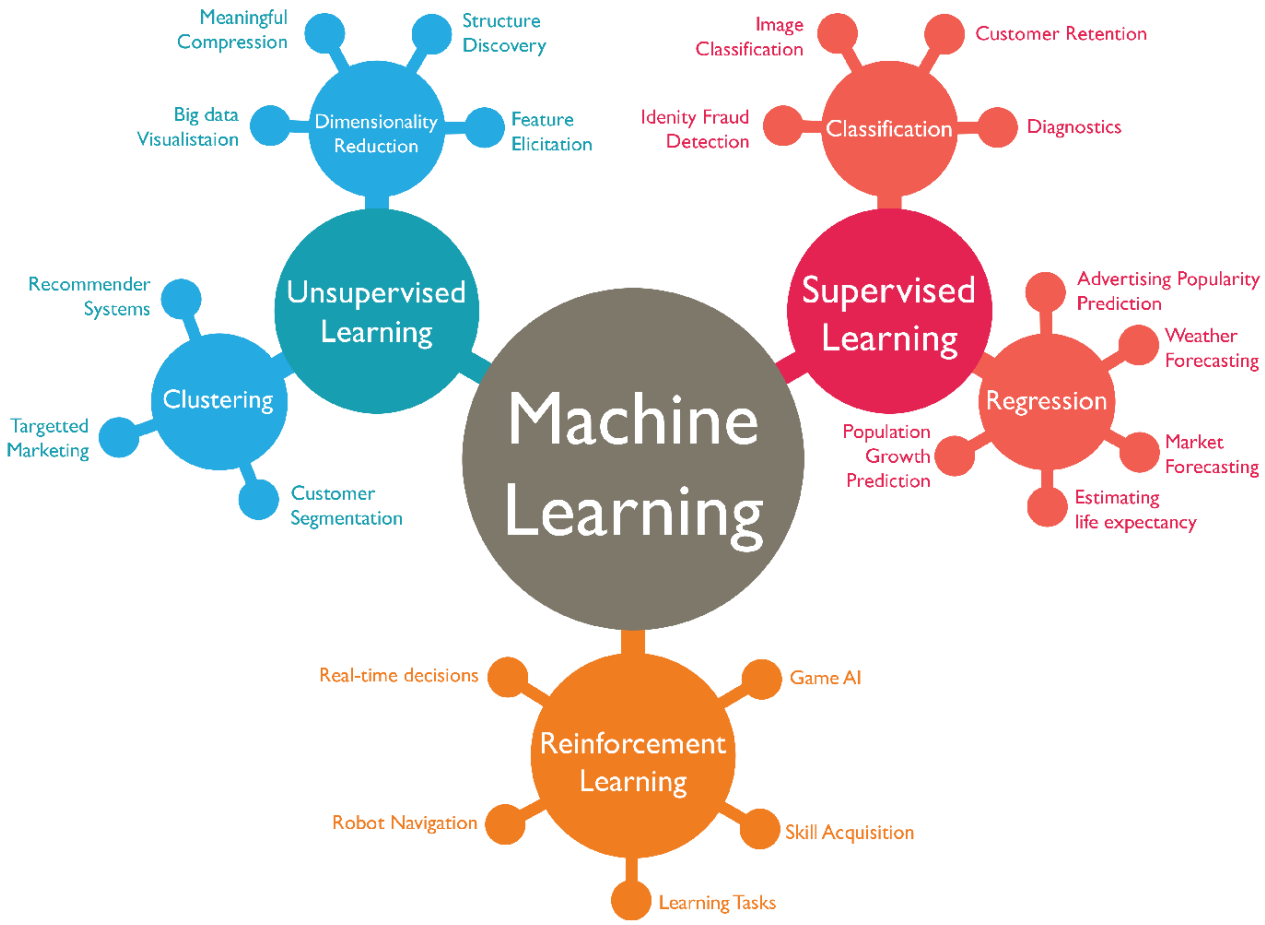
ENCS5341
Machine Learning and Data Science

Clustering

Introduction

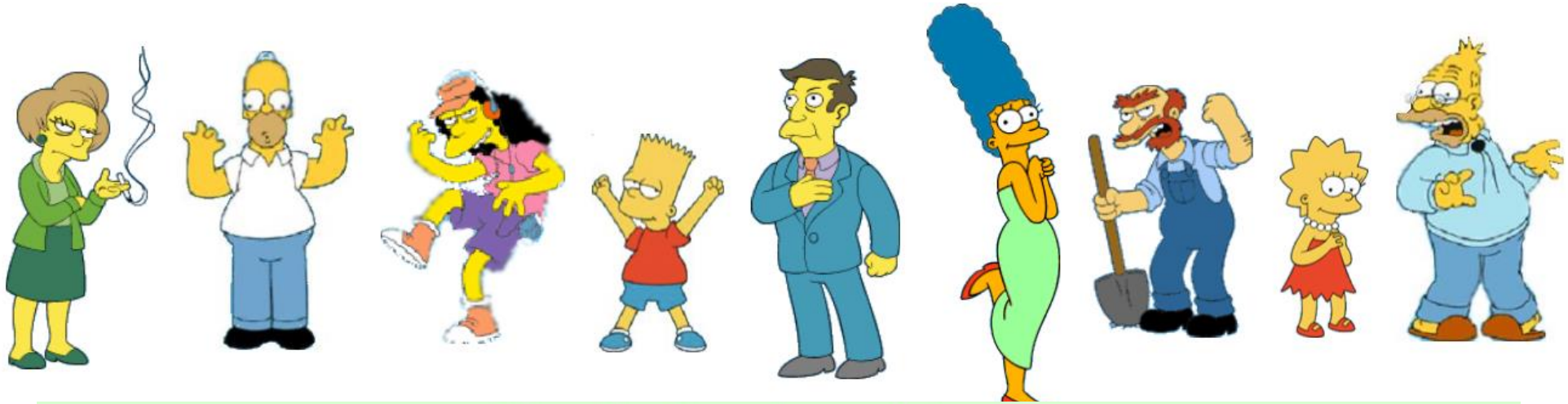
- So far in this course, we only considered supervised learning techniques.
- Supervised learning: All training data samples are annotated with its corresponding label:
  - If the label is continuous → regression.
  - If the label is discrete → classification.
- This lecture is about clustering, which is an example of unsupervised learning tasks.
- Unsupervised learning: we only have the input training data without any labels.



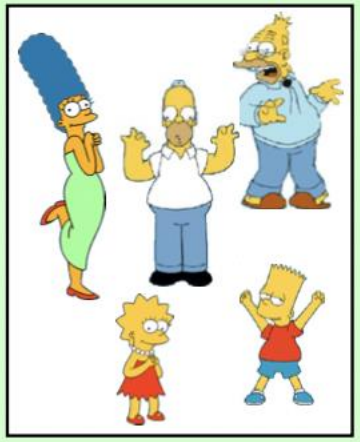
What is Clustering?

- Clustering is the process of organizing a set of physical or abstract objects into classes (called Clusters), such that there is
  - high intra-class similarity.
  - low inter-class similarity
- More informally: finding natural groupings among objects.

What is a natural grouping among these objects?

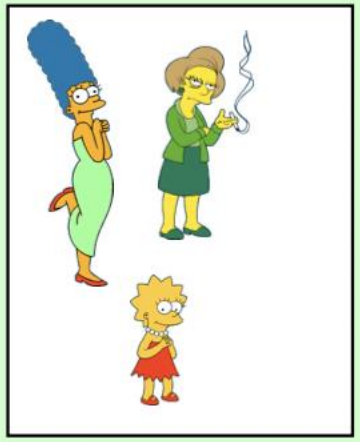


Clustering is subjective



Simpson Family

School Employees



Females

Males

What is similarity?

- The quality or state of being similar; likeness; resemblance; as, a similarity of features.

Webster's Dictionary

- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

- **Definition**: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1, O_2)$

Properties of a distance measure

- Symmetry

  $D(A, B) = D(B, A)$

- Constancy of Self-Similarity

  $D(A, A) = 0$

- Positivity (Separation)

  $D(A, B) \geq 0$

  $D(A, B) = 0$ iff $A = B$

- Triangular Inequality

  $D(A, B) \leq D(A, C) + D(B, C)$

Applications of Clustering

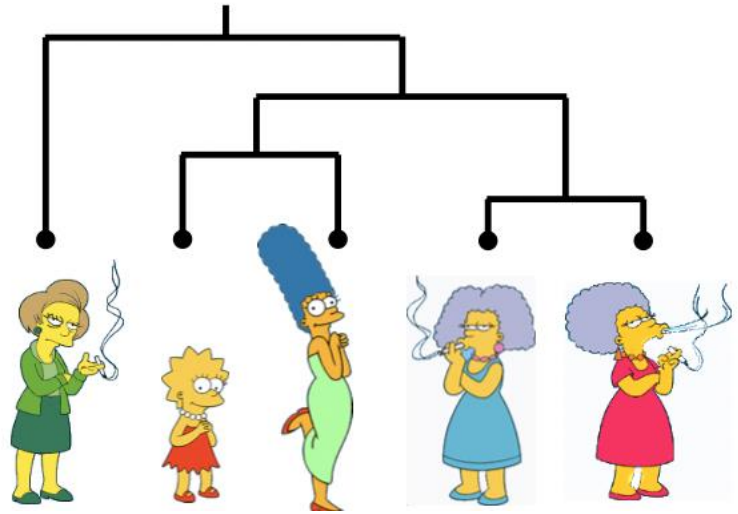
- Business (e.g. customer segmentation).
- Pattern recognition.
- Image segmentation.
- Compression.
- Information retrieval.
- …

Classical Clustering Types

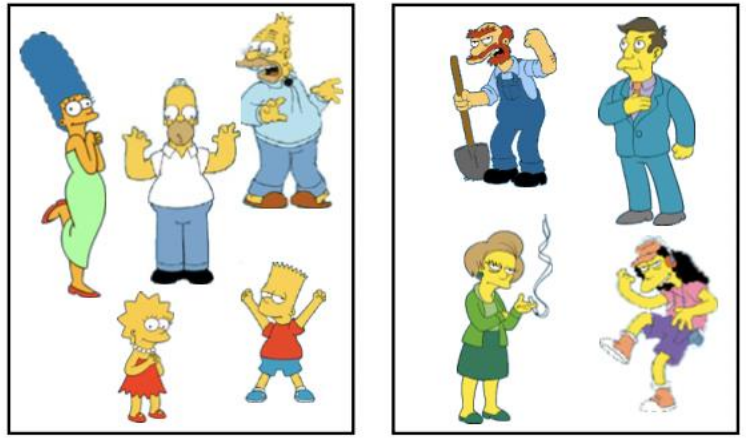
- Hierarchical Clustering: Create a hierarchical decomposition of the set of objects using some criterion.

- Partitional Clustering: Construct various partitions and then evaluate them by some criterion.

Hierarchical



Partitional



In this course

- Partitional Clustering
  - k-means
  - GMM
- Hierarchical Clustering
  - Agglomerative Clustering

K-means

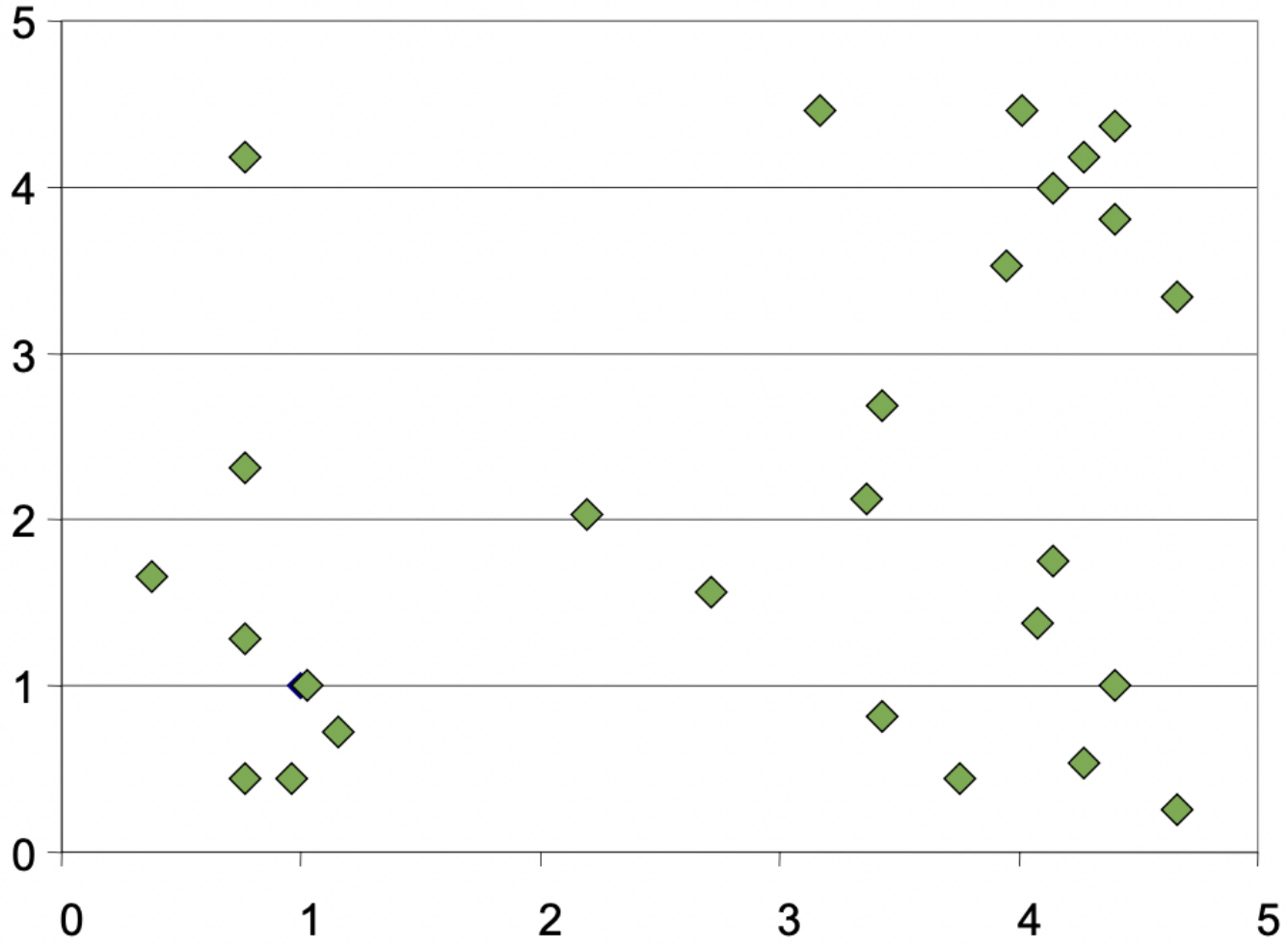
- Given:
  - data set $X = \{x_1,\ldots,x_m\}$
  - squared Euclidean distance as a distance measure
  - number of clusters k

- Goal:
  - Group data points in k clusters, i.e. find an assignment of data points to clusters, as well as a set of cluster centers μ_j , such that the sum of the squares of the distances of each data point to its closest center is minimized.

- Formally: Let r_{ij} be an indicator vector, which is 1 if and only if x_i is assigned to cluster j. Find the values for r_{ij} and μ_i , so as to minimize the objective function

$$J(r,\mu) := \sum_{i=1}^{m}\sum_{j=1}^{k} r_{ij}\|x_i - \mu_j\|^2$$

Data set

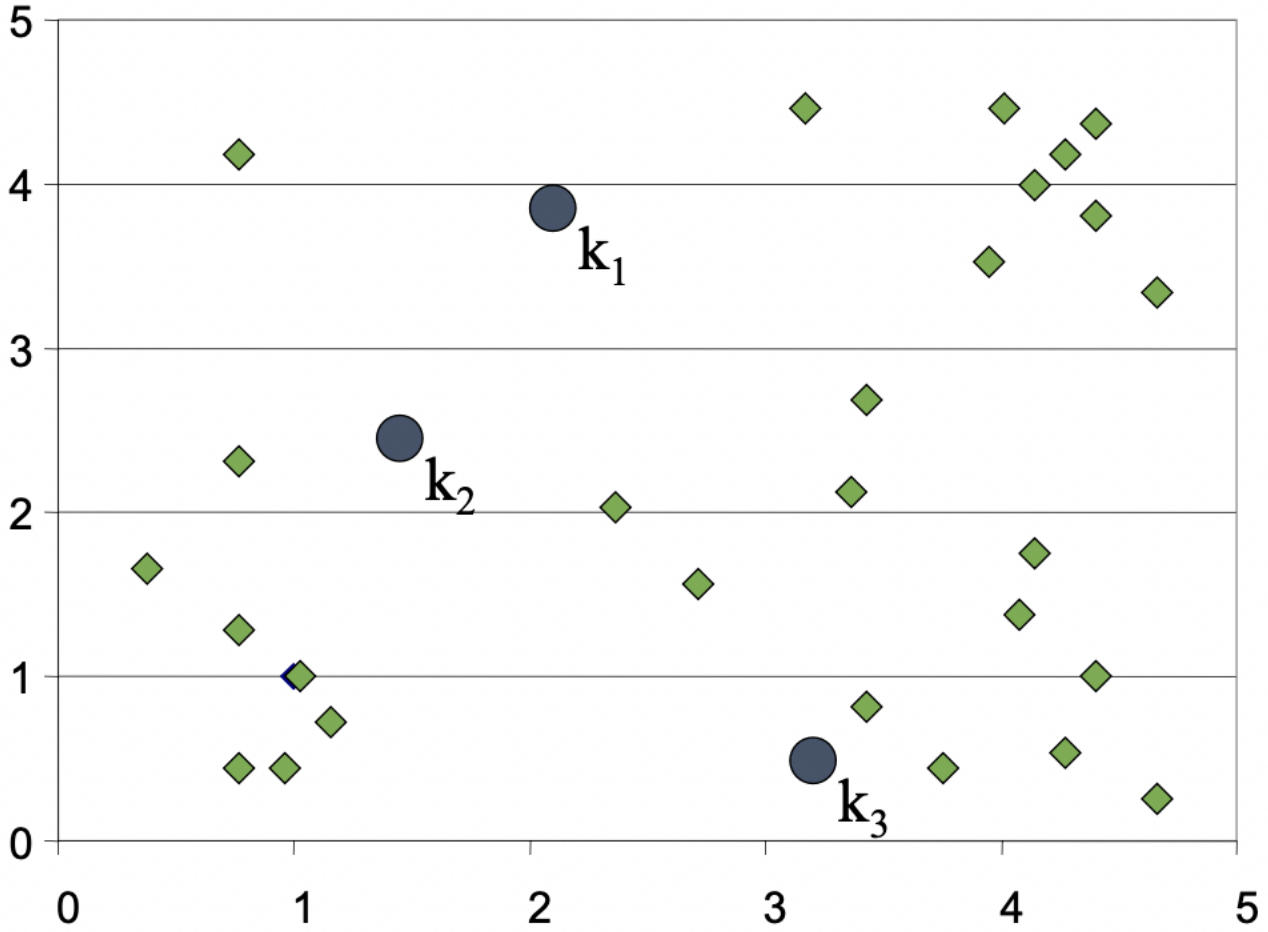


Lloyd's Algorithm for k-means

1. Decide on a value for k.
2. Initialize the k cluster centers.
3. Decide the class memberships of the N objects by assigning them to the nearest cluster center.
4. Re-estimate the k cluster centers as centroids of the clusters resulting from step 3.
5. If none of the N objects changed membership in the last iteration, exit. Otherwise goto 3.

