K-Means Clustering

What is clustering?

- Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

Types of clustering

- **Hierarchical algorithms**: These find successive clusters using previously established clusters.
  - **Agglomerative ("bottom-up")**: Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.
  - **Divisive ("top-down"):** Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
- **Partitional clustering**: Partitional algorithms determine **all clusters at once**. They include:
  - **K-means and derivatives**
  - **Fuzzy c-means clustering**
  - **QT clustering algorithm**

Common Distance measures

- **Distance measure will determine how the similarity of two elements is calculated and it will influence the shape of the clusters.**

They include:

1. The Euclidean distance (also called 2-norm distance) is given by:

$$d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

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

Common Distance measures

3.The **maximum norm** is given by:

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = \max_{1 \le i \le p} |x_i - y_i|$$

4.The **Mahalanobis** distance corrects data for different scales and correlations in the variables.

5. **Inner product space:** The angle between two vectors can be used as a distance measure when clustering high dimensional data

6. **Hamming distance** (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.

K-Means Clustering

- The k-means algorithm is an algorithm to cluster n objects based on attributes into k partitions, where k < n.
- It attempts 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

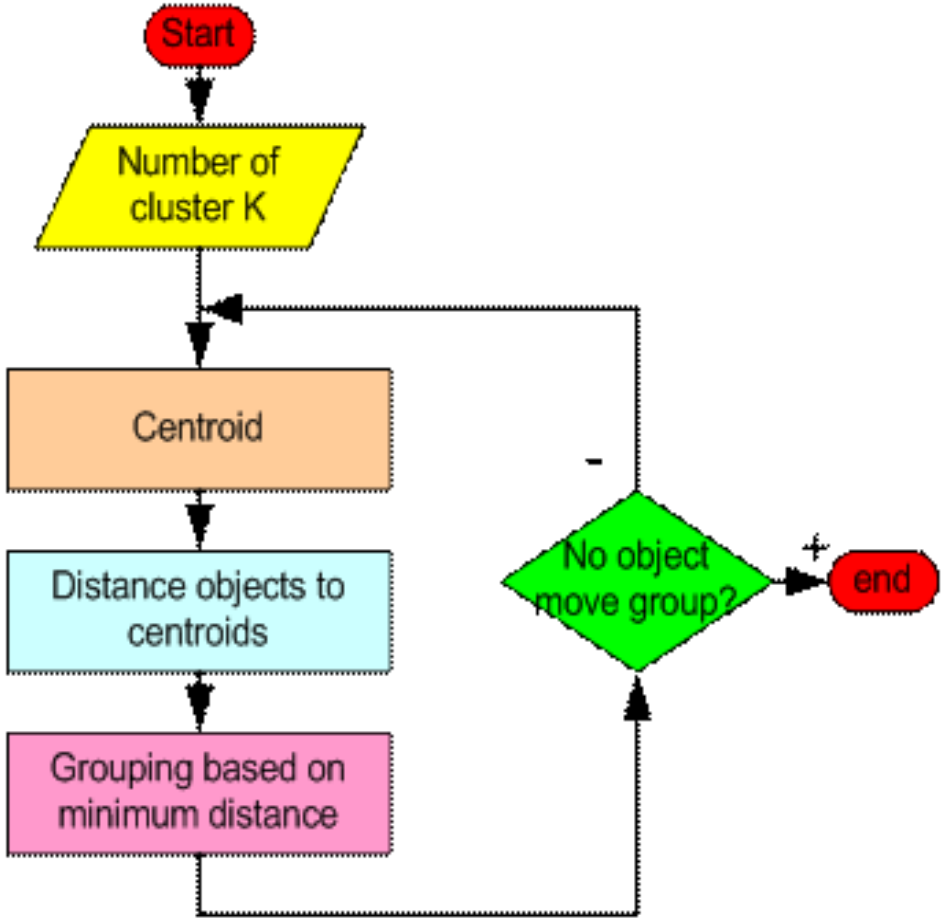
$$J = \sum_{j=1}^{k} \sum_{n \in S_j} |x_n - \mu_j|^2$$

where x_n is a vector representing the nth data point and μ_j is the geometric centroid of the data points in S_j.

