ENCS3340 - Artificial Intelligence

Unsupervised Learning

Unsupervised Learning

1 - Clustering

Unsupervised Learning: Clustering Introduction

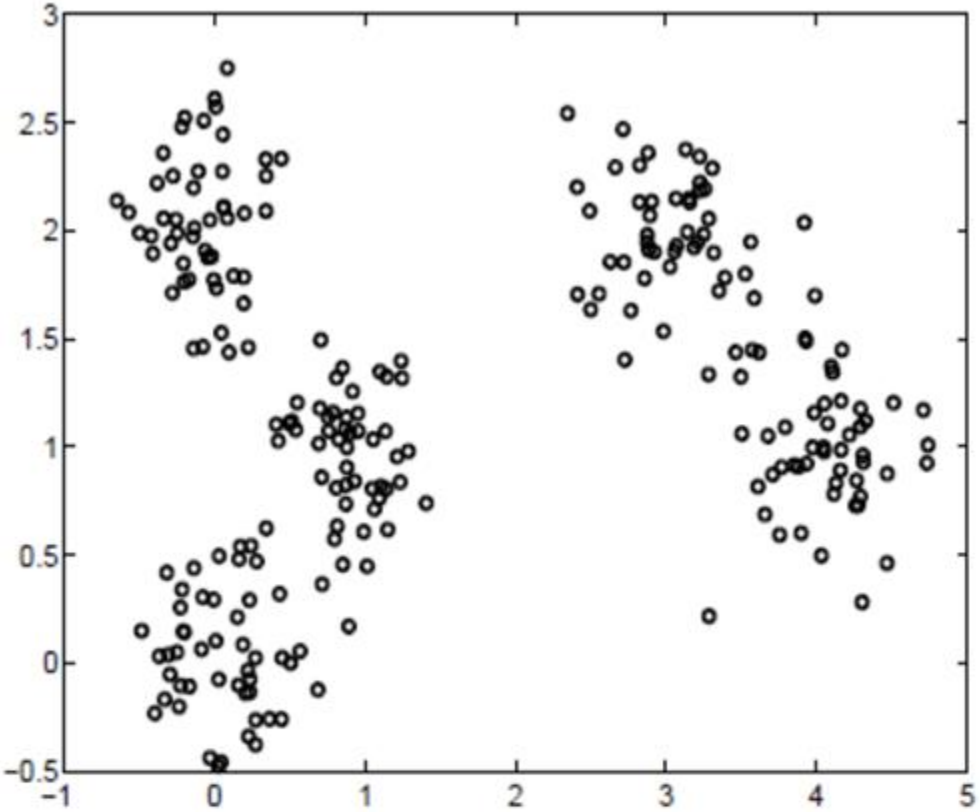
- Cluster: A collection/group of data objects/ points such that:
  - **similar** (related) to one another in same group
  - **dissimilar** (unrelated) to the objects in other groups
- Cluster Analysis
  - find ***similarities*** between data according to characteristics underlying the data and grouping similar data objects into clusters

Unsupervised Learning: Clustering

- Clustering Analysis: Unsupervised learning
  - No predefined classes for a training data set
  - Two general tasks:
    - identify the "natural" clusters **number** and
    - properly grouping objects into **"sensible"** clusters
- Typical applications
  - as a stand-alone tool to gain an insight into data distribution
  - as a preprocessing step of other algorithms in intelligent systems

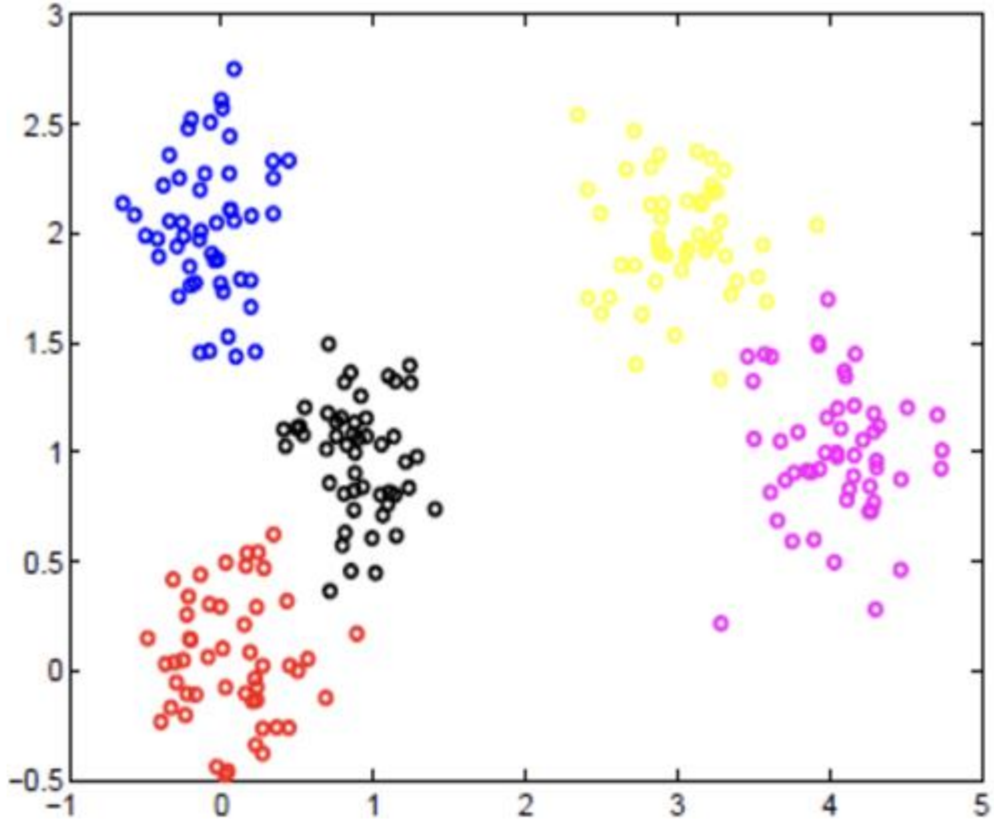
Introduction

- Illustrative Example 1: how many clusters?



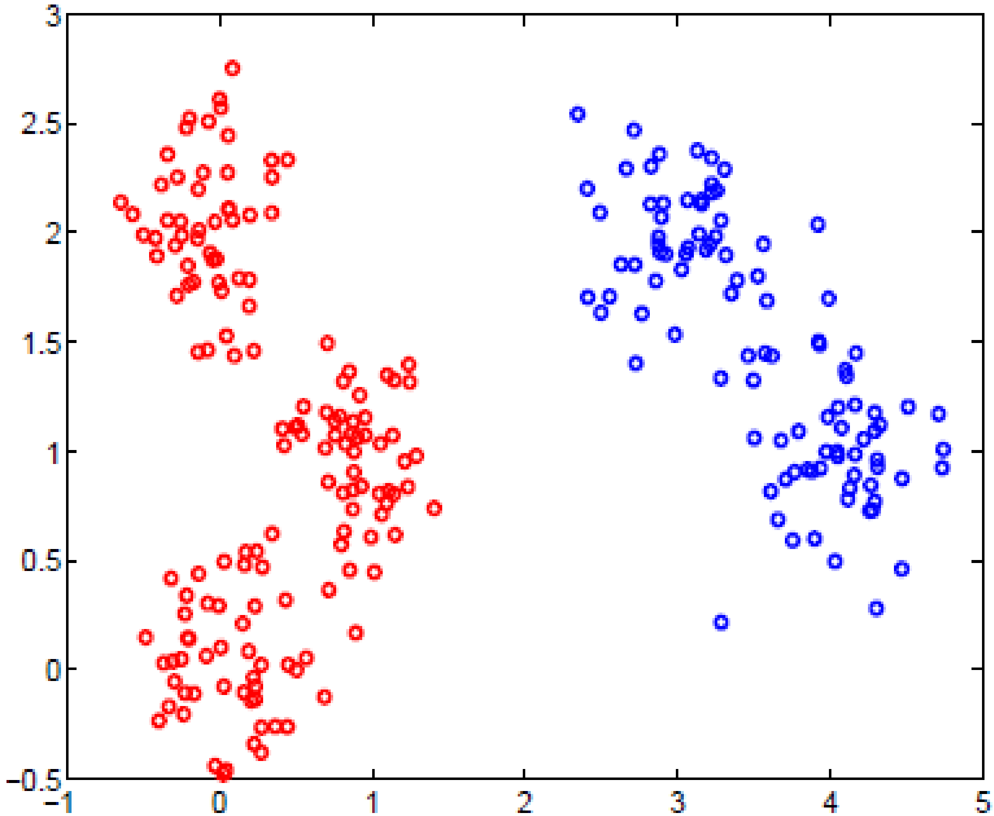
Introduction

- Illustrative Example 1: how many clusters?