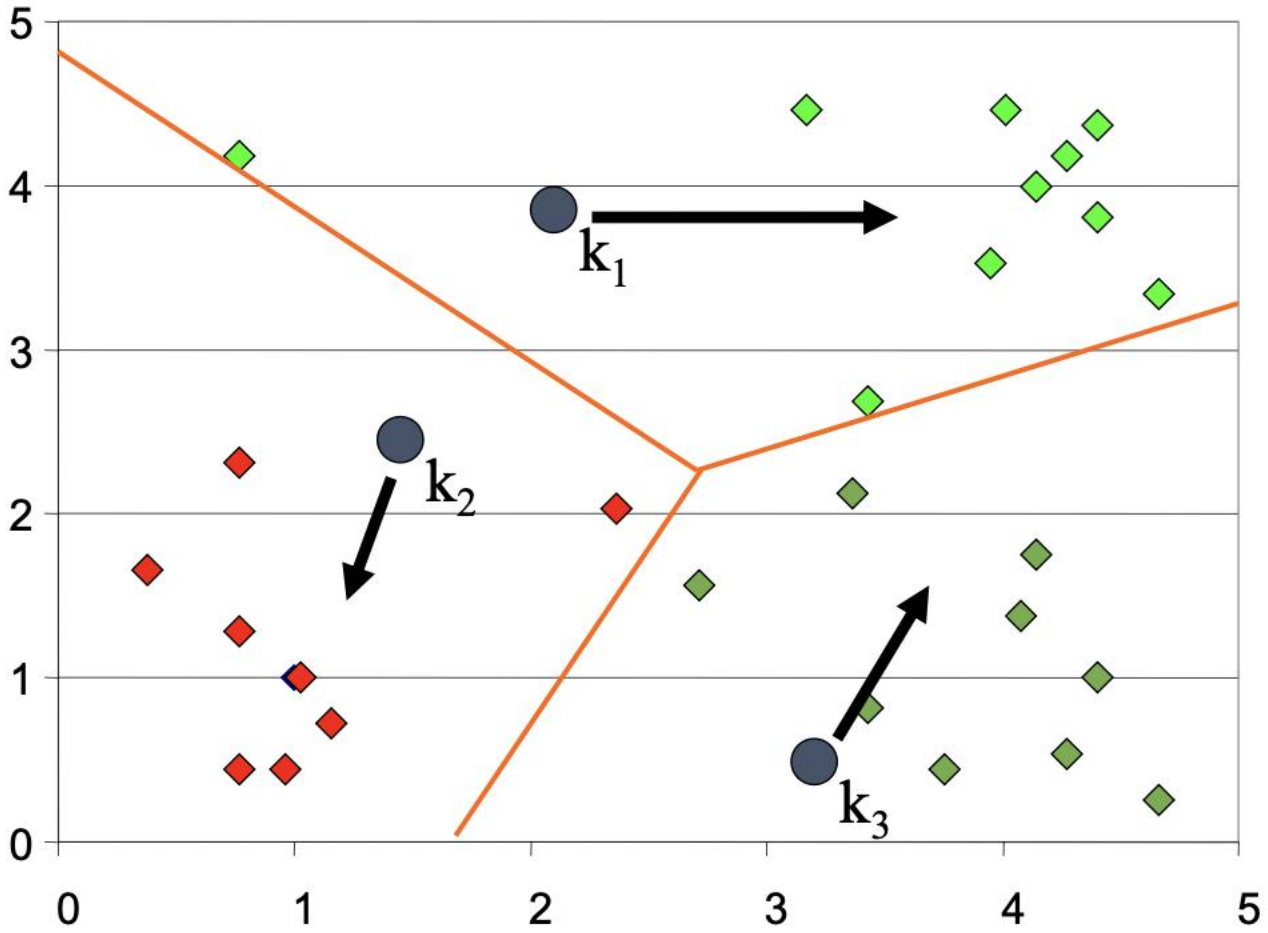
K-means Clustering: Step 1

- Algorithm: k-means, Distance Metric: Euclidean



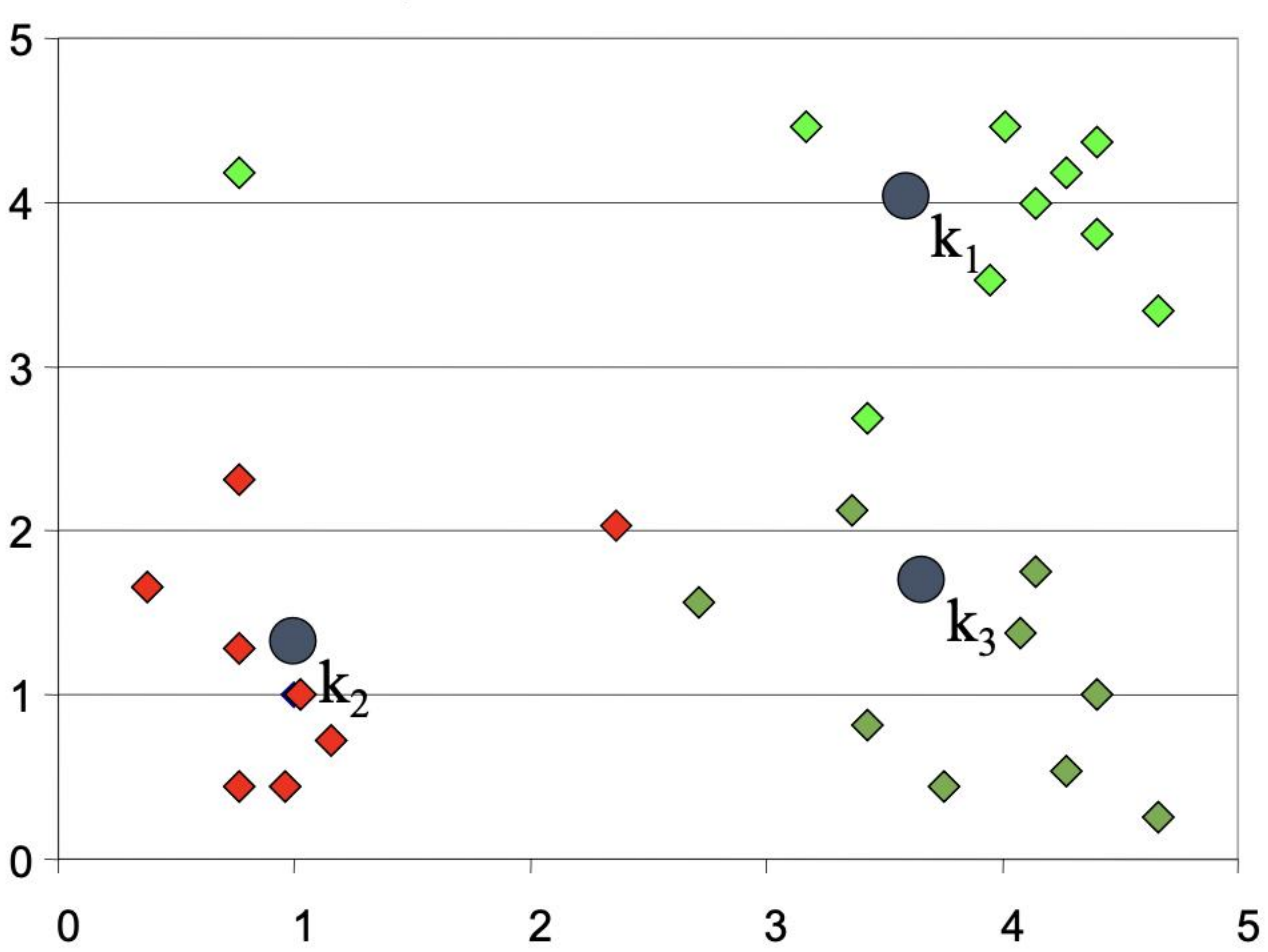
K-means Clustering: Step 2

- Algorithm: k-means, Distance Metric: Euclidean



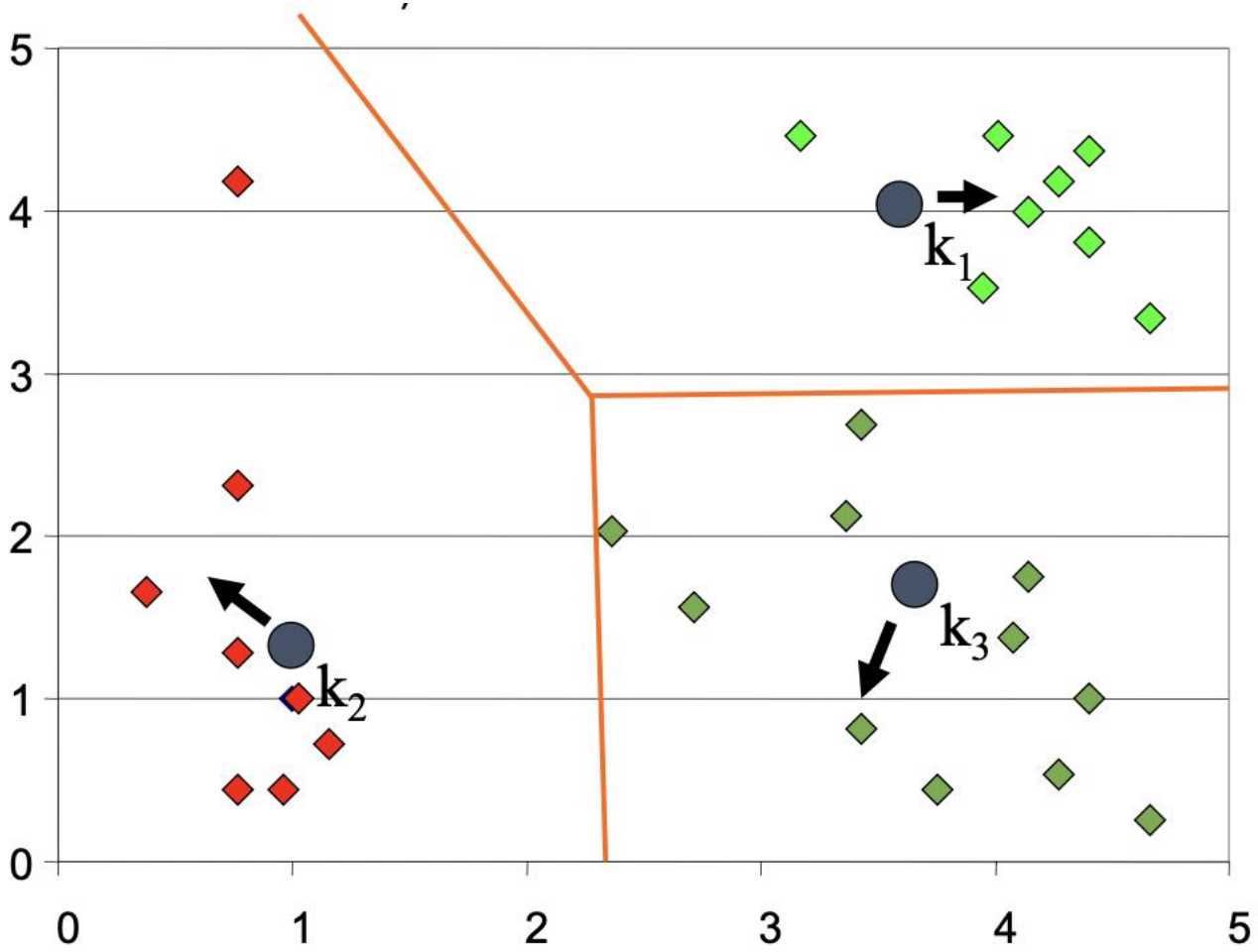
K-means Clustering: Step 3

- Algorithm: k-means, Distance Metric: Euclidean



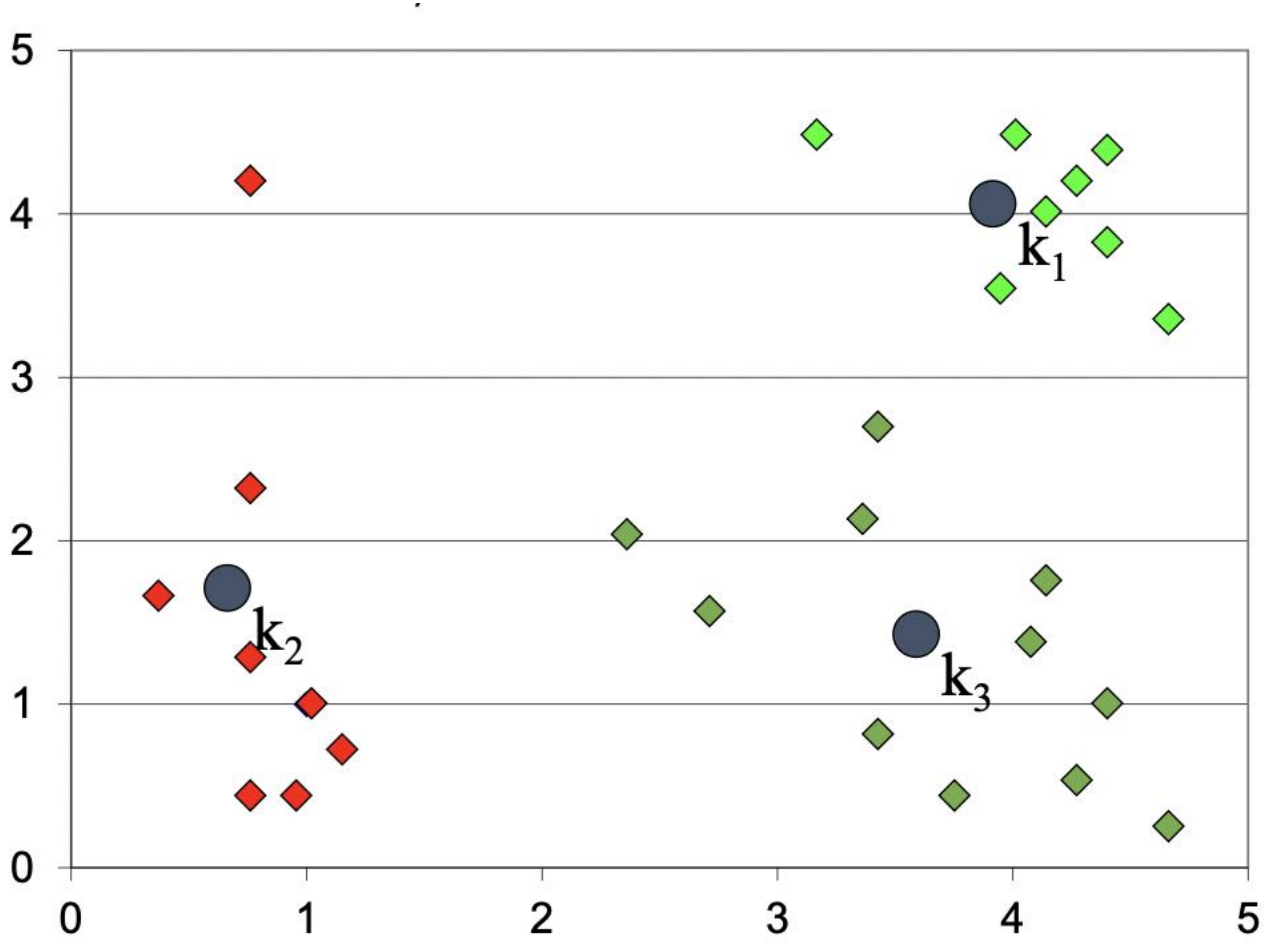
K-means Clustering: Step 4

- Algorithm: k-means, Distance Metric: Euclidean



K-means Clustering: Step 5

- Algorithm: k-means, Distance Metric: Euclidean



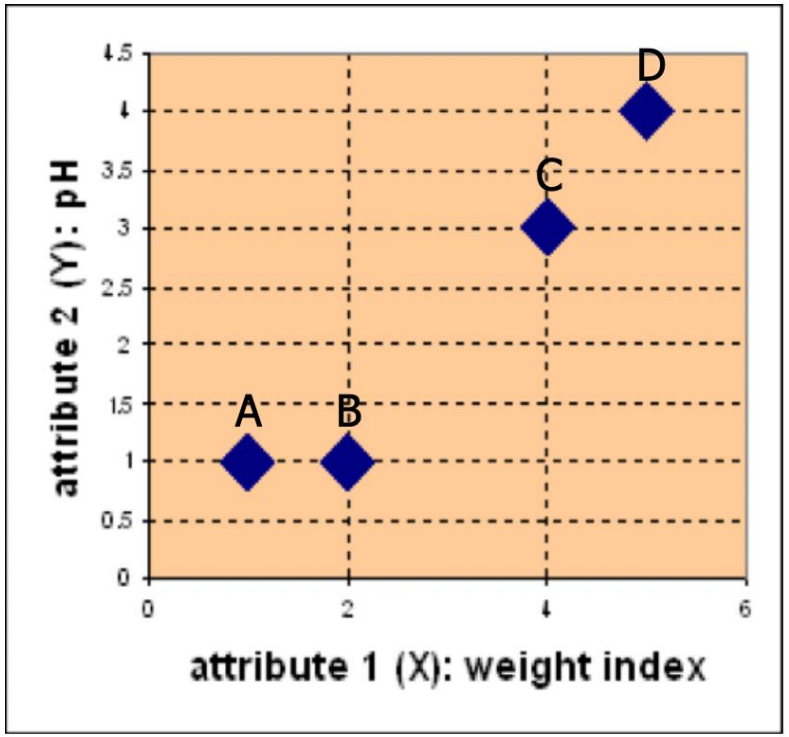
Example: Image Segmentation and Compression

Example

- Problem

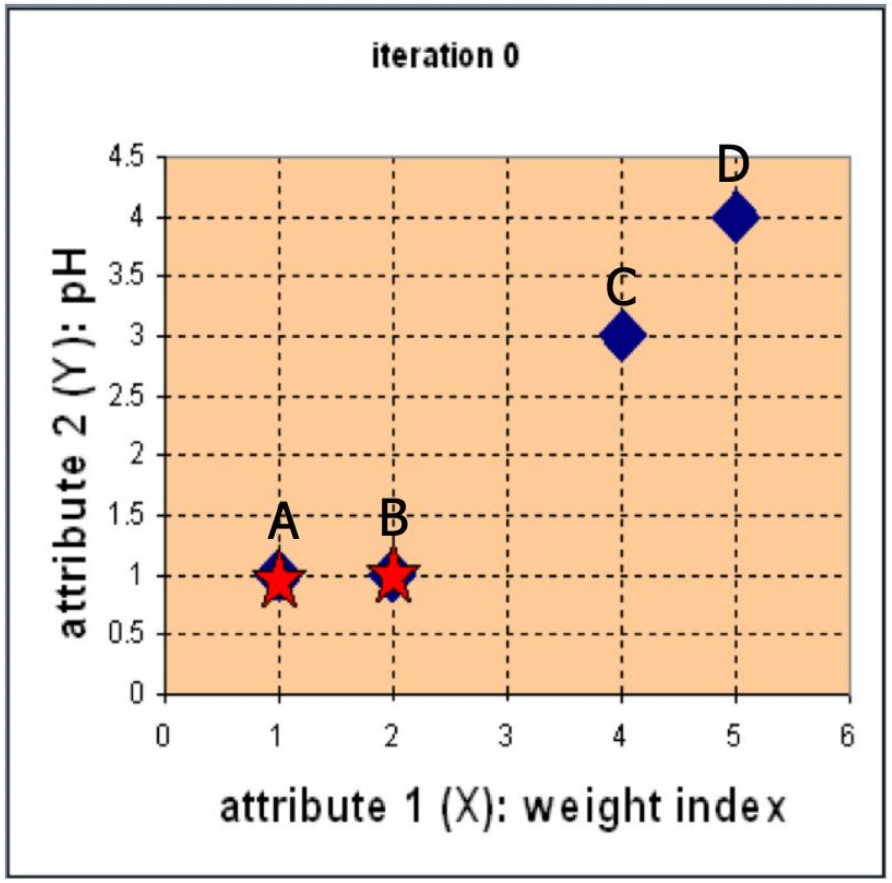
Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into K=2 group of medicine.

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



$$c_1 = A, c_2 = B$$

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

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

Euclidean distance

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

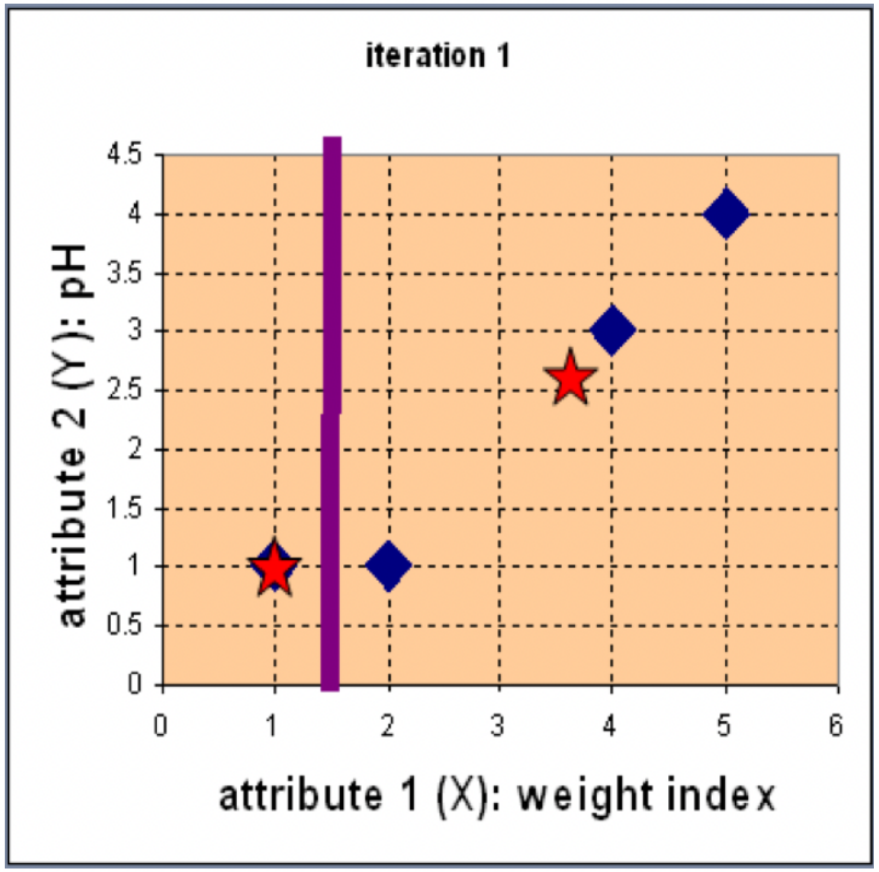
$$d(D,c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

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

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.

