Cluster analysis

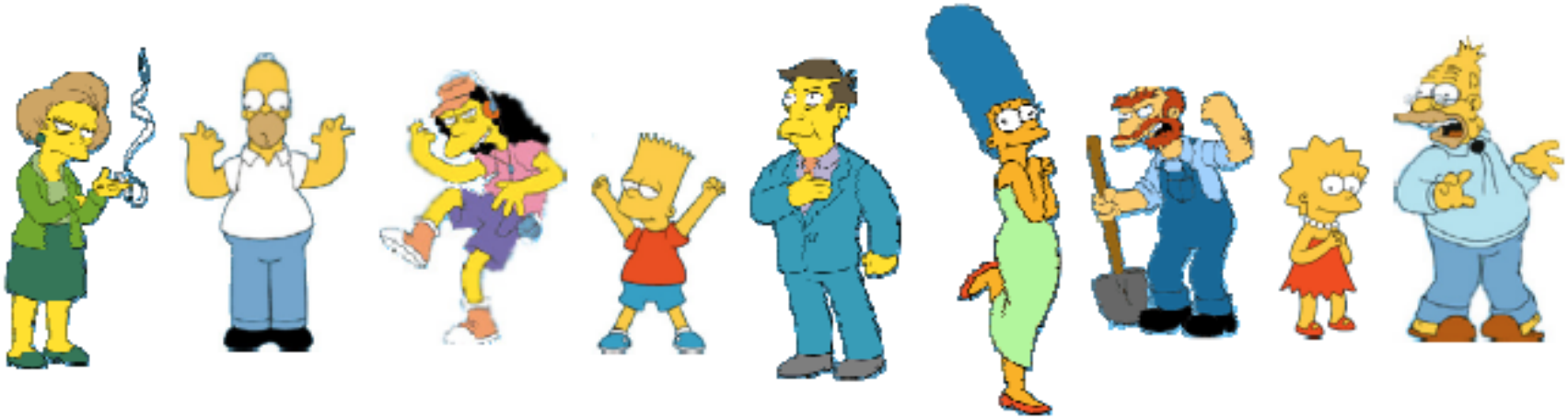
(unsupervised learning)

...just to relax a bit..

Cluster analysis (CA)

- CA is a collection of statistical methods that can be used to **assign units to groups**
- Group members share some characteristic it is hoped that the resultant classification will provide some insight into a research topic
- Some possible applications of clustering
  1. data reduction – reduce data that are homogeneous (similar)
  2. find “natural clusters” and describe their unknown properties
  3. find useful and suitable groupings
  4. find unusual data objects (i.e. outlier detection)

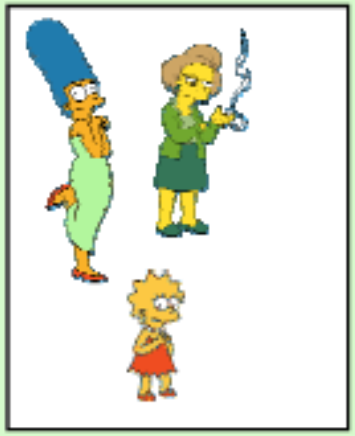
What is a natural grouping among these objects?



Clustering is subjective

Simpson's Family

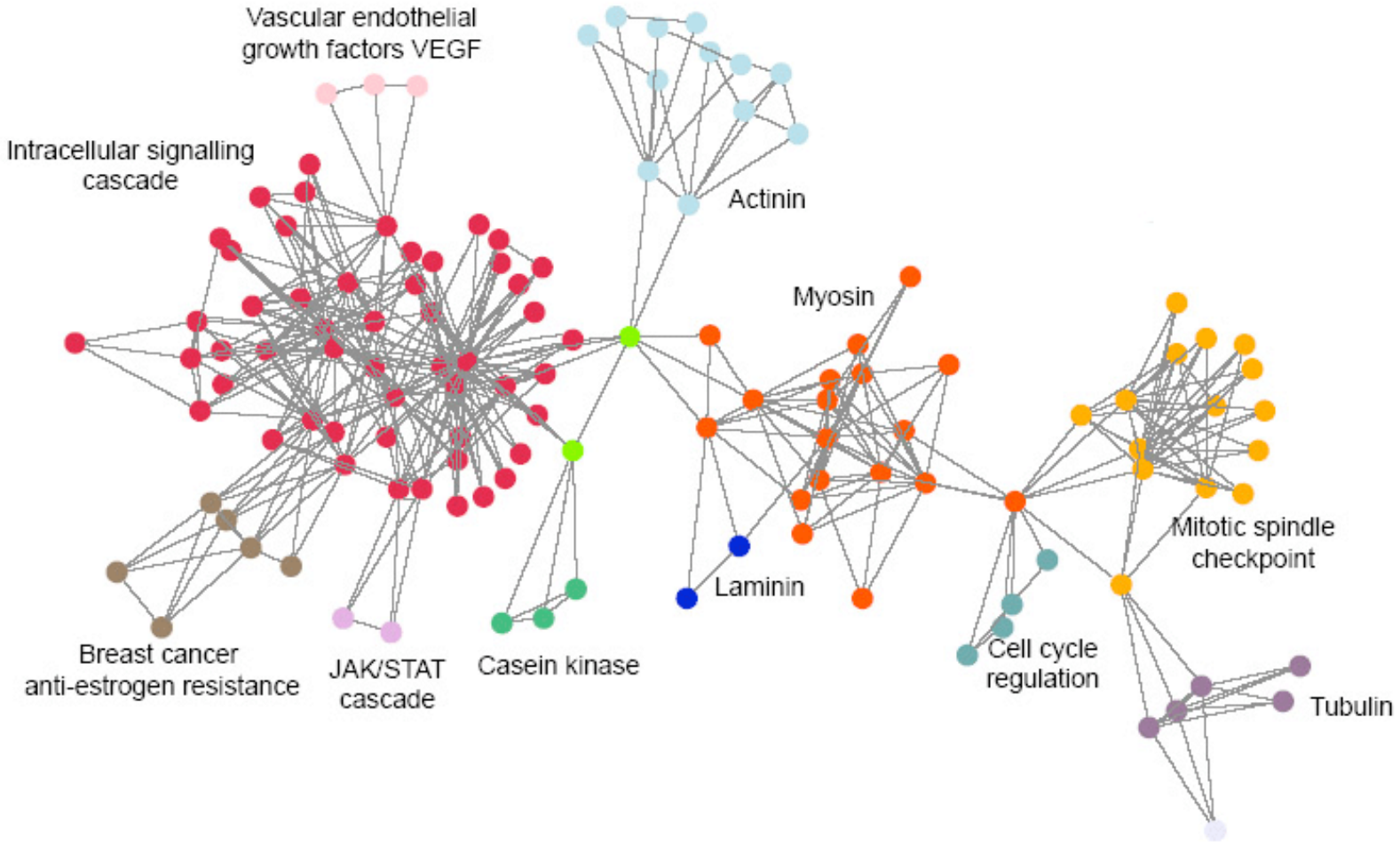
School Employees



Females

Males

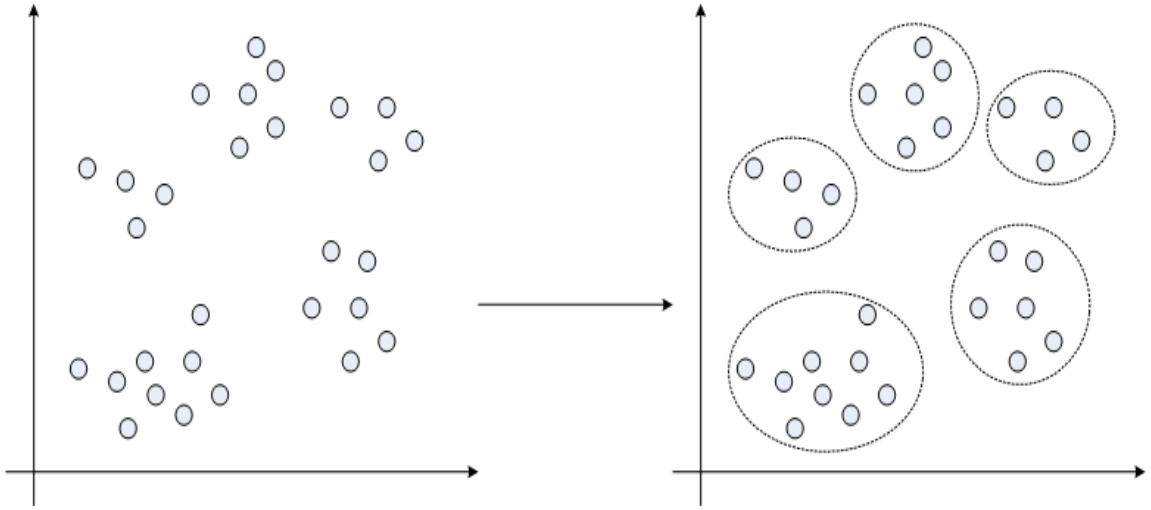
Example



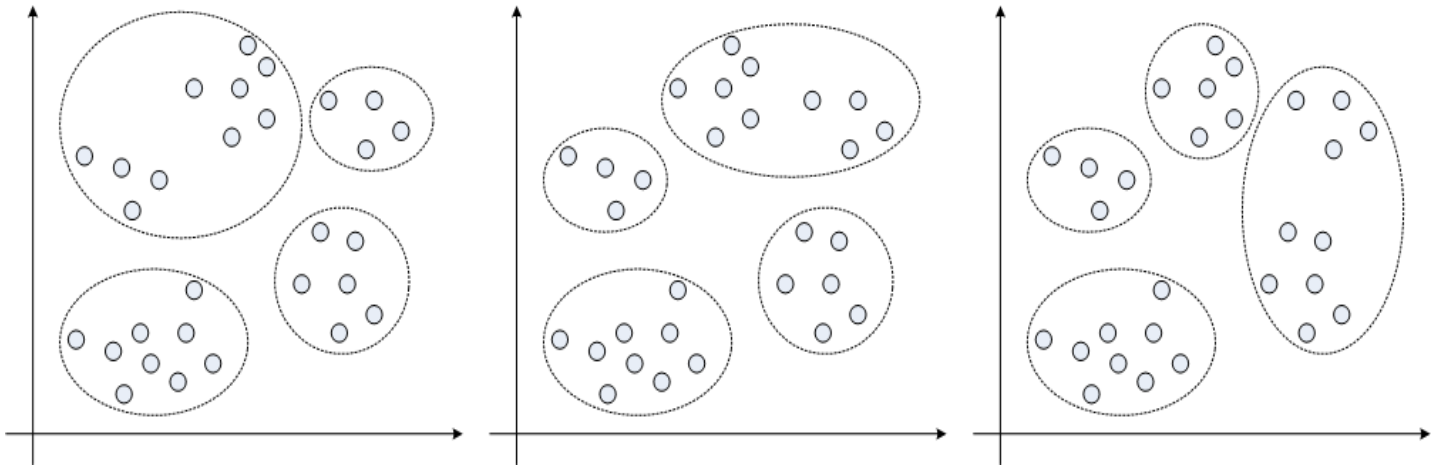
- Examples of communities of interacting proteins. The communities were automatically identified by cluster analysis.

Clustering

- Example: using distance based clustering



- This was easy but how if you had to create 4 clusters? Which of the possibilities below is correct?



Classification via clustering

- Our aim is to find a way of grouping statistical units (or variables) in a data set in such a way that:
  - Units are **similar within** groups and **dissimilar among** groups

We'll focus on units

- Unfortunately, there is no absolute "best" criterion for clustering

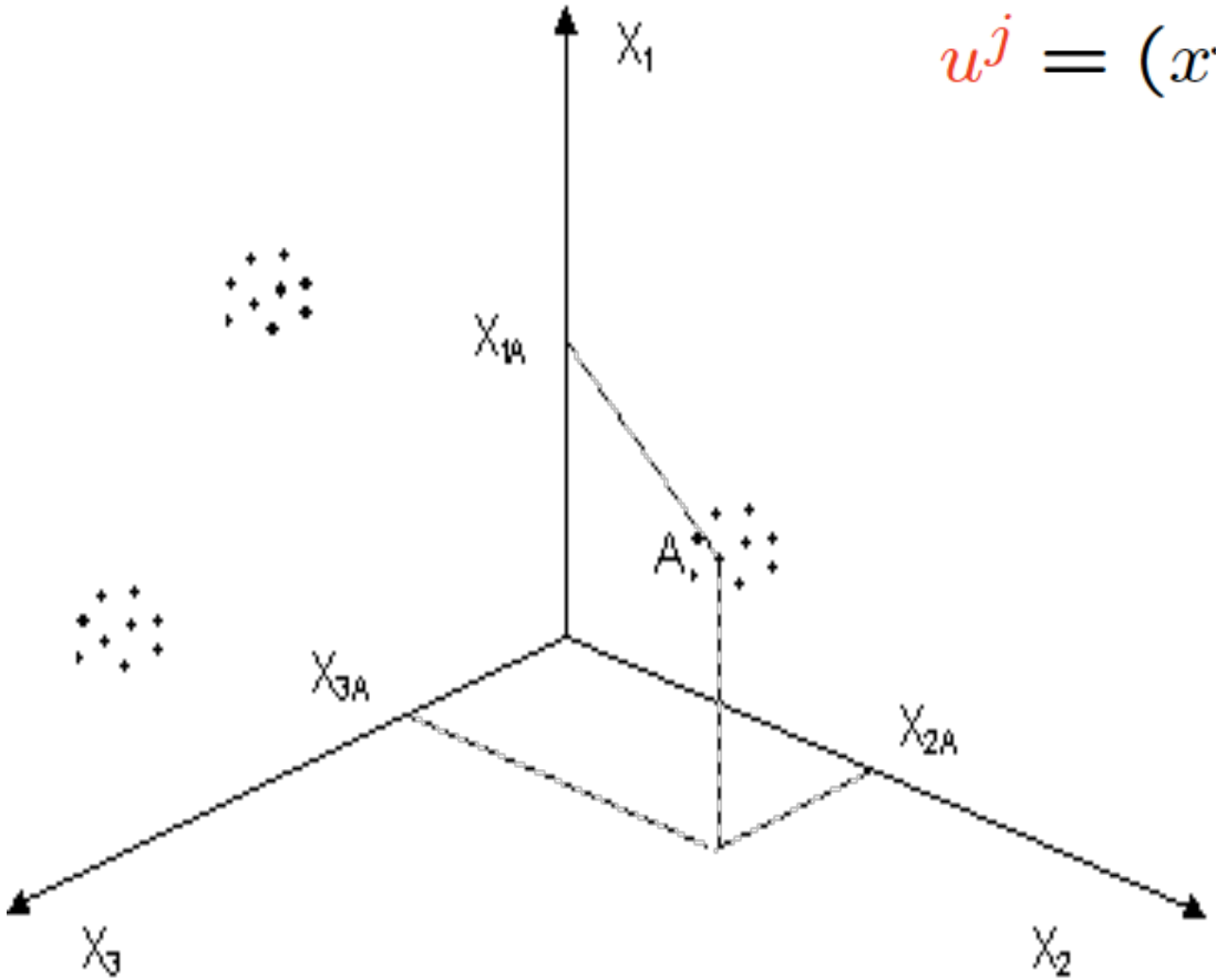
- **Essentials**:
  - Defining a **metric** to evaluate distances
  - Defining an **algorithm** to form groups

Similarity/distance measures

- p-dimensional space

$$u^i = (x_1^i, \ldots, x_p^i)$$

$$u^j = (x_1^j, \ldots, x_p^j)$$



Metric

- A metric on a set X is a **function** $d : X \times X \to \mathbb{R}$

  that satisfies the following properties

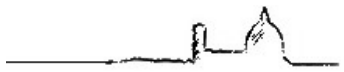
$\forall x, y, z \in X$

distance
- non-negativity: $d(x, y) \geq 0$
- identity: $d(x, y) = 0$ iff $x = y$
- symmetry: $d(x, y) = d(y, x)$

- triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$

- A metric is sufficient but not necessary for cluster analysis
- Replacing the triangle inequality with

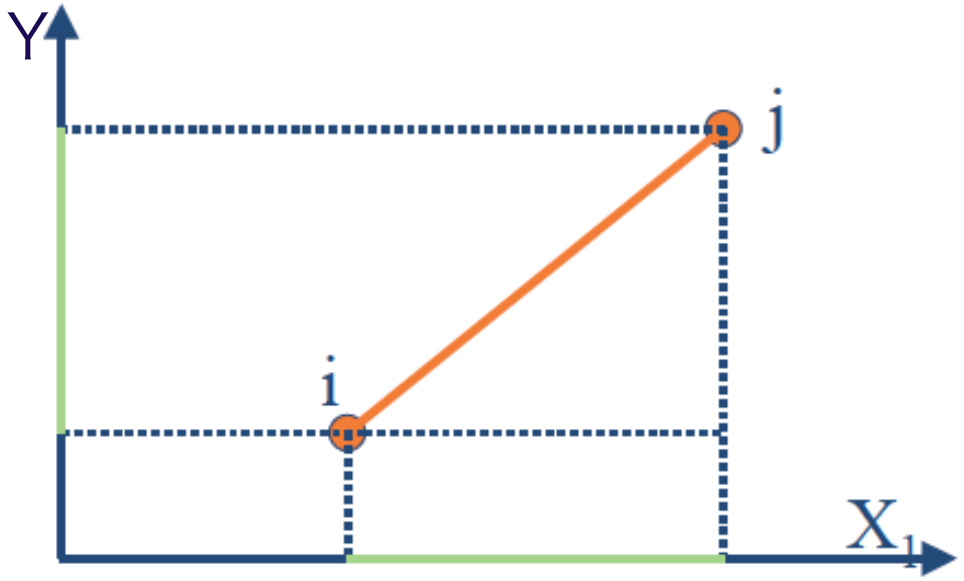
$d(x,z) = \max[\, d(x,y), d(y,z)]$ we have an ultrametric



Frequently adopted distances (quantitative vars)

- **Euclidean distance**

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



- Karl Pearson distance

$$d(x,y) = \sqrt{(x-y)'S^{-1}(x-y)}$$

Euclidean distance with standardized variables

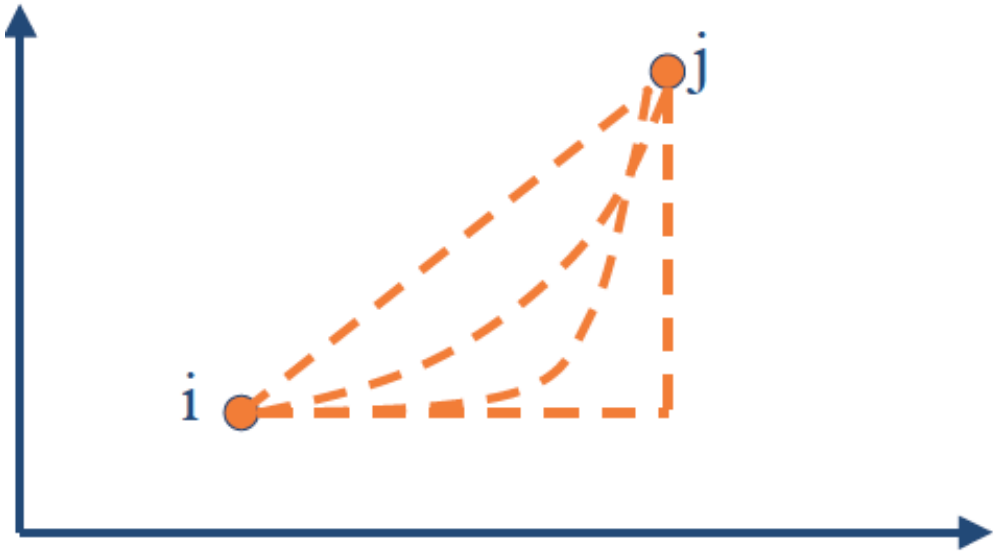
Frequently adopted distances (quant. vars)

- Minkowsky distance and Manhattan distance

$$d(x,y) = \left[\sum_{i=1}^{p} | x_i - y_i |^m \right]^{1/m}$$

m=1

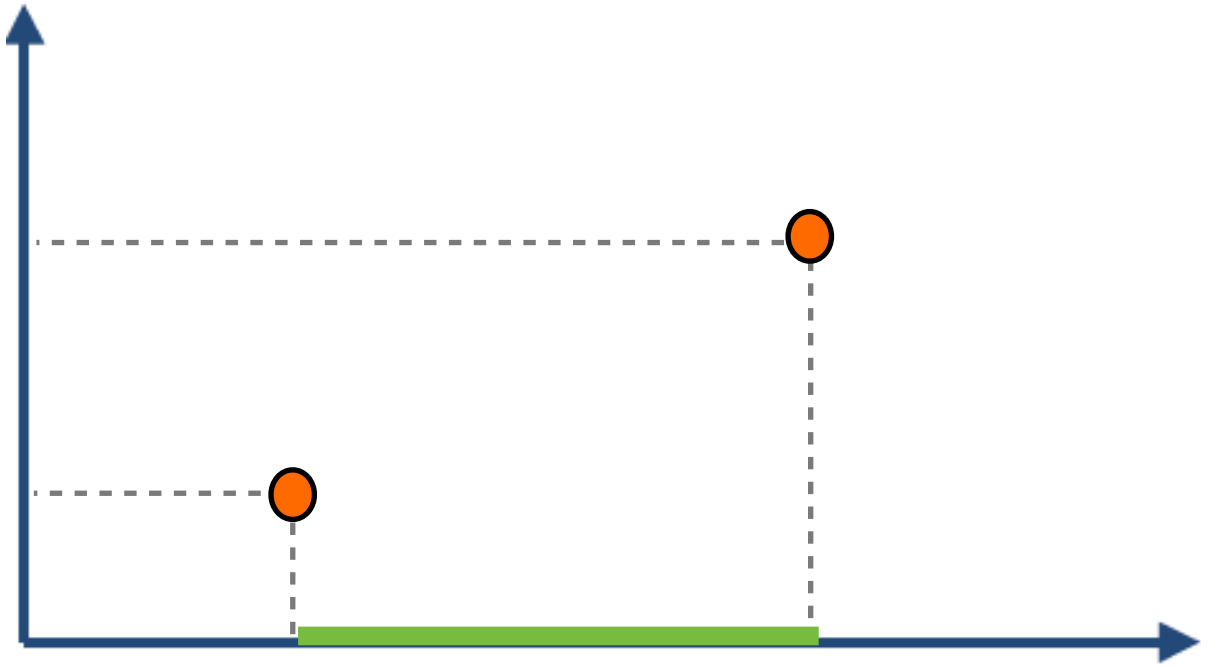
$$d(x,y) = \sum_{i=1}^{p} | x_i - y_i |$$



Supremum distance

- Chebychev distance (supremum)

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



Canberra metric

- Canberra metric for non-negative vars

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

- General version

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

Let's standardize?

