K-means Clustering

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

- Given a *K*, find a partition of *K clusters* to optimise the chosen partitioning criterion (cost function)
    - global optimum: exhaustively search all partitions
- The *K-means* algorithm: a heuristic method
    - K-means algorithm (MacQueen'67): each cluster is represented by the centre of the cluster and the algorithm converges to stable centriods of clusters.
    - K-means algorithm is the simplest partitioning method for clustering analysis and widely used in data mining applications.

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)

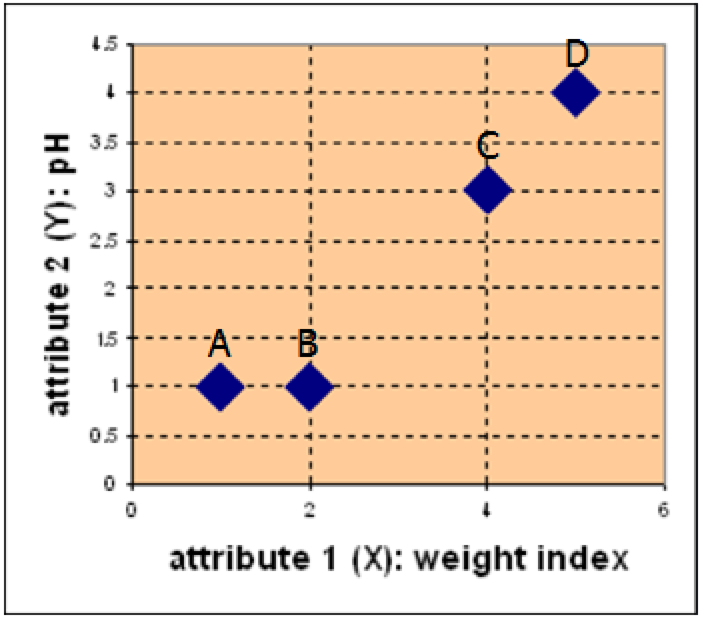
1) Assign each object to the cluster of the nearest seed point measured with a specific distance metric
2) Compute new seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., *mean point*, of the cluster)
3) Go back to Step 1), stop when no more new assignment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