K-Means Clustering

- Simply speaking k-means clustering is an algorithm to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?



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.

How the K-Mean Clustering algorithm works?

Step 1: Choose the number K of clusters.

Step 2: Select at random K points , the centroids (not necessarily from your dataset)

Step 3: Assign each data point to the closest clusters ⟶ that forms K clusters

Step 4: Compute and place the new centroid of each cluster.

Step 5: Reassign each data point to the new closest centroid.

If any reassignment took place, go to step 4, otherwise go to finish state.

A Simple example (k=2)

Individual	Varaible1	Variable2
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

A Simple example (k=2)

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)

A Simple example (k=2)

Step 2: Thus, we obtain two clusters containing: {1,2,3} and {4,5,6,7}. Their new centroids are:

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

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

$$= (4.12, 5.38)$$

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

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

Individual	Varaible1	Variable2
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

A Simple example (k=2)

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)

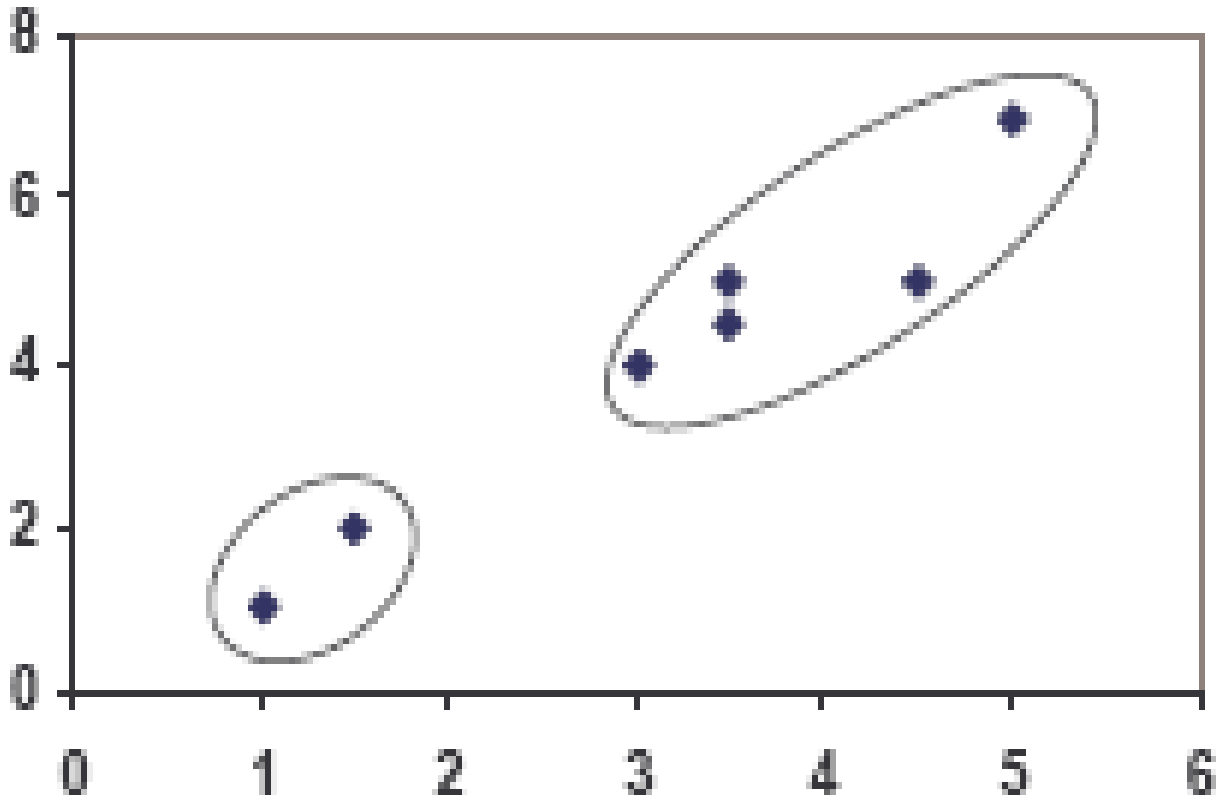
Individual	Varaible1	Variable2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