- The scale of measurement of the variables is an important consideration when using the Euclidean distance measure. Changing the scale can affect the relative distances among the items

- To counter this problem, each variable could be standardized in the usual way (by subtracting the mean and dividing by the standard deviation of the variable)

- As a **drawback**, standardized units are more similar and groups are nearer

Frequently adopted similarities (qualitative vars)

- Simple matching coefficient

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

statistical units

x \ y	0	1
0	a	b
1	c	d

n. vars=1 for both x and y

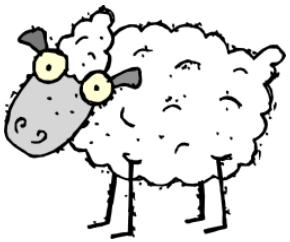
- Jacard's coefficient -> for asymmetric characteristics

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

- The square of the Euclidean distance counts the total number of mismatches

- Distance = 1- similarity

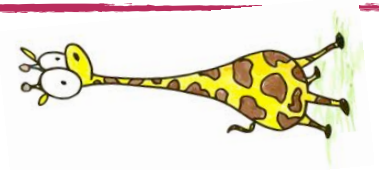
Example: lion, giraffe, human, sheep

- 4 units: (L)  (G)  (H)  (S) 

- 7 binary vars: (1) has a tail, (2) is a wild animal, (3) is a farm animal, (4) eats other animals, (5) has long neck, (6) walks on four legs, (7) provides clothing material without being killed

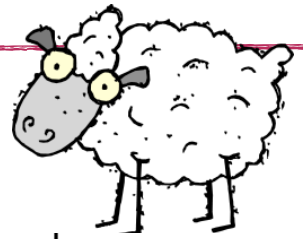
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
lion	1	1	0	1	0	1	0
giraffe	1	1	0	0	1	1	0
human	0	0	0	1	0	0	0
sheep	1	0	1	0	0	1	1

Example: Pairwise association tables



	1	0
1	3	1
0	1	2

	1	0
1	1	3
0	0	3



	1	0
1	2	2
0	2	1

	1	0
1	0	4
0	1	2

	1	0
1	2	2
0	2	1

	1	0
1	0	1
0	4	2

Example: similarities and distances

Similarities

Simple matching

	L	G	H	S
L	1	5/7	4/7	3/7
G		1	2/7	3/7
H			1	2/7
S				1

Jacard's

	L	G	H	S
L	1	3/5	1/4	2/6
G		1	0/5	2/6
H			1	0/5
S				1

Distances

	L	G	H	S
L	0	0.29	0.43	0.57
G		0	0.71	0.57
H			0	0.71
S				0

	L	G	H	S
L	0	0.40	0.75	0.67
G		0	1	0.67
H			0	1
S				0

Multi-categorical variables

	Feature	Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion
Is	small	1	1	1	1	1	1	0	0	0	0	1	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1
	run	0	0	0	0	0	0	0	0	1	1	0	1	1
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0

Frequently adopted distances (mixed vars)

- Gower's general coefficient of similarity (1966)

$$s(x,y) = \frac{\sum_{i=1}^{p} w_i \; s_i(x_i, y_i)}{\sum_{i=1}^{p} w_i} \qquad (x_1, \ldots, x_p), (y_1, \ldots, y_p) \in X_p$$

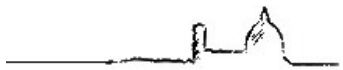
  - Nominal variables

$$s_i(x_i, y_i) = \begin{cases} 1 & \text{if} \quad x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

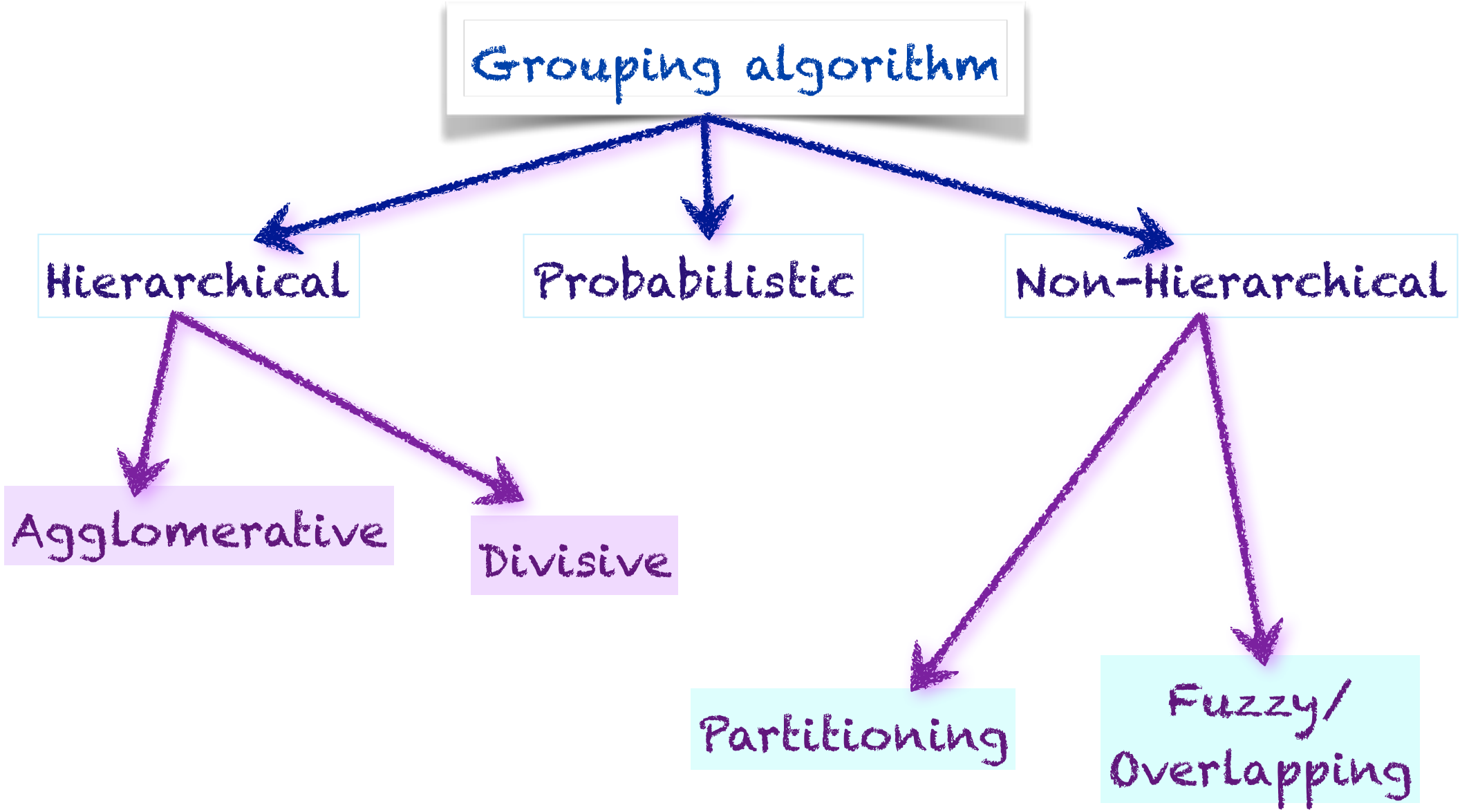
  - Quantitative variables:

$$s_i(x_i, y_i) = 1 - \text{normalized distance}$$

  - Ordinal variables: transform in ranks and use a quantitative measure

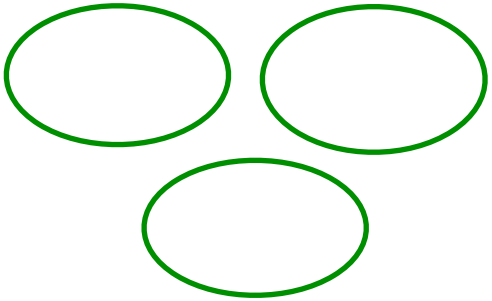


What's up once we got a distance matrix?

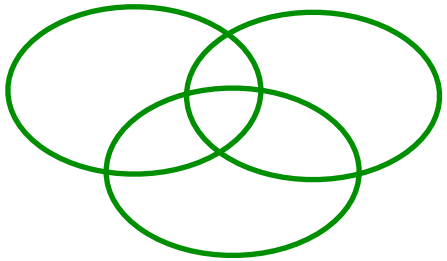


Grouping algorithms

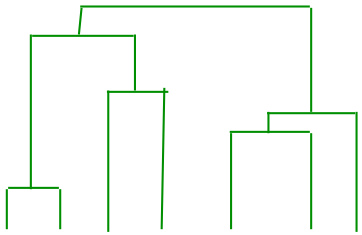
Non overlapping



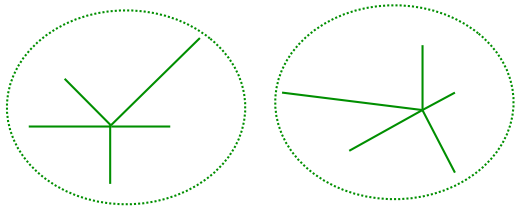
Overlapping



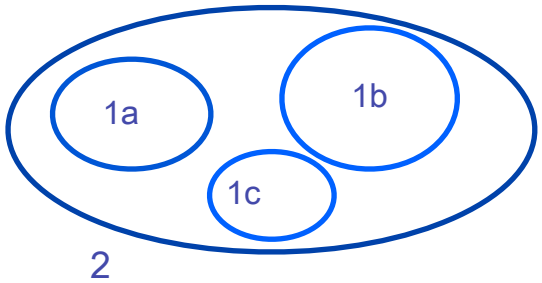
Hierarchical



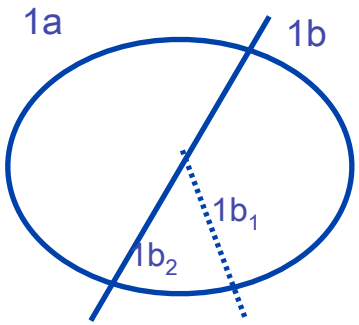
Non-hierarchical



Agglomerative

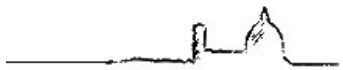


Divisive



Partitioning vs Hierarchical

- Non-hierarchical
  1. Advantage: Provides clusters that satisfy some optimality criterion (approximately)
  2. Disadvantages: Need initial K, long computation time
- Hierarchical
  3. Advantage: Fast computation (agglomerative)
  4. Disadvantages: Rigid, cannot correct later for erroneous decisions made earlier
- Probabilistic
  5. Uses probability distribution measures to create the clusters
  6. Eg: Gaussian mixture model clustering



Hierarchical agglomerative clustering steps

- Given a set of n units to be clustered, hierarchical agglomerative clustering consists of:

1. **Assign** each unit to its own cluster, so that if there are n units, then **n clusters** are composed, each containing just one unit
2. Find the closest distance pair of clusters and **merge** them into a single cluster
3. **Compute** pairwise distances between the new clusters and each of the old clusters
4. **Repeat** steps **2** and **3** until all units are clustered into a single cluster of size n.
5. Draw the **dendogram**, and chose an opportune number of clusters

Distances between groups

- Single linkage clustering (nearest neighbour)
- Complete linkage clustering (further neighbour)
- Centroid
- Average linkage
- Median (weighted pair-group centroid method)
- Ward method
- . . .

Distance measures for cluster

Single Linkage:

min {d(a, b) : a ∈ A, b ∈ B}

The distance between two clusters is the shortest distance from any unit of one cluster to any unit of the other cluster

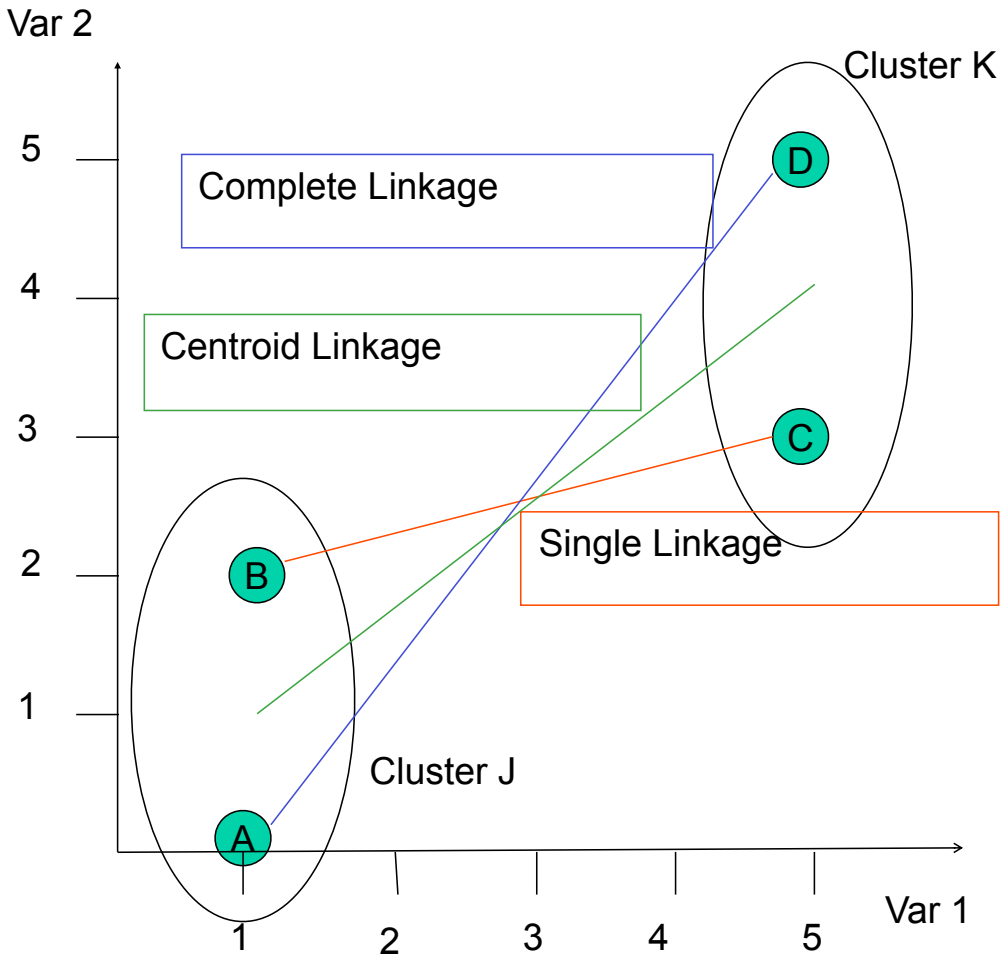
Complete Linkage:

max {d(a, b) : a ∈ A, b ∈ B}

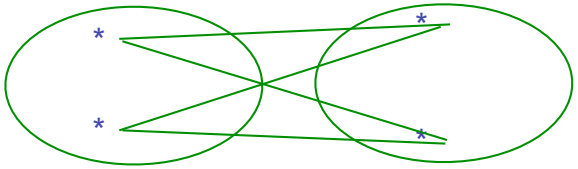
The distance between two clusters is the greatest distance from any unit of one cluster to any unit of the other

Centroid Linkage:

The distance between two clusters is the distance the two cluster centroids

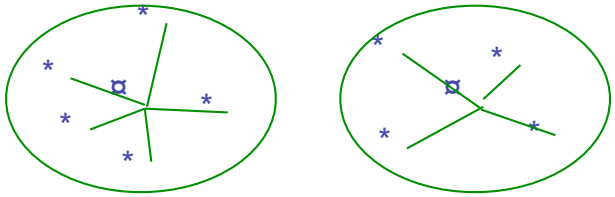


Average and Ward methods



Average Linkage:

The distance between two clusters is the average of the distance between each unit in a group with each unit of the other



Wards method:
Minimization of within-cluster variance

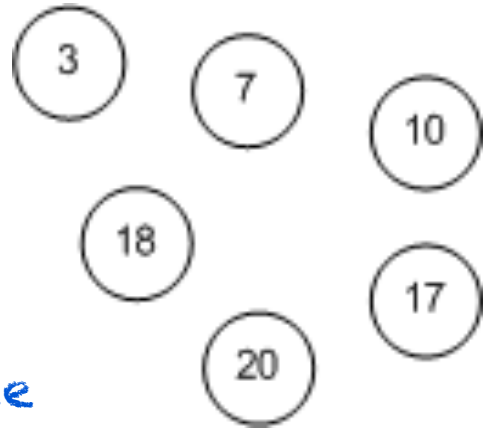
The distance between two clusters (e.g. A and B) consists in how much the sum of squared will increase if they are merged

$$d(A,B) = \sum_{i \in A \cup B} (x_i - \bar{x}_{A \cup B})^2 - \left(\sum_{i \in A} (x_i - \bar{x}_A)^2 + \sum_{i \in B} (x_i - \bar{x}_B)^2 \right)$$

$$= \frac{n_A \, n_B}{n_A + n_B} (\bar{x}_A - \bar{x}_B)^2$$

Example : X = (3, 7, 10, 17, 18, 20)

Step 1 : Assign each unit to a different group



Step 2 : Chose a metric and a linkage

- Manhattan distance
- Single linkage

Step 3 : Construct the distance matrix

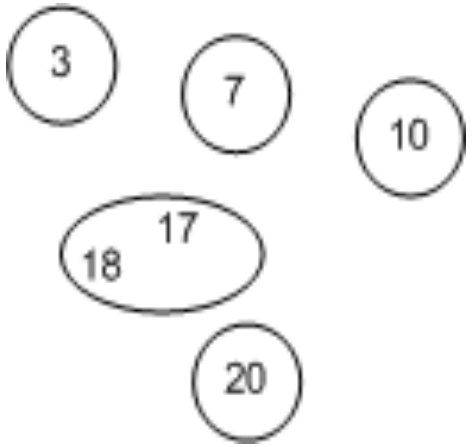
Manhattan distance

	3	7	10	17	18	20
3	0	4	7	14	15	17
7		0	3	10	111	13
10			0	7	8	10
17				0	1	3
18					0	2
20						0

Smallest distance

Example : X = (3, 7, 10, 17, 18, 20)

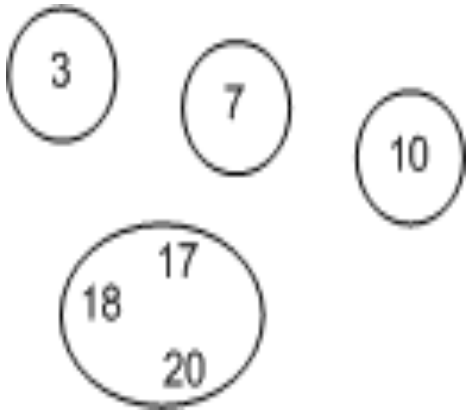
Step 4 : Join nearest groups



Step 5 : Ri-compute distances

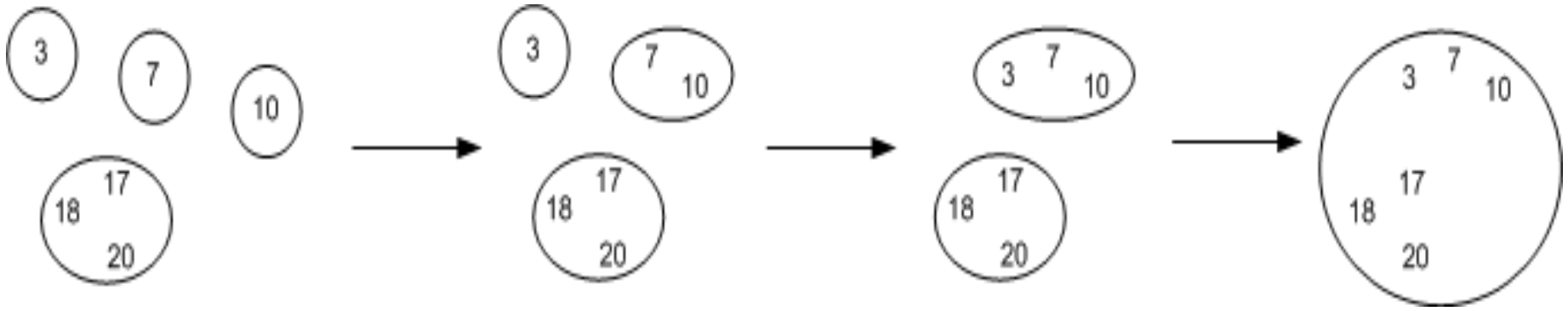
Single linkage

	3	7	10	(17, 18)	20
3	0	4	7	14	17
7		0	3	10	13
10			0	7	10
(17, 18)				0	2
20					0

