

# *K*-means Clustering

Ke Chen

# Outline

- Introduction
- $K$ -means Algorithm
- Example
- How  $K$ -means partitions?
- $K$ -means Demo
- Relevant Issues
- Application: Cell Neulei Detection
- Summary

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

# Introduction

- 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)

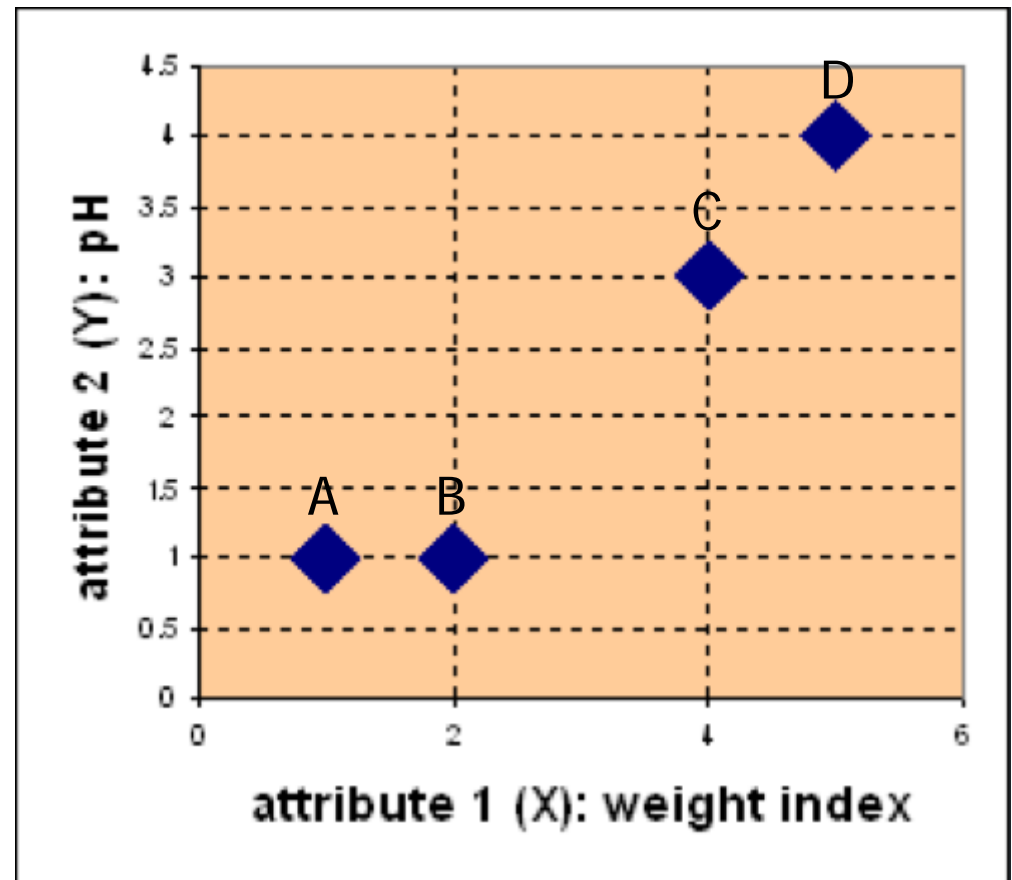
- 1) Assign each object to the cluster of the nearest seed point measured with a specific distance metric
- 2) Compute new seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., *mean point*, of the cluster)
- 3) Go back to Step 1), stop when no more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

# Example

- Problem

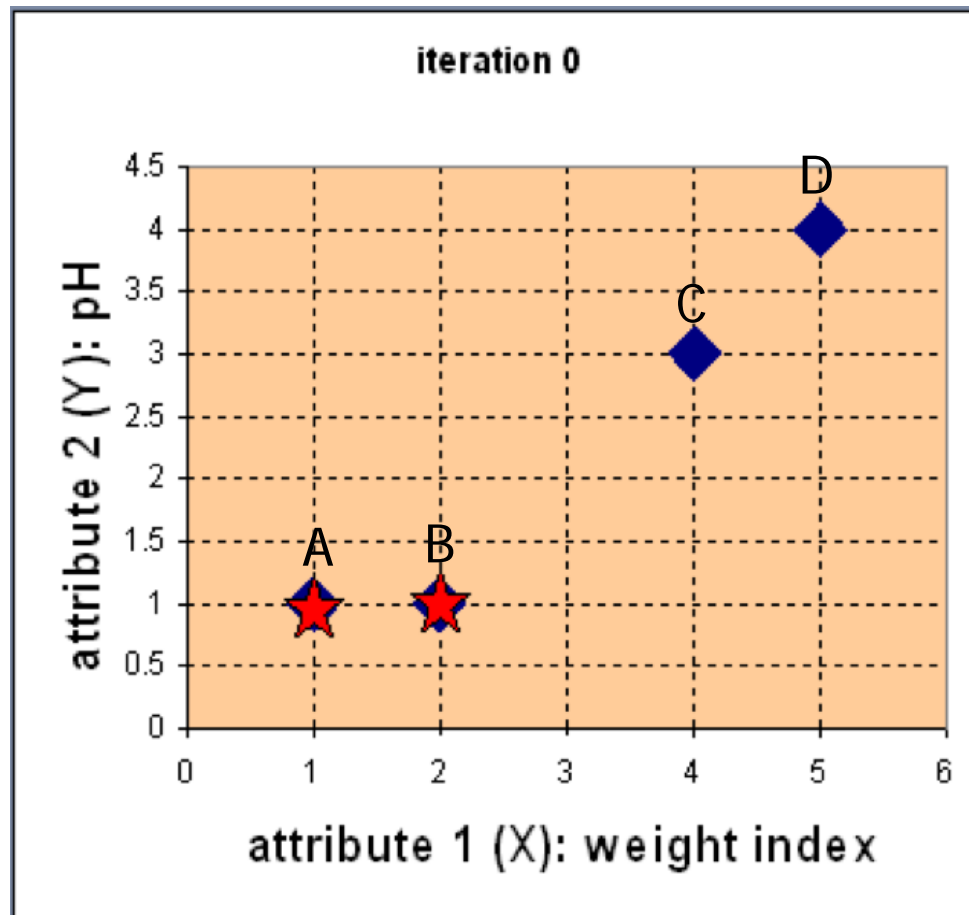
Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into  $K=2$  group of medicine.

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



$$c_1 = A, c_2 = B$$

$D^0 =$	0	1	3.61	5	$c_1 = (1,1)$	group - 1
	1	0	2.83	4.24	$c_2 = (2,1)$	group - 2
	A	B	C	D	Euclidean distance	
	1	2	4	5	X	
	1	1	3	4	Y	

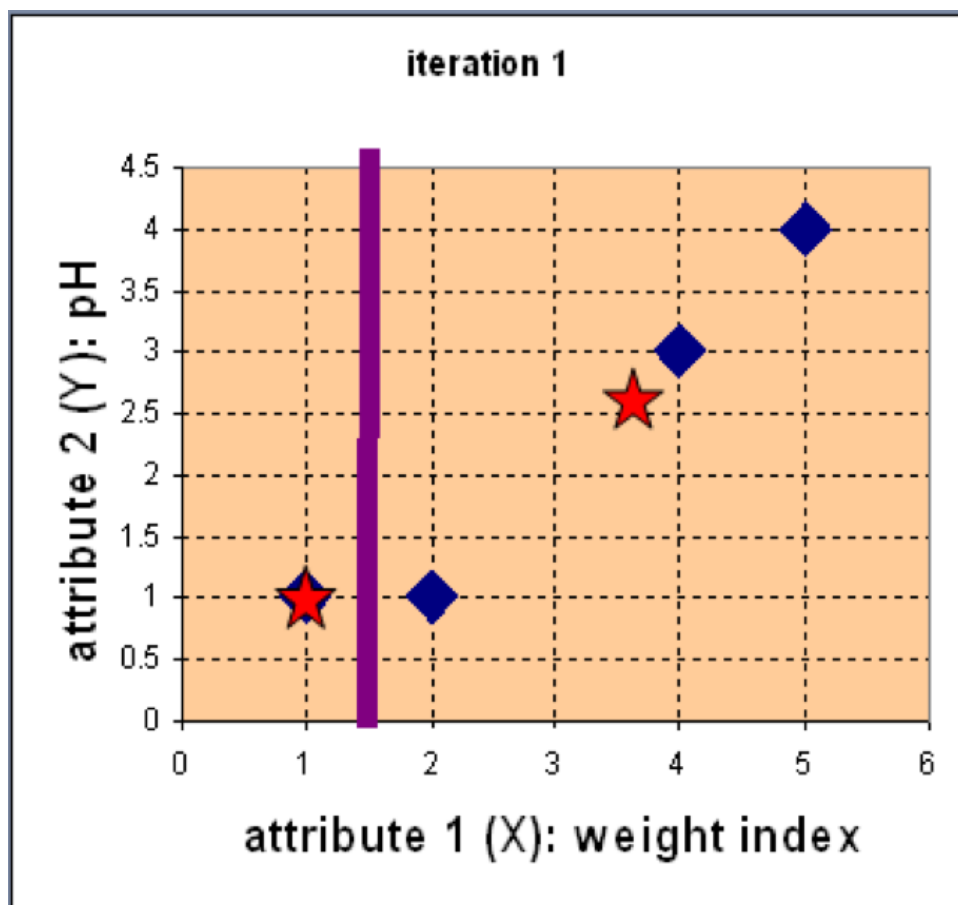
$$d(D, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

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

Assign each object to the cluster with the nearest seed point

# Example

- 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.

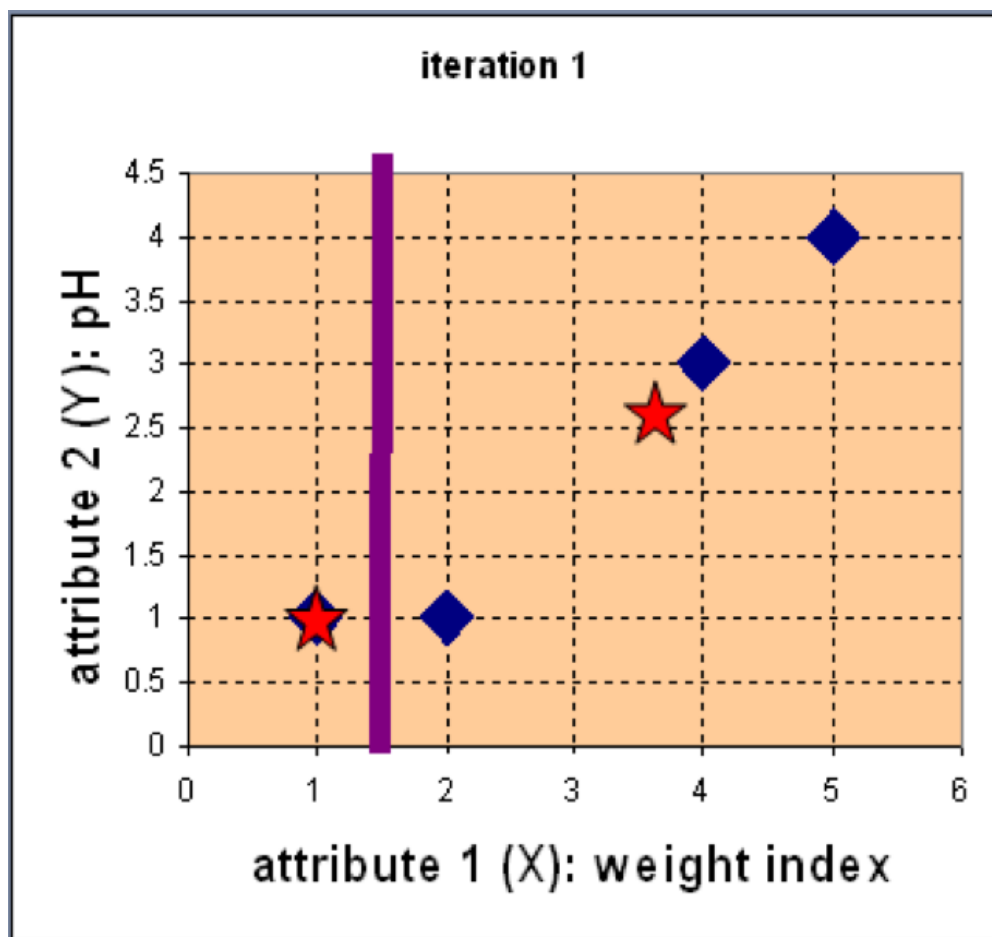
$$c_1 = (1, 1)$$

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



# Example

- Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

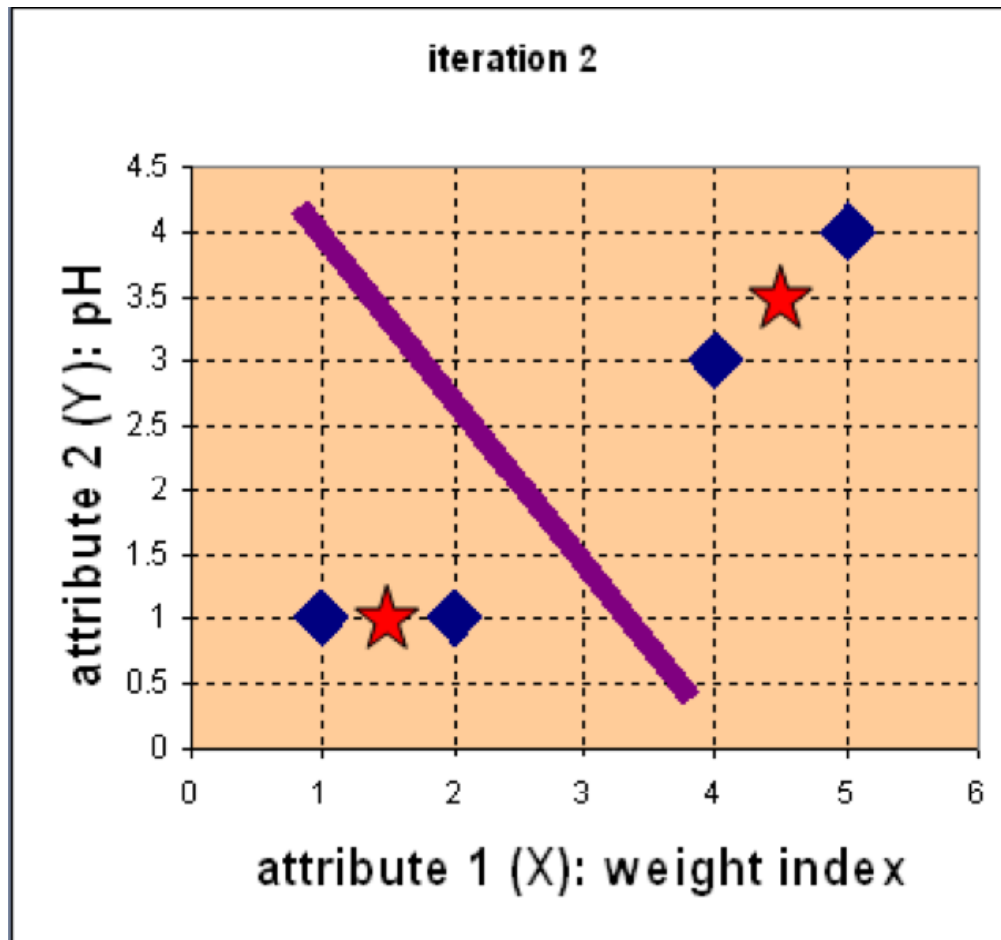
$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{array}{l} \mathbf{c}_1 = (1, 1) \text{ group-1} \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) \text{ group-2} \end{array}$$

	A	B	C	D	
	1	2	4	5	X
	1	1	3	4	Y

Assign the membership to objects

# Example

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



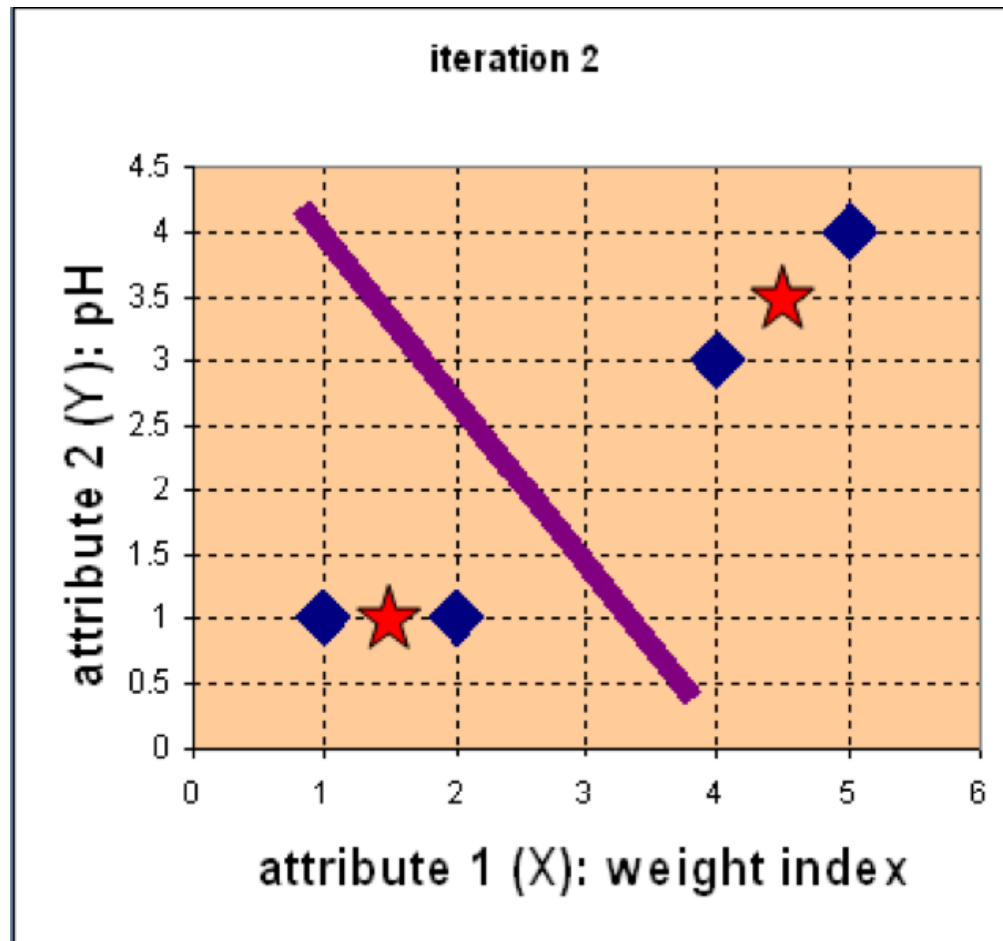
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

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

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

# Example

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



Compute the distance of all objects to the new centroids

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

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

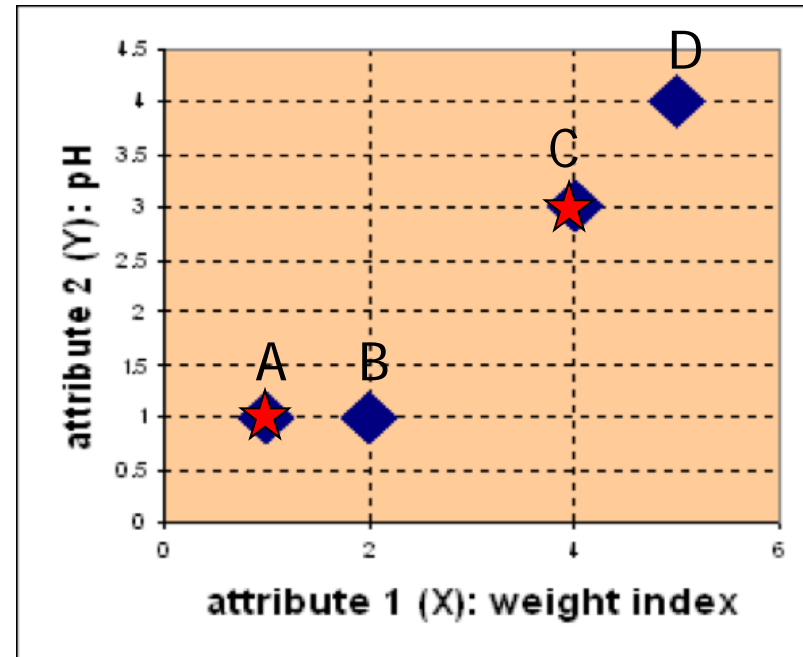
Stop due to no new assignment  
Membership in each cluster no longer change

# Exercise

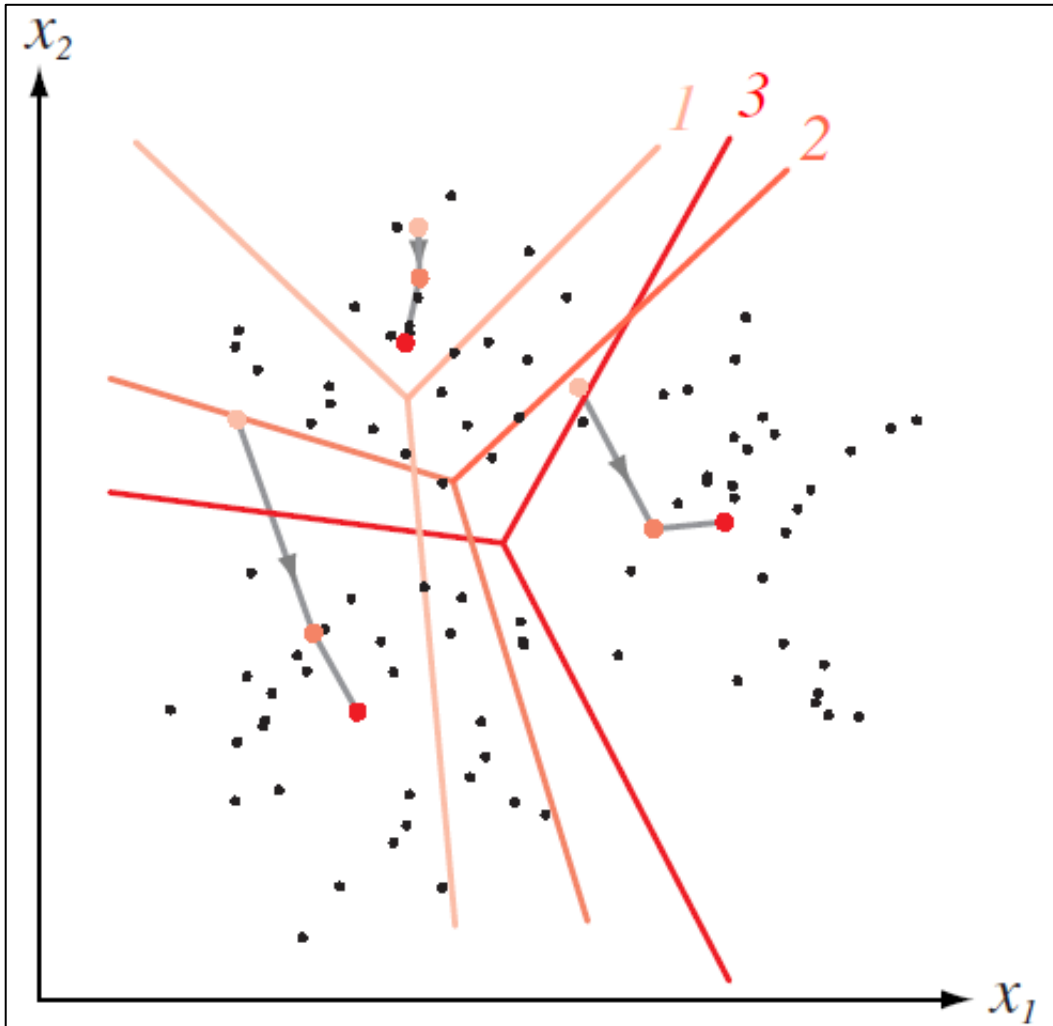
For the medicine data set, use K-means with the **Manhattan** distance metric for clustering analysis by setting  $K=2$  and initialising seeds as  $C_1 = A$  and  $C_2 = C$ . Answer three questions as follows:

1. How many steps are required for convergence?
2. What are memberships of two clusters after convergence?
3. What are centroids of two clusters after convergence?

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



# 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.

# K-means Demo

Clustering - K-means demo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://home.dei.polimi.it/matteucc/Clustering/tutorial\_html/AppletKM.html

Most Visited http://www.robertwhy...

Google Google Maps Bookmarks Check AutoFill

Options Dr.KeChen

Clustering - K-means demo

---

*A Tutorial on Clustering Algorithms*

---

[Introduction](#) | [K-means](#) | [Fuzzy C-means](#) | [Hierarchical](#) | [Mixture of Gaussians](#) | [Links](#)

---

**K-means - Interactive demo**

This applet requires Java Runtime Environment version 1.3 or later. You can download it from the [Sun Java website](#).

Data  Initialize Reset ☐ Show

Clusters  Start Step Run Euclidean

## K-means Demo

GETTING STARTED

- Choose how many data and clusters you want and then click on the **Initialize** button to generate them in random positions.  
OR  
Insert *manually* Data and Clusters using Right and Left mouse buttons. You can also delete them by clicking on them.
- Move data and centers of clusters as you like by clicking and dragging.
- Choose which *metric* the algorithm should use.
- Click on **Start** to begin the simulation. During simulation data and clusters positions are fixed.
- Go on using either **Step** or **Run** until the end of the simulation. Current number of steps is shown.
- Use the **Reset** button to go back to the initial configuration. Now you can move existing data and centers of clusters or generate new ones and then begin another simulation.
- When **Show History** is checked all the steps done until now are shown.

---

[Back to K-means](#)

---

Applet TestApplet started

start | Inbox for kchen@i... | moore-tutorials-fo... | K-mean | kmeans | PowerPoint Slide ... | Clustering - K-me... | EN | 17:35

# 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,  $K, t \ll n$ .
- 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

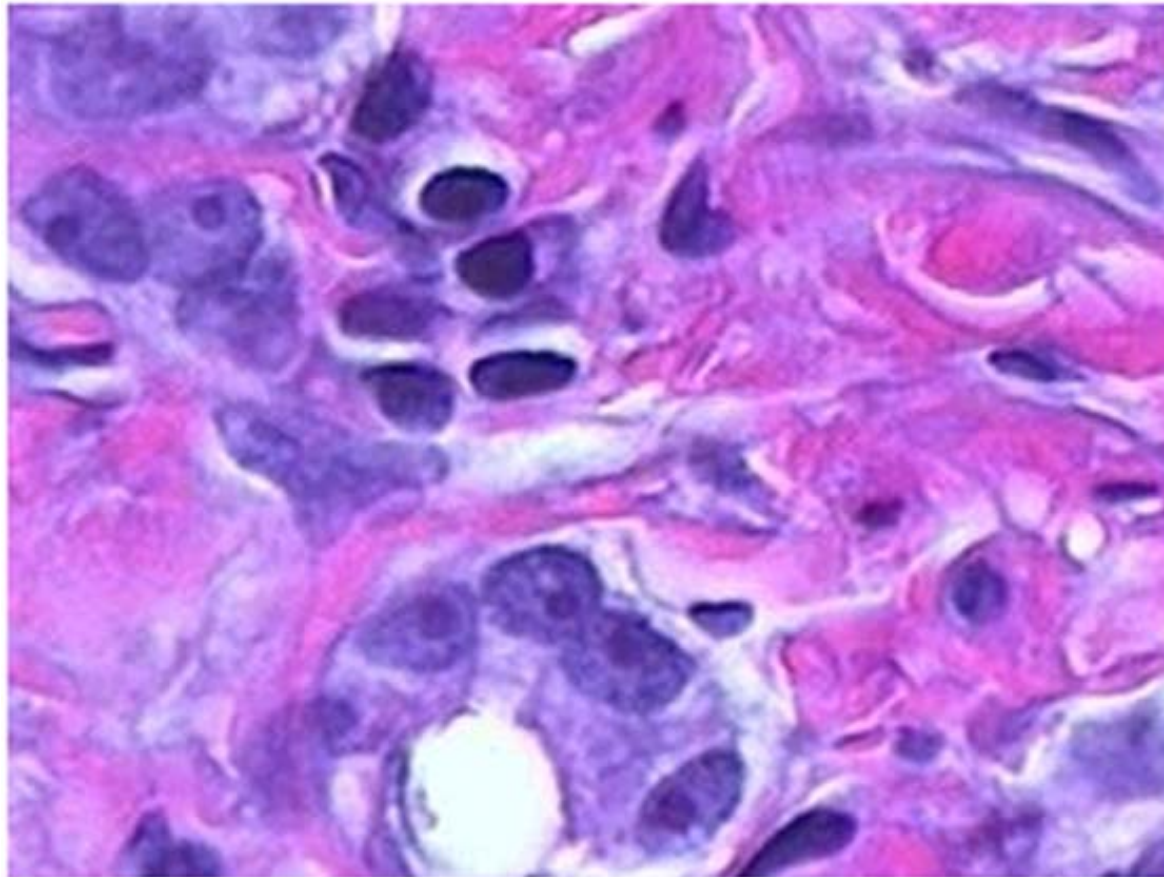


Image courtesy of Alan Partin, Johns Hopkins University



# 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

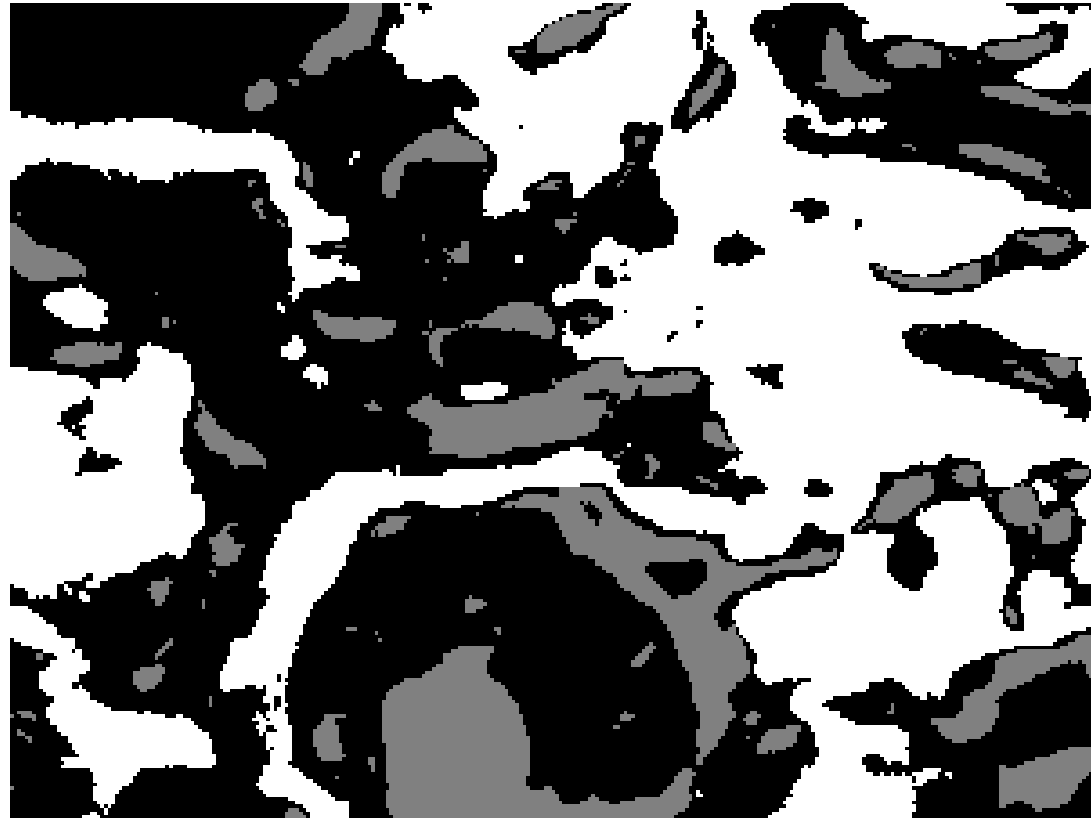
- In the  $L^*a^*b^*$  colour space, each pixel has a properties or feature vector:  $(L^*, a^*, b^*)$ .
- Like feature selection,  $L^*$  feature is discarded. As a result, each pixel has a feature vector  $(a^*, b^*)$ .
- Applying the  $K$ -means algorithm to the image in the  $a^*b^*$  feature space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

