Clustering

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.

Clustering: a definition

"The process of organizing objects into *groups* whose members are *similar in some way*"

J.A. Hartigan, 1975

"An algorithm by which objects are grouped in *classes*, so that intra-class *similarity* is maximized and inter-class similarity is minimized"

J. Han and M. Kamber, 2000

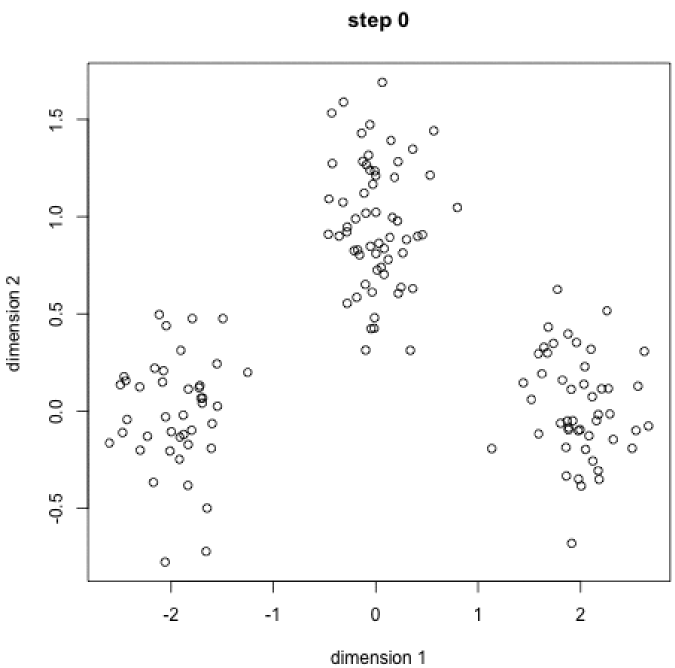
"... grouping or segmenting a collection of objects into subsets or *clusters*, such that those within each cluster are more closely *related* to one another than objects assigned to different clusters"

T. Hastie, R. Tibshirani, J. Friedman, 2009

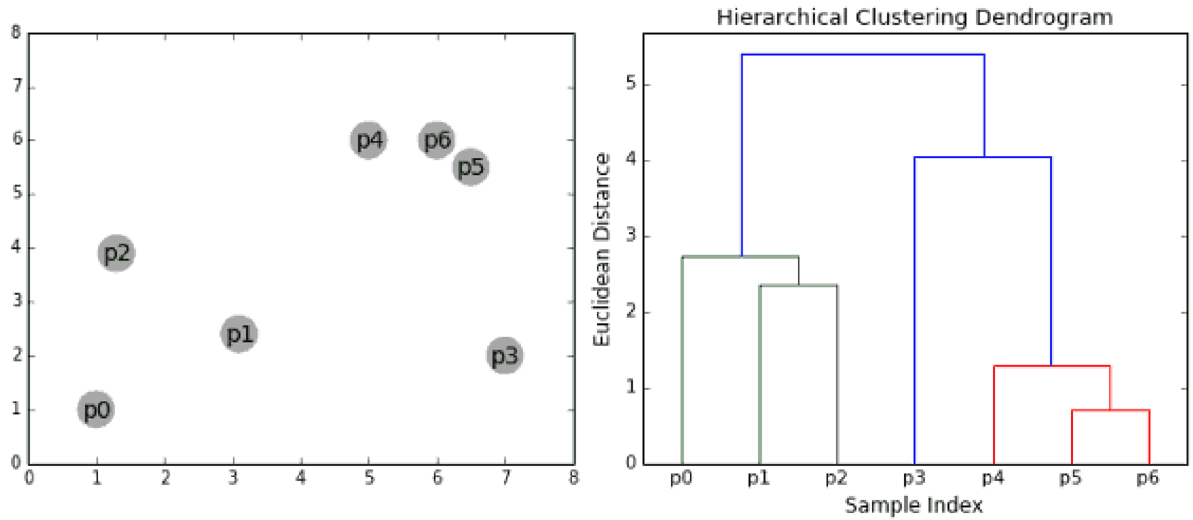
(Some) Applications of Clustering

- Market research/Insurance/Telephone Companies
  - find groups of customers with similar behavior for targeted advertising
- Biology
  - classification of plants and animals given their features
- Social Media
  - identify suggestions
  - cluster the blocked users
- On the Web:
  - document classification
  - cluster Web log data to discover groups of similar access patterns
  - recommendation systems ("If you liked this, you might also like that")

Types of Clustering

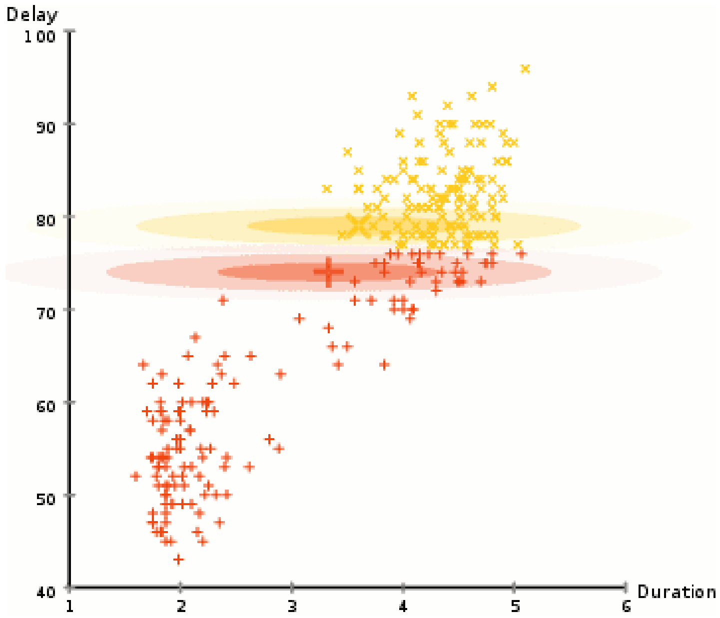


Centroid-based Clustering
(K-means)

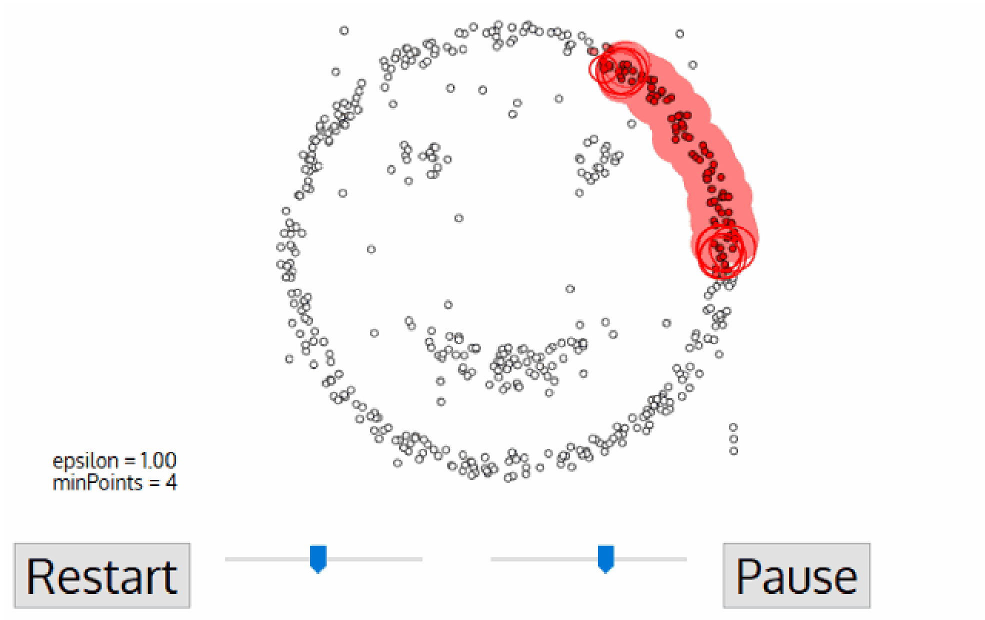


Hierarchical Clustering
(Agglomerative Hierarchical Clustering)

Types of Clustering



Distribution-based Clustering
(Gaussian Mixture-model based)



Density-based Clustering
(DB-SCAN)

Distance Measures

Two major classes of distance measure:

- Euclidean
  - A Euclidean space has some number of real-valued dimensions and "dense" points
  - There is a notion of *average* of two points
  - A Euclidean distance is based on the locations of points in such a space $d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$
- Non-Euclidean
  - A Non-Euclidean distance is based on properties of points, but not on their *location* in a space

Distances vs Similarities

- Distances are normally used to measure the similarity or dissimilarity between two data objects...
- ... However they are two different things!
- i.e. Dissimilarities can be judged by a set of users in a survey
    - they do not necessarily satisfy the triangle inequality
    - they can be 0 even if two objects are not the same
    - they can be asymmetric (in this case their average can be calculated)

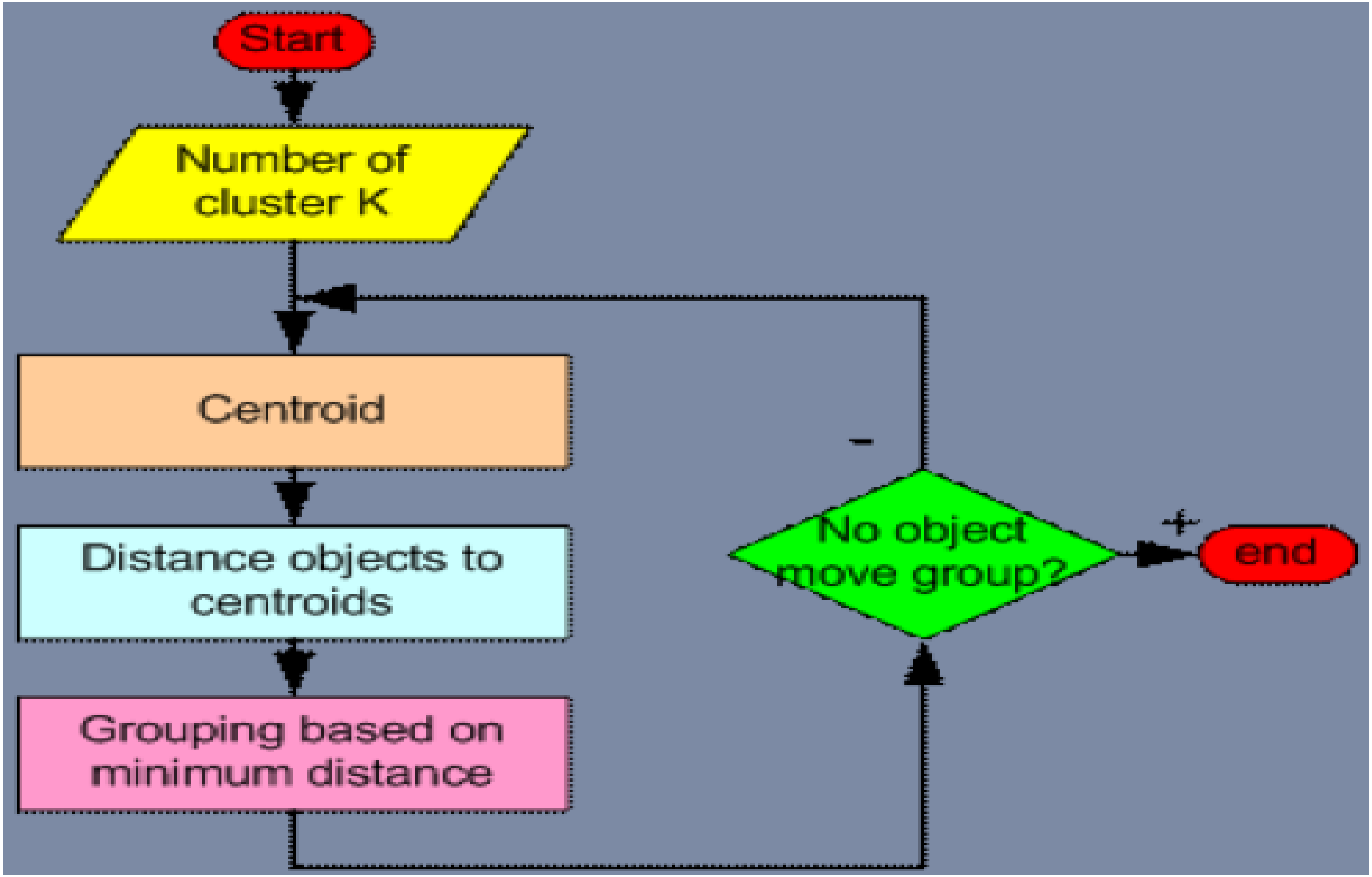
K-Means Algorithm

- One of the simplest unsupervised learning algorithms
- Assumes Euclidean space *(works with numeric data only)*
- Number of clusters fixed a priority
- **How does it work?**