A Simple example (k=2)

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	Varaible1	Variable2
1	0.58	5.02
2	0.58	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

Plot



A Simple example (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	1.61	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

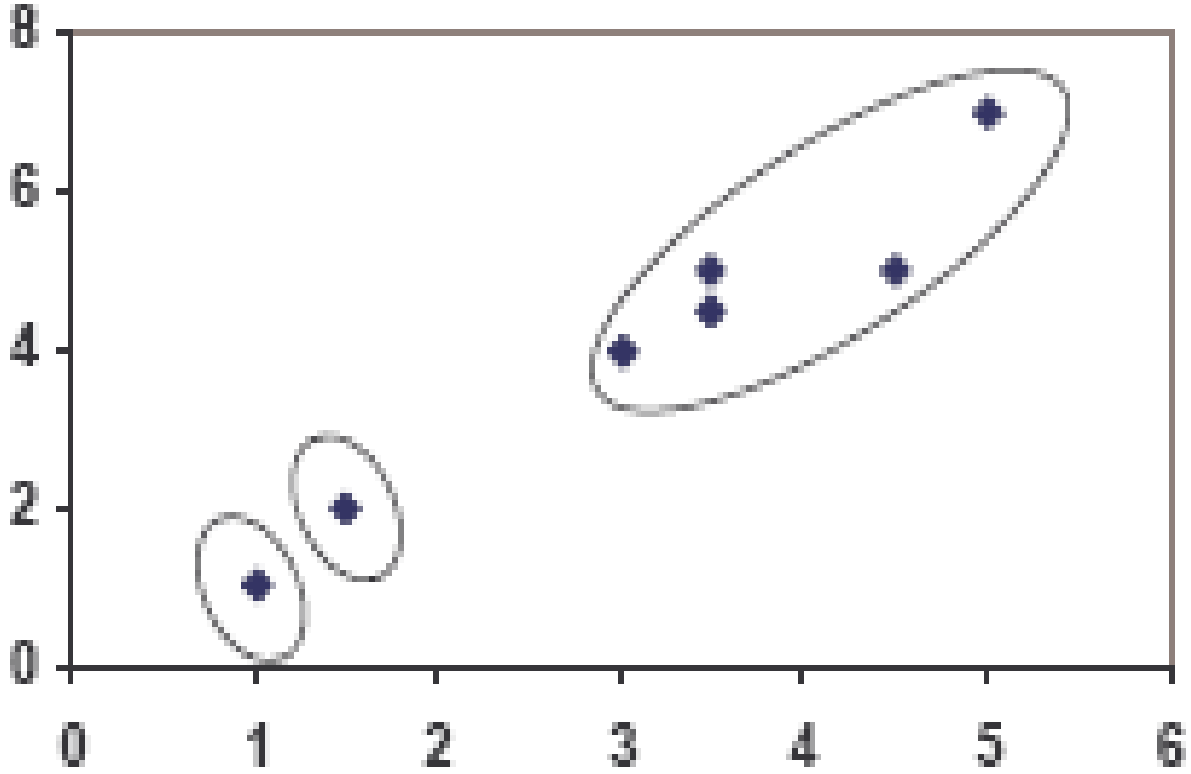
Clustering with initial centroids (1, 2, 3)

