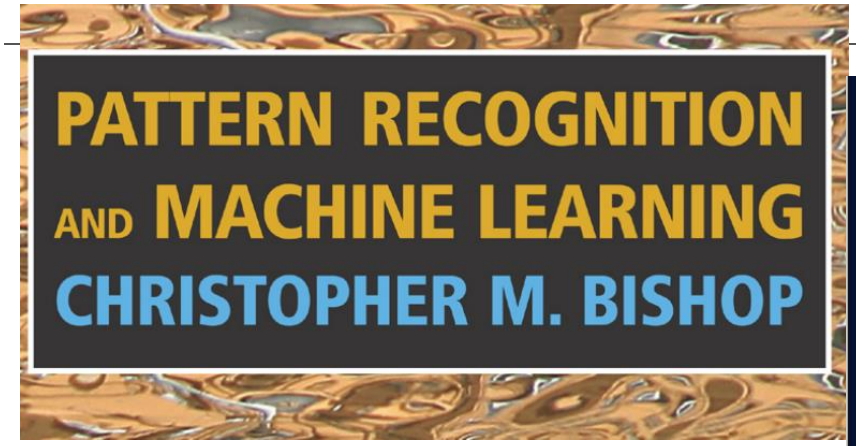
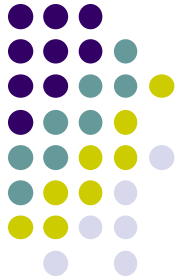
K-MEANS CLUSTERING