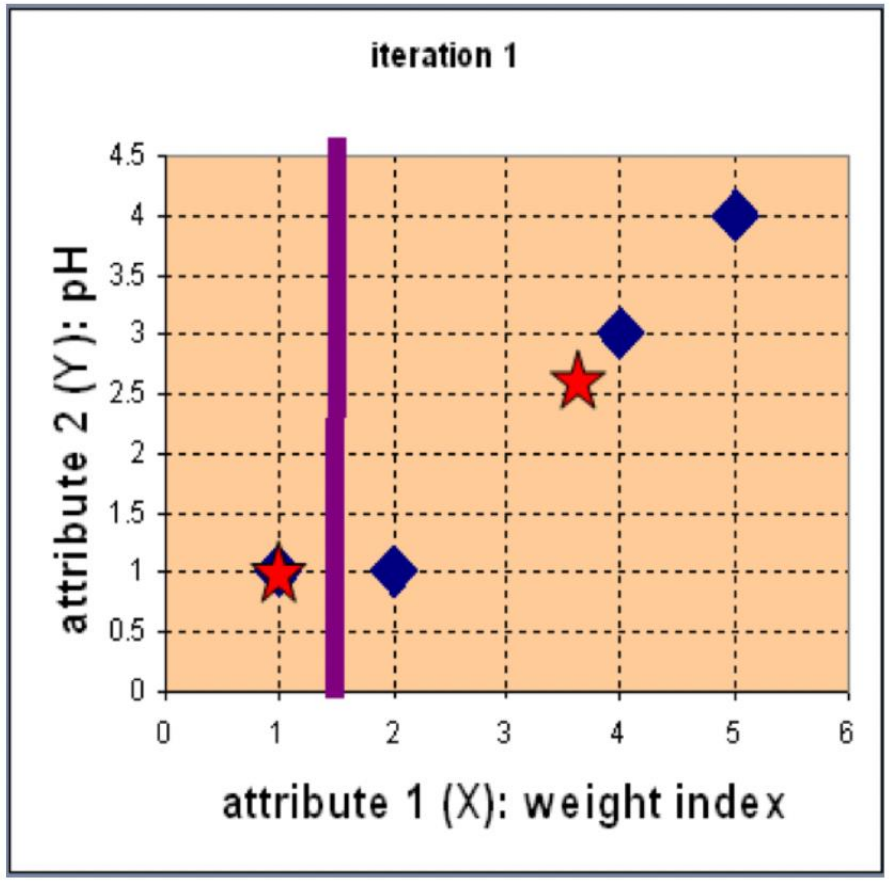
$$c_1 = (1,\ 1)$$

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

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

Example

- Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

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

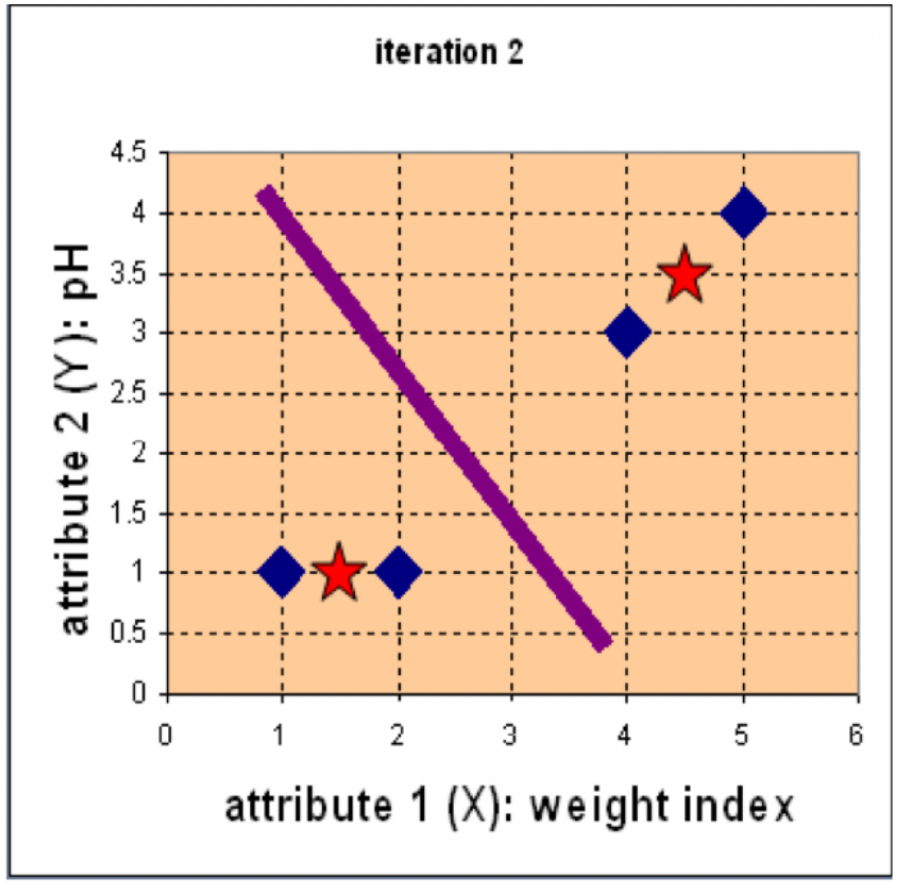
$$\begin{matrix} A & B & C & D \end{matrix}$$

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

Assign the membership to objects

Example

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



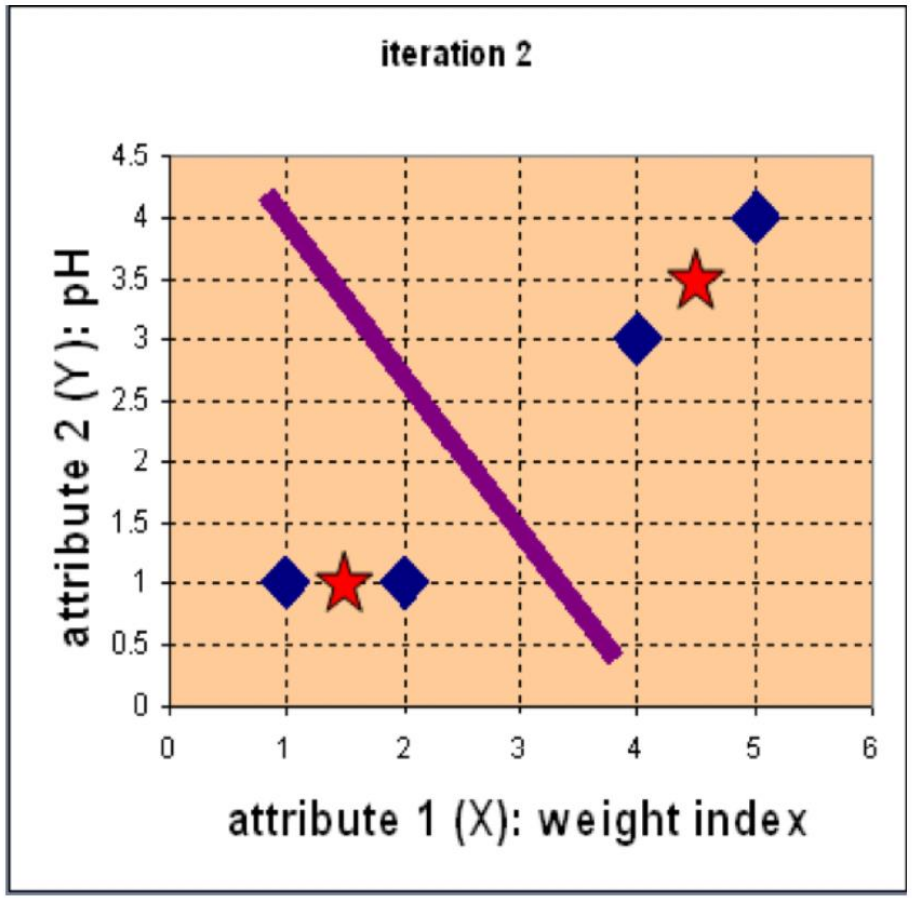
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

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

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

Example

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



Compute the distance of all objects to the new centroids

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

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

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

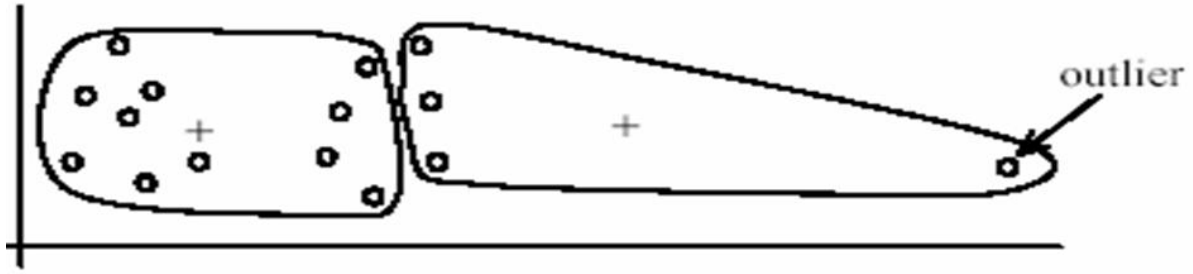
Stop due to no new assignment
Membership in each cluster no longer change

K-means: pros and cons

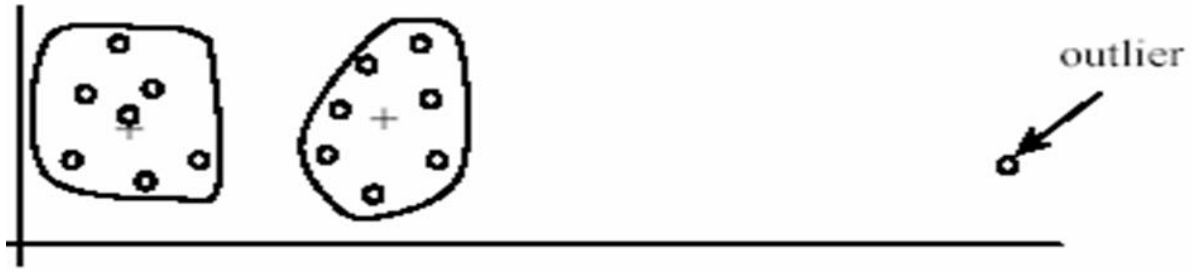
- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error

- Cons/issues
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters
  - Assuming means can be computed

Sensitive to outliers

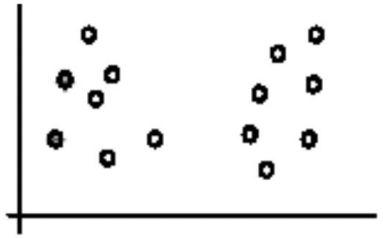


(A): Undesirable clusters

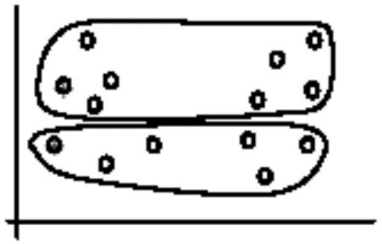


(B): Ideal clusters

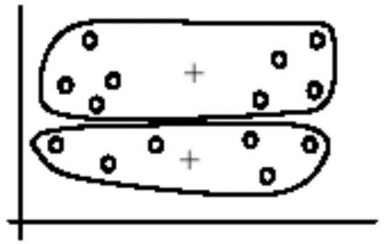
Sensitive to initial centers



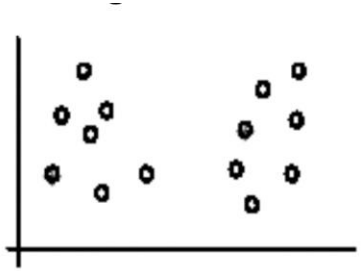
(A). Random selection of seeds (centroids)



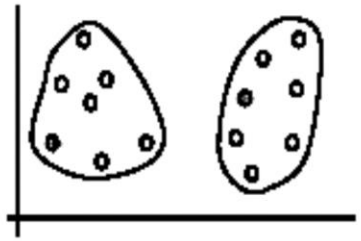
(B). Iteration 1



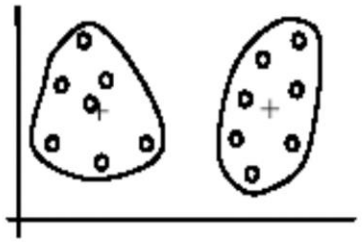
(C). Iteration 2



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

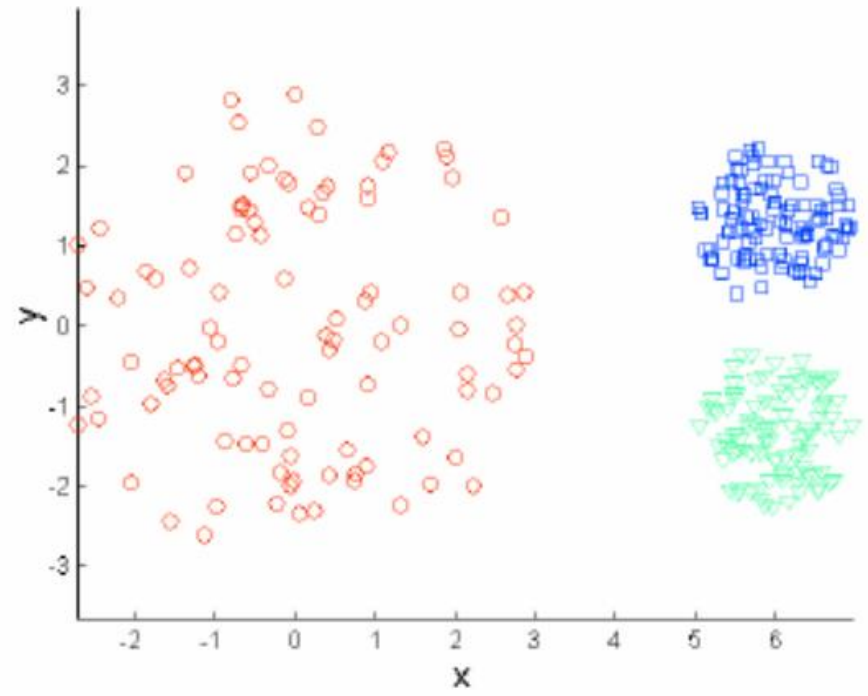


(B). Iteration 1

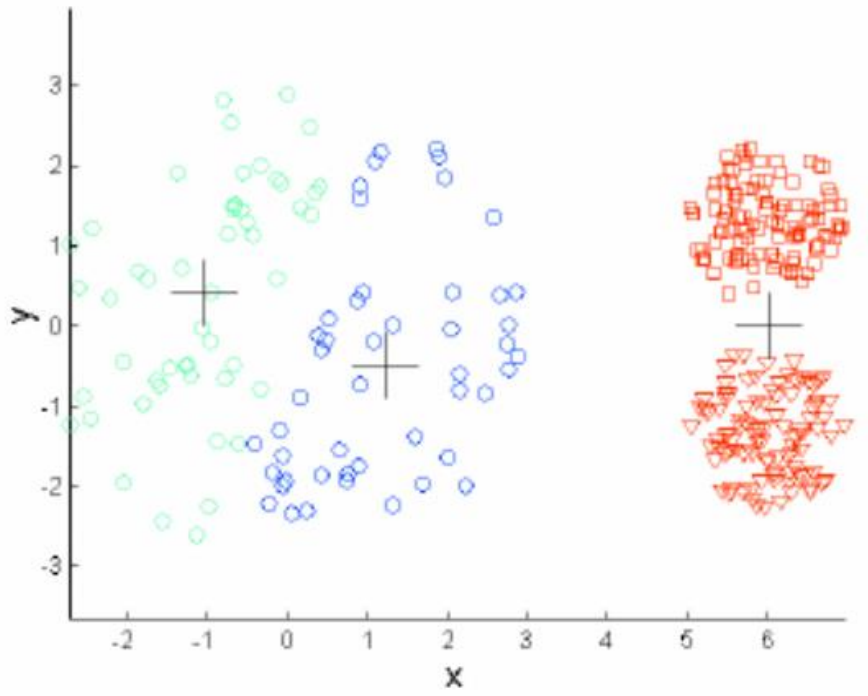


(C). Iteration 2

Differing density

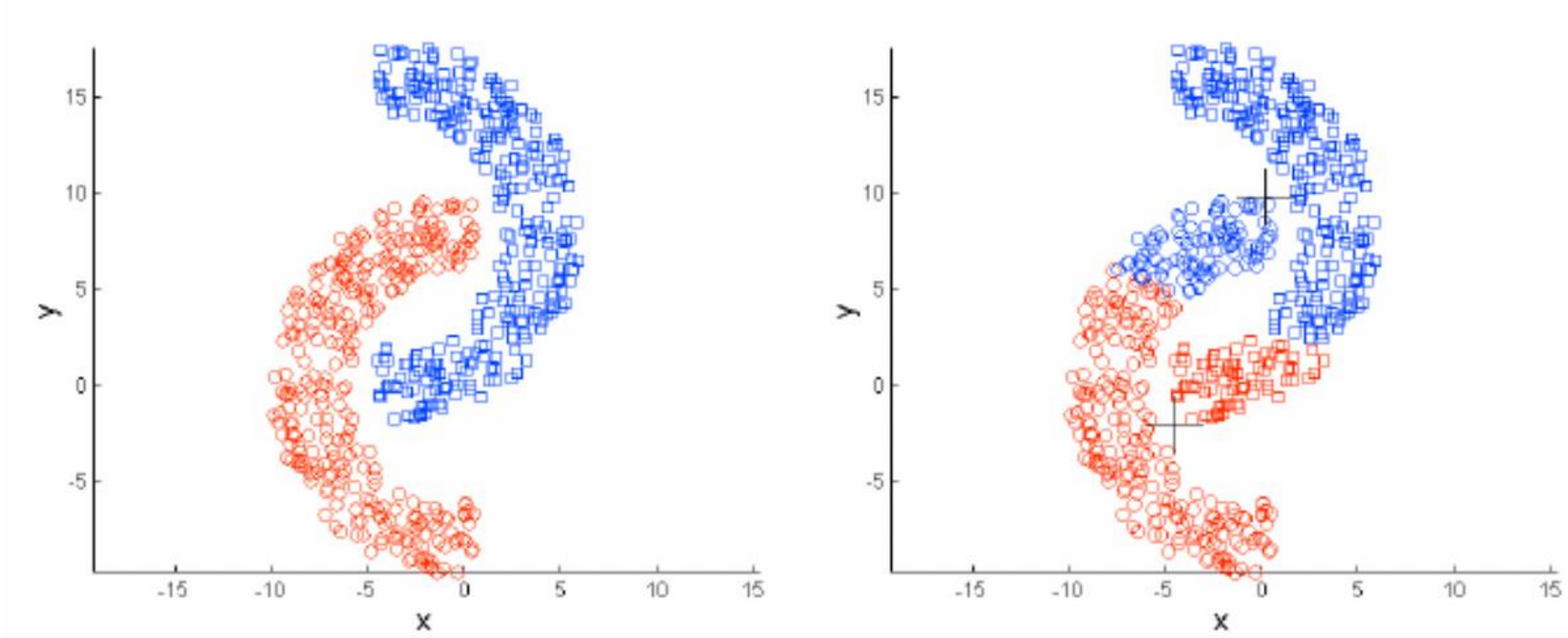


Original Points



K-means (3 Clusters)

Non-convex clusters



Original Points

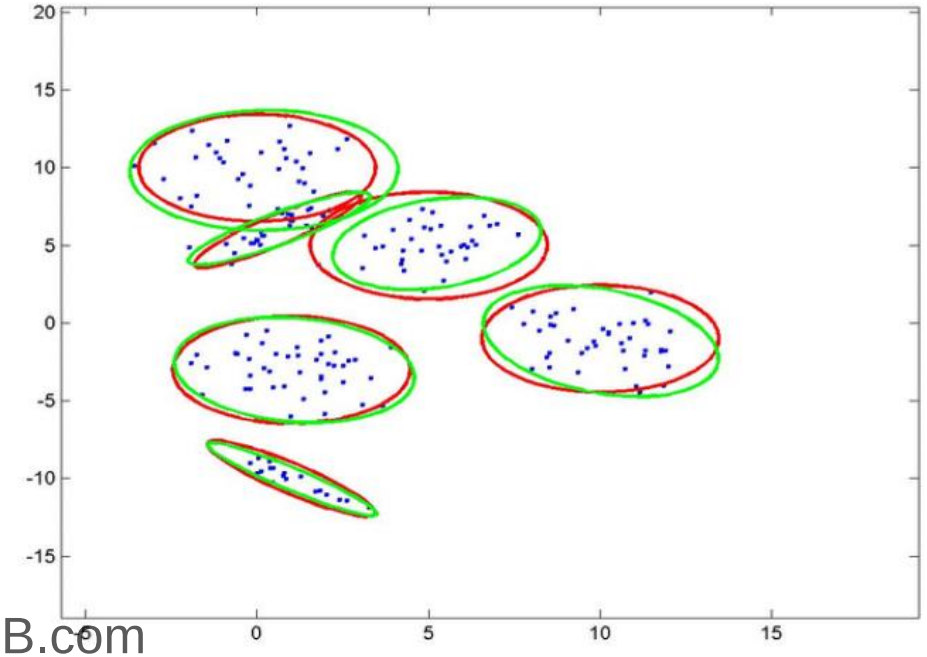
K-means (2 Clusters)