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

Plot



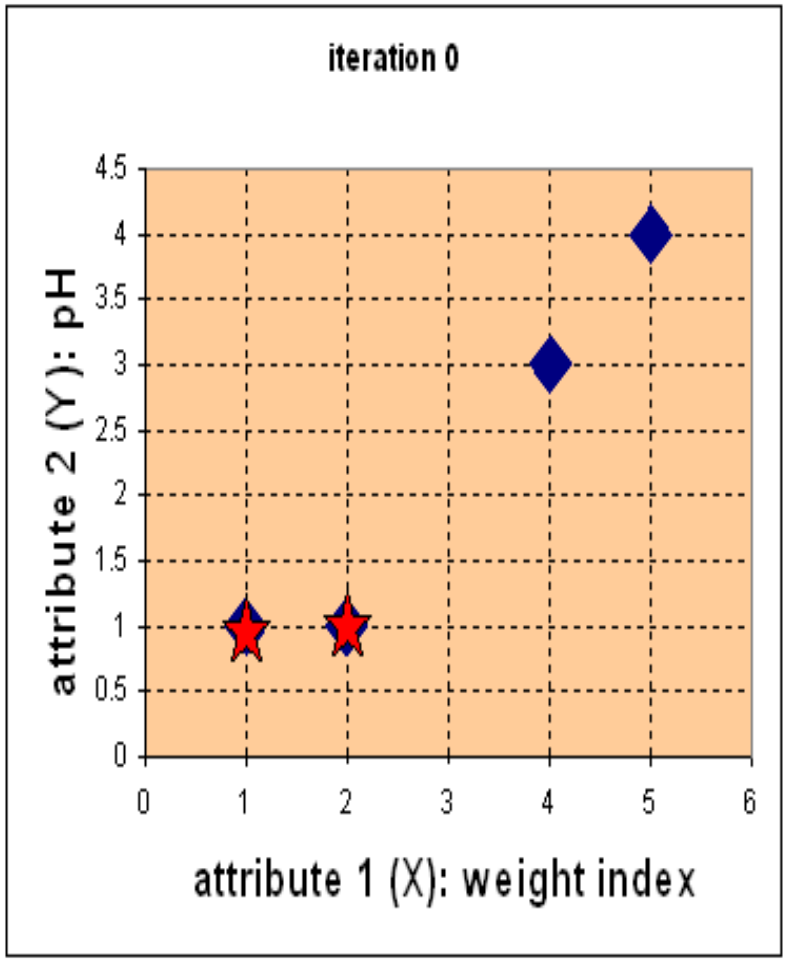
Real-Life Numerical Example of K-Means Clustering

- We have 4 medicines as our training data points object and each medicine has 2 attributes.
- Each attribute represents coordinate of the object. We have to determine which medicines belong to cluster 1 and which medicines belong to the other cluster.

Object	Attribute1 (X): weight index	Attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Real-Life Numerical Example of K-Means Clustering

Step 1: Initial value of centroids : Suppose we use medicine A and medicine B as the first centroids.
Let and c1 and c2 denote the coordinate of the centroids, then c1=(1,1) and c2=(2,1)



Real-Life Numerical Example of K-Means Clustering

Objects-Centroids distance : we calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration o is

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

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Each column in the distance matrix symbolizes the object.
The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid.
For example, distance from medicine C = (4, 3) to the first centroid $\mathbf{c}_1 = (1,1)$ is, $\sqrt{(4-1)^2+(3-1)^2} = 3.61$ and its distance to the second centroid $\mathbf{c}_2 = (2,1)$ is , $\sqrt{(4-2)^2+(3-1)^2} = 2.83$ etc.

