Unsupervised Learning

Big Data Training,

27 Februari 2019

Pusdiklat BPS

Outline

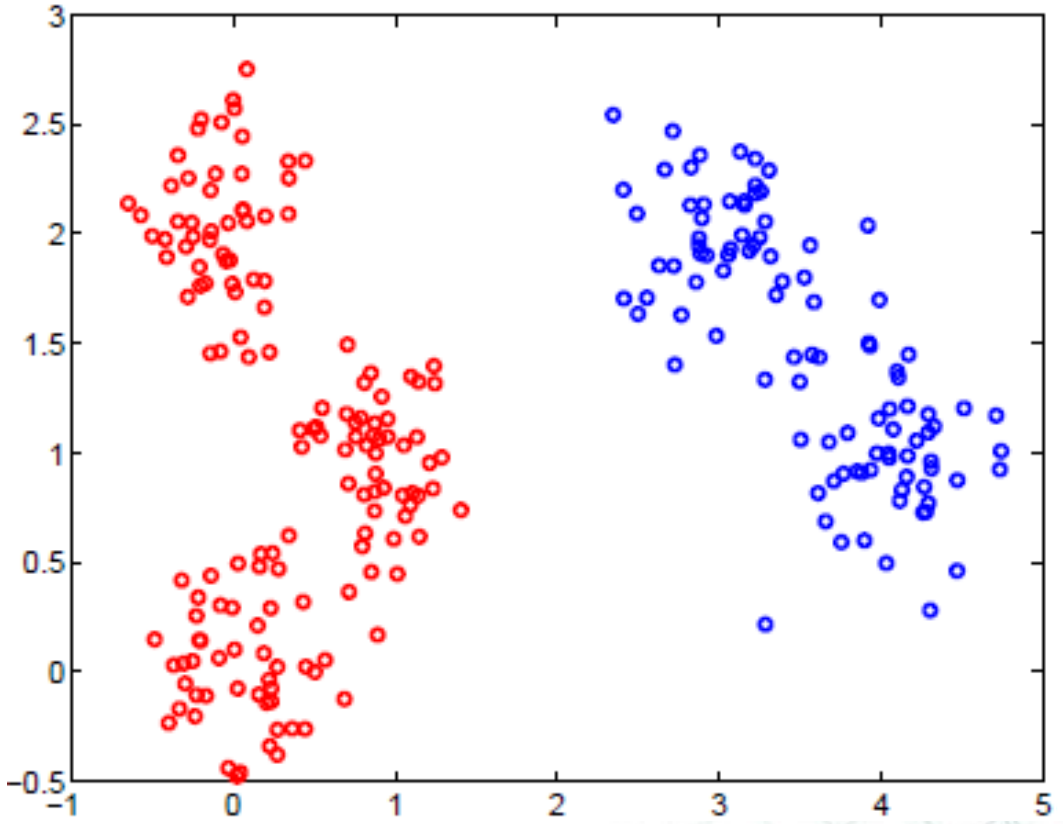
- Introduction
- Data Types and Representations
- Distance Measures
- Major Clustering Approaches
- Summary

Introduction

- Cluster: A collection/group of data objects/points
  - similar (or related) to one another within the same group
  - dissimilar (or 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
- Clustering Analysis: Unsupervised learning
  - no predefined classes for a training data set
  - Two general tasks: identify the “natural” clustering 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

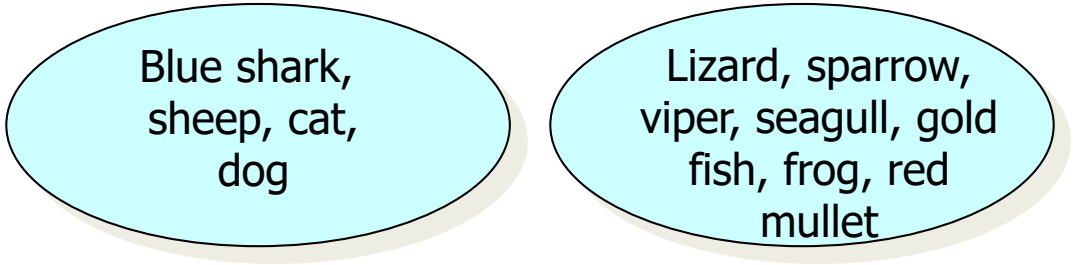
Introduction

- Illustrative Example 1: how many clusters?

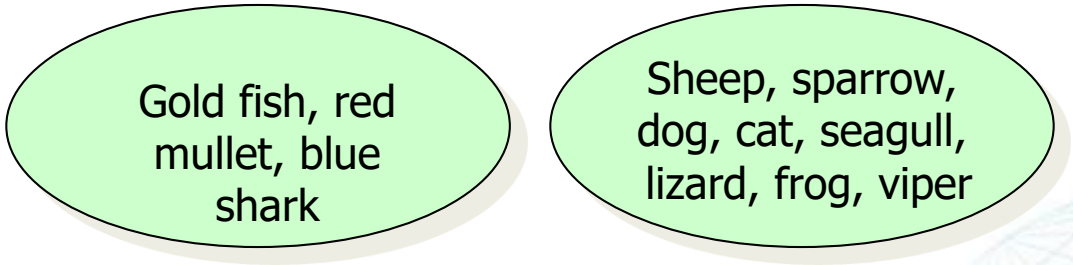


Introduction

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



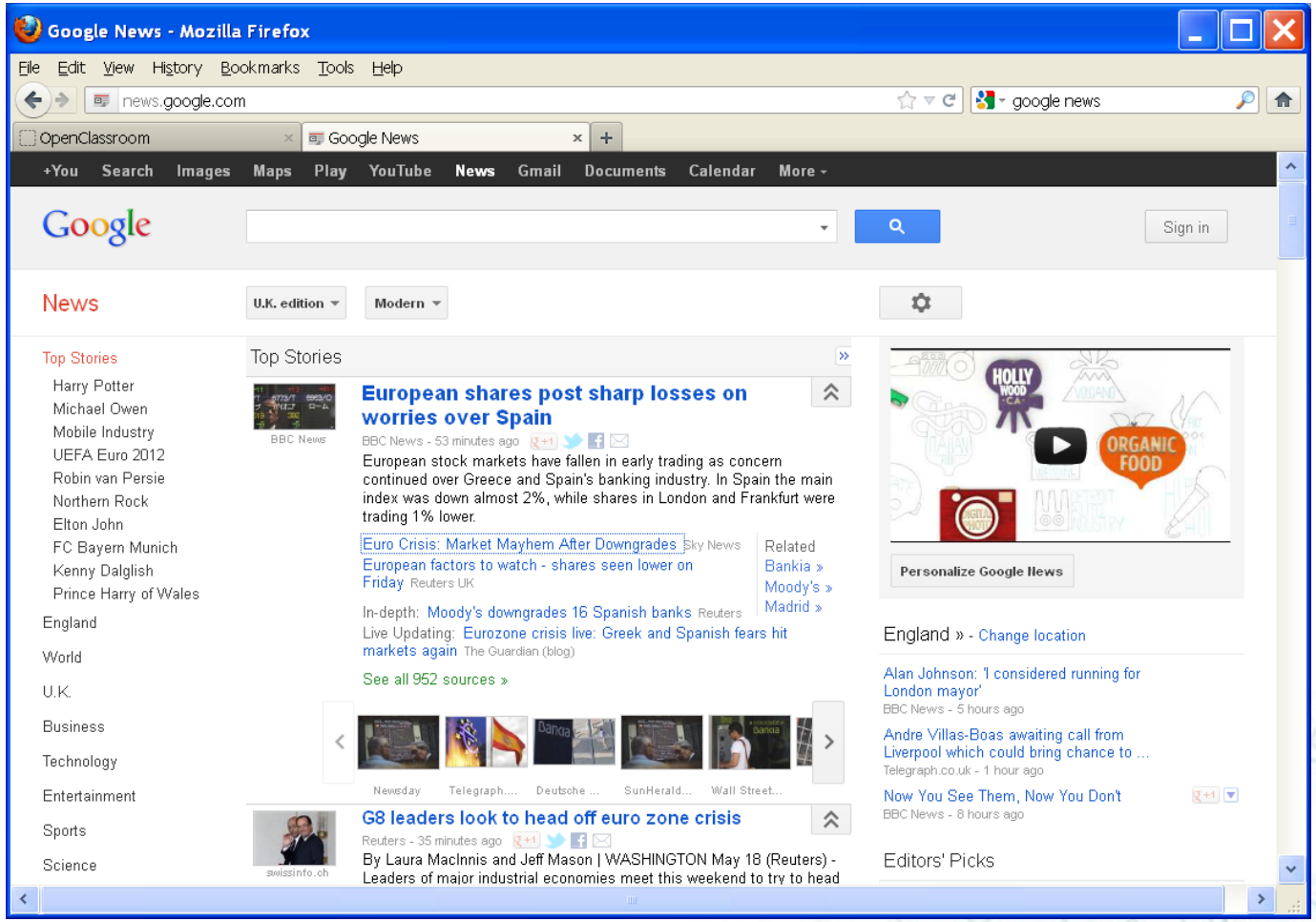
1. Two clusters
2. Clustering criterion: How animals bear their progeny



1. Two clusters
2. Clustering criterion: Existence of lungs

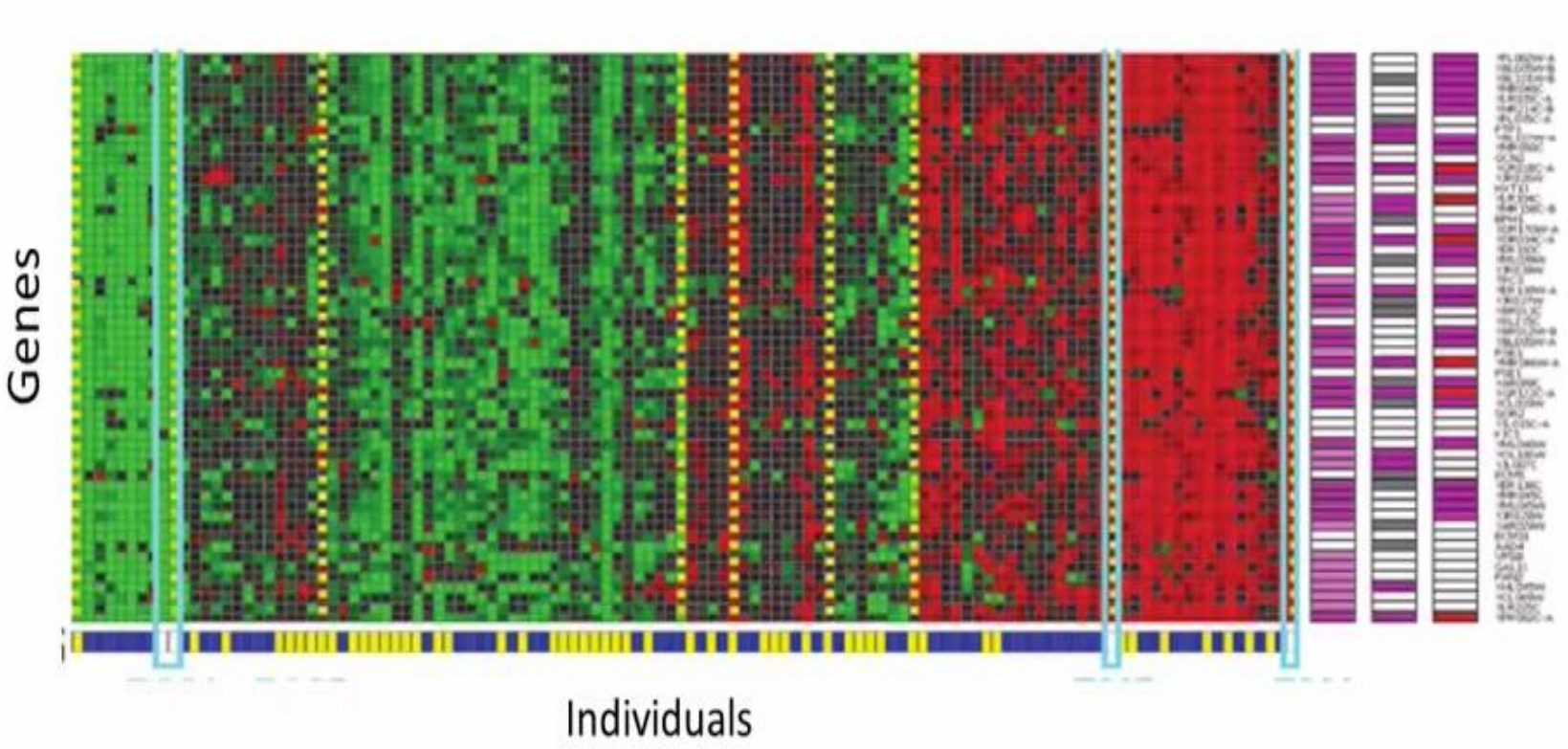
Introduction

- Real Applications: Google News



Introduction

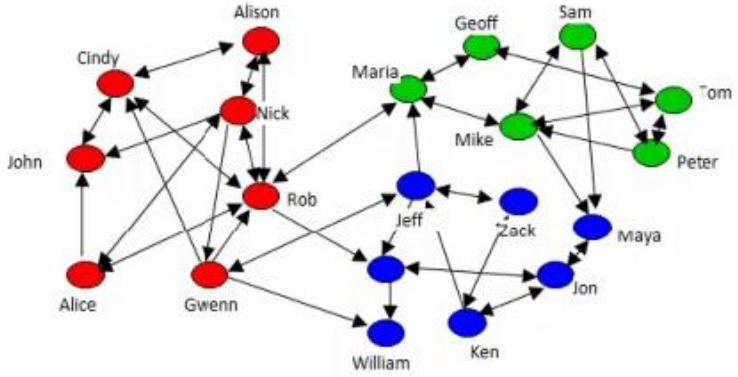
- Real Applications: Genetics Analysis



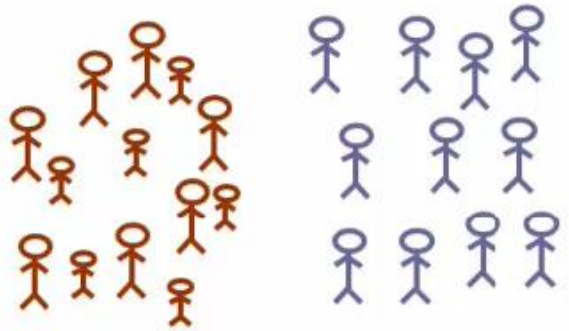
Introduction

- Real Applications: Emerging Applications

Organize computing clusters



Social network analysis



Market segmentation.

