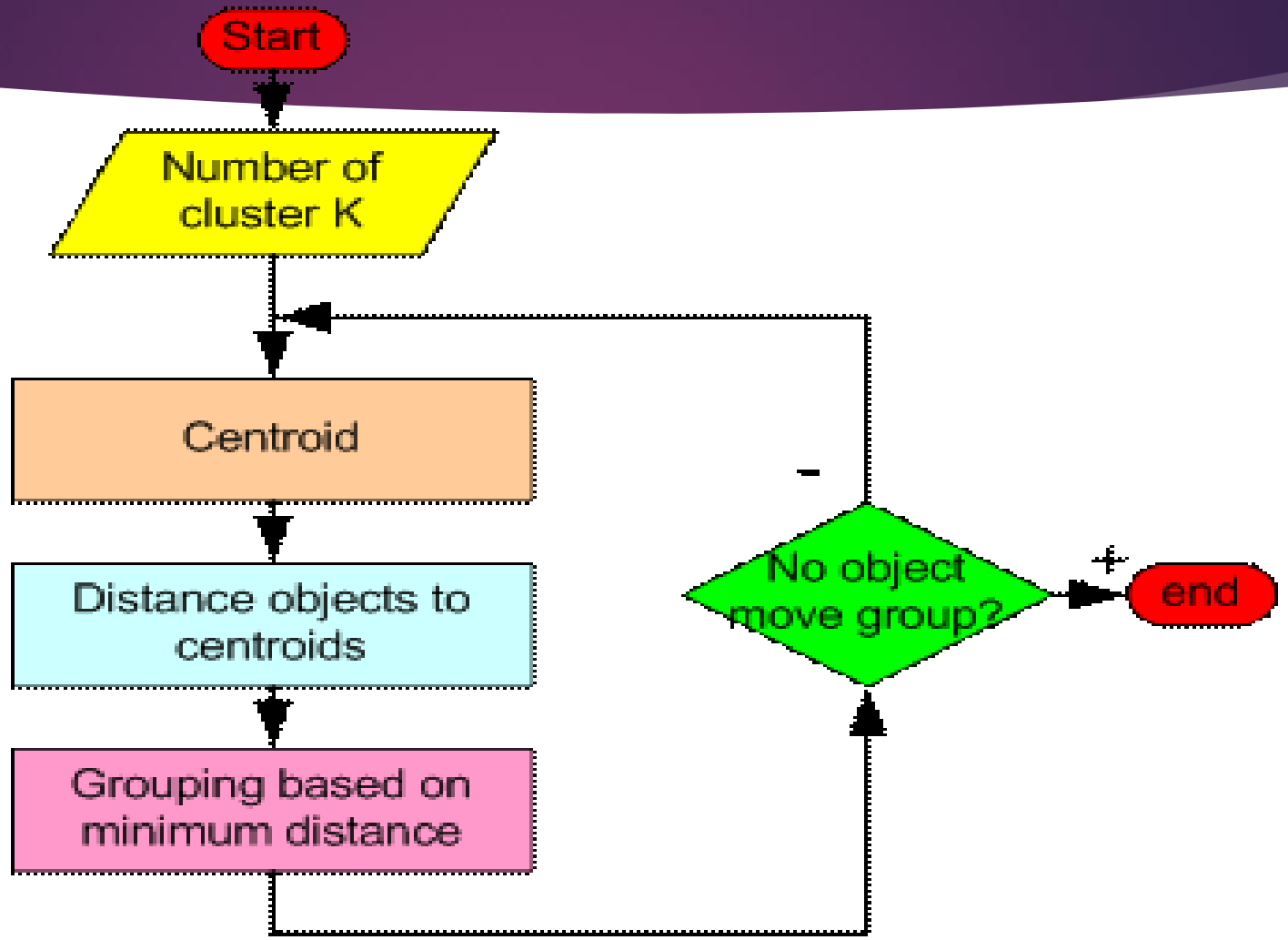
- The **k-means algorithm** is an algorithm to cluster *n* objects based on attributes into *k* partitions, where $k < n$.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.
- It assumes that the object attributes form a vector space.

- An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2,$$

where x_n is a vector representing the the n^{th} data point and u_j is the geometric centroid of the data points in S_j.

How the K-Mean Clustering algorithm works?



- **Step 1:** Begin with a decision on the value of k = number of clusters .

- **Step 2**:

Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

1.Take the first k training sample as single-element clusters

2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

- **Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters.

  If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

A Simple example showing the implementation of k-means algorithm (using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.

- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

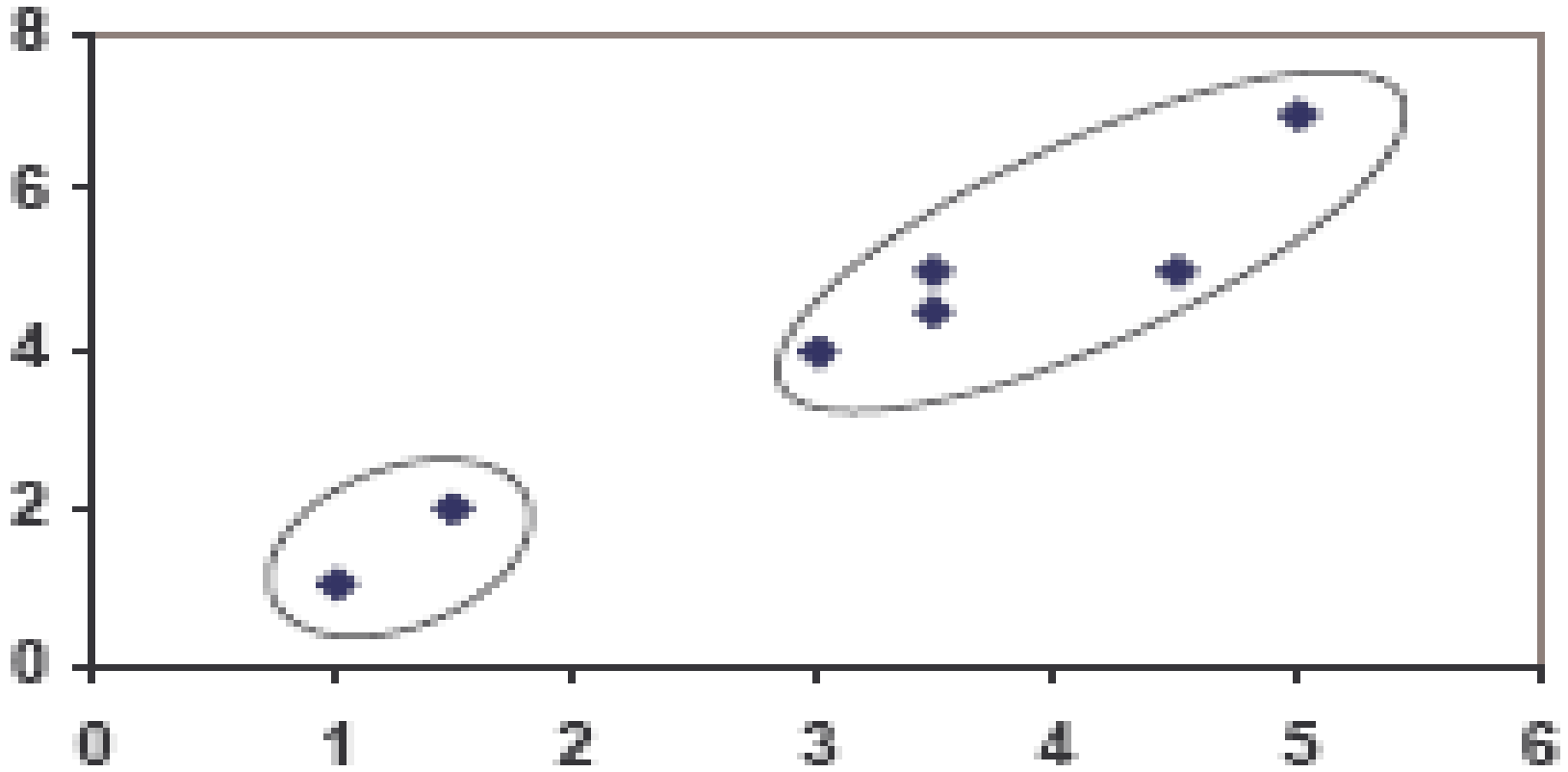
  The clusters obtained are:

  {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.86	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT

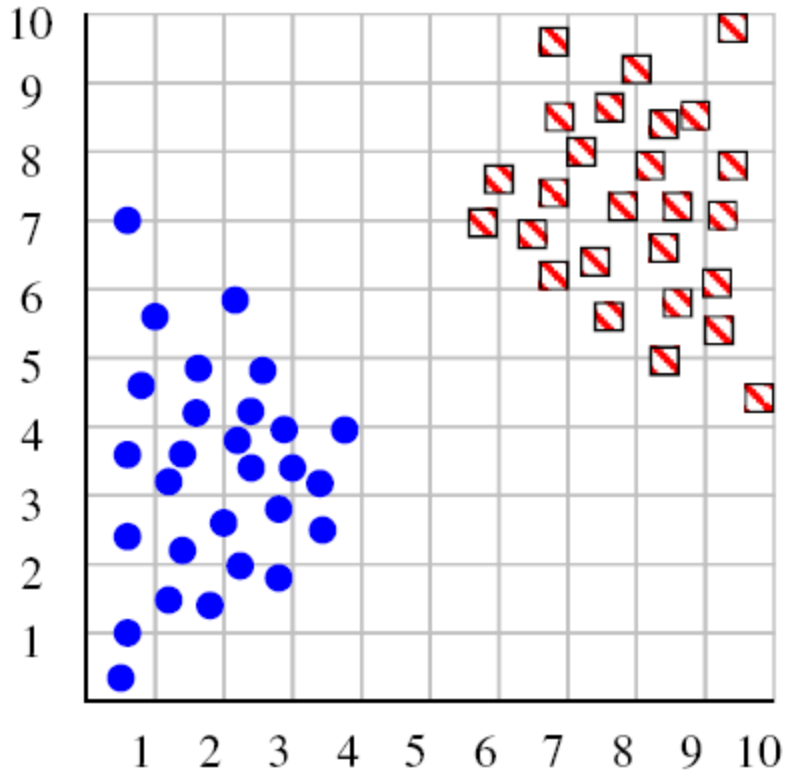


K-means: summary

- Algorithmically, very simple to implement
- *K*-means converges, but it finds a local minimum of the cost function
- Works only for numerical observations
- *K* is a user input; alternatively BIC (Bayesian information criterion) or MDL (minimum description length) can be used to estimate *K*
- Outliers can considerable trouble to *K*-means

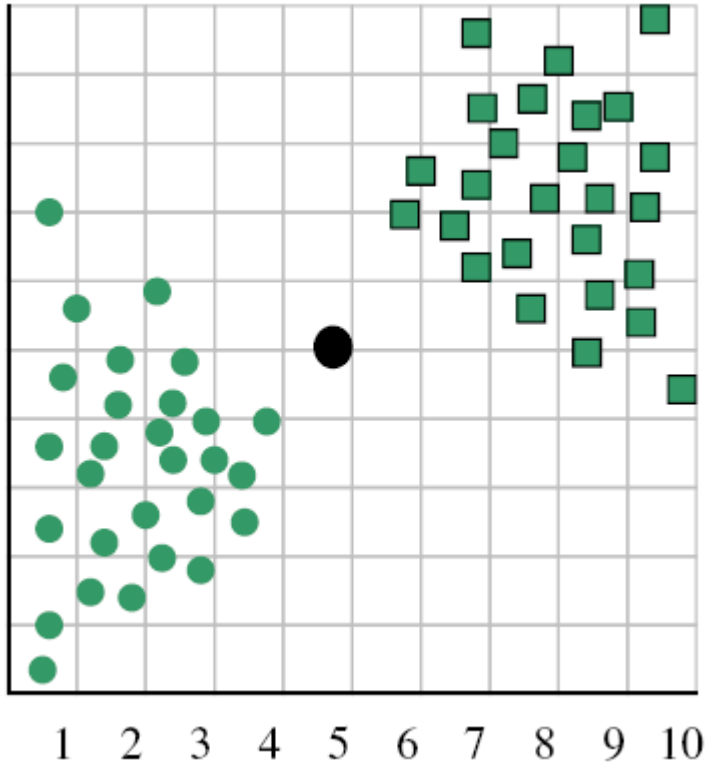
Deciding K

- Try a couple of K



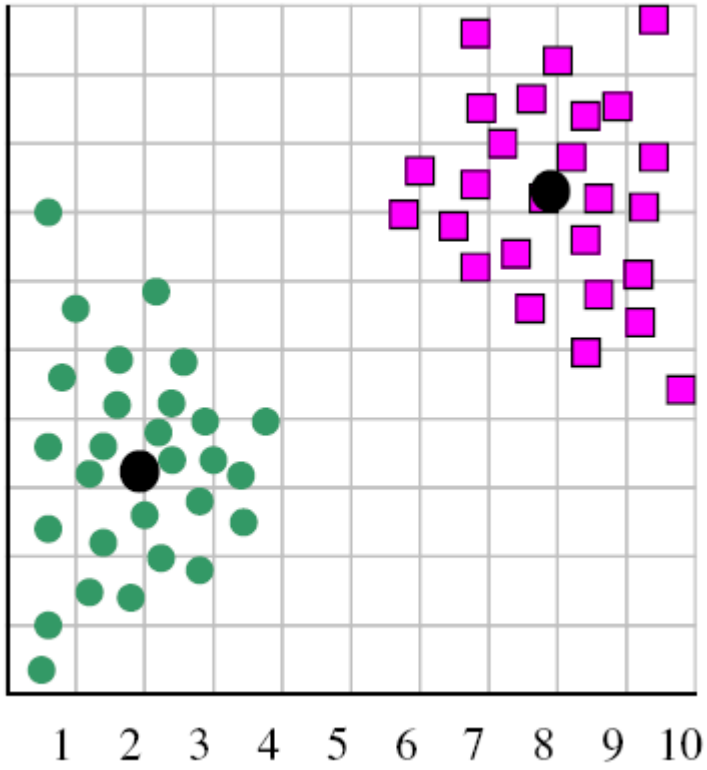
Deciding K

- When k = 1, the objective function is 873.0



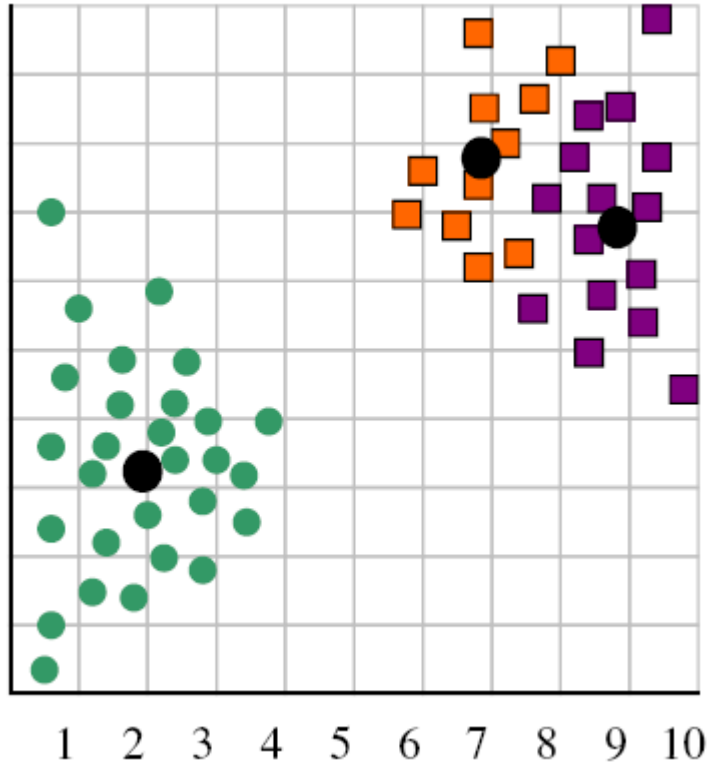
Deciding K

- When k = 2, the objective function is 173.1



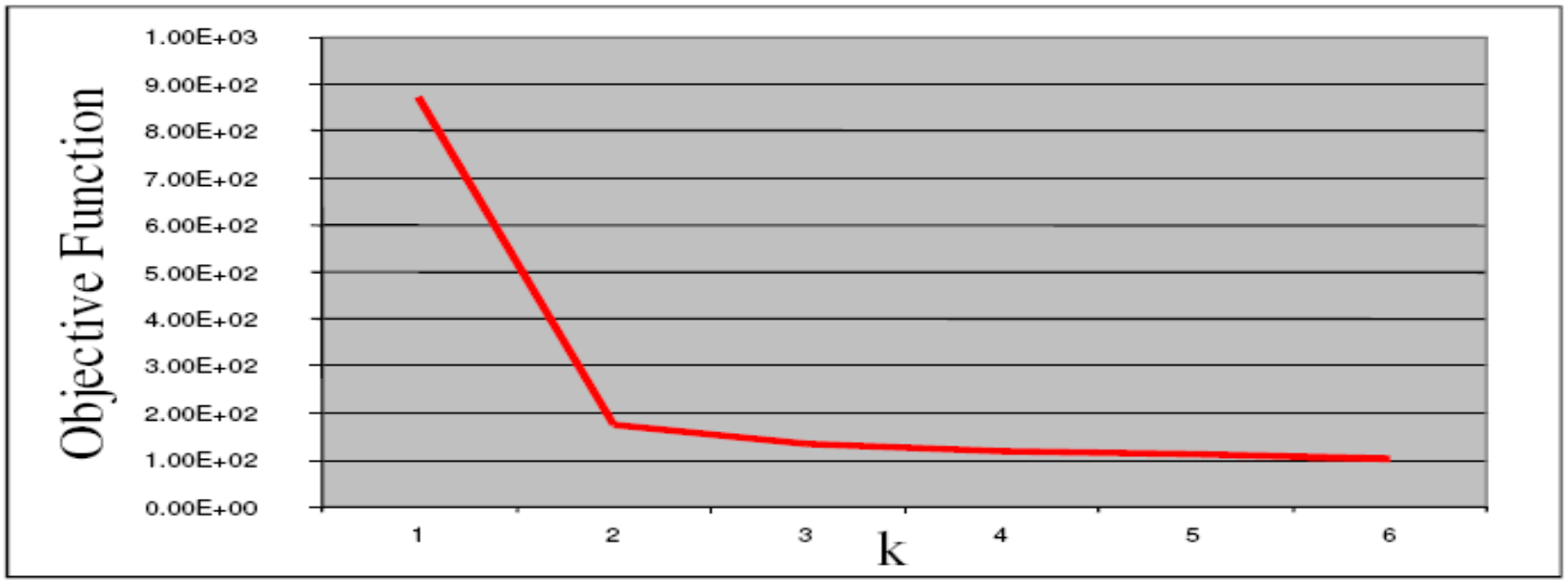
Deciding K

- When k = 3, the objective function is 133.6



Deciding K

- We can plot objective function values for k=1 to 6
- The abrupt change at k=2 is highly suggestive of two clusters
- "knee finding" or "elbow finding"
- Note that the results are not always as clear cut as in this toy example