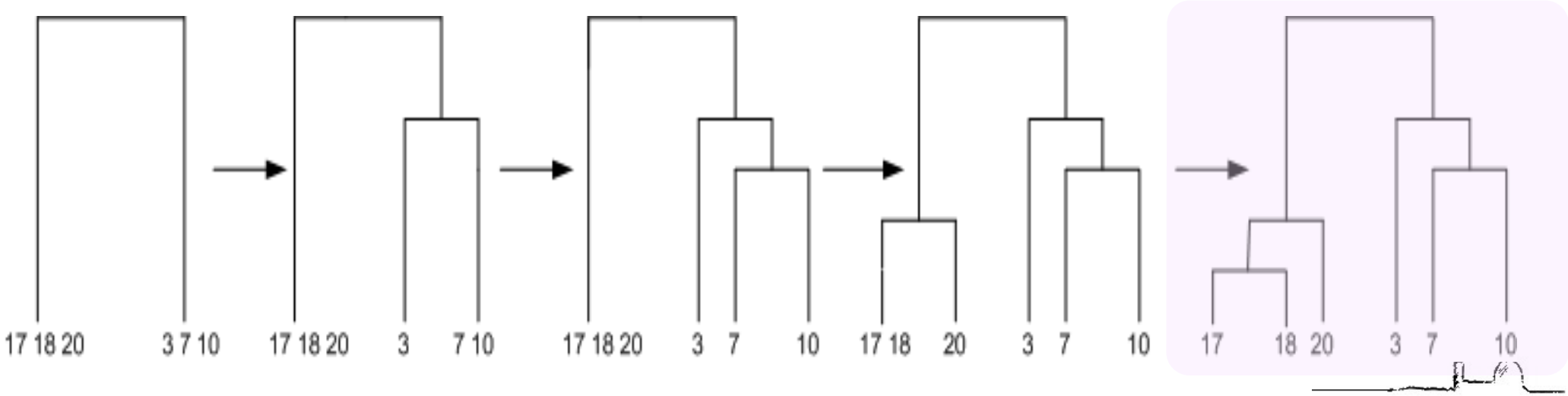


Step 6 : Repeat step 4 and 5 until all units are grouped

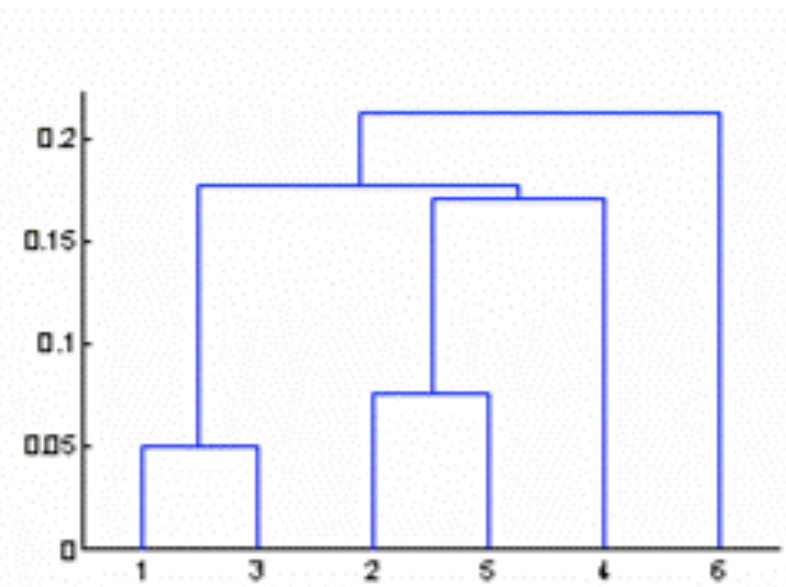
Example : X = (3, 7, 10, 17, 18, 20)

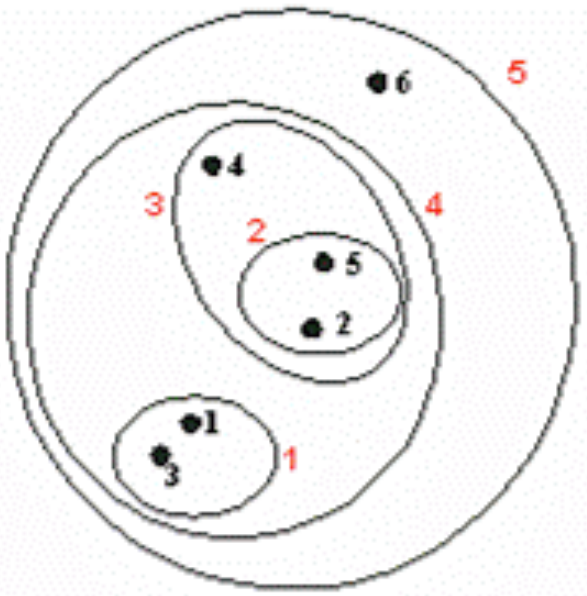


Final Step : Plot the **dendrogram** and chose the "optimal" number of groups



Dendrogram





- Dendrogram : graphical representation of the nested clusters
- A clustering of the data is obtained by cutting the dendrogram at a desired level.
- Each connected component forms a cluster

Dendrogram cutting

- The dendrogram cutting is an **arbitrary** step
- Usually the cut is made where the dendrogram has **jump**, corresponding to the max distance
- The good cut should also bring well cohesive groups
- Some indexes can be used to evaluate the quality of clustering

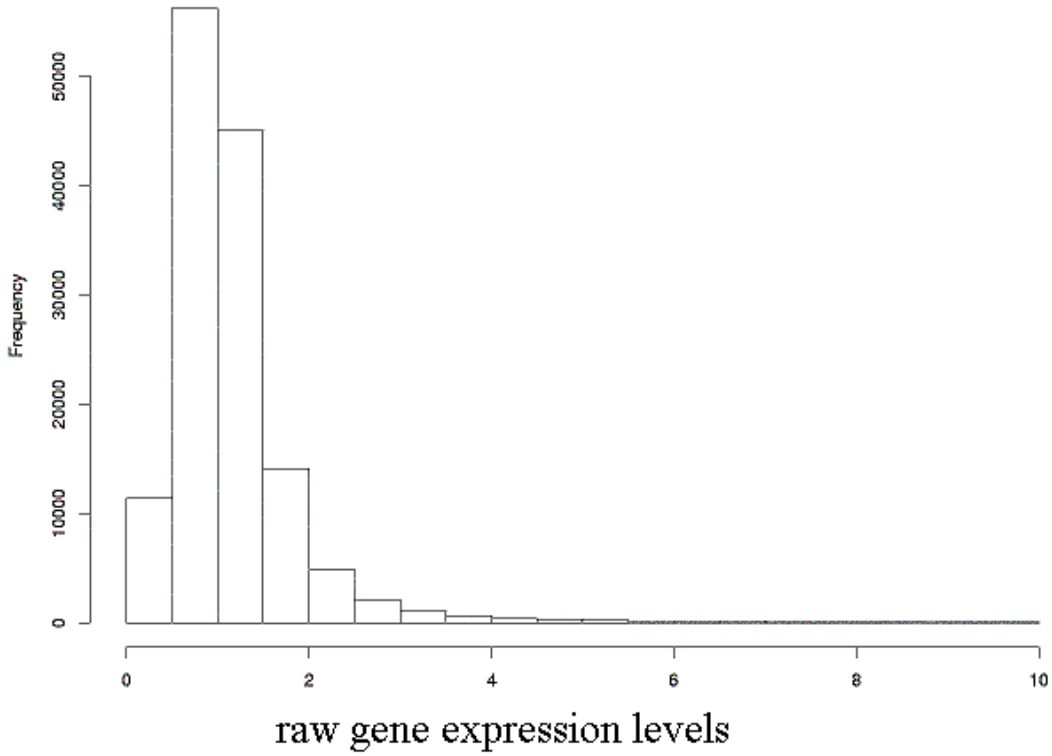
$$cohesion(C_i) = \frac{1}{\sum_{\mathbf{x},\mathbf{y} \in C_i} \left(distance(\mathbf{x},\mathbf{y})\right)}$$

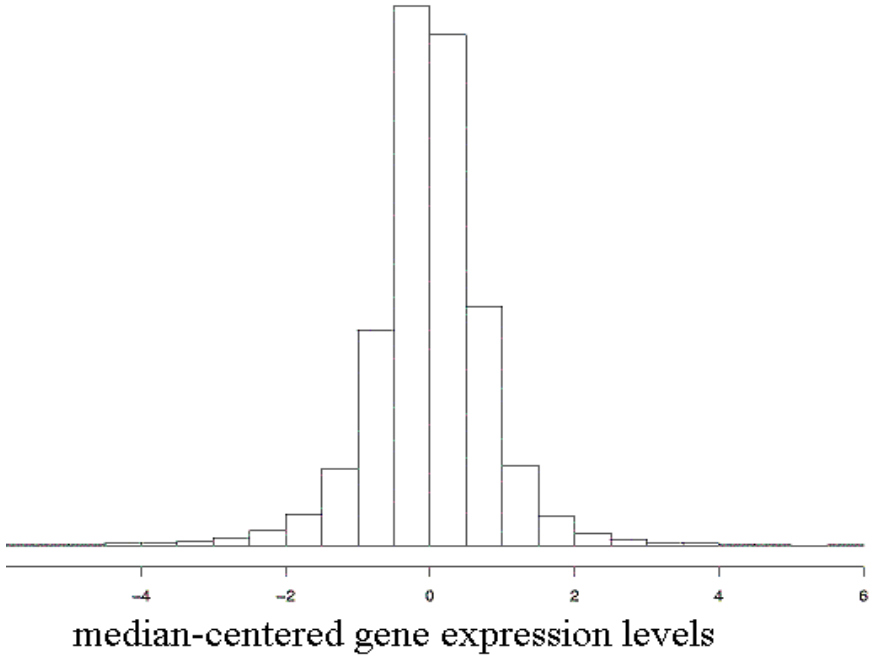
$$separation(C_i, C_j) = \sum_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} \left(distance(\mathbf{x},\mathbf{y})\right)$$

Example: genes and cancer (Bittner & al., 2000)

- It has been proposed (by many) that a cancer taxonomy can be identified from gene expression experiments.
- Data:
  - 31 melanomas (from a variety of tissues/cell lines)
  - 7 controls
  - 8150 cDNAs
  - 6971 unique genes
  - 3613 genes ‘strongly detected’

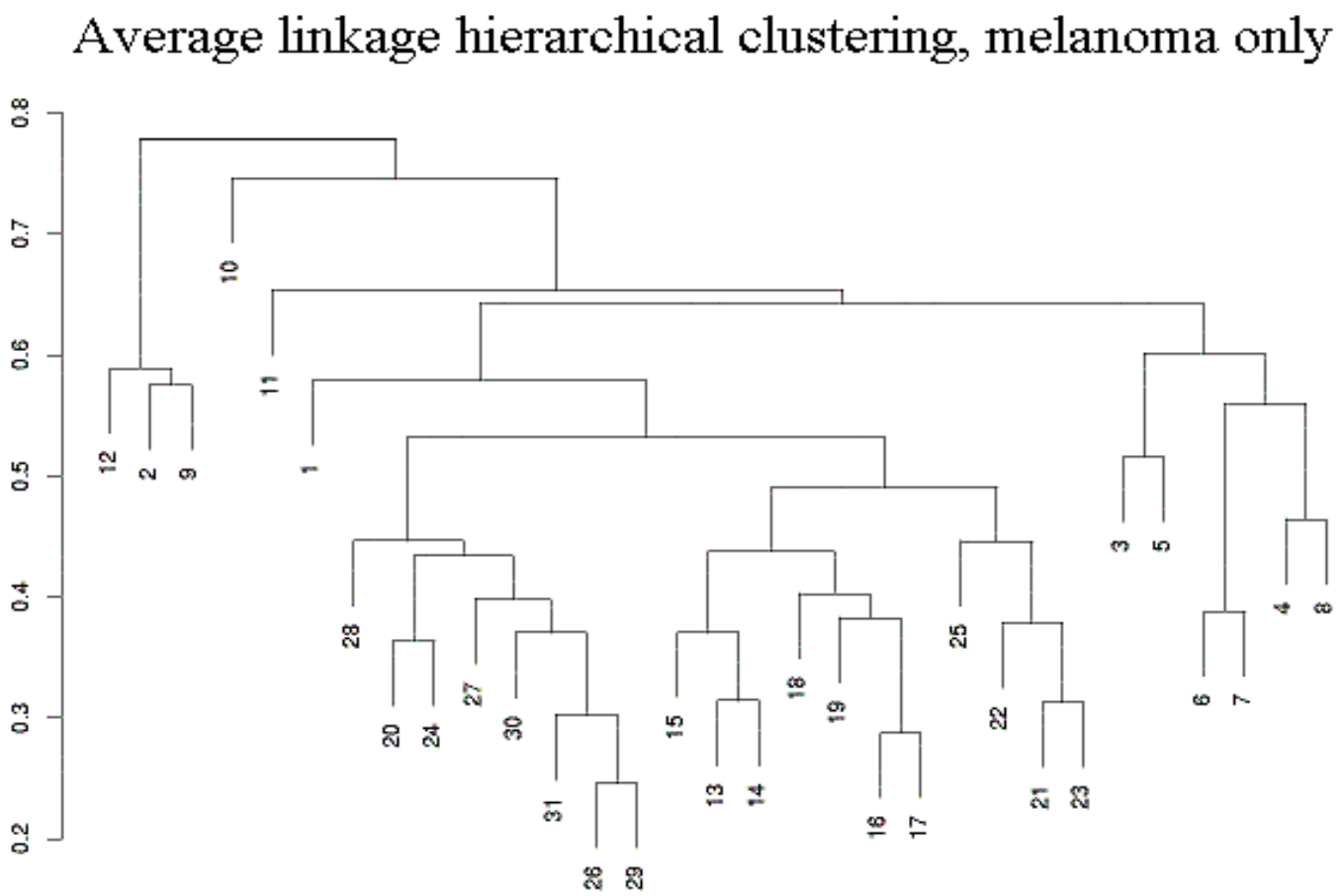
A look to data . . .





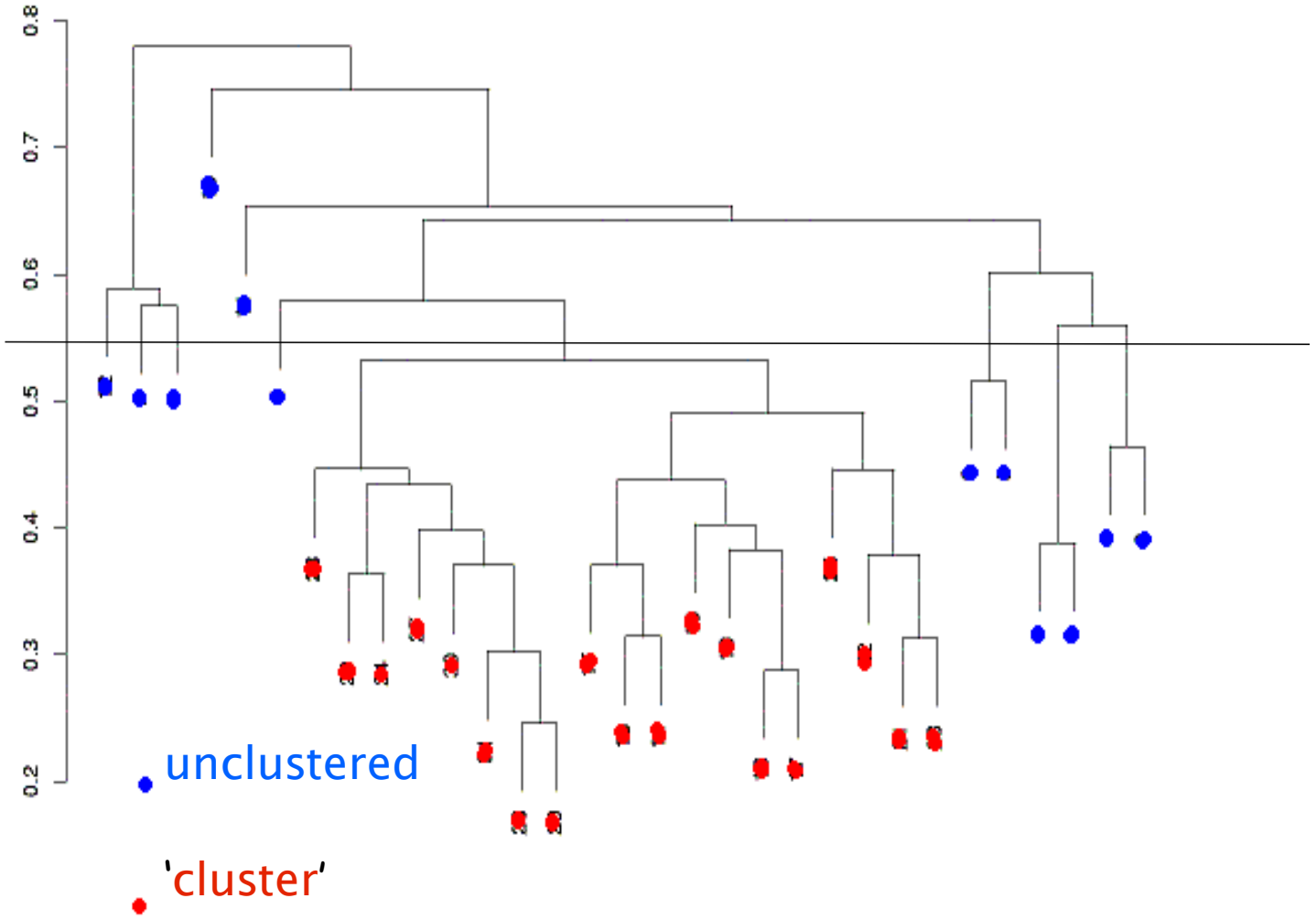
. . . then take the logs . . .

Dendrogram

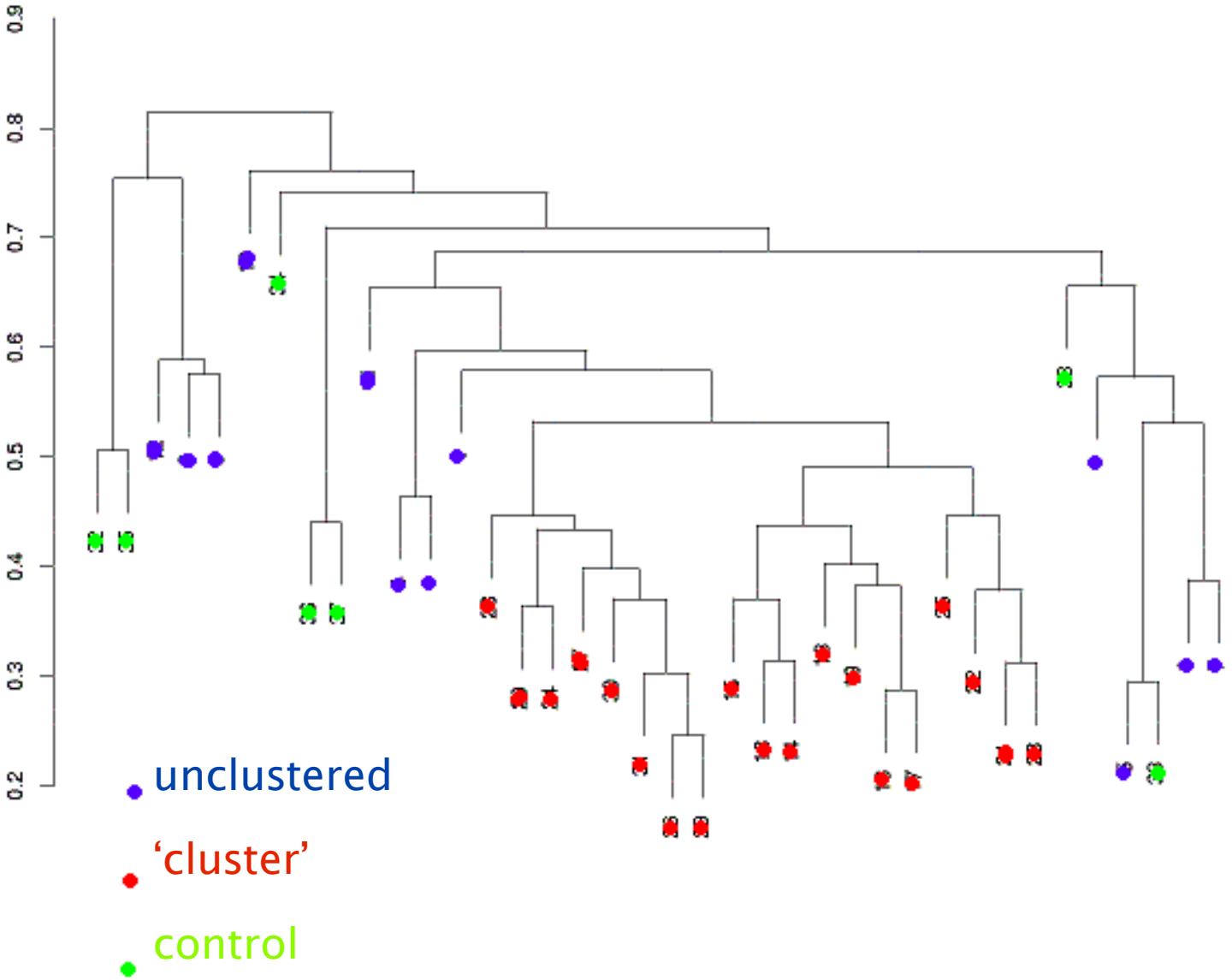

Average linkage hierarchical clustering, melanoma only
0.8
0.7
0.6
0.5
0.4
0.3
0.2
10
11
12
2
9
1
28
20
24
27
30
31
26
29
15
13
14
18
19
16
17
25
22
21
23
3
5
6
7
4
8