Astronomical data analysis

Introduction

- A technique demanded by many 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

Clustering as a Preprocessing Tool (Utility)

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - It is hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective

Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Quiz

Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

- ☐ Given email labeled as spam/not spam, learn a spam filter.
- ☑ Given a set of news articles found on the web, group them into set of articles about the same story.
- ☑ Given a database of customer data, automatically discover market segments and group customers into different market segments.
- ☐ Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

Data Types and Representations

- Discrete vs. Continuous
  - Discrete Feature
    - Has only a finite set of values

      e.g., zip codes, rank, or the set of words in a collection of documents
    - Sometimes, represented as integer variable
  - Continuous Feature
    - Has real numbers as feature values

      e.g, temperature, height, or weight
    - Practically, real values can only be measured and represented using a finite number of digits
    - Continuous features are 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

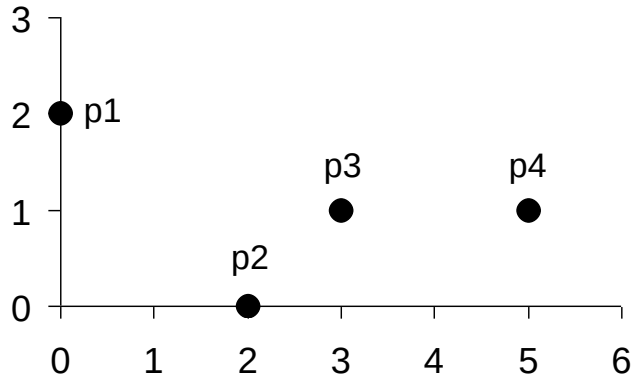
  - Distance/dissimilarity matrix (object-by-object structure)

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

    - *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 $\mathbf{y} = (y_1\ y_2 \cdots y_n)$

  $$d(\mathbf{x}, \mathbf{y}) = \sqrt[p]{|\ x_1 - y_1\ |^p + |\ x_2 - y_2\ |^p \cdots + |\ x_n - y_n\ |^p}, \quad p > 0$$

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

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

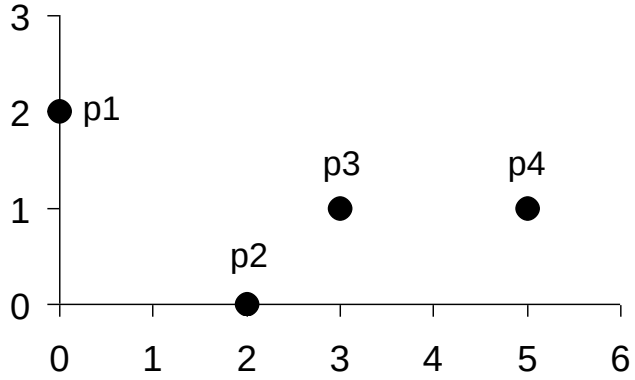
  - $p = 2$: Euclidean distance

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

  - 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: Manhattan and Euclidean distances



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

Data Matrix

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

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 $\mathbf{y} = (y_1\ y_2 \cdots y_n)$

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

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

  - Property: $0 \le d(\mathbf{x}, \mathbf{y}) \le 2$
  - Nonmetric vector objects: keywords in documents, gene features in micro-arrays, …
  - Applications: information retrieval, biologic taxonomy, ...

Distance Measures

- Example: Cosine measure

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

$$\mathbf{x}_1 \bullet \mathbf{x}_2 = 3 \times 1 + 2 \times 0 + 0 \times 0 + 5 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 2 = 5$$

$$\|\mathbf{x}_1\| = \sqrt{3^2 + 2^2 + 0^2 + 5^2 + 2^2 + 0^2 + 0^2} = \sqrt{42} \approx 6.48$$

$$\|\mathbf{x}_2\| = \sqrt{1^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2} = \sqrt{6} \approx 2.45$$

$$\cos(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \bullet \mathbf{x}_2}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|} = \frac{5}{6.48 \times 2.45} \approx 0.32$$

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

Distance Measures

- Distance for Binary Features
  - For binary features, their value can be converted into 1 or 0.
  - Contingency table for binary feature vectors, $\mathbf{x}$ and $\mathbf{y}$

		$\mathbf{y}$ = 1	$\mathbf{y}$ = 0
$\mathbf{x}$	1	a	b
	0	c	d

a : number of features that equal 1 for both $\mathbf{x}$ and $\mathbf{y}$

b : number of features that equal 1 for $\mathbf{x}$ but that are 0 for $\mathbf{y}$

c : number of features that equal 0 for $\mathbf{x}$ but that are 1 for $\mathbf{y}$

d : number of features that equal 0 for both $\mathbf{x}$ and $\mathbf{y}$

Distance Measures

- Distance for Binary Features
  - Distance for symmetric binary features

    Both of their states equally valuable and carry the same weight; i.e., no preference on which outcome should be coded as 1 or 0 , e.g. gender

$$d(\mathbf{x},\mathbf{y}) = \frac{b+c}{a+b+c+d}$$

  - Distance for asymmetric binary features

    Outcomes of the states not equally important, e.g., the *positive* and *negative* outcomes of a disease

$$d(\mathbf{x},\mathbf{y}) = \frac{b+c}{a+b+c}$$

et to 1 and the other is 0.

Distance Measures

- Example: Distance for binary features

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	1	0	1	0	0	0
Mary	F	1	0	1	0	1	0
Jim	M	1	1	0	0	0	0

- "Y": yes
- "P": positive
- "N": negative

- gender is a symmetric feature (less important)
- the remaining features are asymmetric binary
- set the values "Y" and "P" to 1, and the value "N" to 0

Mary

Jack	2	0
	1	3

Jim

Jack	1	1
	1	3

Mary

Jim	1	1
	2	2

$$d(\text{Jack},\text{Mary}) = \frac{0+1}{2+0+1} = 0.33$$

$$d(\text{Jack},\text{Jim}) = \frac{1+1}{1+1+1} = 0.67$$

$$d(\text{Jim},\text{Mary}) = \frac{1+2}{1+1+2} = 0.75$$

Distance Measures

- Distance for nominal features
  - A generalization of the binary feature so that it can take more than two states/values, e.g., red, yellow, blue, green, ……
  - There are two methods to handle variables of such features.
    - **Simple mis-matching**

$$d(\mathbf{x}, \mathbf{y}) = \frac{\text{number of mis-matching features between } \mathbf{x} \text{ and } \mathbf{y}}{\text{total number of features}}$$

    - **Convert it into binary variables**

creating new binary features for all of its nominal states

e.g., if an feature has three possible nominal states: red, yellow and blue, then this feature will be expanded into three binary features accordingly.

Thus, distance measures for binary features are now applicable!

Distance Measures

- Distance for nominal features (cont.)
  - Example: Play tennis

	Outlook	Temperature	Humidity	Wind
D_1	010	100	10	10
D_2	100	100	01	10

  - **Simple mis-matching**

$$d(D_1, D_2) = \frac{2}{4} = 0.5$$

  - **Creating new binary features**
    - **Using the same number of bits as those features can take**

Outlook = {Sunny, Overcast, Rain} ⟶ (100, 010, 001)

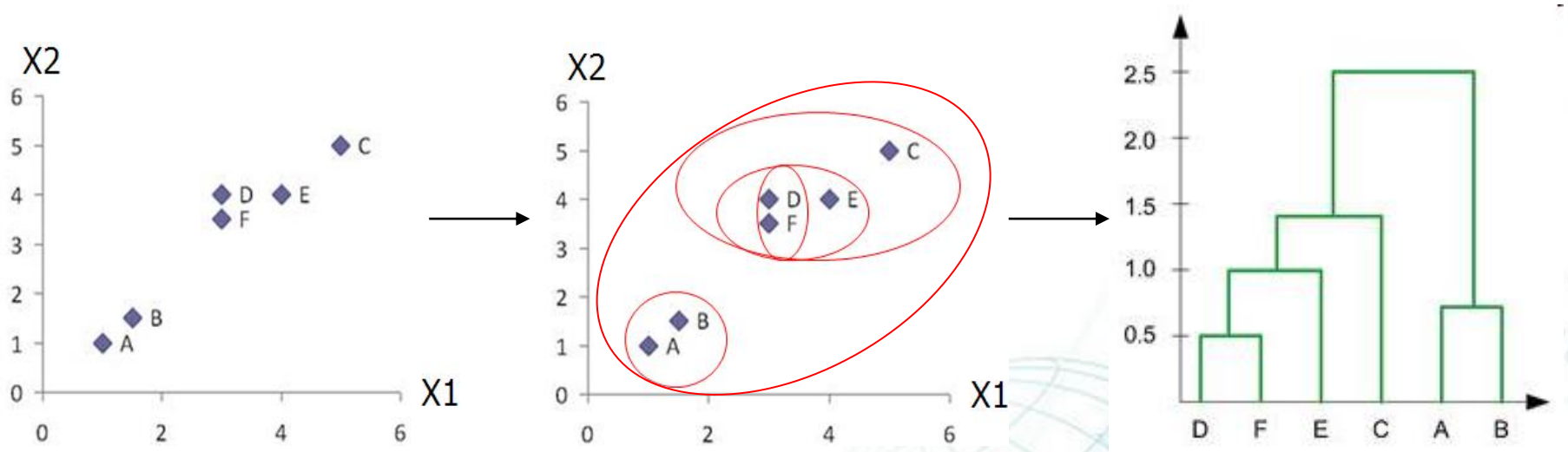
Temperature = {High, Mild, Cool} ⟶ (100, 010, 001)

Humidity = {High, Normal} ⟶ (10, 01)

Wind = {Strong, Weak} ⟶ (10, 01)

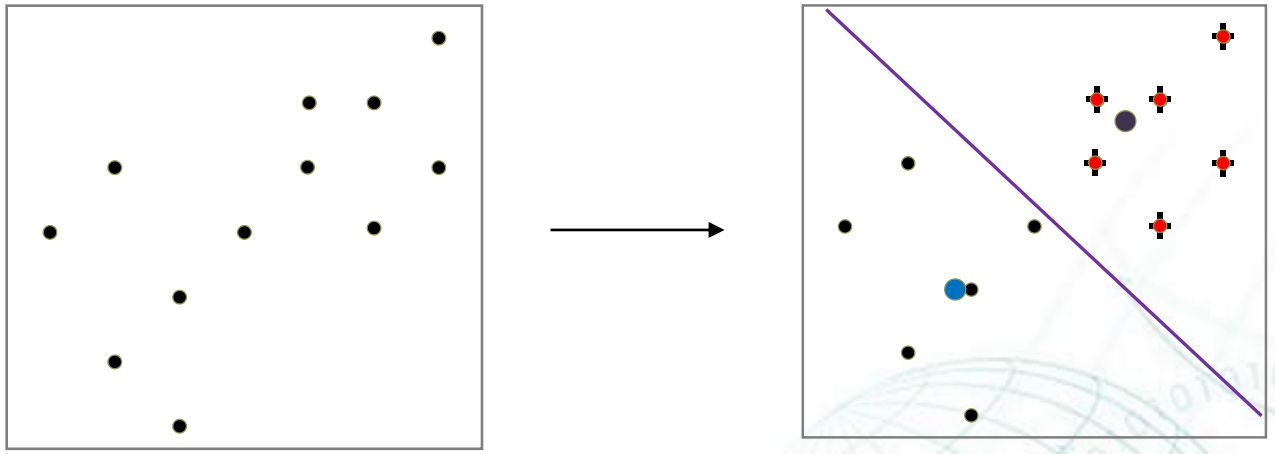
Major Clustering Approaches

- Hierarchical approach
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Agglomerative, Diana, Agnes, BIRCH, ROCK, ……



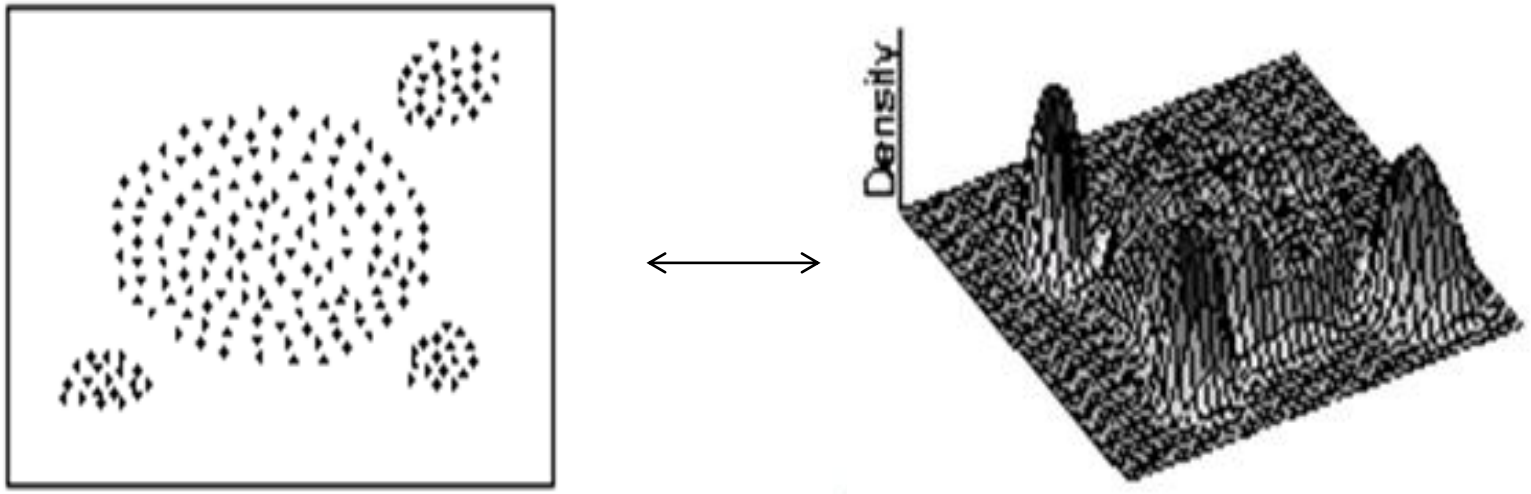
Major Clustering Approaches

- Partitioning approach
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square distance cost
  - Typical methods: k-means, k-medoids, CLARANS, ……