Real-Life Numerical Example of K-Means Clustering

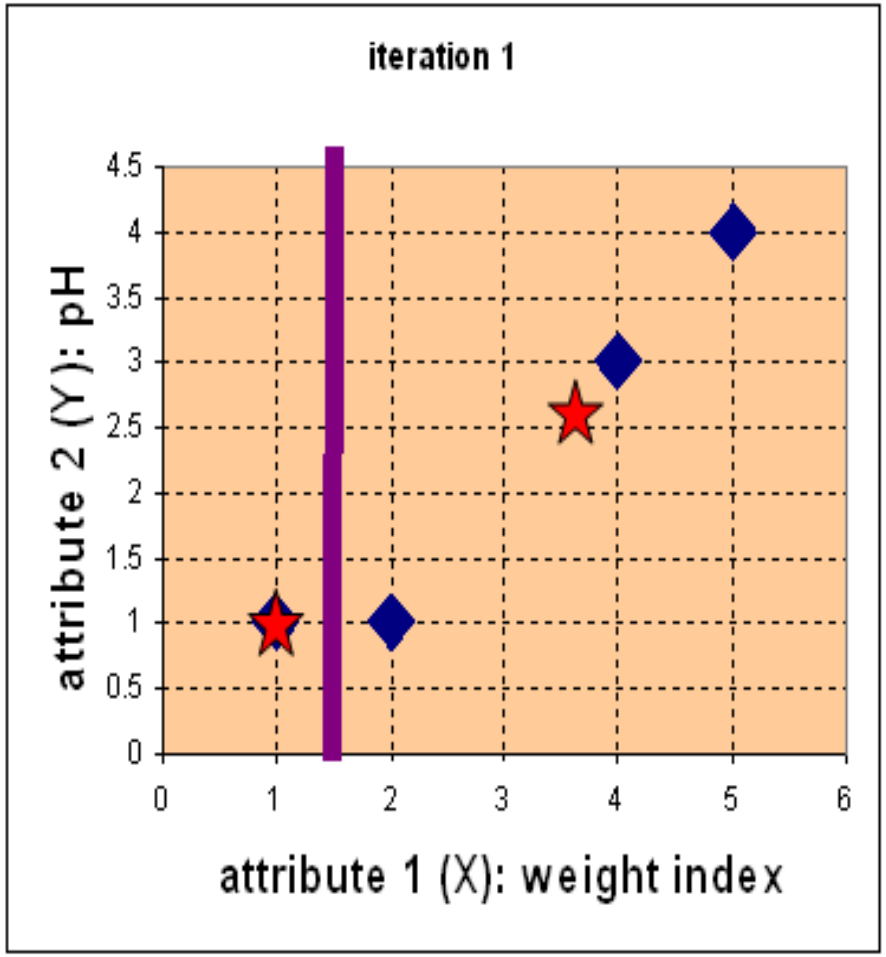
Step 2: Objects clustering : We assign each object based on the minimum distance.

Medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2.

The elements of Group matrix below is 1 if and only if the object is assigned to that group.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$A \quad B \quad C \quad D$$



Real-Life Numerical Example of K-Means Clustering

Iteration-1, Objects-Centroids distances : The next step is to compute the distance of all objects to the new centroids.
Similar to step 2, we have distance matrix at iteration 1 is

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

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

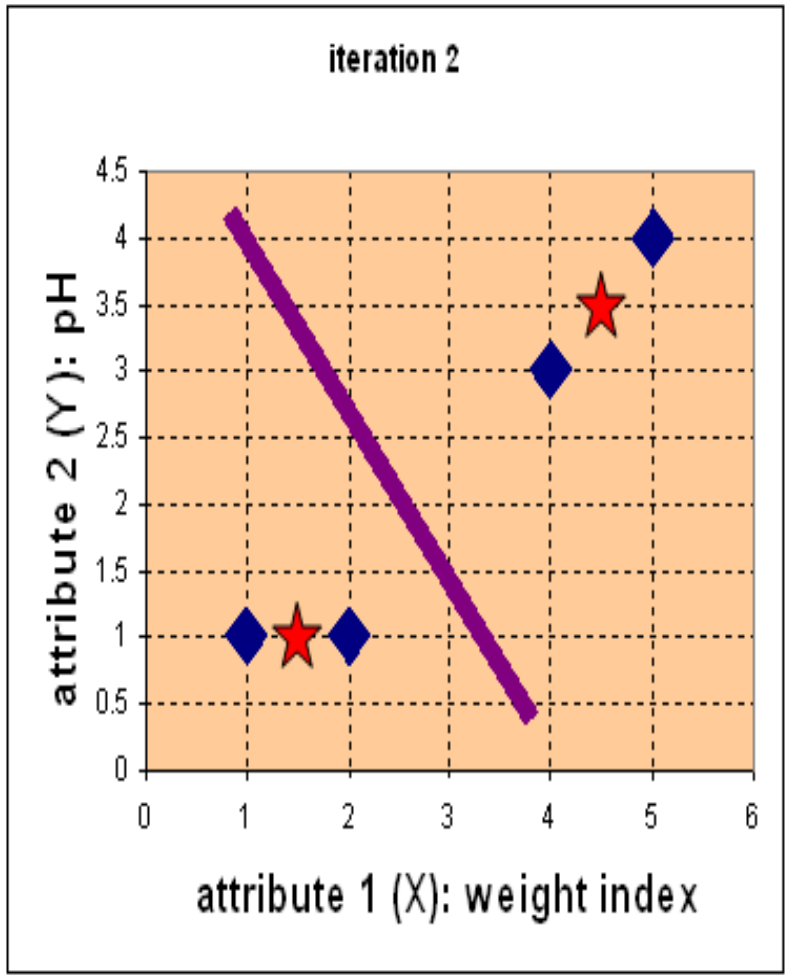
Real-Life Numerical Example of K-Means Clustering

Iteration-1,Objects clustering: Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$A \quad B \quad C \quad D$

Iteration 2, determine centroids: Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new centroids are and $c_1 = (\frac{1+2}{2}, \frac{1+1}{2}) = (1\frac{1}{2}, 1)$ $c_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\frac{1}{2}, 3\frac{1}{2})$



Real-Life Numerical Example of K-Means Clustering

Iteration-2, Objects-Centroids distances : Repeat step 2 again, we have new distance matrix at iteration 2 as

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

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Real-Life Numerical Example of K-Means Clustering

Iteration-2, Objects clustering: Again, we assign each object based on the minimum distance.

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$A \quad B \quad C \quad D$$

We obtain result that $\mathbf{G}^2 = \mathbf{G}^1$. Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore.

Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed..

Real-Life Numerical Example of K-Means Clustering

We get the final grouping as the results as:

Object	Feature1(X): weight index	Feature2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

Relevant Issues

- Computational complexity
  - 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?

Application

Colour-Based Image Segmentation Using K-means

- **Step 1: Loading a colour image of tissue stained with hemotoxylin and eosin (H&E)**

H&E image

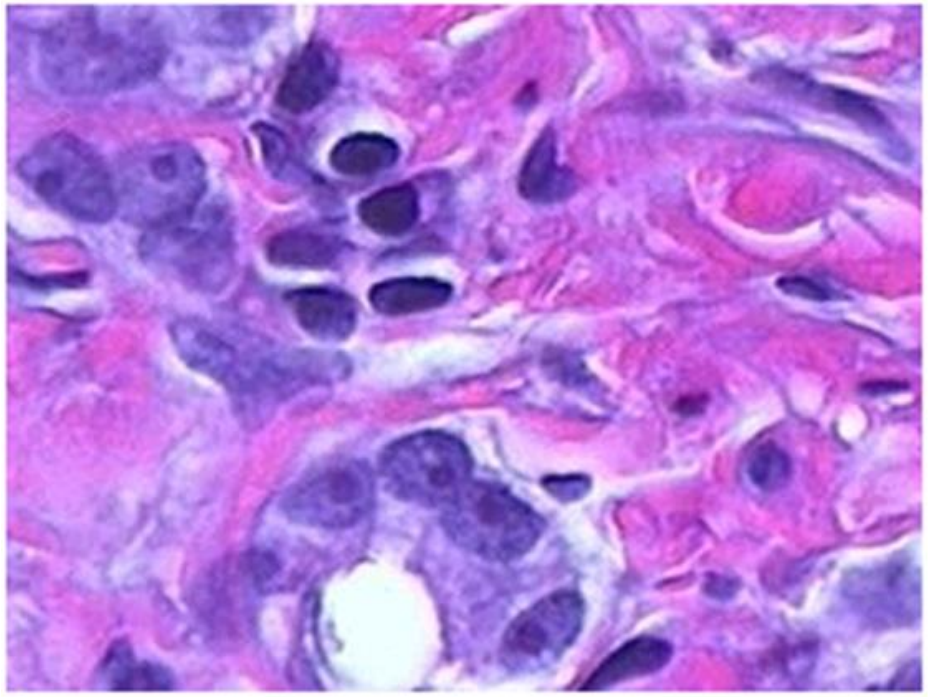


Image courtesy of Alan Partin, Johns Hopkins University

Application

Colour-Based Image Segmentation Using K-means

- **Step 2: Convert the image from RGB colour space to L*a*b* colour space**
  - **Unlike the RGB colour model, L*a*b* colour is designed to approximate human vision.**
  - **There is a complicated transformation between RGB and L*a*b*. (L*, a*, b*) = T(R, G, B).**

$$(R, G, B) = T'(L^*, a^*, b^*).$$

Application

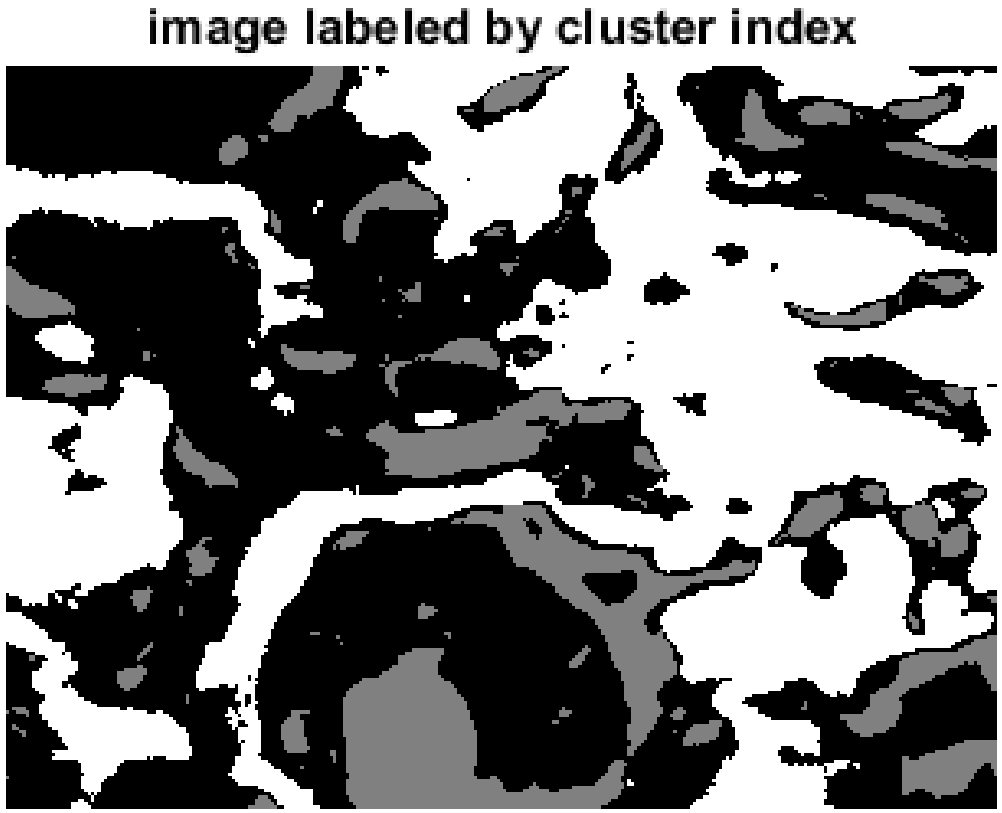
Colour-Based Image Segmentation Using K-means

Step 3: Undertake clustering analysis in the (a*, b*) colour space with the K-means algorithm

- **In the L*a*b* colour space, each pixel has a properties or feature vector: (L*, a*, b*).**
- **Like feature selection, L* feature is discarded. As a result, each pixel has a feature vector (a*, b*).**
- **Applying the K-means algorithm to the image in the a*b* feature space where K = 3 by applying the domain knowledge.**