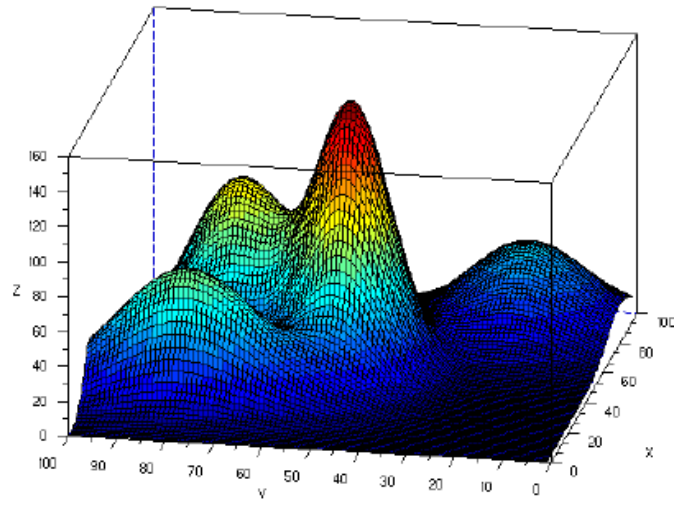
Soft vs Hard Clustering

- K-means assigns each data point exactly to one of the clusters.
- There may be data points that lie roughly midway between cluster centers.
- Use probabilistic approach to obtain “soft” assignments of data points to clusters in a way that reflects the level of uncertainty over the most appropriate assignment

Idea

- Rather than identifying clusters by “nearest” centroids.
- Fit a Set of k Gaussians to the data.
- Maximum Likelihood over a mixture model

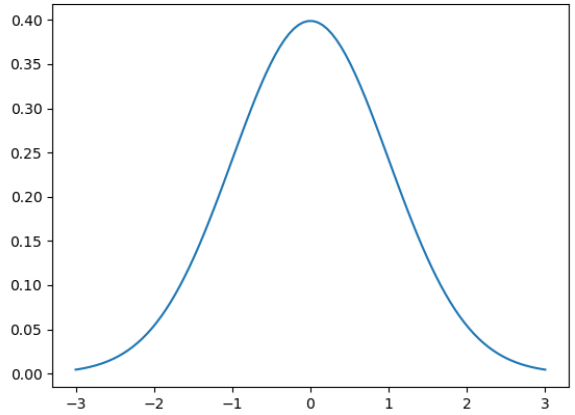




Multi-Variate Gaussian Distribution

- Recall: one-dimensional (univariate) Gaussian distribution:

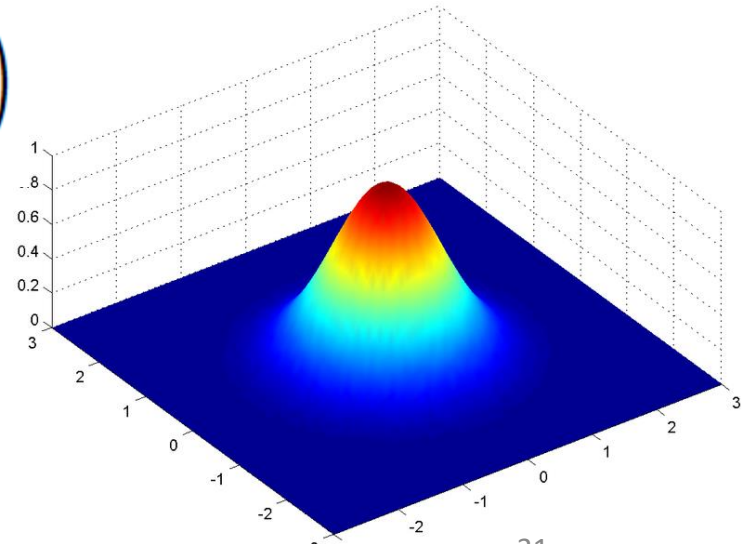
$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$



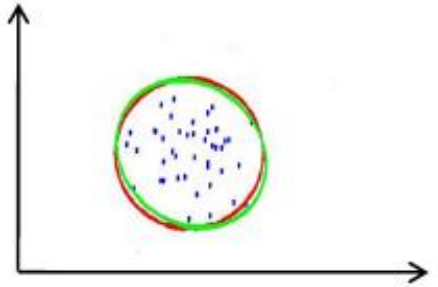
- Multivariate case:

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

Mean $\mu \in \mathbb{R}^d$, covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$

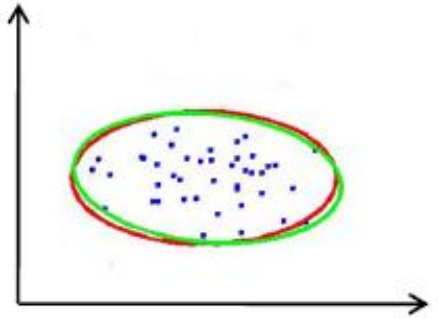


Different covariance matrices



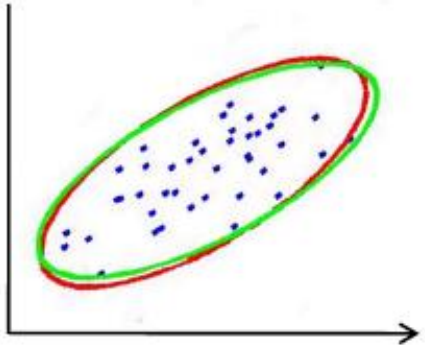
Covariance matrix $\Sigma =$

$$\begin{matrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{matrix}$$



Covariance matrix $\Sigma =$

$$\begin{matrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{matrix}$$



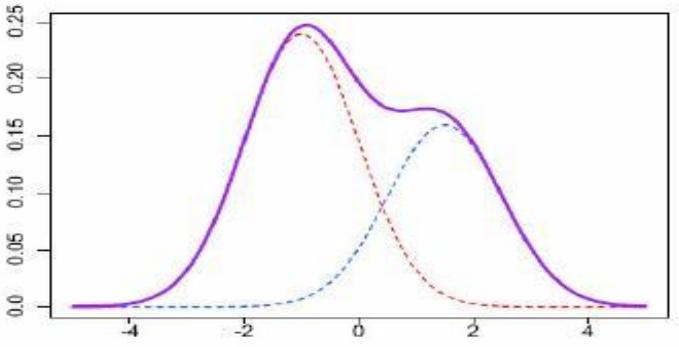
Covariance matrix $\Sigma =$

$$\begin{matrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{matrix}$$

Using different forms of covariance matrix allows for clusters of different shapes

Gaussian Mixture Model

- Often a simple Gaussian distribution is unable to capture the structure of a data set, whereas a linear superposition of several Gaussians gives a good characterization:



- GMM: the weighted sum of a number of Gaussians where the weights are determined by a distribution π

$$p(x|\pi,\mu,\Sigma) := \sum_{j=1}^{k} \pi_j \mathcal{N}(x|\mu_j,\Sigma_j)$$

$$\text{where} \sum_{i=0}^{k} \pi_i = 1$$

Mixture of Gaussians

- Our goal is to find the parameters (π, μ, Σ)
- One way to set the values of these parameters is to use maximum likelihood
- Given the data $X = \{x_1,...,x_m\}$, log of the likelihood function is given by

$$\log p(X|\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)$$

- where θ = set of parameters ($(\pi,\mu,\Sigma)_1$ ... $(\pi,\mu,\Sigma)_k$)

Maximizing the likelihood function

- Find partial derivatives
- Set them to zero
- We get:

$$\mu_j = \frac{1}{m_j} \sum_{i=1}^{m} \gamma_{ij} x_i$$

$$\Sigma_j = \frac{1}{m_j} \sum_{i=1}^{m} \gamma_{ij} (x_i - \mu_j)(x_i - \mu_j)^\top$$

$$\pi_j = \frac{m_j}{m}$$

with responsibilities:

$$\gamma_{ij} = \frac{\pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}{\sum_{j'} \pi_{j'} \mathcal{N}(x_i | \mu_{j'}, \Sigma_{j'})}$$

$$m_j = \sum_{i=1}^{m} \gamma_{ij}$$

Expectation Maximization Algorithm

Given:

- Gaussian Mixture Model
- Number of clusters k
- Goal: maximize the likelihood function with respect to the parameters (mean and the covariance of the components and its mixing coefficients)

Expectation Maximization Algorithm

1. Initialize the means, covariances and mixing coefficients; evaluate the initial value of the log likelihood

2. E-Step: evaluate the responsibilities

$$\gamma_{ij} = \frac{\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}{\sum_{j'} \pi_{j'} \mathcal{N}(x_i|\mu_{j'}, \Sigma_{j'})}$$

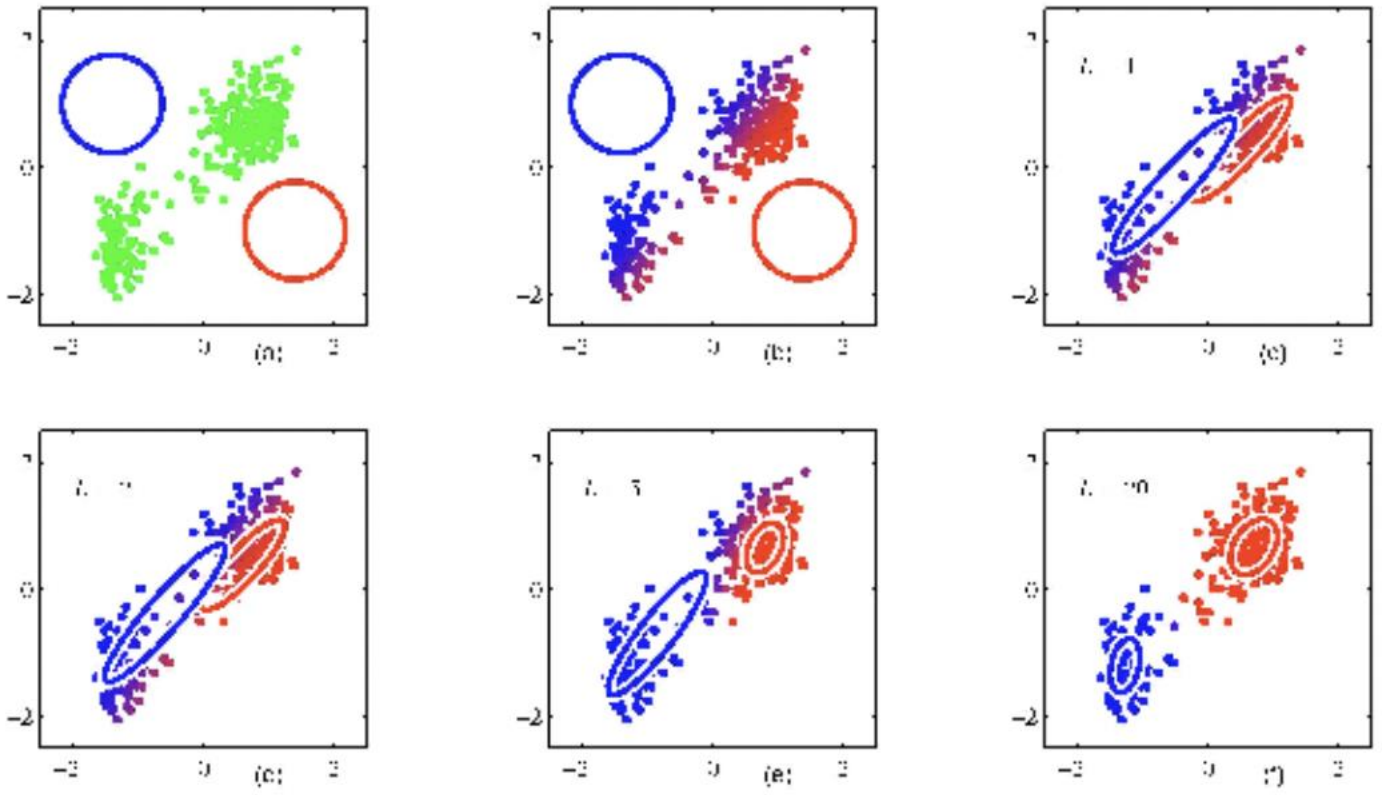
$$m_j = \sum_{i=1}^{m} \gamma_{ij}$$

3. M-Step: Re-estimate parameters

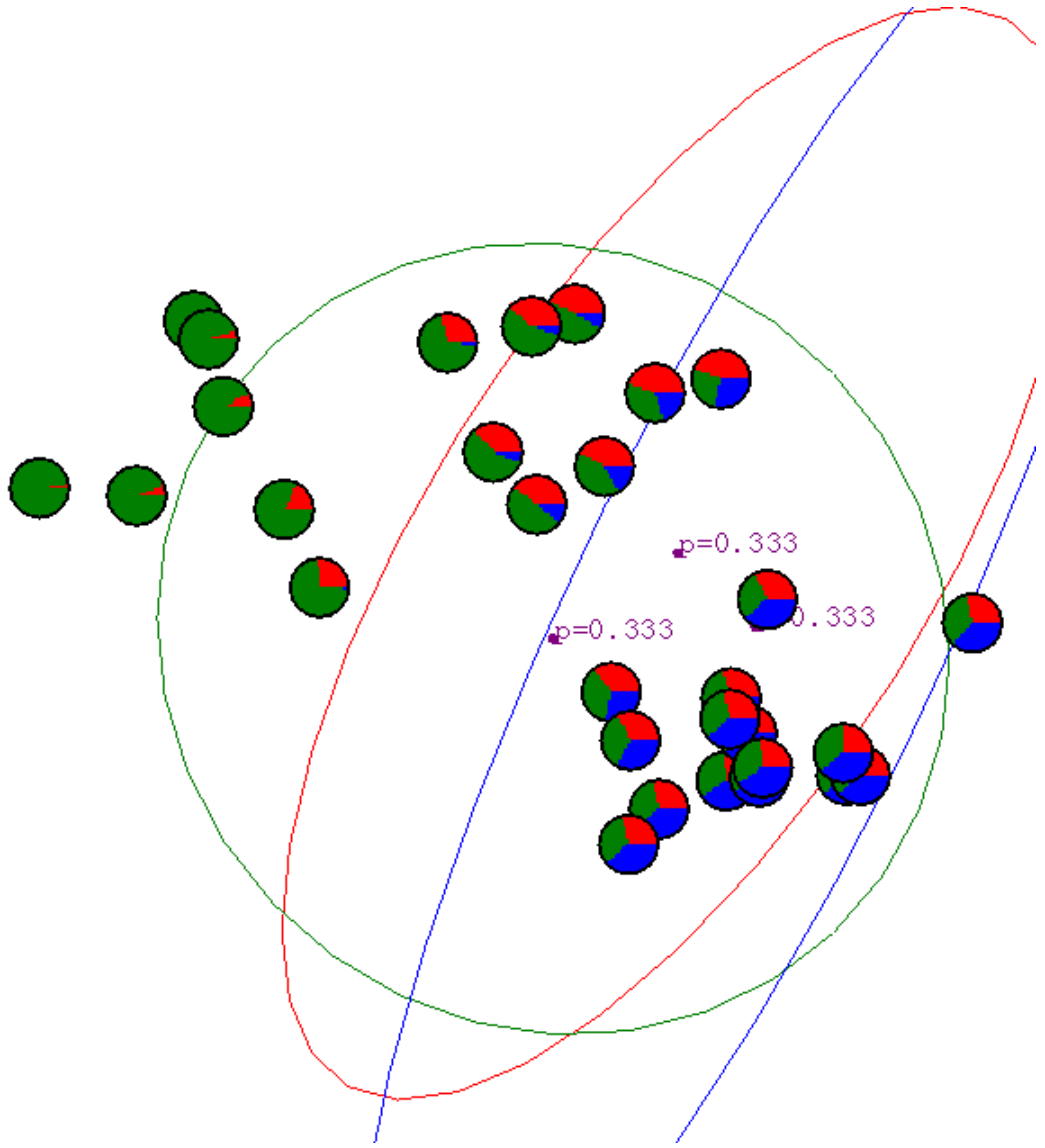
$$\mu_j = \frac{1}{m_j} \sum_{i=1}^{m} \gamma_{ij} x_i \qquad \Sigma_j = \frac{1}{m_j} \sum_{i=1}^{m} \gamma_{ij} (x_i - \mu_j)(x_i - \mu_j)^\top \qquad \pi_j = \frac{m_j}{m}$$