Average linkage on melanomas

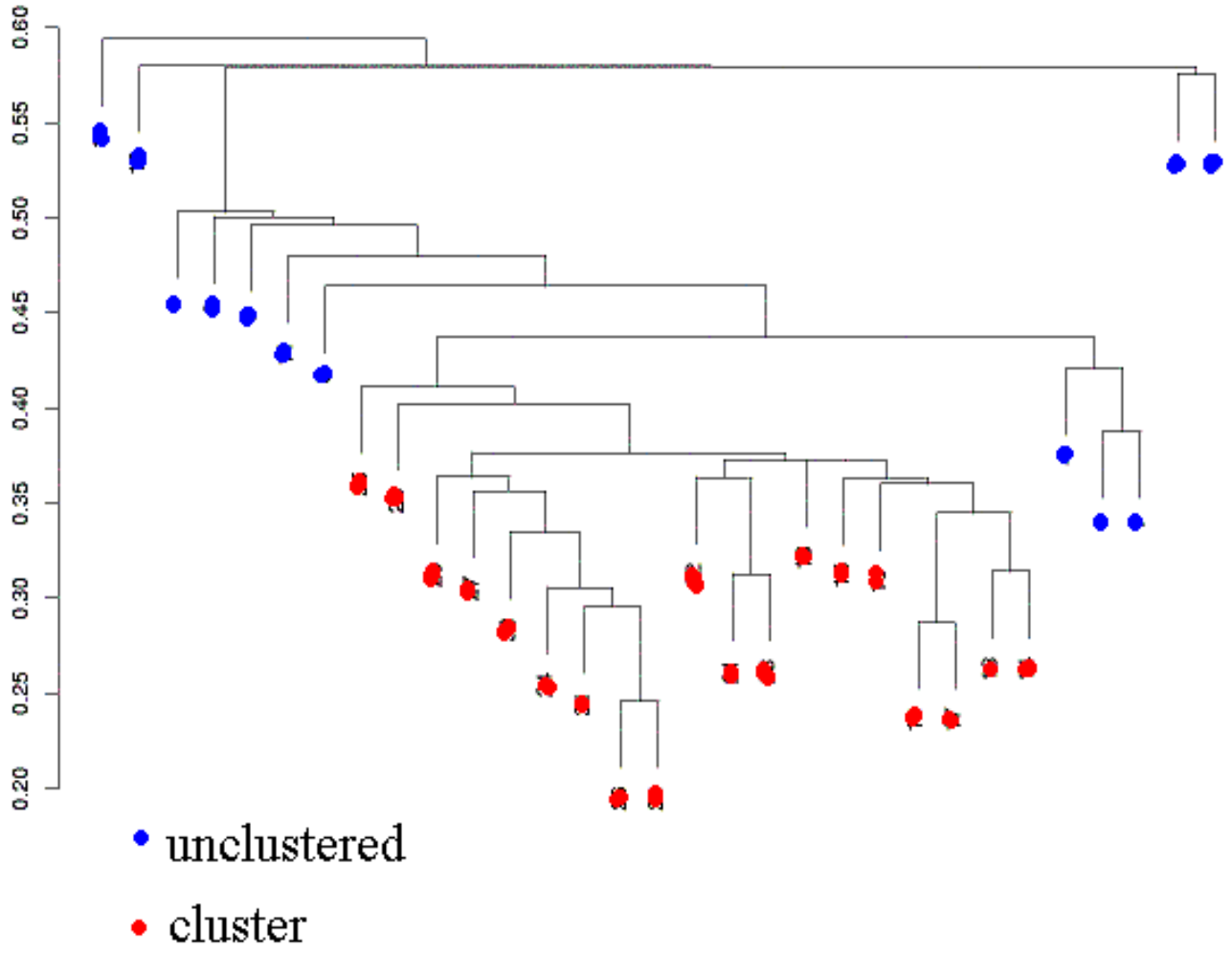
$1-\rho = .54$



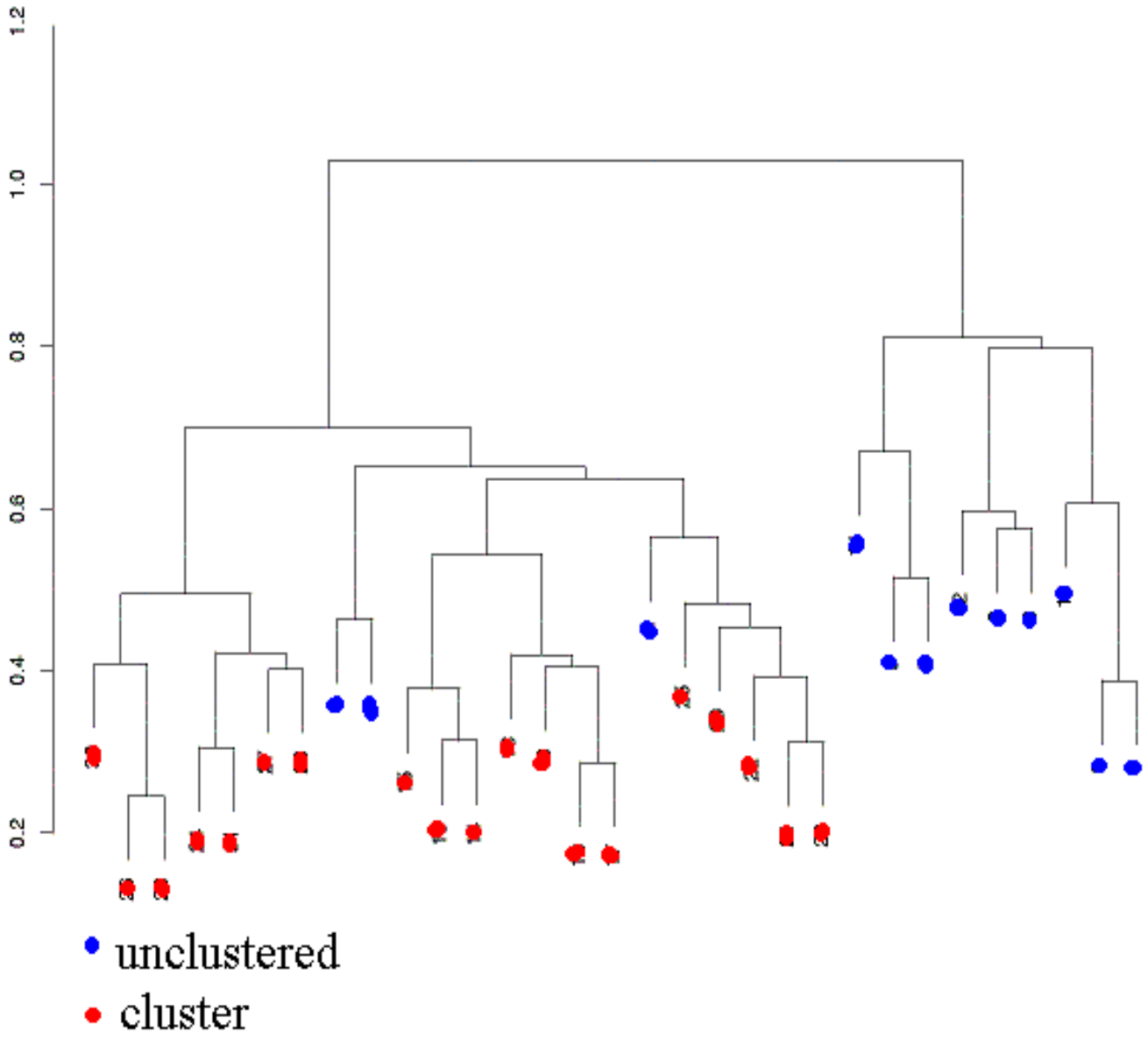
Average linkage, melanoma & controls



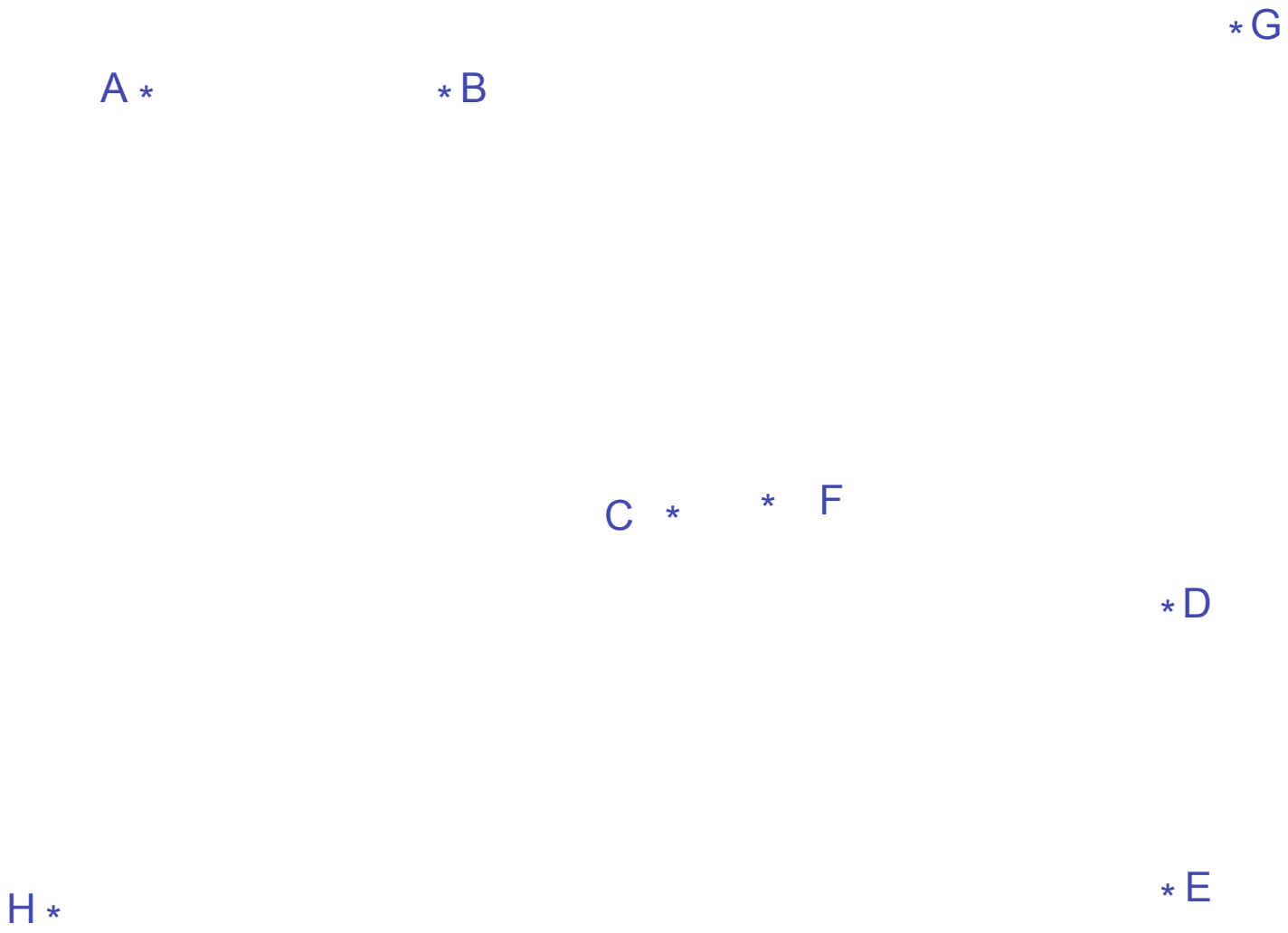
Complete linkage hierarchical clustering



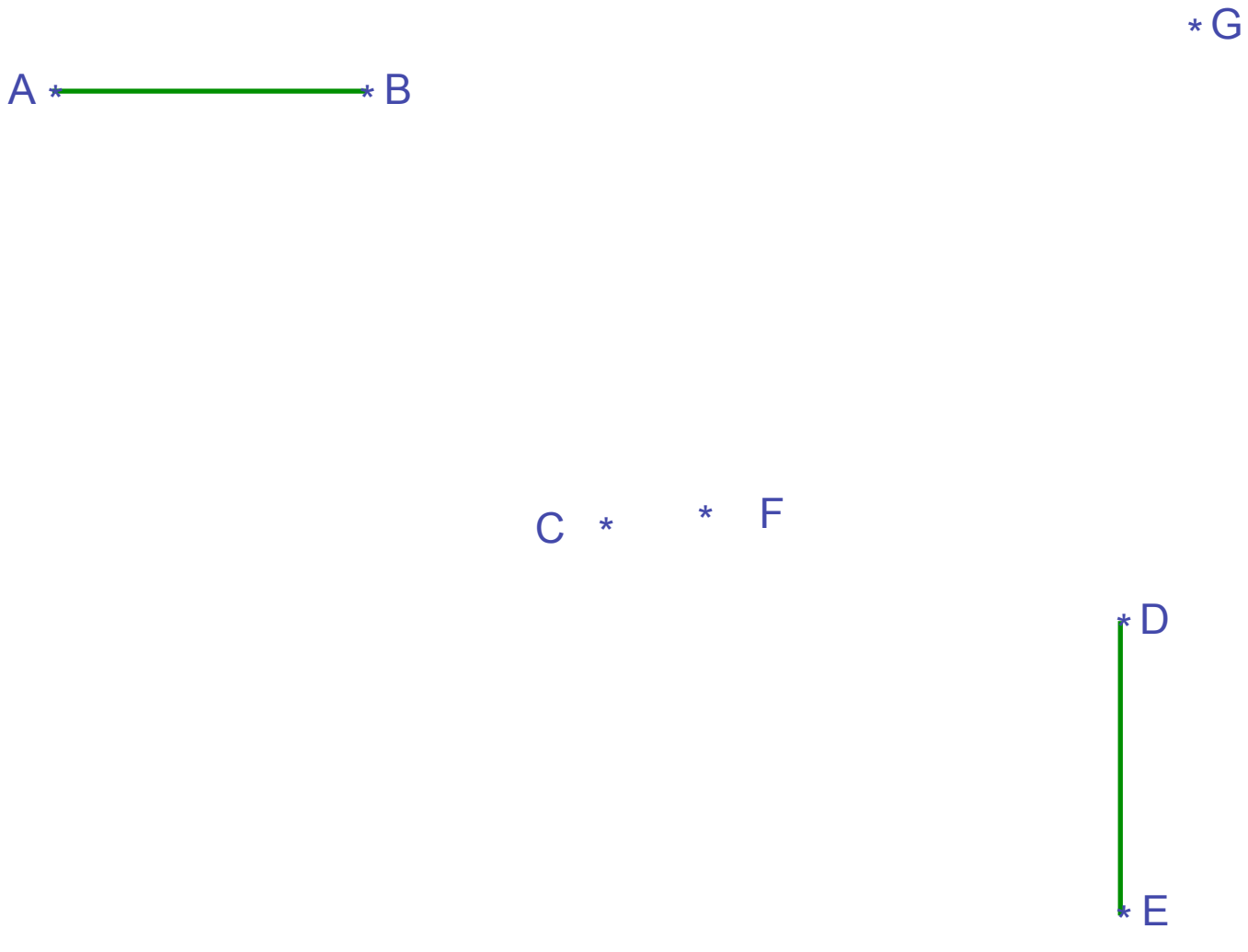
Single linkage hierarchical clustering



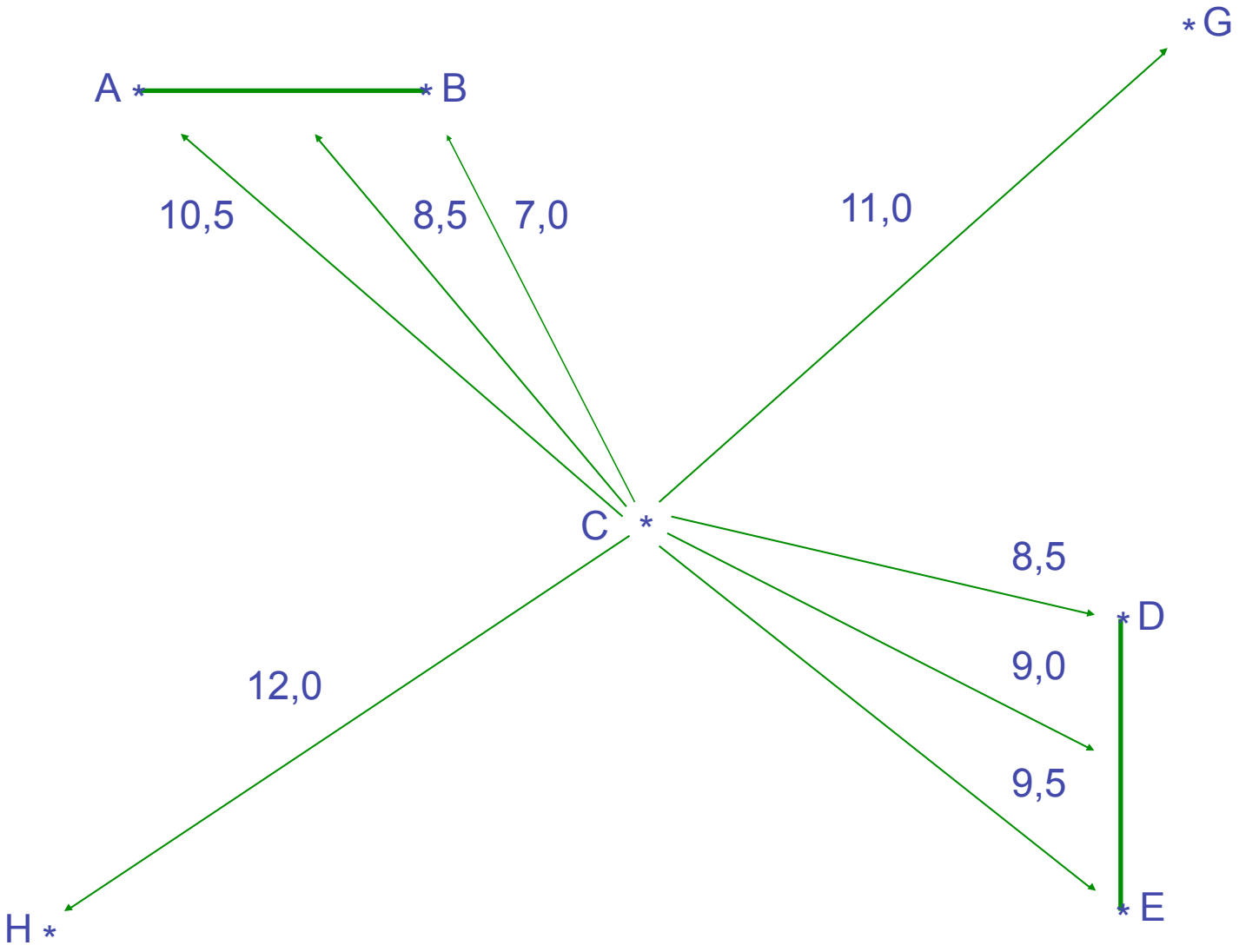
Which is the effect of the linkage?



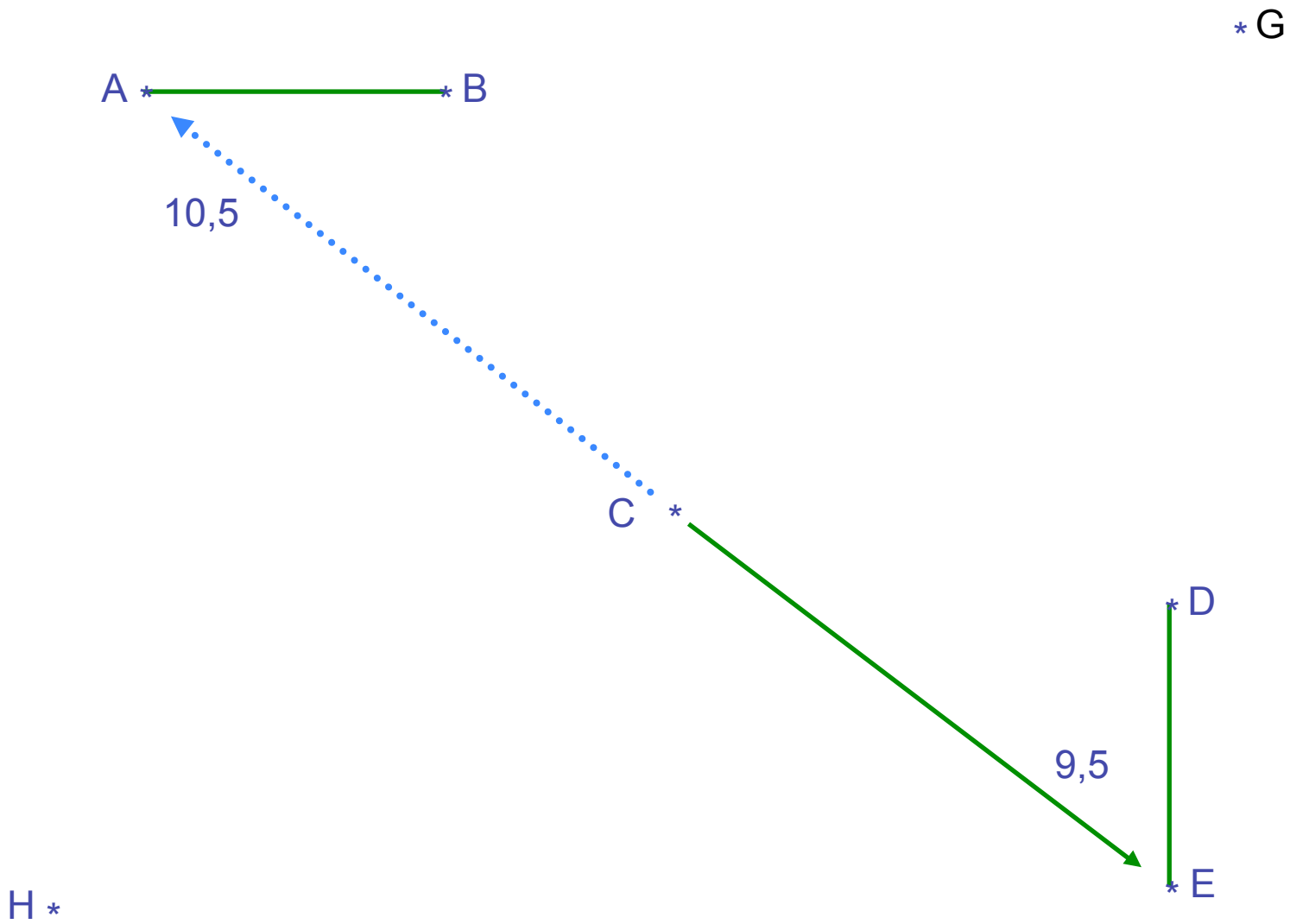
First joining . . .