Introduction

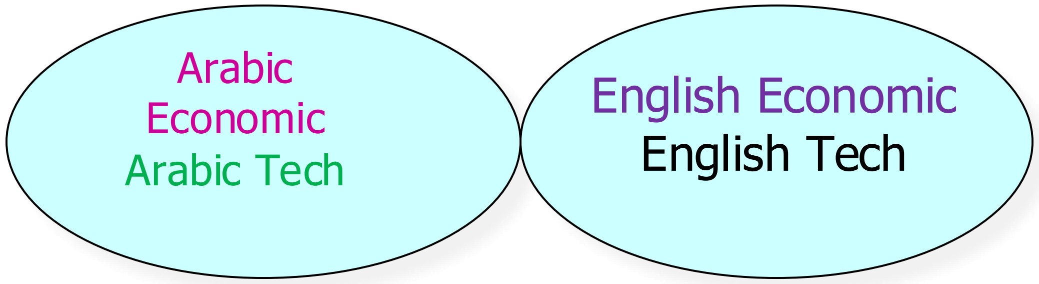
- Illustrative Example 1: how many clusters?



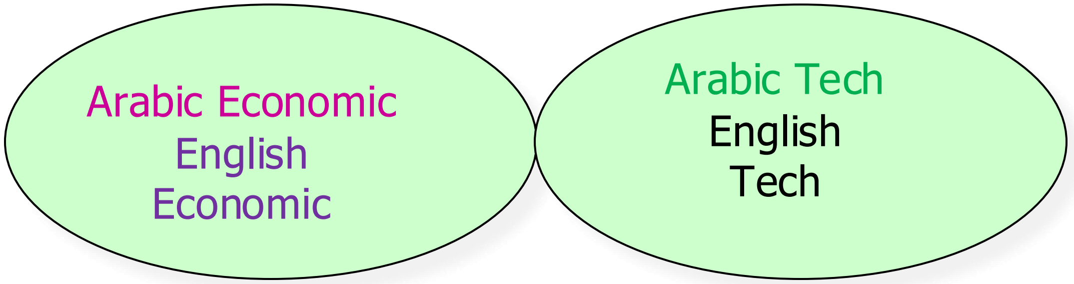
Introduction (Cont.)

- Illustrative Example 2: are they in the same cluster?

  Which Features are important?



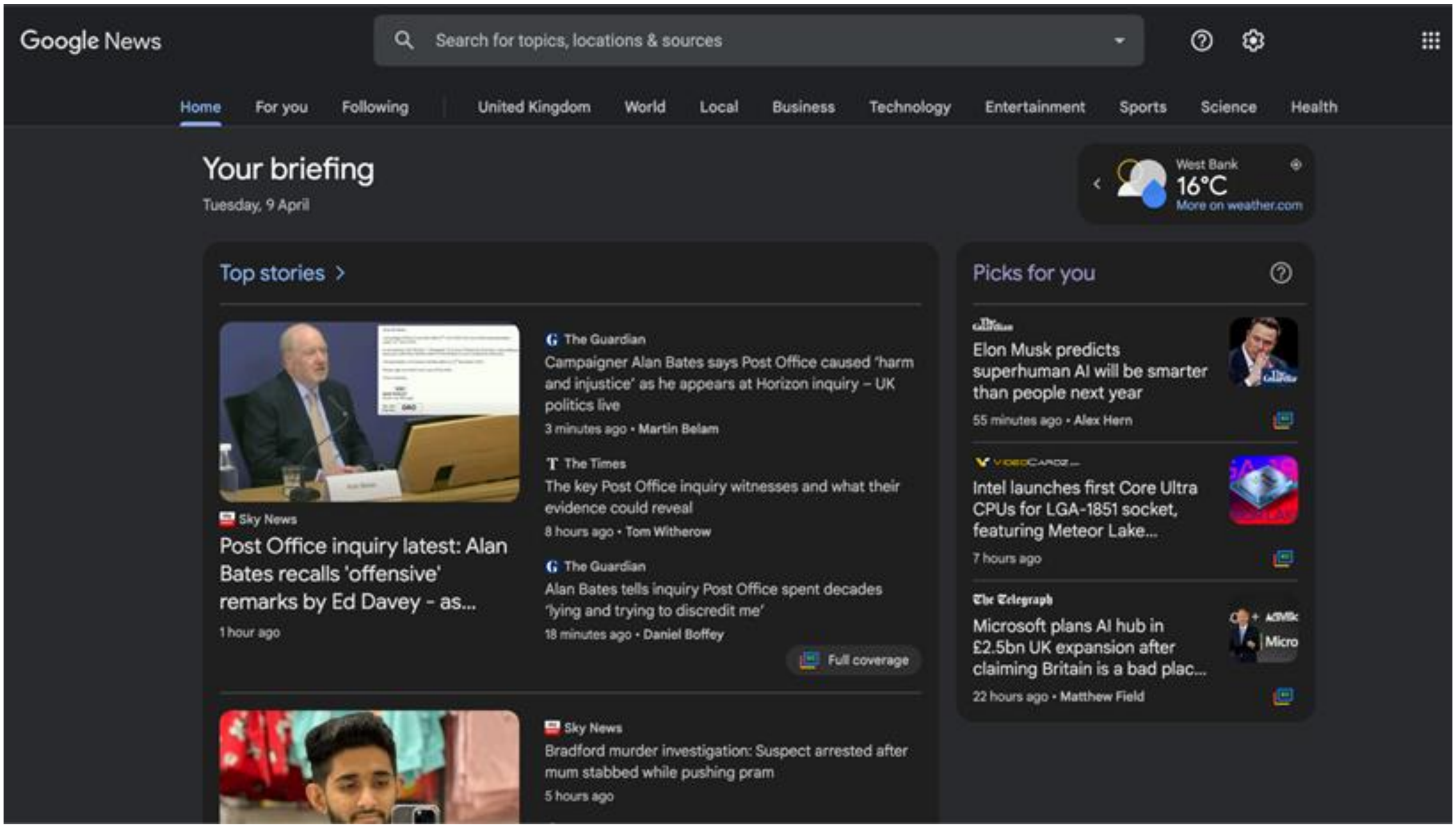
1. Two clusters
2. Clustering criterion:
   Article Language



1. Two clusters
2. Clustering criterion:
   Article Topic

Introduction (Cont.)

- Real Applications: [Google News]



Introduction (Cont.)