K-Means Algorithm

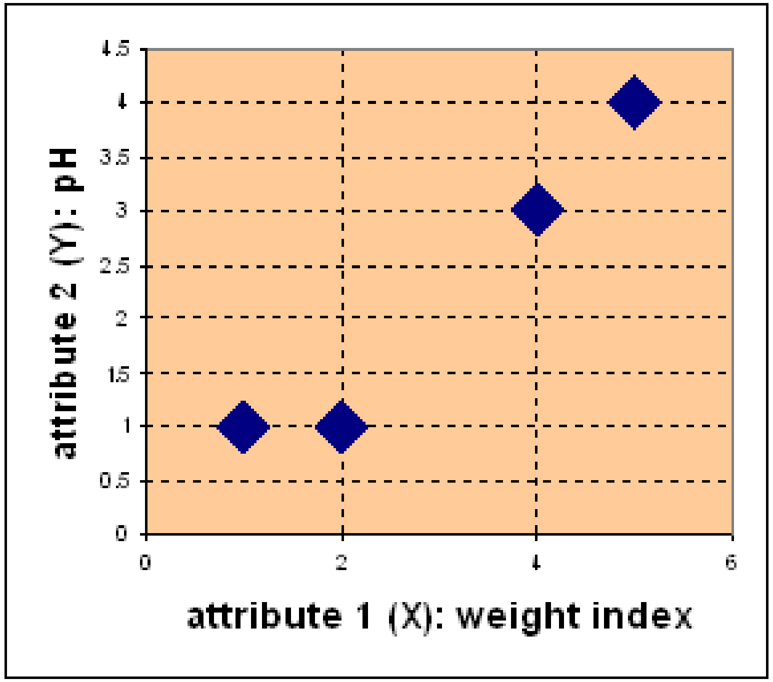
- One of the simplest unsupervised learning algorithms
- Assumes Euclidean space *(works with numeric data only)*
- Number of clusters fixed a priority
- **How does it work?**

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group *centroids*.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

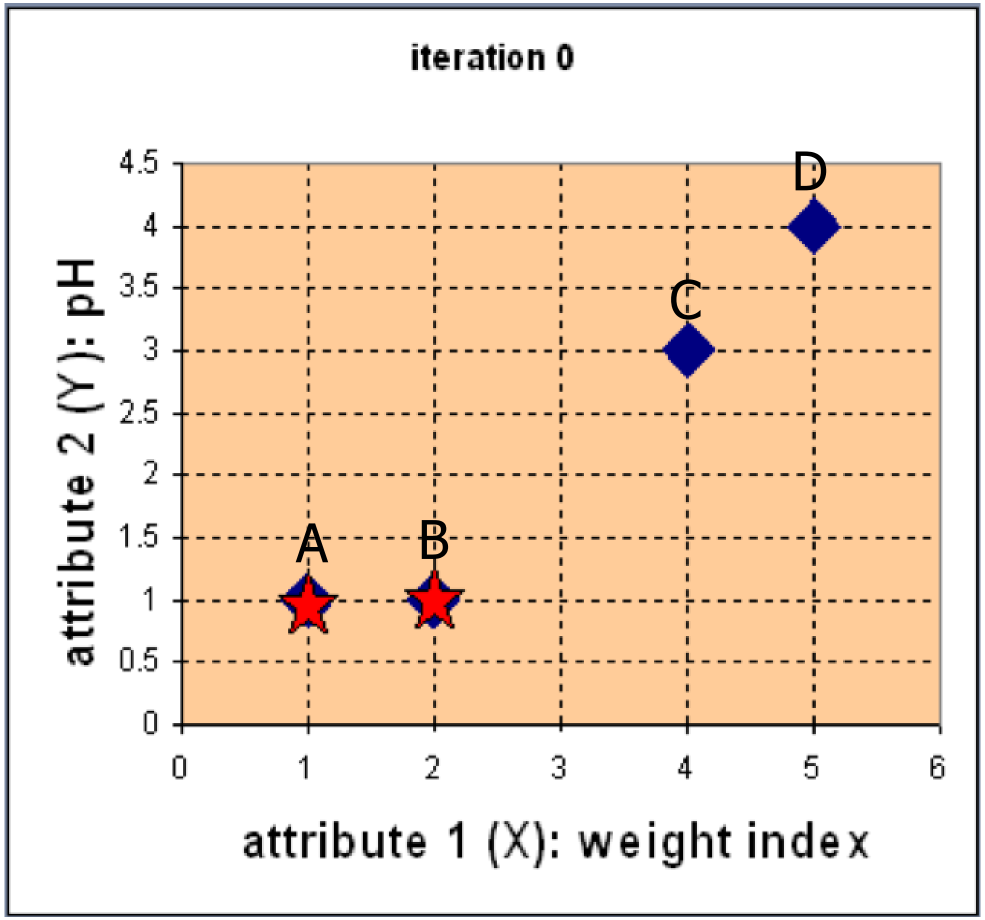
K-Means: A numerical example

Object	Attribute 1 (X)	Attribute 2 (Y)
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4



Example

Step 1: Use initial seed points for partitioning



$$c_1 = A, c_2 = B$$

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

Euclidean distance

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

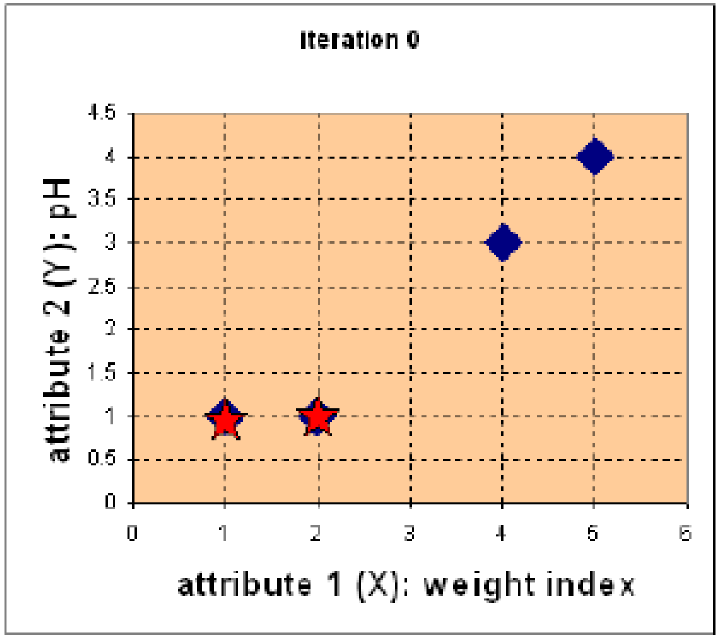
$$d(D, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

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

Assign each object to the cluster with the nearest seed point

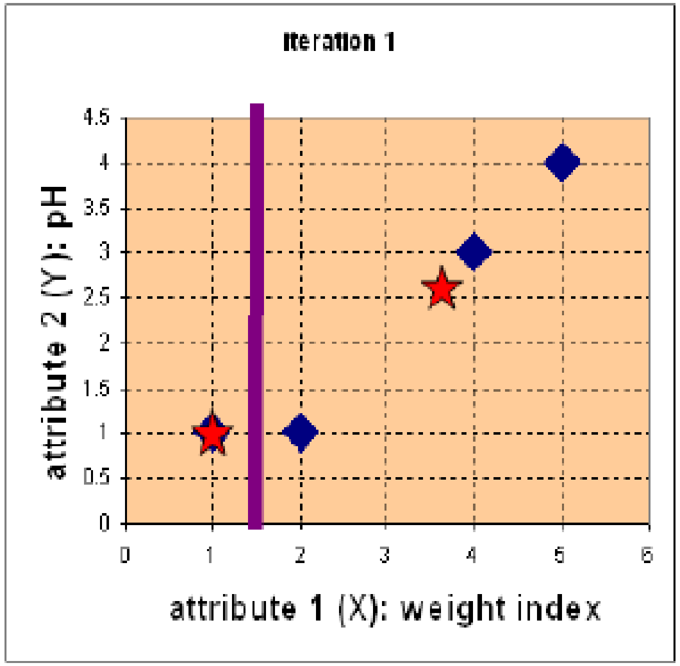
K-Means: A numerical example

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



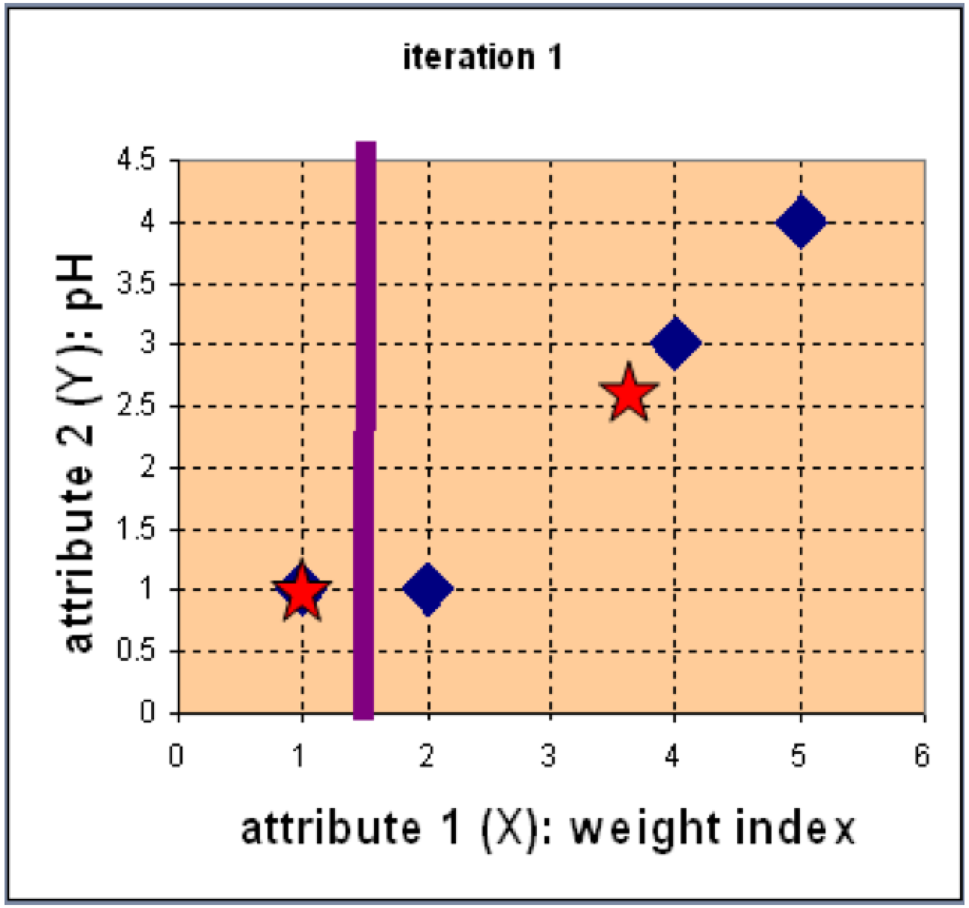
K-Means: A numerical example

- Object Clustering
  - $$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{array}{l} group1 \\ group2 \end{array}$$



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.