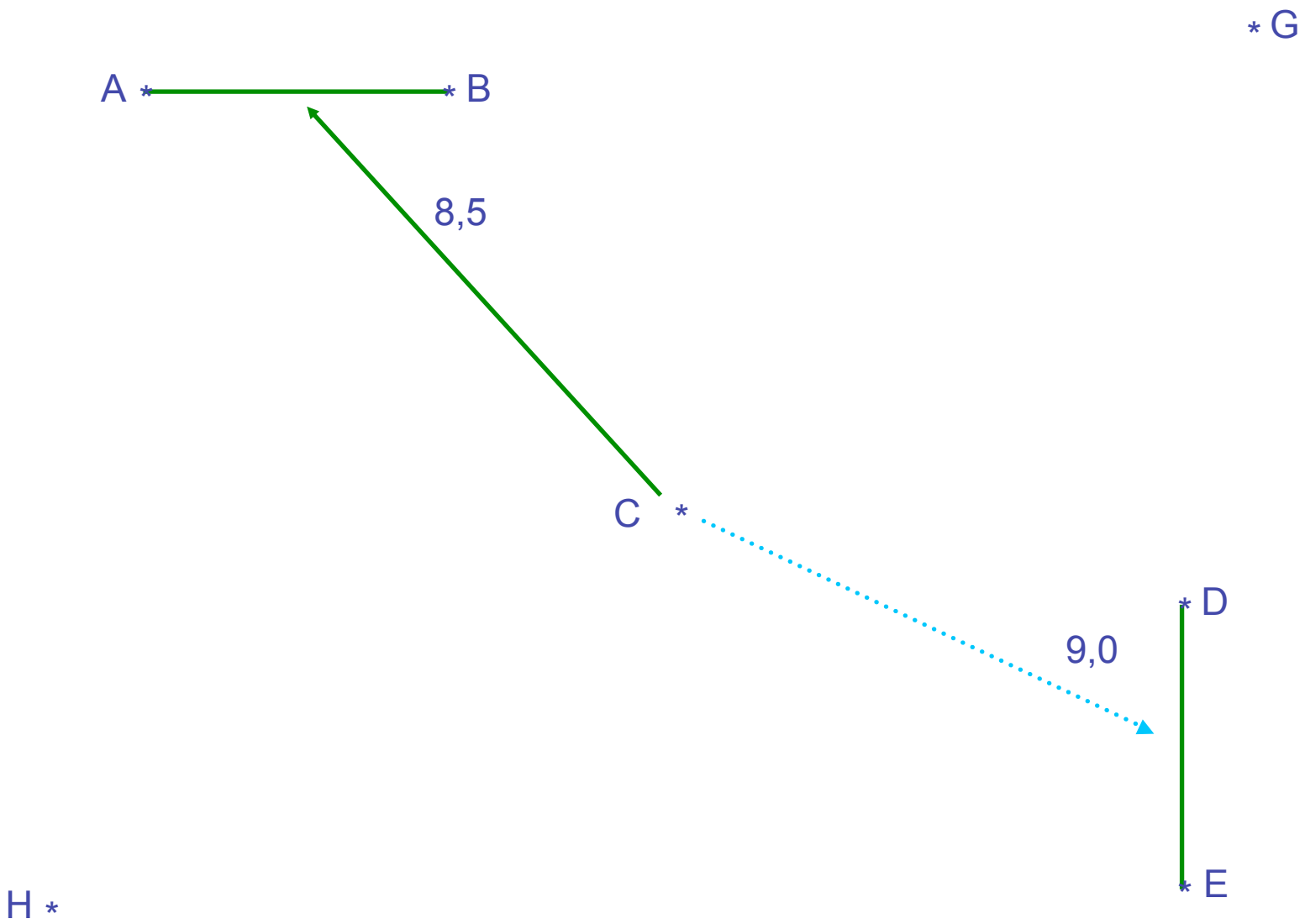
Where C will go?



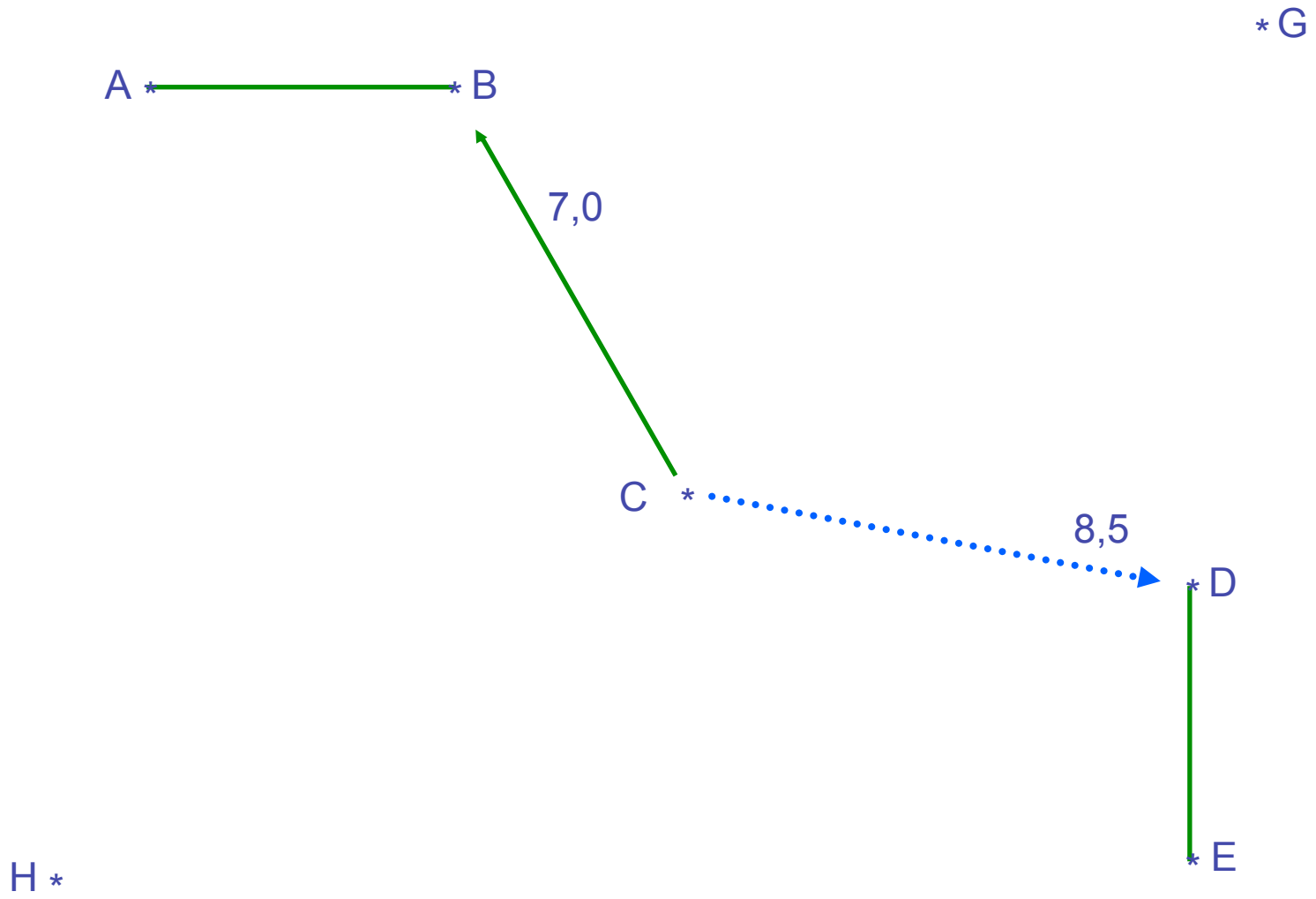
Complete linkage: minimize longest distance from cluster to point



Average linkage

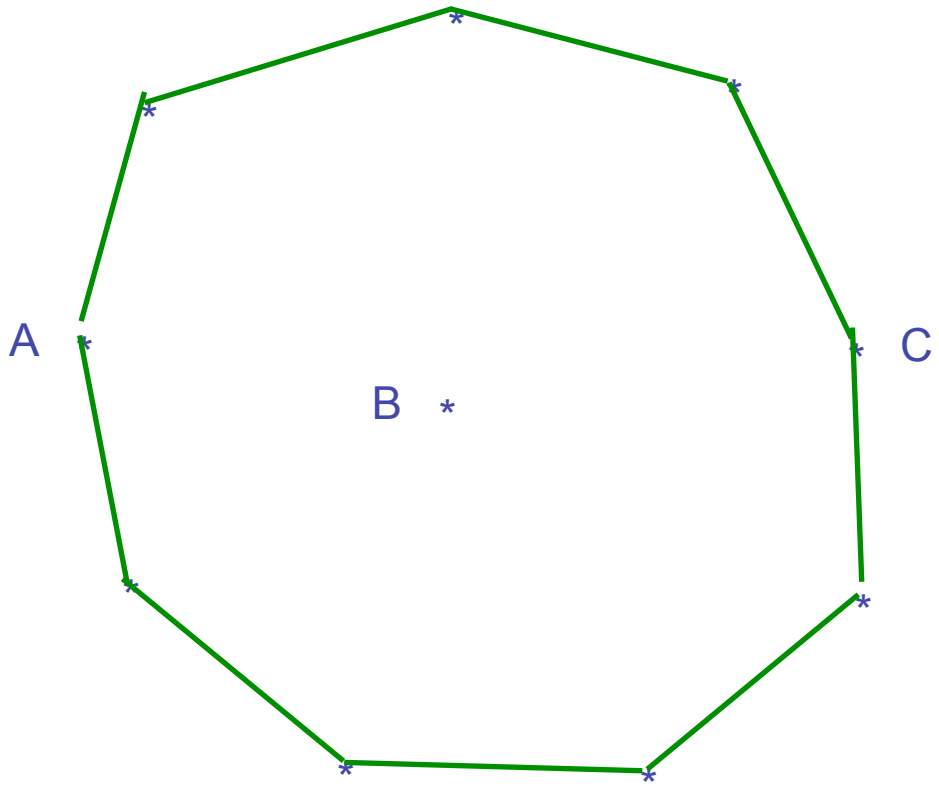


Single linkage



Single linkage: Pitfall

- Cluster formation begins . .
- All the time the closest observation is put into the existing cluster(s)
- Snake-like clusters
- A and C merge into the same cluster omitting B!



. . . but it is good to identify outliers

Summarizing . . .

- With Euclidean distance, one should prefer Ward or centroid linkage
- If groups have spheric form, complete linkage should be preferred
- At the presence of outliers, single linkage works better

Strengths and weakness of hierarchical clustering

Strengths

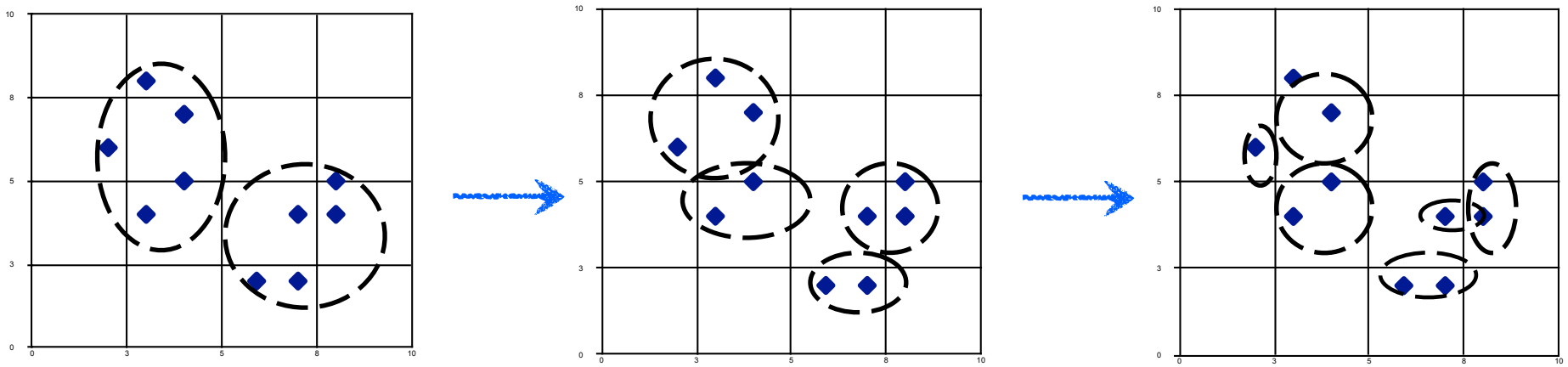
- Do not have to assume any particular number of clusters
  -- Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level.
- They may correspond to meaningful taxonomies
  -- (e.g., animal kingdom, phylogeny reconstruction, …)

Weekness

- Not efficient
- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized.

Hierarchical divisive methods

- DIANA (Divisive Analysis)
- Introduced in Kaufmann and Rousseeuw (1990)
- Start with one group and eventually each unit forms a cluster on its own



Recent Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  1. do not scale well: time complexity of at least $O(n^2)$, where n is the number of total units
  2. can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  3. BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  4. ROCK (1999): clustering categorical data by neighbour and link analysis
  5. CHAMELEON (1999): hierarchical clustering using dynamic modelling

Non-hierarchical clustering

- Partitioning: → K-means

- K-means (MacQueen'67) is one of the commonly used clustering algorithm

  - Aim: Construct a partition of a data set of n units into a set of K clusters

  - Method: Given a K, find a partition of K clusters that optimizes the chosen partitioning criterion

K-means clustering algorithm

Given K, the K-means algorithm is implemented in four steps:

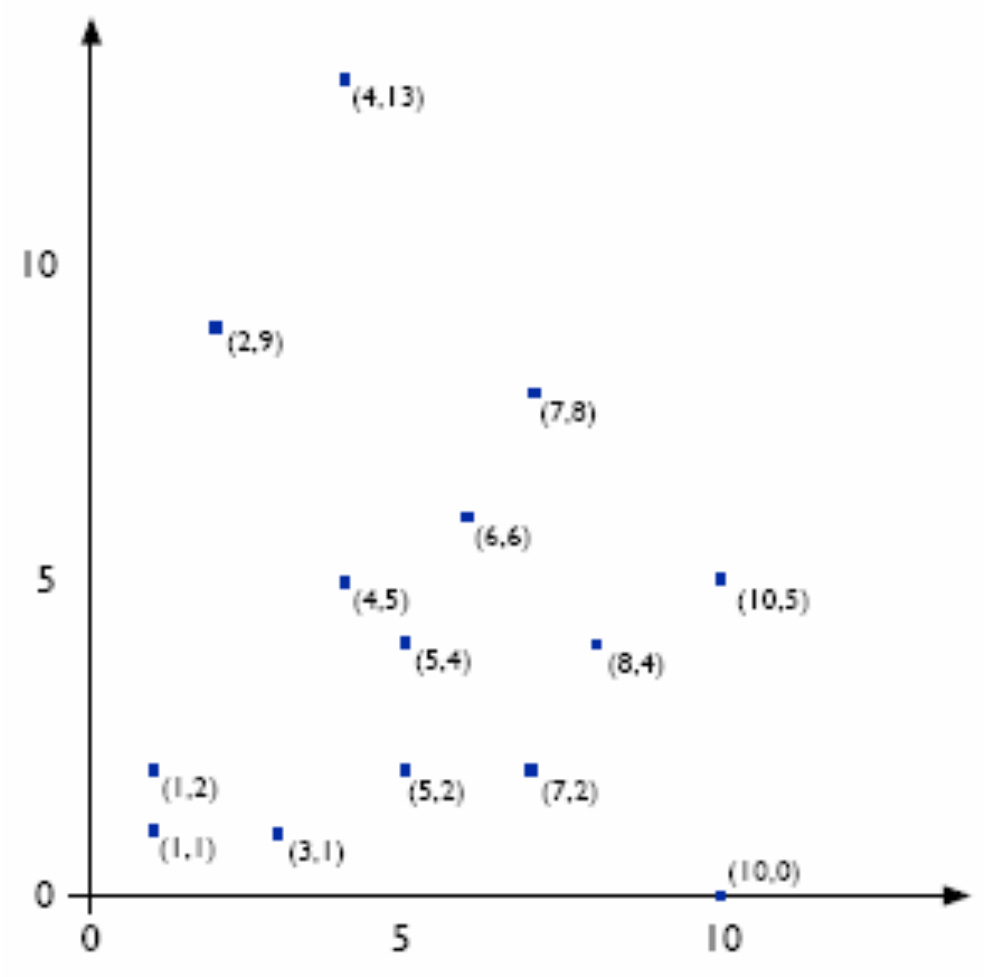
Step 1 Choose K points at random as cluster centers (centroids)

Step 2 Assign each unit to its closest cluster center using an opportune distance measure (usually Euclidean or Manhattan)

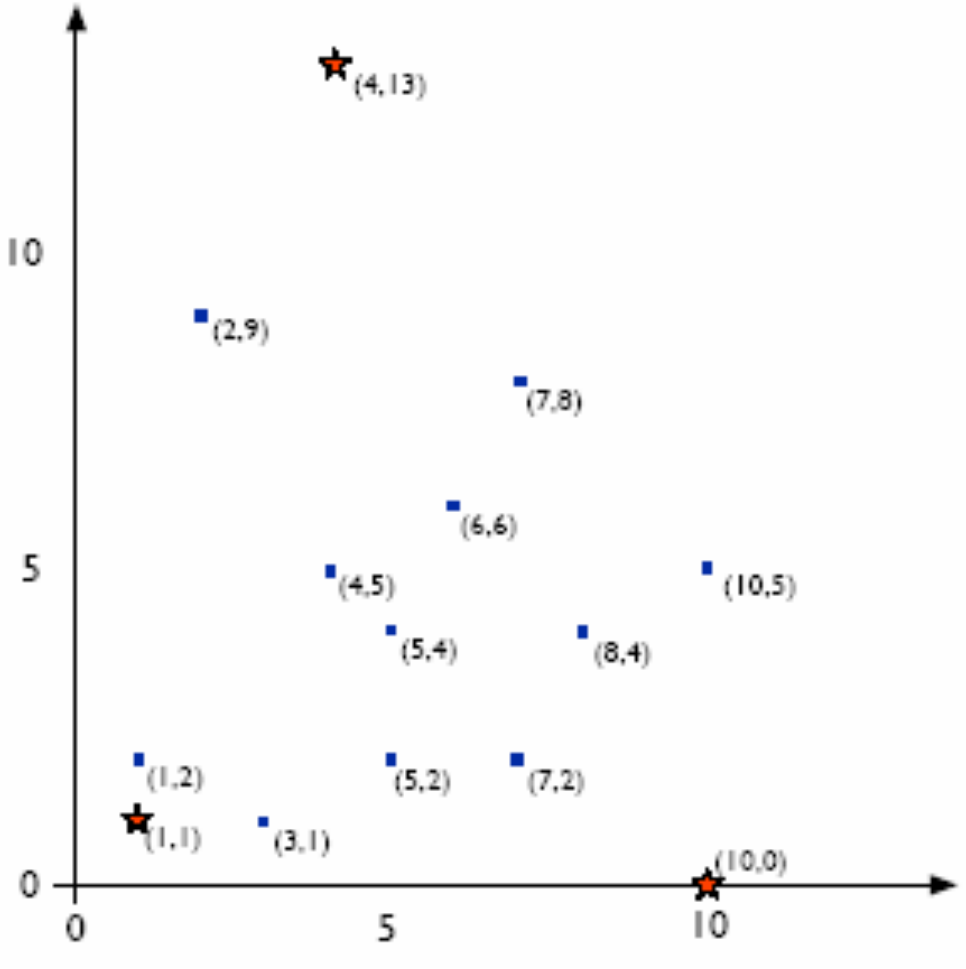
Step 3 Calculate the centroid of each cluster, use it as the new cluster center