4. Evaluate the log likelihood and check for convergence. If not converged, go to step 2.

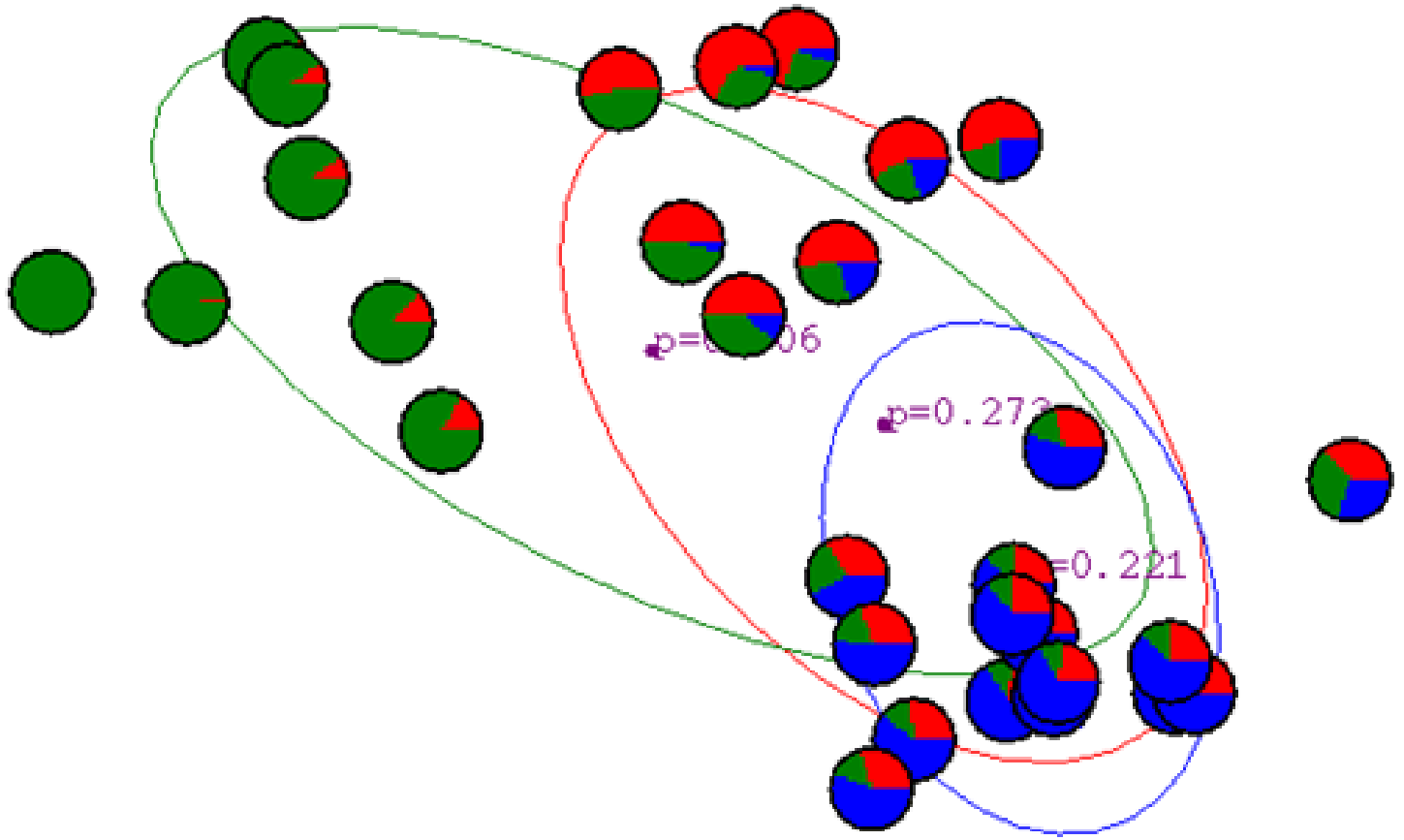
Visual example of EM



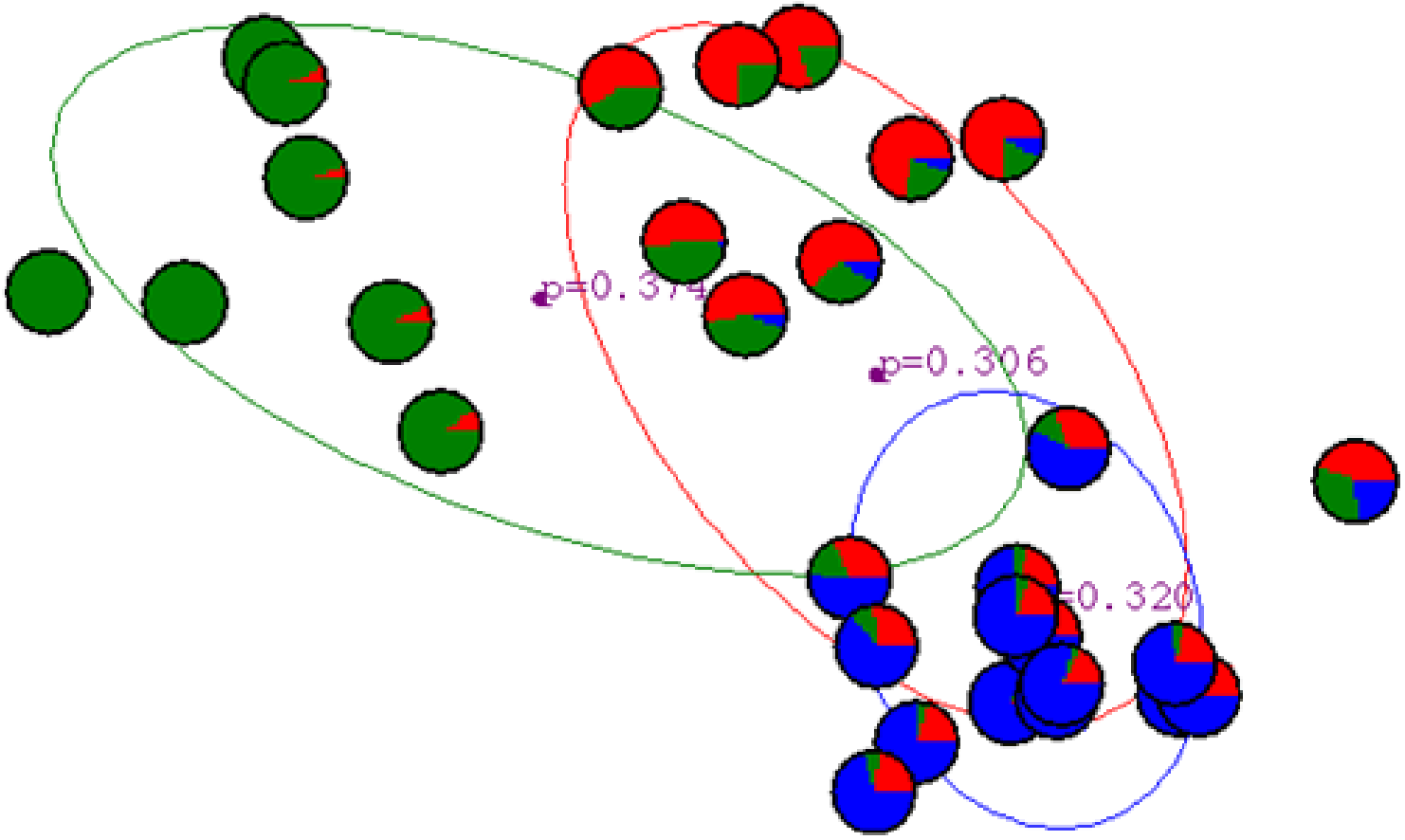
Gaussian Mixture Example: Start



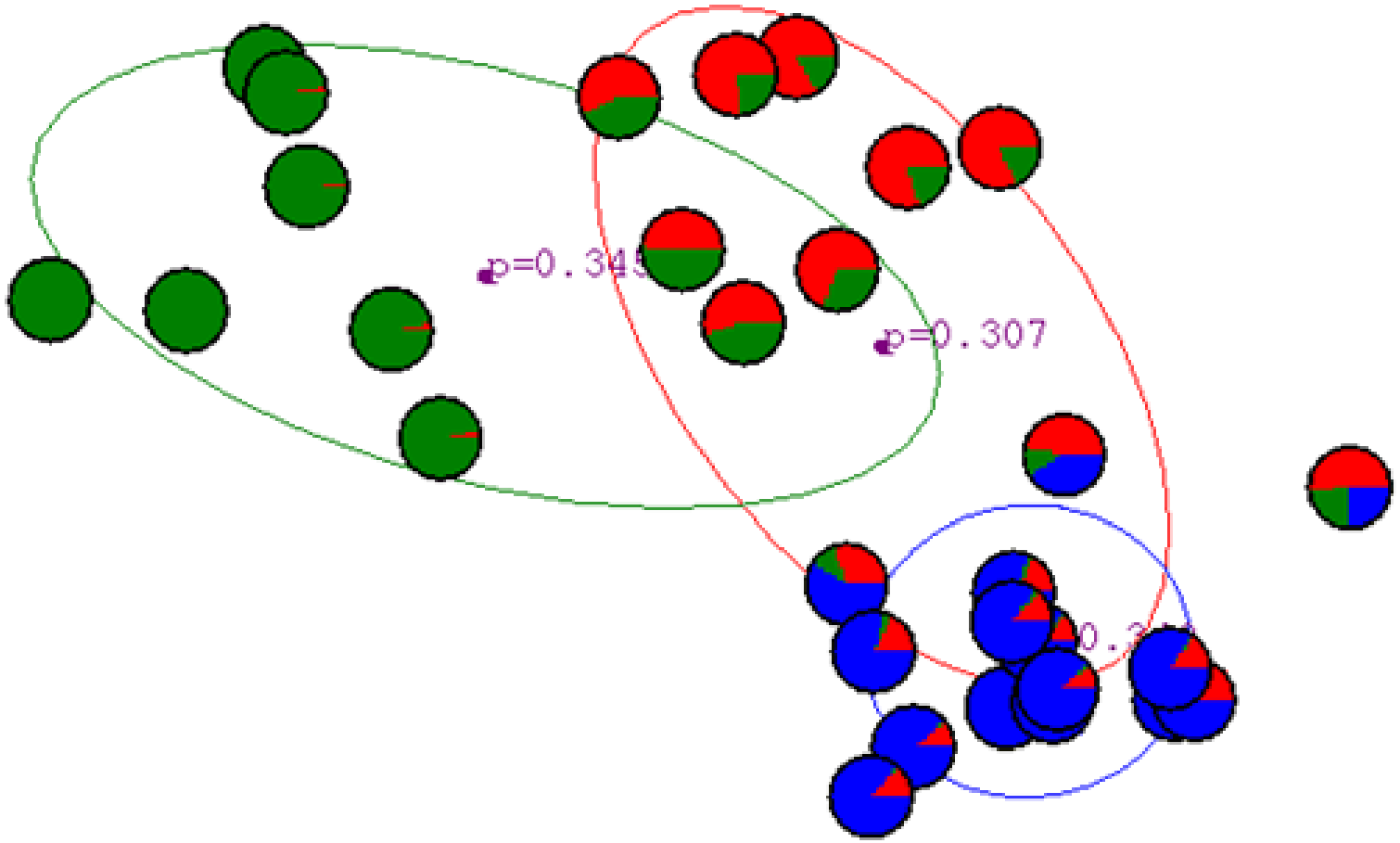
After First Iteration



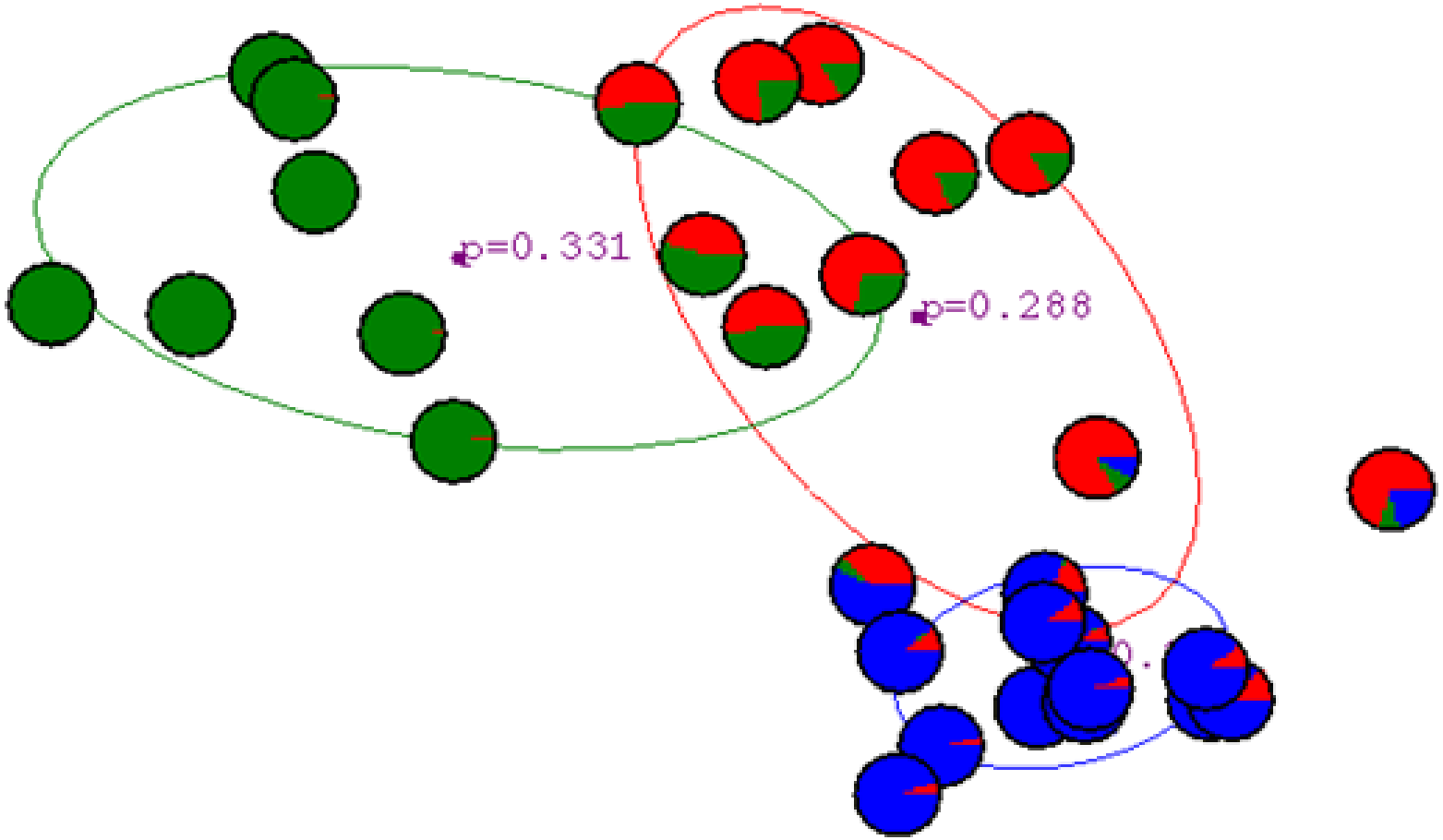
After 2nd Iteration



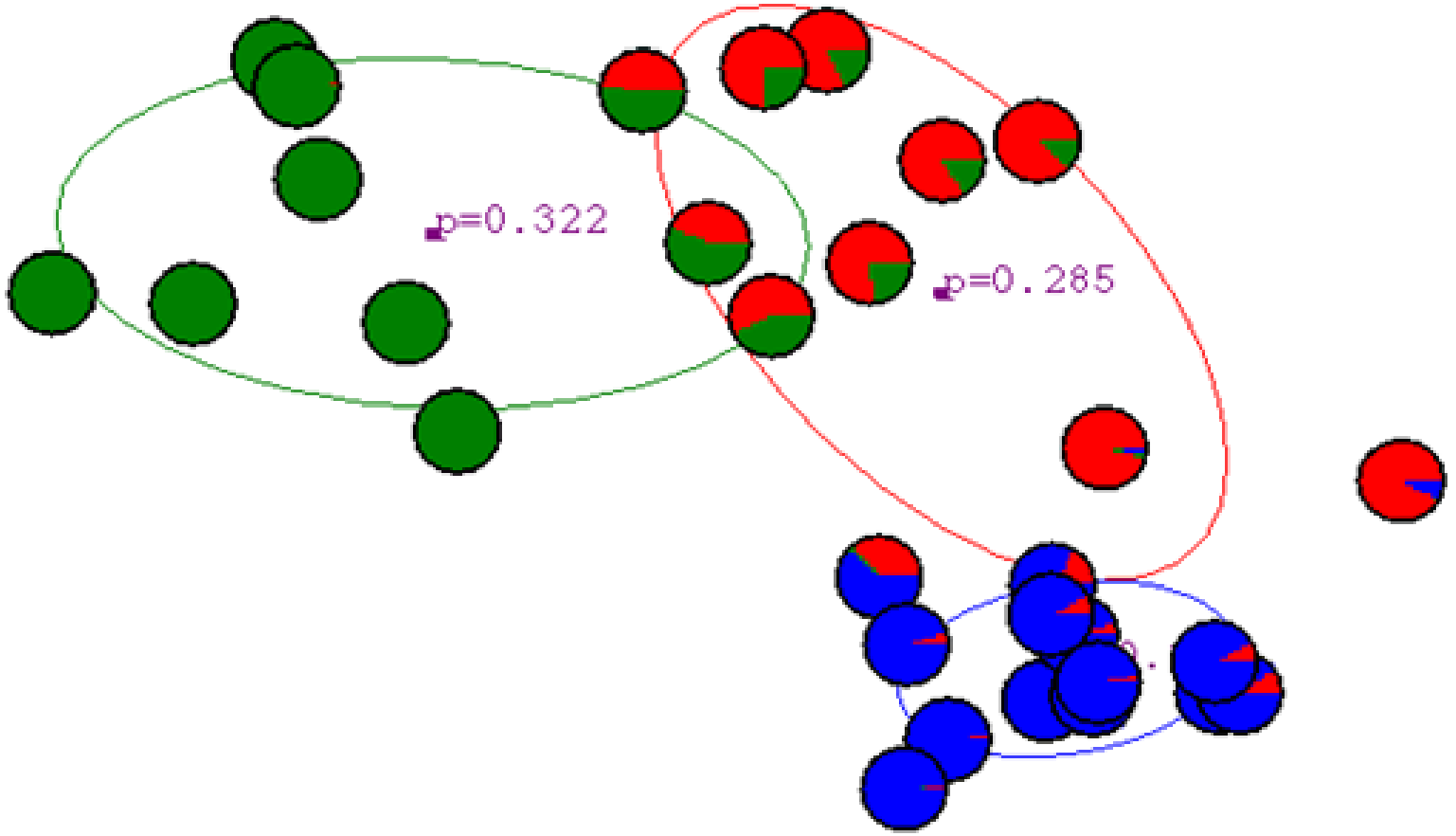
After 3rd Iteration



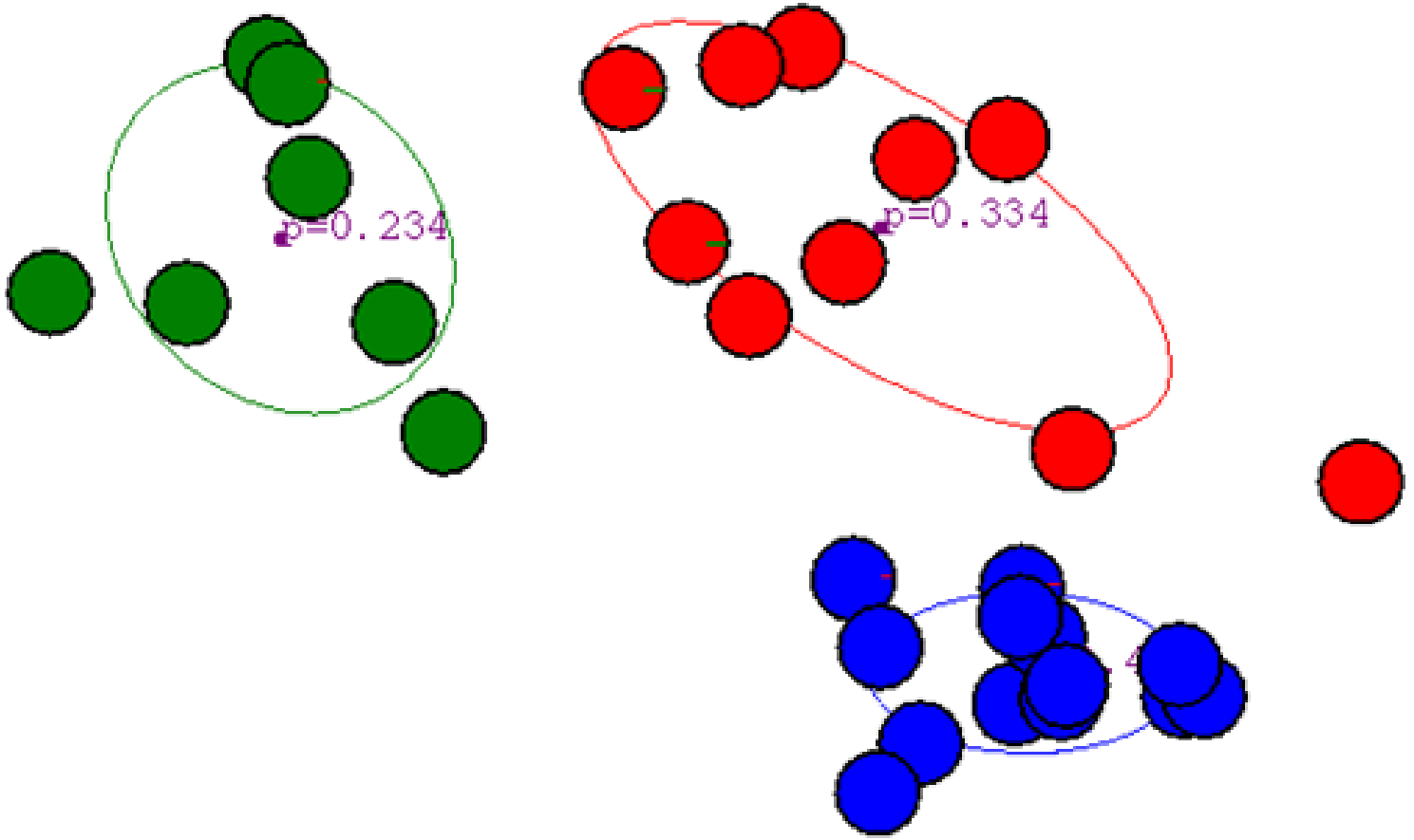
After 4th Iteration



After 5th Iteration



After 20th Iteration



Problems

- EM algorithm takes many more iterations to reach convergence compared with k-means and each cycle requires significantly more computation

  → run k-means to find suitable initialization for EM

- Incorrect number of Mixture Components

Hierarchial Clustering

Hierarchical Clustering

hierarchical clustering of a set $I = \{x_1, \ldots, x_n\}$ of instances:

a sequence $\mathcal{C}_1, \ldots, \mathcal{C}_n$ of nested partitions of I, where

- $\mathcal{C}_1$ contains n clusters, i.e., $\mathcal{C}_1 = \{\{x_1\}, \ldots, \{x_n\}\}$

  $\mathcal{C}_2$ contains $n-1$ clusters

  $\vdots$

  $\mathcal{C}_n$ contains 1 cluster, i.e., $\mathcal{C}_n = \{\{x_1, \ldots, x_n\}\}$

- if x, x' belong to the same cluster in $\mathcal{C}_k$ then they belong to the same cluster in $\mathcal{C}_l$ for all $l = k, k+1, \ldots, n$

dendrogram: natural (binary) tree representation of a hierarchial clustering

Dendrograms

set of nested partitions can be visualized as a dendrogram

height: proportional to the distance/similarity at which two clusters have been merged

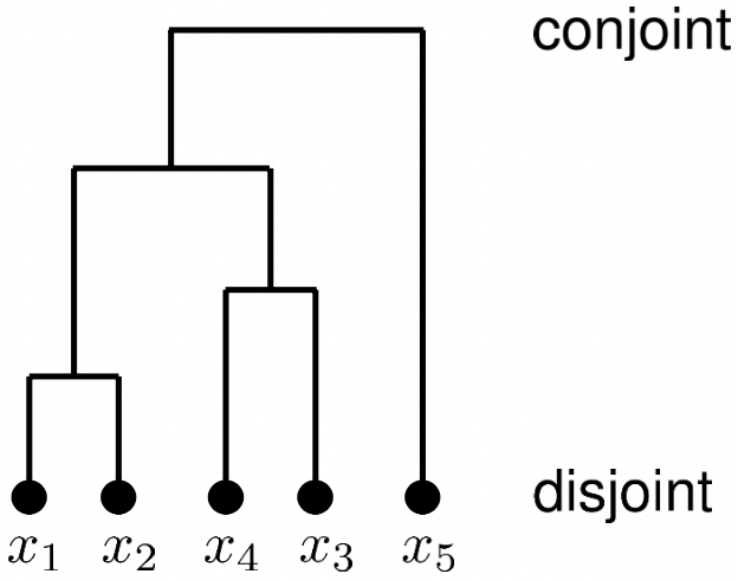
$$\mathcal{C}_5 = \{\{x_1, x_2, x_3, x_4, x_5\}\}$$

$$\mathcal{C}_4 = \{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}$$

$$\mathcal{C}_3 = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}\}$$

$$\mathcal{C}_2 = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$$

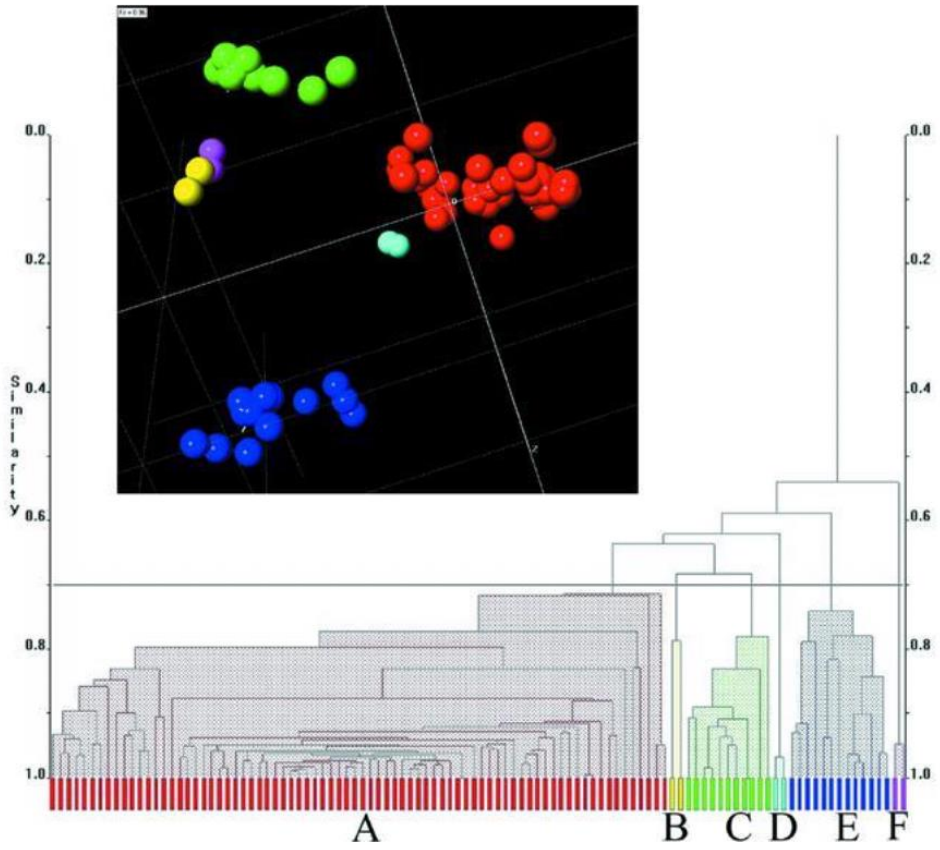
$$\mathcal{C}_1 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$$