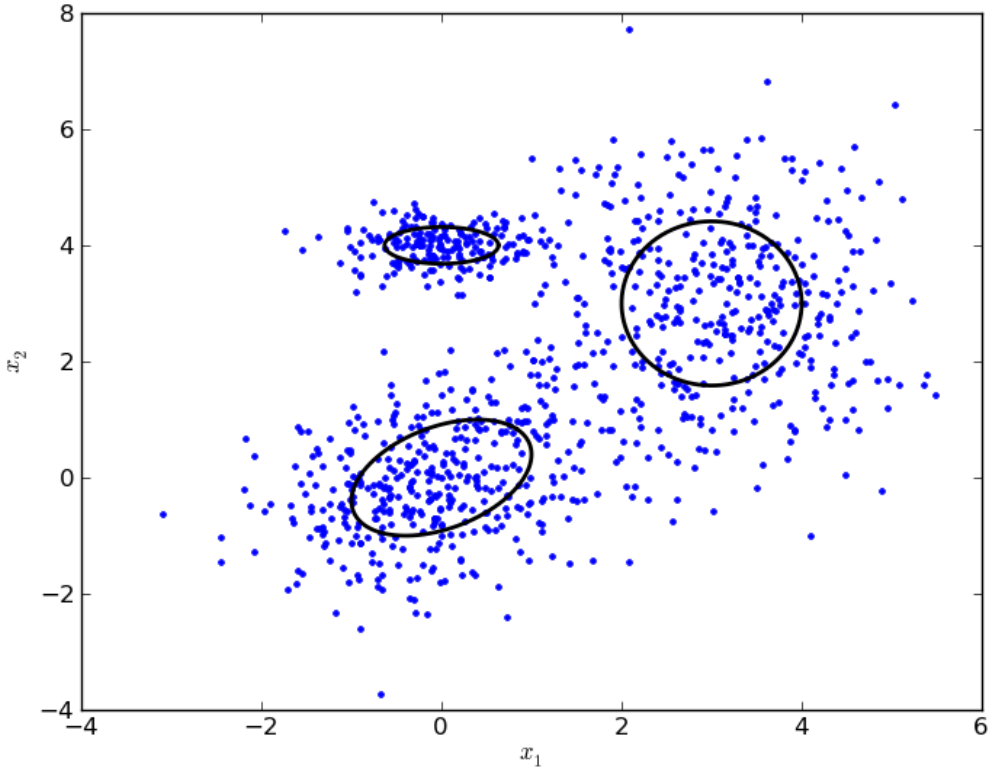
Major Clustering Approaches

- Density-based approach
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue, ……



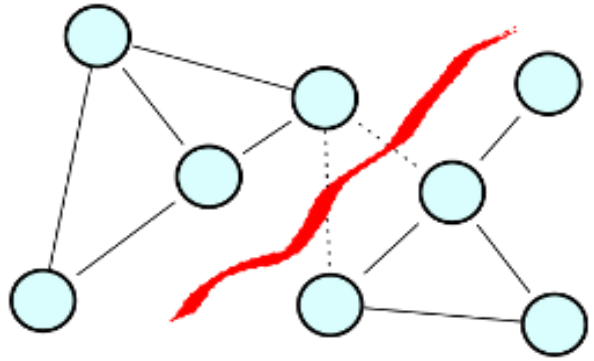
Major Clustering Approaches

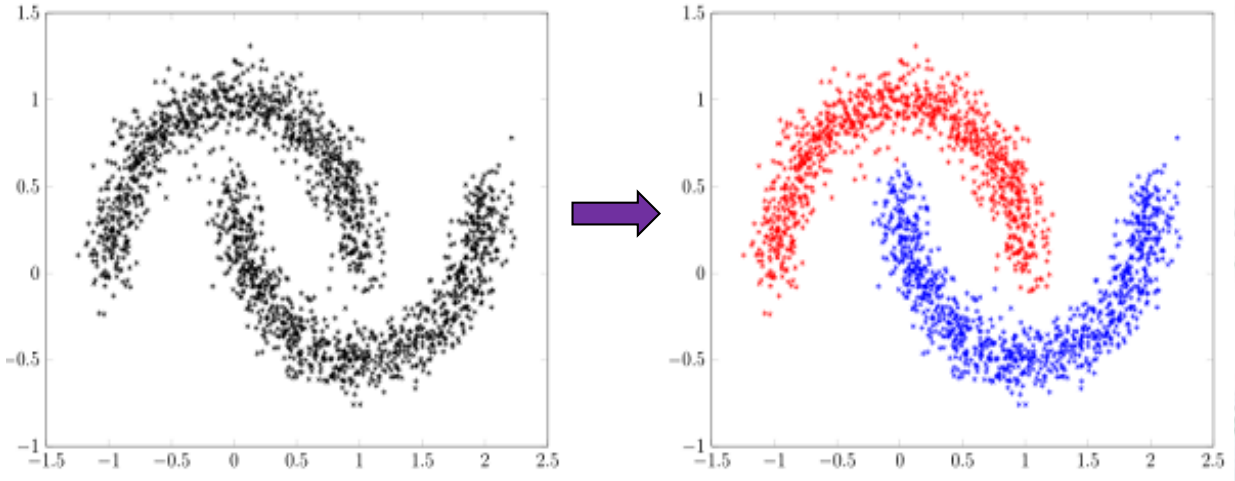
- Model-based approach
  - A generative model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: Gaussian Mixture Model (GMM), COBWEB, ……



Major Clustering Approaches

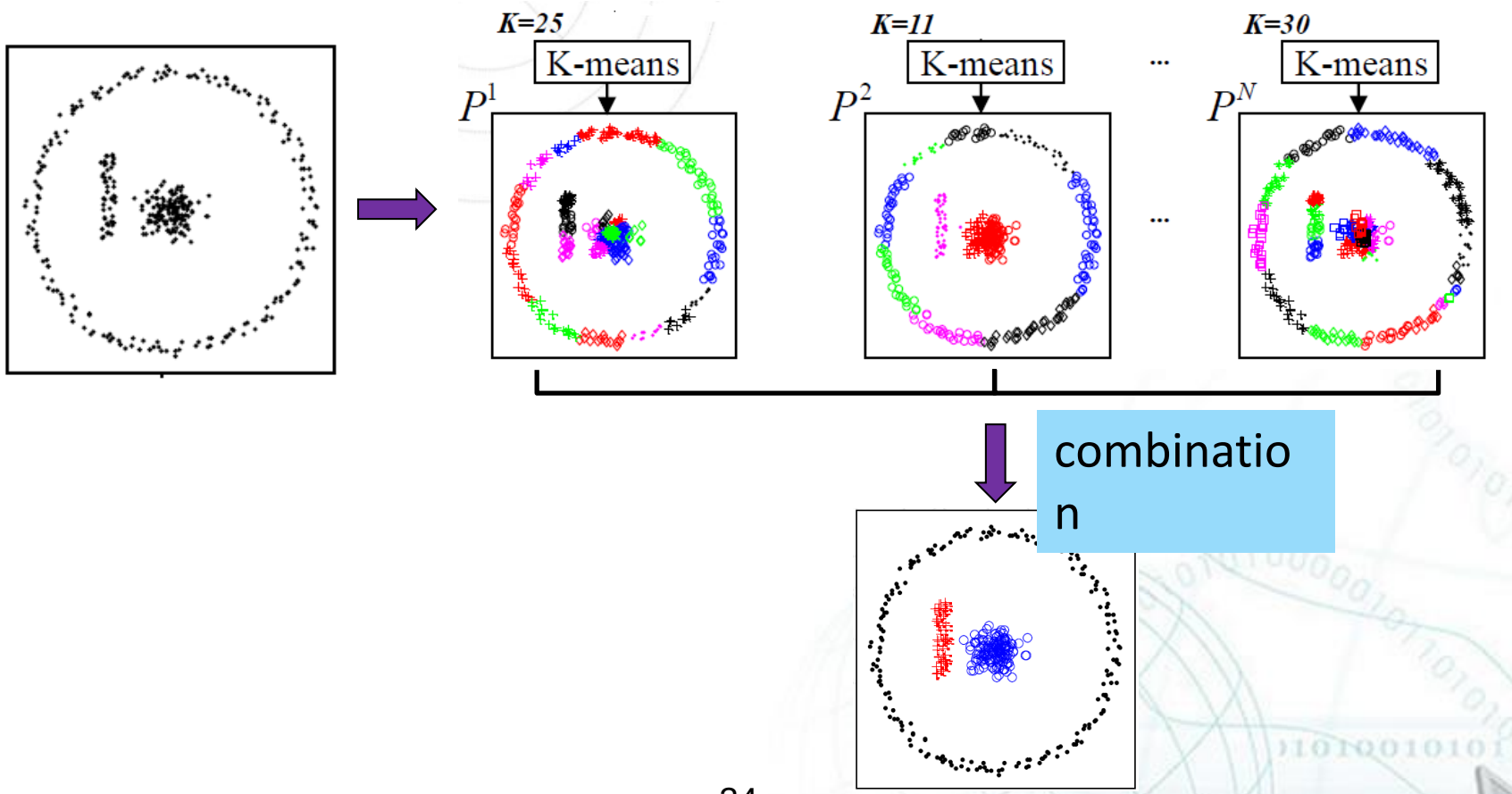
- Spectral clustering approach
  - Convert data set into weighted graph (vertex, edge), then cut the graph into sub-graphs corresponding to clusters via spectral analysis
  - Typical methods: Normalised-Cuts ……





Major Clustering Approaches

- Clustering ensemble approach
  - Combine multiple clustering results (different partitions)
  - Typical methods: Evidence-accumulation based, graph-based ……



Summary

- Clustering analysis groups objects based on their (dis)similarity and has a broad range of applications.
- Measure of distance (or similarity) plays a critical role in clustering analysis and distance-based learning.
- Clustering algorithms can be categorized into partitioning, hierarchical, density-based, model-based, spectral clustering as well as ensemble approaches.
- There are still lots of research issues on cluster analysis;
  - finding the number of “natural” clusters with arbitrary shapes
  - dealing with mixed types of features
  - handling massive amount of data – Big Data
  - coping with data of high dimensionality
  - performance evaluation (especially when no ground-truth available)

Hierarchical Clustering

Outline

- Introduction
- Cluster Distance Measures
- Agglomerative Algorithm
- Example and Demo
- Relevant Issues
- Summary

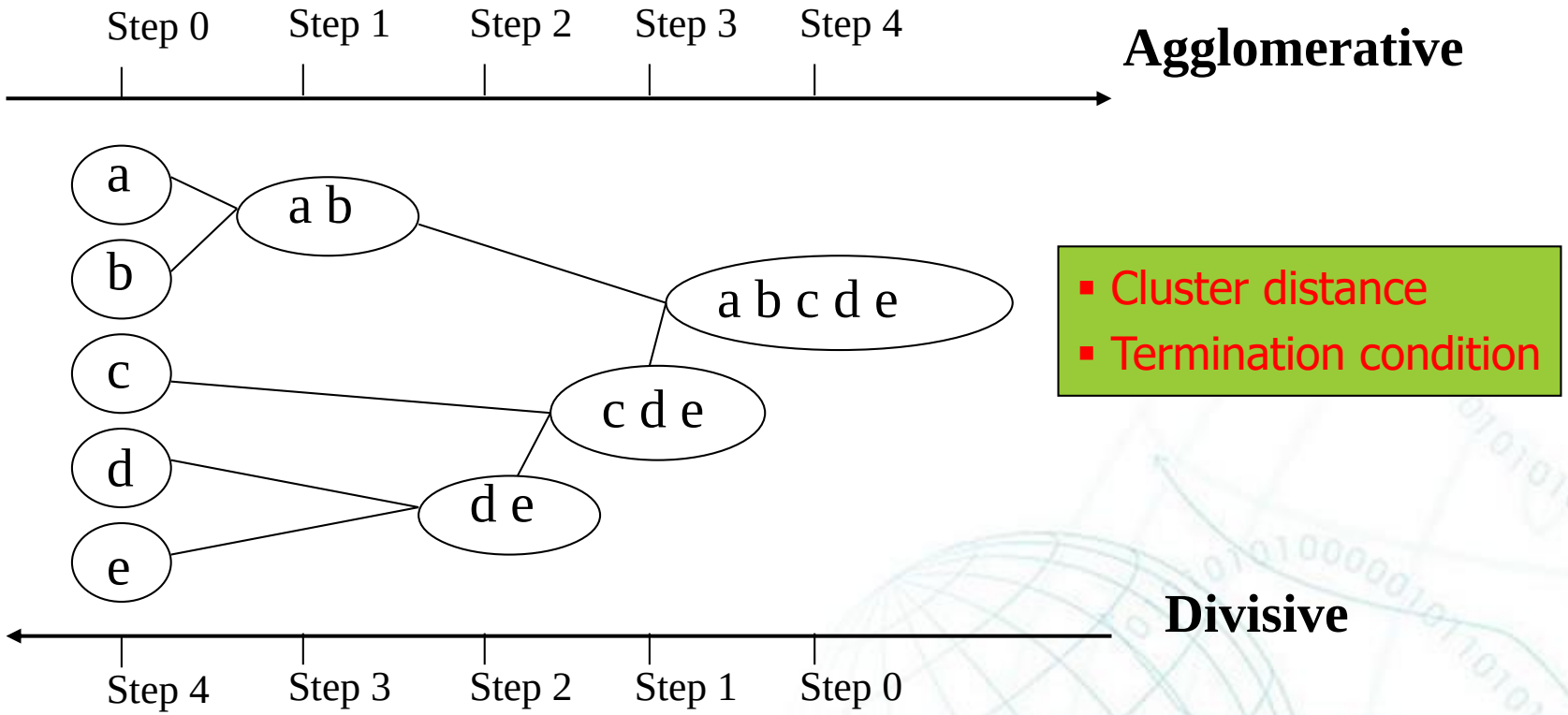
Introduction

- Hierarchical Clustering Approach
  - A typical clustering analysis approach via partitioning data set sequentially
  - Construct nested partitions layer by layer via grouping objects into a tree of clusters (without the need to know the number of clusters in advance)
  - Use (generalised) distance matrix as clustering criteria
- Agglomerative vs. Divisive
  - Two sequential clustering strategies for constructing a tree of clusters
  - Agglomerative: a bottom-up strategy
    - Initially each data object is in its own (atomic) cluster
    - Then merge these atomic clusters into larger and larger clusters
  - Divisive: a top-down strategy
    - Initially all objects are in one single cluster
    - Then the cluster is subdivided into smaller and smaller clusters