Step 4 Go back to Step 2, stop when cluster centers do not change any more

Example: k-means

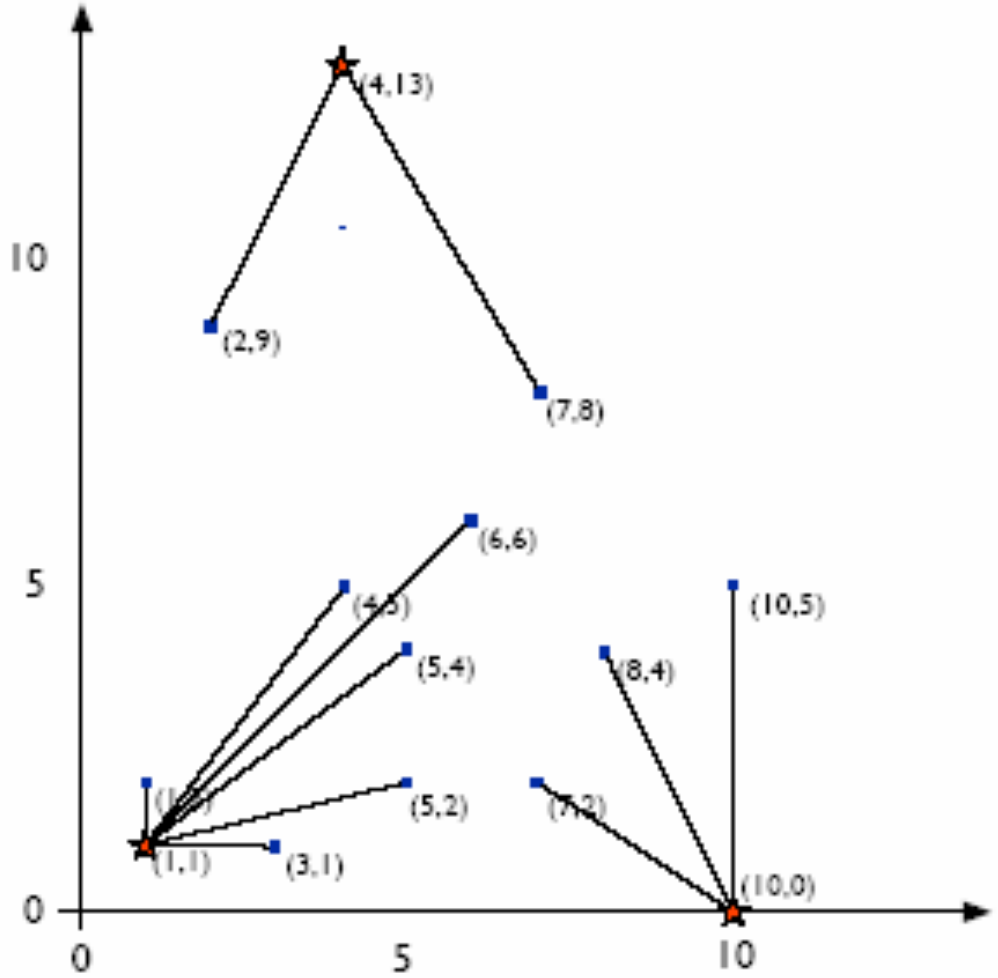


Original data

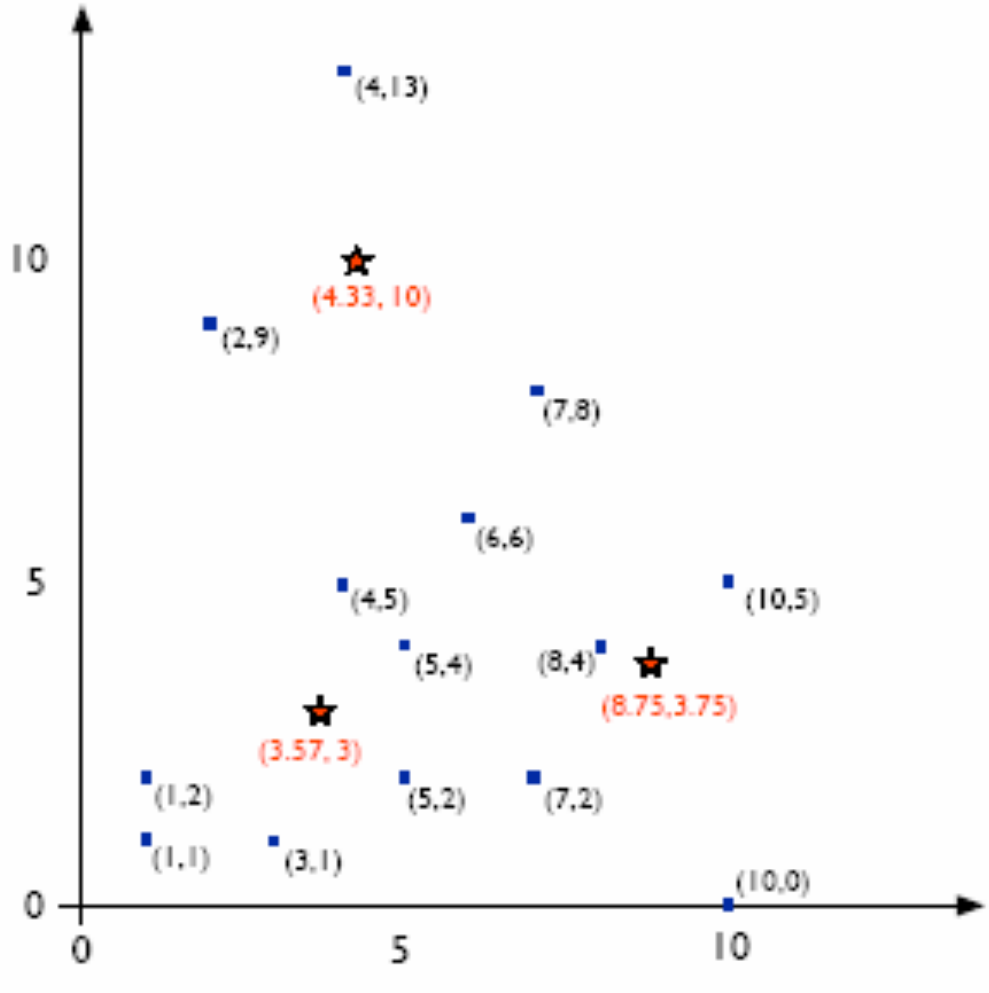


k random points
k=3

Example: k-means

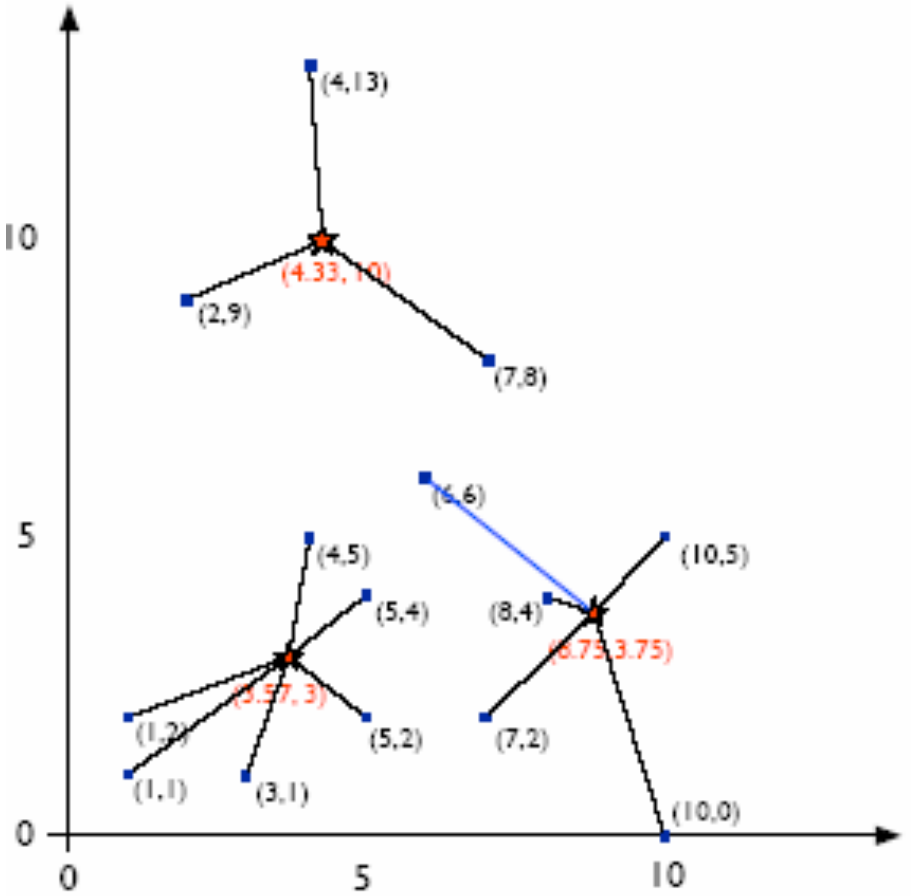


Assign units to the nearest center

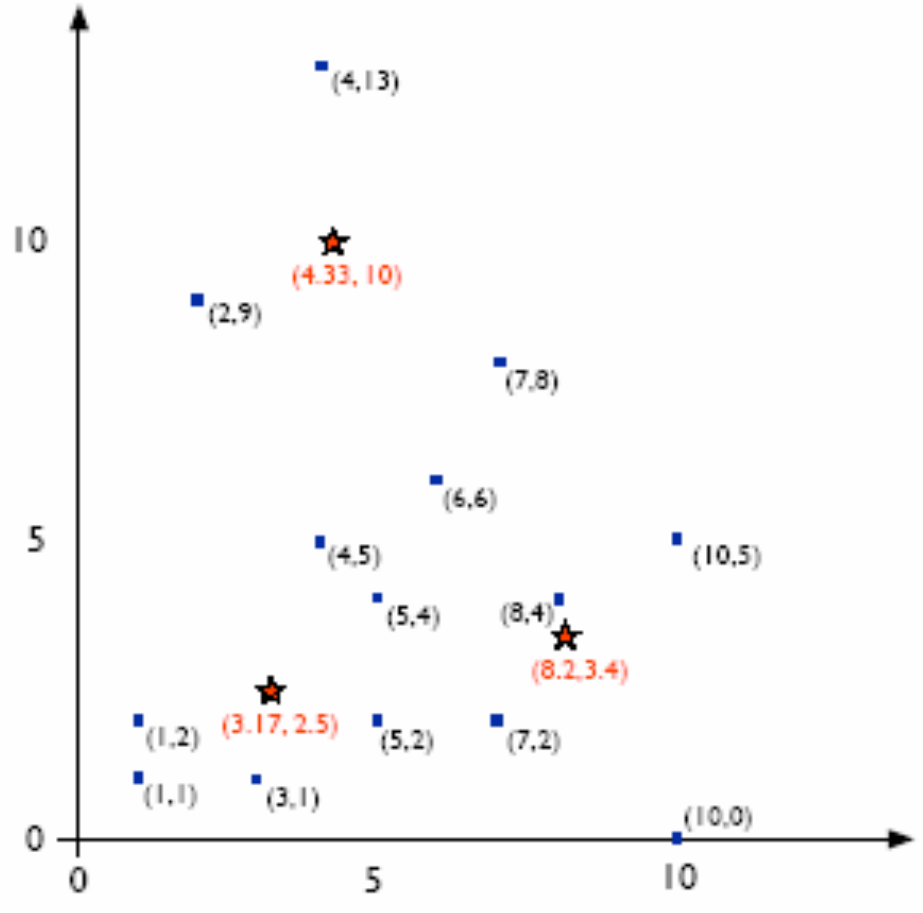


Re-compute centers

Example: k-means

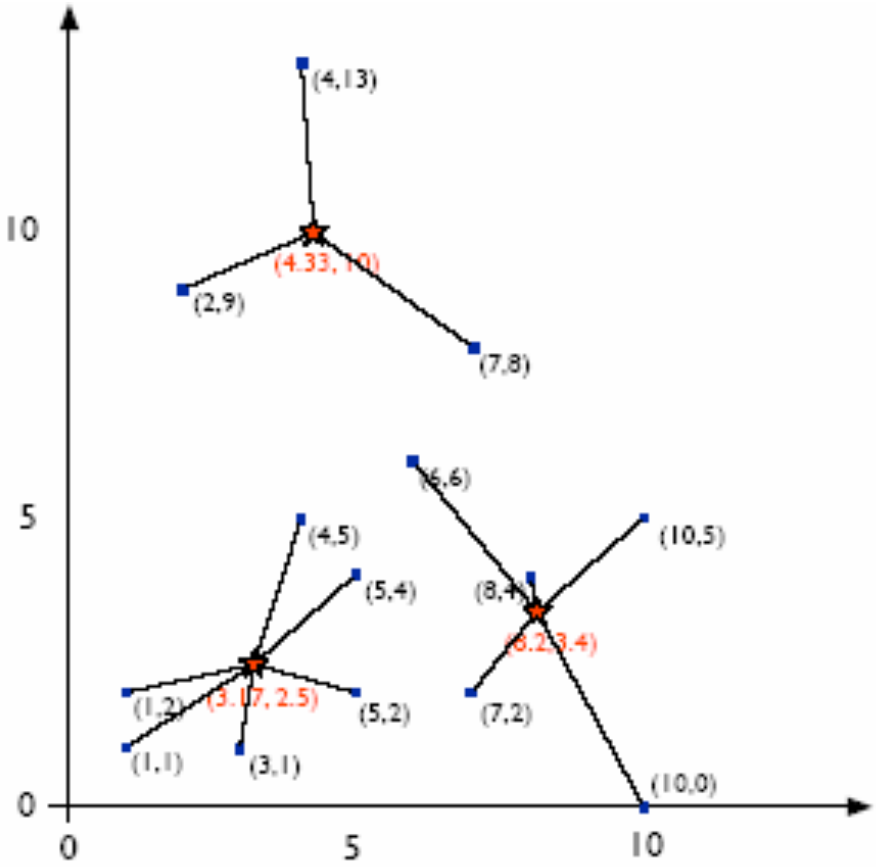


Assign again units to the nearest center

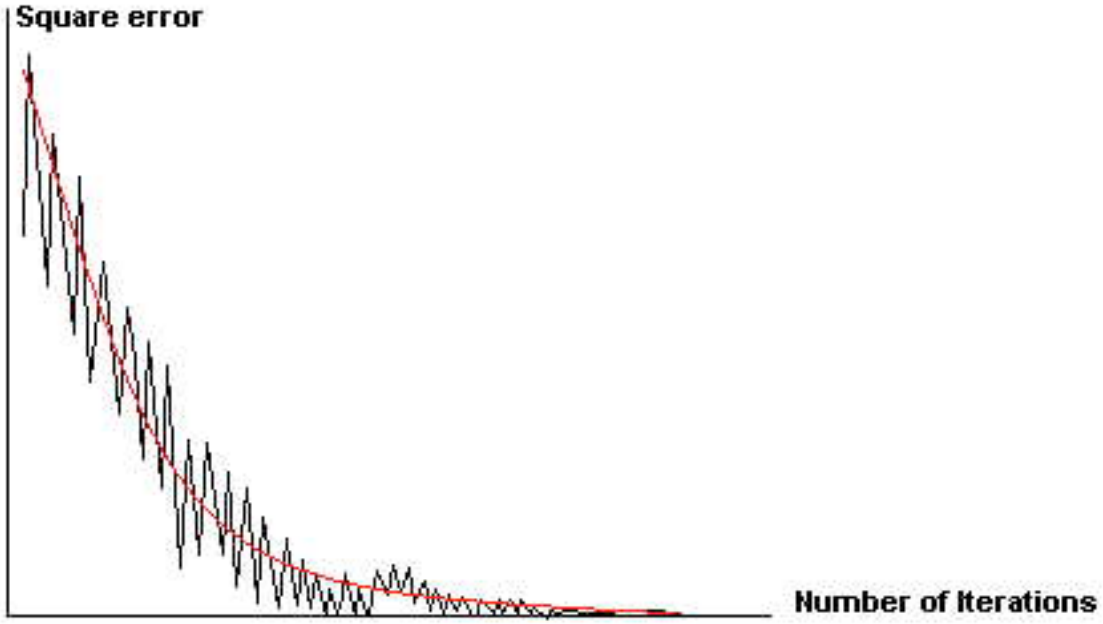


Re-compute new centers

Example: k-means



The algorithm tries to minimize the within-cluster sum-squared error (i.e. the sum of squared distances of each point from its cluster centroid



k-means method: some comments

Good properties:

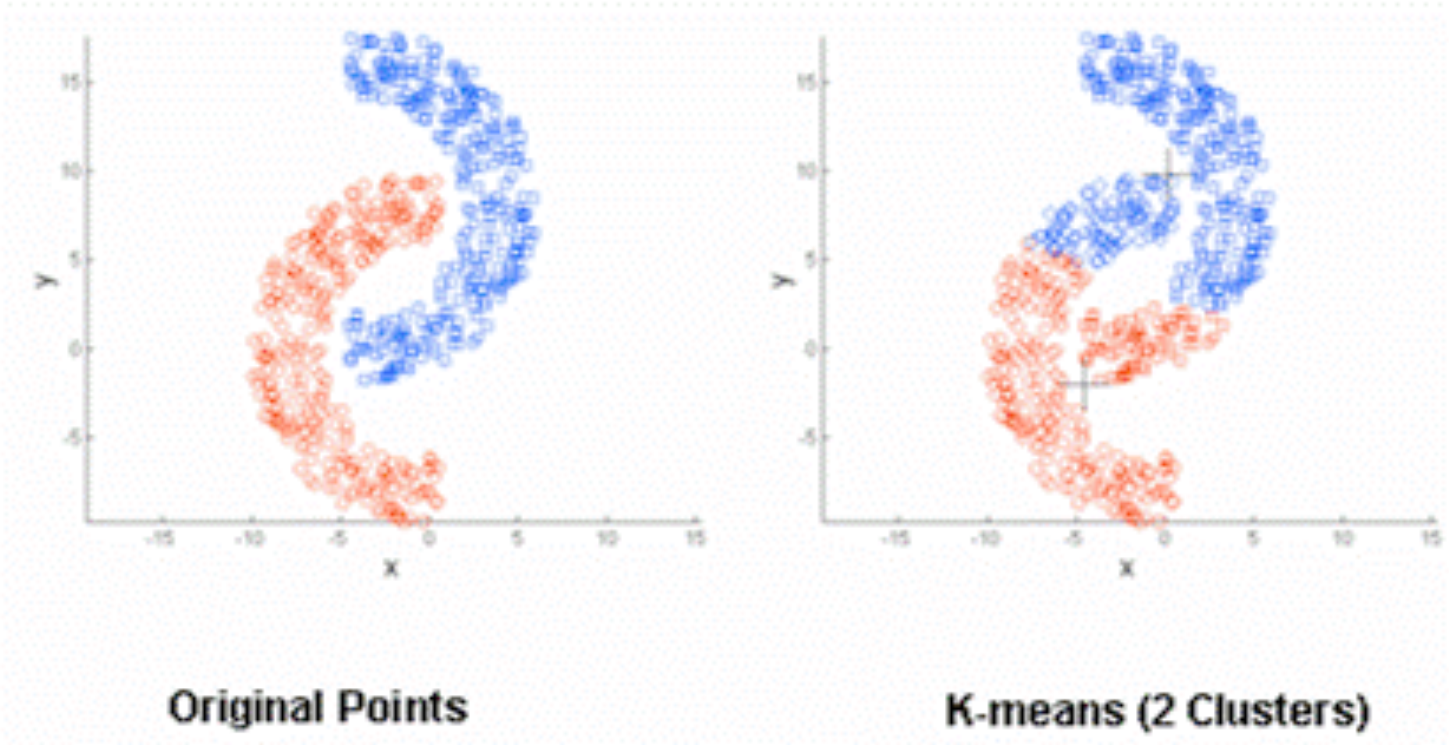
- Relatively efficient
- Guaranteed to converge (although to a local minima)

But some **issues**:

- Different initial partitions can result in different final clusters
- K must be pre-specified
- Does not work well with clusters (in the original data) of:
  - Different size
  - Different density
  - **Non-globular** shapes