# Application

- Colour-Based Image Segmentation Using *K*-means

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

image labeled by cluster index



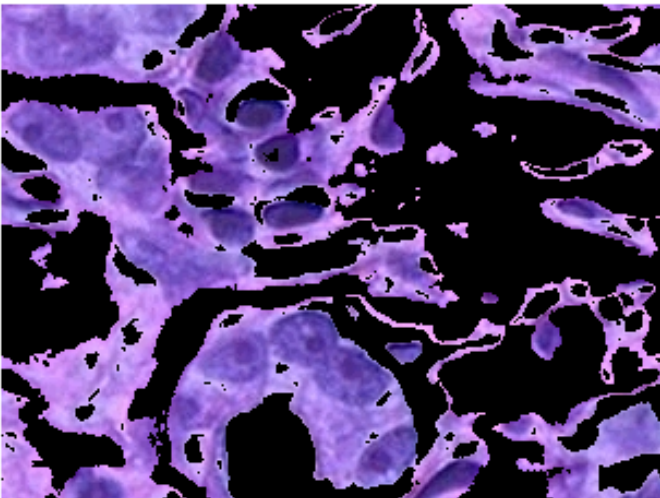
# Application

- Colour-Based Image Segmentation Using *K*-means

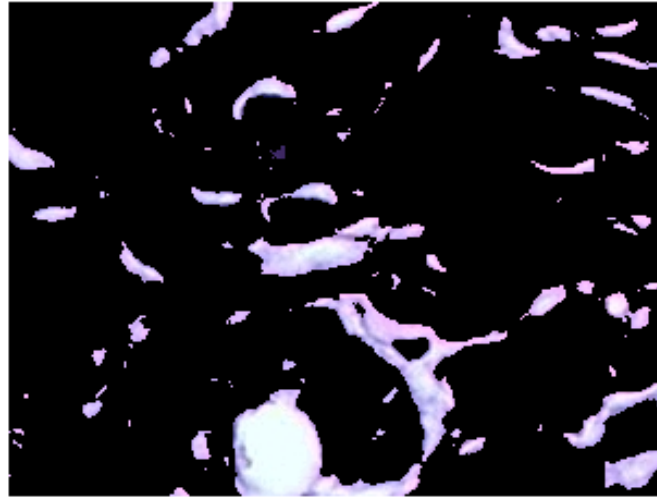
**Step 5:** Create Images that Segment the H&E Image by Colour

- Apply the label and the colour information of each pixel to achieve separate colour images corresponding to three clusters.

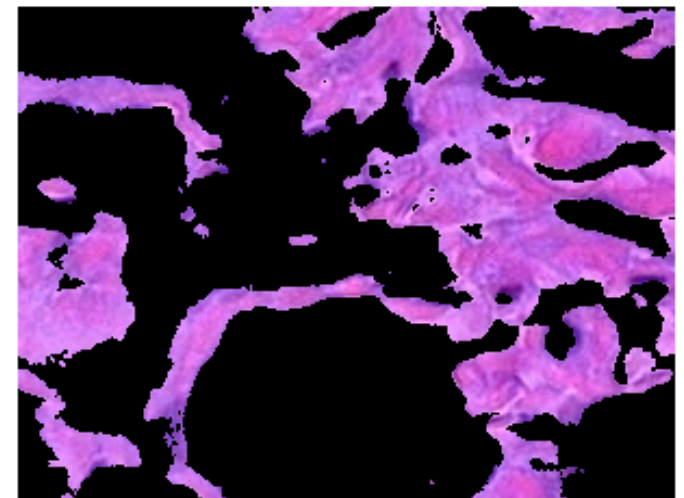
objects in cluster 1



objects in cluster 2



objects in cluster 3



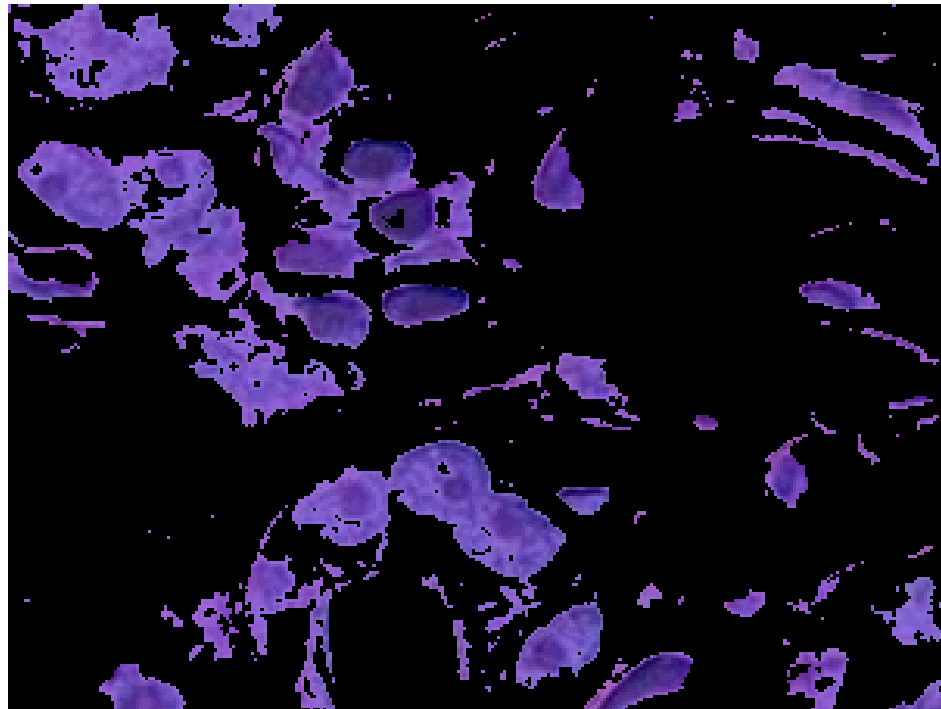
# Application

- Colour-Based Image Segmentation Using *K*-means

**Step 6:** Segment the nuclei into a separate image with the  $L^*$  feature

- In cluster 1, there are **dark** and **light blue** objects. The **dark blue** objects 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**



# 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

**Online tutorial:** the **K-means** function in Matlab

<https://www.youtube.com/watch?v=aYzjenNNOcc>