Introduction

- Illustrative Example

  Agglomerative and divisive clustering on the data set {a, b, c, d ,e }

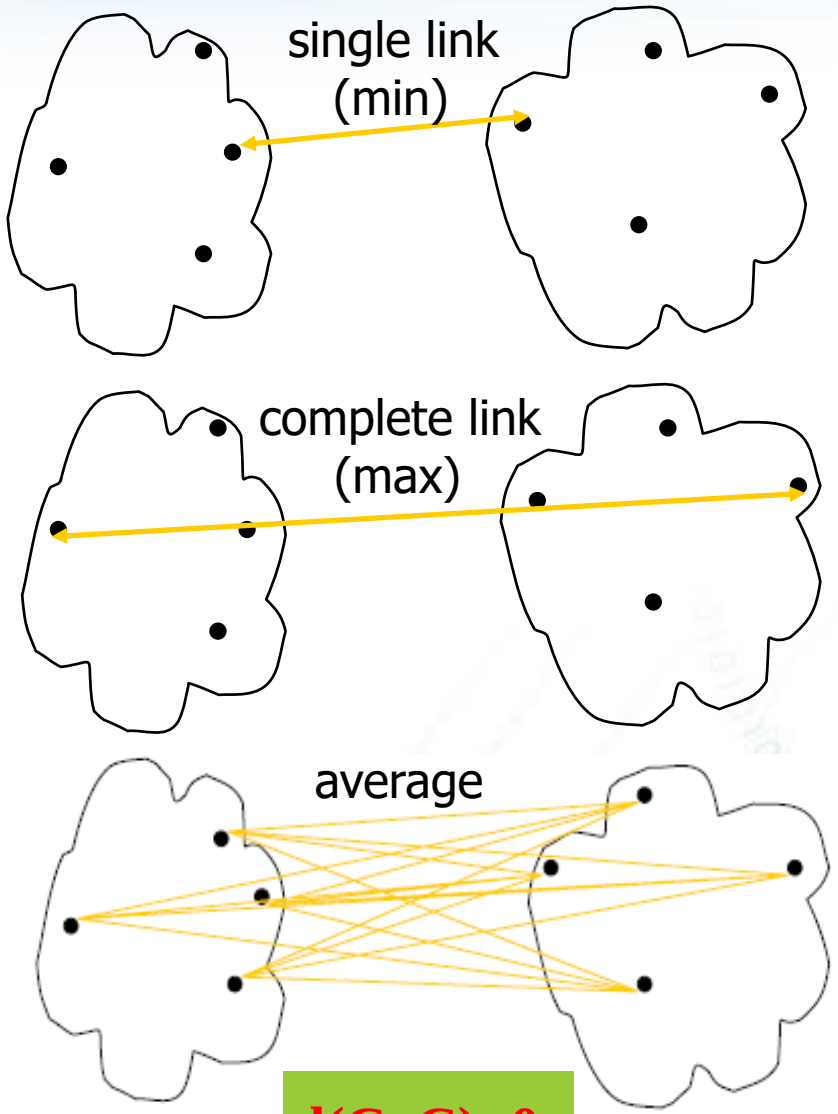


- Cluster distance
- Termination condition

Cluster Distance Measures

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$
- Average: avg distance between elements in one cluster and elements in the other, i.e.,

$d(C_i, C_j) = \text{avg}\{d(x_{ip}, x_{jq})\}$



Cluster Distance Measures

Example: Given a data set of five objects characterised by a single feature, assume that there are two clusters: C_1: {a, b} and C_2: {c, d, e}.

	a	b	c	d	e
Feature	1	2	4	5	6

1. Calculate the distance matrix.

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

2. Calculate three cluster distances between C_1 and C_2.

Single link

$$\begin{aligned}\text{dist}(C_1, C_2) &= \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \min\{3, 4, 5, 2, 3, 4\} = 2\end{aligned}$$

Complete link

$$\begin{aligned}\text{dist}(C_1, C_2) &= \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \max\{3, 4, 5, 2, 3, 4\} = 5\end{aligned}$$

Average

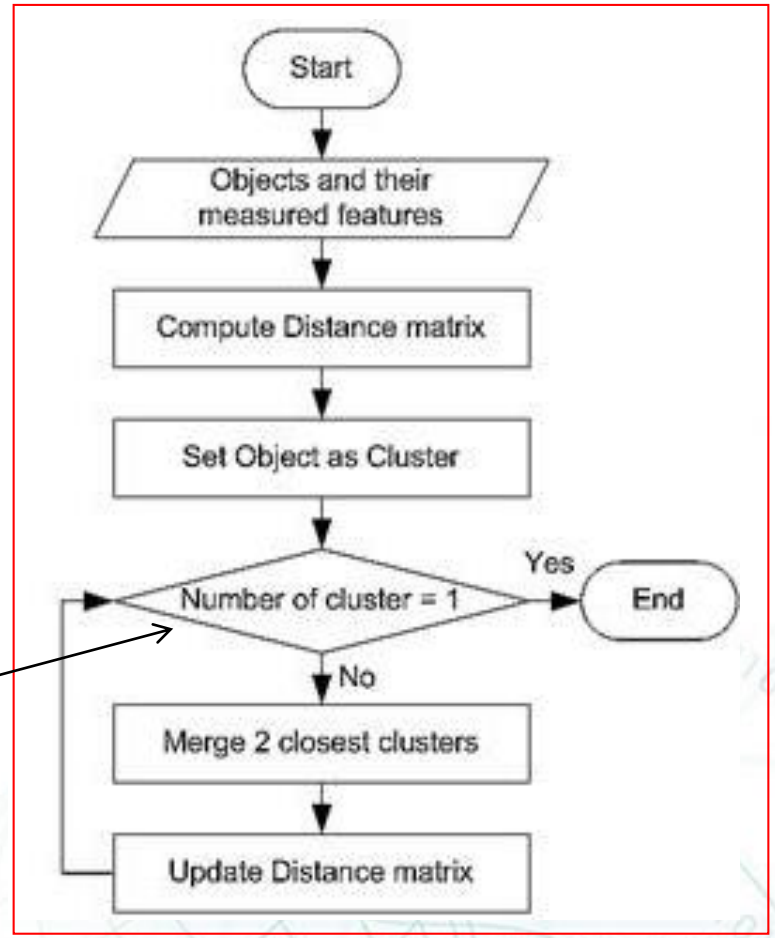
$$\text{dist}(C_1, C_2) = \frac{d(a,c) + d(a,d) + d(a,e) + d(b,c) + d(b,d) + d(b,e)}{6}$$

$$= \frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5$$

Agglomerative Algorithm

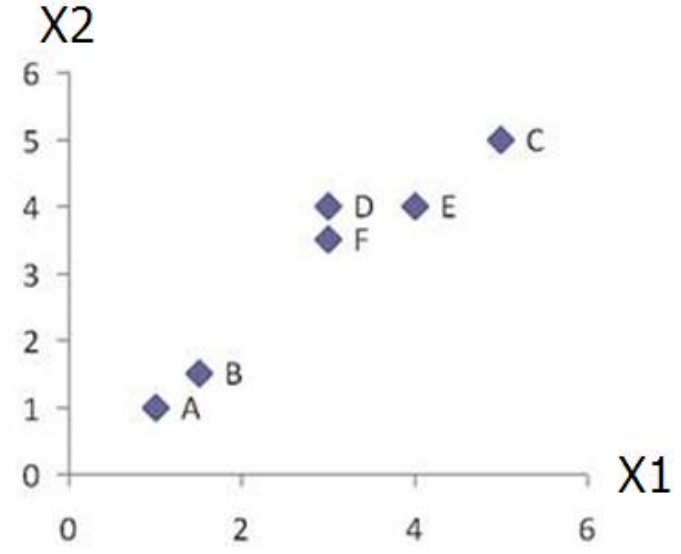
- The *Agglomerative* algorithm is carried out in three steps:

1) Convert all object features into a distance matrix
2) Set each object as a cluster (thus if we have *N* objects, we will have *N* clusters at the beginning)
3) Repeat until number of cluster is one (or known # of clusters)
   - Merge two closest clusters
   - Update "distance matrix"



Example

- Problem: clustering analysis with agglomerative algorithm



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

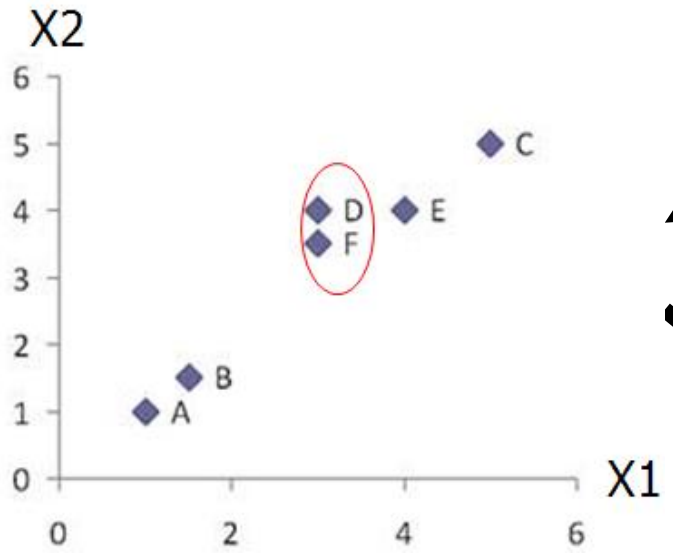
Euclidean distance

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

Example

- Merge two closest clusters (iteration 1)



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Example

- Update distance matrix (iteration 1)

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\to A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\to B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\to C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

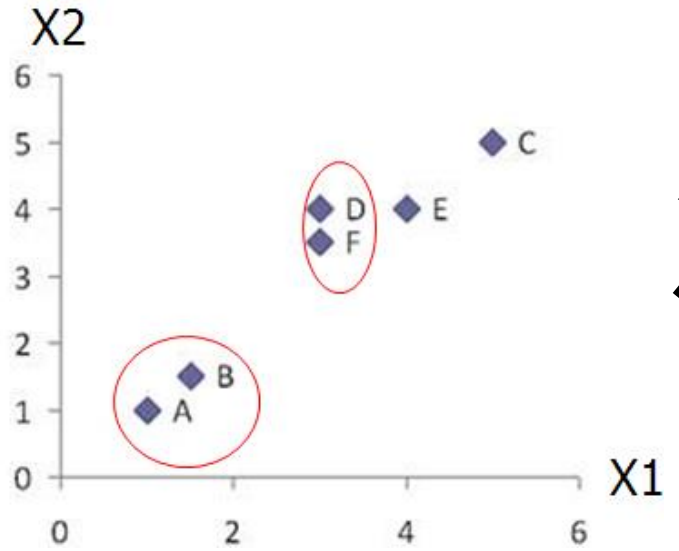
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Example

- Merge two closest clusters (iteration 2)



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Example

- Update distance matrix (iteration 2)

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

$$d_{(D,F)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

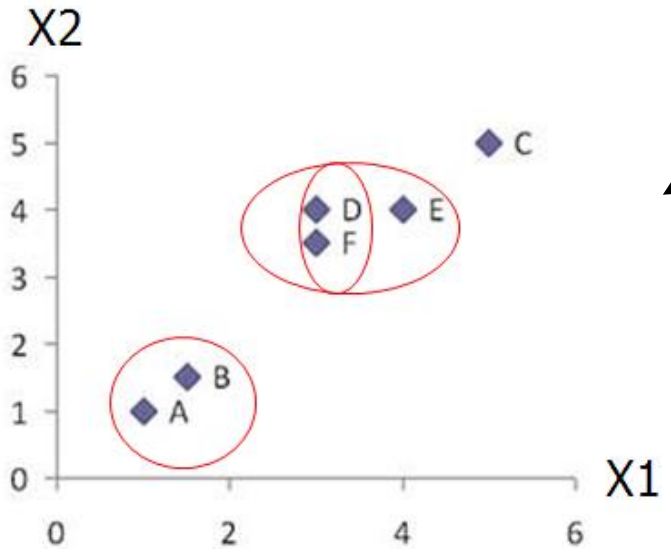
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Example

- Merge two closest clusters/update distance matrix (iteration 3)



Min Distance (Single Linkage)

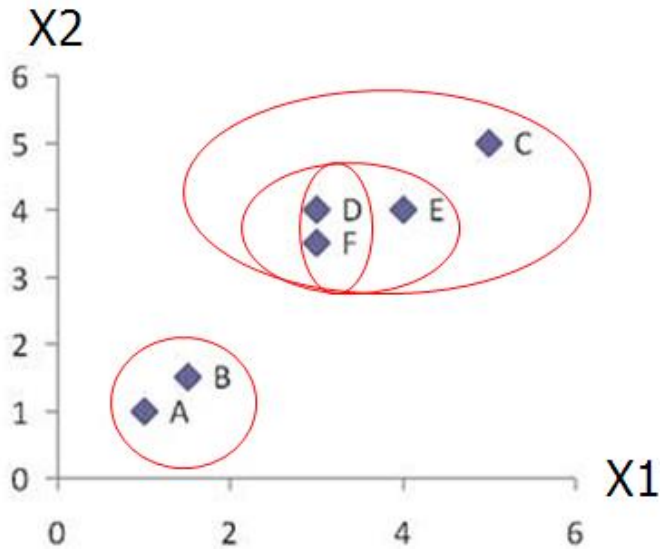
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Example

- Merge two closest clusters/update distance matrix (iteration 4)



Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

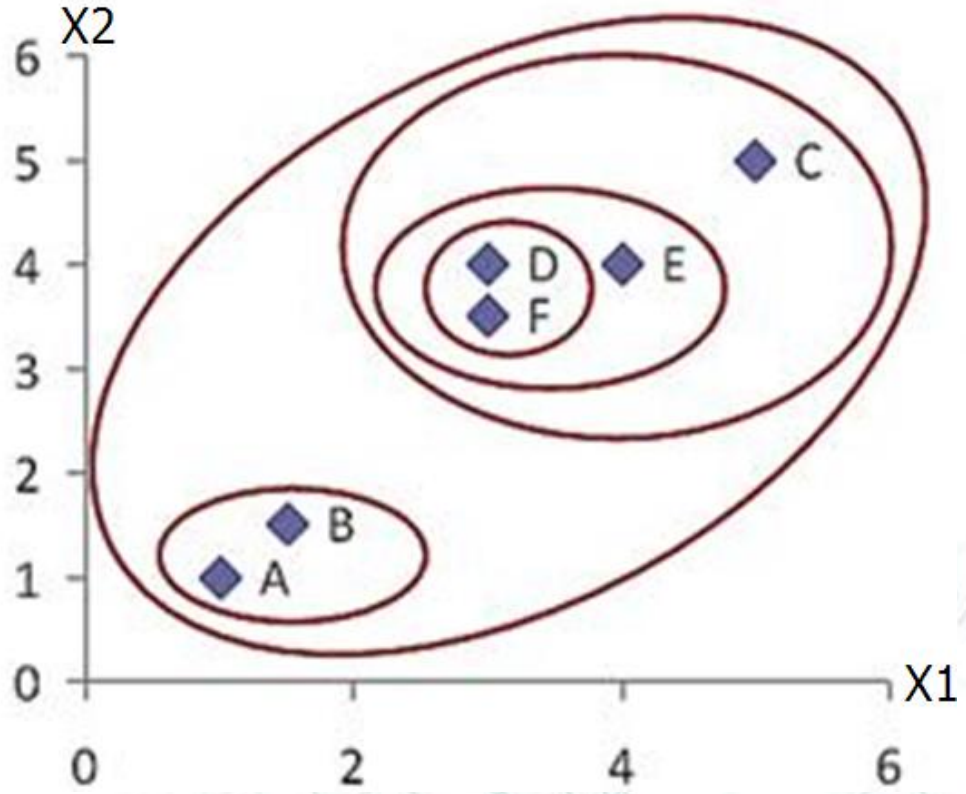
Min Distance (Single Linkage)

Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

Example

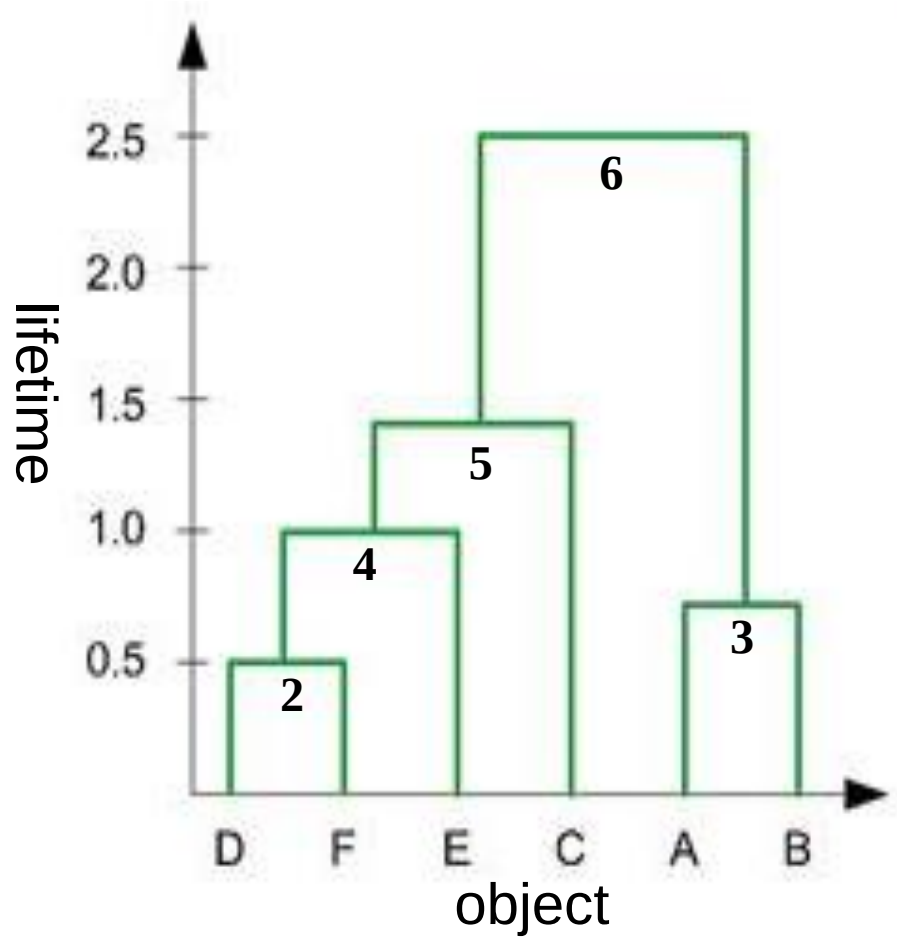
- Final result (meeting termination condition)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



Example

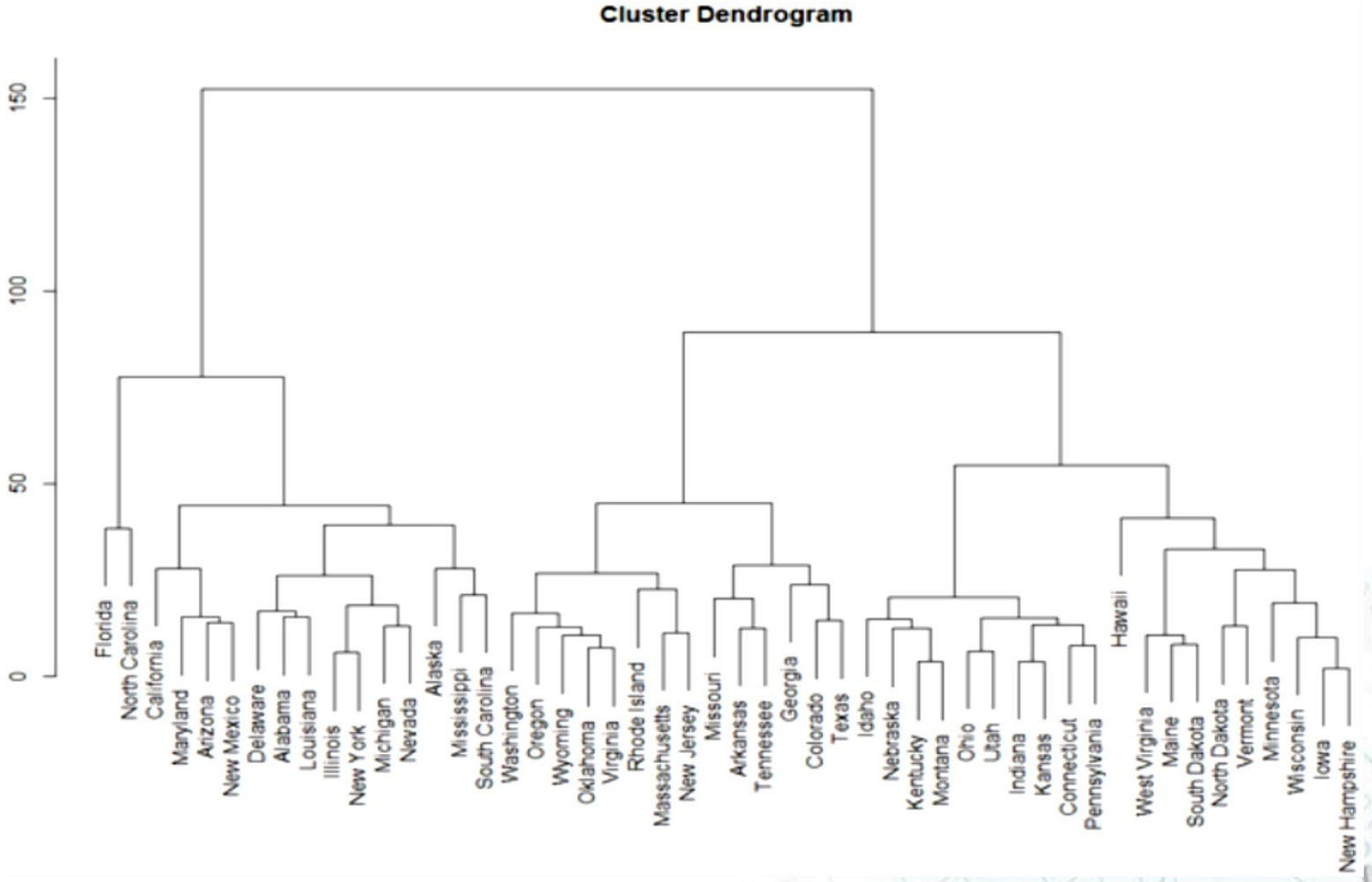
- Dendrogram tree representation



1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge clusters D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
5. We merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge clusters (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation

Example

- Dendrogram tree representation: “clustering” USA states



Exercise

Given a data set of five objects characterised by a single feature:

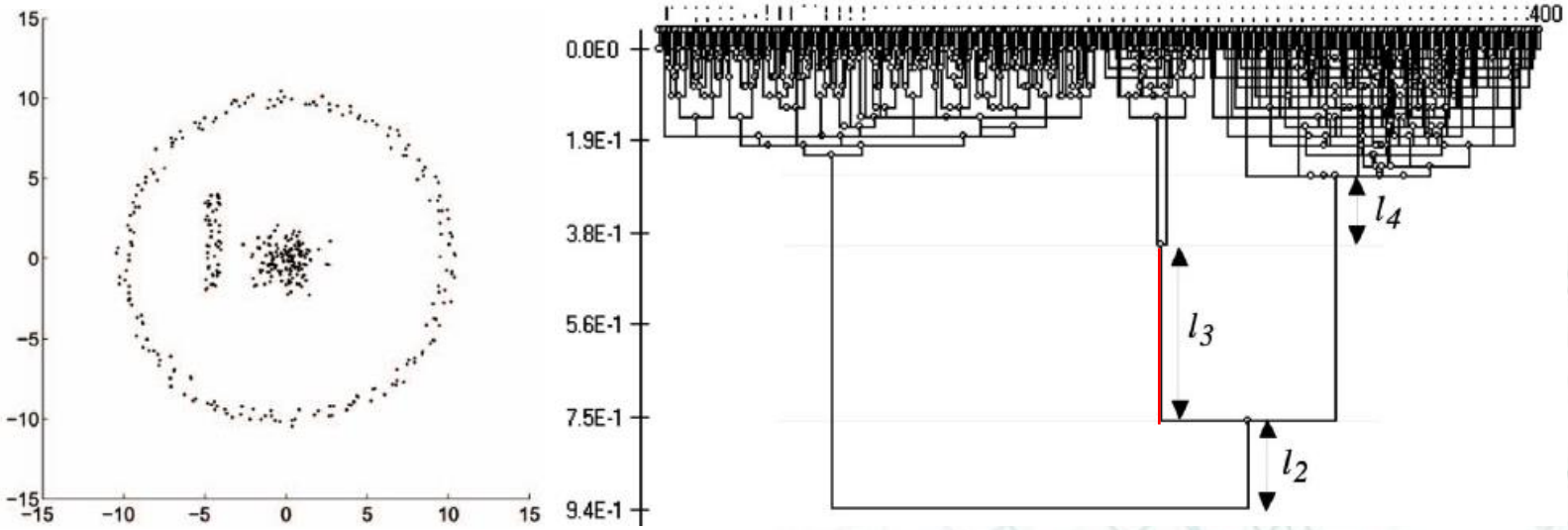
	a	b	C	d	e
Feature	1	2	4	5	6

Apply the agglomerative algorithm with single-link, complete-link and averaging cluster distance measures to produce three dendrogram trees, respectively.

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

Relevant Issues

- How to determine the number of clusters
  - If the number of clusters known, termination condition is given!
  - The *K*-cluster lifetime as the range of threshold value on the dendrogram tree that leads to the identification of *K* clusters
  - Heuristic rule: cut a dendrogram tree with maximum *K*-cluster life time



Summary

- Hierarchical algorithm is a sequential clustering algorithm
  - Use distance matrix to construct a tree of clusters (dendrogram)
  - Hierarchical representation without the need of knowing # of clusters (can set termination condition with known # of clusters)
- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Sensitive to cluster distance measures and noise/outliers
  - Less efficient: $O(n^2)$, where n is the number of total objects
- There are several variants to overcome its weaknesses
  - BIRCH: scalable to a large data set
  - ROCK: clustering categorical data
  - CHAMELEON: hierarchical clustering using dynamic modelling

K-means Clustering

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)

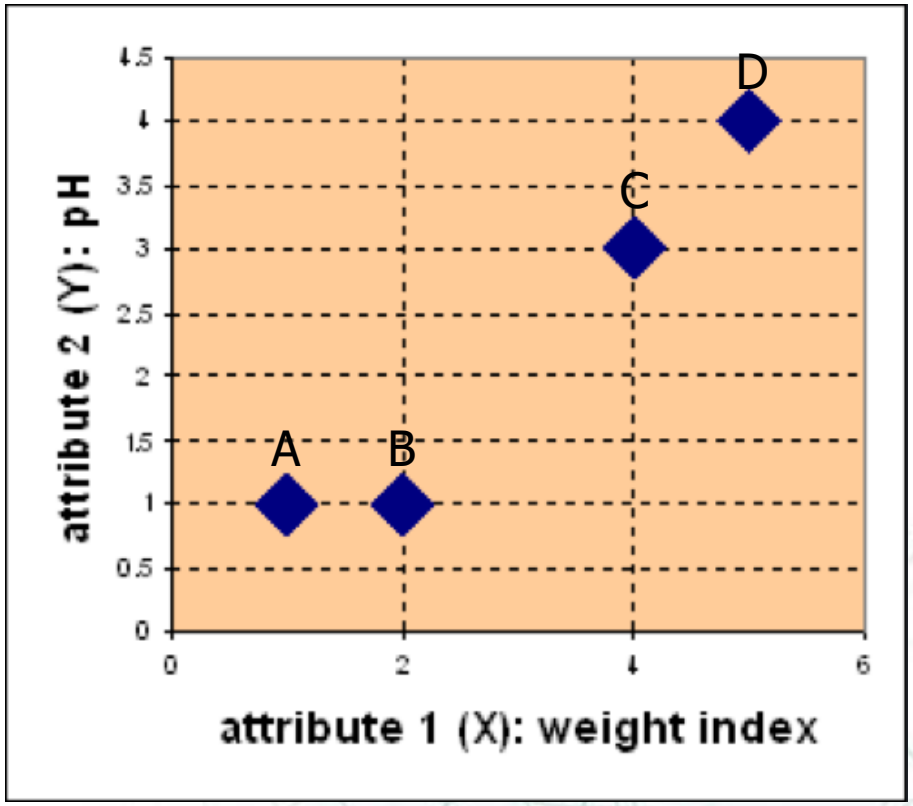
1) Assign each object to the cluster of the nearest seed point measured with a specific distance metric
2) Compute seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., *mean point*, of the cluster)
3) Go back to Step 1), stop when no more new assignment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

Example

- Problem

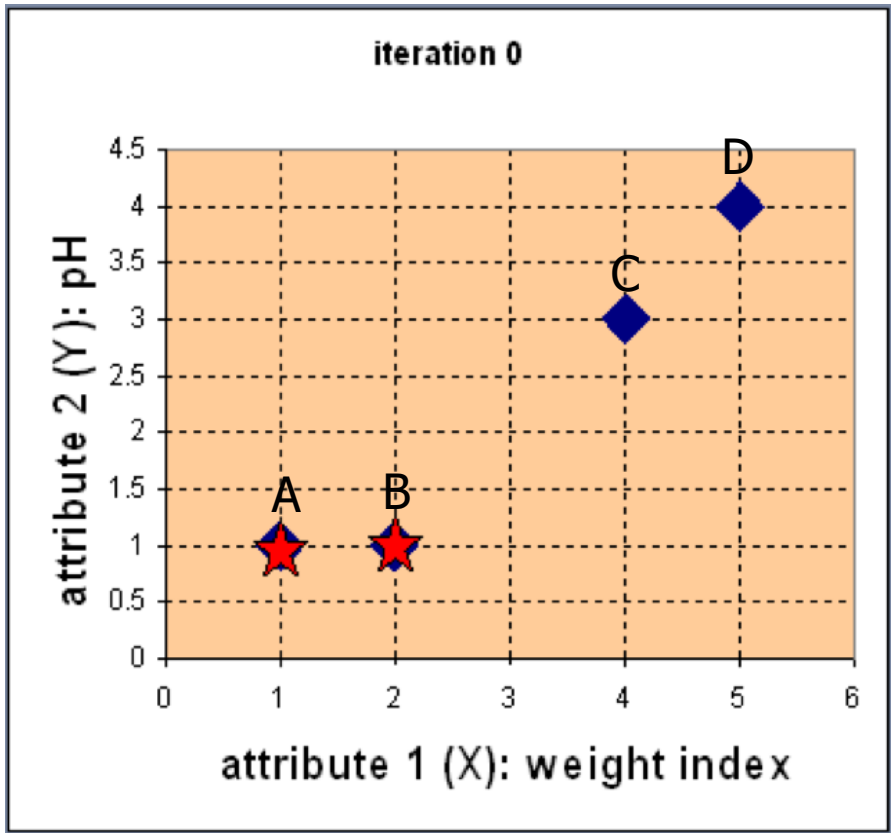
  Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into *K=2* group of medicine.

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



$$c_1 = A, c_2 = B$$

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

Euclidean distance

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

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

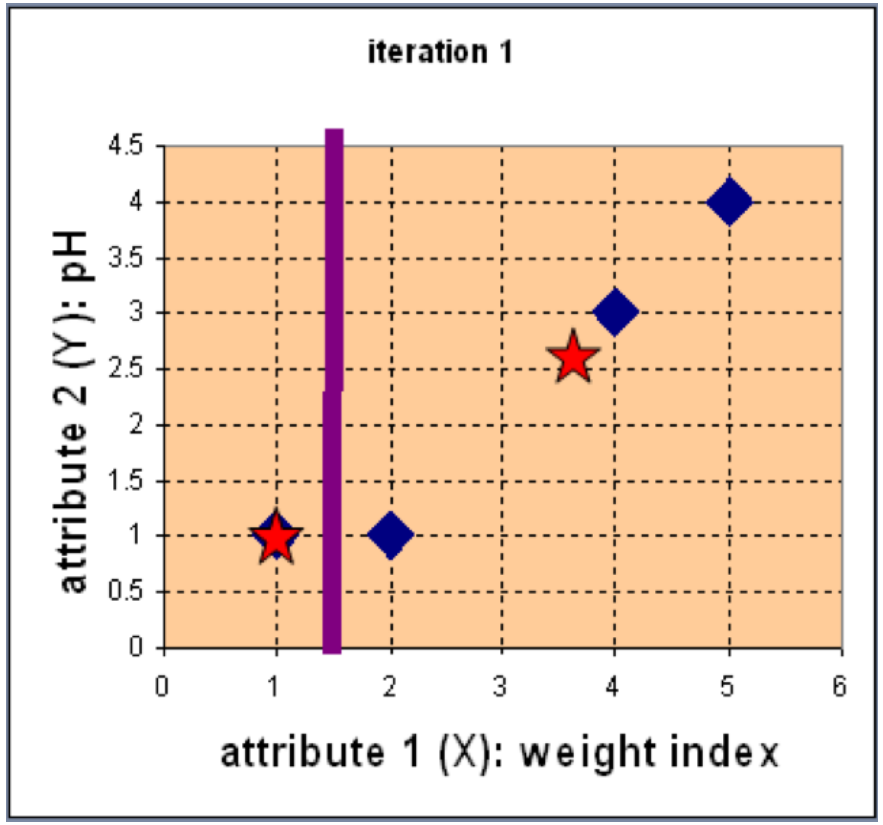
$$d(D,c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

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

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.

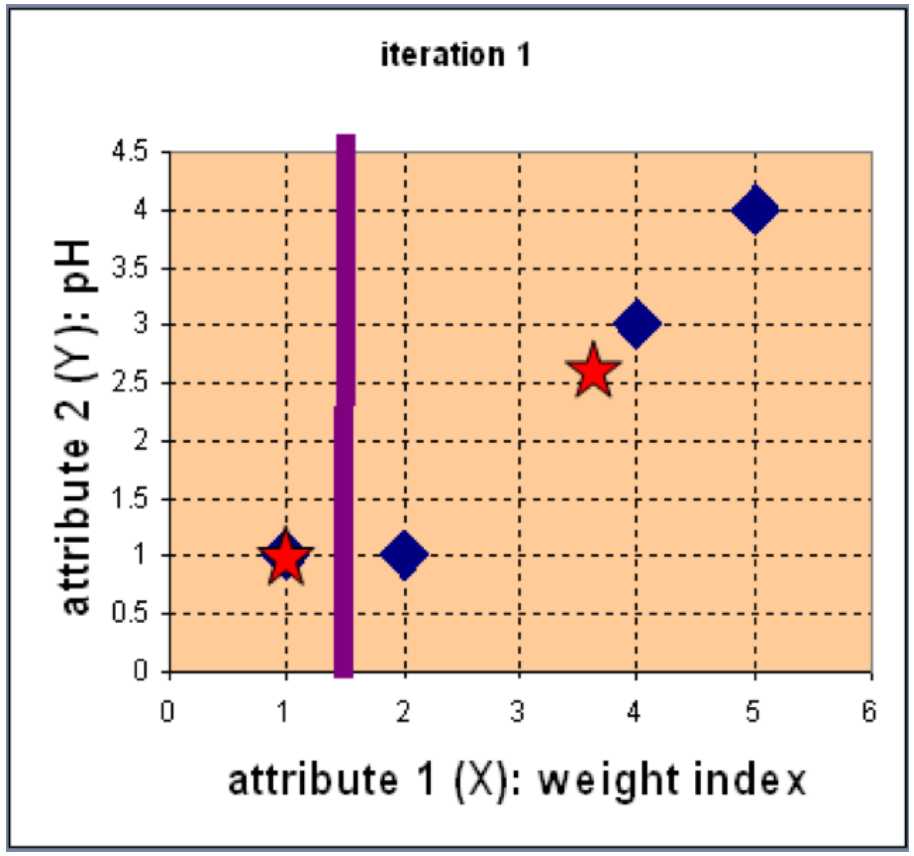
$$c_1 = (1,\ 1)$$

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

$$= \left(\frac{11}{3}, \frac{8}{3}\right)$$

Example

- Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

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

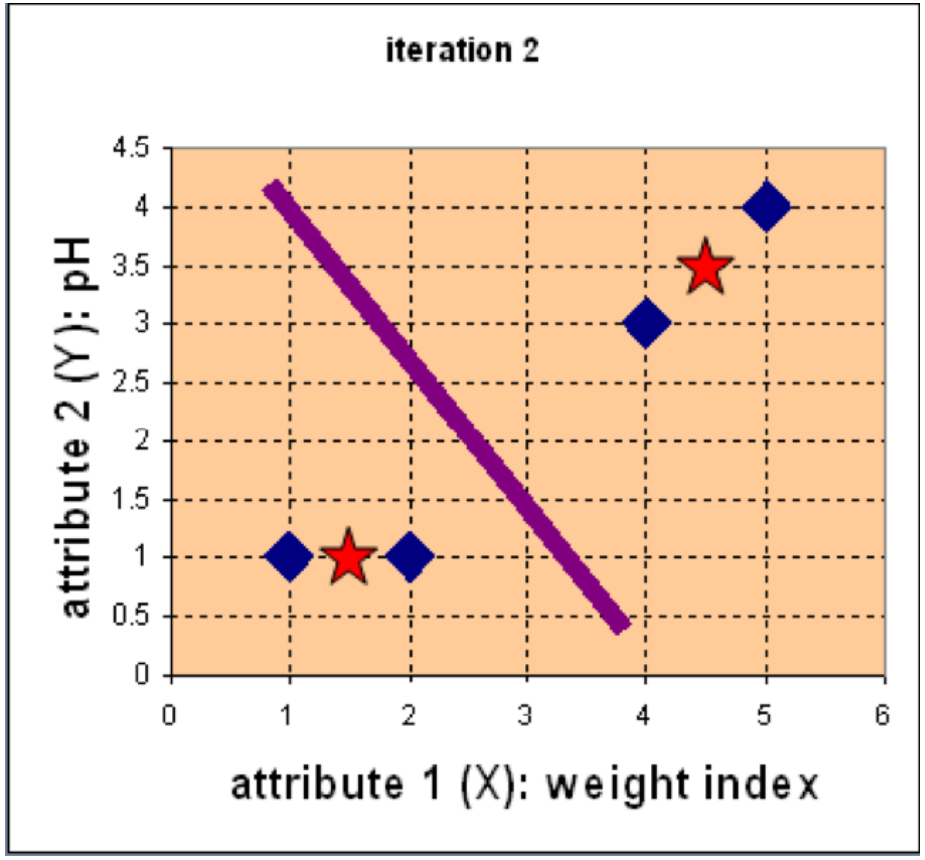
$$\begin{matrix} A & B & C & D \end{matrix}$$

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

Assign the membership to objects

Example

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



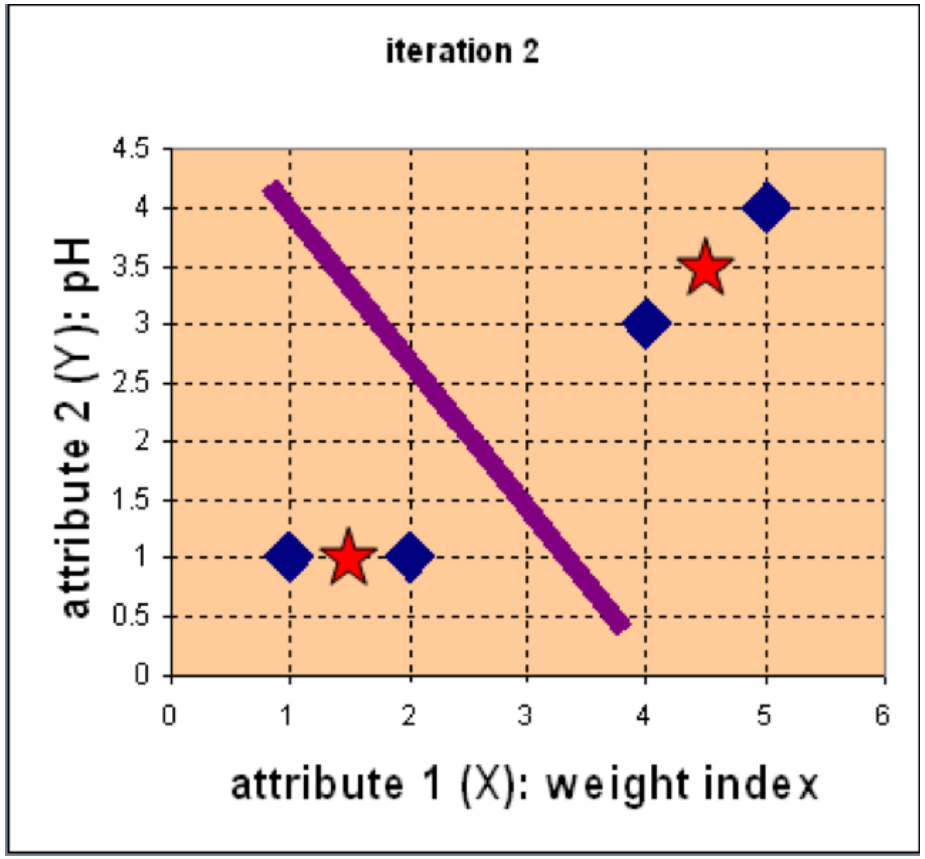
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

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

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

Example

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



Compute the distance of all objects to the new centroids

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

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

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

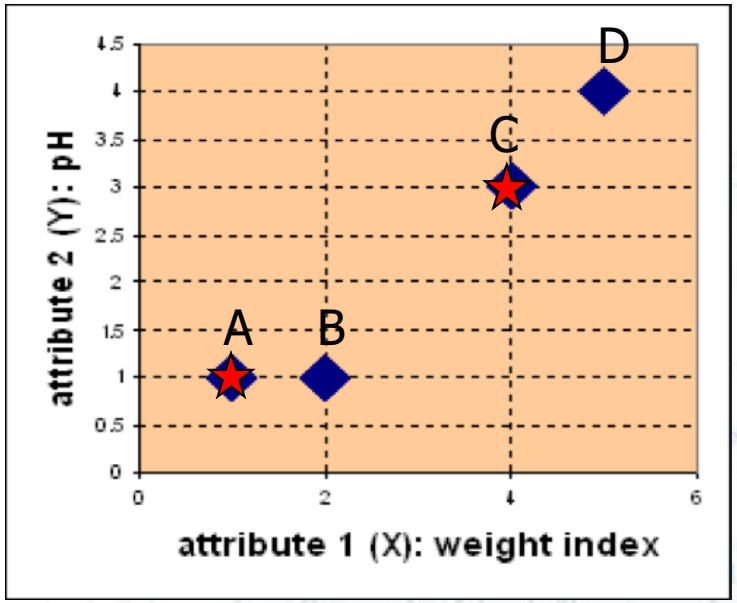
Stop due to no new assignment
Membership in each cluster no longer change

Exercise

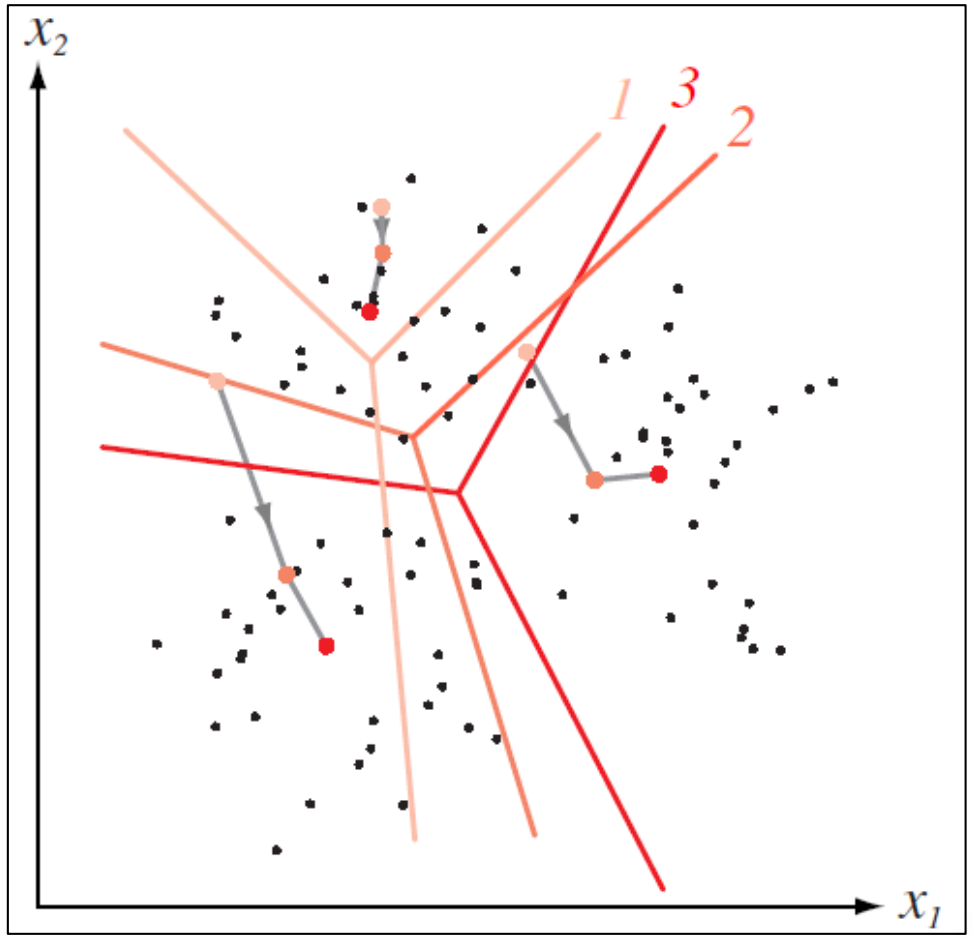
For the medicine data set, use K-means with the Manhattan distance metric for clustering analysis by setting $K=2$ and initialising seeds as C_1 = A and C_2 = C. Answer three questions as follows:

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?



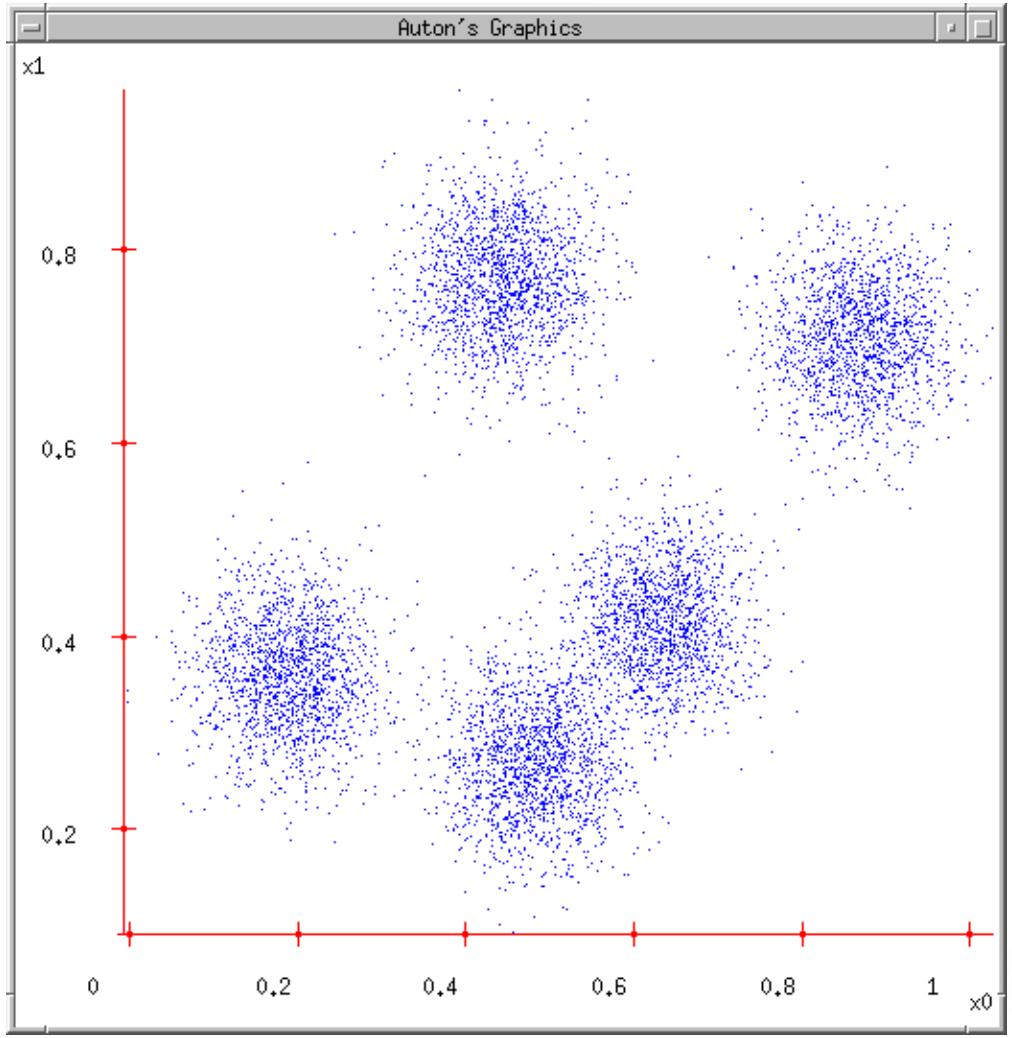
When *K* centroids are set/fixed, they partition the whole data space into *K* mutually exclusive subspaces to form a partition.

A partition amounts to a

Voronoi Diagram

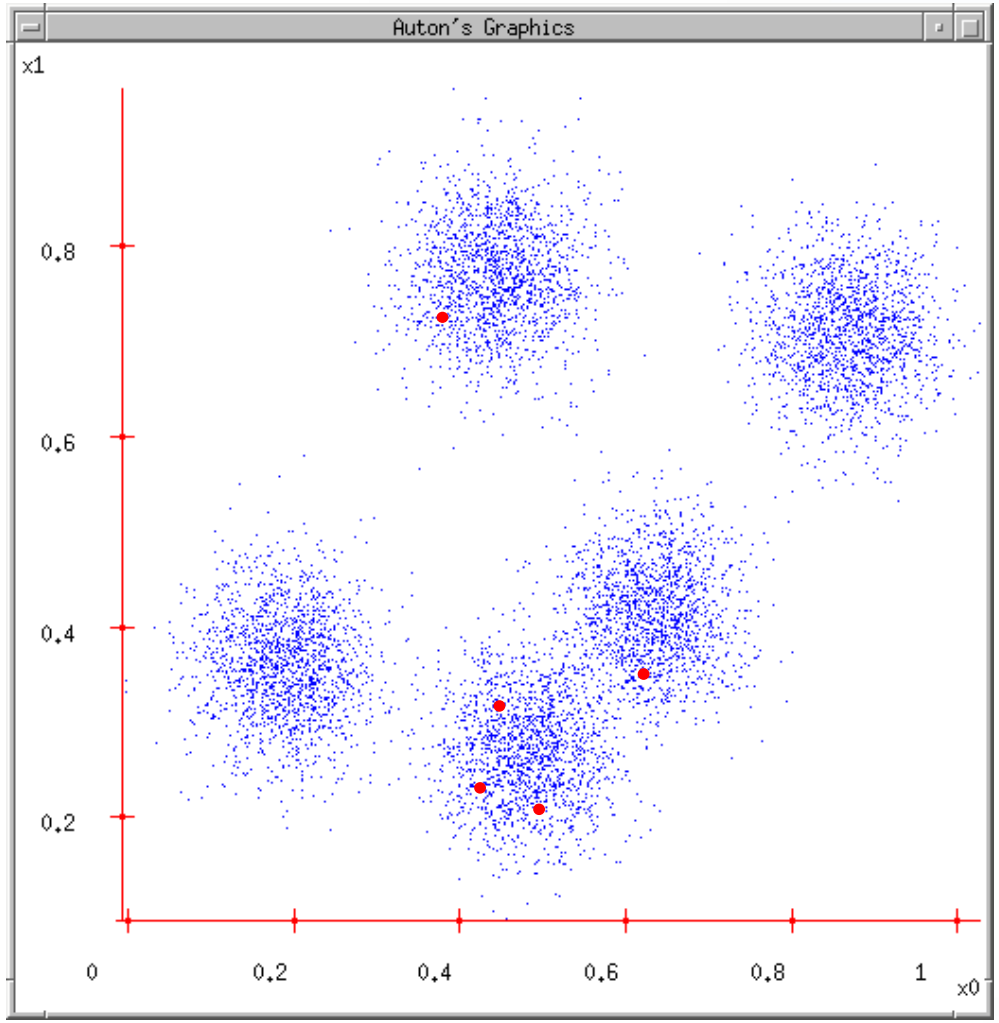
Changing positions of centroids leads to a new partitioning.

K-means Demo



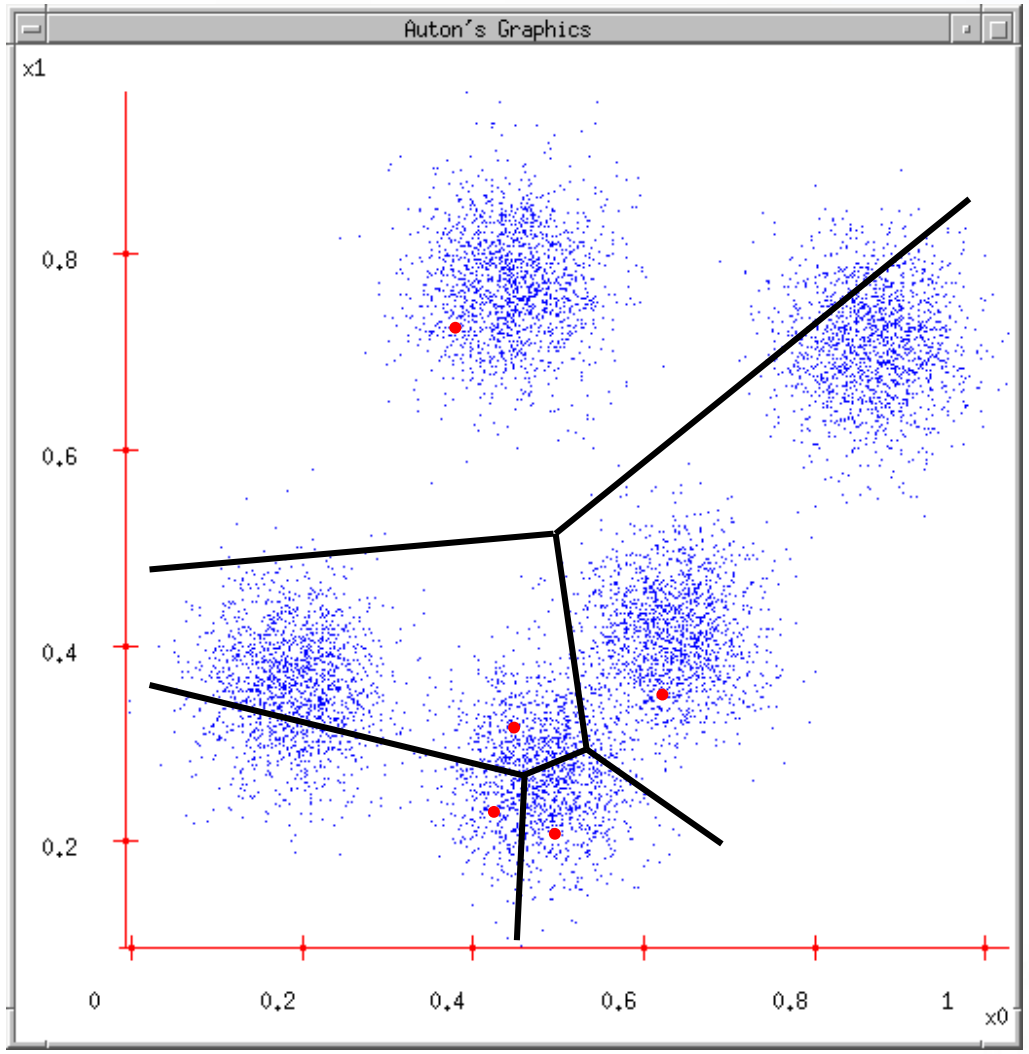
1. User set up the number of clusters they'd like. *(e.g. k=5)*

K-means Demo



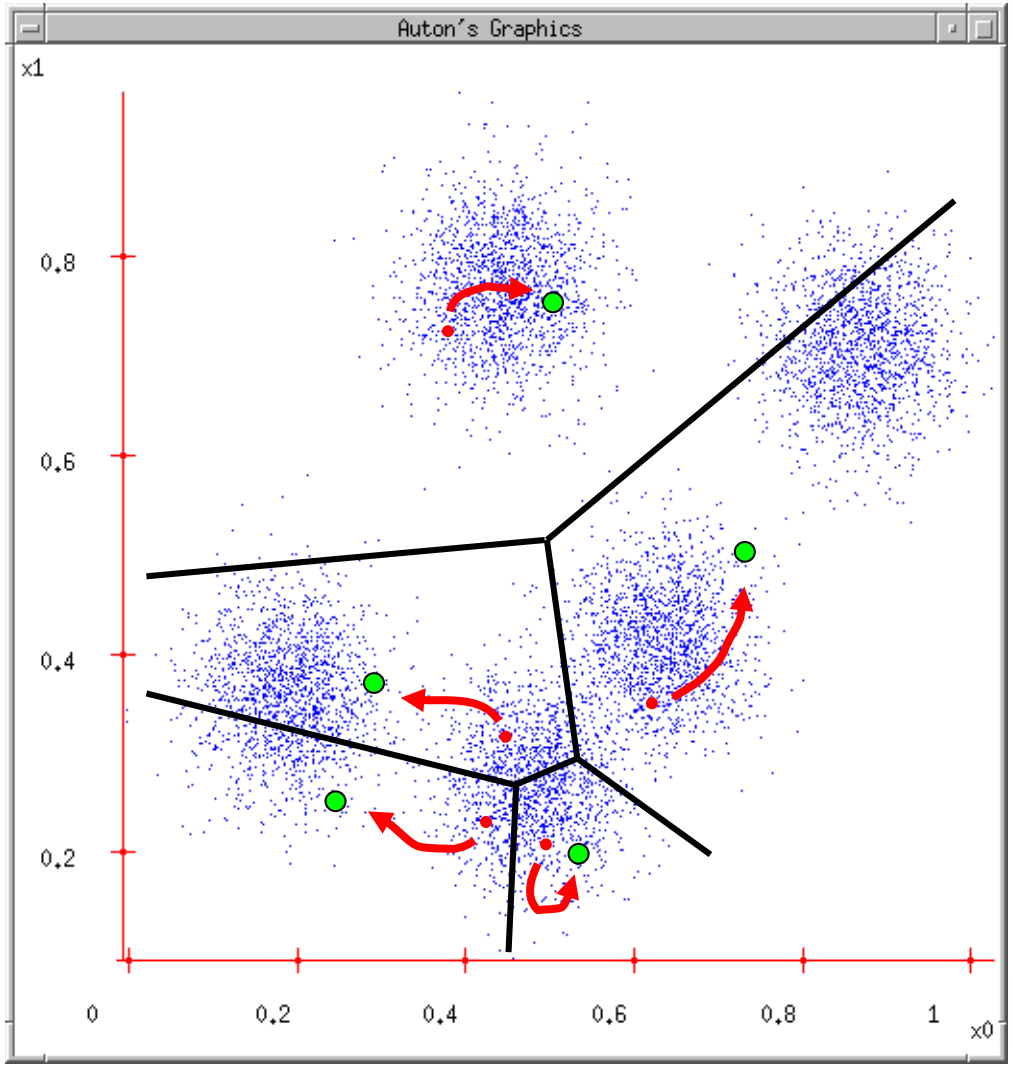
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations

K-means Demo



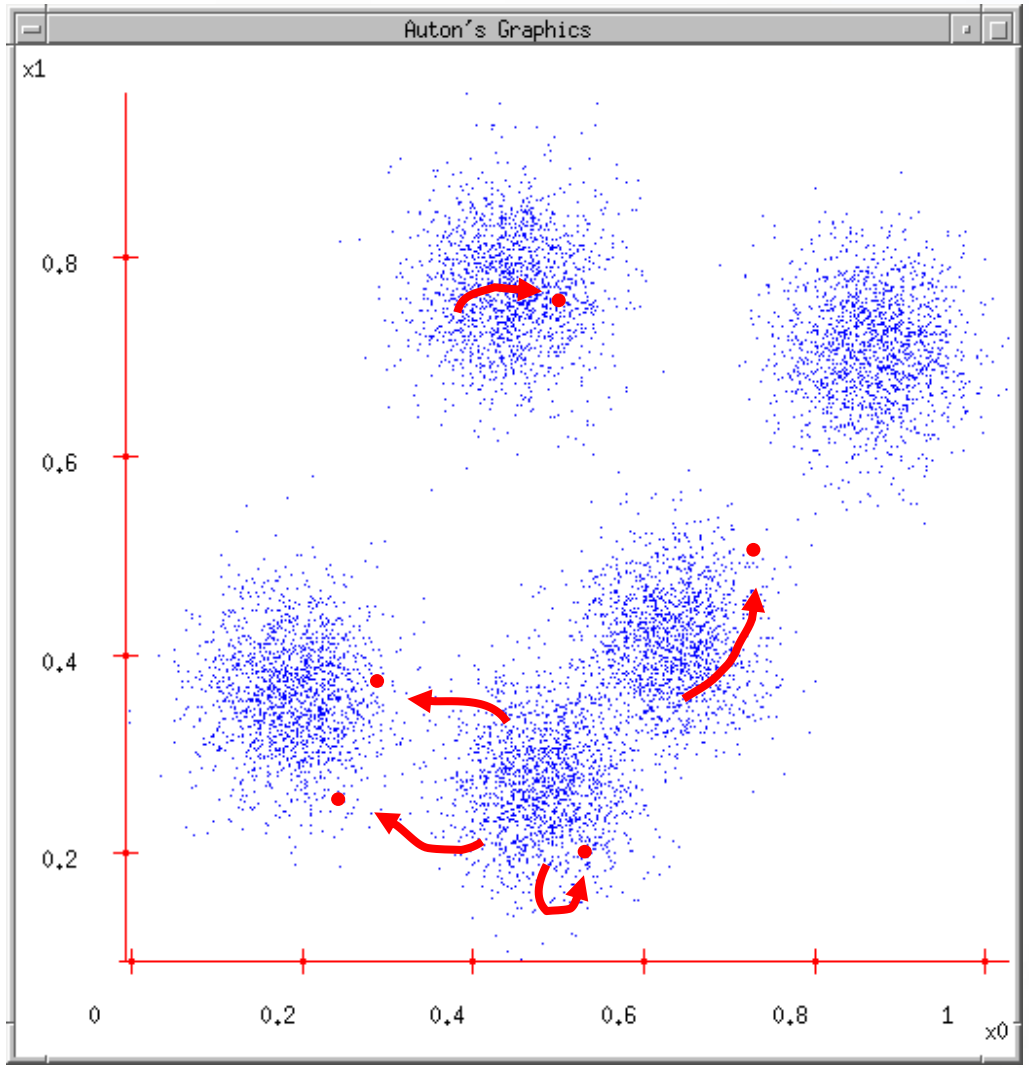
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster Center locations
3. Each data point finds out which Center it's closest to. (Thus each Center "owns" a set of data points)

K-means Demo



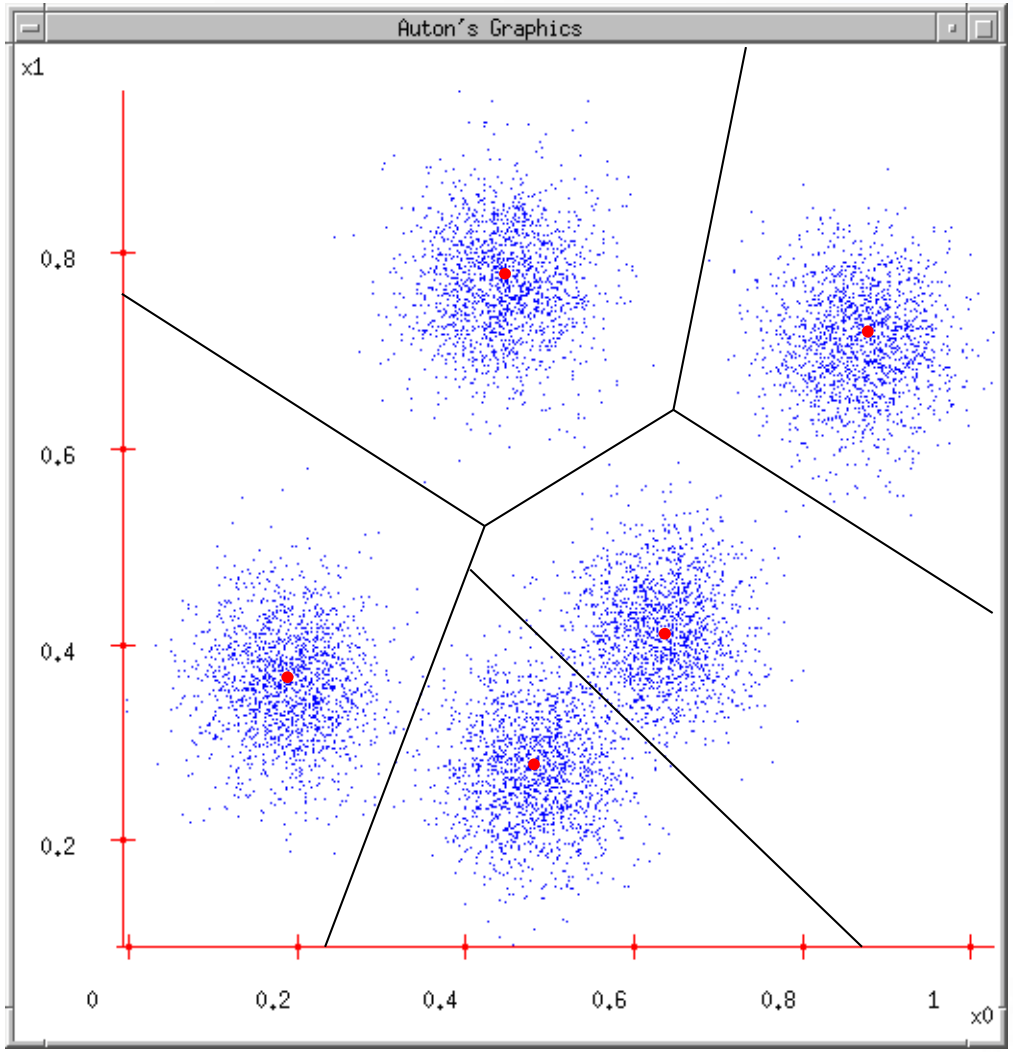
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each Center "owns" a set of data points)
4. Each centre finds the centroid of the points it owns

K-means Demo



1. User set up the number of clusters they'd like. (e.g. *K=5*)
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there

K-means Demo



1. User set up the number of clusters they'd like. (e.g. *K=5*)
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there
6. …Repeat until terminated!

K-means Demo Weka

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?

Application

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

  Step 1: Read Image

  Step 2: Convert Image from RGB Colour Space to L*a*b* Colour Space

  Step 3: Classify the Colours in 'a*b*' Space Using *K*-means Clustering

  Step 4: Label Every Pixel in the Image Using the Results from *K*-means Clustering (KMEANS)

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

  Step 6: Segment the Nuclei into a Separate Image

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