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

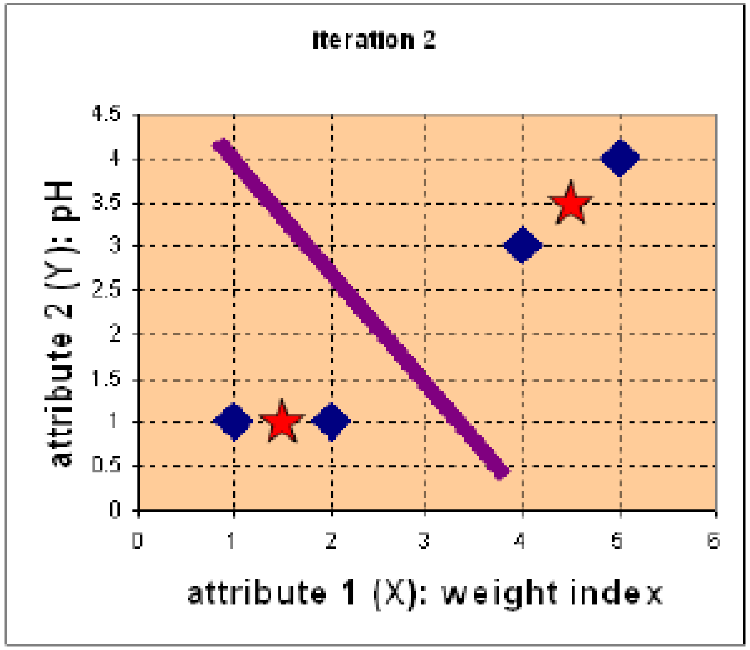
$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$

$$= (11/3,\ 8/3)$$

$$= (3.67,\ 2.67)$$

K-Means: A numerical example

- $D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix}$ $\begin{array}{l} c_1 = (1, 1) \\ c_2 = (\frac{11}{3}, \frac{8}{3}) \end{array}$
- $G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \Rightarrow$ $\begin{array}{l} c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5, 1) \\ c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4.5, 3.5) \end{array}$

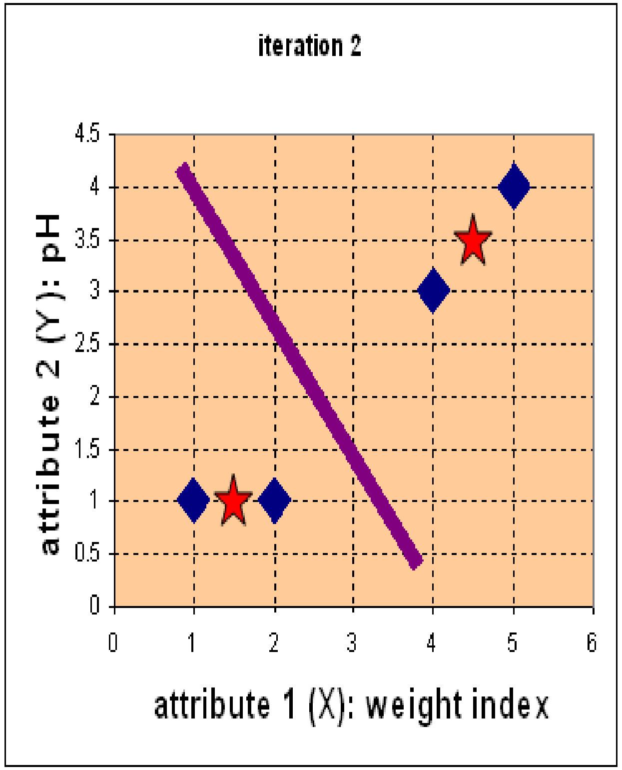


Next: determine next centroids:

Now we repeat the step to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new centroids are

$$\mathbf{c}_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\tfrac{1}{2}, 1)$$

$$\mathbf{c}_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\tfrac{1}{2}, 3\tfrac{1}{2})$$



Repeat step 2 again, we have new distance matrix at iteration 2 as

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

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

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

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

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

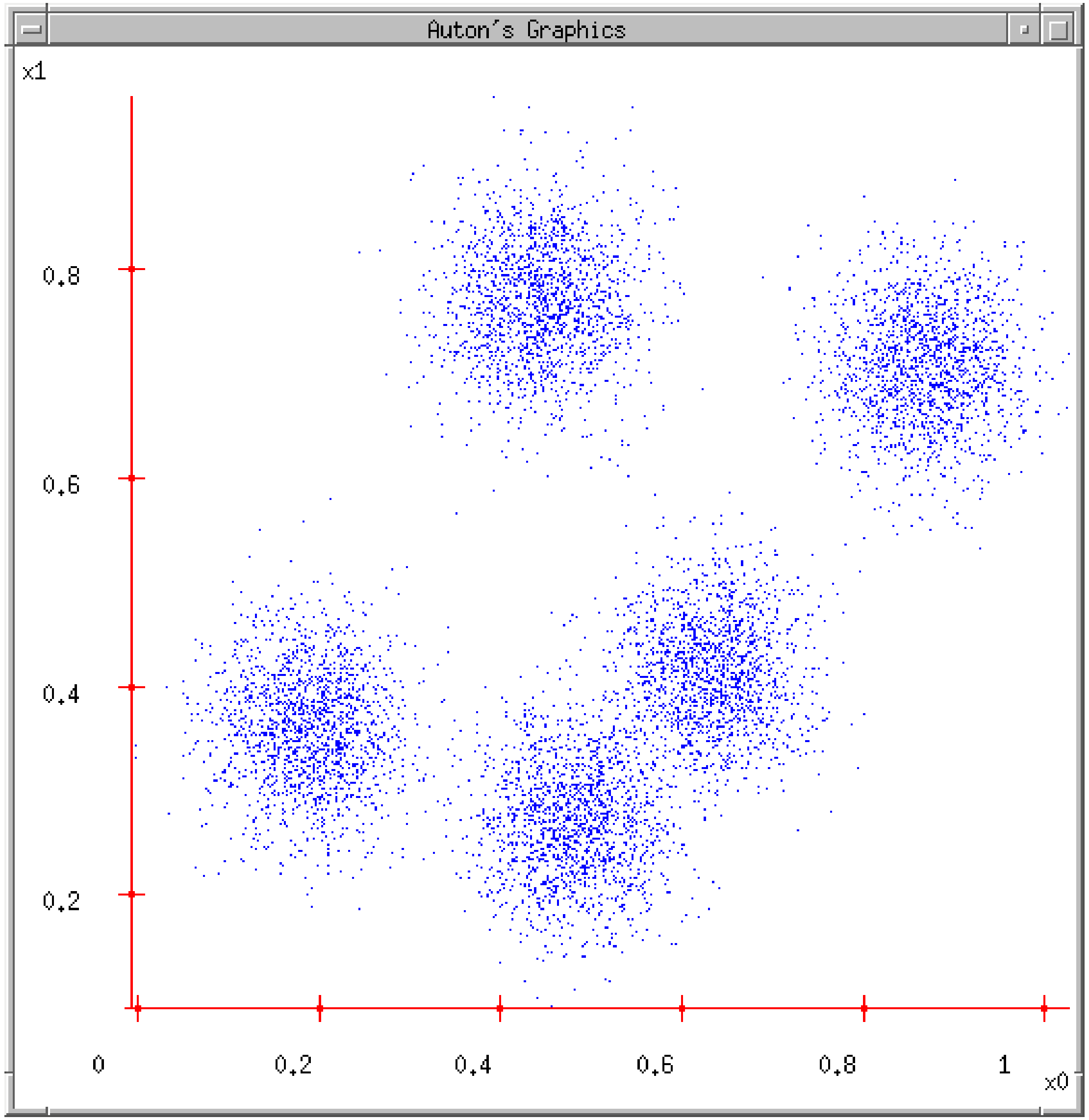
$$\mathbf{G}^2 = \mathbf{G}^1$$

- Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore.
- Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed..

We get the final grouping as the results as:

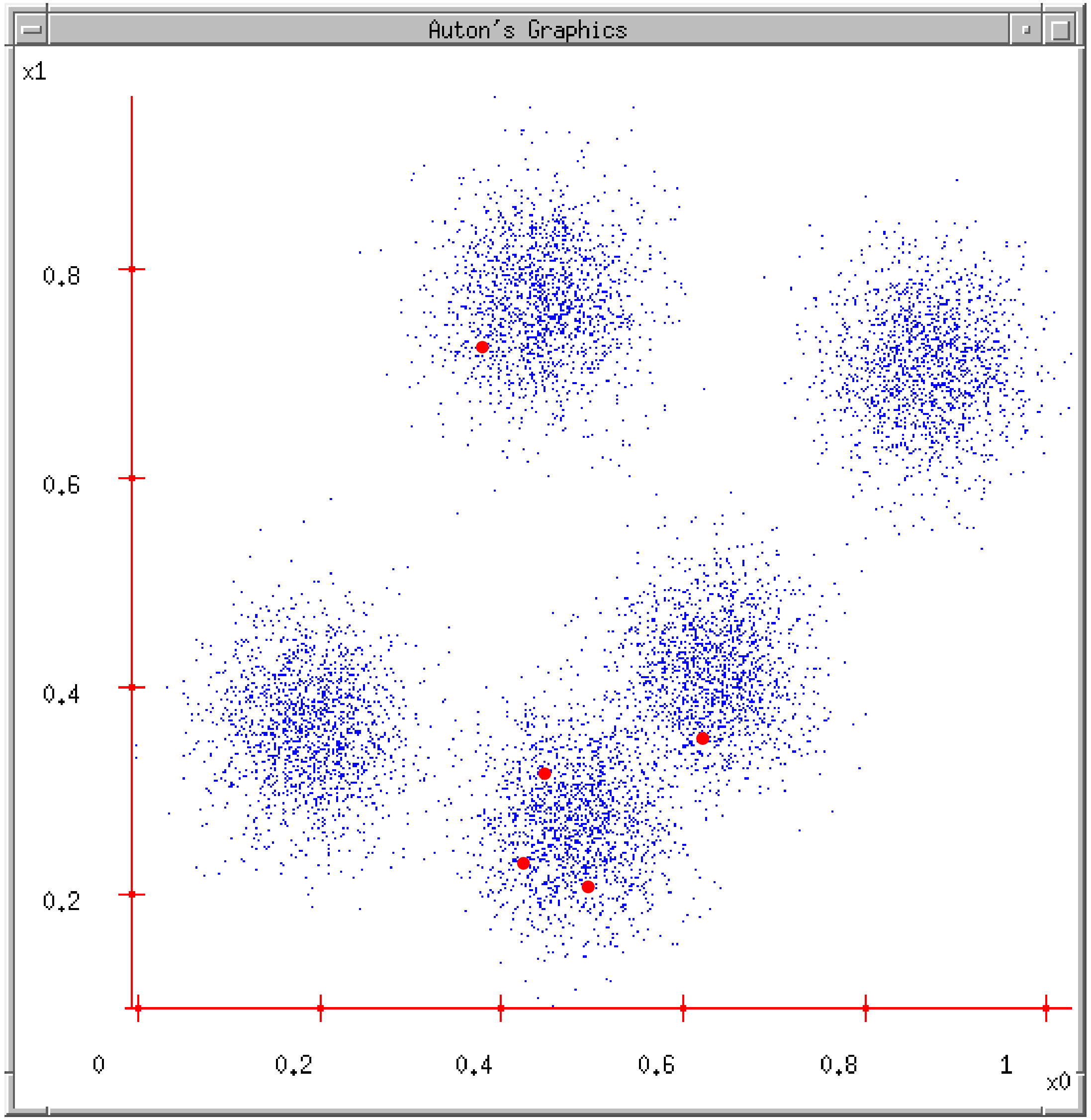
Object	Group (result)
A	1
B	1
C	2
D	2

K-means Demo



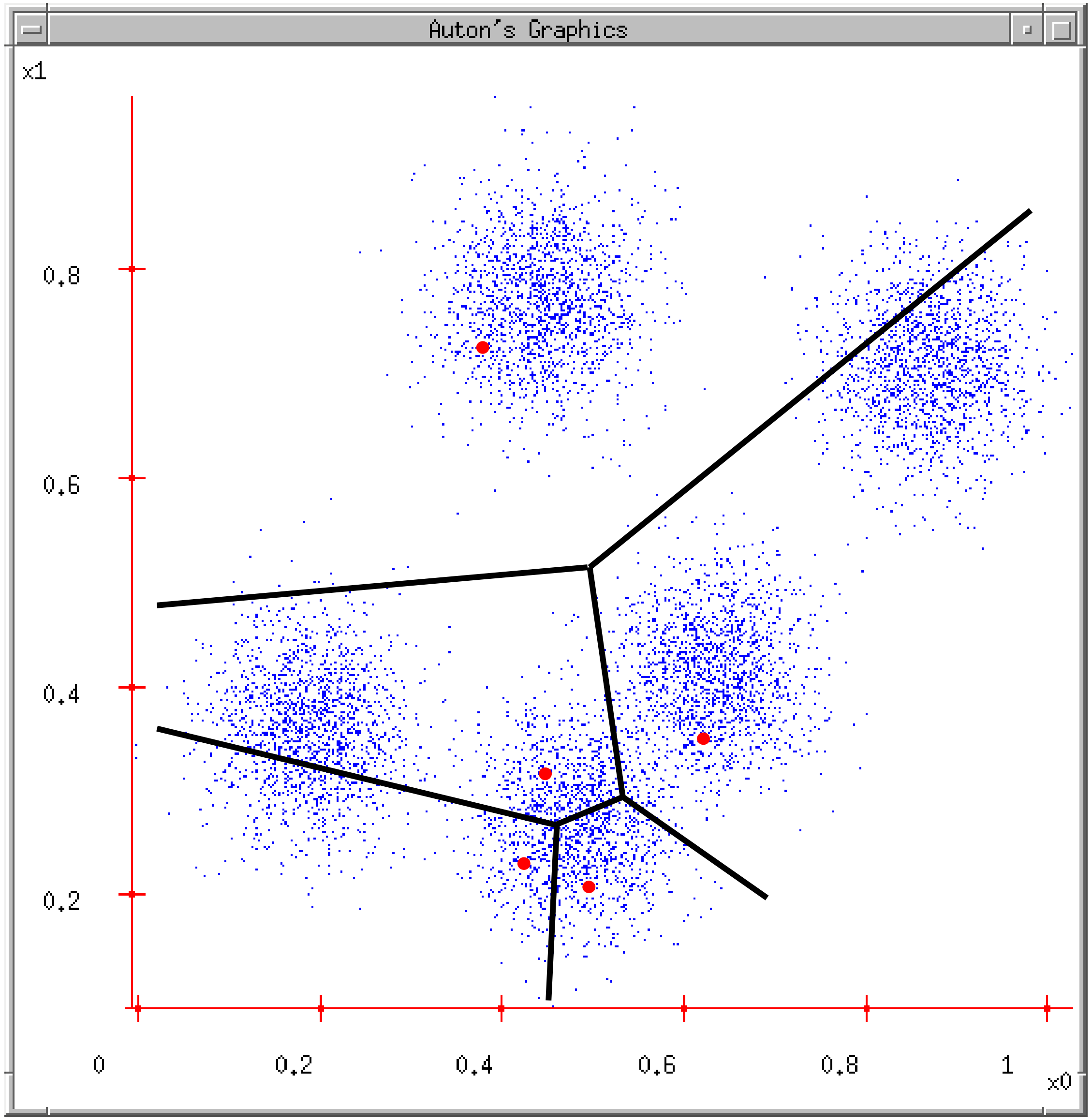
1. User set up the number of clusters they'd like. *(e.g. k=5)*

K-means Demo



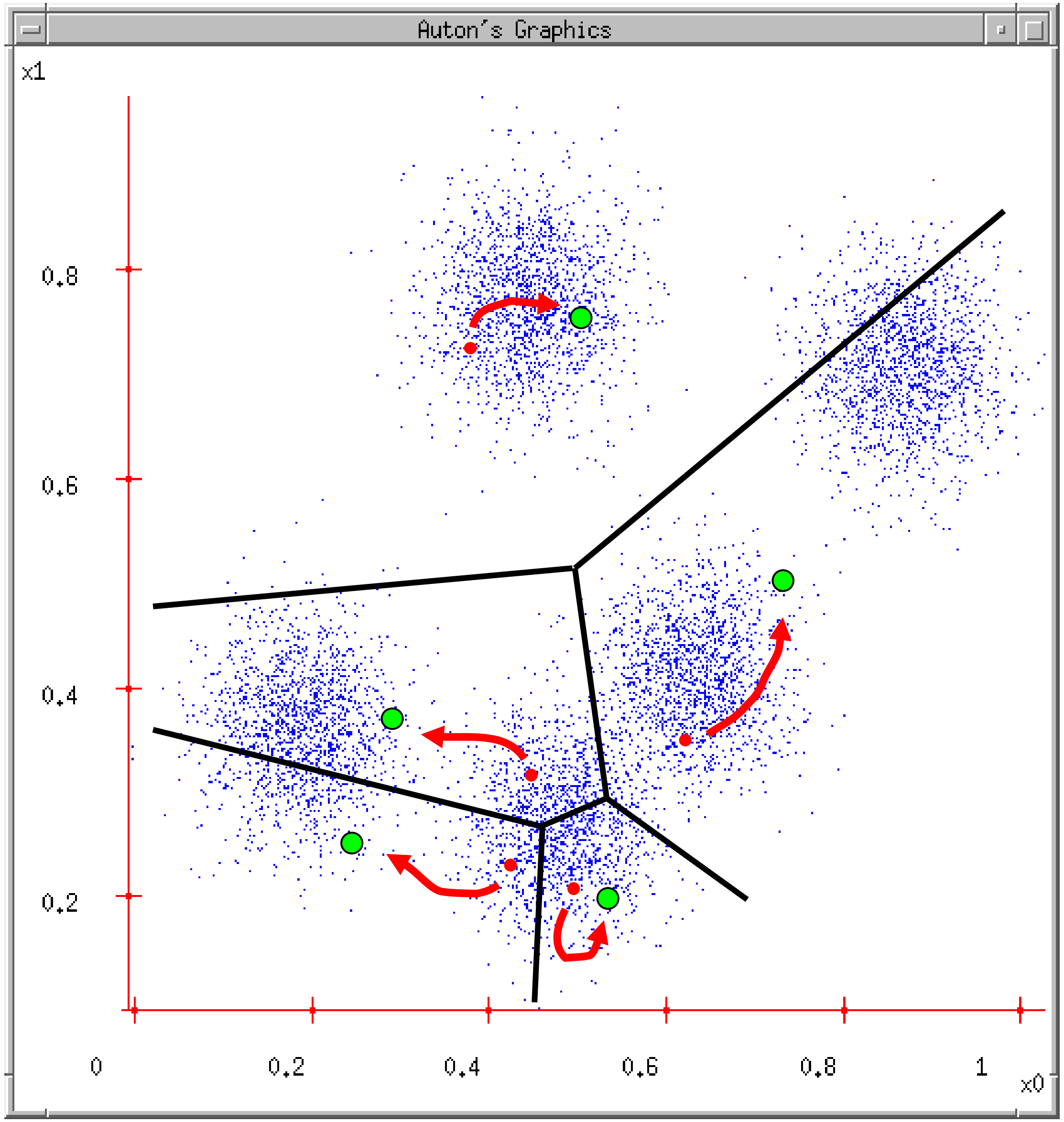
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations

K-means Demo



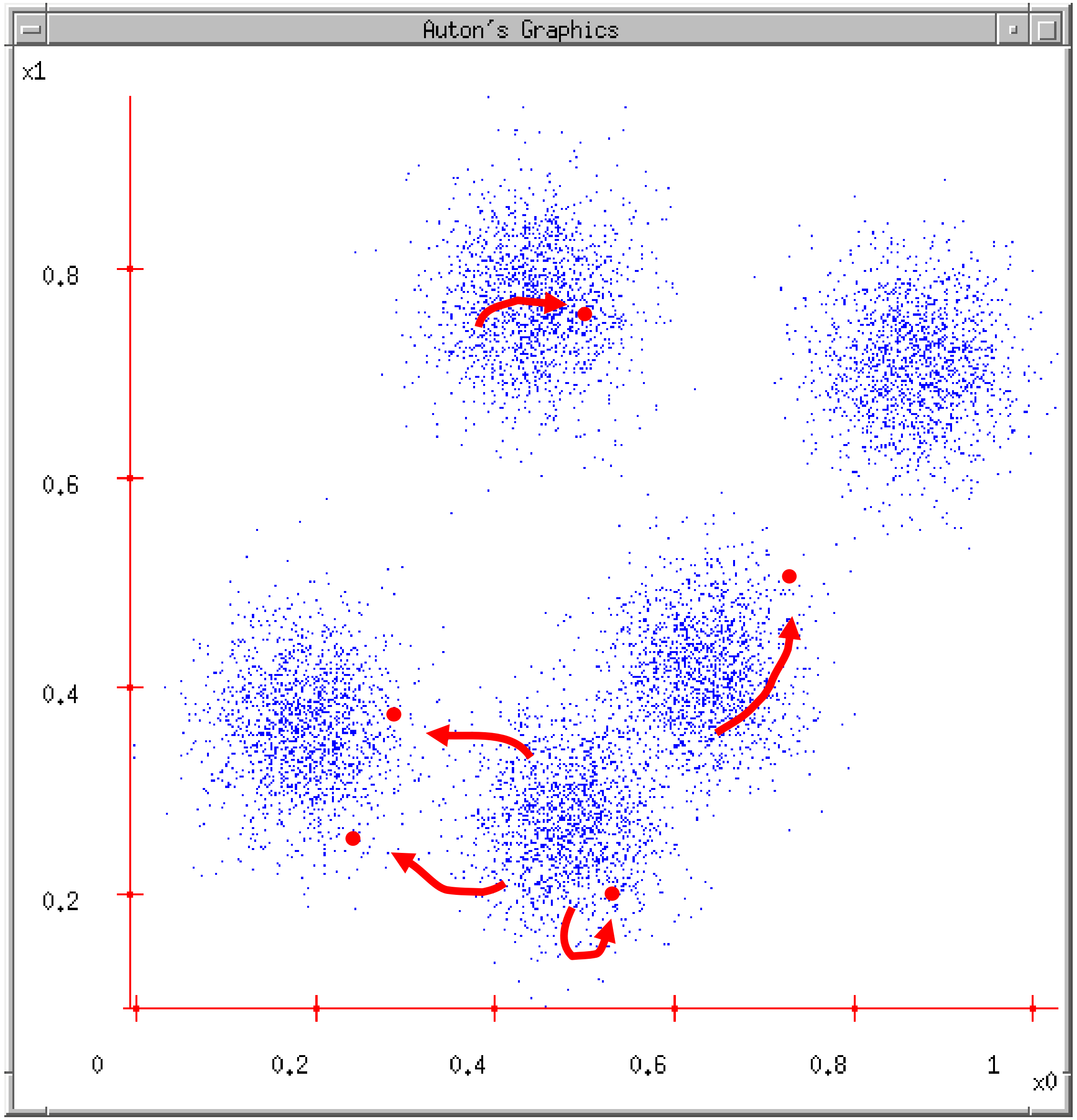
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster Center locations
3. Each data point finds out which Center it's closest to. (Thus each Center "owns" a set of data points)