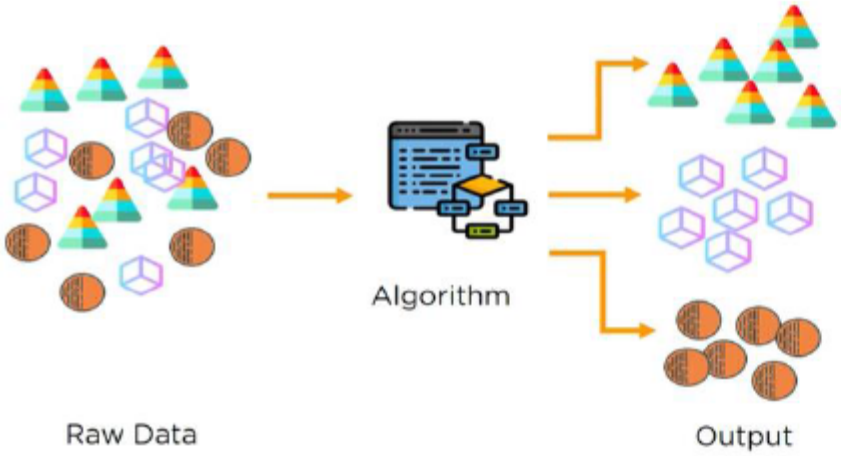


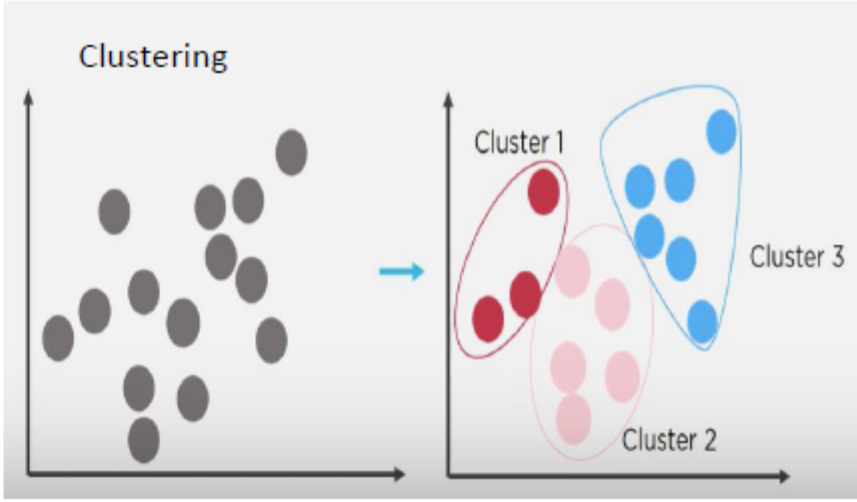
Unsupervised learning

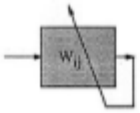


- Unsupervised learning is a type of algorithm that learns patterns from untagged data.
- Unsupervised learning or “learning without a teacher.” In this case one has a set of N observations (x1, x2, . . . , xN) of a random p-vector X having joint density Pr(X). The goal is to directly infer the properties of this probability density without the help of a supervisor or teacher providing correct answers or degree-of-error for each observation.
- Unsupervised learning is where you only have input data (X) and no corresponding output variables.
- The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.
- These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.

Unsupervised learning







Unsupervised learning

Finding the hidden structure from unlabled data

RV College of Engineering

Types of Unsupervised learning

Go, change the world

Clustering

Clustering is used for analyzing and grouping data which does not include pre-labeled class or even a class attribute at all

Algorithms used:

- K-means
- Hierarchical Clustering
- Hidden Markov model

Association

Discovers the probability of the co-occurrence of items in a collection

Algorithms used:

- Apriori algorithm
- FP-Growth

•An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Unsupervised learning applications

Learn clusters/groups without any label

Customer segmentation (i.e. grouping)

Image compression

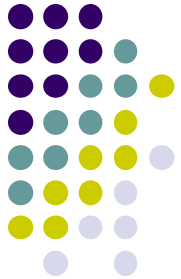
Bioinformatics: learn motifs

Find important features

Identifying Cancerous Data.

Prediction of Students' Academic Performance.

Drug Activity Prediction

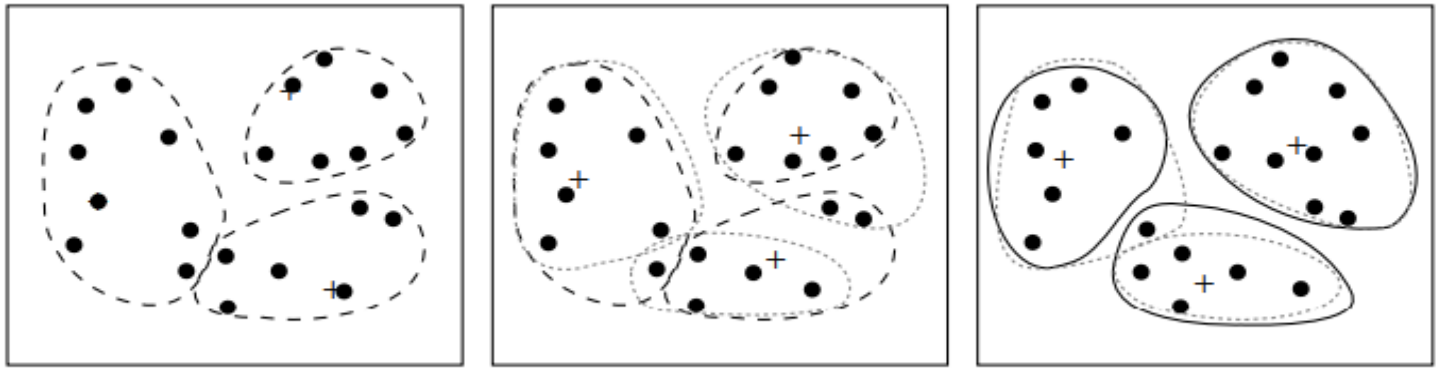


- Let us say we have an image that is stored with 24 bits/pixel and can have up to 16 million colors. Assume we have a color screen with 8 bits/pixel that can display only 256 colors. We want to find the best 256 colors among all 16 million colors such that the image using only the 256 colors color quantization in the palette looks as close as possible to the original image. This is *color quantization* where we map from high to lower resolution.