Example

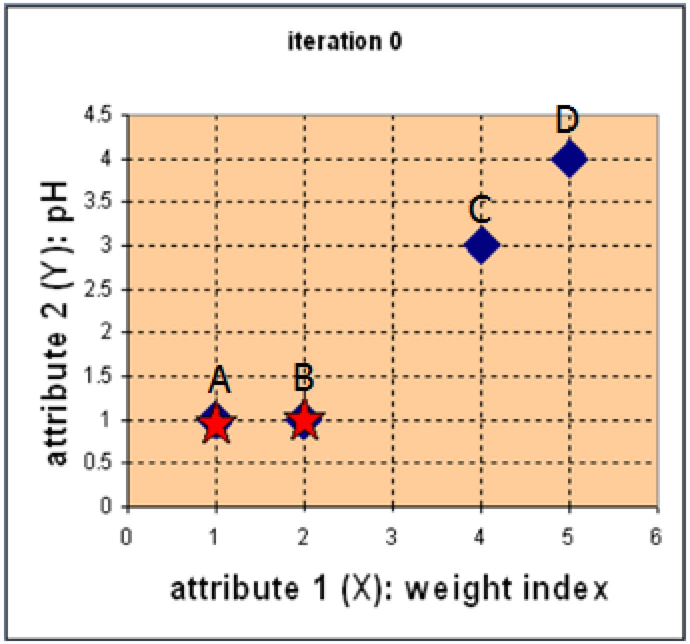
- Problem

  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



Example



Step 1: Use initial seed points for partitioning

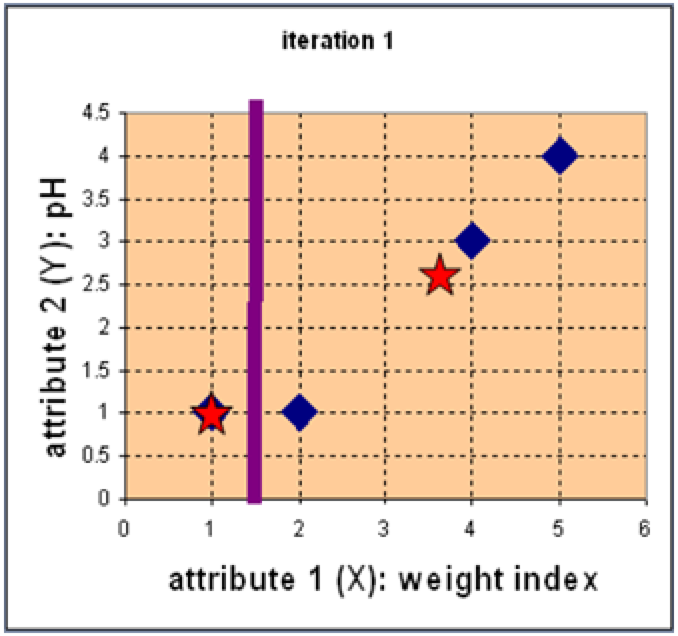
$$c_1 = A,\ c_2 = B$$

Step 2 Assign each object to the cluster with the nearest seed point

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

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

Step 3 Compute new centroids of the current partition



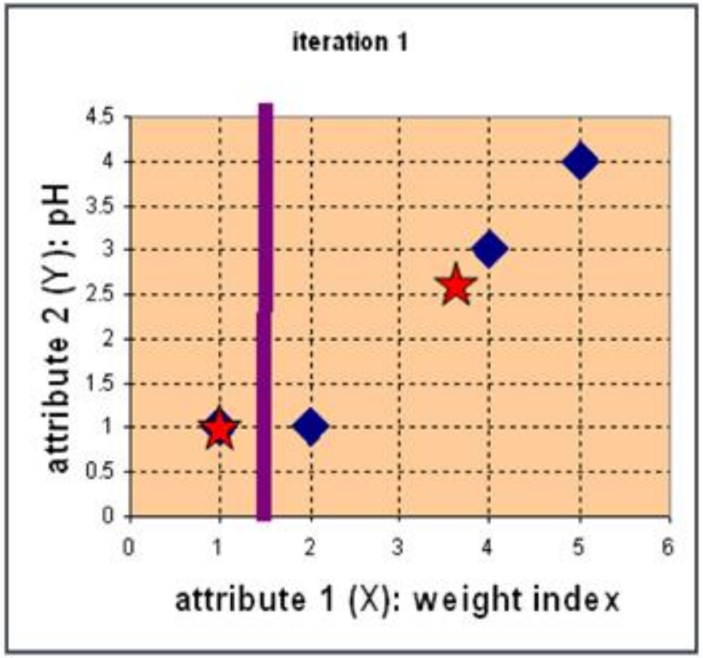
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

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

$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$
$$= (\frac{11}{3},\ \frac{8}{3})$$

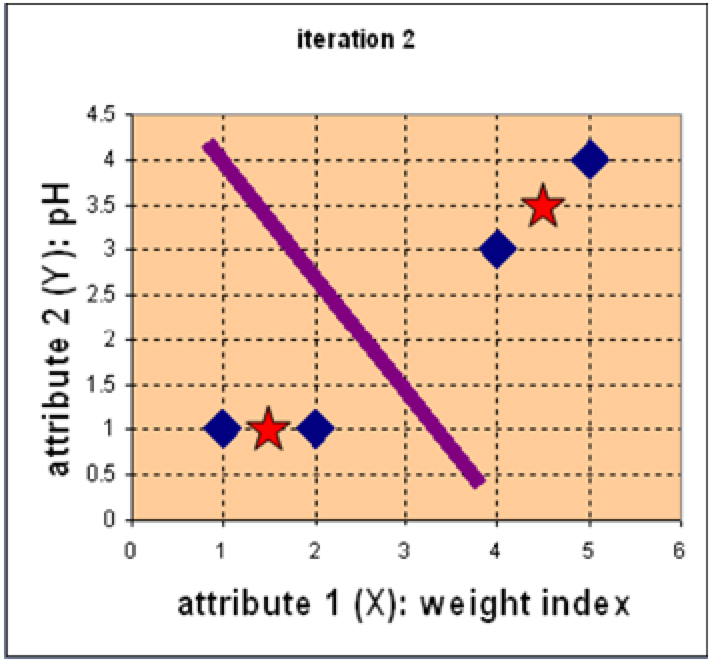
Step 2 Assign each object to the cluster with the nearest seed point