K-means Demo



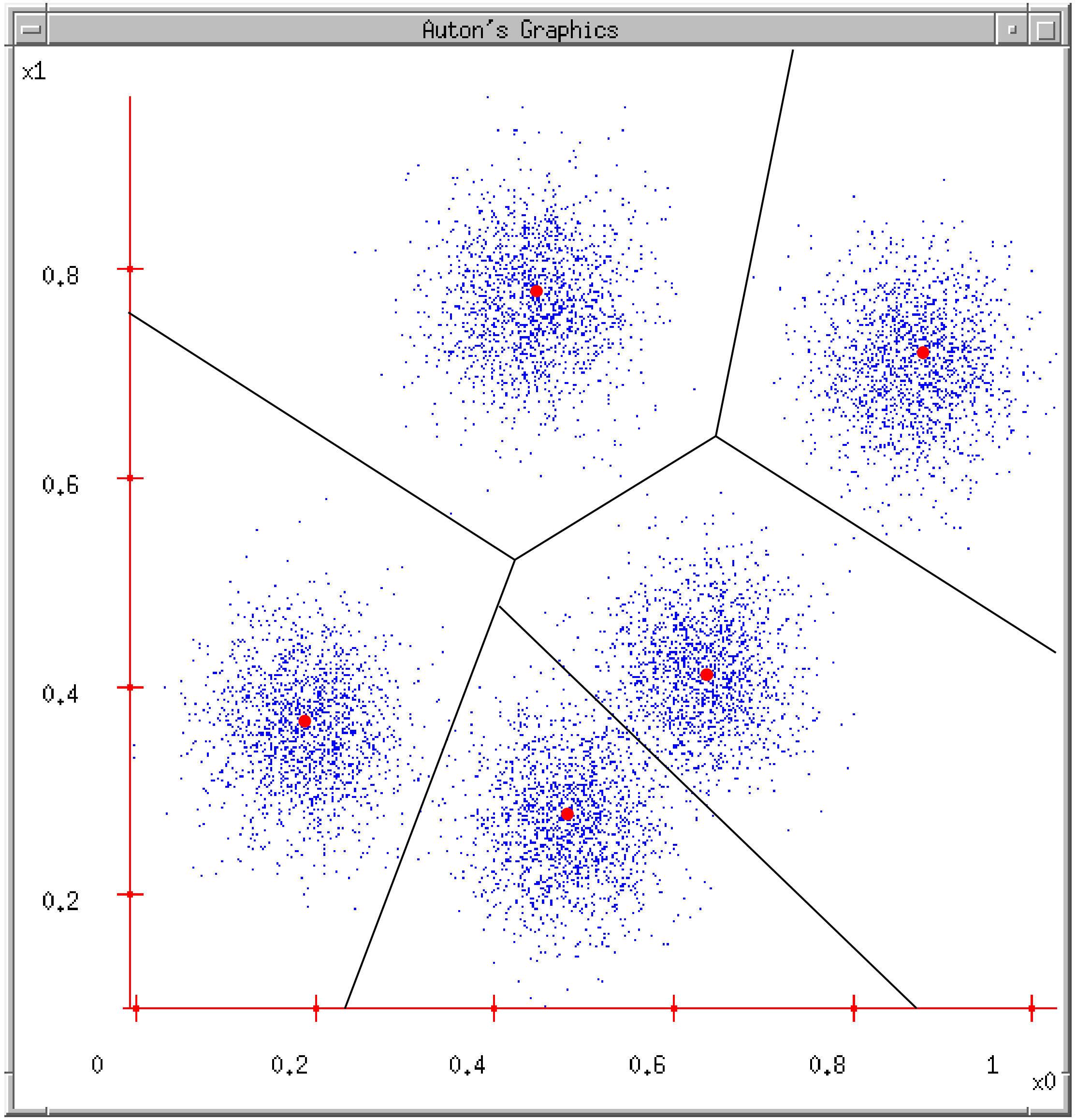
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each Center "owns" a set of data points)
4. Each centre finds the centroid of the points it owns

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

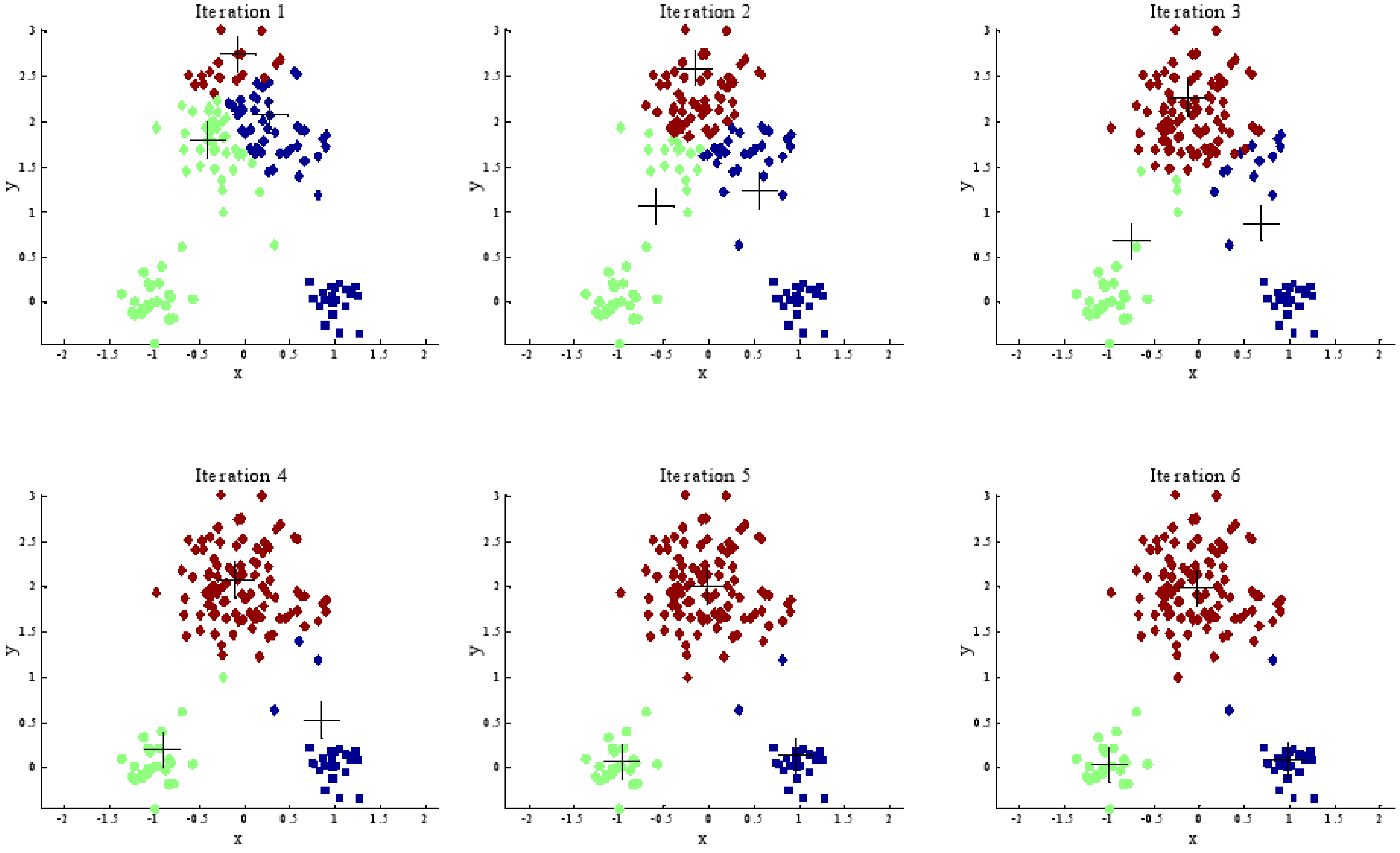
K-means Demo



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!