Three-dimensional clustering algorithm can be applied to color quantization

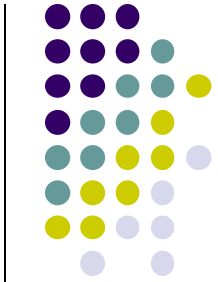
INTRODUCTION- What is clustering?

- **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.

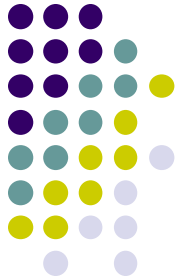


(a) Initial clustering (b) Iterate (c) Final clustering

Examples of Clustering Applications



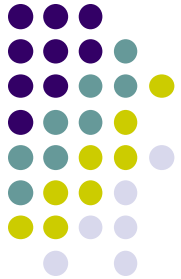
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with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



Clustering:

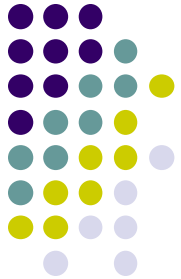
- Introduction
- Partitioning methods
- Hierarchical methods
- Model-based methods
- Density-based methods

Major Clustering Approaches



- Partitioning: Construct various partitions and then evaluate them by some criterion
- Hierarchical: Create a hierarchical decomposition of the set of objects using some criterion
- Model-based: Hypothesize a model for each cluster and find best fit of models to data
- Density-based: Guided by connectivity and density functions

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:

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

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

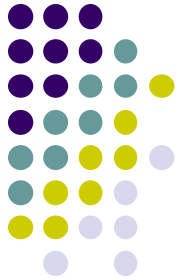
$$d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

3.The maximum norm is given by:

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

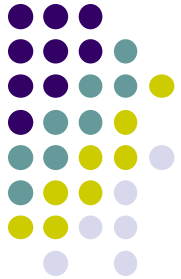
4. The Mahalanobis distance corrects data for different scales and correlations in the variables.
5. Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data
6. Hamming distance (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.

Good Clustering?



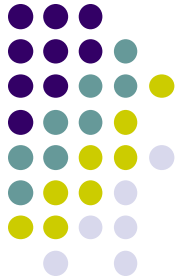
- A good clustering method will produce clusters with
  - High intra-class similarity
  - Low inter-class similarity
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

Requirements for Clustering in Data Mining



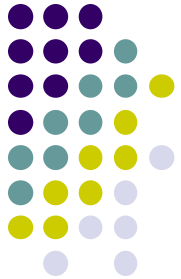
- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal domain knowledge required to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- Robustness wrt high dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Partitioning Algorithms

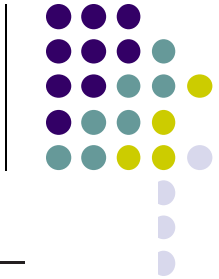


- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen, 1967): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw, 1987): Each cluster is represented by one of the objects in the cluster

K-Means Clustering



- Given *k*, the *k-means* algorithm consists of four steps:
  - Select initial centroids at random.
  - Assign each object to the cluster with the nearest centroid.
  - Compute each centroid as the mean of the objects assigned to it.
  - Repeat previous 2 steps until no change.