Renew membership based on new centroids



Step 3 Compute new centroids of the current partition

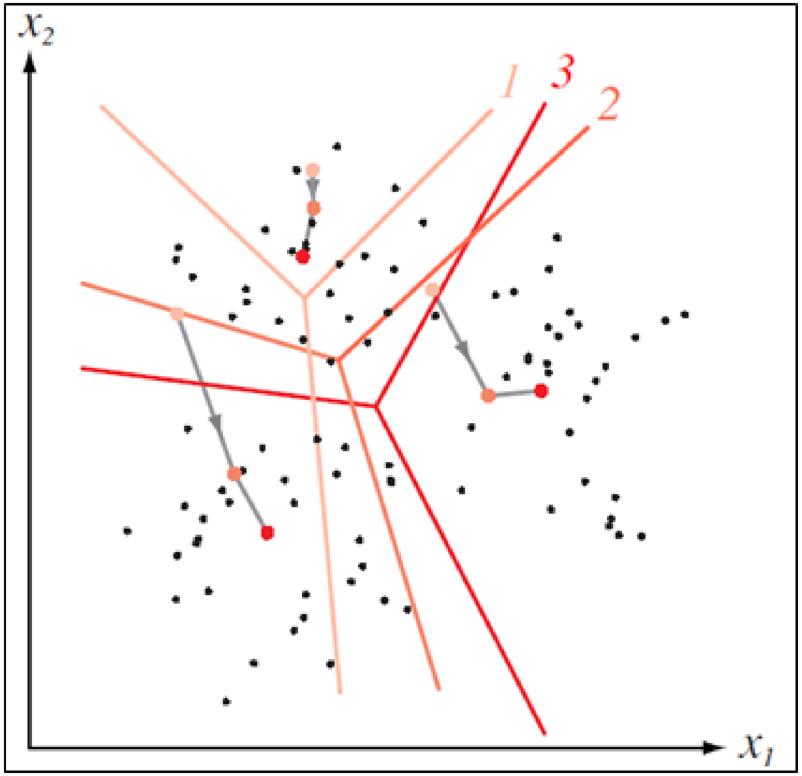
Repeat until its convergence



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

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

How K-means partitions?



When K centroids are set/fixed, they partition the whole data space into K mutually exclusive subspaces to form a partition.

A partition amounts to a

[Voronoi Diagram]

Changing positions of centroids leads to a new partitioning.

Relevant Issues

- Efficient in computation
  - O(*tKn*), where *n* is number of objects, *K* is number of clusters, and *t* is number of iterations. Normally, *K*, *t* << *n*.
- Local optimum
  - sensitive to initial seed points
  - converge to a local optimum: maybe an unwanted solution
- Other problems
  - Need to specify *K*, the *number* of clusters, in advance
  - Unable to handle noisy data and outliers (*K-Medoids* algorithm)
  - Not suitable for discovering clusters with non-convex shapes
  - Applicable only when mean is defined, then what about categorical data? (*K-mode* algorithm)
  - how to evaluate the *K*-mean performance?

Summary

- *K*-means algorithm is a simple yet popular method for clustering analysis
- Its performance is determined by initialisation and appropriate distance measure
- There are several variants of *K*-means to overcome its weaknesses
  - *K*-Medoids: resistance to noise and/or outliers
  - *K*-Modes: extension to categorical data clustering analysis
  - CLARA: extension to deal with large data sets
  - Mixture models (EM algorithm): handling uncertainty of clusters