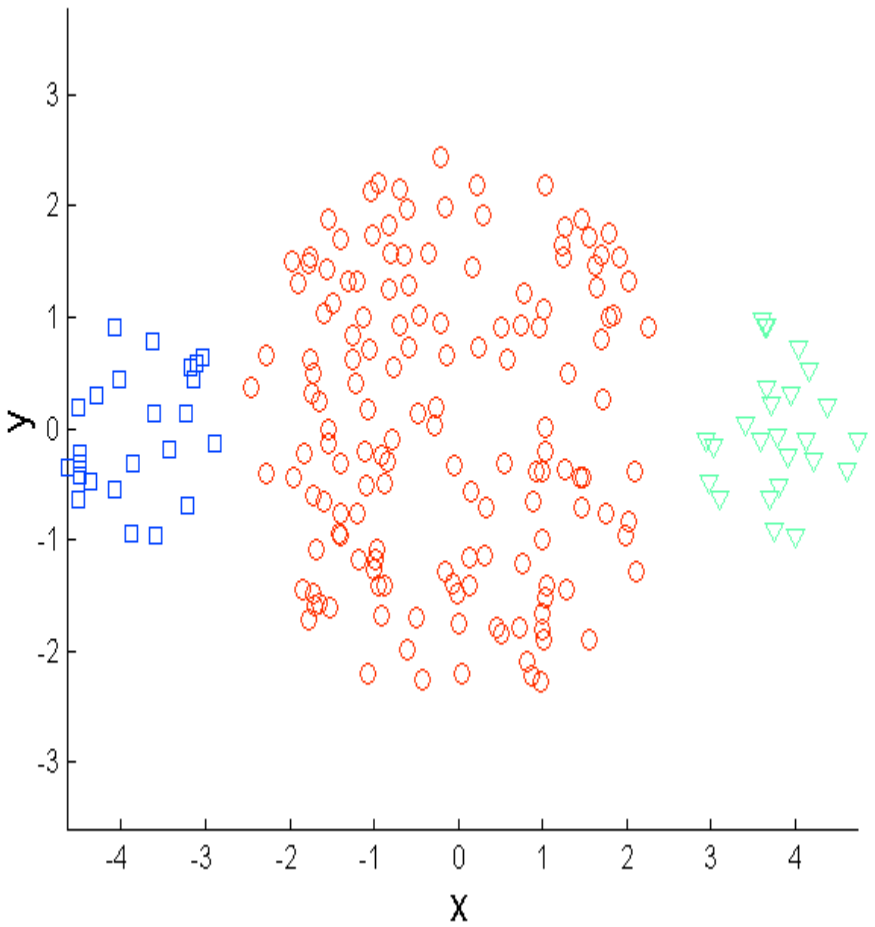
k-means method: some issues

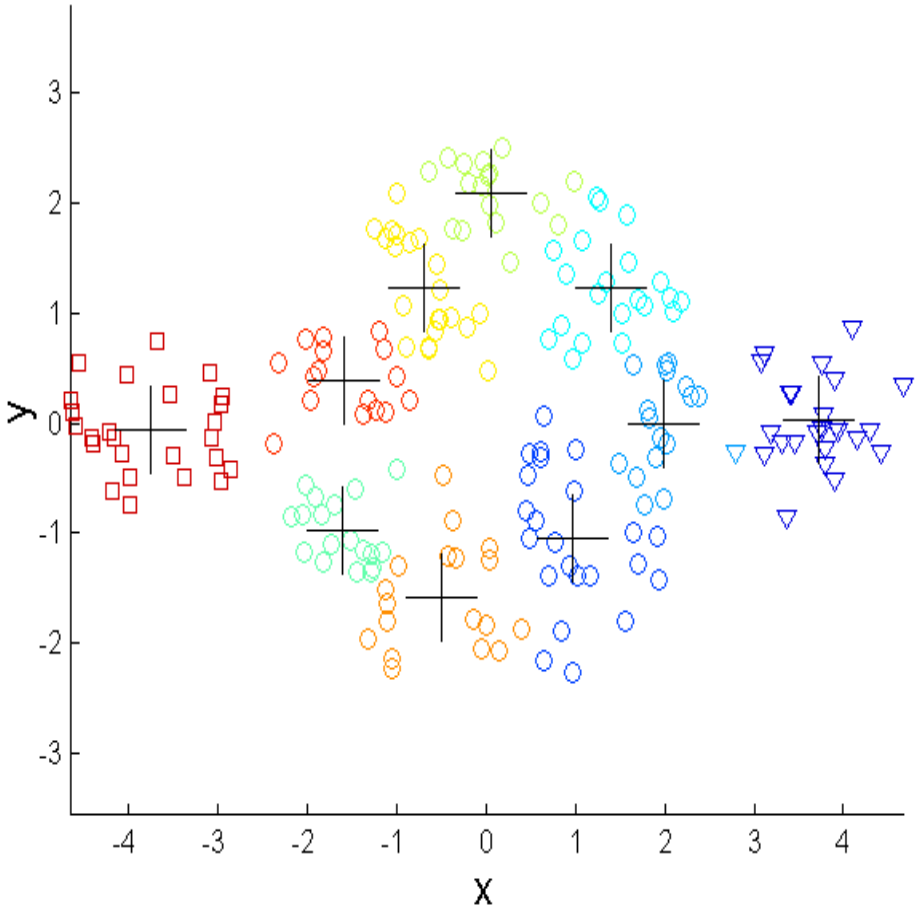
Non globular clusters



- Moreover this method has problem with outliers

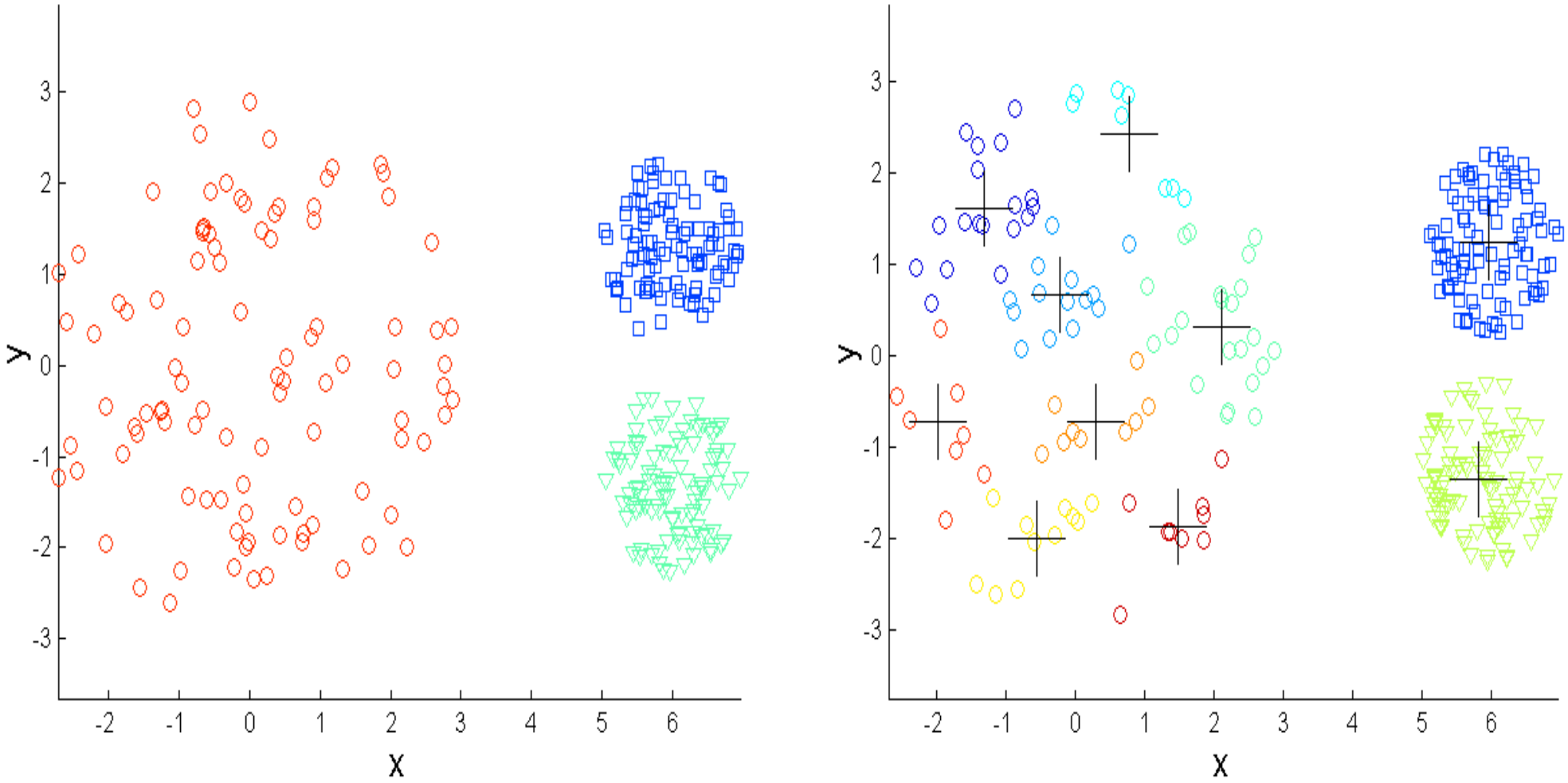
k-means method: some issues





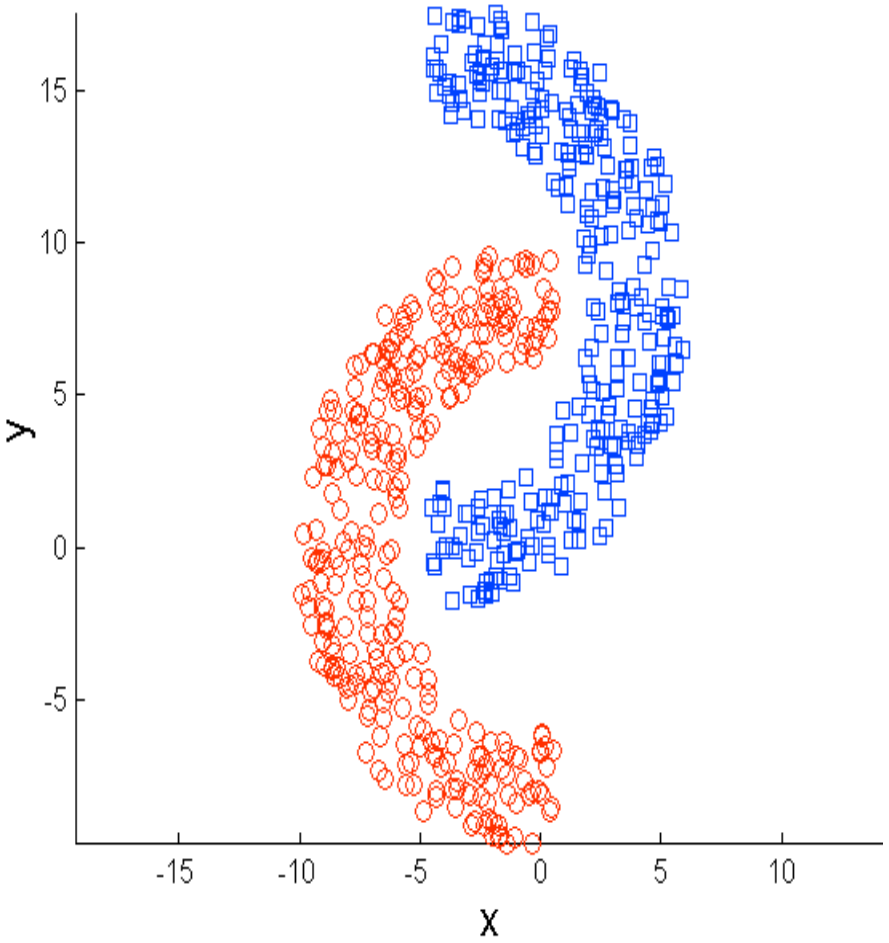
- How to overcome the different sizes problem: increase k and then apply a merging strategy

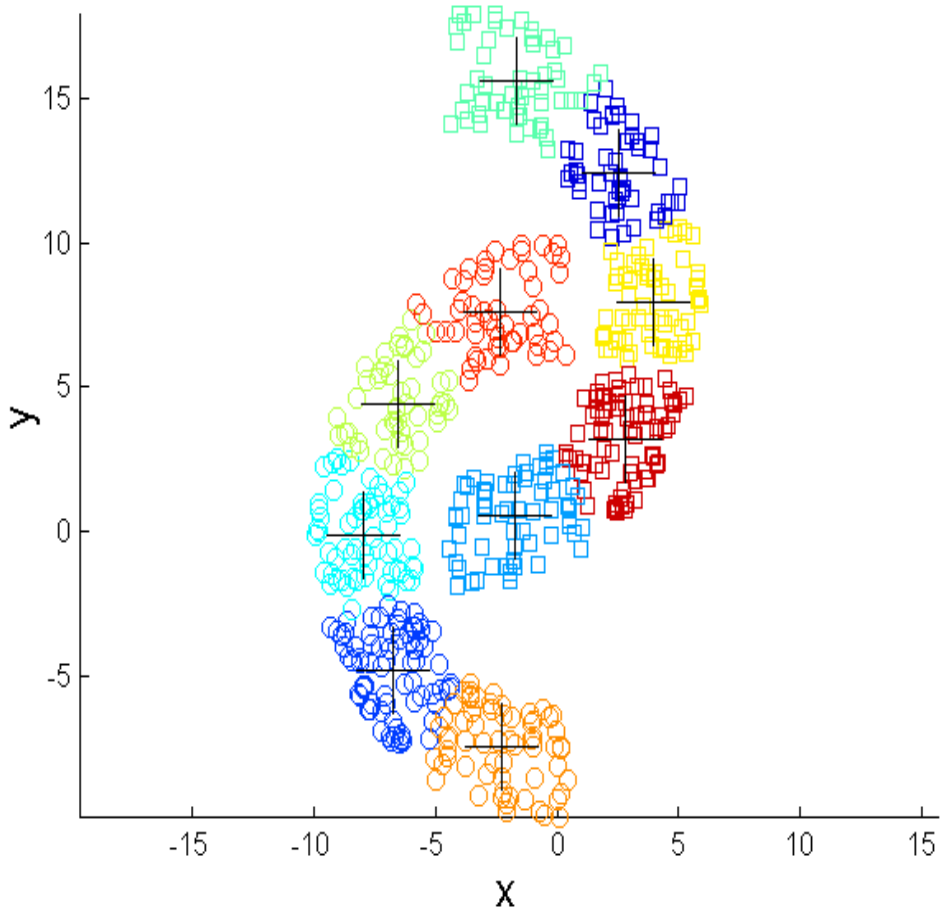
k-means method: some issues



- How to overcome the different sizes problem: increase k and then apply a merging strategy

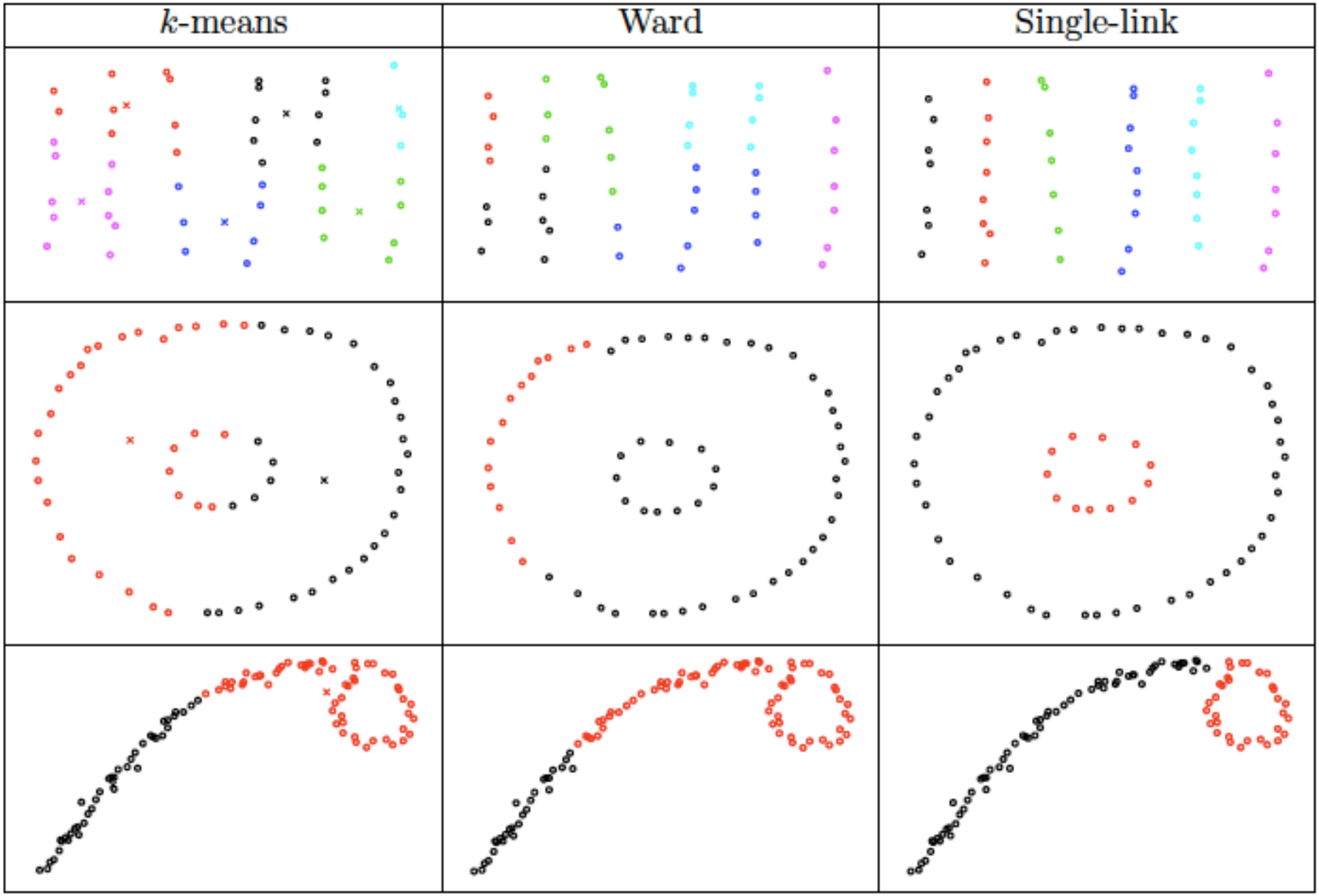
k-means method: some issues



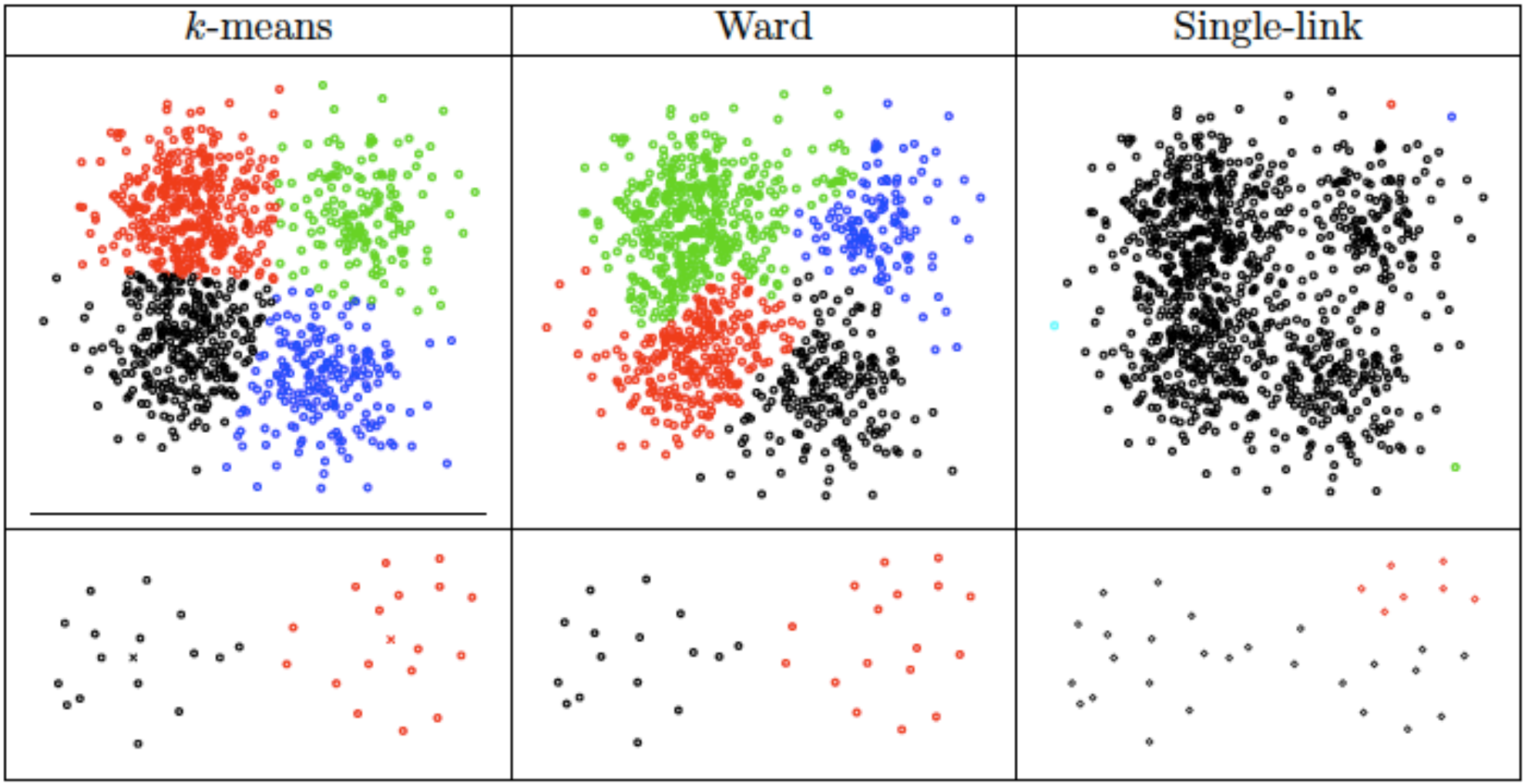


- How to overcome the non-spherical groups: increase k and then apply a merging strategy

Sometimes single link is better . . .



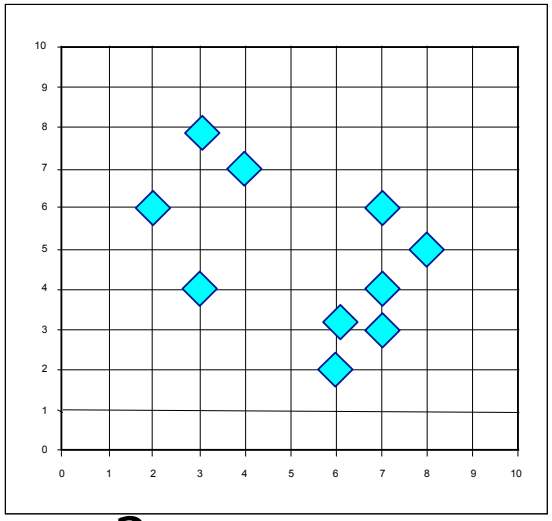
. . . sometimes it's not



Variations of the K-Means Method

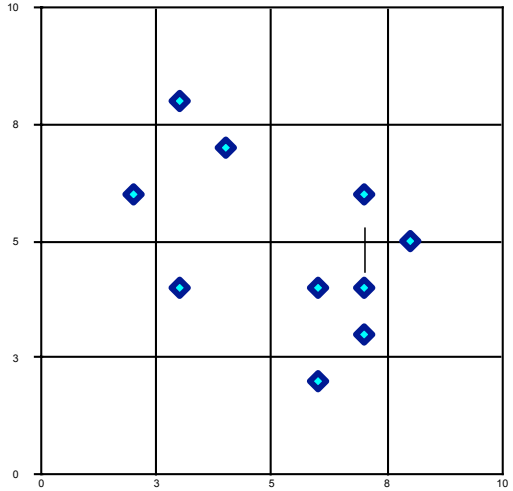
- Handling categorical data: **k-modes** (Huang'98)
  1. Replacing means of clusters with modes
  2. Using new dissimilarity measures to deal with categorical objects
  3. Using a frequency-based method to update modes of clusters
- Robust method: **K-medoids** and **Partitioning-Around-medoids** (PAM).
  4. See http://en.wikipedia.org/wiki/K-medoids for details

A Typical K-Medoids Algorithm (PAM)



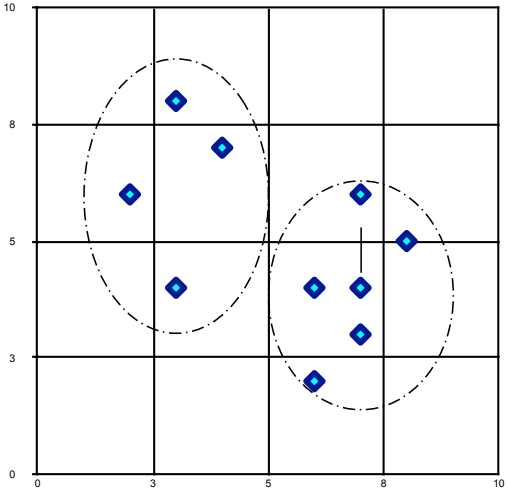
p=2

Arbitrary choose k object as initial medoids

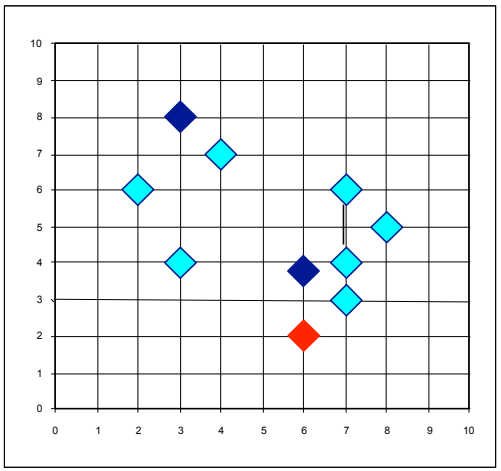


Assign each remaining object to nearest medoids

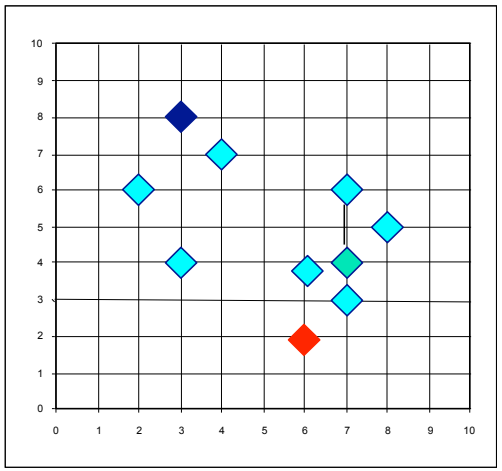
Total Cost = 20



Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping



Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

Do loop until no change

Cost=sum all changes in differences

Overlapping clustering – Fuzzy C-means algorithm

- Methods considered until now assign units only to one cluster
- Fuzzy C-means (FCM) is a method which allows units to belong to more than one cluster, with a certain degree of membership
- This method (Dunn (1973) & Bezdek (1981)) is frequently used in pattern recognition

Fuzzy C-means algorithm

- Choose:
  - κ = num. clusters
  - q = degree of fuzzification
- Randomly select k centroids c_i (ora membership matrix)
- Compute the membership matrix (n x k)

$$m_{ij} = \left(\sum_{s=1}^{k} \left(\frac{d_{ij}}{d_{is}} \right)^{2/(q-1)} \right)^{-1}$$

Membership value of unit i to cluster j = relative gravitation to cluster j

$$d_{ij} = \|x_i - c_j\|$$

Distance of unit i to centroid of cluster j

Fuzzy C-means algorithm

- **Assign** each unit to the cluster with higher membership (smallest distance)

$$m_{ij}^{*} = \begin{cases} 1 & \text{if } m_{ij} > m_{ij'} \\ 0 & \text{otherwise} \end{cases}$$

- Compute the new center of each group
- Repeat until centers are stable
- Compute the final membership matrix

NB. Objective function: Highest degree of membership =>

Fuzzy evaluation

"Good" clusters are actually not very fuzzy

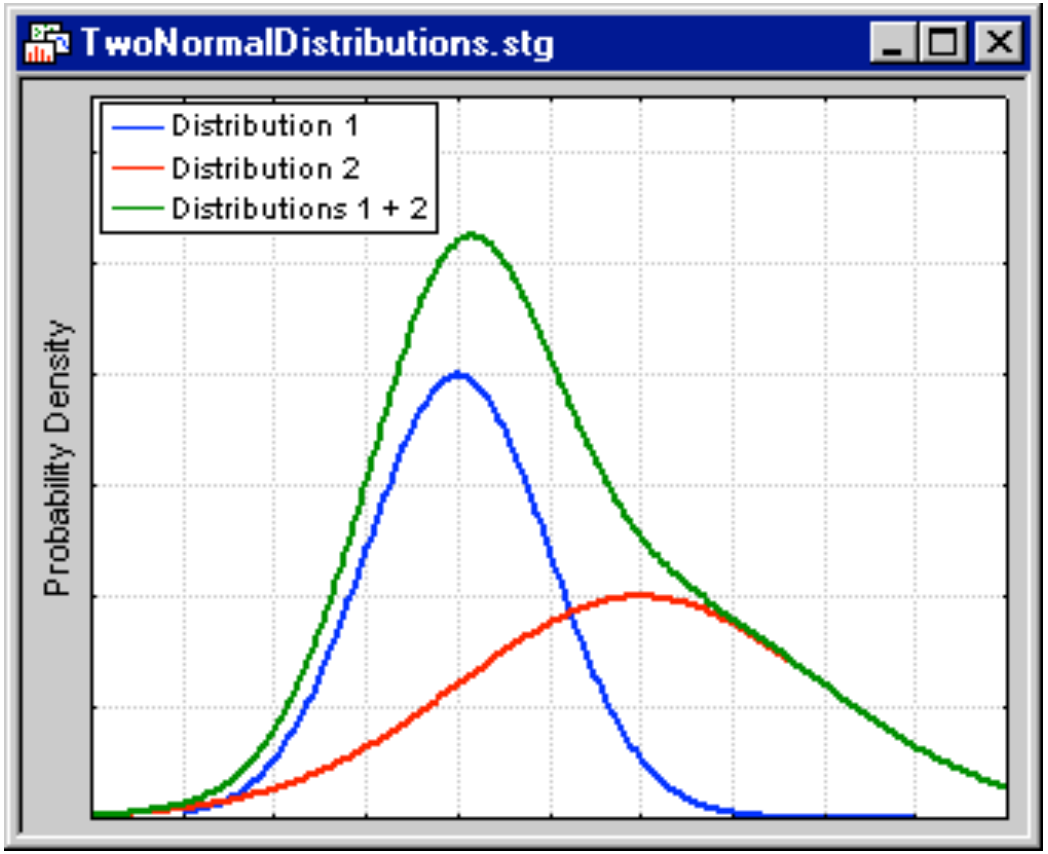
- The criteria for the definition of "optimal partition" of the data into subgroups are based on the following requirements:

  1. Clear separation between the resulting clusters
  2. Minimal volume of the clusters
  3. Maximal number of data points concentrated in the vicinity of the cluster centroid

Probabilistic clustering methods

- Assume that data is generated by a weighted combination of random processes.
- Aim: to find the weights and parameter values that best fit the data.