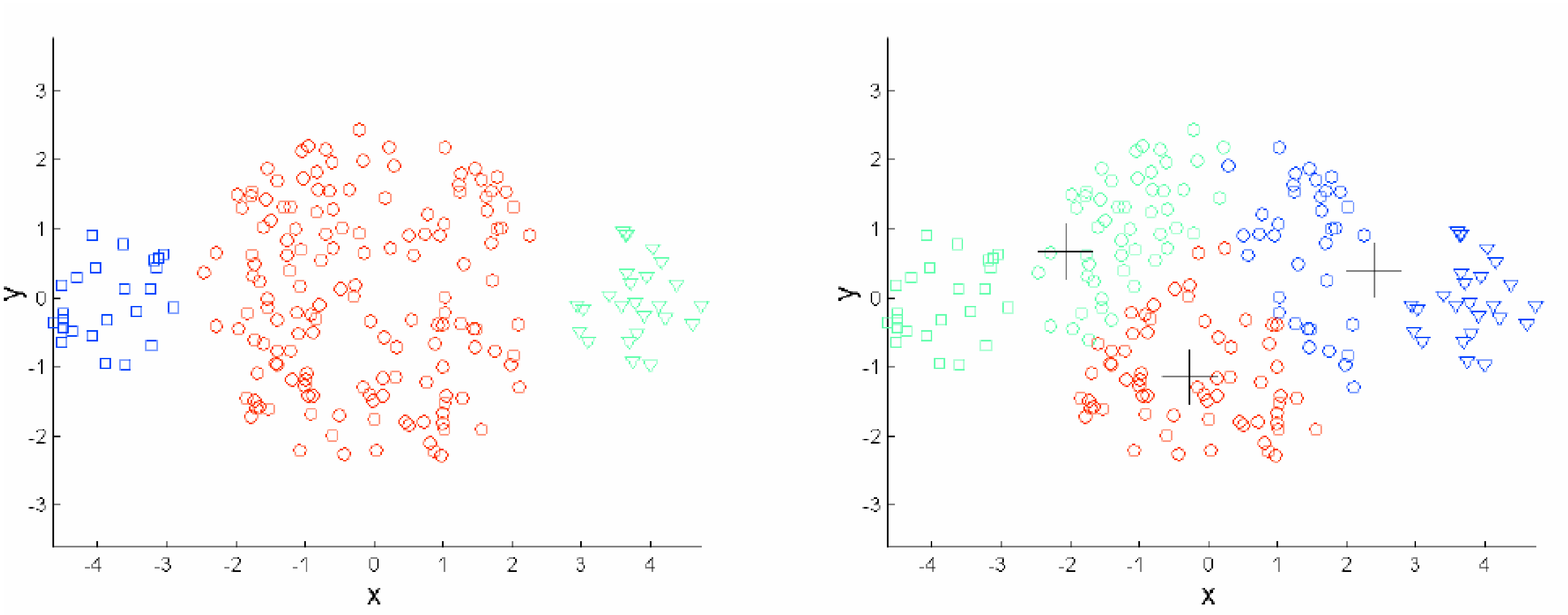
K-Means limits

Importance of choosing initial centroids



K-Means limits

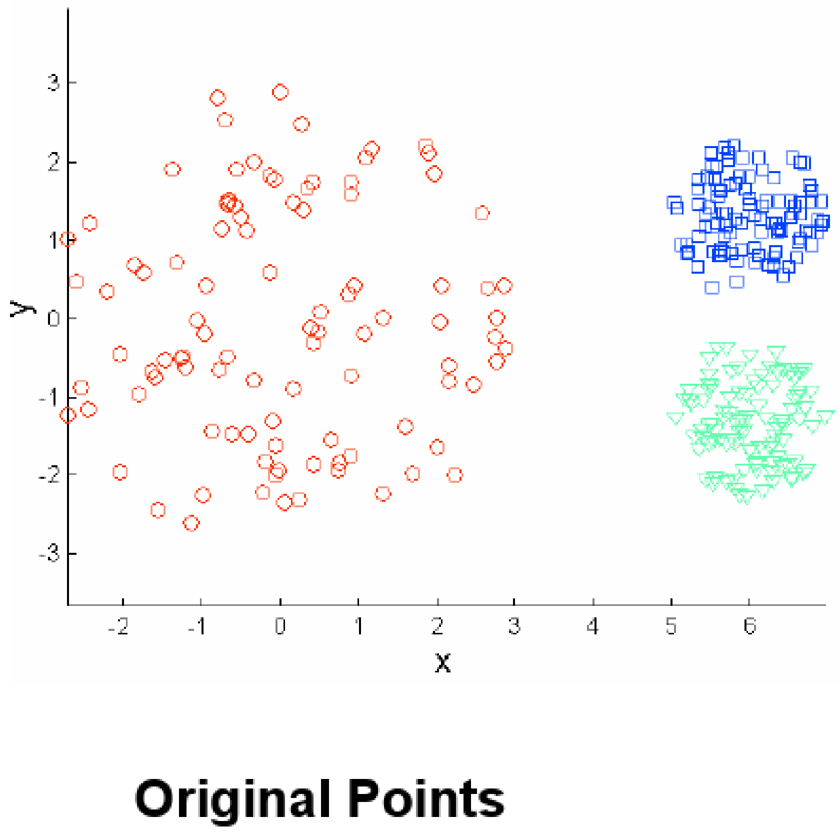
Differing sizes

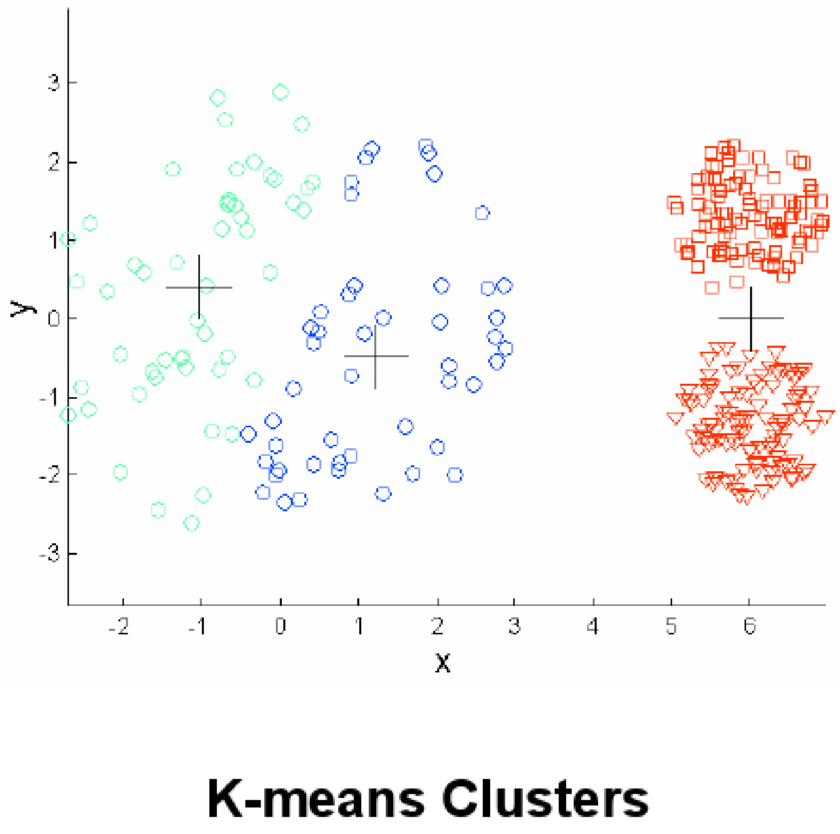


Original Points **K-means Clusters**

K-Means limits

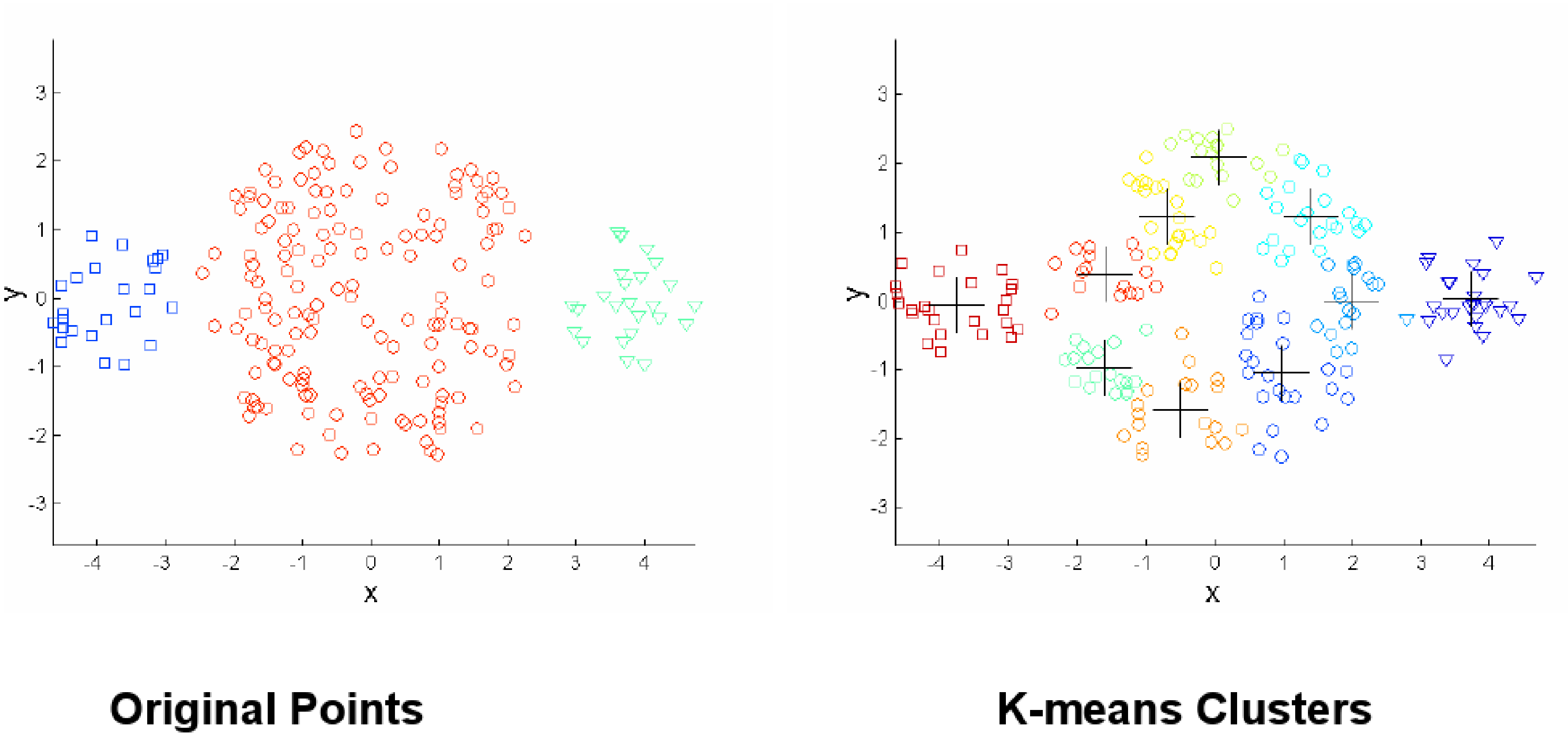
Differing density





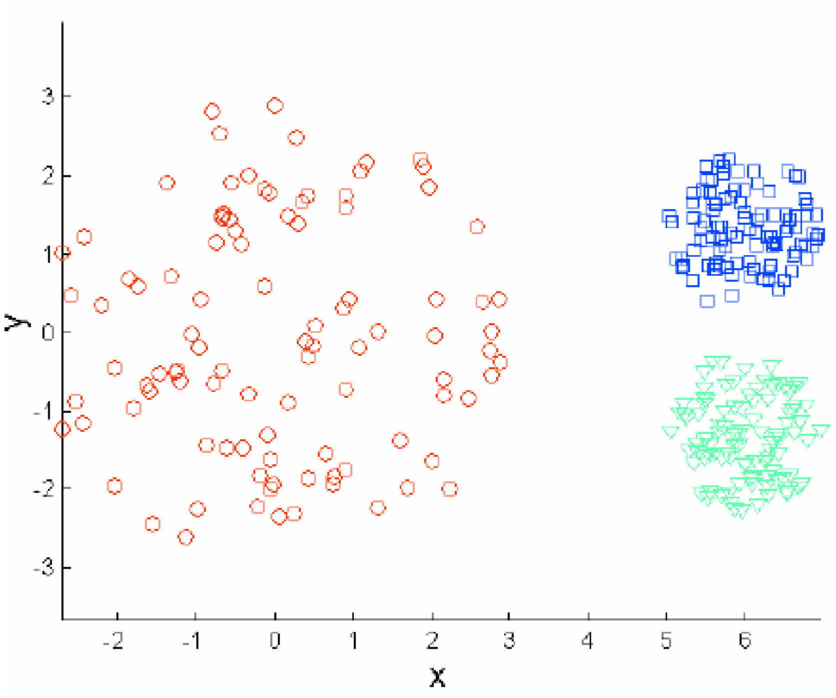
K-Means: higher K

What if we tried to increase K to solve K-Means problems?

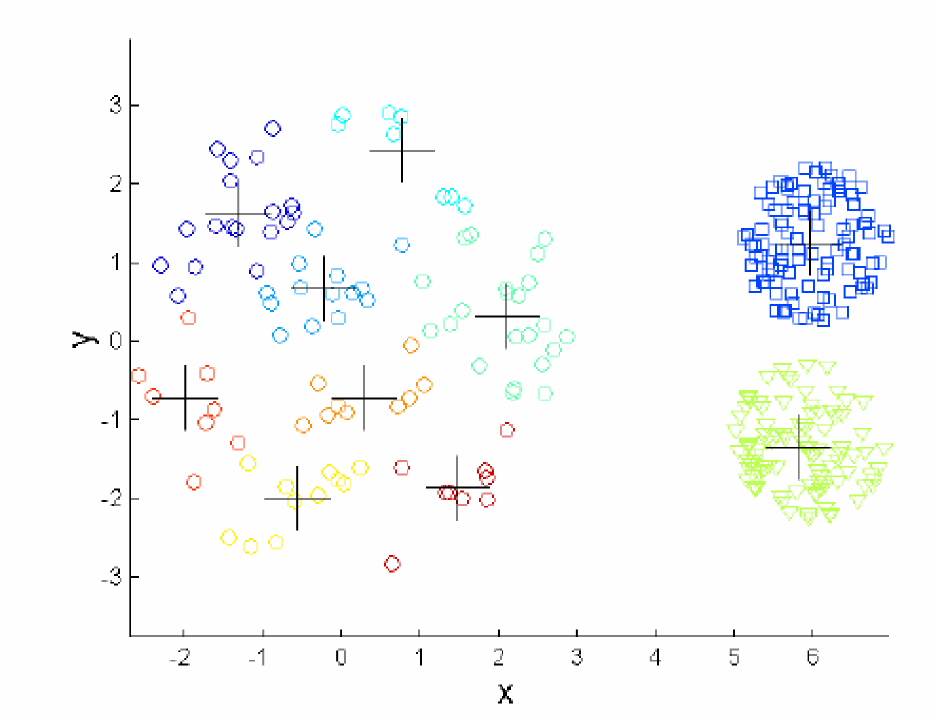


K-Means: higher K

What if we tried to increase K to solve K-Means problems?



Original Points

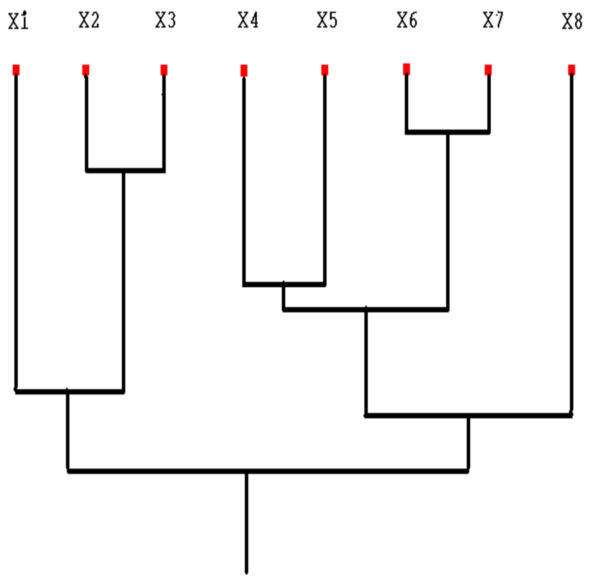


K-means Clusters

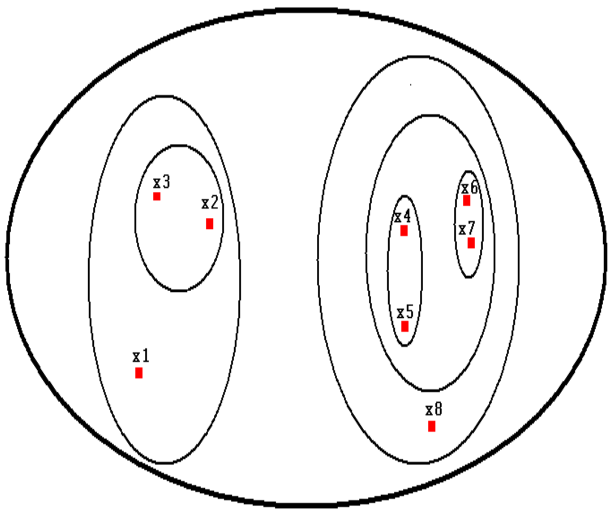
K-Means: Summary

- Advantages:
  - Simple, understandable
  - Relatively efficient: $O(tkn)$, where n is #objects, k is #clusters, and t is #iterations ($k, t \ll n$)
  - Often terminates at a local optimum
- Disadvantages:
  - Works only when mean is defined (what about categorical data?)
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data (too sensible to outliers)
  - Not suitable to discover clusters with non-convex shapes
  - Results depend on the metric used to measure distances and on the value of k
- Suggestions
  - Choose a way to initialize means (i.e. randomly choose k samples)
  - Start with *distant* means, run many times with different starting points
  - Use another algorithm ;-)