Back

Weaknesses of K-Means Clustering

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

Applications of K-Means Clustering

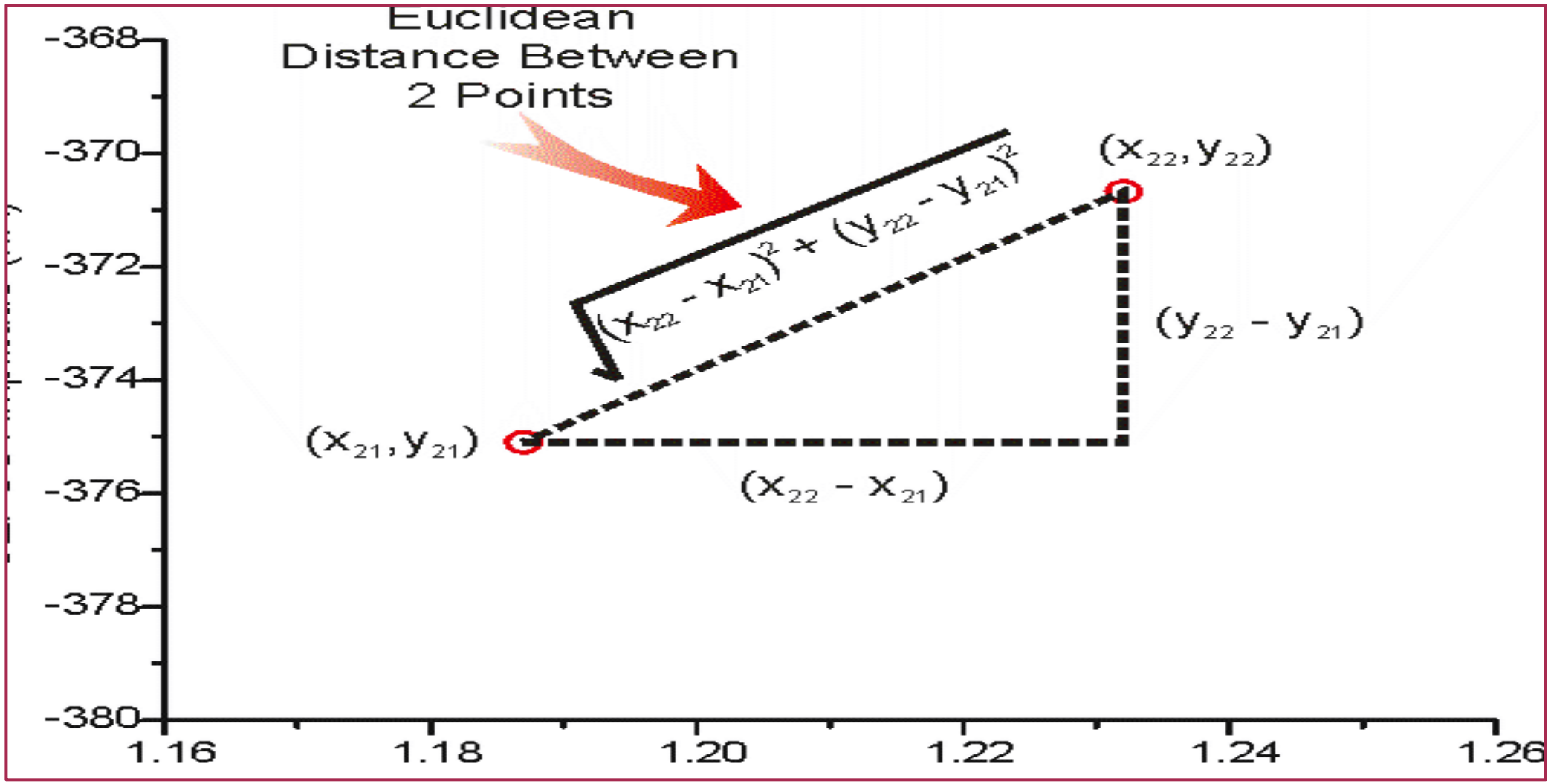
- It is relatively *efficient and fast*. It computes result at **O(tkn)**, where n is number of objects or points, k is number of clusters and t is number of iterations.
- k-means clustering can be applied to *machine learning or data mining*
- *Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation).*
- *Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.*

Geometric Distance Measure

- Geometric distance metrics, primarily, tends to measure the similarity between two or more vectors solely based on the distance between two points in multi-dimensional space.
- The examples of such type of geometric distance measures are Minkowski distance, Euclidean distance and Manhattan distance.
- Minkowski distance is the general form of Euclidean and Manhattan distance. Mathematically, it can be represented as the following:

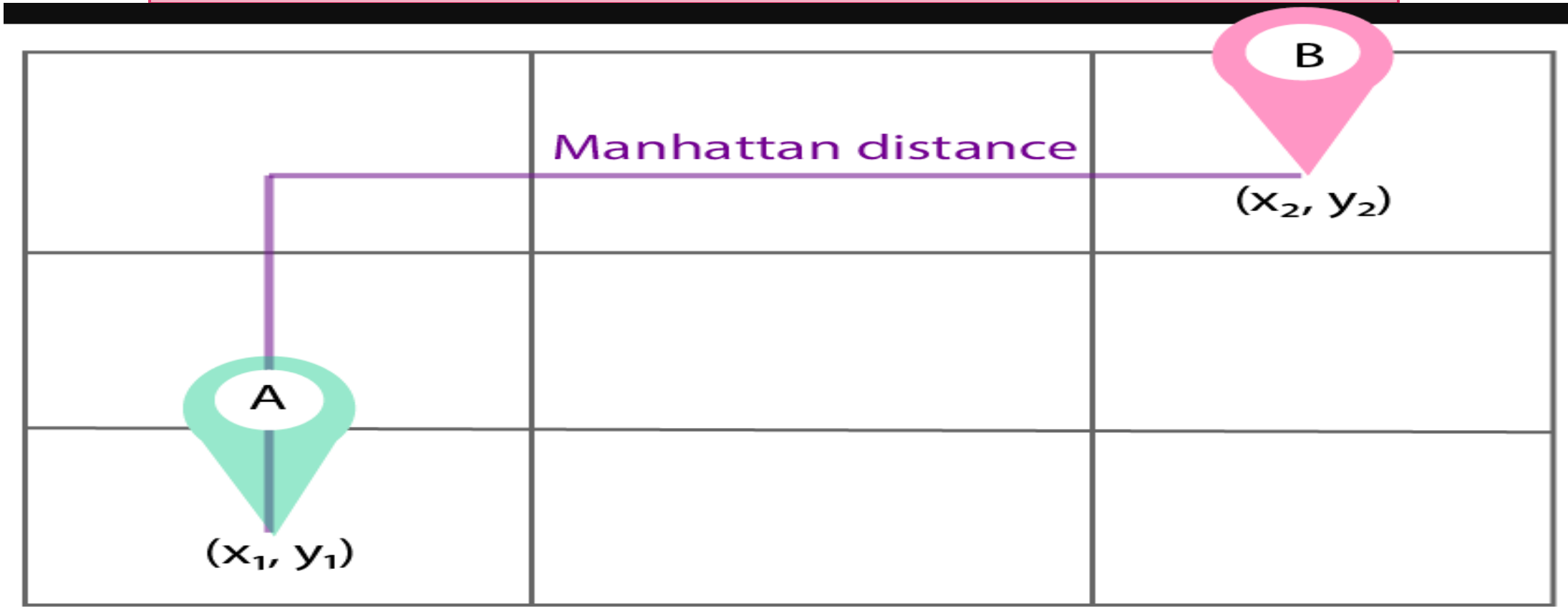
$$D(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^{1/P} \right)^{P}$$

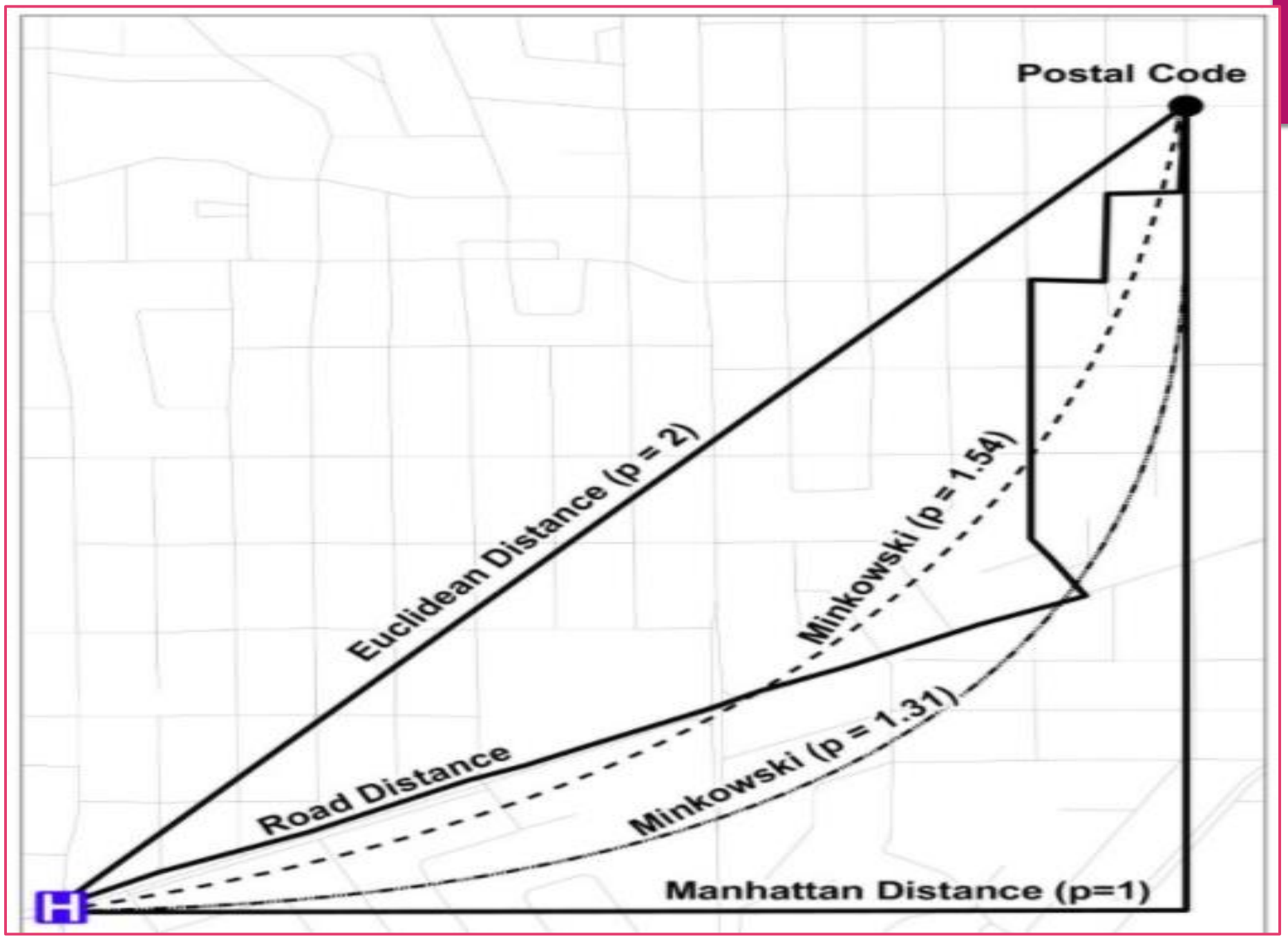
$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$



Manhattan Distance

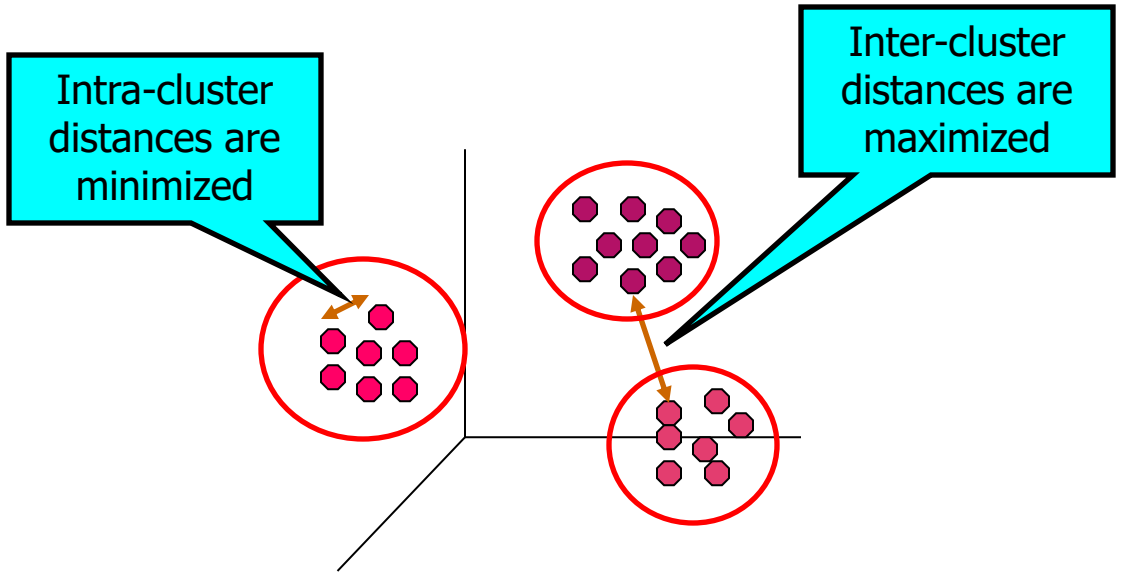
- If you want to find Manhattan distance between two different points (x1, y1) and (x2, y2) such as the following, it would look like the following:
- Manhattan distance = (x2 – x1) + (y2 – y1)




Postal Code
Euclidean Distance (p = 2)
Minkowski (p = 1.54)
Road Distance
Minkowski (p = 1.31)
Manhattan Distance (p=1)
H

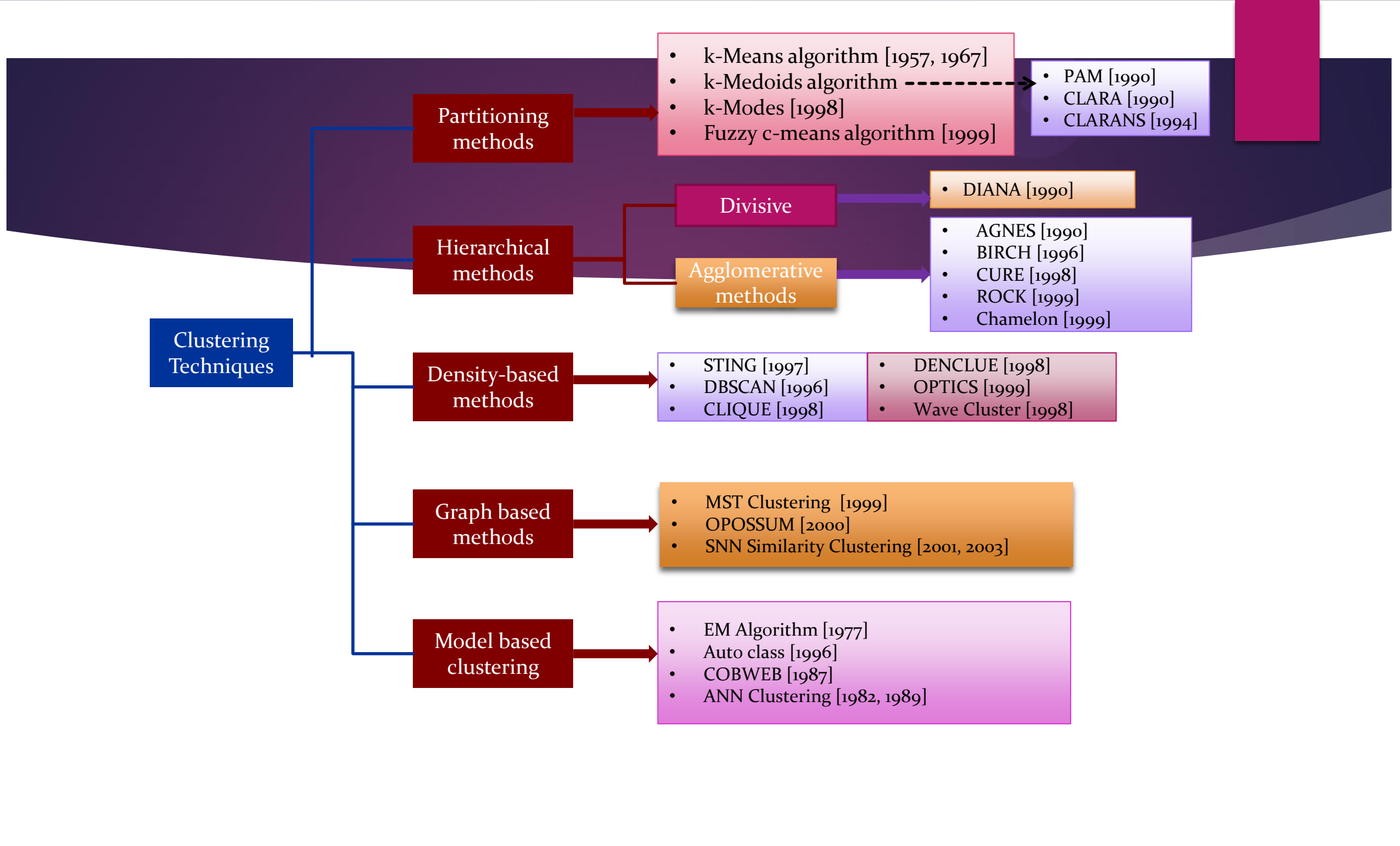
What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns


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]

K-MEDOIDS CLUSTERING

The mean in k-means clustering is sensitive to outliers. Since an object with an extremely high value may substantially distort the distribution of data.