- Real world tasks:
  - Bank/Internet Security: fraud/spam pattern discovery
  - Biology: taxonomy of living things such as kingdom, phylum, class, order, family, genus and species
  - City-planning: Identifying groups of houses according to their house type, value, and geographical location
  - Climate change: understanding earth climate, find patterns of atmospheric and ocean
  - Finance: stock clustering analysis to uncover correlation underlying shares
  - Image Compression/segmentation: coherent pixels grouped
  - Information retrieval/organisation: Google search, topic-based news
  - Land use: Identification of areas of similar land use in an earth observation database
  - Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
  - Social network mining: special interest group automatic discovery

Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
  - …
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance ⇒ maximized
  - Intra-clusters distance ⇒ minimized
- Clustering Quality depends on algorithm, distance function, and application.

Data Types and Representations

- **Discrete vs. Continuous**
  - Discrete Feature
    - Has only a finite set of value e.g., zip codes, rank, or the set of words in a collection of documents
    - Sometimes, represented as integer variable
  - Continuous Feature
    - Real numbers as feature values e.g. temperature, height, or weight, location: practically measured and represented using a finite number of digits
    - Typically represented as floating-point variables

Data Types and Representations

- Data representations
  - **Data** matrix (object-by-feature structure)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- n data points (objects) with p dimensions (features)
- Two modes: row and column represent different entities
- E.g. Document/word matrix

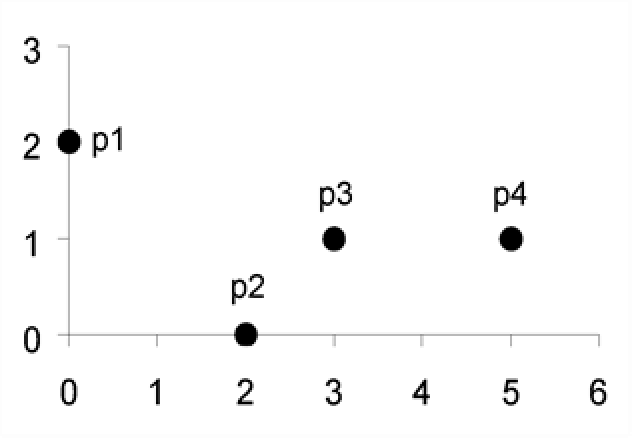
Distance/dissimilarity matrix: object-by-object structure

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

- n data points, but registers only the distance
- A symmetric/triangular matrix
- Single mode: row and column for the same entity (distance)

Data Types and Representations

- Examples



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Data Matrix

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix (i.e., Dissimilarity Matrix) for Euclidean Distance

Distance Measures

- Minkowski Distance (http://en.wikipedia.org/wiki/Minkowski_distance)

  For $\mathbf{x} = (x_1\ x_2 \cdots x_n)$ and $\mathbf{y} = (y_1\ y_2 \cdots y_n)$

$$d(\mathbf{x}, \mathbf{y}) = \left(|x_1 - y_1|^p + |x_2 - y_2|^p \cdots + |x_n - y_n|^p \right)^{\frac{1}{p}}, \quad p > 0$$

  - $p = 1$: Manhattan (city block) distance

$$d(\mathbf{x}, \mathbf{y}) = |x_1 - y_1| + |x_2 - y_2| \cdots + |x_n - y_n|$$

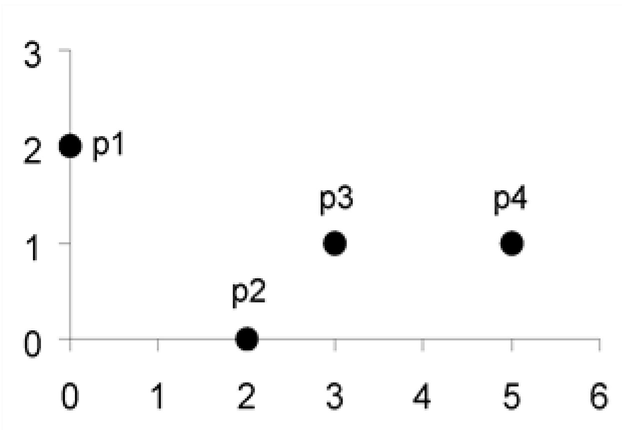
  - $p = 2$: Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2 \cdots + |x_n - y_n|^2}$$

  - Do not confuse p with n, i.e., all these distances are defined based on all numbers of features (dimensions).
  - A generic measure: use appropriate p in different applications

Distance Measures

- Example: Manhatten and Euclidean distances



L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

Distance Matrix for Manhattan Distance

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Data Matrix

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix for Euclidean Distance

Distance Measures

- Cosine Measure (Similarity vs. Distance)

For $\mathbf{x} = (x_1\, x_2 \cdots x_n)$ and $\mathbf{y} = (y_1\, y_2 \cdots y_n)$

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{x_1 y_1 + \cdots + x_n y_n}{\sqrt{x_1^2 + \cdots + x_n^2}\,\sqrt{y_1^2 + \cdots + y_n^2}}$$

$$d(\mathbf{x}, \mathbf{y}) = 1 - \cos(\mathbf{x}, \mathbf{y})$$

$$0 \leq d(\mathbf{x}, \mathbf{y}) \leq 2$$

- *x=(1,0,1), y=(0,1,1): cos(x,y)=1/2*
- Property:
    - Nonmetric vector objects: keywords in documents, gene features in micro-arrays, …
    - Applications: information retrieval, biologic taxonomy

Distance Measures

- Example: Cosine measure

$$\mathbf{x}_1 = (3, 2, 0, 5, 2, 0, 0), \mathbf{x}_2 = (1,0, 0, 0, 1, 0, 2)$$

$$3\times1+2\times0+0\times0+5\times0+2\times1+0\times0+0\times2=5$$

$$\sqrt{3^2+2^2+0^2+5^2+2^2+0^2+0^2}=\sqrt{42}\approx 6.48$$

$$\sqrt{1^2+0^2+0^2+0^2+1^2+0^2+2^2}=\sqrt{6}\approx 2.45$$

$$\cos(\mathbf{x}_1,\mathbf{x}_2)=\frac{5}{6.48\times2.45}\approx 0.32$$

$$d(\mathbf{x}_1,\mathbf{x}_2)=1-\cos(\mathbf{x}_1,\mathbf{x}_2)=1-0.32=0.68$$

Unsupervised Learning

2 - K-means Clustering

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", **or centroid**, 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

- The *K-means* algorithm: a heuristic method
  - K-means algorithm (MacQueen'67): each cluster is represented by the center of the cluster and the algorithm converges to stable centroids of clusters.
  - K-means algorithm is the simplest partitioning method for clustering analysis: widely used in data mining applications.

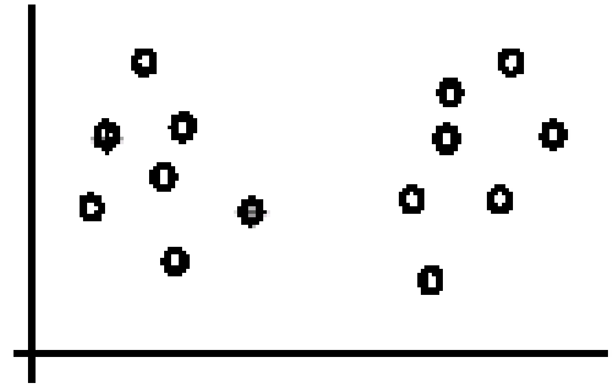
K-means Algorithm

- Given the cluster number *K*, the *K-means* algorithm works as follows:

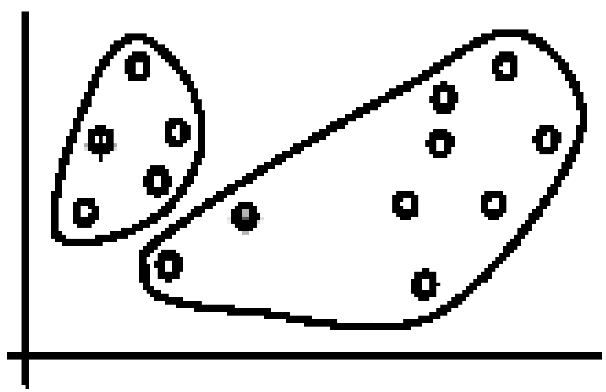
0) Initialisation: set initial **K** seed points –centroids- (randomly)

1) Assign each object to the cluster of the nearest **seed** point using the specific **distance metric**
2) Compute the new seed points as the centroids of the clusters of the current partition (the centroid is the center, or *mean point*, of the cluster after additions)
3) Stop when no more new assignment (i.e., membership in each cluster no longer changes) else Go back to Step

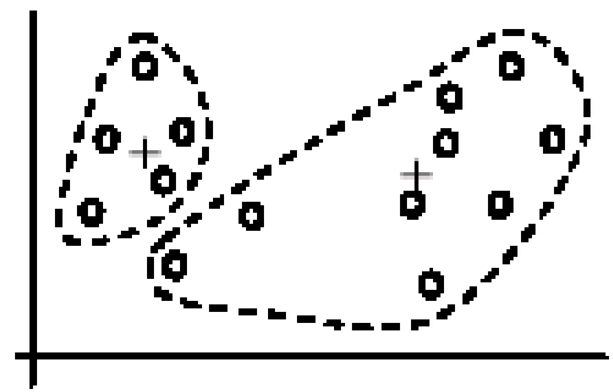
An example



(A). Random selection of *k* centers

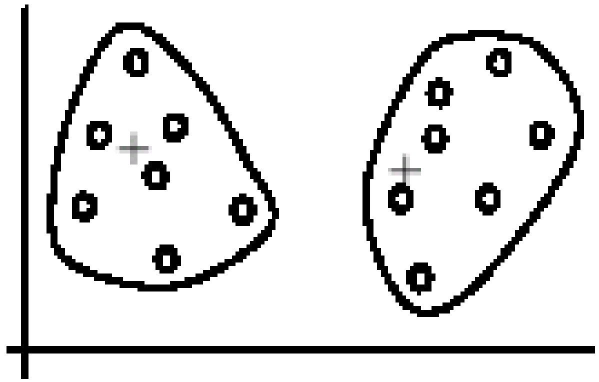


Iteration 1: (B). Cluster assignment

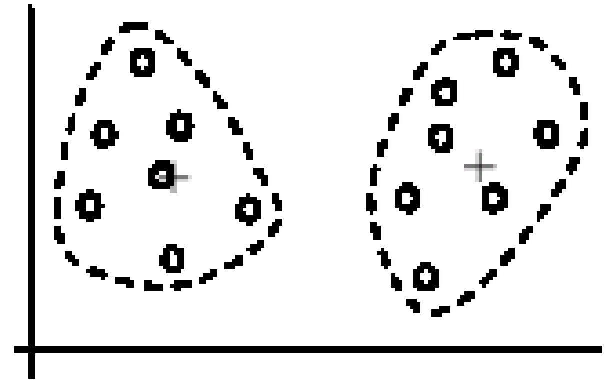


(C). Re-compute centroids

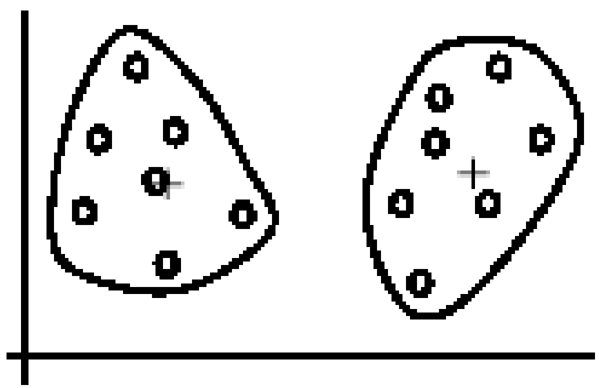
An example (cont …)



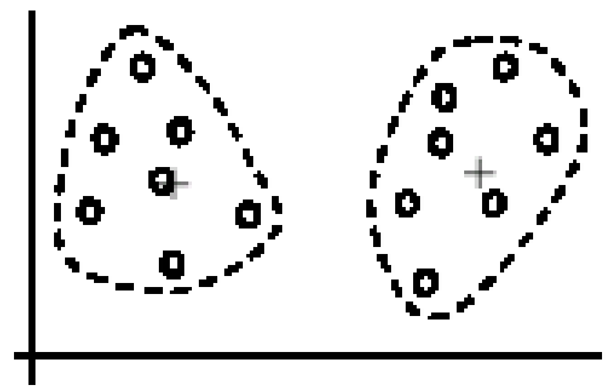
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

Stopping/convergence criterion

1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE) over all clusters,

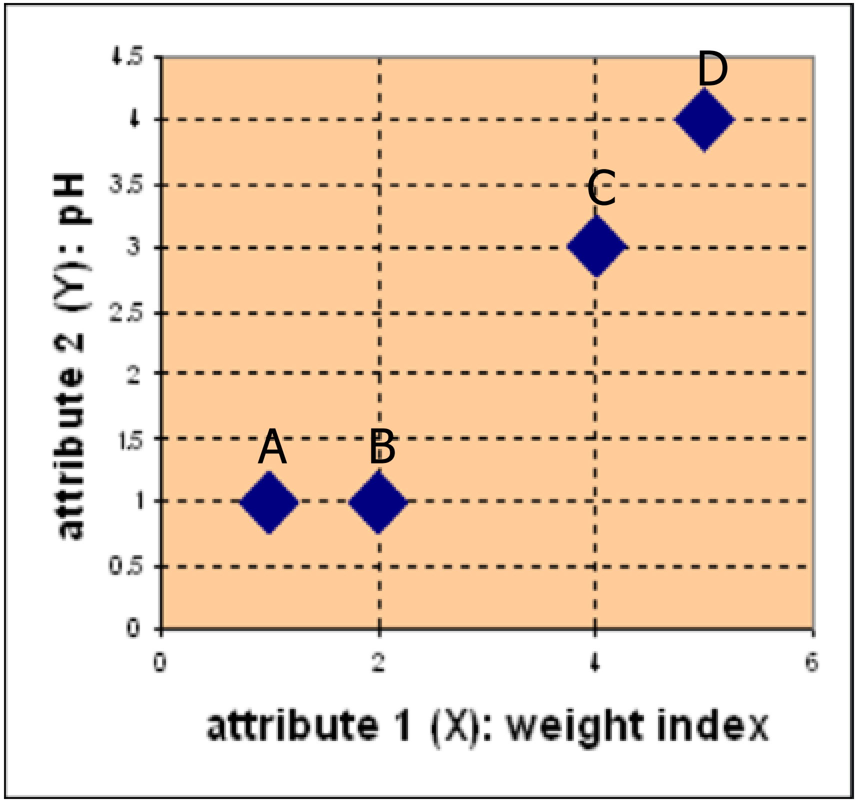
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

a. C_j is the *jth* cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and *dist*($\mathbf{x}$, $\mathbf{m}_j$) is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

Example

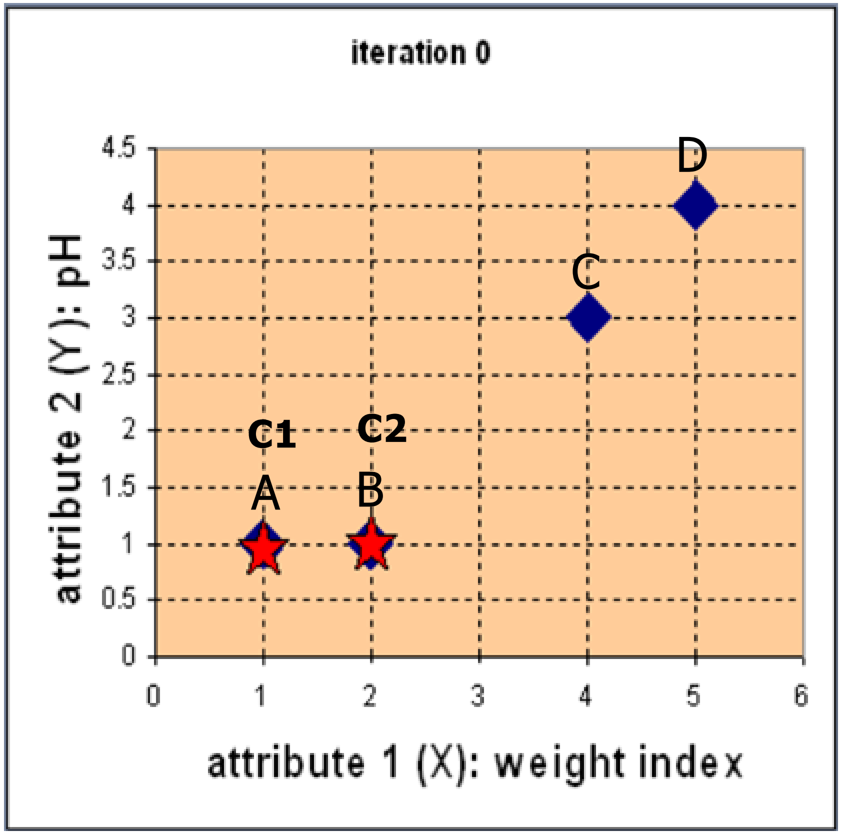
- **Problem**
- Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into *K=2* group of medicine.

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



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

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

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

Euclidean distance

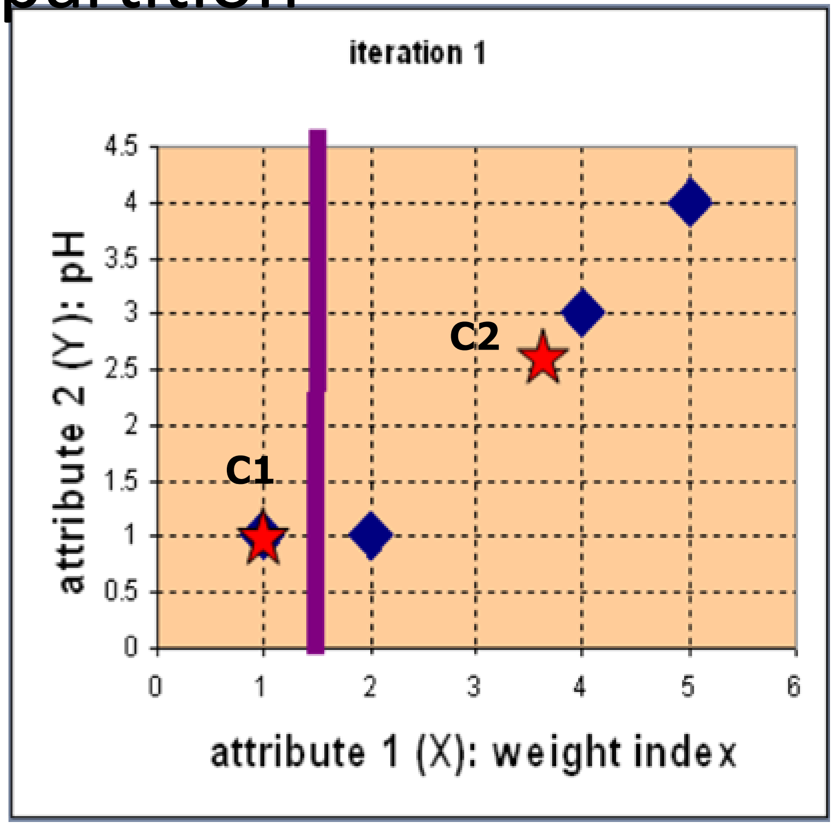
$$d(D, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

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

Assign each object to the cluster with the nearest seed point

Example

- Step 2: Compute new centroids of the current partition



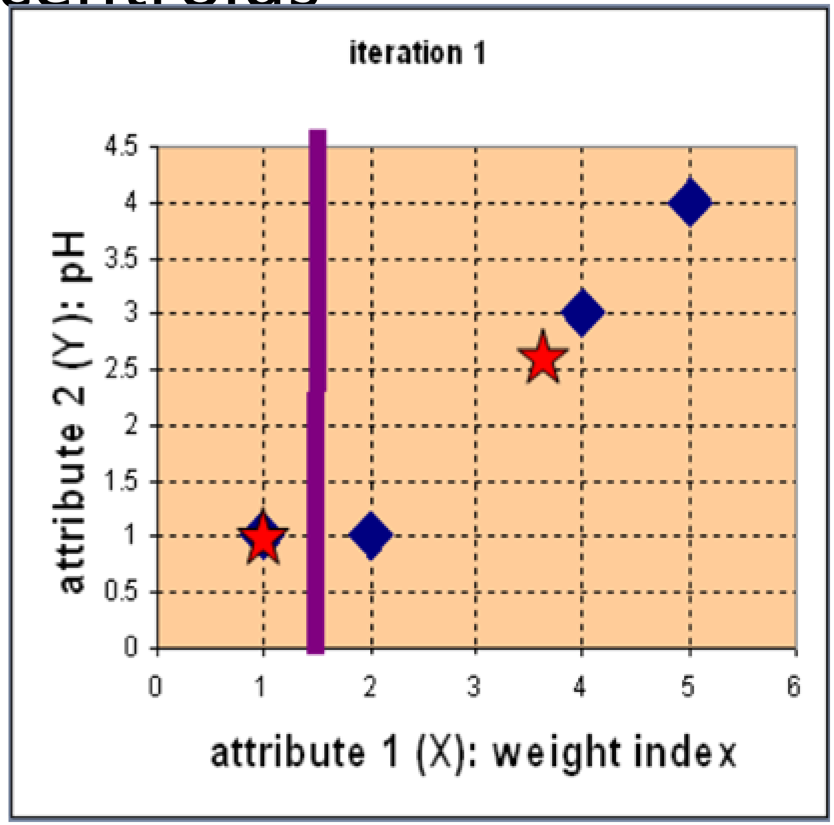
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

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

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

Example

- Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

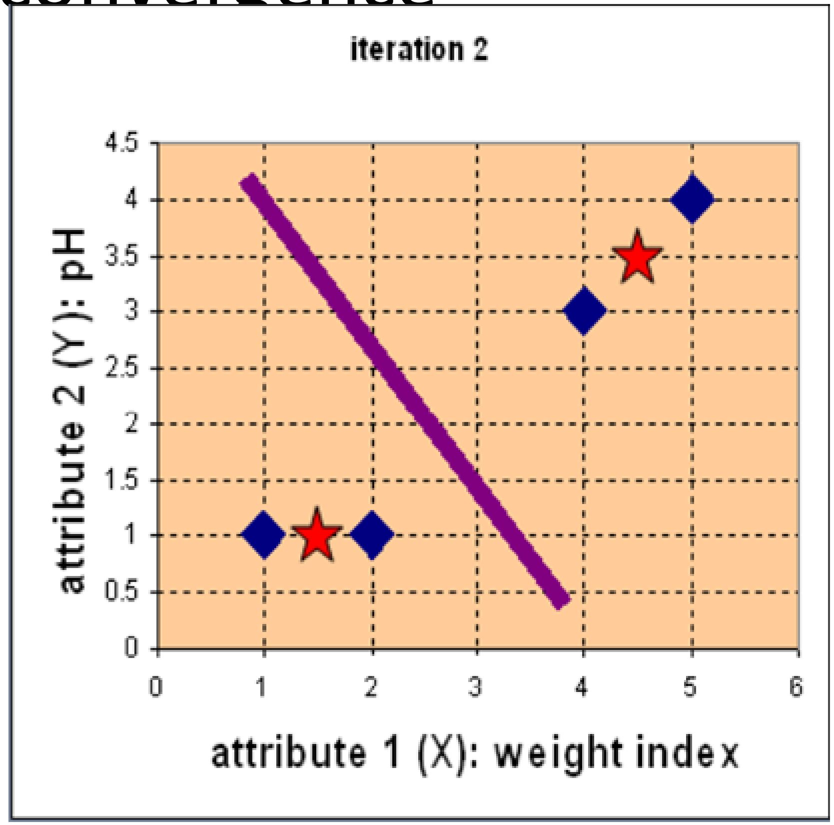
$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

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

Assign the membership to objects

Example

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



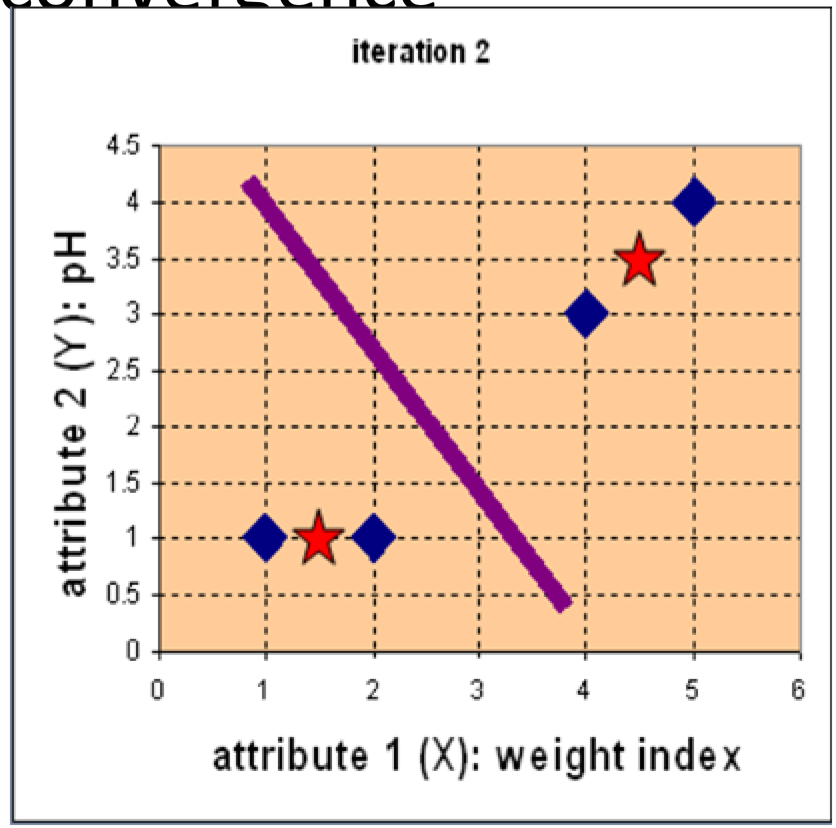
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

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

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

Example

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



Compute the distance of all objects to the new centroids

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

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

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

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

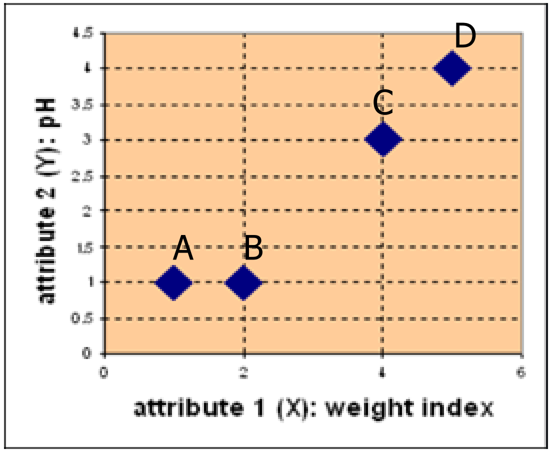
Exercise: Different Distance Metric

Use K-means with the Manhattan distance metric for clustering analysis by setting $K=2$ and initial seeds 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



Another example: (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

Another example

Step 1:

Initialization: Randomly we choose following two centroids m1[#1]=(1.0,1.0) and m2 [#4]=(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)

Another example

Step 2: From the distances:

- we obtain two clusters: {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

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

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

Another example

Step 3:

- Now using these **new** centroids we compute the Euclidean distance of each object, as in next table.
- $m_1=(1.83,2.33)$, $m_2=(4.12,5.33)$
- 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) (**WHY?**)

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

Another example

- <u>Step 4</u> :

  Now using the new centroids we compute the Euclidean distance of each object, as in next table.

The clusters obtained are:

{1,2} and {3,4,5,6,7}

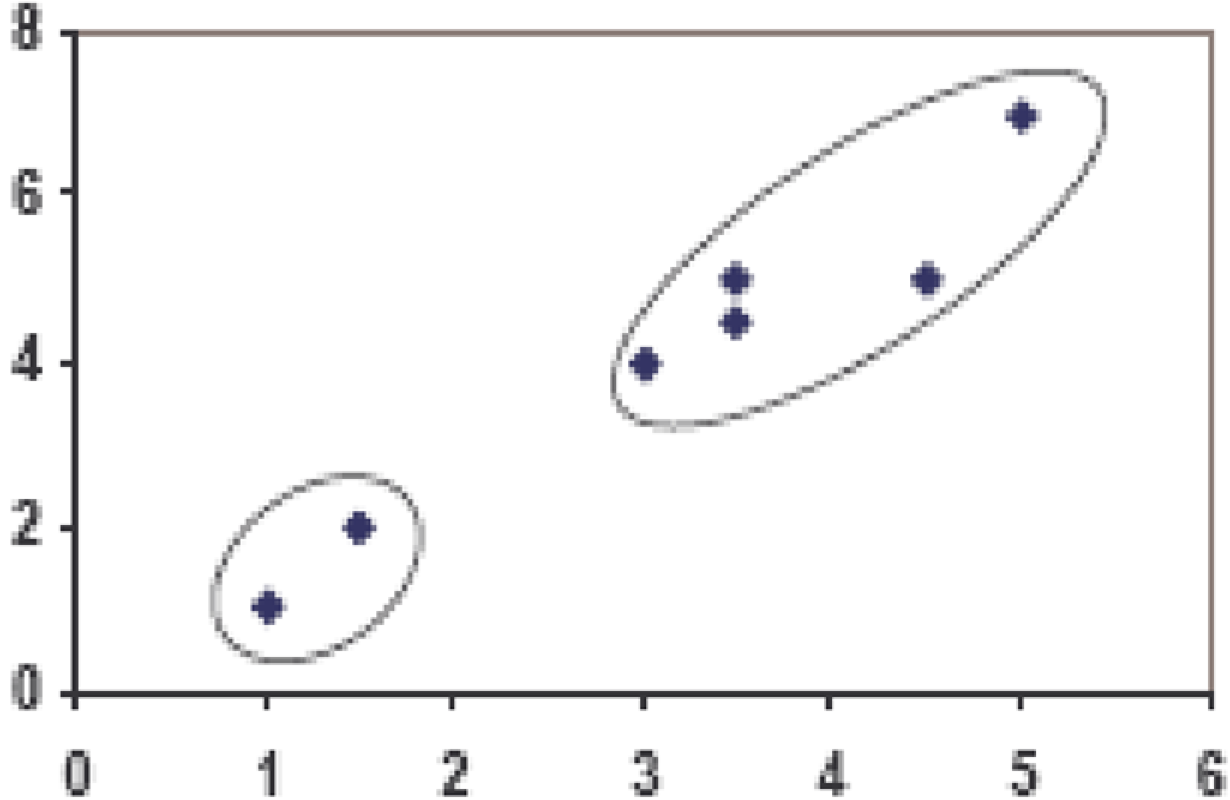
- There is no change in the cluster structure.
- Thus, the algorithm stops here and final result consist of 2 clusters

  {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

Another example

- PLOT



Another example

- (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)

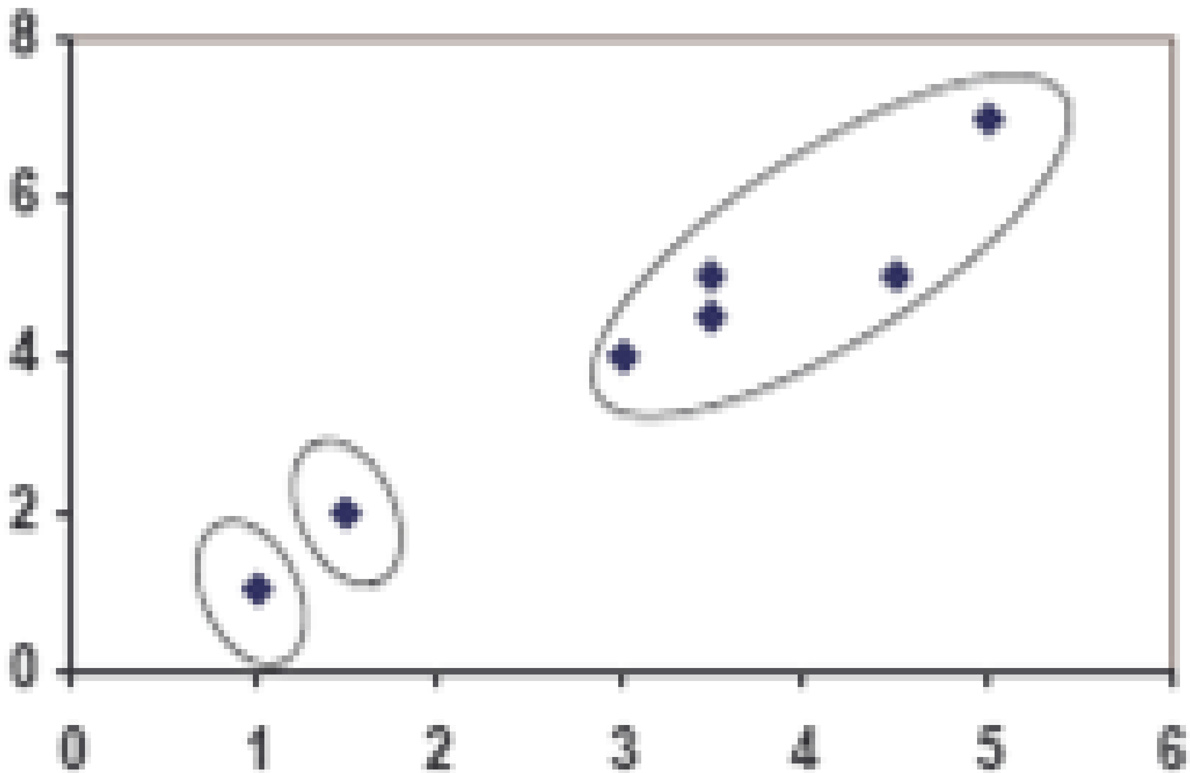
Step 1

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

Another example

- PLOT



Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

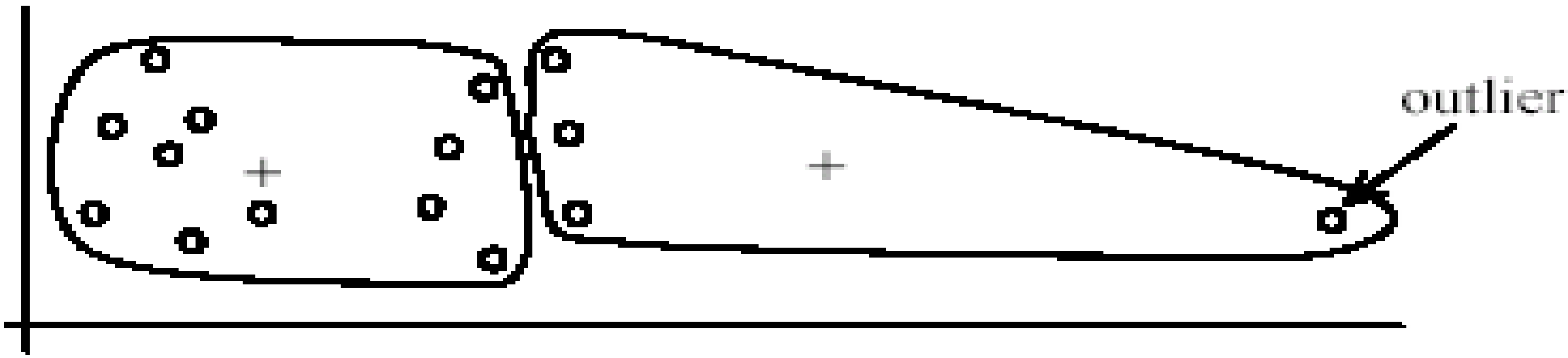
    k is the number of clusters, and

    t is the number of iterations (conversion can be slow!).
  - Since both k and t are usually small. k-means is considered a linear algorithm.
- K-means: most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE (Sum of Squared Errors) is used. The global optimum is hard to find due to complexity.

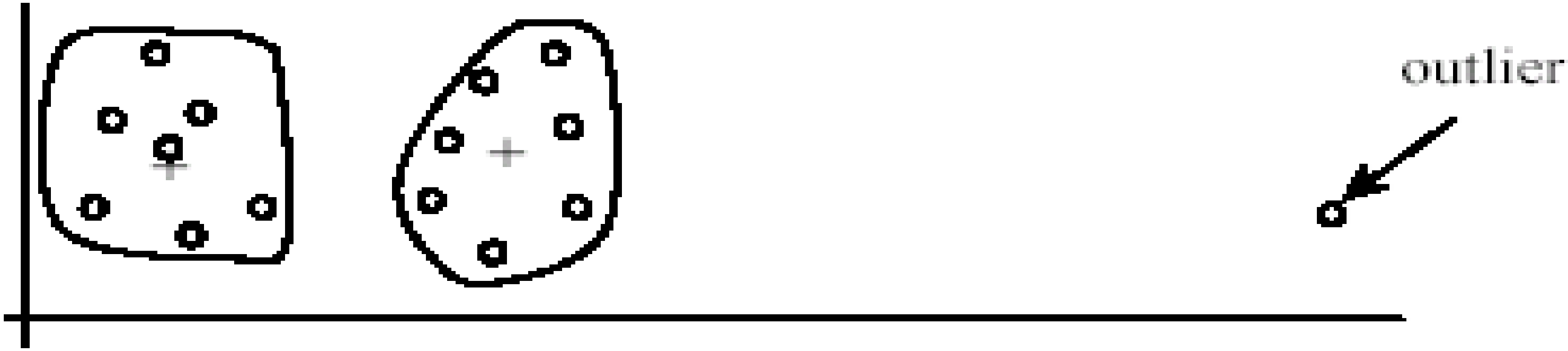
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers: data points very far away from other data points.
  - Outliers could be errors in data recording or special data points with very different values.

Weaknesses of k-means: Problems with outliers



(A): Undesirable clusters



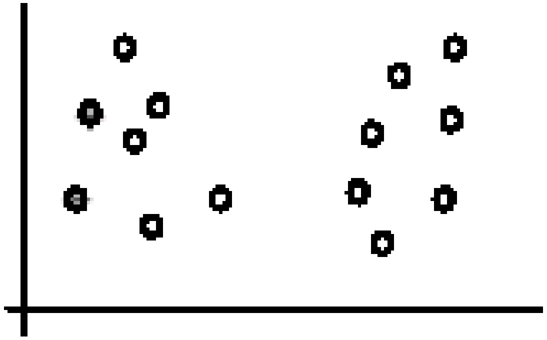
(B): Ideal clusters

Weaknesses of k-means: Outliers

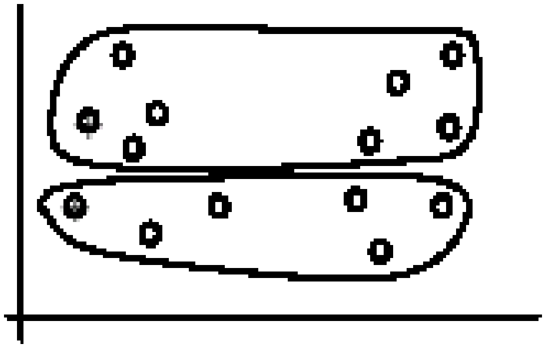
- Remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Or perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small (*larger data sets*).
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont.)

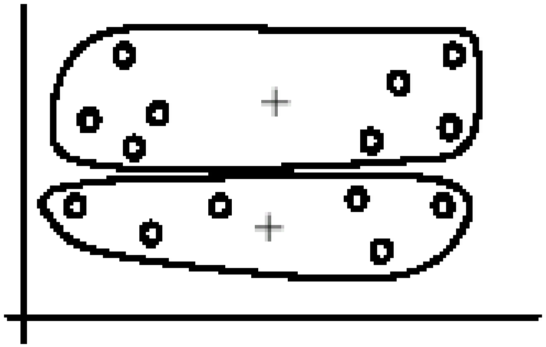
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



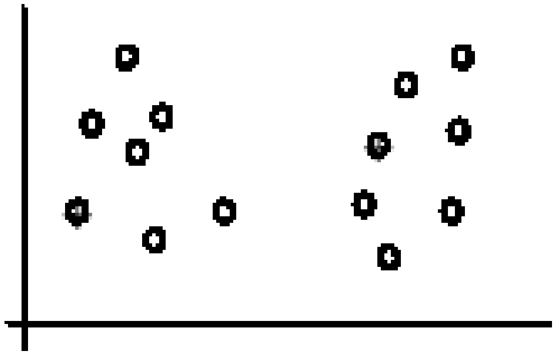
(B). Iteration 1



(C). Iteration 2

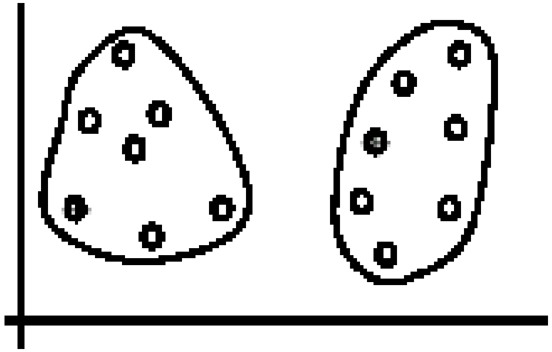
Weaknesses of k-means (cont.)

- If we use different seeds: good results

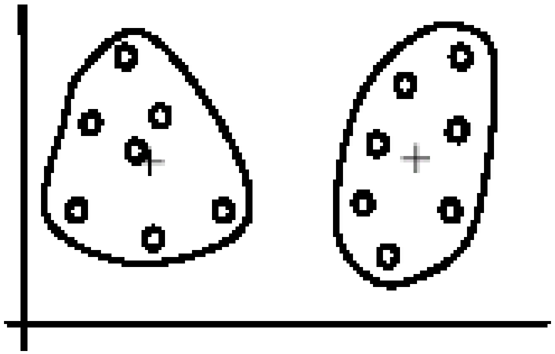


(A). Random selection of *k* seeds (centroids)

There are some methods to help choose good seeds



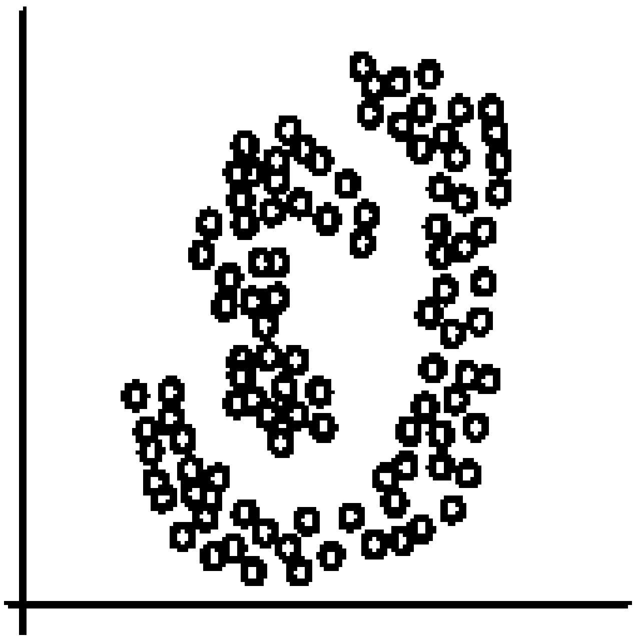
(B). Iteration 1



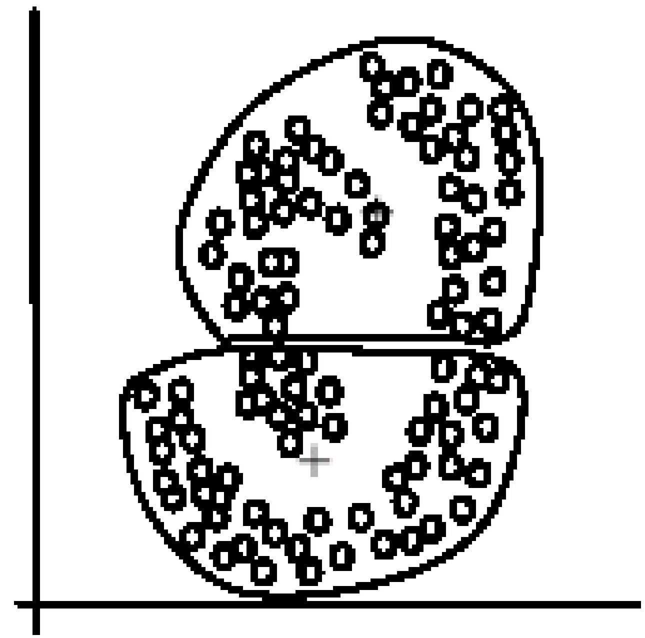
(C). Iteration 2

Weaknesses of k-means (cont.)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

Cluster Evaluation: hard problem

- The quality of clustering is very hard to evaluate because
  - We do not know the correct clusters
- Some methods, however, are used:
  - User inspection
    - Study centroids, and spreads
    - Rules from a decision tree.
    - For text documents, one can read some documents in clusters.

Cluster evaluation: ground truth

- We use some labeled data (for classification)
- Assumption: Each class is a cluster.
- After clustering, a confusion matrix is constructed. From the matrix, we compute various measurements, entropy, purity, precision, recall and F-score.
  - Let the classes in the data D be $C = (c_1, c_2, \ldots, c_k)$. The clustering method produces k clusters, which divides D into k disjoint subsets, $D_1, D_2, \ldots, D_k$.

What Is A Good Clustering?

Internal criterion: A good clustering will produce high quality clusters in which:

- the intra-class (intra-cluster) similarity is high
- the inter-class similarity is low

How would you evaluate clustering?