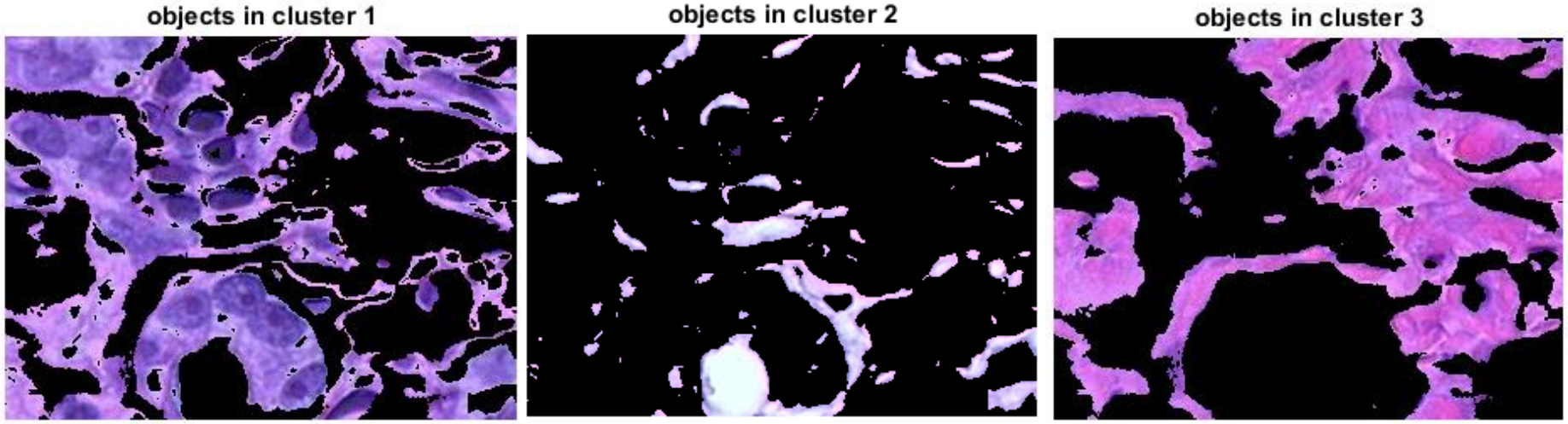
Application

Step 4: Label every pixel in the image using the results from K-means clustering (indicated by three different grey levels)



Application

Step 5: Create Images that Segment the H&E Image by Colour. Apply the label and the colour information of each pixel to achieve separate colour images corresponding to three clusters.



"blue" pixels

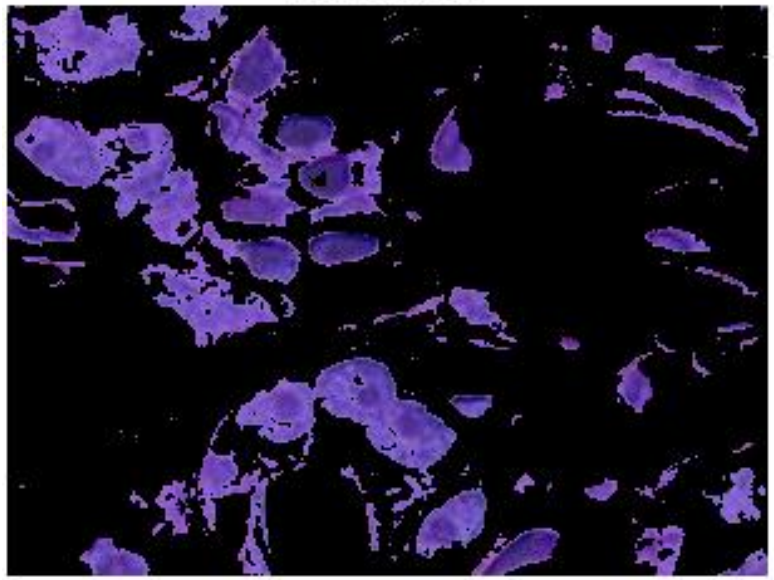
"white" pixels

"pink" pixels

Application

Step 6: Segment the nuclei into a separate image with the L* feature
In cluster 1, there are dark and light blue objects (pixels). The dark blue objects (pixels) correspond to nuclei (with the domain knowledge). L* feature specifies the brightness values of each colour. With a threshold for L*, we achieve an image containing the nuclei only.

blue nuclei



Weaknesses of K-Mean Clustering

- When the numbers of data are not so many, initial grouping will determine the cluster significantly.
- 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.
- 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.
- It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the local optimum.

Summary

- K-means algorithm is a simple yet popular method for clustering analysis
- Its performance is determined by initialization 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