Hence we move to k-medoids.

Instead of taking mean of cluster we take the most centrally located point in cluster as it's center.

These are called medoids.

k-Medoids

- It is also called as Partitioning Around Medoid algorithm.
- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.
- The dissimilarity of the medoid(Ci) and object(Pi) is calculated by using $E = |Pi - Ci|$

Algorithm:

1. Initialize: select k random points out of the n data points as the medoids.
2. Associate each data point to the closest medoid by using any common distance metric methods.
3. While the cost decreases:

For each medoid m, for each data o point which is not a medoid:

1. Swap m and o, associate each data point to the closest medoid, recompute the cost.
2. If the total cost is more than that in the previous step, undo the swap.

K-MEDOIDS - PAM ALGORITHM

PAM stands for **Partitioning Around Medoids**.

GOAL: To find Clusters that have minimum average dissimilarity between objects that belong to same cluster.

ALGORITHM:

1. Start with initial set of medoids.
2. Iteratively replace one of the medoids with a non-medoid if it reduces total sum of SSE of resulting cluster.

SSE is calculated as below:

$$SSE(X) = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

Where k is number of clusters and x is a data point in cluster C_i and M_i is medoid of C_i

ADVANTAGES and DISADVANTAGES of PAM

Advantages:

PAM is more flexible as it can use any similarity measure.

PAM is more robust than k-means as it handles noise better.

Disadvantages:

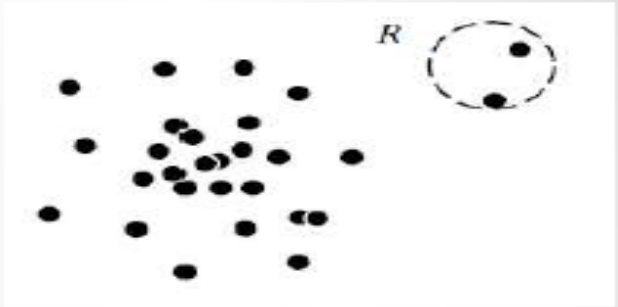
PAM algorithm for K-medoid clustering works well for dataset but cannot scale well for large data set due to high computational overhead.

PAM COMPLEXITY : $O(k(n-k)^2)$ this is because we compute distance of n-k points with each k point, to decide in which cluster it will fall and after this we try to replace each of the medoid with a non medoid and find it's distance with n-k points.

Outlier Detection in Clustering

- An **outlier** is a data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism.
- Outliers are referred as "abnormal" data.
- Outliers are different from noisy data. Noise is a random error or variance in a measured variable.
- Outliers are interesting because they are suspected of not being generated by the same mechanisms as the rest of the data.
- Outlier detection is also related to novelty detection in evolving data sets.

The objects in region R are outliers



Types of Outliers

- ✓ Global Outliers
- ✓ Contextual Outliers
- ✓ Collective Outliers

Outlier detection techniques.

Supervised, Semi-Supervised, and Unsupervised Methods

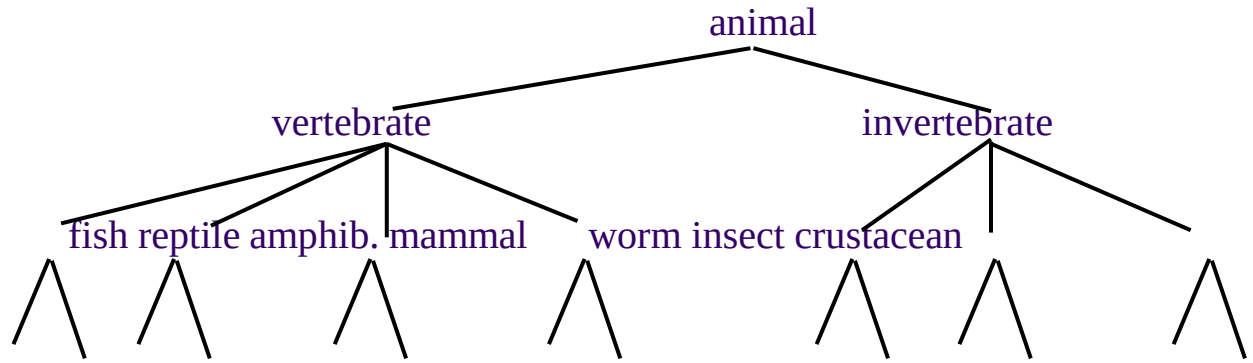
- Supervised methods model data normality and abnormality.
- In some application scenarios, objects labeled as "normal" or "outlier" are not available. Thus, an unsupervised learning method has to be used.
- In some cases where only a small set of the normal and/or outlier objects are labeled, but most of the data are unlabeled.

Statistical Methods, Proximity-Based Methods and Clustering-Based Methods

- Statistical methods (also known as model-based methods) make assumptions of data normality.
- The effectiveness of proximity-based methods relies heavily on the proximity (or distance) measure used.
- Clustering-based methods assume that the normal data objects belong to large and dense clusters, whereas outliers belong to small or sparse clusters, or do not belong to any clusters.

Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.

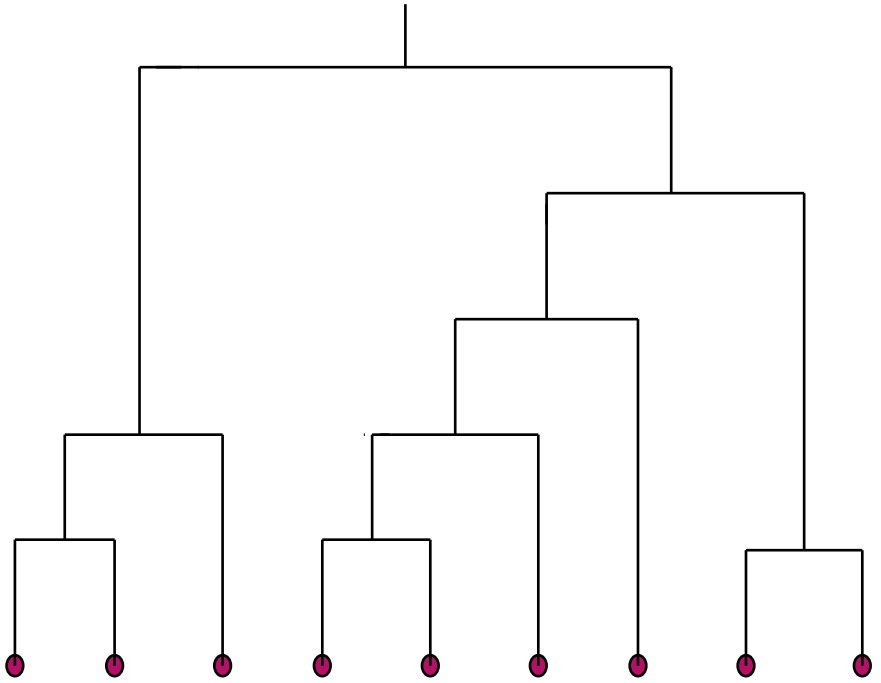


- One approach: recursive application of a partitional clustering algorithm.

Dendrogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.