Hierarchical Clustering



Dendrogram



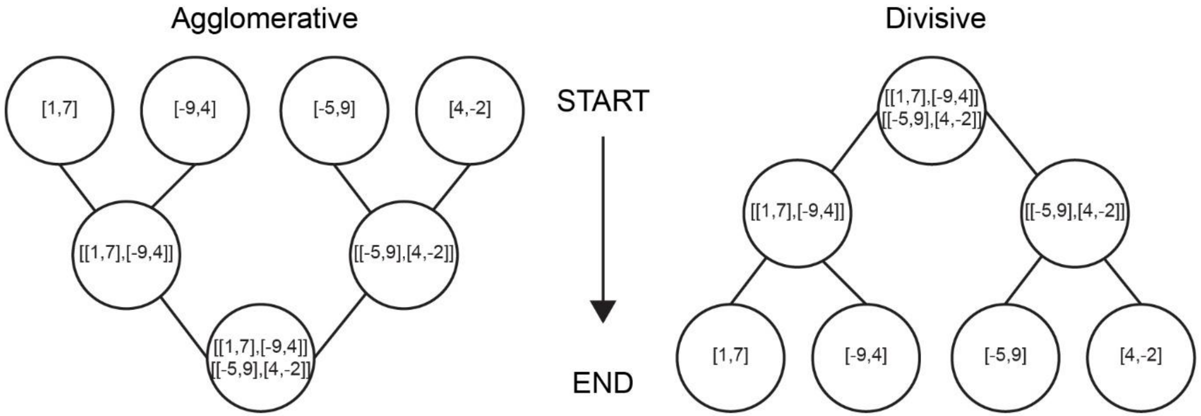
Venn Diagram of Clustered Data

Hierarchical Clustering (Cont.)

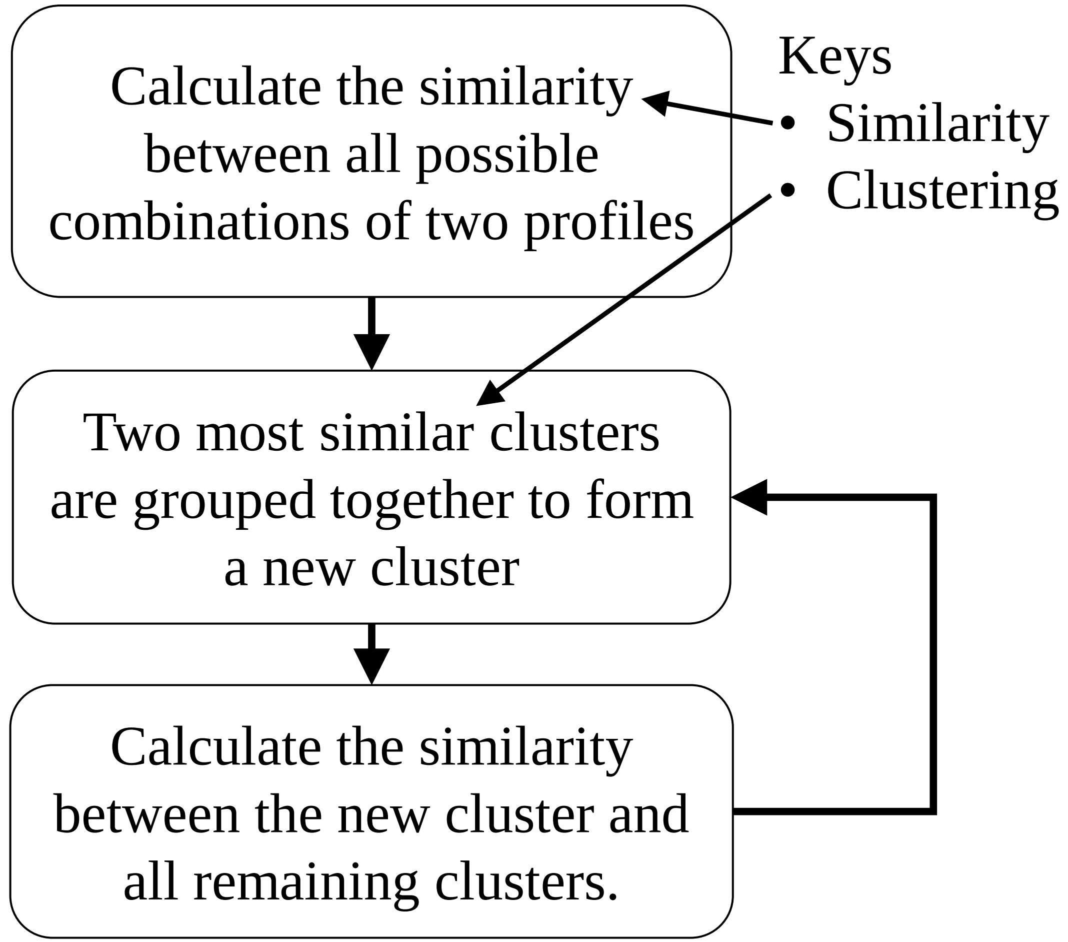
Multilevel clustering:
level 1 has *n* clusters → level *n* has one cluster.

Agglomerative HC:
starts with singleton and merge clusters.

Divisive HC:
starts with one sample and split clusters.

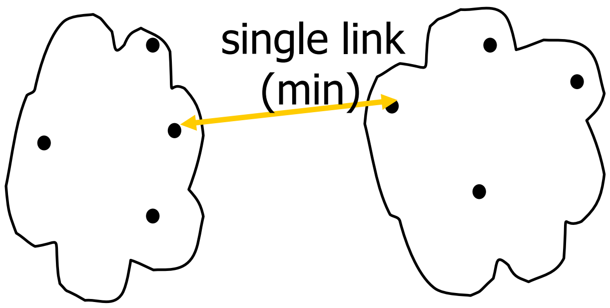


Hierarchical Clustering



Cluster Distance Measures

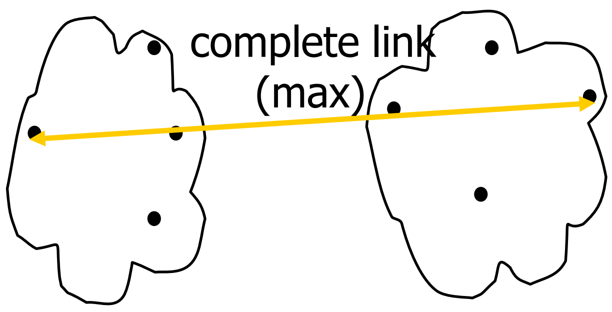
Single link: smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$



Cluster Distance Measures

Complete link: largest distance between an element in one cluster and an element in the other, i.e.,

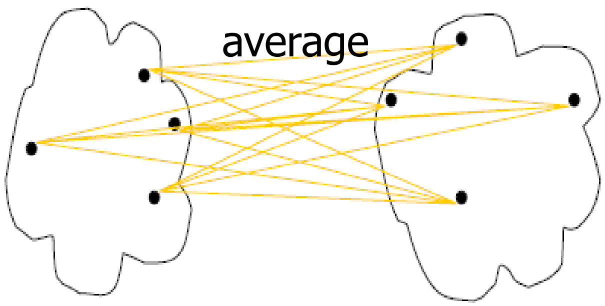
$d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$



Cluster Distance Measures

Average: avg distance between elements in one cluster and elements in the other, i.e.,

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



d(C, C)=0

Cluster Distance Measures

Example: Given a data set of five objects characterized by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}.

1. Calculate the distance matrix.
2. 2. Calculate three cluster distances between C1 and C2.

	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

	a	b	c	d	e
Feature	1	2	4	5	6

Single link

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

Cluster Distance Measures

Example: Given a data set of five objects characterized by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}.

1. Calculate the distance matrix.
2. 2. Calculate three cluster distances between C1 and C2.

	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

	a	b	c	d	e
Feature	1	2	4	5	6

Complete link

$$\begin{aligned}&\mathrm{dist}(\mathrm{C}_1, \mathrm{C}_2)\\&= \max\{\, d(\mathrm{a,c}), d(\mathrm{a,d}), d(\mathrm{a,e}), d(\mathrm{b,c}), d(\mathrm{b,d}), d(\mathrm{b,e})\}\\&= \max\{3, 4, 5, 2, 3, 4\} = 5\end{aligned}$$

Cluster Distance Measures

Example: Given a data set of five objects characterised by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}.

1. Calculate the distance matrix.
2. 2. Calculate three cluster distances between C1 and C2.

	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

	a	b	c	d	e
Feature	1	2	4	5	6

Average

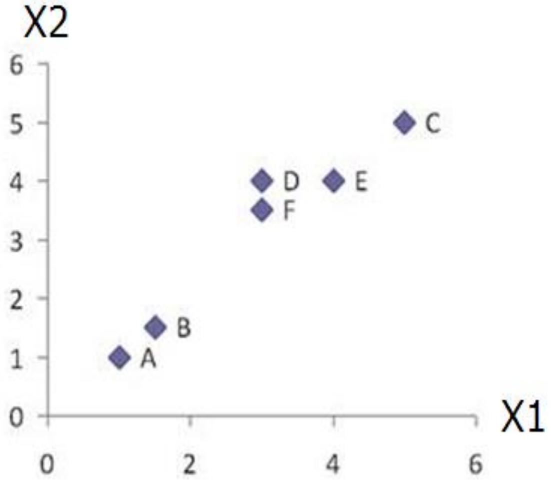
$$\mathrm{dist}(C_1, C_2) = \frac{d(\mathrm{a,c}) + d(\mathrm{a,d}) + d(\mathrm{a,e}) + d(\mathrm{b,c}) + d(\mathrm{b,d}) + d(\mathrm{b,e})}{6}$$

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

Example

Problem: clustering analysis with agglomerative algorithm

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



$$d_{AB} = ((1-1.5)^2 + (1-1.5)^2)^{\frac{1}{2}} = \sqrt{\frac{1}{2}} = 0.707$$

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

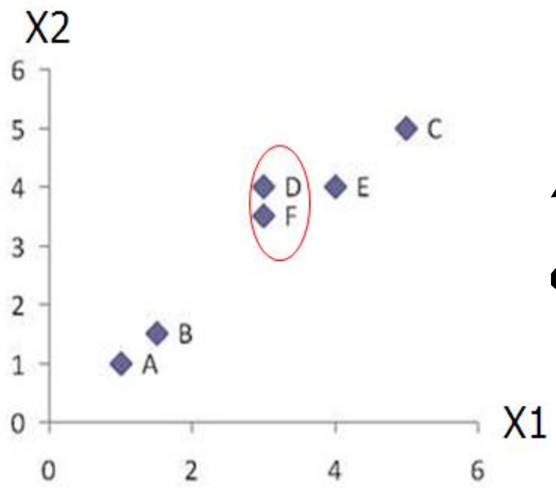
Euclidean distance

Dist	A	B	C	D	E	F
A	0.0	0.71	5.66	3.61	4.24	3.2
B		0.0	4.95	2.92	3.54	2.50
C			0.0	2.24	1.41	2.50
D				0.0	1.00	0.50
E					0.0	1.12
F						0.0

distance matrix

Example

Merge two closest clusters (iteration 1)



Dist	A	B	C	D	E	F
A	0.0	0.71	5.66	3.61	4.24	3.2
B		0.0	4.95	2.92	3.54	2.50
C			0.0	2.24	1.41	2.50
D				0.0	1.00	0.50
E					0.0	1.12
F						0.0

Dist	A	B	C	D,F	E
A	0.0	0.71	5.66	?	4.24
B		0.0	4.95	?	3.54
C			0.0	?	1.41
D,F				0.0	?
E					0.0

Example

Update distance matrix (iteration 1)