Dendrograms

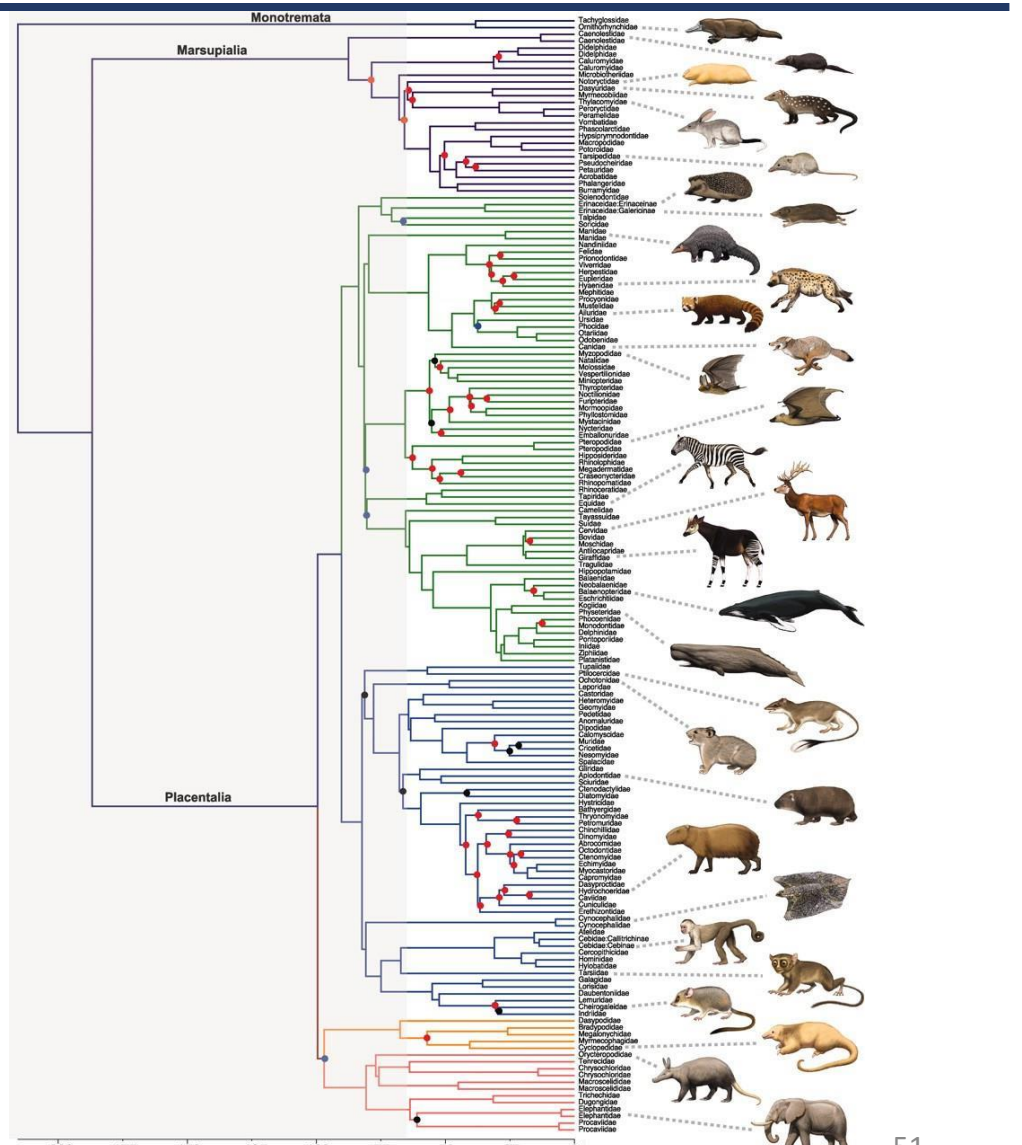
set of nested partitions can be visualized as a dendrogram

height: proportional to the distance/similarity at which two clusters have been merged



Strengths of Hierarchical Clustering

- no assumptions on the number of clusters (in contrast to e.g. k-Means)
  - any desired number of clusters can be obtained by “cutting” the dendogram at the proper level
- hierarchical clusterings may correspond to meaningful taxonomies, e.g.,
  - in biological sciences
    - e.g., phylogeny reconstruction
  - web
    - e.g., product catalogs
  - …



Hierarchical Clustering

- two main types of hierarchical clustering
  - **agglomerative (bottom-up, clumping):**
    - start with the instances as singleton clusters
    - at each step, merge the closest pair of clusters until only one cluster left
  - **divisive (top-down, splitting):**
    - start with one, all-inclusive cluster
    - at each step, split a cluster until each cluster becomes a singleton
- traditional hierarchical algorithms use a similarity or distance matrix to merge/split one cluster at a time

Agglomerative Clustering

algorithm:

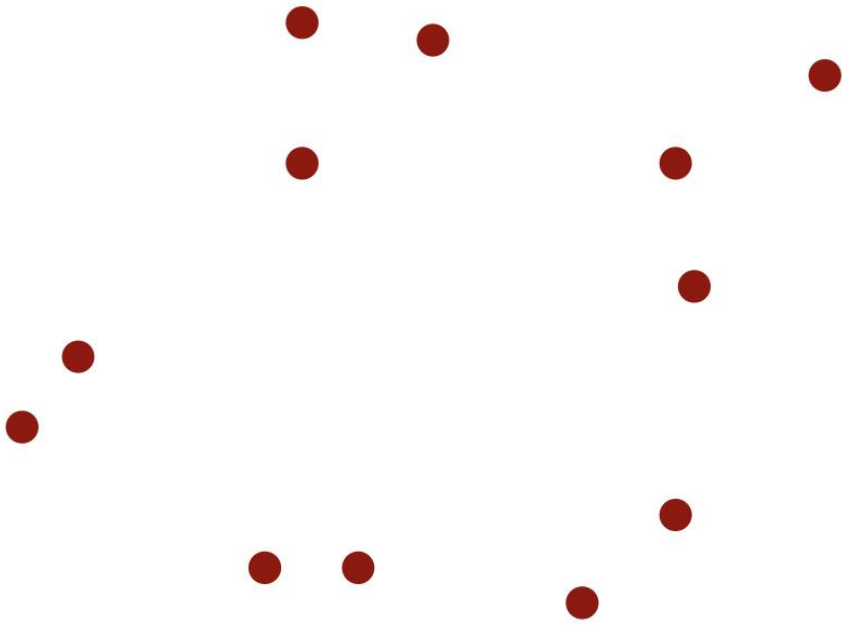
1. $\mathcal{C}_1 = \{\{x_1\}, \ldots, \{x_n\}\}$
2. $k := 1$
3. **for** $(k = 1;\ |\mathcal{C}_k| > 1;\ k = k + 1)$ **do**
4. find “closest" pair $C_i, C_j \in \mathcal{C}_k$ with $i \neq j$
5. $\mathcal{C}_{k+1} := \mathcal{C}_k \setminus \{C_i, C_j\}$
6. $\mathcal{C}_{k+1} := \mathcal{C}_{k+1} \cup \{C_i \cup C_j\}$
7. **return** $\mathcal{C}_1, \ldots, \mathcal{C}_n$

key operation: computing the distance between two clusters

- different distance definitions between clusters lead to different algorithms

Input/Initial setting

- start with clusters of individual points and a distance/proximity matrix

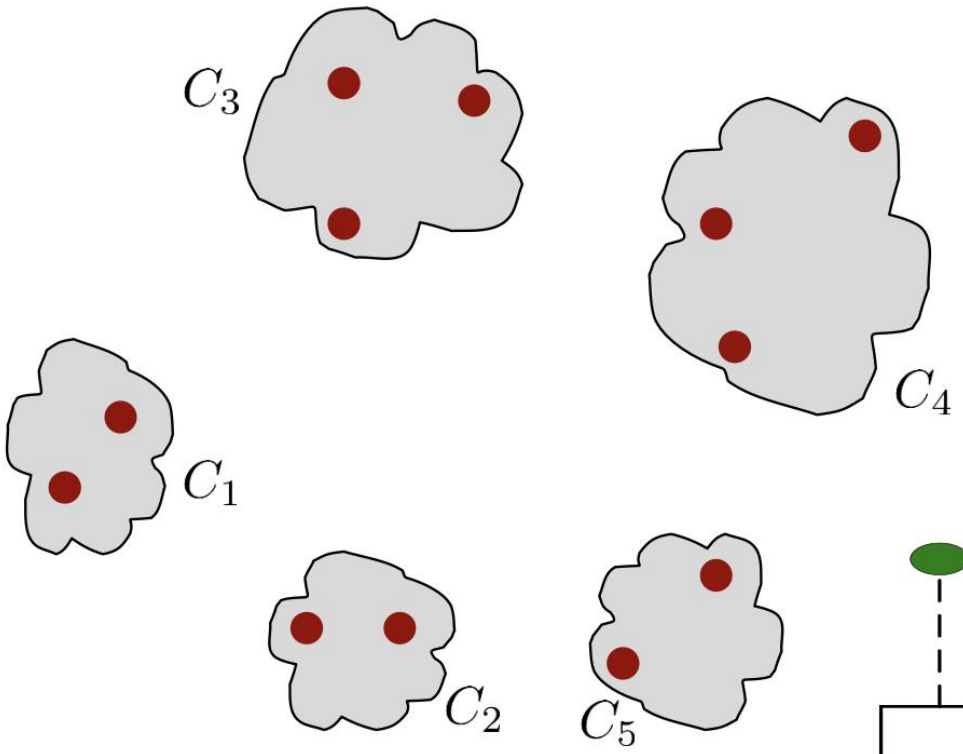


	p_1	p_2	p_3	p_4	p_5	...
p_1						
p_2						
p_3						
p_4						
p_5						
⋮						

distance/proximity matrix

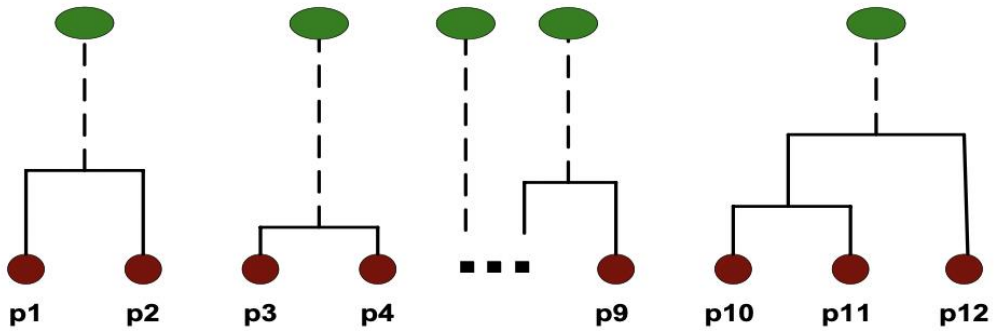
Intermediate State

- after some merging steps, we have some clusters



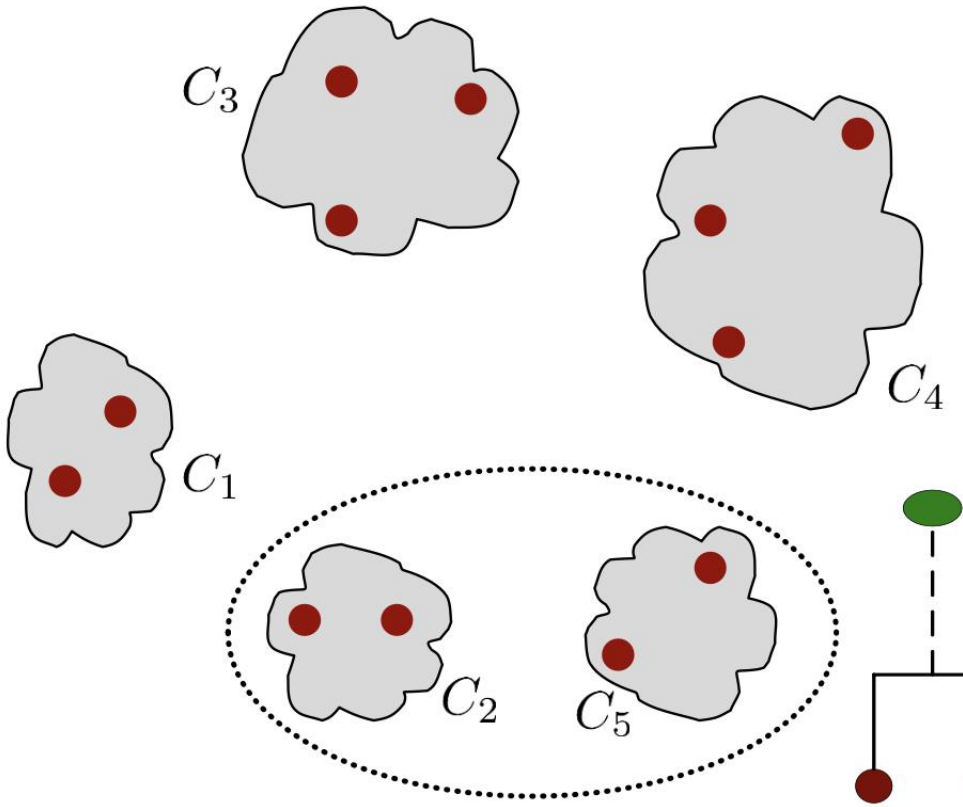
	C_1	C_2	C_3	C_4	C_5
C_1					
C_2					
C_3					
C_4					
C_5					

distance/proximity matrix



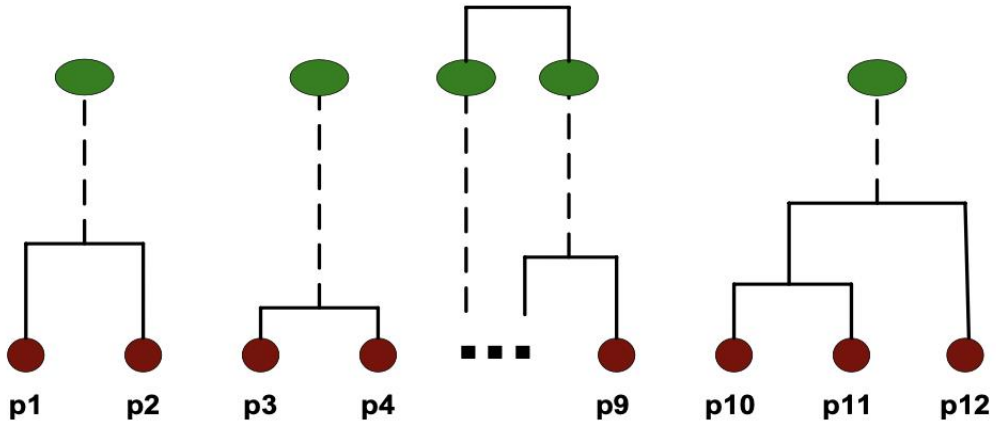
Intermediate State

- merge the two closest clusters (C_2 and C_5) and update the distance/proximity matrix



	C_1	C_2	C_3	C_4	C_5
C_1					
C_2					
C_3					
C_4					
C_5					

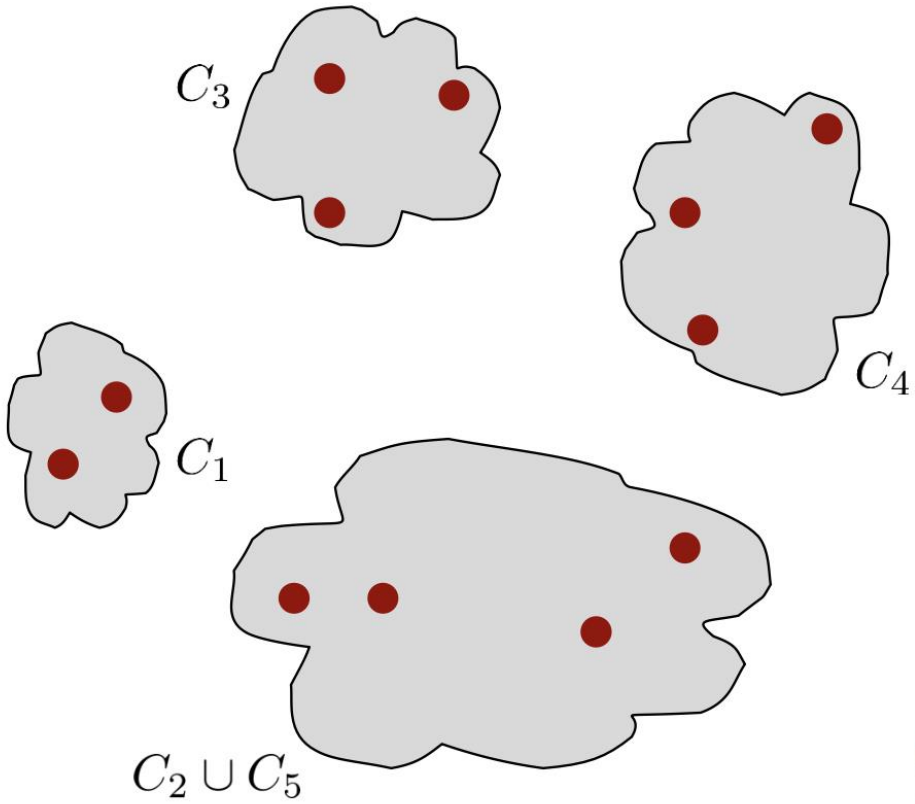
distance/proximity matrix



After Merging

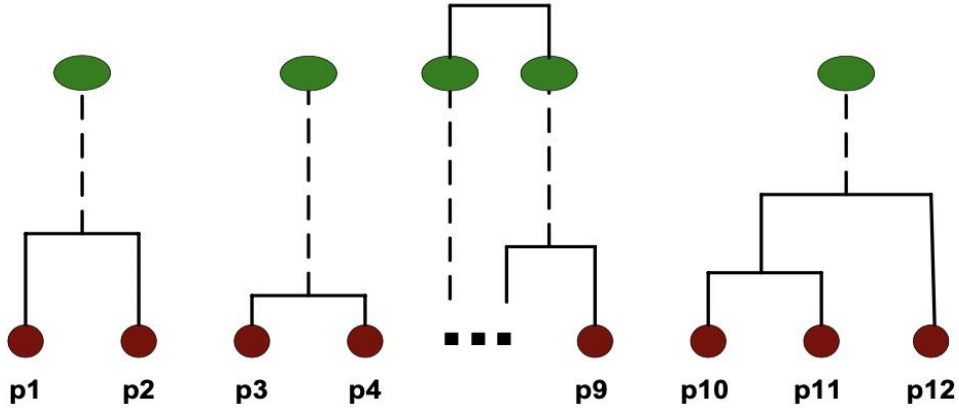
"How do we update the distance matrix?"

- depends on the distance between clusters



	C_1	$C_2 \cup C_5$	C_3	C_4
C_1		?		
$C_2 \cup C_5$	?	?	?	?
C_3		?		
C_4		?		

distance/proximity matrix

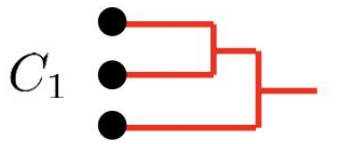


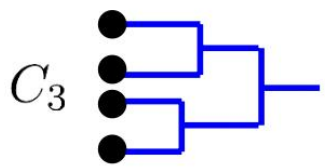
Distance between Two Clusters

each cluster is a set of points

Q: How do we define distance between two sets of points?

A: lots of alternatives . . .





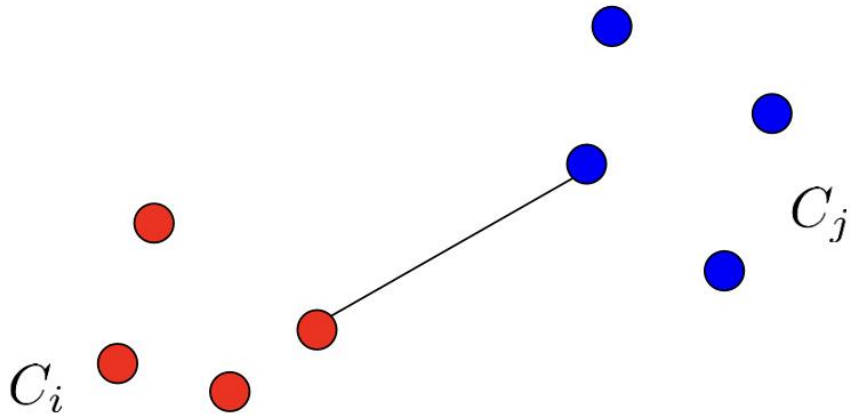
Merge which pair of clusters?

- Depends on the choice of the inter-cluster distance!

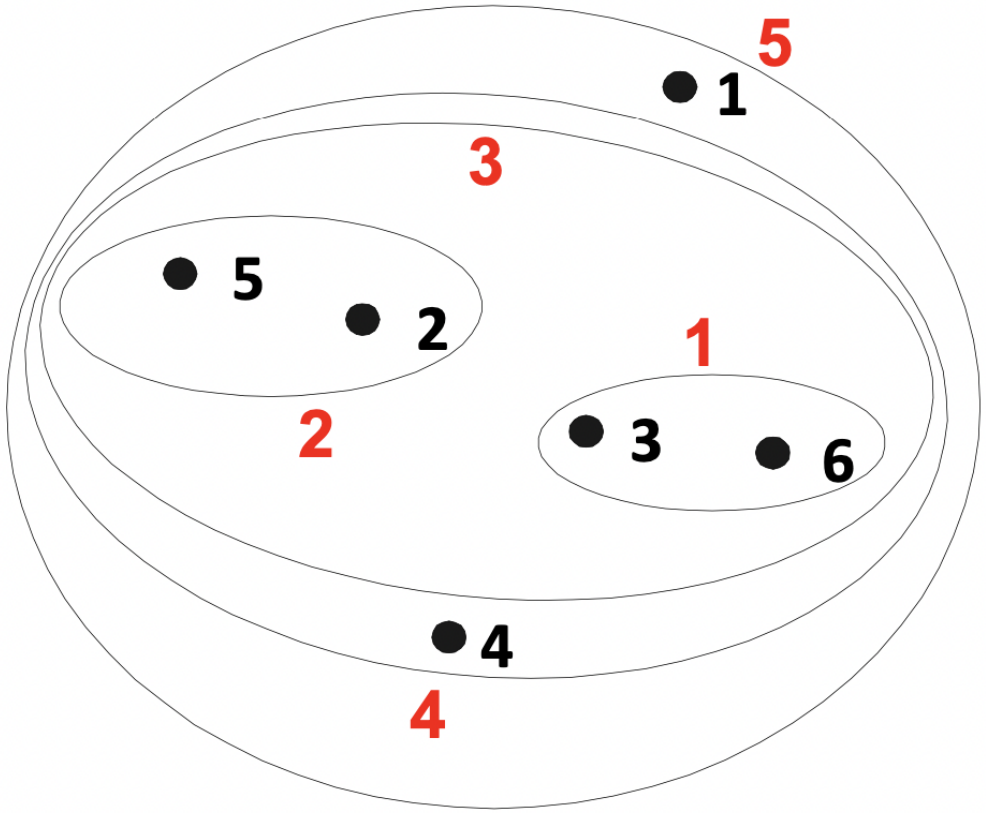
1. Inter-Cluster Distances: Single-Link Distance

single-link distance between two clusters: **minimum** distance between the members of two cluster, i.e.,

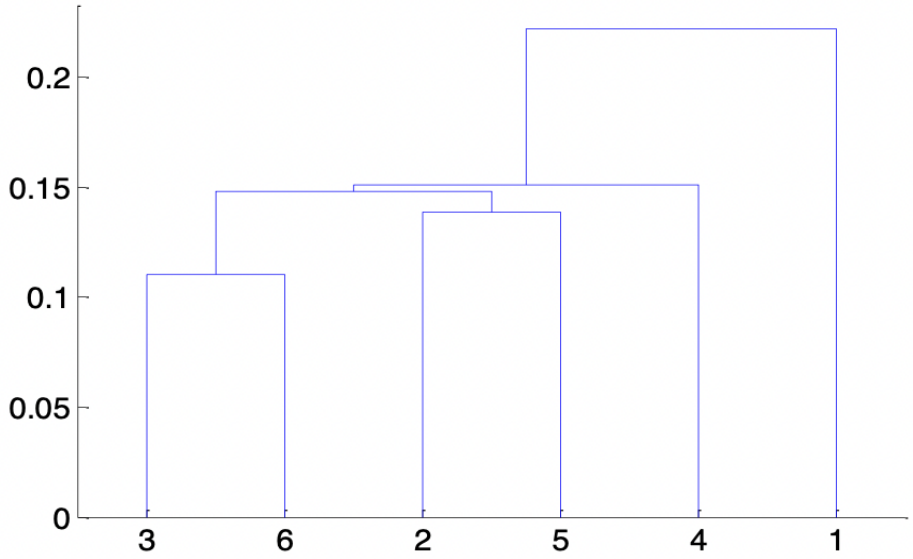
$$d_{\mathrm{SL}}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$



Single-Link Clustering: Example

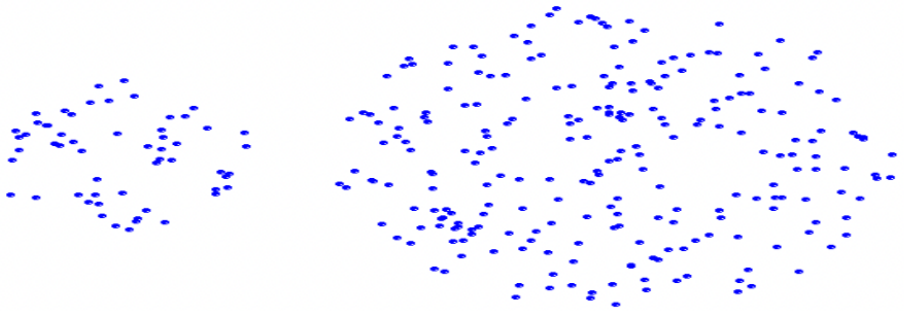


nested clusters

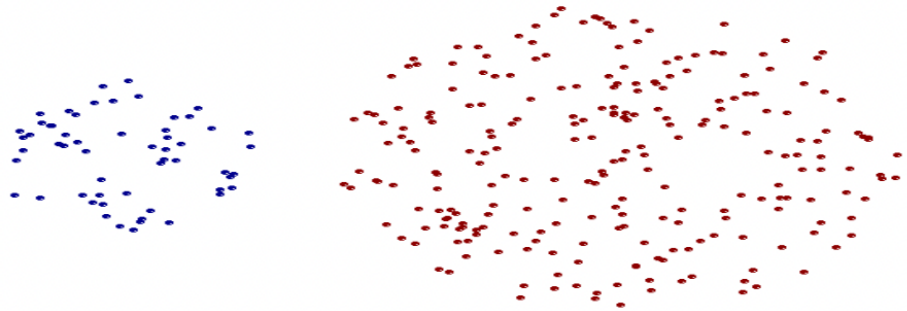


dendrogram

Strengths of Single-Link Clustering



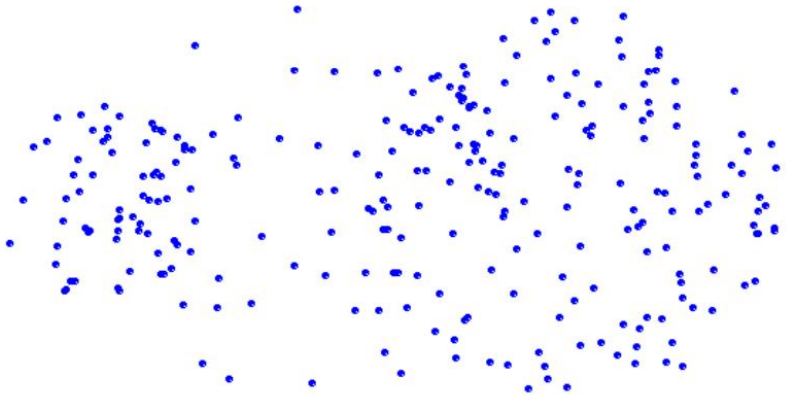
original points



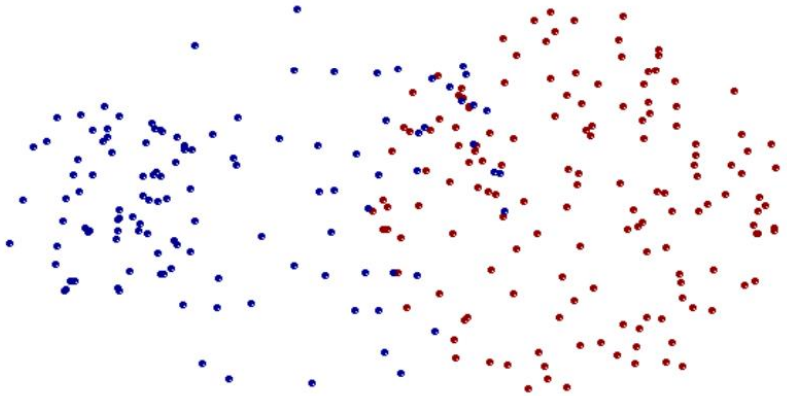
two clusters

☺ Can handle non-elliptical (unsymmetrical) shapes, clusters can be even concentric !

Limitations of Single-Link Clustering



original points



two clusters

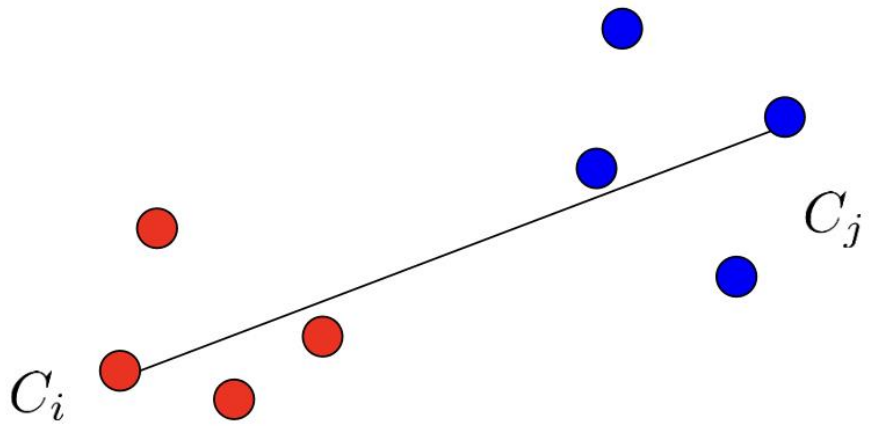
☹ sensitive to noise and outliers

☹ elongated clusters (chaining effect)

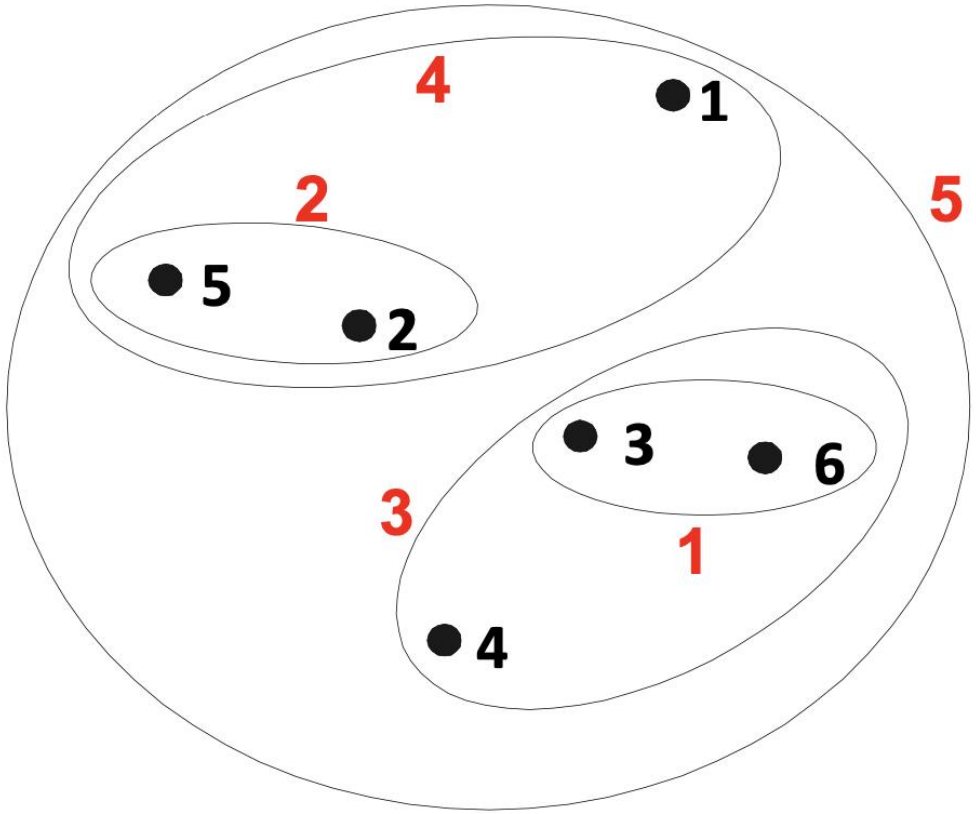
2. Inter-Cluster Distances: Complete-Link Distance

complete-link distance between two clusters: **maximum** distance between the members of two cluster, i.e.,

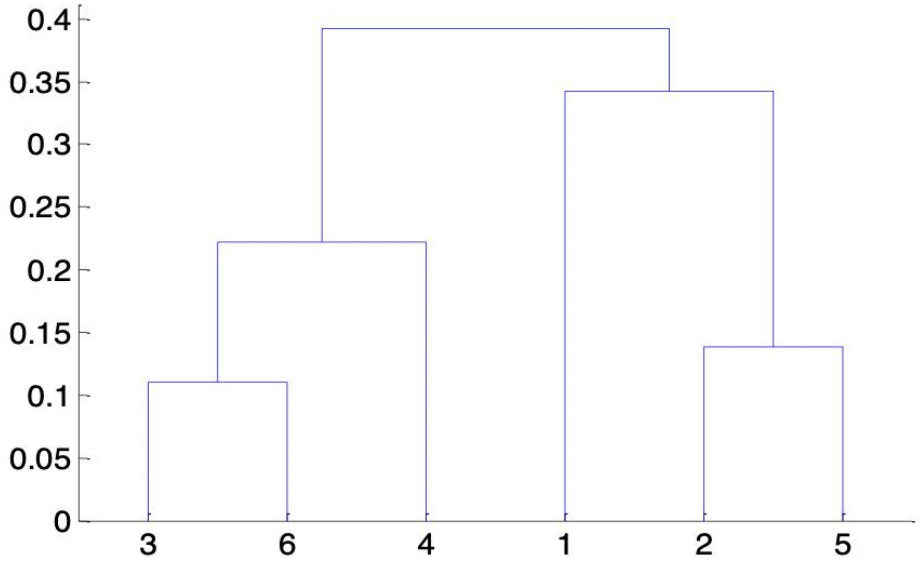
$$d_{\mathrm{CL}}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$



Complete-Link Clustering: Example

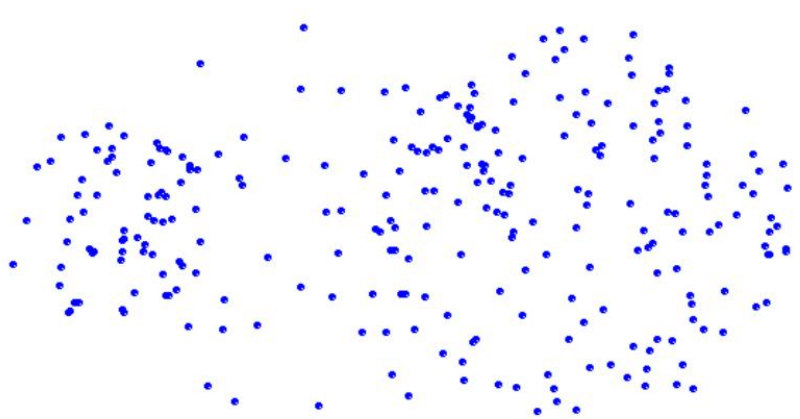


nested clusters

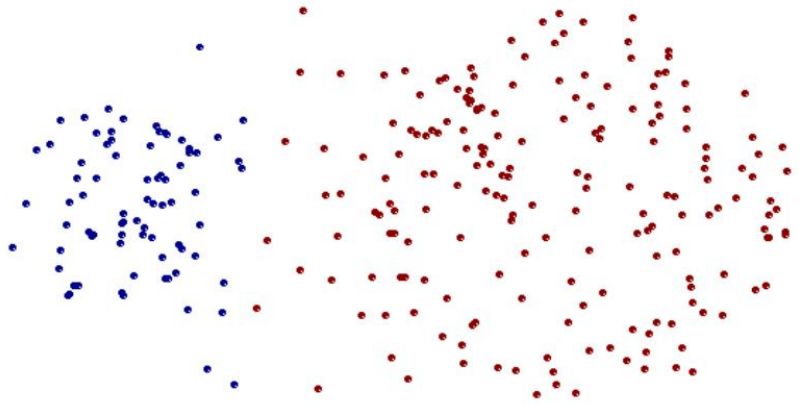


dendrogram

Strengths of Complete-Link Clustering



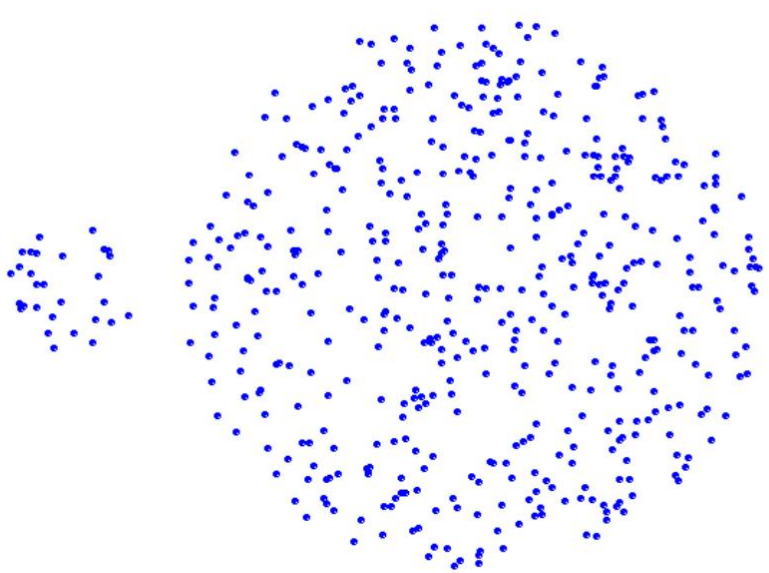
original points



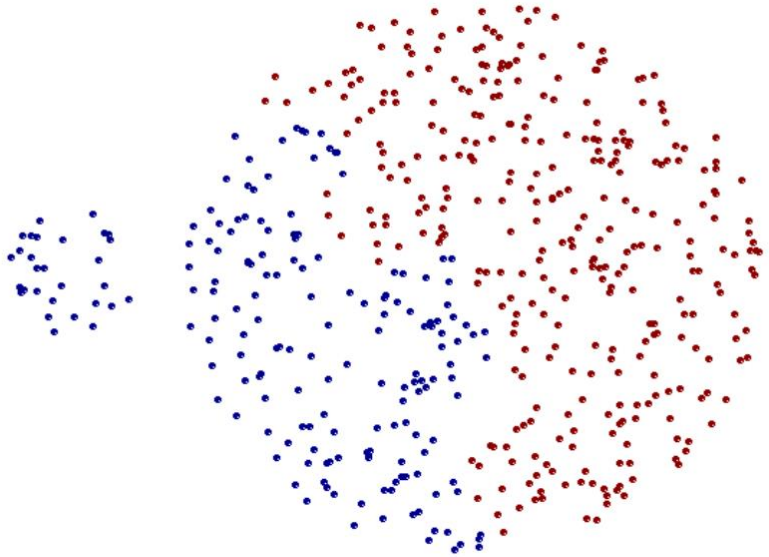
two clusters

- ☺ more balanced clusters (with equal diameter)
- ☺ less sensitive to noise

Limitations of Complete-Link Clustering



original points



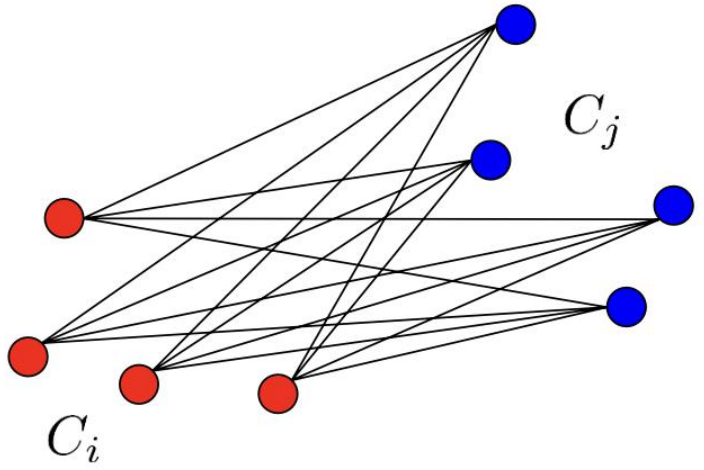
two clusters

- ☹ tends to break large clusters
- ☹ all clusters tend to have the same diameter
  - small clusters are merged with larger ones

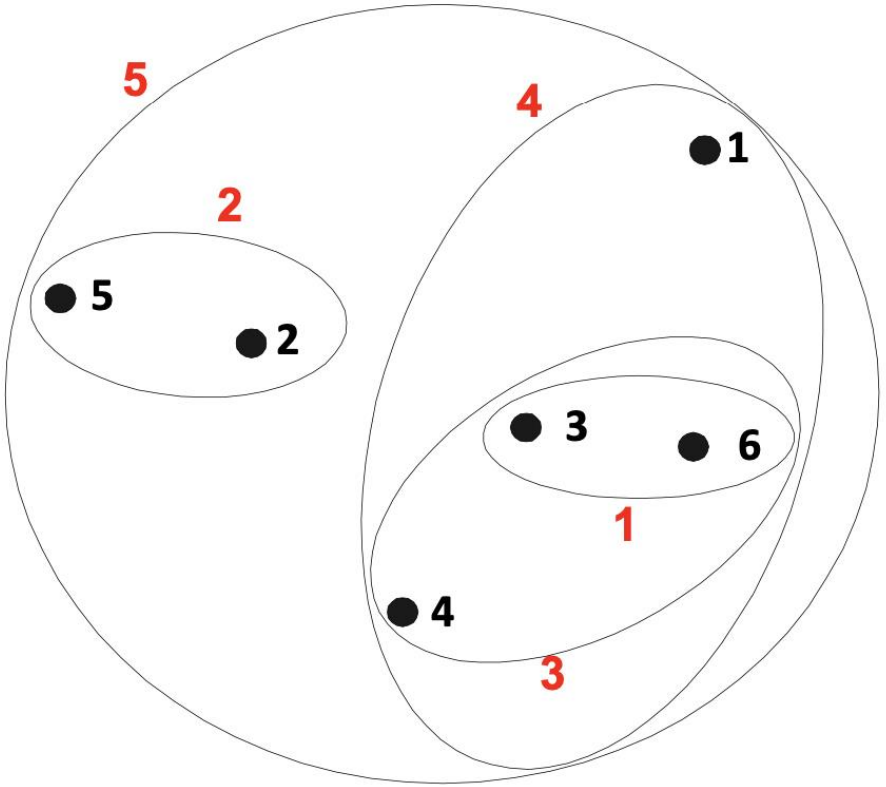
3. Inter-Cluster Distances: Average-Link Distance

average-link distance between two clusters: **averaged** distances of all pairs of objects (one from each cluster), i.e.,

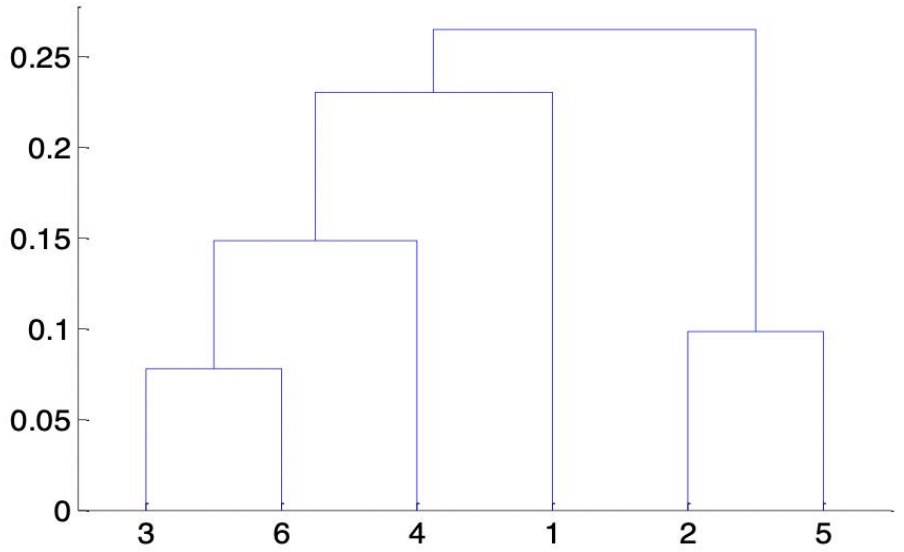
$$d_{\mathrm{AL}}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$



Average-Link Clustering: Example



nested clusters



dendrogram

Average-link Clustering: Discussion

- compromise between single and complete link

☺ strengths

- less susceptible to noise and outliers

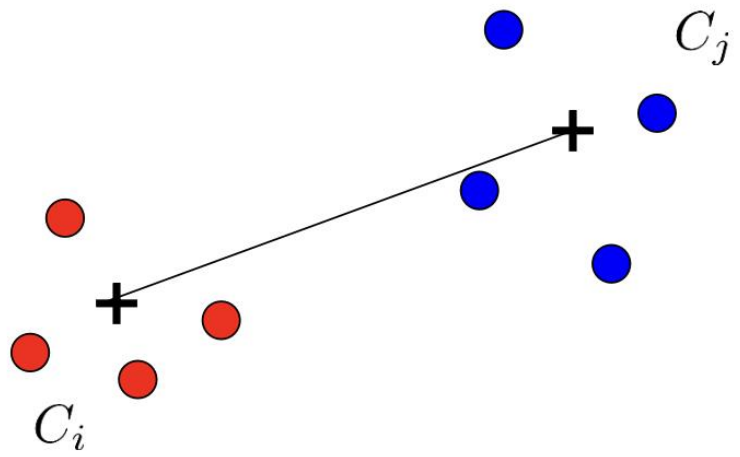
☹ limitations

- may cause elongated clusters to split and for portions of neighboring elongated clusters to merge

4. Inter-Cluster Distances: Centroid Distance

centroid distance between two clusters: distance between two cluster centroids, i.e.,

$$d_{\mathrm{C}}(C_i, C_j) = d\left(\frac{\sum_{x \in C_i} x}{|C_i|}, \frac{\sum_{x \in C_j} x}{|C_j|}\right)$$



5. Inter-Cluster Distances: Ward's Distance

Ward's distance between clusters C_i and C_j:

increase in sum of squared errors when C_i and C_j are merged i.e.,

$$d_W(C_i, C_j) = \sum_{x \in C_{ij}} d^2(x - r_{ij}) - \sum_{x \in C_i} d^2(x - r_i) - \sum_{x \in C_j} d^2(x - r_j)$$

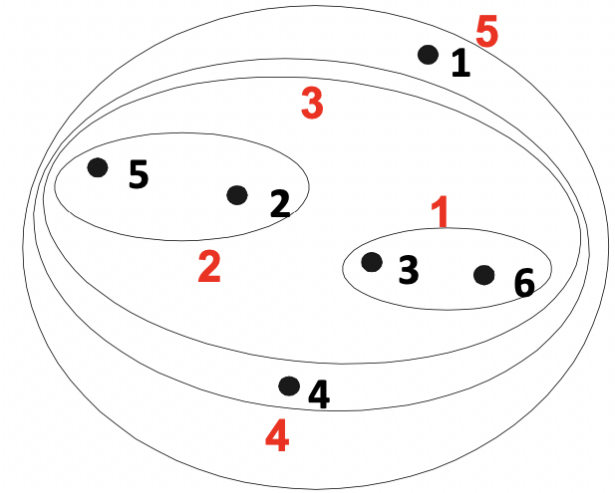
where

- r_i: centroid of C_i
- r_j: centroid of C_j
- r_{ij}: centroid of $C_{ij} = C_i \cup C_j$

Ward's distance for clusters

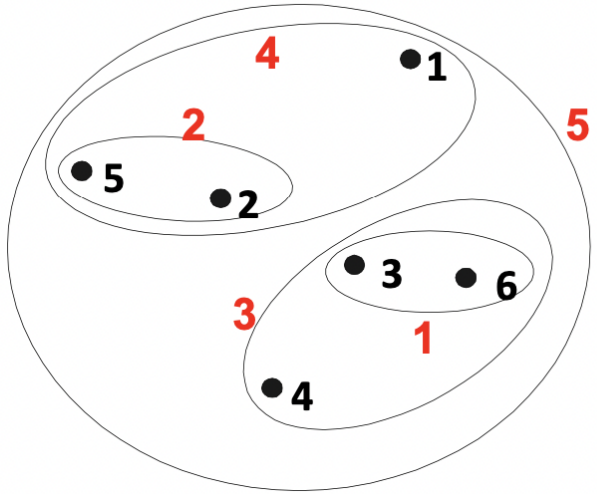
- similar to group average and centroid distance
- less susceptible to noise and outliers
- hierarchical analogue of k-means
  - can be used to initialize k-means

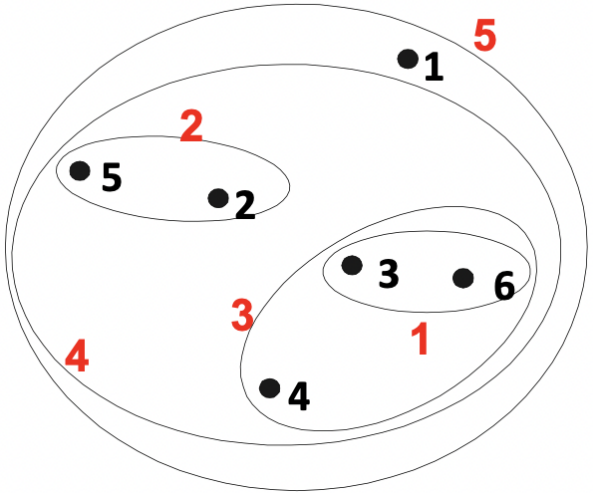
Hierarchical Clustering: Comparison



single-link

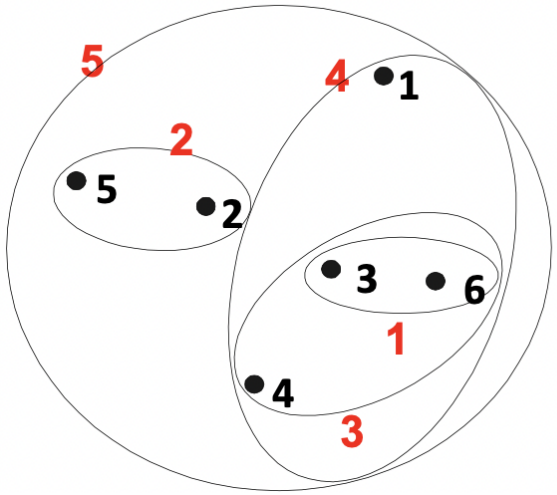
complete-link





average-link

Ward's



Inter-Cluster Distances

general case: various other inter-cluster distance definitions can be used

general agglomerative clustering algorithm: iteratively update a distance matrix

- for distance function d on the instances and inter-cluster distance f, start the agglomerative clustering with the distance matrix

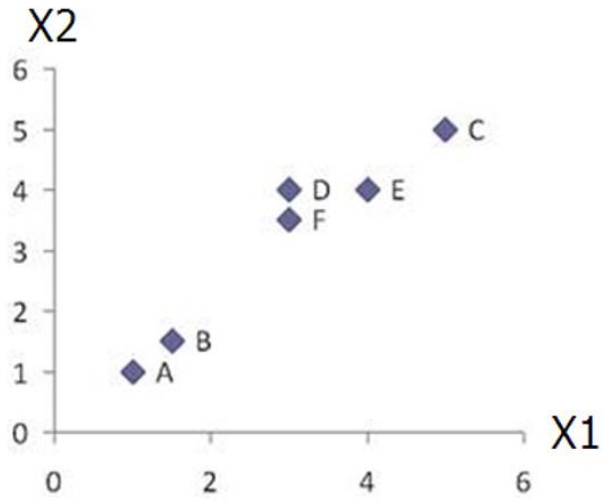
$$D = (d(x_i, x_j))_{1 \leq i \leq, j \leq n}$$

  - set of instances: $I = \{x_1, \ldots, x_n\}$
  - upper triangular, as distances are symmetric

- merging clusters C_i and C_j: merge the corresponding rows and columns in the matrix and update the values according to f

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

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

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

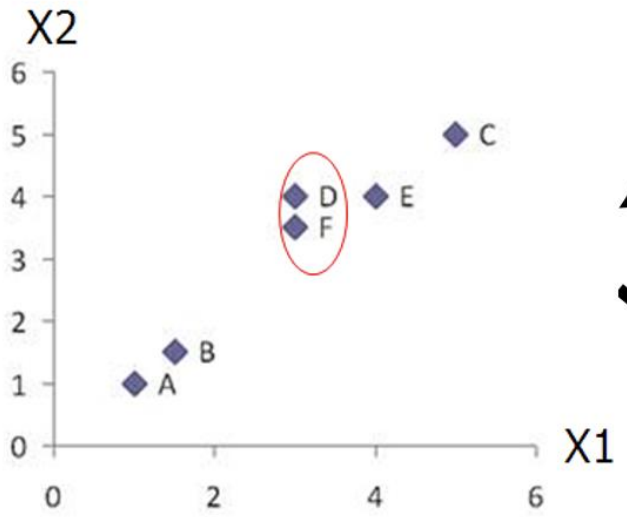
Euclidean distance

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

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

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

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

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

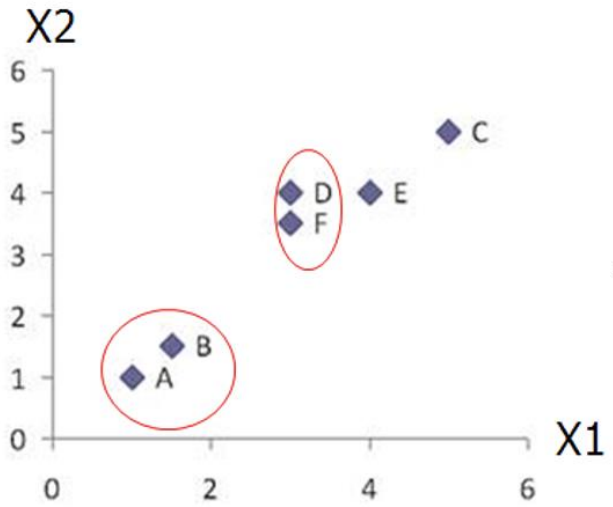
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

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

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

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

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

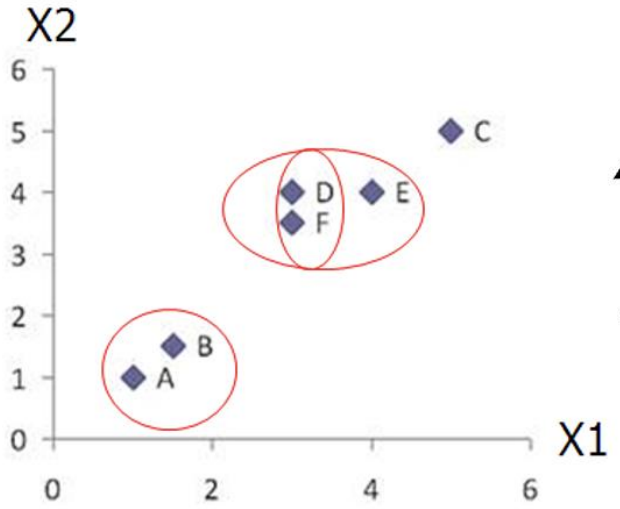
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

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)

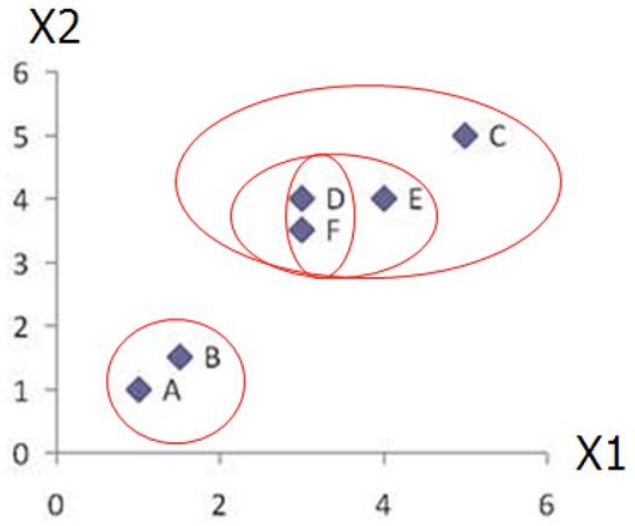
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

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

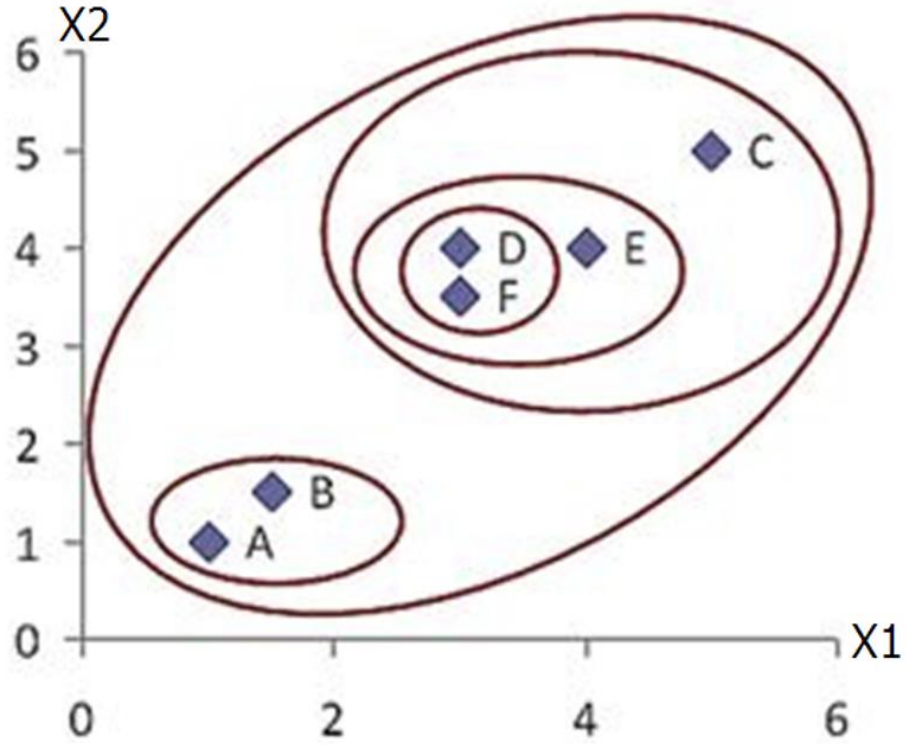
Min Distance (Single Linkage)

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

Example

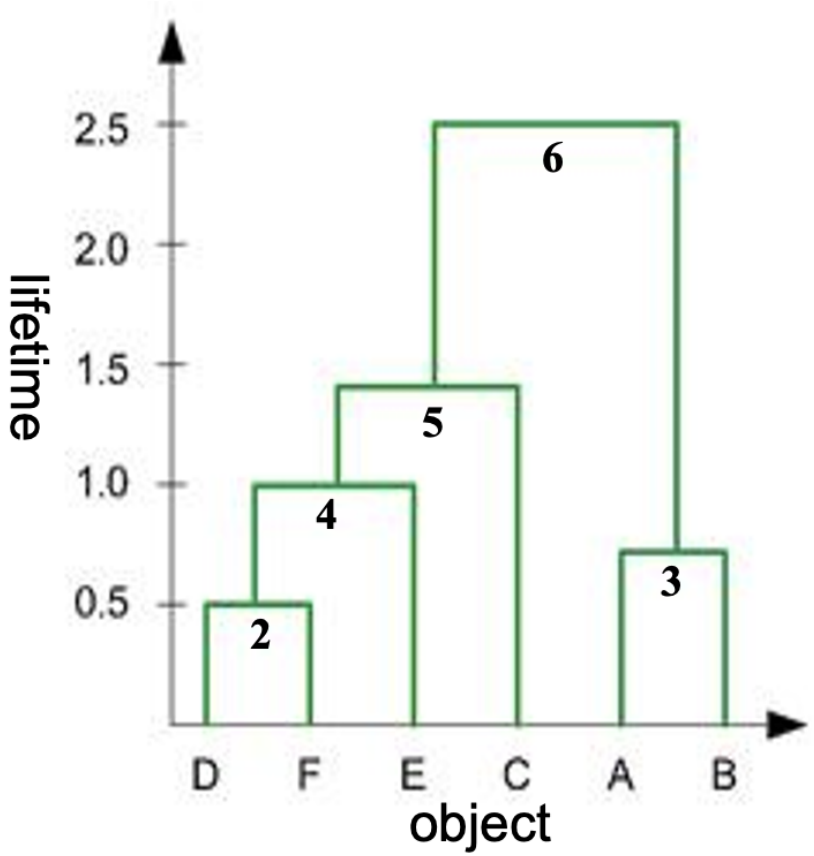
- Final result (meeting termination condition)

	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

Hierarchical Clustering: Time and Space Requirements

for a dataset I consisting of n instances (points)

- $O(n^2)$ **space**
  - it requires storing the distance matrix
- $O(n^3)$ **time** in most of the cases
  - complexity can be reduced to $O(n^2 \log n)$ time for some approaches by using appropriate data structures

Discussion

- weighted/unweighted:
  - **unweighted:** each instance counted equally
  - **weighted:** each cluster counted equally, i.e. instances weighted by cluster size
- only needs distance matrix
  - can work with specialized distances
- user can and has to pick granularity
- nice visualization (for small datasets)
- not very suitable for very large instance sets
- shapes of clusters very much depend on chosen method, no clear cluster "model" as in K-means