Algorithm 14.1 *K-means Clustering.*

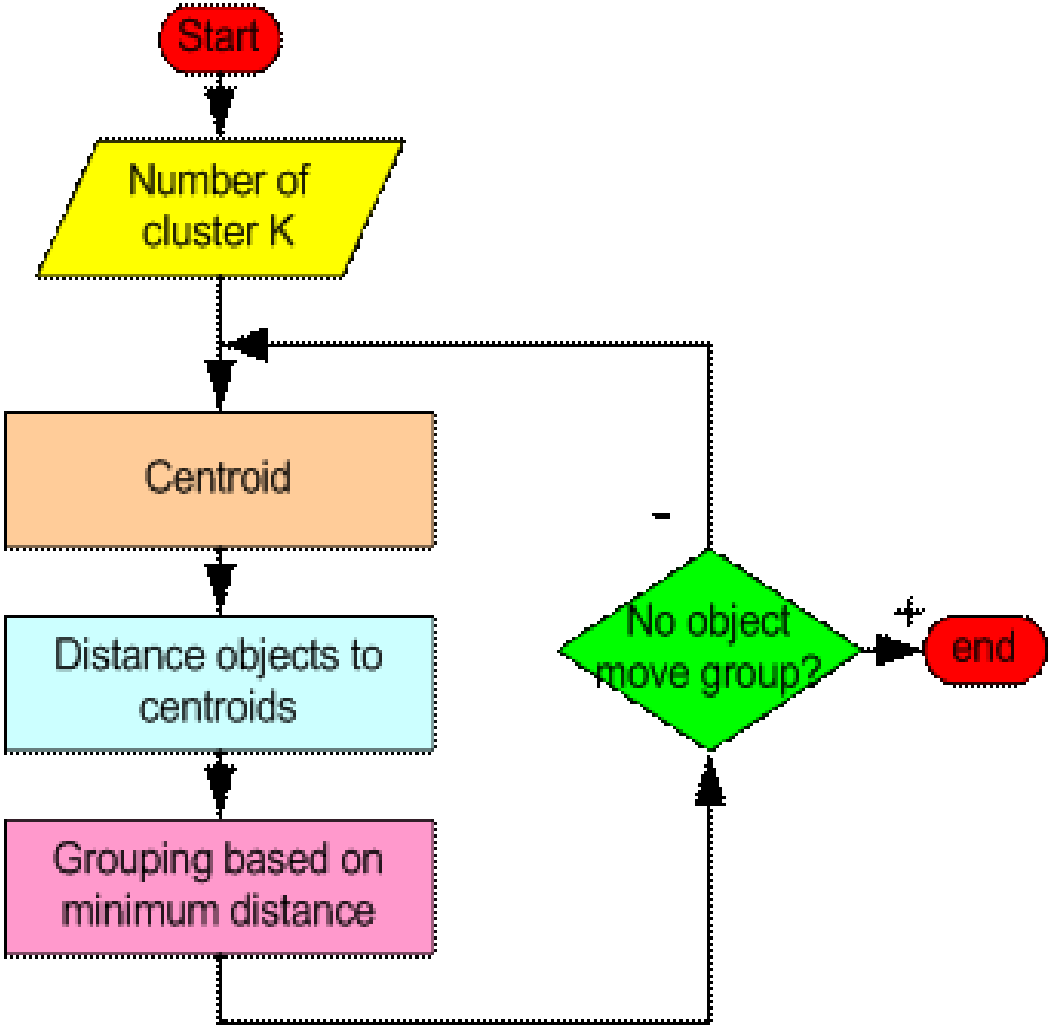
1. For a given cluster assignment C, the total cluster variance (14.33) is minimized with respect to $\{m_1, \ldots, m_K\}$ yielding the means of the currently assigned clusters (14.32).

2. Given a current set of means $\{m_1, \ldots, m_K\}$, (14.33) is minimized by assigning each observation to the closest (current) cluster mean. That is,

$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \|x_i - m_k\|^2. \tag{14.34}$$

3. Steps 1 and 2 are iterated until the assignments do not change.

How the K-Mean Clustering algorithm works?



Let the data set is: {x1,..., xN }

Objective is:
To partition the data set into some number K of clusters.
To find an assignment of data points to clusters
To find mean / centroid of vectors {μk}, such that the sum of the squares of the distances of each data point to its closest vector μk, is a minimum.

The objective function(distortion measure) is given by

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

binary indicator variables $r_{nk} \in \{0, 1\}$,

EM:

- Two-stage optimization is then repeated until convergence.
- Updating r and updating μk
- E -expectation
- M -maximization steps of the EM algorithm

Christopher

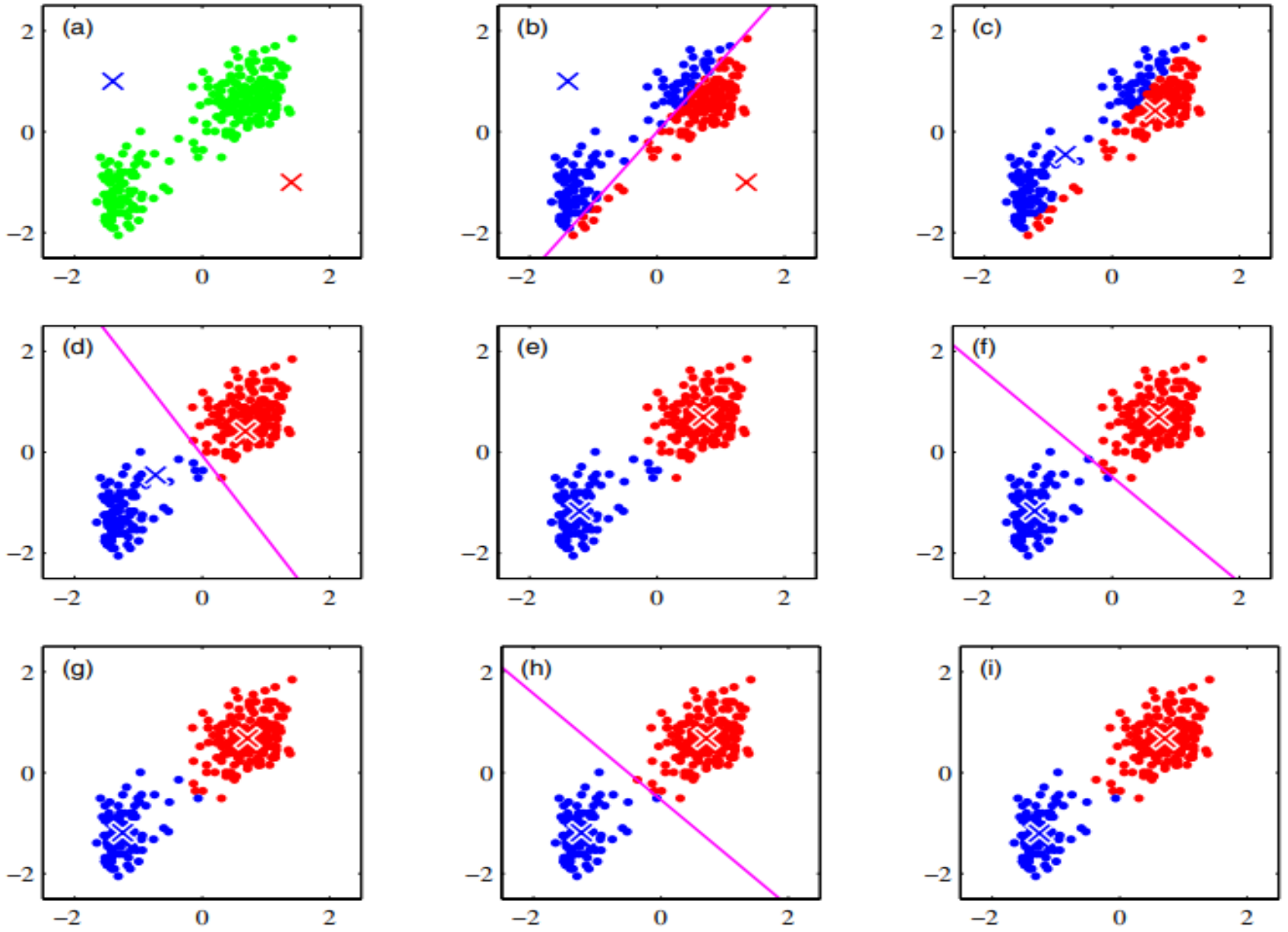


Figure 9.1 Illustration of the K-means algorithm using the re-scaled Old Faithful data set. (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are shown by the red and blue crosses, respectively. (b) In the initial E step, each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on. (c) In the subsequent M step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster. (d)–(i) show successive E and M steps through to final convergence of the algorithm.

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:

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

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

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

$$= (4.12, 5.38)$$

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

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

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}

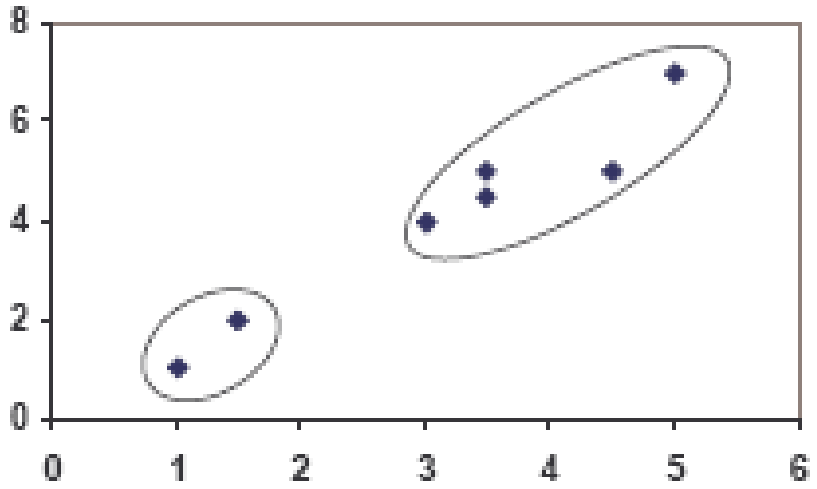
- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

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

- A = (1, 1), B = (1, 2), C = (2, 2), D = (6, 2), E = (7, 2), F = (6, 6), G = (7, 6)
- K=2 $C_1^{(0)} = (1,1)$ $C_2^{(0)} = (1,2)$

			$D_1^{(1)}$	$D_2^{(1)}$	$D_1^{(2)}$	$D_2^{(2)}$	$D_1^{(3)}$	$D_2^{(3)}$
A	1	1	0	1	0	4.486		
B	1	2	1	0	1	4.05		
C	2	2	1.414	1	1.414	3.129		
D	6	2	5.1	5	5.1	1.769		
E	7	2	6.08	6	6.08	2.542		
F	6	6	7.07	6.4	7.07	2.94		
G	7	6	7.81	7.21	7.81	3.646		

1st iteration:

B_1 $\sqrt{(1-1)^2 + (1-2)^2} = 1$

$G_1^{(1)} = 1$, $G_2^{(1)} = \{B, C, D, E, F, G\}$

$C_1^{(1)} = 1$ $C_2^{(1)} = \frac{2+2+2+2+6+6}{6} = 3.3$

$C_1^{(1)} = (1,1)$ $C_2^{(1)} = (4.83, 3.3)$

2nd iteration.

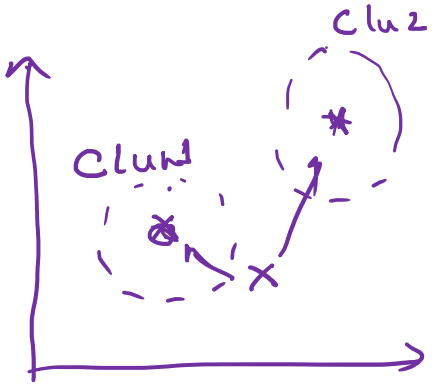
$G_1^{(2)} = (A, B, C)$

$G_2^{(2)} = (D, E, F, G)$

$C_1^{(2)} = \left(\frac{4}{3}, \frac{5}{3}\right)$

$C_2^{(2)} = \left(\frac{26}{4}, \frac{16}{4}\right)$

$X = [2.5, 4.5]$



(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

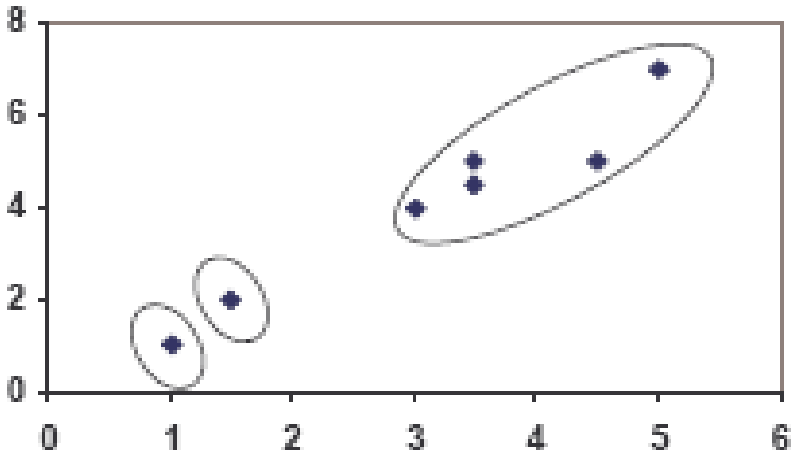
C_3

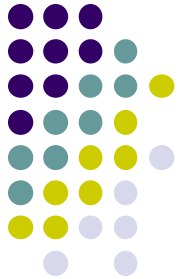
clustering with initial centroids (1, 2, 3)

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

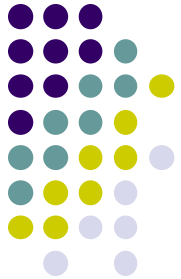
Step 2





- A = (1, 1), B = (1, 2), C = (2, 2), D = (6, 2), E = (7, 2), F = (6, 6), G = (7, 6)

- K=3



- Consider the collection of three-dimensional patterns: (1, 1, 1), (1, 2, 1), (1, 1, 2), (6, 6, 1), (6, 7, 1), (7, 6, 1).
- Find the 2-partition obtained by the k-means algorithm with k = 2.

(K = 3) patterns
Initial centriods
(10, 3.5, 2.0), (63, 5.4, 1.3), (10.4, 3.5, 2.1)

After iteration-1

Cluster1: {(10, 3.5, 2.0)}

Cluster2: {(63, 5.4, 1.3), (73.5, 5.8, 1.2), (81, 6.1, 1.3), (71, 6.4, 1.0)}

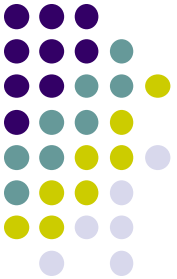
Cluster3: {(10.4, 3.5, 2.1), (10.3, 3.3, 2.0), (10.4, 3.3, 2.3), (10.4, 3.5, 2.3), (10.5, 3.3, 2.1)}

Cluster Centroids: (10, 3.5, 2.0), (72.1, 5.9, 1.2), (10.4, 3.4, 2.2

Table 7.1. A dataset of 10 patterns.

Pattern numbers	Feature1	Feature2	Feature3
1	10	3.5	2.0
2	63	5.4	1.3
3	10.4	3.5	2.1
4	10.3	3.3	2.0
5	73.5	5.8	1.2
6	81	6.1	1.3
7	10.4	3.3	2.3
8	71	6.4	1.0
9	10.4	3.5	2.3
10	10.5	3.3	2.1

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

Let and c_1 and c_2 denote the coordinate of the centroids, then c_1=(1,1) and c_2=(2,1)

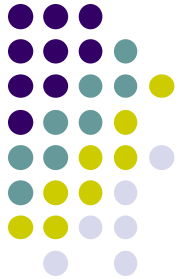
$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (2,1) & group-2 \end{matrix}$$

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

$$)^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

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

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



We 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

A drawback of k-means. (Pg No 454 – Data mining)

Consider six points in 1-D space having the values 1,2,3,8,9,10, and 25, respectively. Intuitively, by visual inspection we may imagine the points partitioned into the clusters {1,2,3} and {8,9,10}, where point 25 is excluded because it appears to be an outlier.

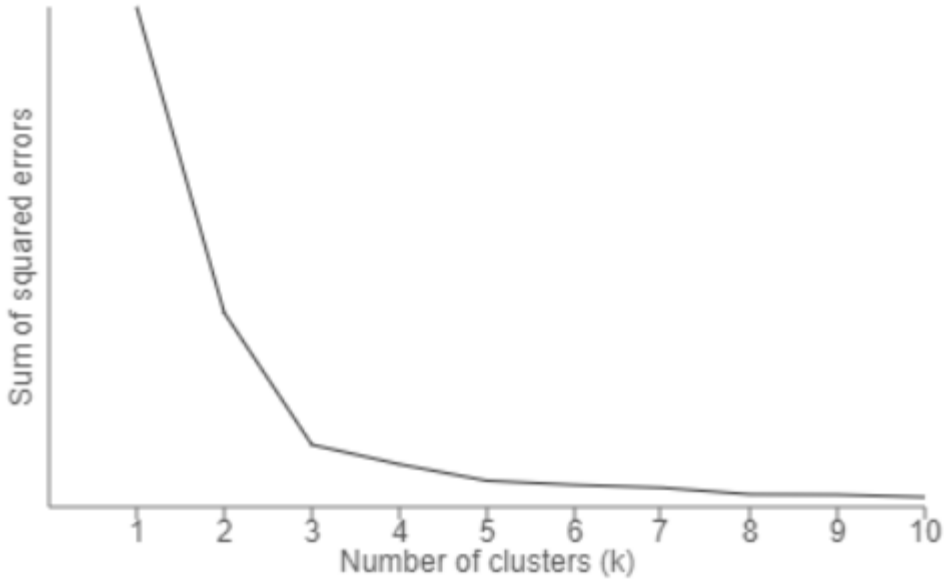
How would k-means partition the values?

If we apply k-means using k = 2 the partitioning {{1,2,3},{8,9,10,25}} has the within-cluster variation (1 − 2) 2 + (2 − 2) 2 + (3 − 2) 2 + (8 − 13) 2 + (9 − 13) 2 + (10 − 13) 2 + (25 − 13) 2 = 196, given that the mean of cluster {1,2,3} is 2 and the mean of {8,9,10,25} is 13.

Compare this to the partitioning {{1,2,3,8},{9,10,25}} for which k-means computes the within cluster variation as (1 − 3.5) 2 + (2 − 3.5) 2 + (3 − 3.5) 2 + (8 − 3.5) 2 + (9 − 14.67) 2 + (10 − 14.67) 2 + (25 − 14.67) 2 = 189.67

given that 3.5 is the mean of cluster {1,2,3,8} and 14.67 is the mean of cluster {9,10,25}. The latter partitioning has the lowest within-cluster variation; therefore, the k-means method assigns the value 8 to a cluster different from that containing 9 and 10 due to the outlier point 25. Moreover, the center of the second cluster, 14.67, is substantially far from all the members in the cluster.

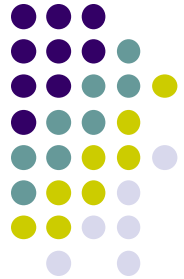
Choosing K



Comments on the *K-Means* Method

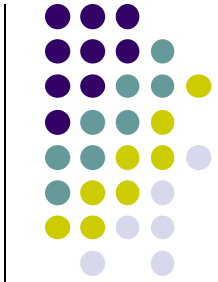
- Strengths
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as *simulated annealing* and *genetic algorithms*
- Weaknesses
  - Applicable only when *mean* is defined (what about categorical data?)
  - Need to specify *k,* the *number* of clusters, in advance
  - Trouble with noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Weaknesses of K-Mean 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-Mean 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.*

CONCLUSION

- *K-means algorithm is* useful for undirected knowledge discovery and is relatively simple. K-means has found wide spread usage in lot of fields, ranging from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligence, image processing, machine vision, and many others.