Dist	A	B	C	D	E	F
A	0.0	0.71	5.66	3.61	4.24	3.2
B		0.0	4.95	2.92	3.54	2.50
C			0.0	2.24	1.41	2.50
D				0.0	1.00	0.50
E					0.0	1.12
F						0.0

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

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

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

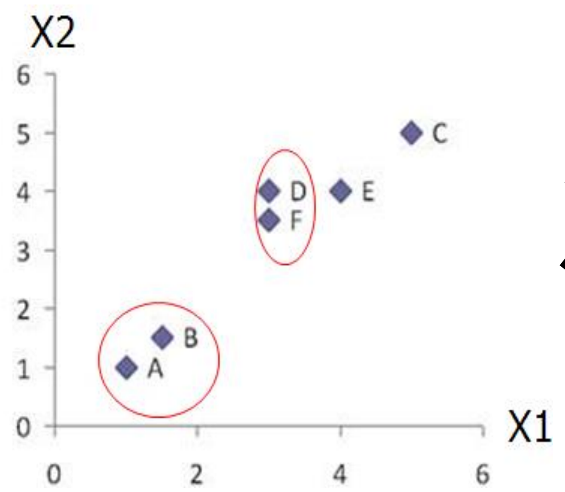
$$d_{(D,F)\to E} = \min(d_{DE}, d_{FE}) = \min(1.00, 1.12) = 1.00$$

Dis	A	B	C	D,F	E
A	0.0	0.71	5.66	?	4.24
B		0.0	4.95	?	3.54
C			0.0	?	1.41
D,F				0.0	?
E					0.0

Dist	A	B	C	D,F	E
A	0.0	0.71	5.66	3.20	4.24
B		0.0	4.95	2.50	3.54
C			0.0	2.24	1.41
D,F				0.0	1.00
E					0.0

Example

Merge two closest clusters (iteration 2)



Dist	A	B	C	D,F	E
A	0.0	**0.71**	5.66	3.20	4.24
B		0.0	4.95	2.50	3.54
C			0.0	2.24	1.41
D,F				0.0	1.00
E					0.0

Dist	A,B	C	D,F	E
A,B	0.0	?	?	?
C		0.0	2.24	1.41
D,F			0.0	1.00
E				0.0

Example

Update distance matrix (iteration 2)

Dist	A	B	C	D,F	E
A	0.0	0.71	5.66	3.20	4.24
B		0.0	4.95	2.50	3.54
C			0.0	2.24	1.41
D,F				0.0	1.00
E					0.0

$$d_{(A,B)\to C} = \min(d_{AC}, d_{BC}) = \min(\,5.66{,}4.95) = 4.95$$

$$d_{(A,B)\to (D,F)} = \min(d_{AD}, d_{AF}, d_{BD}, d_{BF})$$
$$= \min(\,3.61{,}2.92{,}3.20{,}2.50) = 2.50$$

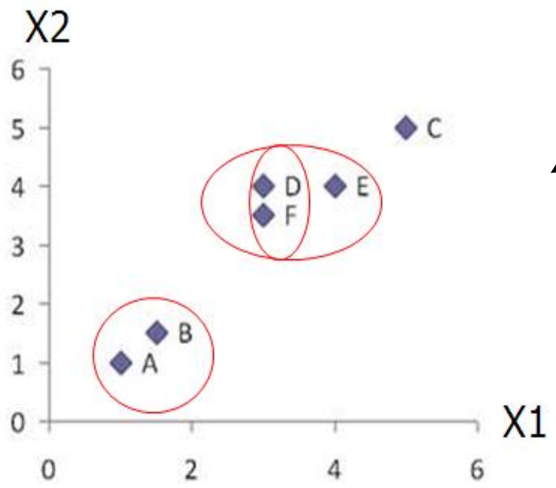
$$d_{(A,B)\to E} = \min(d_{AE}, d_{BE}) = \min(\,4.24{,}3.54) = 3.54$$

Dist	A,B	C	D,F	E
A,B	0.0	?	?	?
C		0.0	2.24	1.41
D,F			0.0	1.00
E				0.0

Dist	A,B	C	D,F	E
A,B	0.0	4.95	2.50	3.54
C		0.0	2.24	1.41
D,F			0.0	1.00
E				0.0

Example

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

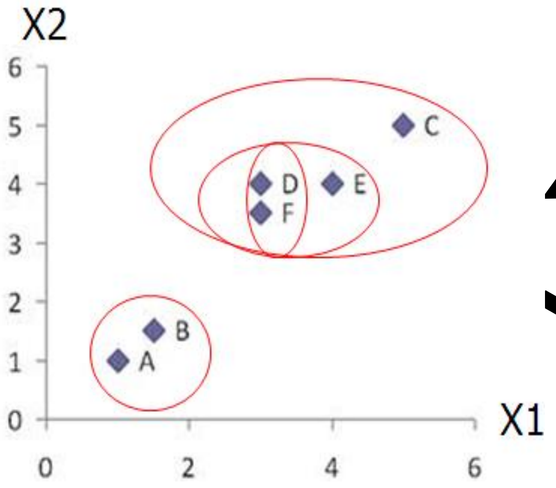


Dist	A,B	C	D,F	E
A,B	0.0	4.95	2.50	3.54
C		0.0	2.24	1.41
D,F			0.0	1.00
E				0.0

Dist	A,B	C	(D,F),E
A,B	0.0	4.95	2.50
C		0.0	1.41
(D,F),E			0.0

Example

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



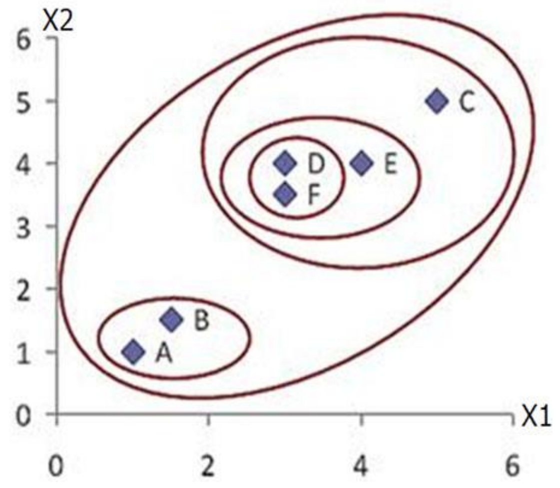
Dist	A,B	C	(D,F),E
A,B	0.0	4.95	2.50
C		0.0	1.41
(D,F),E			0.0

Dist	A,B	((D,F),E),C
A,B	0.0	2.50
((D,F),E),C		0.0

Example

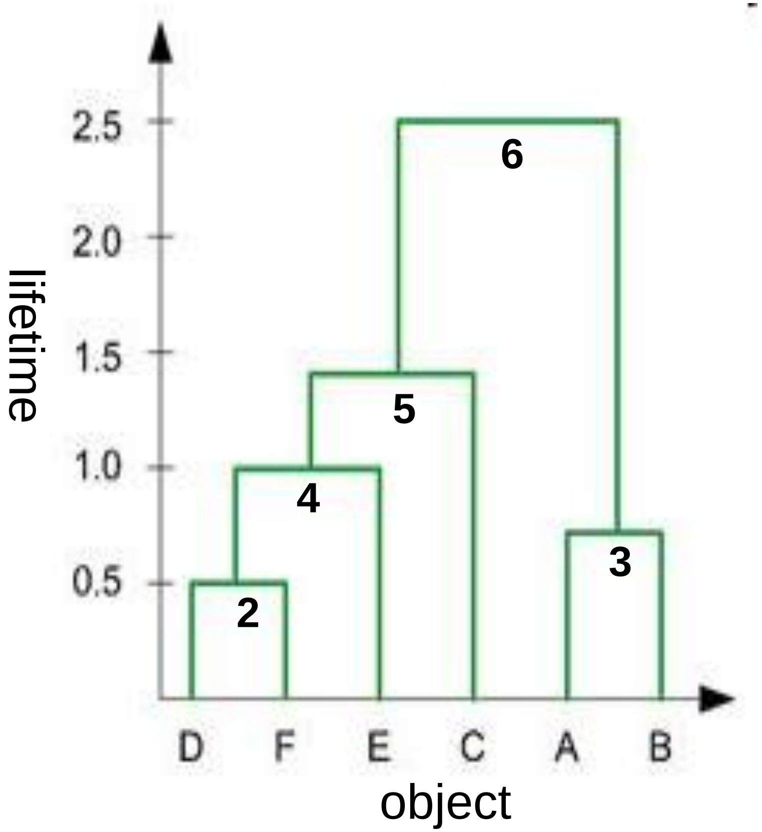
Final result (meeting termination condition)

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



Example

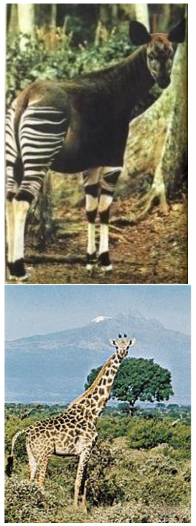
Dendrogram tree representation

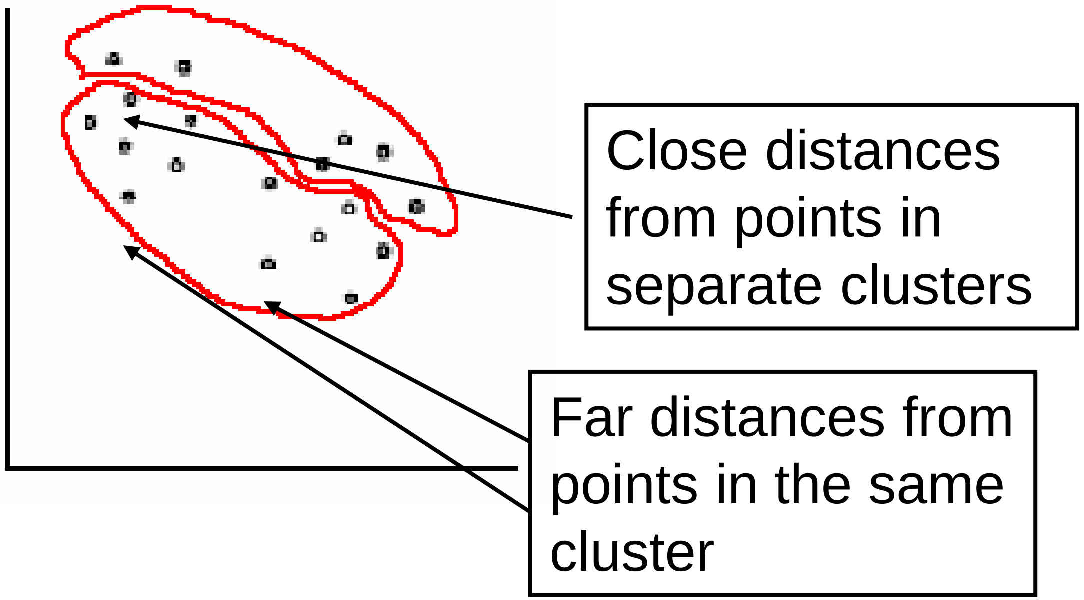


1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge clusters D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
5. We merge clusters ((D, F), E) and C into (((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

Bad Clustering

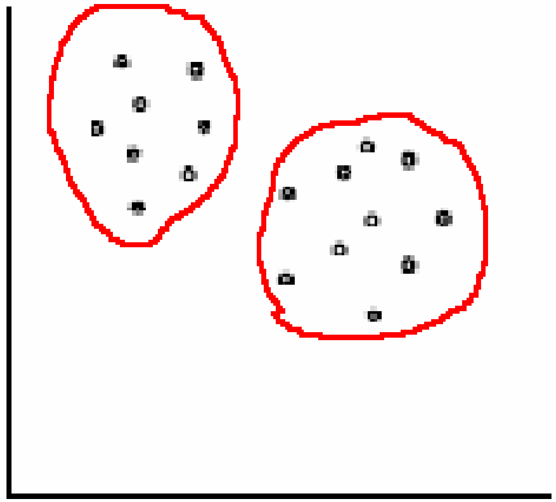
This clustering violates both
Homogeneity and Separation principles





Good Clustering

This clustering satisfies both
Homogeneity and Separation principles



- The end