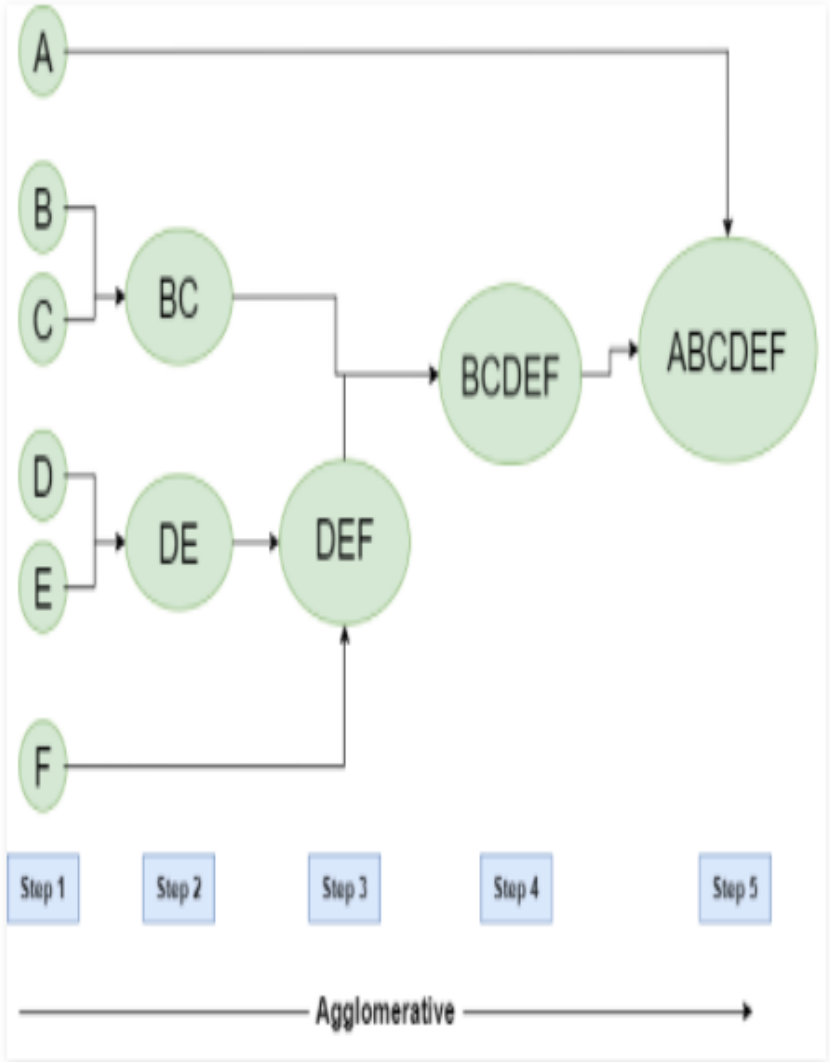
- dendrogram

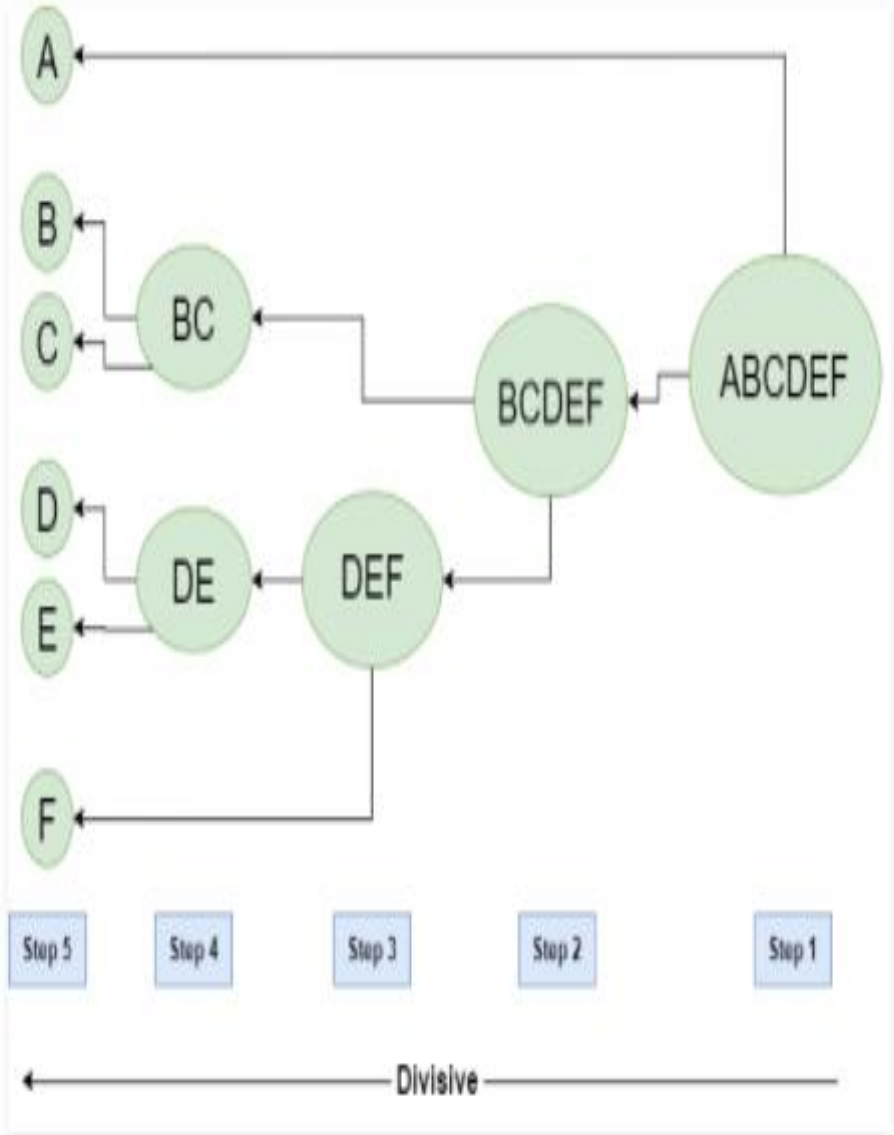


Hierarchical Agglomerative Clustering (HAC)

- Starts with each doc in a separate cluster
  - then repeatedly joins the *closest pair* of clusters, until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

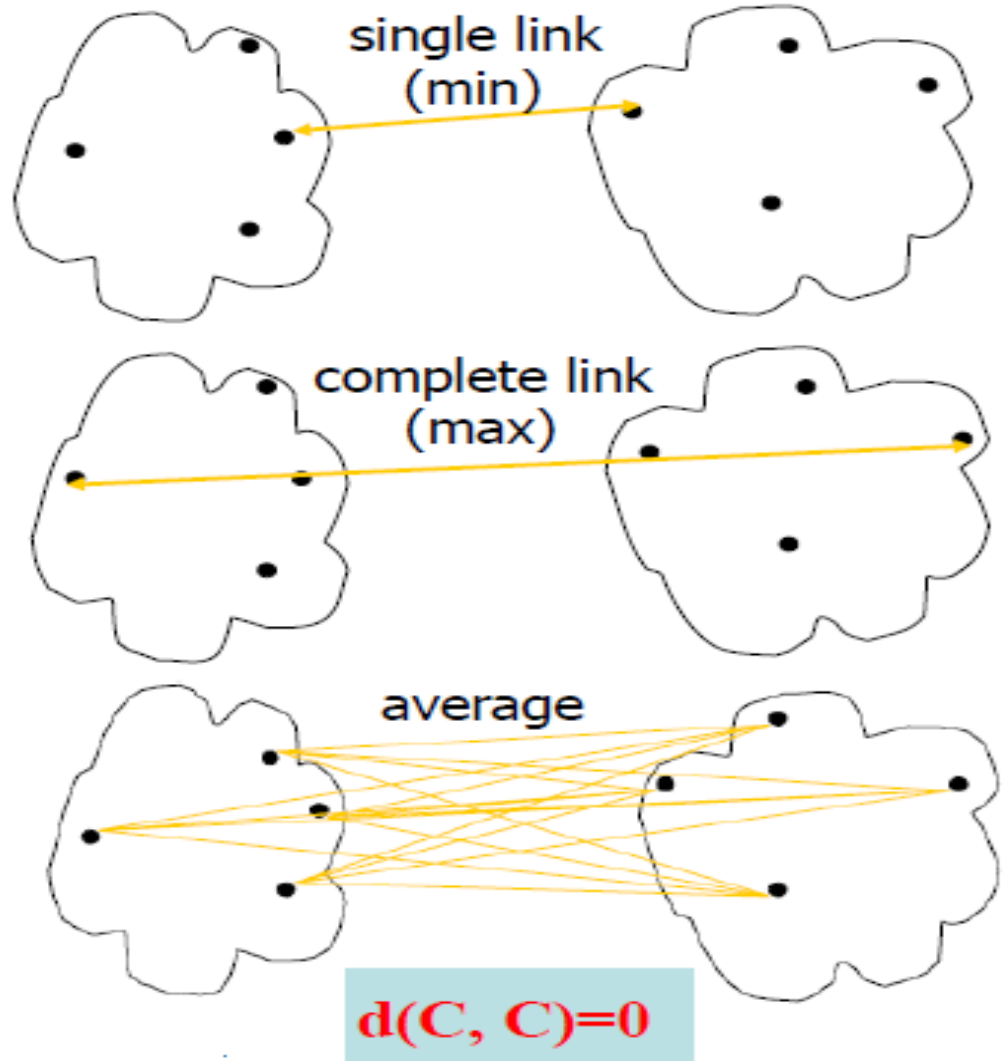
Let's say we have six data points **A, B, C, D, E, F**.





Cluster Distance Measures

- **Single link**: smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$
- **Complete link**: largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$
- **Average**: avg distance between elements in one cluster and elements in the other, i.e., $d(C_i, C_j) = \text{avg}\{d(x_{ip}, x_{jq})\}$



Cluster Distance Measures

Example: Given a data set of five objects characterised by a single continuous feature, assume that there are two clusters: C_1: {a, b} and C_2: {c, d, e}. (Minkowski distance for distance matrix)

	a	b	c	d	e
Feature	1	2	4	5	6

1. Calculate the distance matrix .

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

2. Calculate three cluster distances between C_1 and C_2.

Single link

$$\begin{aligned}\text{dist}(C_1, C_2) &= \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \min\{3, 4, 5, 2, 3, 4\} = 2\end{aligned}$$

Complete link

$$\begin{aligned}\text{dist}(C_1, C_2) &= \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \max\{3, 4, 5, 2, 3, 4\} = 5\end{aligned}$$

Average

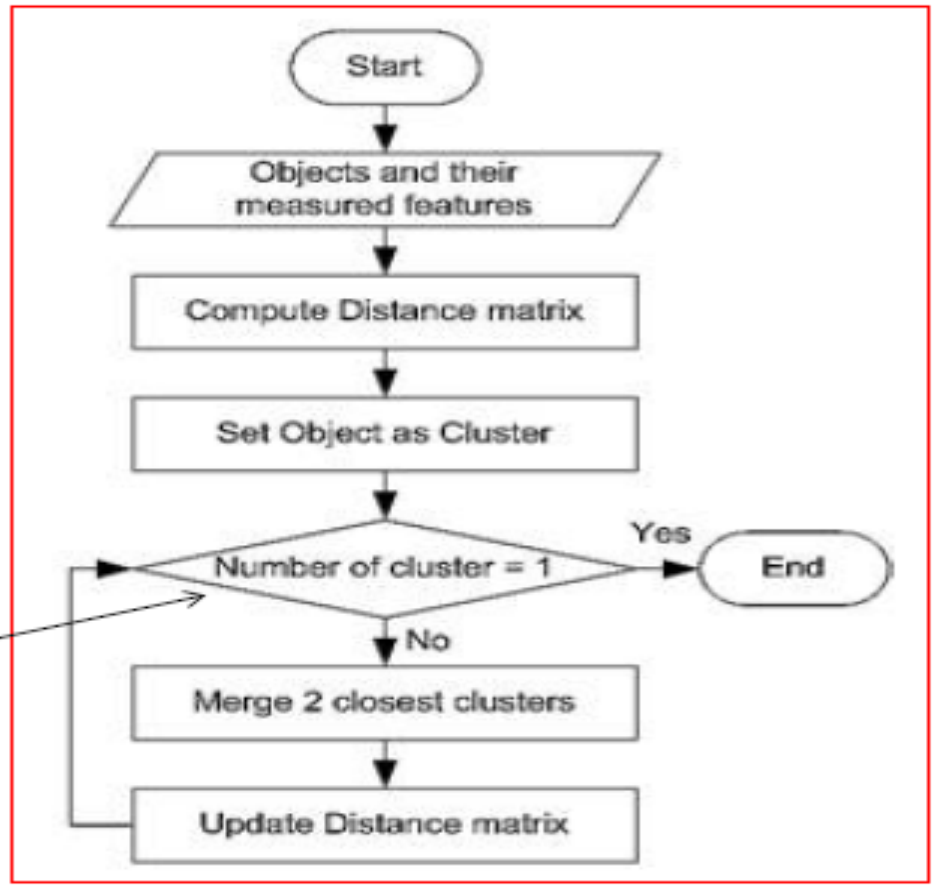
$$\text{dist}(C_1, C_2) = \frac{d(a,c) + d(a,d) + d(a,e) + d(b,c) + d(b,d) + d(b,e)}{6}$$

$$= \frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5$$

Agglomerative Algorithm

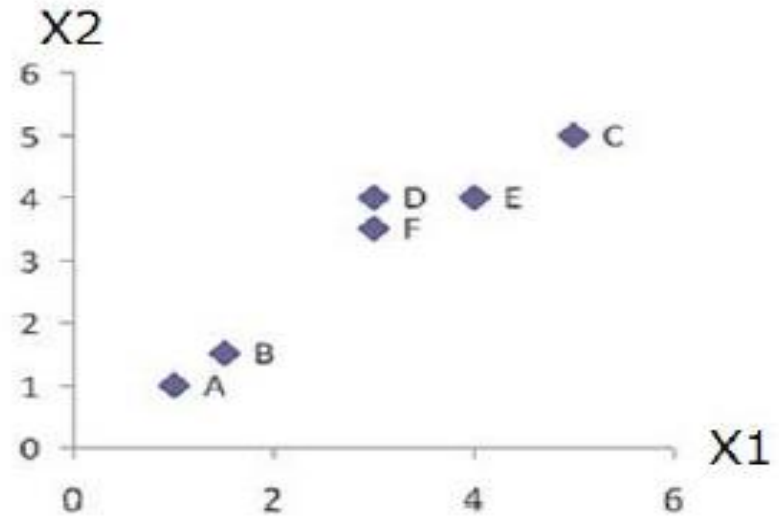
The *Agglomerative* algorithm is carried out in three steps:

1) Convert all object features into a distance matrix
2) Set each object as a cluster (thus if we have *N* objects, we will have *N* clusters at the beginning)
3) Repeat until number of cluster is one (or known # of clusters)
   - Merge two closest clusters
   - Update "distance matrix"



Example

Problem: clustering analysis with agglomerative algorithm



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

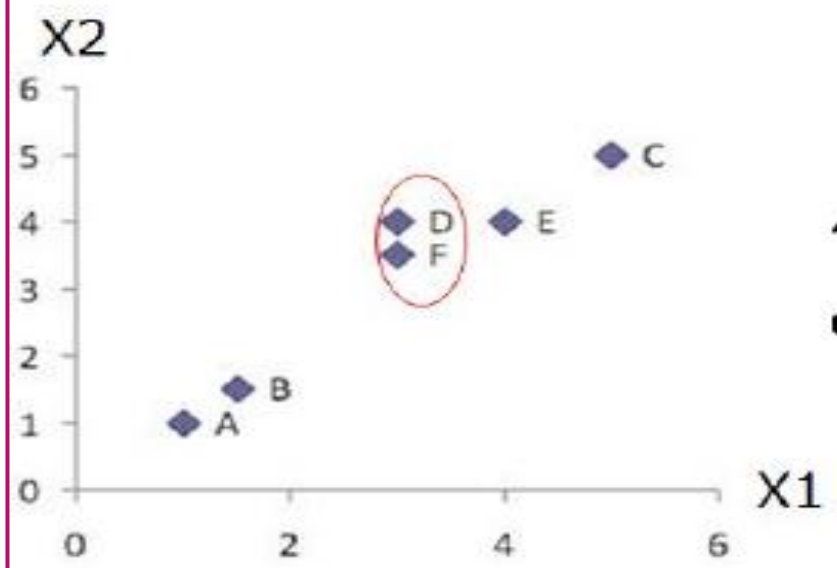
Euclidean distance

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

Example

Merge two closest clusters (iteration 1)



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

- Update distance matrix (iteration 1)

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\to A} = \min(d_{DA}, d_{FA}) = \min(3.61, 3.20) = 3.20$$

$$d_{(D,F)\to B} = \min(d_{DB}, d_{FB}) = \min(2.92, 2.50) = 2.50$$

$$d_{(D,F)\to C} = \min(d_{DC}, d_{FC}) = \min(2.24, 2.50) = 2.24$$

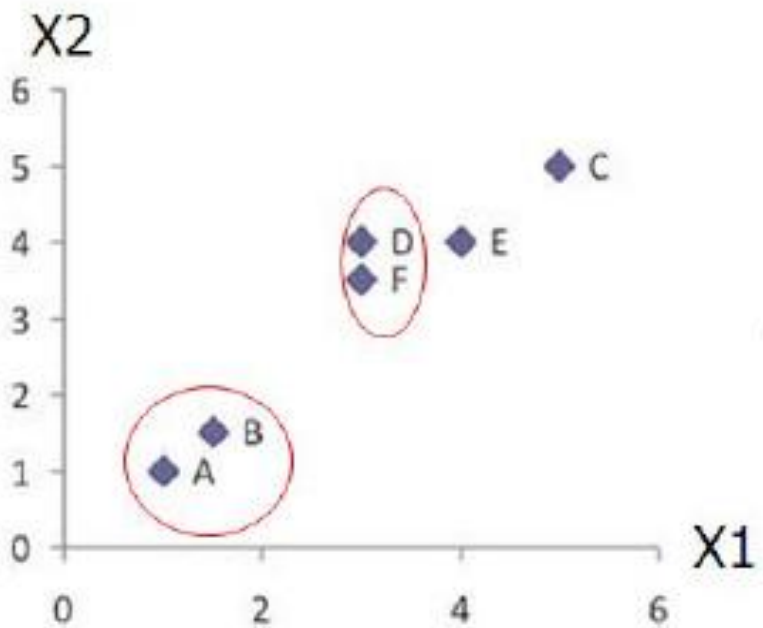
$$d_{E\to(D,F)} = \min(d_{ED}, d_{EF}) = \min(1.00, 1.12) = 1.00$$

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	**1.00**
E	4.24	3.54	1.41	**1.00**	0.00

- Merge two closest clusters (iteration 2)



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

- Update distance matrix (iteration 2)

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

$$d_{(D,F)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

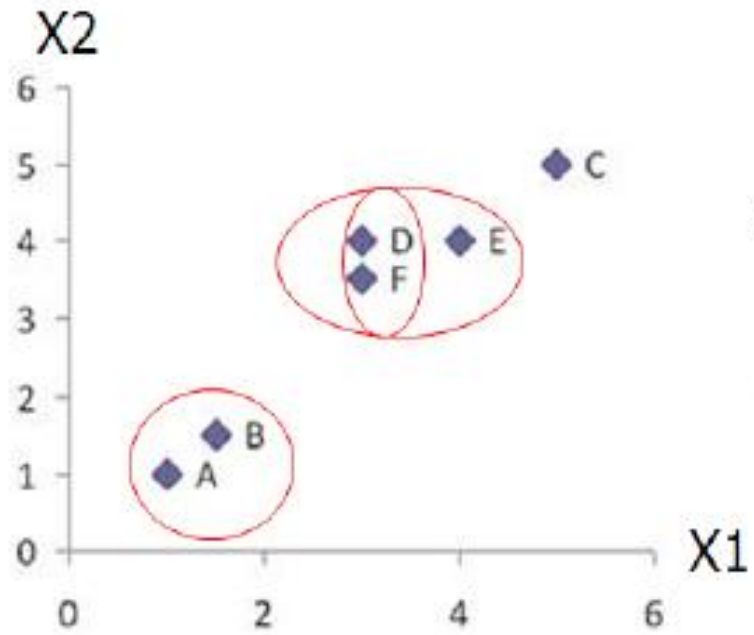
$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

- Merge two closest clusters/update distance matrix (iteration 3)



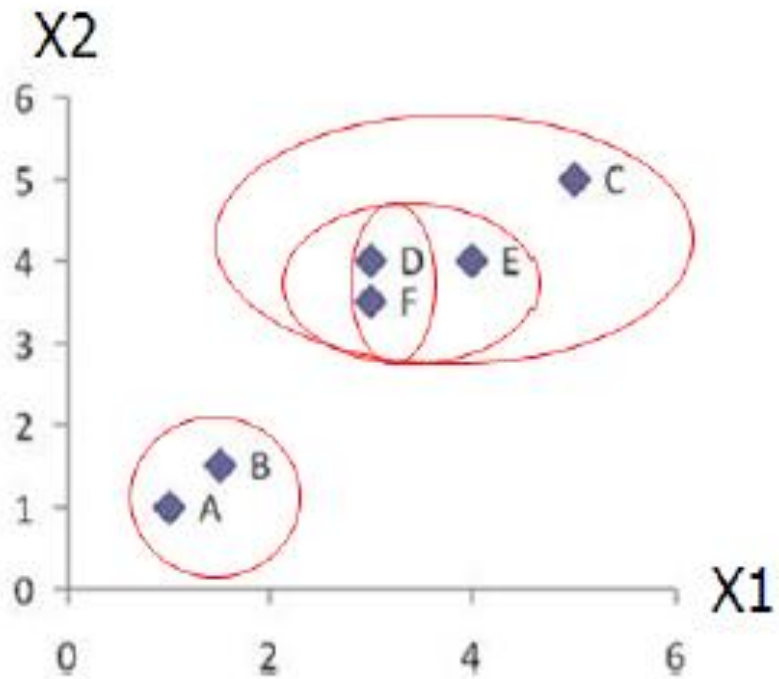
Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

- Merge two closest clusters/update distance matrix (iteration 4)



Min Distance (Single Linkage)

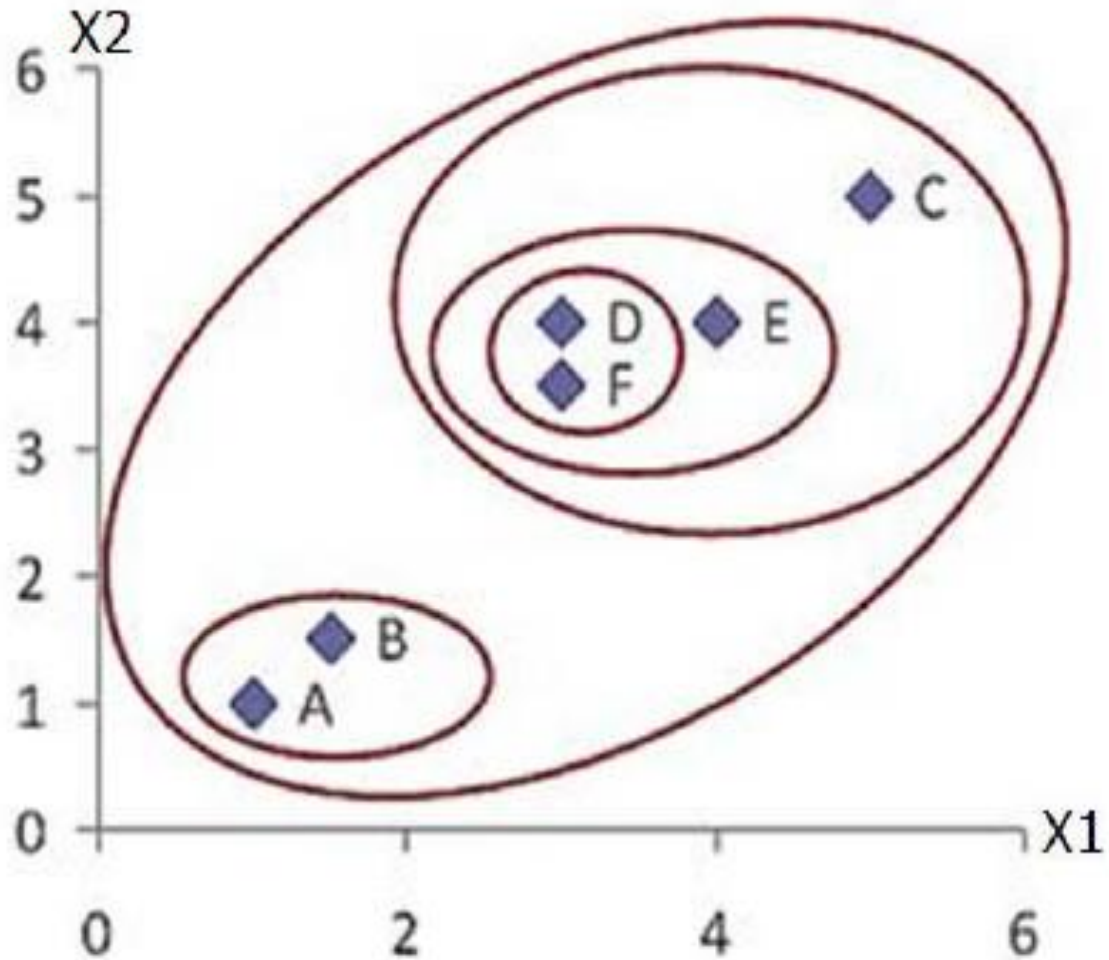
Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Min Distance (Single Linkage)

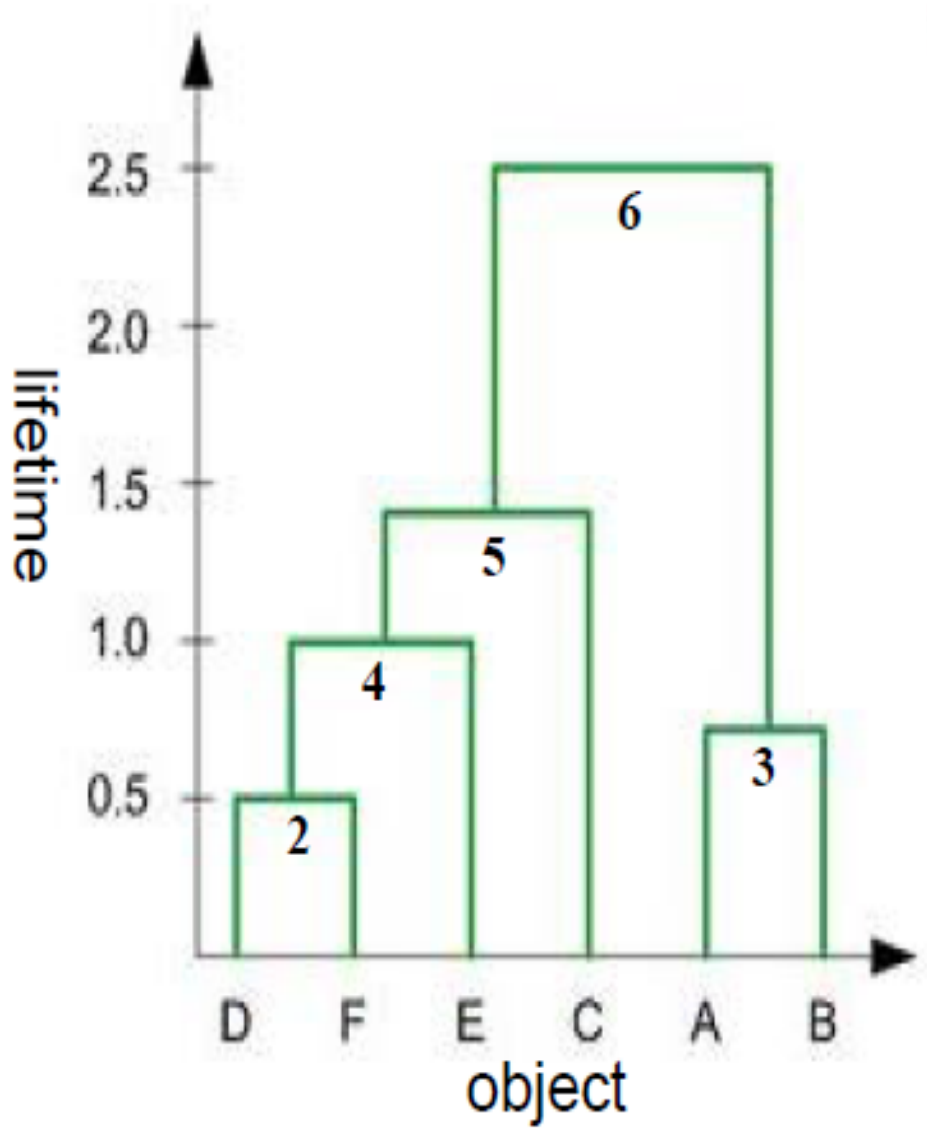
Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

- Final result (meeting termination condition)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



- Dendrogram tree representation



1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge clusters D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
5. We merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge clusters (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation

It will be Continued….