A cluster is considered a distribution (e.g. Gaussian) around a center

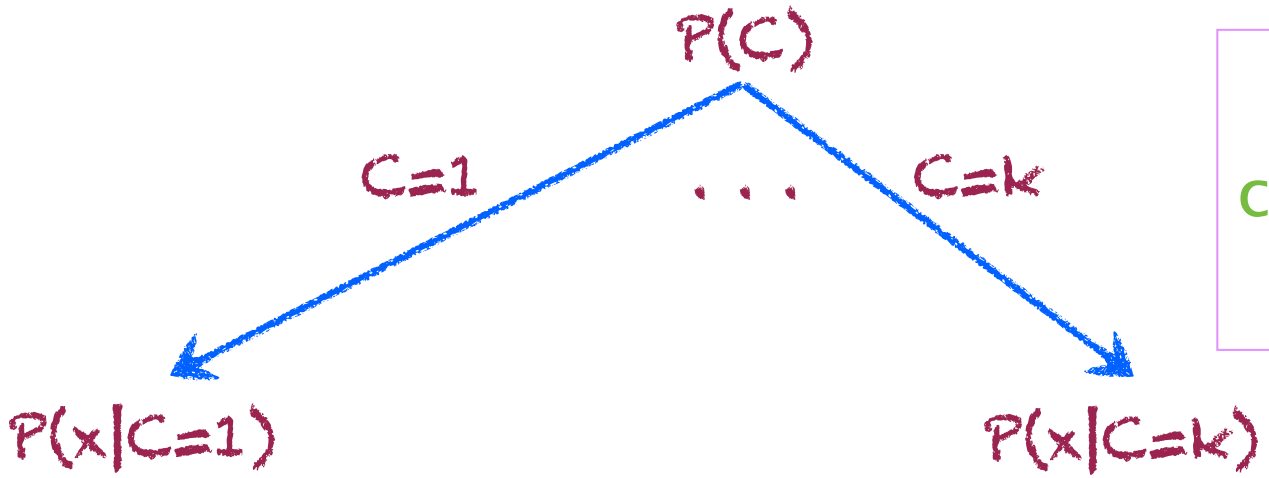


NB. ≠ density-based methods

Mixture models

- Assume a cluster indicator variable $C = 1, \ldots, K$ with probability $p_1, \ldots p_k$, summing to one.

- Data generating process assumed



Any unit x could be generated in k ways

- Generic mixture: $$P(x) = \sum_{i=1}^{k} p_i \, P(x \mid C = i)$$

Gaussian mixture models

- The conditional density of $x = (x_1, \ldots, x_p)$ when $C = i$ is

$$f(x \mid C = i) = \phi(x \mid \mu_i, \Sigma_i) \longrightarrow N_p(\mu_i, \Sigma_i)$$

- The marginal density of x is

$$f(x) = \sum_{i=1}^{k} p_i \, \phi(x \mid \mu_i, \Sigma_i)$$

- The parameters of interest are

$$p_1, \ldots, p_k, \; \mu_1, \ldots, \mu_k, \; \Sigma_1, \ldots, \Sigma_k,$$

- If we had labels indicating who belongs to group j, to reach these estimates would be an easy task, but...

Gaussian mixture models

- If we had labels (group indicators) , then

$$\hat{p}_i = \frac{\hat{n}_i}{n}$$

$$\hat{\mu}_i = \frac{1}{\hat{n}_i} \sum_{u:C_u=i} x_u$$

$$\hat{\Sigma}_i = \frac{1}{\hat{n}_i} \sum_{u:C_u=i} (x_u - \hat{\mu}_i)(x_u - \hat{\mu}_i)'$$

- Luckily these labels can be estimated as posterior probabilities

$$\hat{p}_{u,i} = P(C_u = i \mid x_u, \theta) =$$

$$= \frac{P(x_u|C=i,\theta)\, P(C=i)}{\sum_{j=1}^{k} P(x_u|C=k,\theta)\, P(C=k)}$$

EM algorithm for Gaussian mixture models

Step 1 Initialize parameters

Step 2: E-step Compute the posterior probabilities for each $u = 1, \ldots n$ and $i = 1, \ldots k$

$$p_{u,i} = \frac{\phi\left(x_u \mid \mu_i^{(s)}, \Sigma_i^{(s)}\right) \cdot p_i^{(s)}}{\sum_{j=1}^{k} \phi\left(x_u \mid \mu_j^{(s)}, \Sigma_j^{(s)}\right) \cdot p_j^{(s)}}$$

Iteration s

Step 3: M-step

$$p_i^{(s+1)} = \frac{\sum_u p_{u,i}}{n} \qquad \mu_i^{(s+1)} = \frac{\sum_u p_{u,i} x_u}{\sum_u p_{u,i}}$$

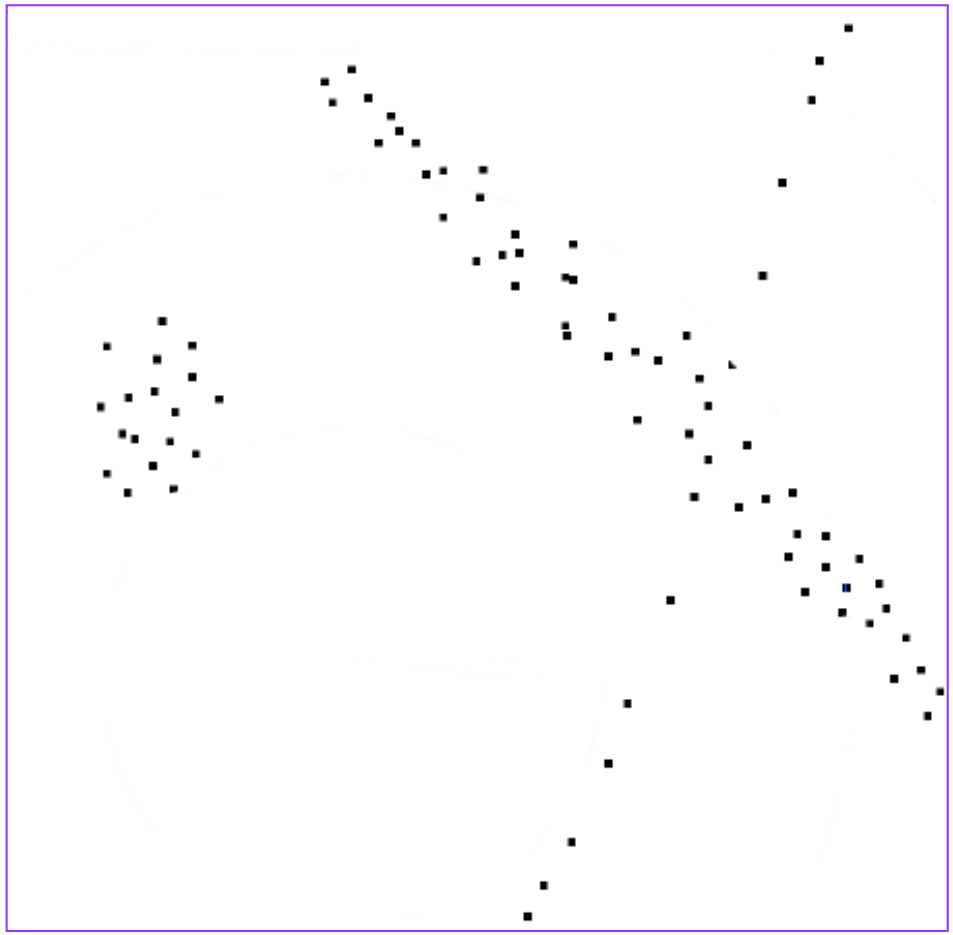
$$\Sigma_i^{(s+1)} = \frac{\sum_u p_{u,i} (x_u - \mu_i^{(s+1)})(x_u - \mu_i^{(s+1)})'}{\sum_u p_{u,i}}$$

Step 4 Repeat 2 and 3 until convergence

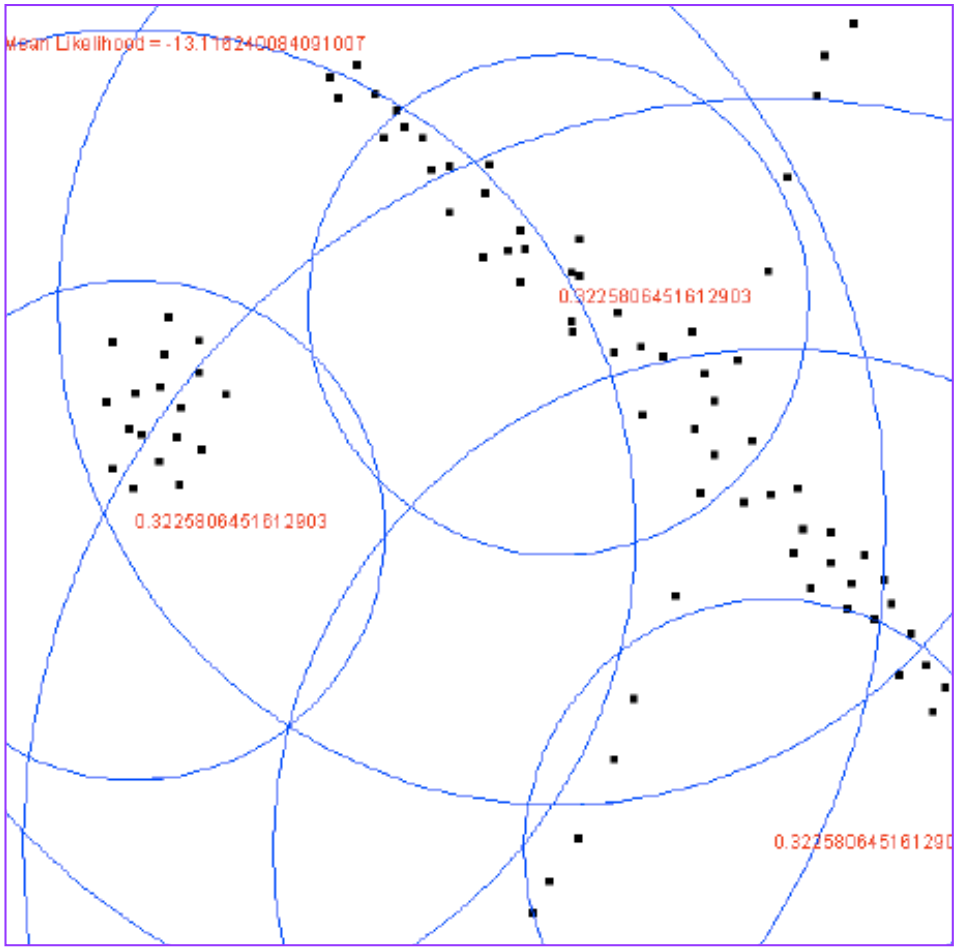
Some comments

- In the E-step we are estimating the membership probabilities subject to sum to 1 constraint on the clusters. This is the optima estimate
- In the M-step we are optimizing the marginal likelihood function, which is the objective function for this clustering method
- Sometimes one assumes equal covariance matrixes
- The EM algorithm always converge, but can “fall” in local optima. Try different starting values or initialize by k-means clustering
- We obtain “soft” cluster like in fuzzy clustering

Example

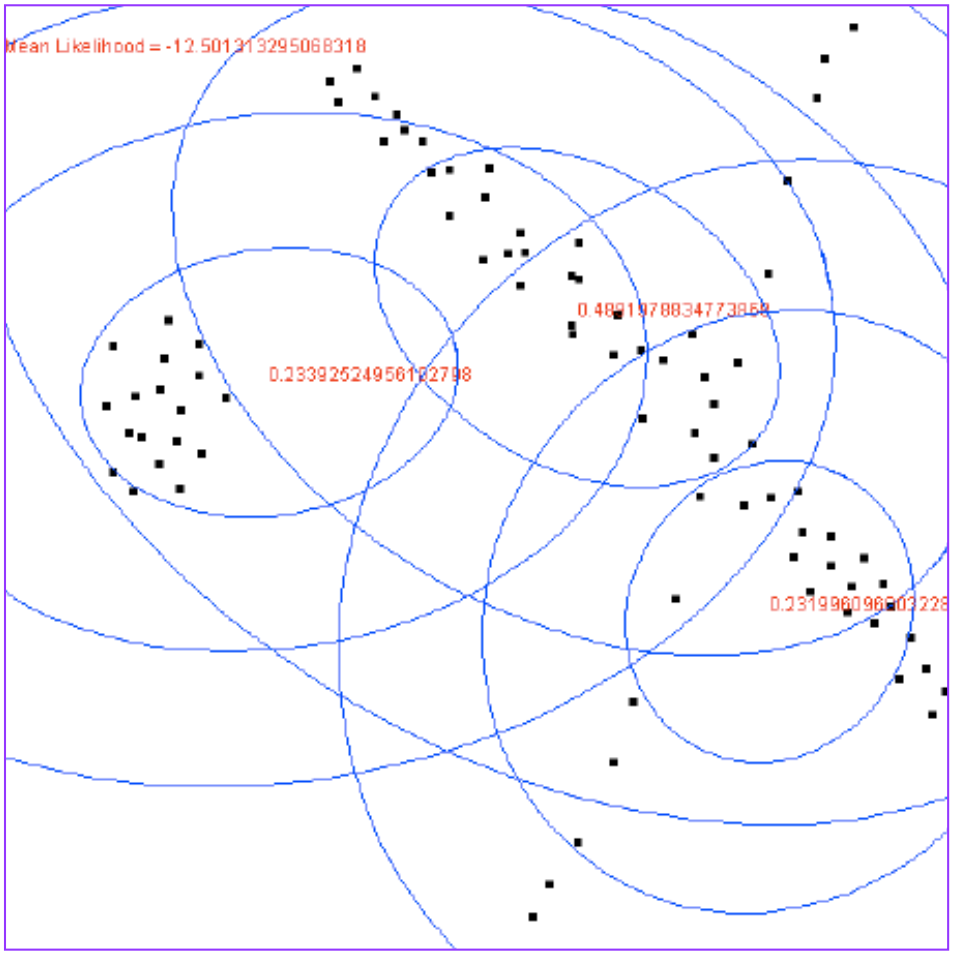


Data

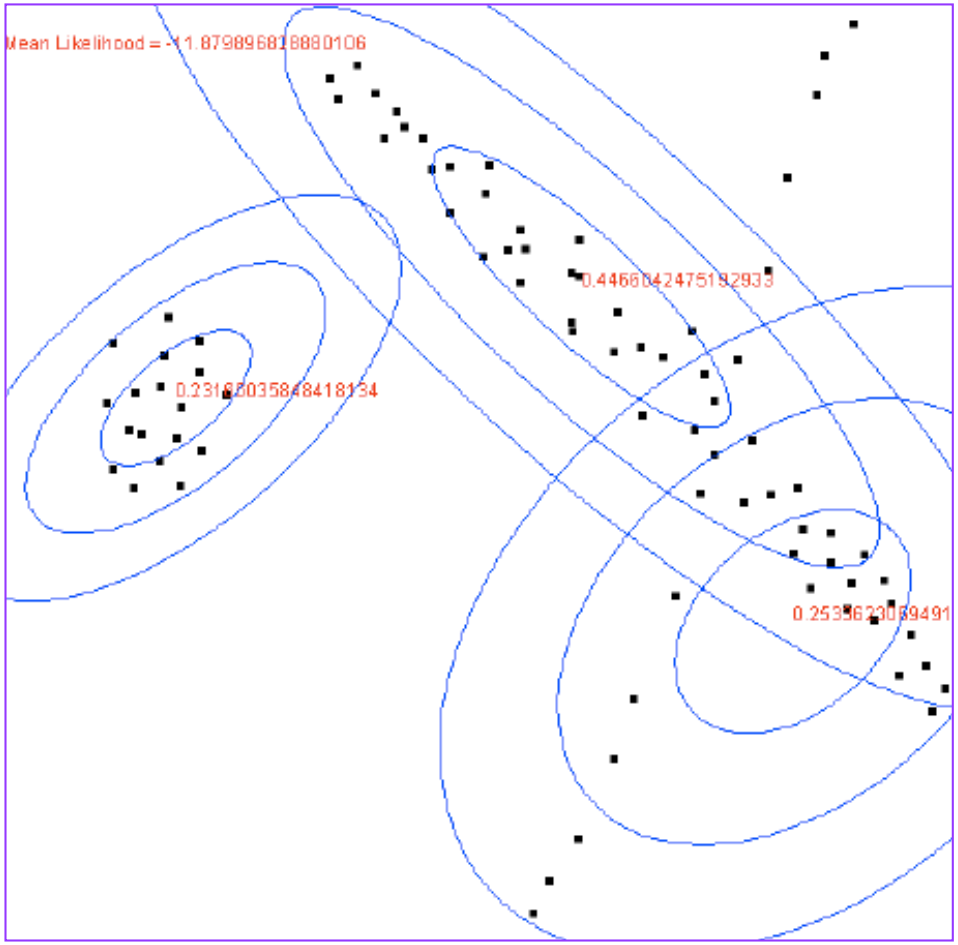


Iteration 1
means are randomly assignes

Example

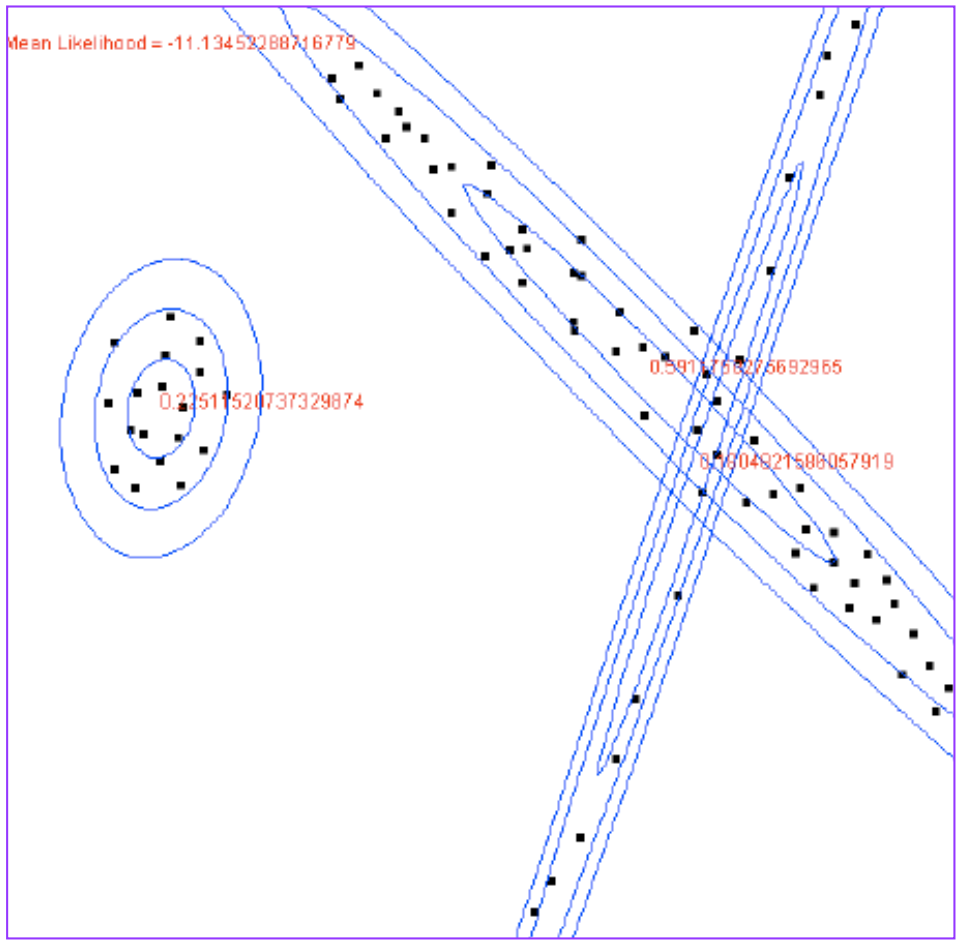


Iteration 2



Iteration 3

Example



Iteration 25

Quality: What is good clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns