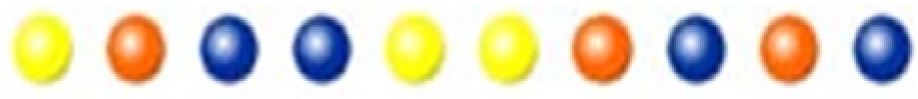
CLUSTERING – PART I: Partitioning Methods
Lecture 2

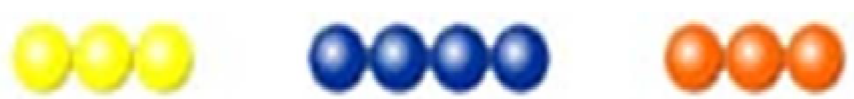
Dr. Vassilis S. Kodogiannis
Reader in Computational Intelligence
Email: V.Kodogiannis@westminster.ac.uk
https://scholar.google.co.uk/citations?user=meTTcLAAAAAJ&hl=en&oi=ao

What is clustering?

- Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data.

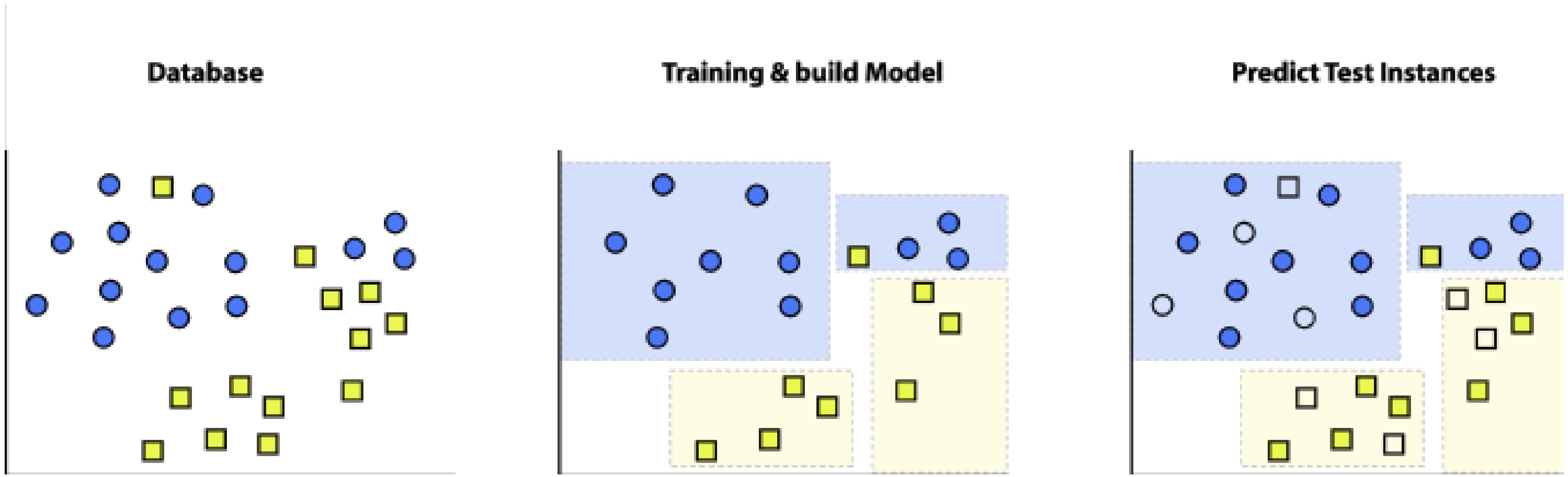


- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters



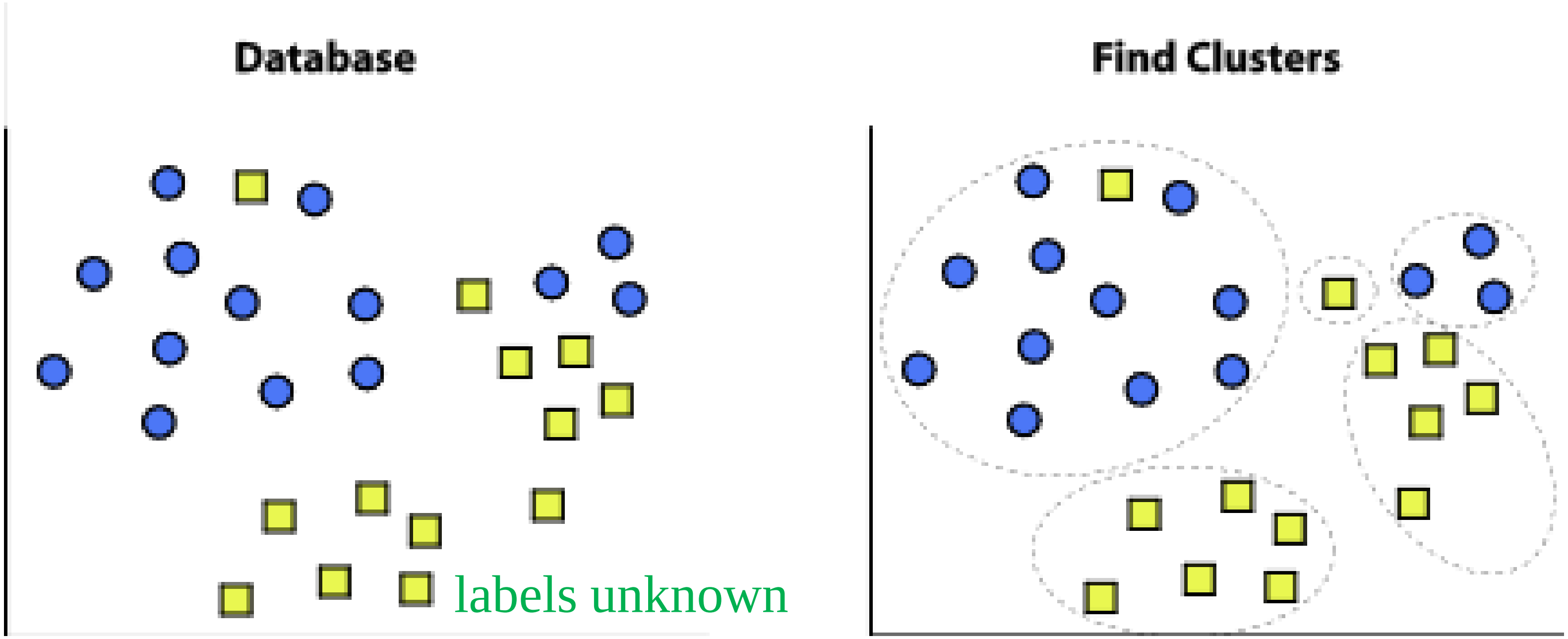
- Clustering is unsupervised classification: no predefined classes

Classification vs. Clustering



Classification: Supervised learning

Classification vs. Clustering



labels unknown

Clustering: Unsupervised learning

No labels, find “natural” grouping of instances

What is clustering?

- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

- Use cluster detection when you suspect that there are natural groupings that may represent groups of customers or products that have lot in common.

- When there are many competing patterns in the data, making it hard to spot a single pattern, creating clusters of similar records reduces the complexity within clusters so that other data mining techniques are more likely to succeed.

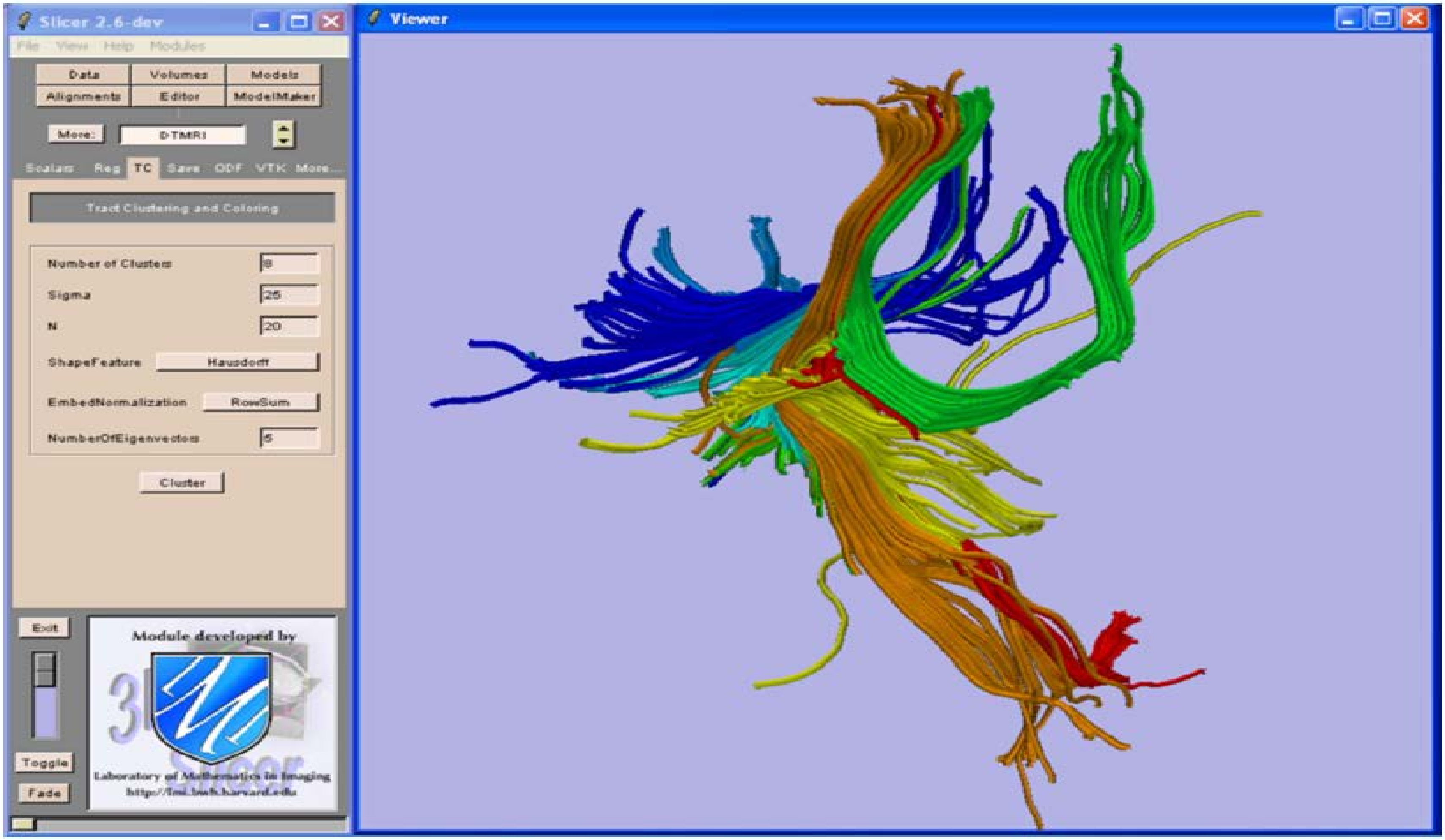
- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that:
  - data points in one cluster are more similar to one another (high intra-class similarity)
  - data points in separate clusters are less similar to one another (low inter-class similarity)

- Similarity measures: e.g. Euclidean distance if attributes are continuous.

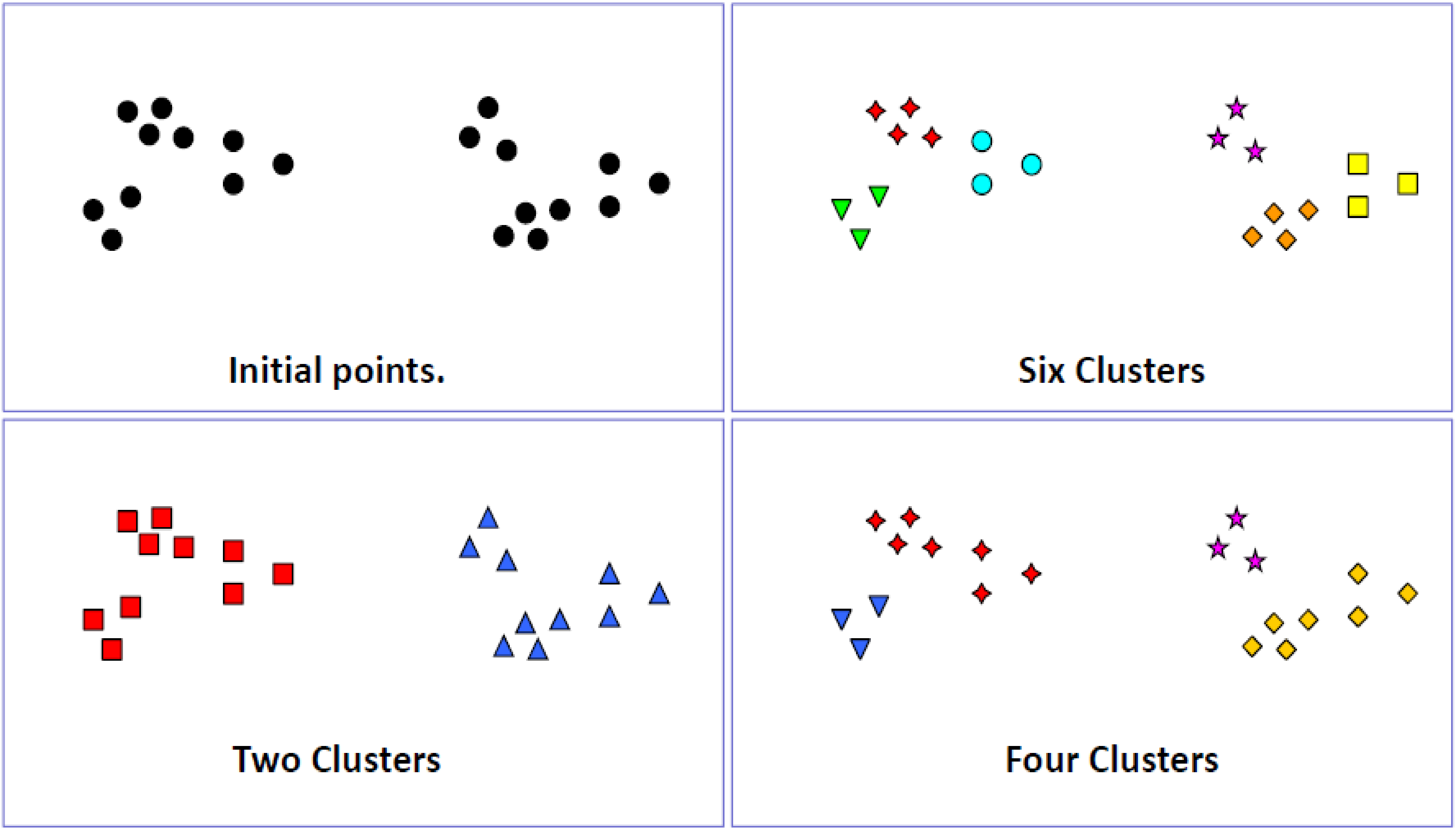
Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

http://wiki.na-mic.org/Wiki/index.php/Progress_Report:DTI_Clustering
Project aiming at developing tools in the 3D Slicer for automatic clustering of tractographic paths through diffusion tensor MRI (DTI) data.
'characterize the strength of connectivity between selected regions in the brain'

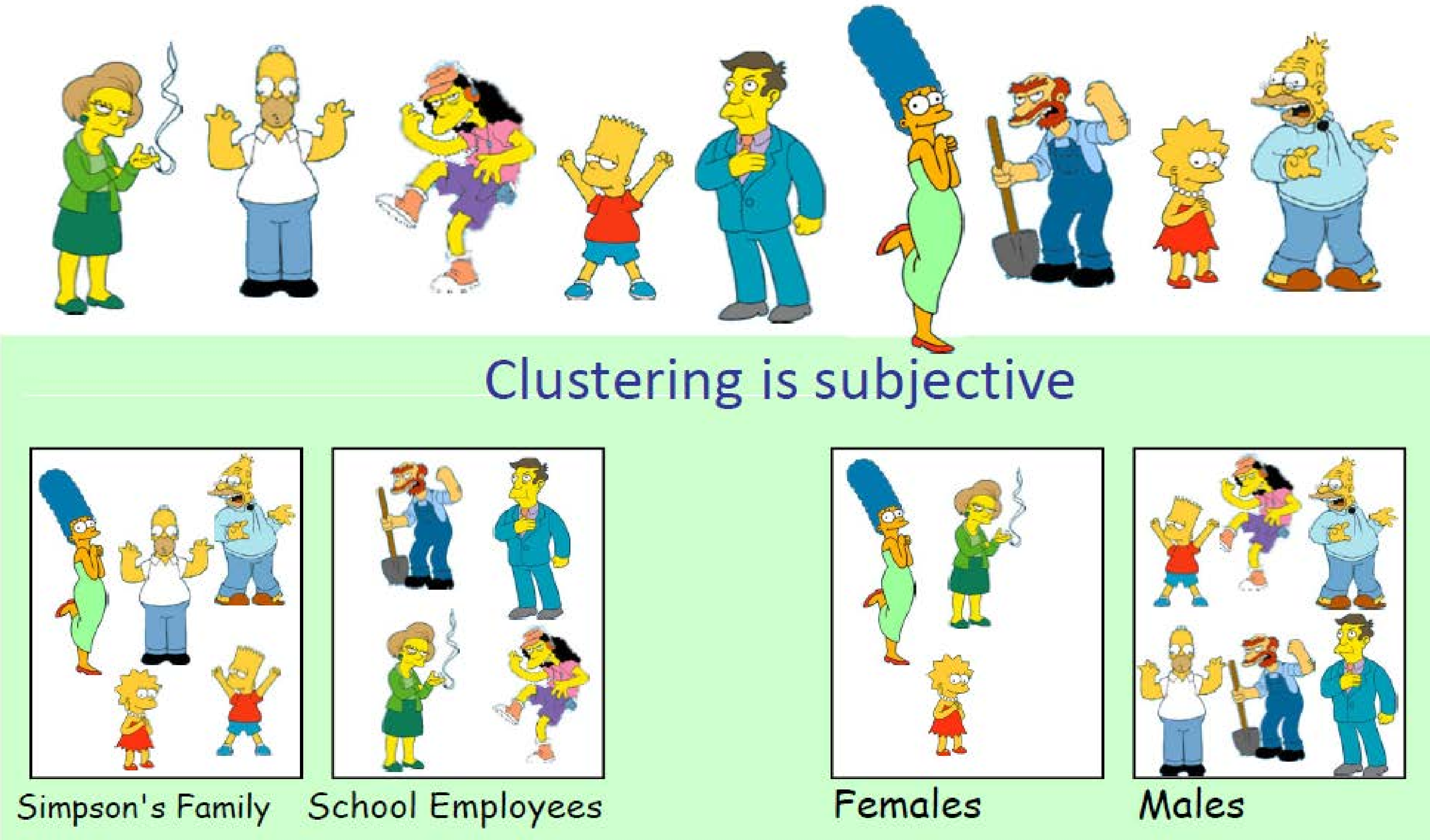


Notion of a Cluster is Ambiguous



Clustering is subjective

What is a natural grouping among these objects?



Types of Data in Cluster Analysis

Data Matrix

Represents n objects with p variables (attributes, measures)

Name	Energy	Protein	Fat	Calcium	Iron
Braised beef	340	20	28	9	2.6
Hamburger	245	21	17	9	2.7
Roast beef	420	15	39	7	2
Beefsteak	375	19	32	9	2.6
Canned beef	180	22	10	17	3.7
Broiled chicken	115	20	3	8	1.4
Canned chicken	170	25	7	12	1.5
Beef heart	160	26	5	14	5.9
Roast lamb leg	265	20	20	9	2.6
Roast lamb shoulder	300	18	25	9	2.3
Smoked ham	340	20	28	9	2.5
Pork roast	340	19	29	9	2.5
Pork simmered	355	19	30	9	2.4
Beef tongue	205	18	14	7	2.5
Veal cutlet	185	23	9	9	2.7
Baked bluefish	135	22	4	25	0.6
Raw clams	70	11	1	82	6
Canned clams	45	7	1	74	5.4
Canned crabmeat	90	14	2	38	0.8
Fried haddock	135	16	5	15	0.5
Broiled mackerel	200	19	13	5	1
Canned mackerel	155	16	9	157	1.8
Fried perch	195	16	11	14	1.3
Canned salmon	120	17	5	159	0.7
Canned sardines	180	22	9	367	2.5
Canned tuna	170	25	7	7	1.2
Canned shrimp	110	23	1	98	2.6

Dissimilarity Matrix

- Proximities of pairs of objects
- d(i,j): dissimilarity between objects i and j
- Nonnegative
- Close to 0: similar

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1																											
2	0.27																										
3	0.31	0.42																									
4	0.21	0.31	0.27																								
5	0.37	0.27	0.52	0.41																							
6	0.41	0.33	0.52	0.46	0.32																						
7	0.41	0.32	0.52	0.45	0.28	0.26																					
8	0.50	0.40	0.65	0.54	0.30	0.39	0.32																				
9	0.24	0.20	0.38	0.28	0.30	0.34	0.34	0.43																			
10	0.22	0.26	0.32	0.25	0.37	0.39	0.39	0.50	0.23																		
11	0.17	0.27	0.31	0.21	0.37	0.41	0.41	0.51	0.24	0.22																	
12	0.18	0.29	0.30	0.20	0.39	0.42	0.42	0.52	0.25	0.22	0.18																
13	0.20	0.30	0.28	0.19	0.40	0.43	0.43	0.53	0.27	0.22	0.19	0.18															
14	0.31	0.23	0.41	0.33	0.27	0.31	0.31	0.40	0.24	0.26	0.31	0.30	0.32														
15	0.35	0.25	0.49	0.39	0.22	0.29	0.24	0.32	0.28	0.34	0.35	0.36	0.38	0.25													
16	0.45	0.35	0.56	0.49	0.31	0.23	0.25	0.38	0.38	0.43	0.45	0.46	0.47	0.34	0.29												
17	0.62	0.54	0.68	0.65	0.45	0.45	0.53	0.39	0.55	0.58	0.62	0.62	0.63	0.49	0.49	0.50											
18	0.65	0.56	0.70	0.67	0.48	0.47	0.55	0.44	0.58	0.61	0.65	0.64	0.66	0.51	0.52	0.52	0.23										
19	0.51	0.43	0.54	0.54	0.41	0.27	0.35	0.48	0.44	0.45	0.51	0.50	0.51	0.37	0.39	0.28	0.38	0.41									
20	0.46	0.38	0.50	0.48	0.36	0.25	0.30	0.43	0.39	0.40	0.46	0.45	0.46	0.32	0.34	0.23	0.45	0.48	0.24								
21	0.35	0.28	0.44	0.38	0.30	0.27	0.28	0.43	0.29	0.31	0.35	0.35	0.35	0.23	0.28	0.28	0.53	0.56	0.33	0.28							
22	0.46	0.38	0.50	0.48	0.36	0.33	0.34	0.46	0.39	0.40	0.46	0.45	0.46	0.32	0.34	0.35	0.45	0.48	0.33	0.30	0.32						
23	0.38	0.30	0.42	0.41	0.30	0.28	0.28	0.44	0.31	0.32	0.38	0.38	0.38	0.24	0.29	0.30	0.48	0.51	0.30	0.24	0.22	0.27					
24	0.52	0.44	0.58	0.54	0.42	0.29	0.36	0.49	0.45	0.46	0.52	0.51	0.52	0.38	0.40	0.29	0.46	0.49	0.28	0.25	0.33	0.24	0.32				
25	0.51	0.41	0.65	0.55	0.37	0.44	0.40	0.49	0.44	0.50	0.50	0.52	0.53	0.40	0.35	0.42	0.58	0.62	0.51	0.48	0.43	0.35	0.43	0.41			
26	0.42	0.33	0.53	0.46	0.29	0.26	0.18	0.33	0.35	0.40	0.42	0.43	0.44	0.31	0.25	0.25	0.54	0.56	0.35	0.29	0.27	0.35	0.28	0.35	0.41		
27	0.45	0.36	0.60	0.50	0.32	0.28	0.31	0.37	0.39	0.45	0.46	0.47	0.48	0.35	0.28	0.29	0.40	0.43	0.34	0.36	0.39	0.33	0.39	0.33	0.37	0.32	

Similarity/Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Euclidean distance is probably the most commonly chosen type of distance. It is the geometric distance in the multidimensional space:

$$d(i,j)=\sqrt{\sum_{k=1}^{p}(x_{ki}-x_{kj})^2}$$

- Required properties for a distance function
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$

- City-block (Manhattan) distance. This distance is simply the sum of differences across dimensions. In most cases, this distance measure yields results similar to the Euclidean distance. However, note that in this measure, the effect of single large differences (outliers) is dampened (since they are not squared).

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+...+|x_{ip}-x_{jp}|$$

- **Weighted distances**
- If we have some idea of the relative importance that should be assigned to each variable, then we can weight them and obtain a weighted distance measure.

$$d(i,j)=\sqrt{w_1(x_{i1}-x_{j1})^2+\cdots+w_p(x_{ip}-x_{jp})^2}$$

Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion

- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion

- Density-based: Based on connectivity and density functions. Able to find clusters of arbitrary shape. Continues growing a cluster as long as the density of points in the neighborhood exceeds a specified limit.

- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database D of n objects into a set of k clusters

- Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: k-means and k-medoids algorithms
    - k-means: Each cluster is represented by the center of the cluster
    - k-medoids or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

K-means: Introduction

- Partitioning Clustering Approach
  - a typical clustering analysis approach via iteratively partitioning training data set to learn a partition of the given data space
  - learning a partition on a data set to produce several non-empty clusters (usually, the number of clusters given in advance)
  - in principle, optimal partition achieved via minimising the sum of squared distance to its "representative object" in each cluster

$$E = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} d^2(\mathbf{x}, \mathbf{m}_k)$$

e.g., Euclidean distance

$$d^2(\mathbf{x}, \mathbf{m}_k) = \sum_{n=1}^{N} (x_n - m_{kn})^2$$

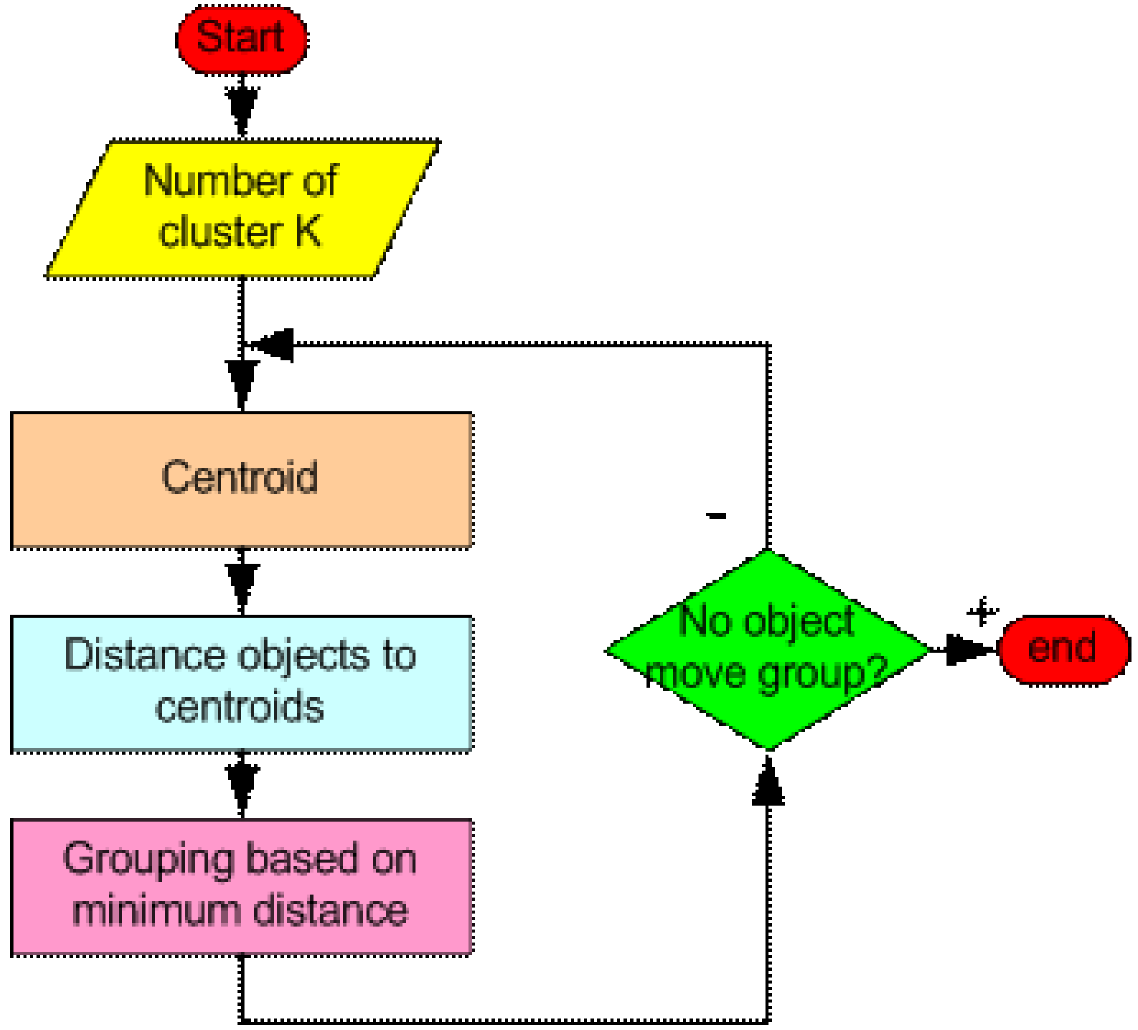
K-means Algorithm

- Given the cluster number *K*, the *K-means* algorithm is carried out in three steps after initialisation:

Initialisation: set seed points (randomly)

1) Assign each object to the cluster of the nearest seed point measured with a specific distance metric
2) Compute new seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., *mean point*, of the cluster)
3) Go back to Step 1), stop when no more new assignment (i.e., membership in each cluster no longer changes)

How the K-Mean Clustering algorithm works?



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

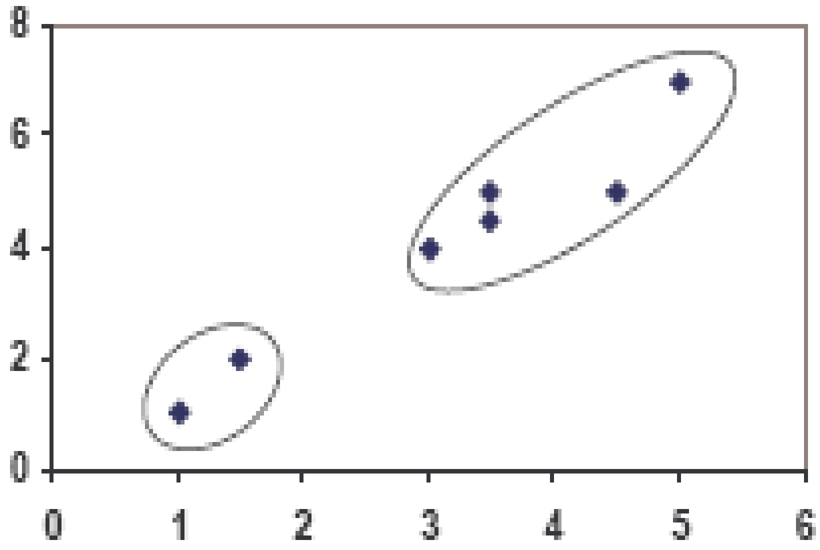
- Step 4 :

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	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72



(with K=3)

Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

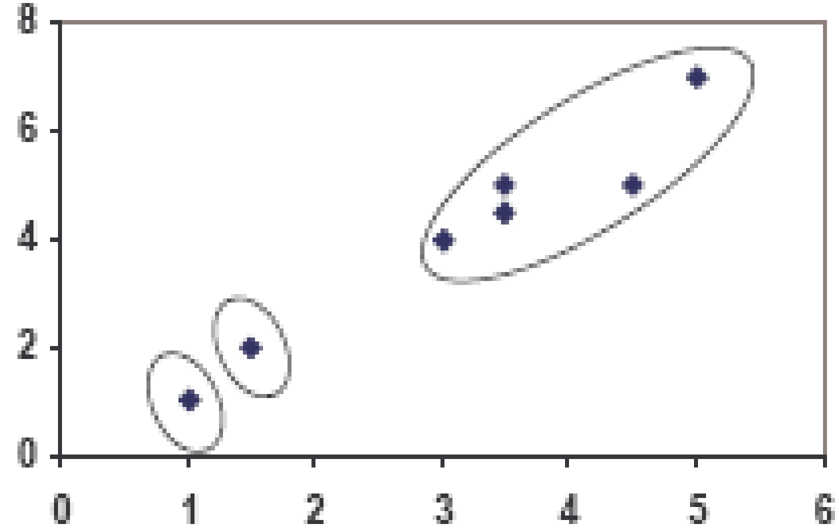
C_3

clustering with initial centroids (1, 2, 3)

Step 1

Individual	m_1 (1.0, 1.0)	m_2 (1.5, 2.0)	m_3 (3.9,5.1)	cluster
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.20	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.61	3
7	4.30	3.20	0.72	3

Step 2

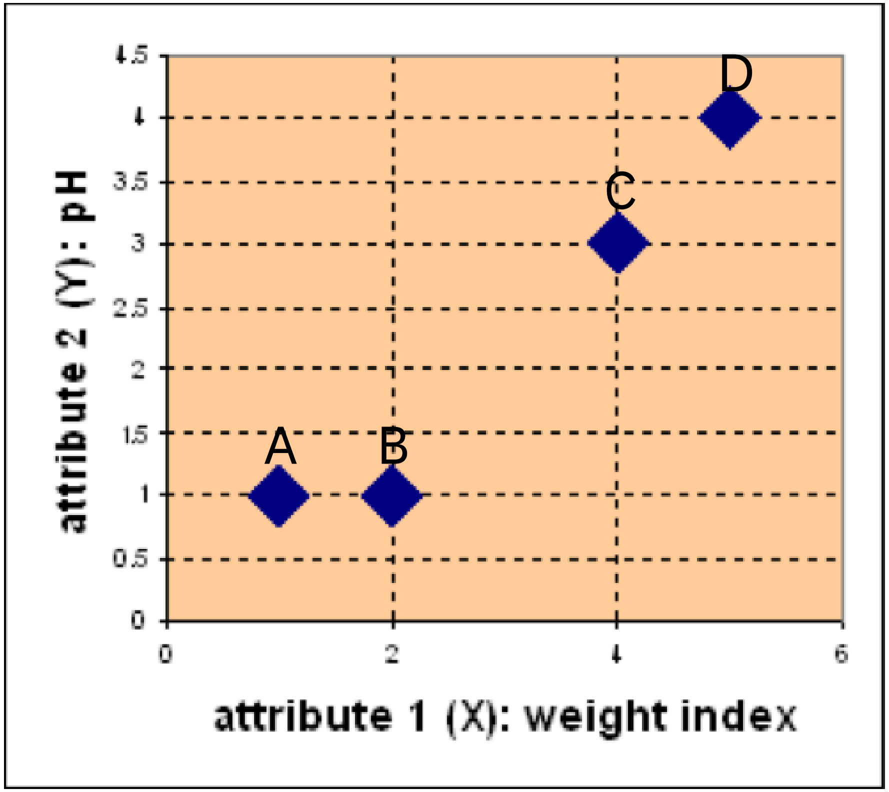


Real Example

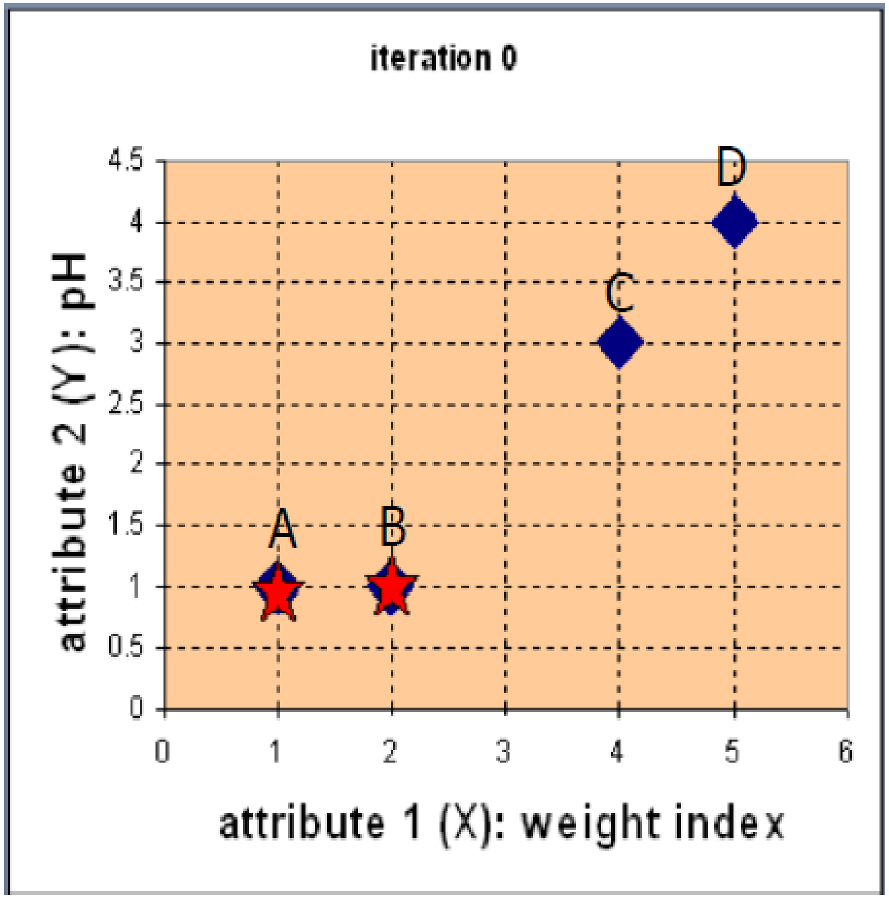
- **Problem**

  Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into $K=2$ group of medicine.

Medicine	Weight	pH-Index
A	1	1
B	2	1
C	4	3
D	5	4



- Step 1: Use initial seed points for partitioning



$$c_1 = A, c_2 = B$$

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & \boxed{5} \\ 1 & 0 & 2.83 & \boxed{4.24} \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1,1) & group - 1 \\ \mathbf{c}_2 = (2,1) & group - 2 \end{array}$$

$$\begin{array}{ccccc} A & B & C & D & \\ \end{array}$$

$$\begin{array}{c} \\ \end{array}\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{c} X \\ Y \end{array}$$

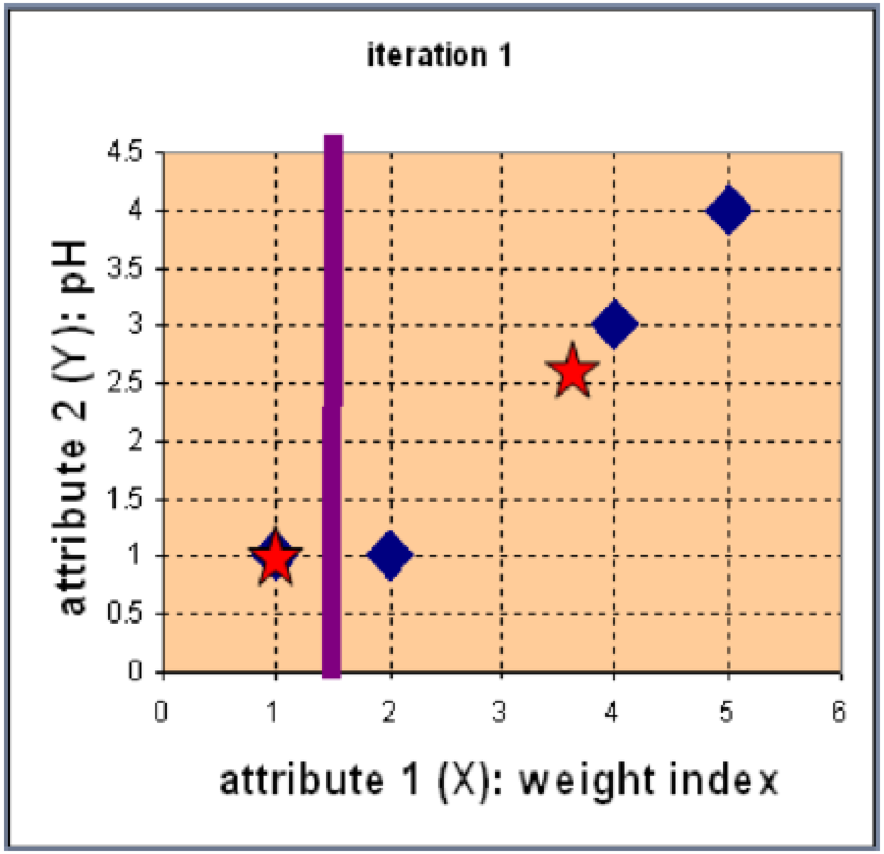
Euclidean distance

$$d(D, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$d(D, c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Assign each object to the cluster with the nearest seed point

- Step 2: Compute new centroids of the current partition



Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

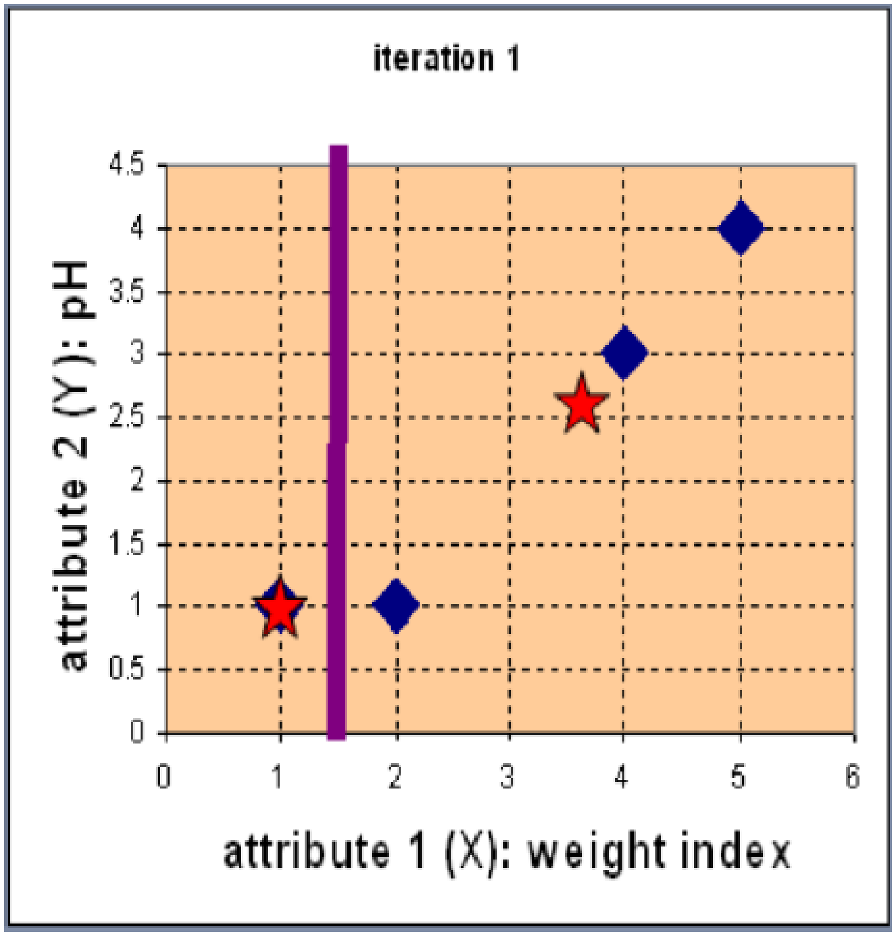
$$c_1 = (1,\ 1)$$

$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$

$$= \left(\frac{11}{3},\ \frac{8}{3}\right)$$

Vassilis S. Kodogiannis

- Step 2: Renew membership based on new centroids



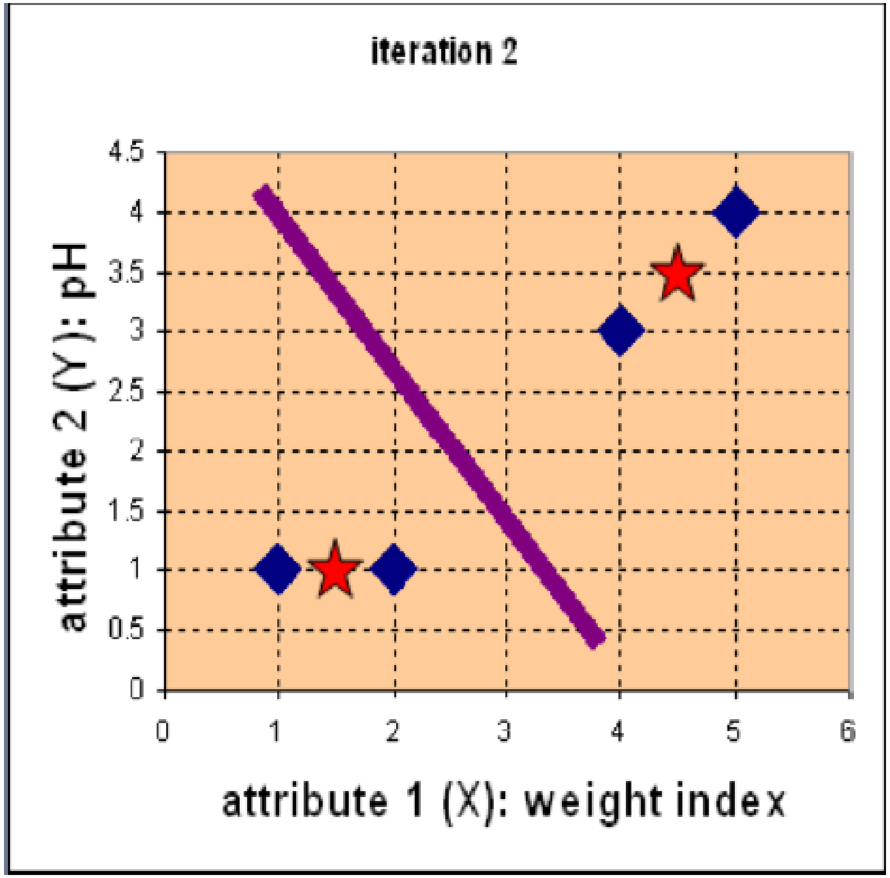
Compute the distance of all objects to the new centroids

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

$$\begin{array}{c c} \begin{matrix} A & B & C & D \end{matrix} & \\ \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} & \begin{matrix} X \\ Y \end{matrix} \end{array}$$

Assign the membership to objects

- Step 3: Repeat the first two steps until its convergence

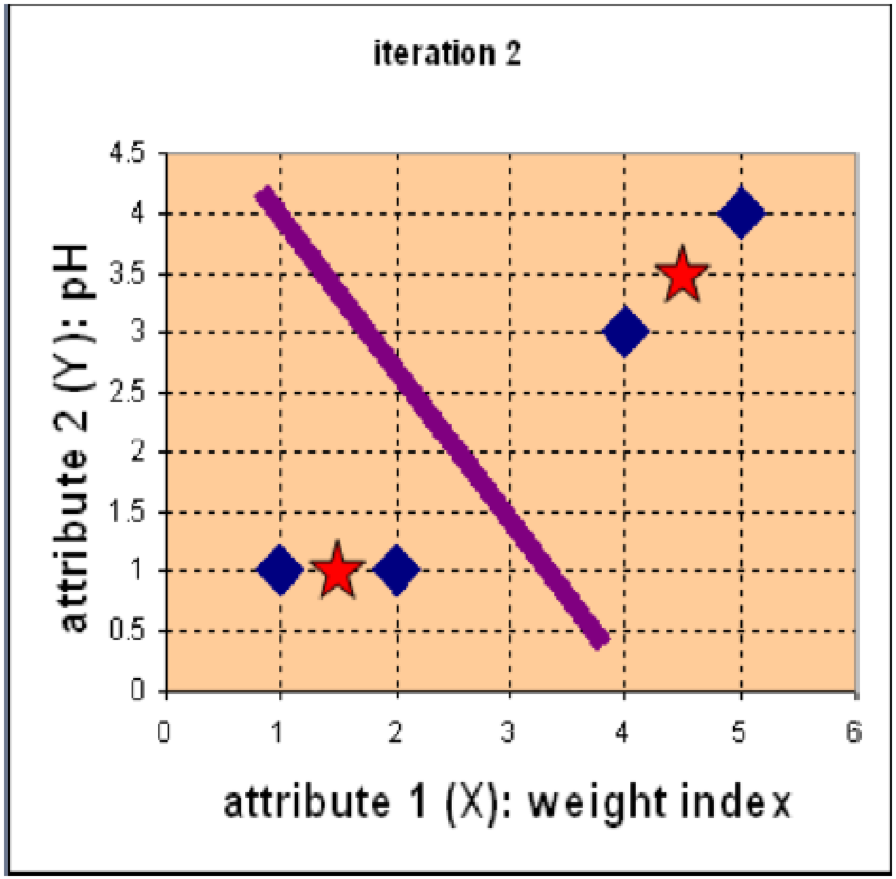


Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$$

- Step 3: Repeat the first two steps until its convergence



Compute the distance of all objects to the new centroids

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1\tfrac{1}{2}, 1) & group-1 \\ \mathbf{c}_2 = (4\tfrac{1}{2}, 3\tfrac{1}{2}) & group-2 \end{array}$$

$$\begin{array}{ccccc} A & B & C & D & \\ \end{array}$$

$$\begin{array}{c} \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \end{array} \begin{array}{l} X \\ Y \end{array}$$

Stop due to no new assignment Membership in each cluster no longer change

K-means clustering summary

Advantages

- Simple, understandable
- Instances automatically assigned to clusters
- Fast

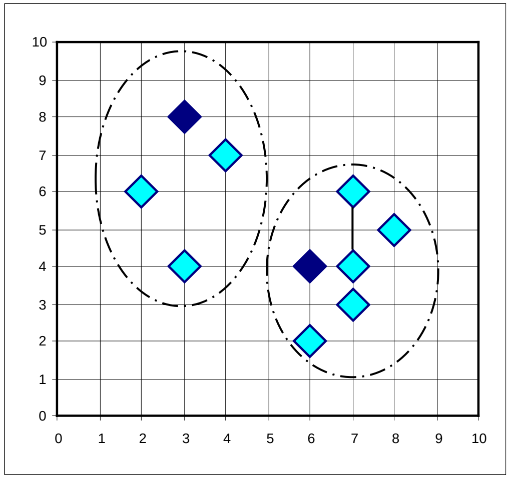
Disadvantages

- Must pick number of clusters beforehand
- All instances forced into a single cluster
- Sensitive to outliers
- Random algorithm
  - Random results
- Not always intuitive
  - Higher dimensions

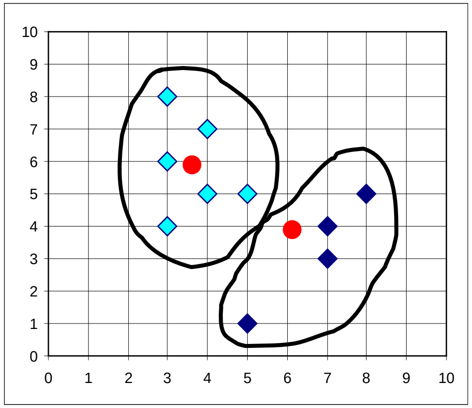
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

K-Medoids

- Partition based clustering (K partitions)
- Effective, why ?
  - Resistant to outliers
  - Do not depend on order in which data points are examined
  - Cluster center is part of dataset, unlike k-means where cluster center is gravity based
  - Experiments show that large data sets are handled efficiently



K-medoids



K-means

K-Medoids

- Minimize the sensitivity of k-means to outliers
- Pick actual objects to represent clusters instead of mean values
- Each remaining object is clustered with the representative object (**Medoid**) to which is the most similar
- The algorithm minimizes the sum of the dissimilarities between each object and its corresponding reference point

$$E = \sum_{i-1}^{k} \sum_{p \in C_i} | p - o_i |$$

→ **E**: the sum of absolute error for all objects in the data set

→ **P**: the data point in the space representing an object

→ **O_i**: is the representative object of cluster C_i

K-Medoids

- Initial representatives are chosen randomly

- The iterative process of replacing representative objects by no representative objects continues as long as the quality of the clustering is improved

- For each representative Object O

  → For each non-representative object R, swap O and R

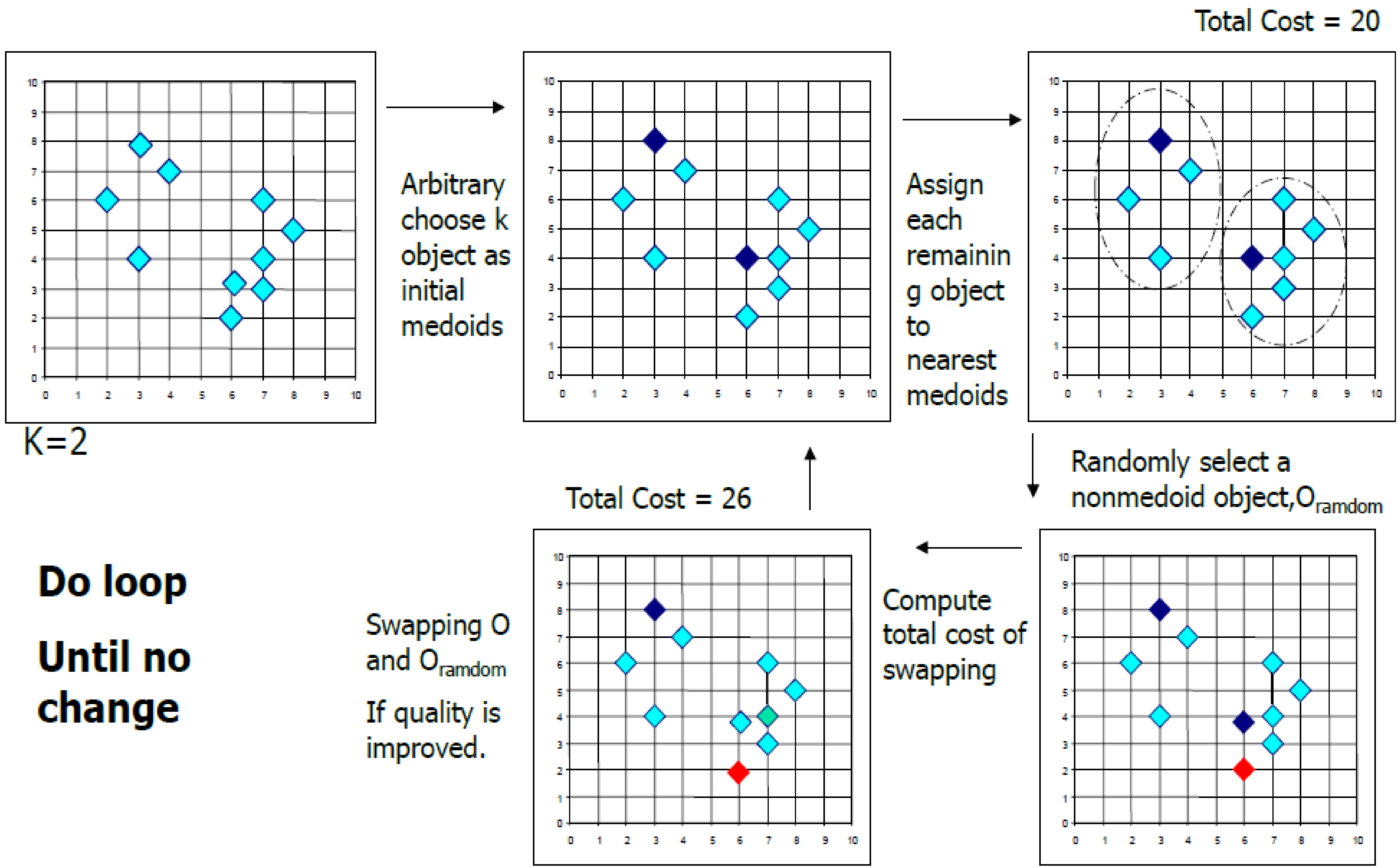
- Choose the configuration with the lowest cost

- Cost function is the difference in absolute error-value if a current representative object is replaced by a non-representative object

The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - 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 (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

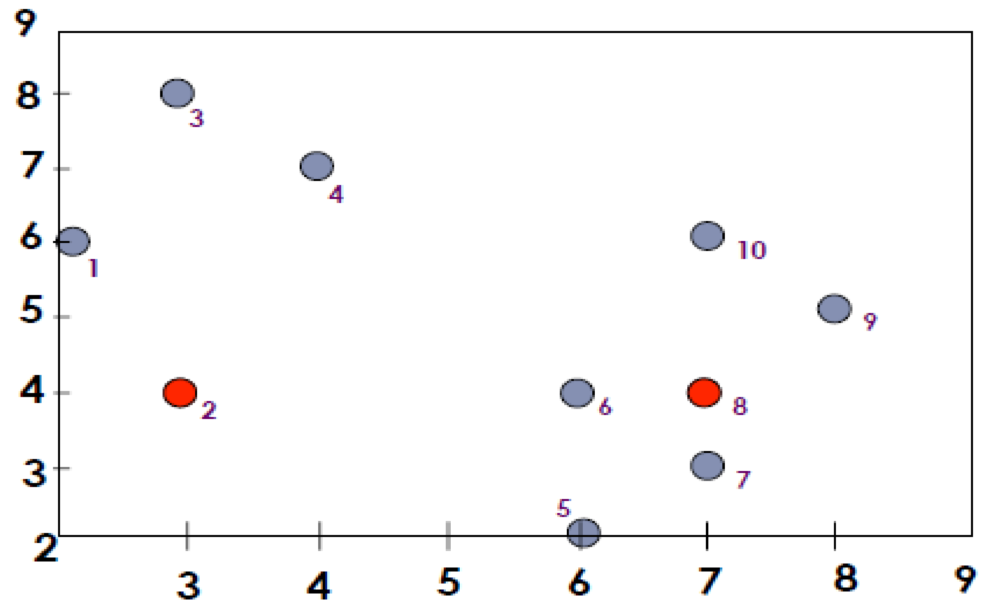
PAM: A Typical K-Medoids Algorithm



K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



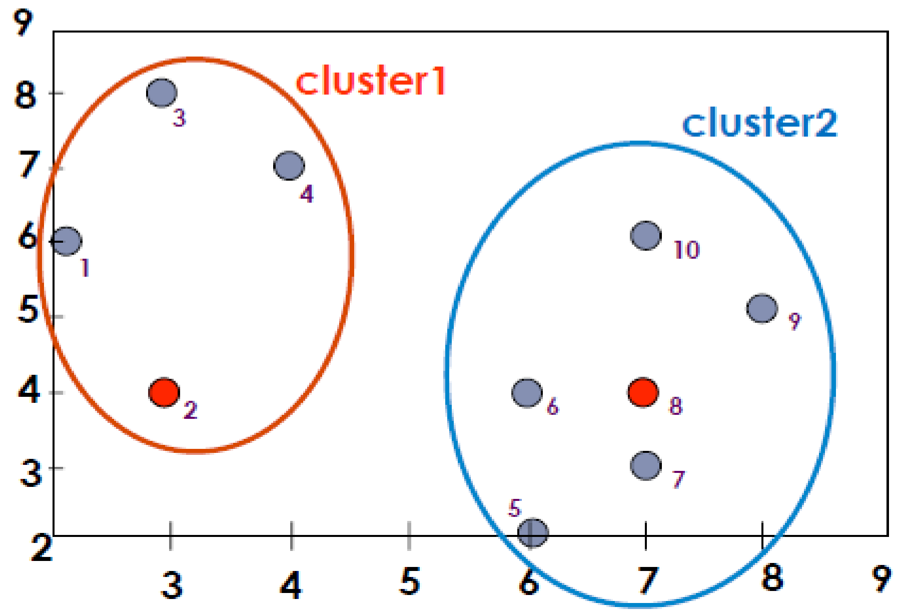
Goal: create two clusters

Choose randmly two mediods

$O_2 = (3,4)$
$O_8 = (7,4)$

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Assign each object to the closest representative object

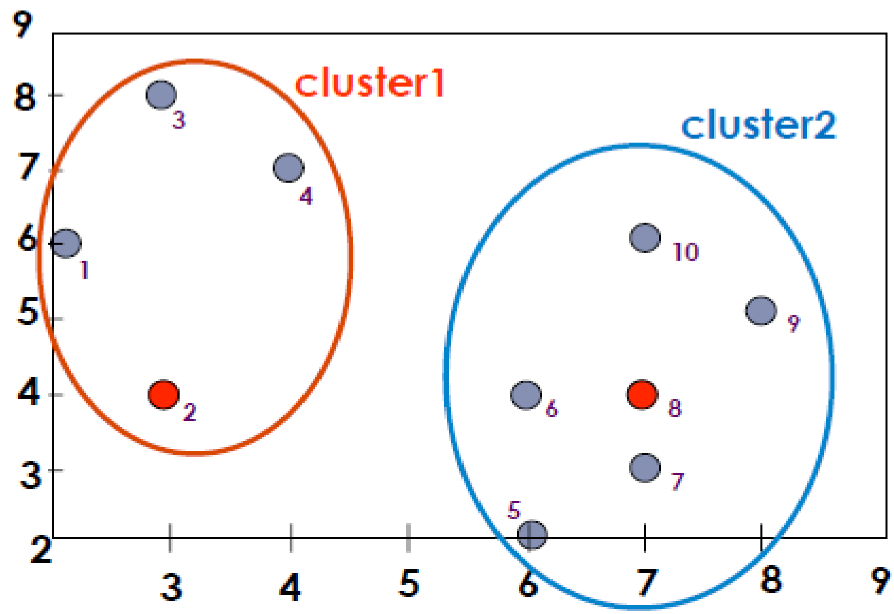
→Using L1 Metric (Manhattan), we form the following clusters

Cluster1 = $\{O_1, O_2, O_3, O_4\}$

Cluster2 = $\{O_5, O_6, O_7, O_8, O_9, O_{10}\}$

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



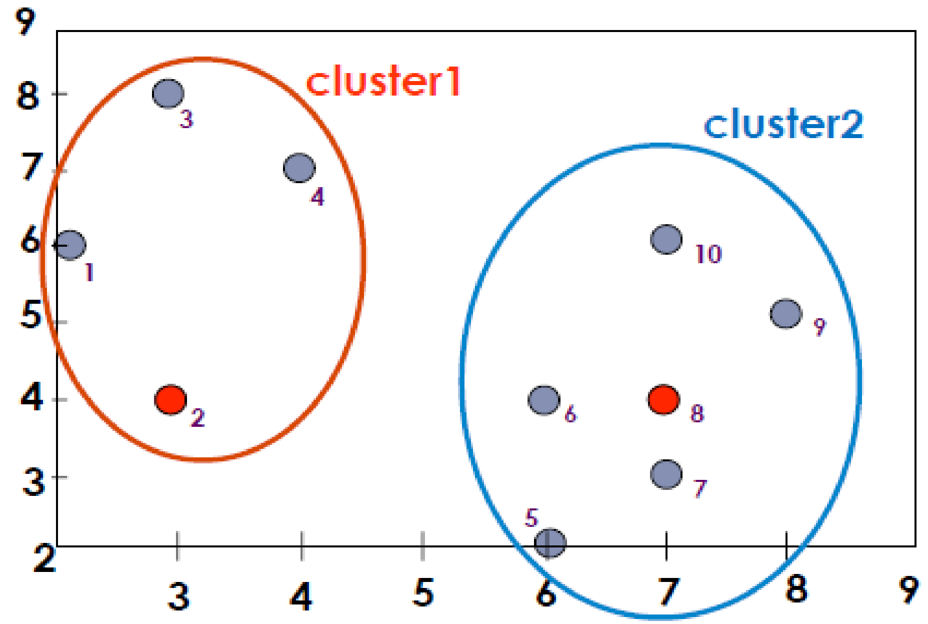
→Compute the absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E=\sum_{i-1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6

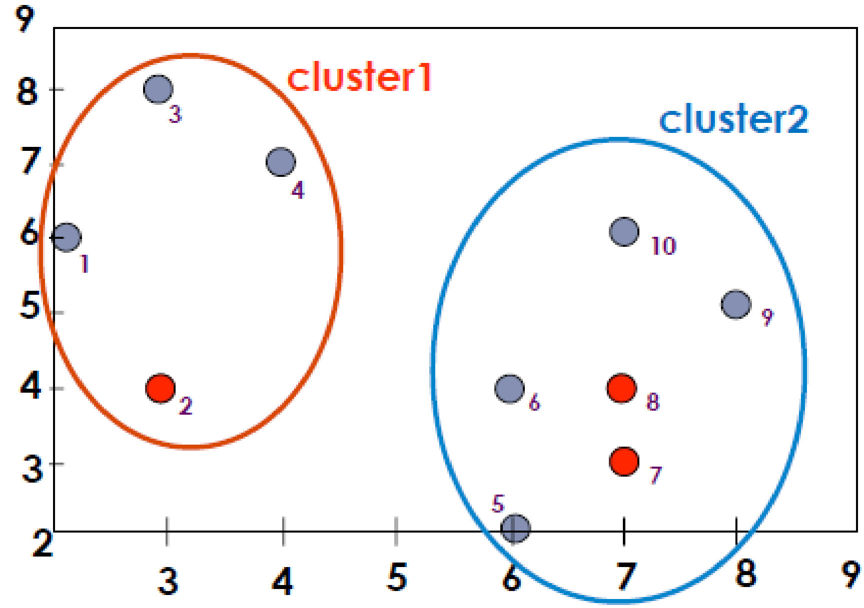


→The absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E = (3+4+4)+(3+1+1+2+2) = 20$$

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Choose a random object O_7

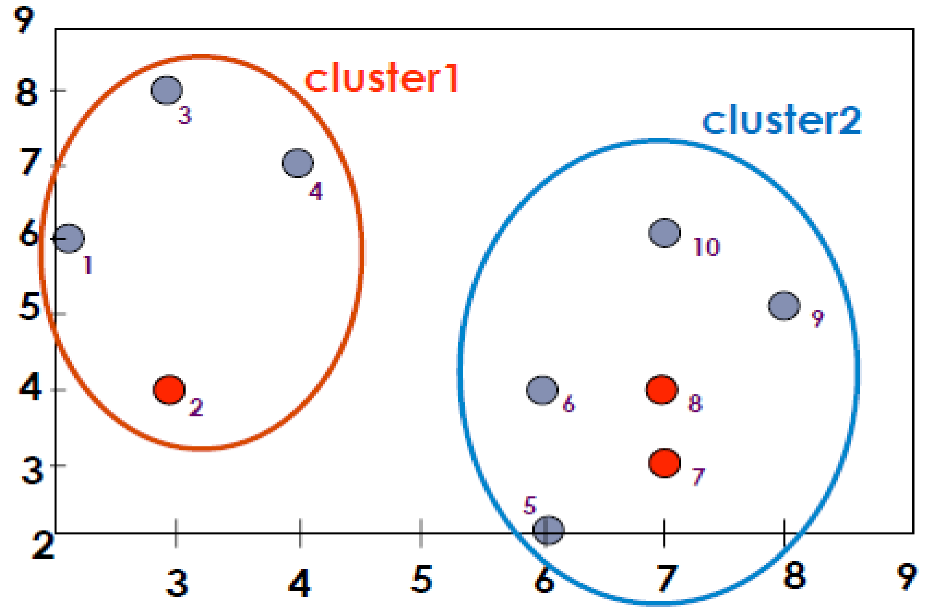
→Swap **O8** and **O7**

→Compute the absolute error criterion **[for the set of Medoids (O2,O7)]**

$$E = (3+4+4)+(2+2+1+3+3) = 22$$

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the cost function

Absolute error [for O_2,O_7] – Absolute error [O_2,O_8]

$$S = 22 - 20$$

$S > 0 \Rightarrow$ it is a bad idea to replace O_8 by O_7

Suggested Books:

- ***Data Mining: Practical Machine Learning Tools and Techniques***, Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal, Morgan Kaufmann, 2016

- ***An Introduction to Statistical Learning: with Applications in R***, Gareth James, Daniela Witten, Trevor Hastie, Springer; 2013.

- ***The R Book***, Michael J. Crawley, Wiley-Blackwell, 2012