Theory and applications of Clustering algorithms.

Master in High Performance Computing

Alex Rodriguez.
alexrod@sissa.it
SISSA, via Bonomea 265.
Office 418
+39 040 3787 874

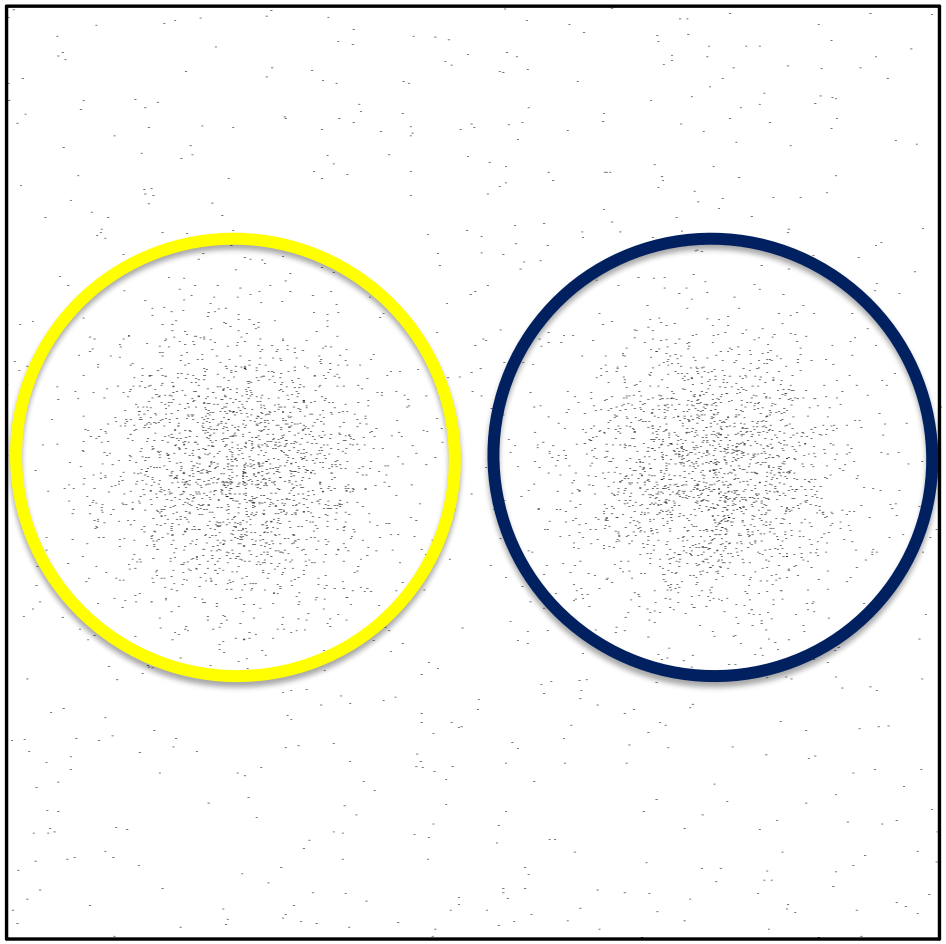
Theory and applications of Clustering algorithms.

- Motivation for Clustering
- Feature selection and Dimensional reduction
- Similarities and Distances
- Flat, fuzzy and Hierarchical clustering methods
- Clustering methods examples
- External and Internal Validation
- Clustering applications

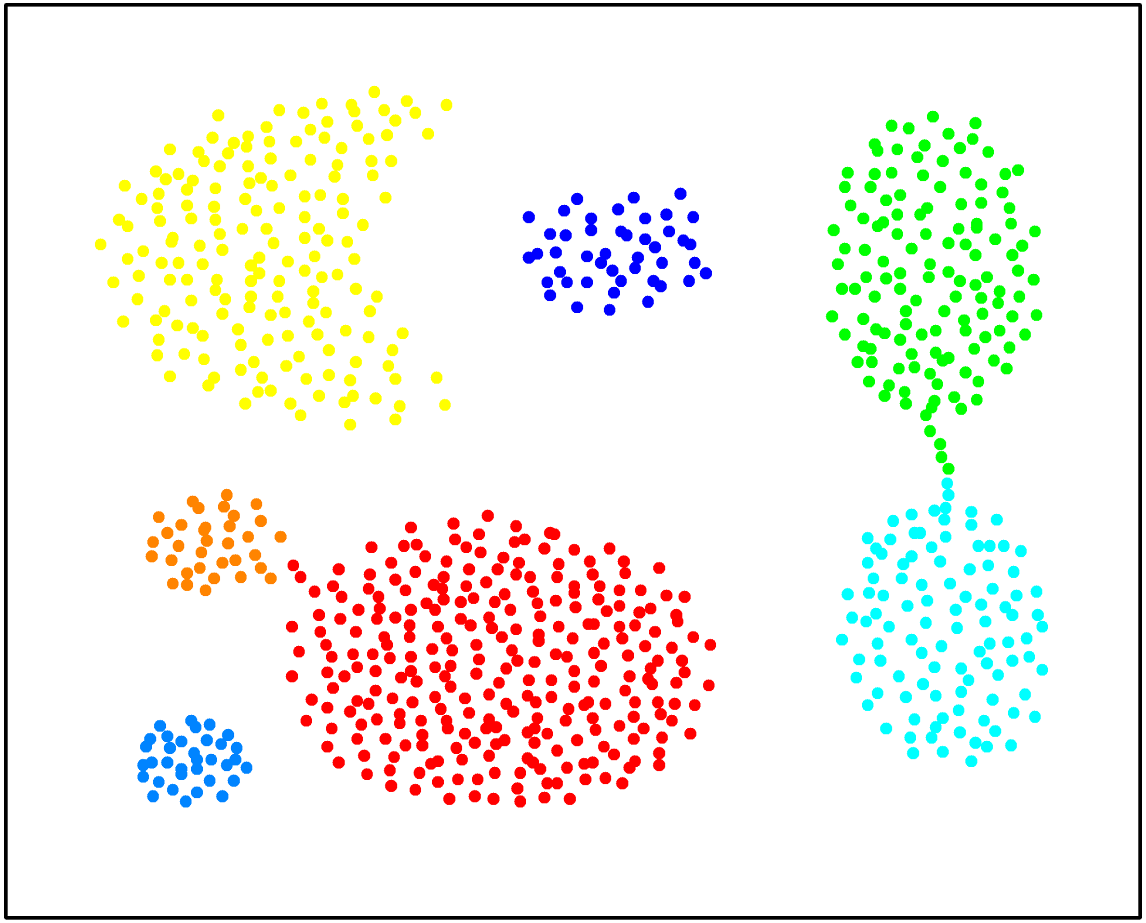
Motivation for clustering

Master in High Performance Computing

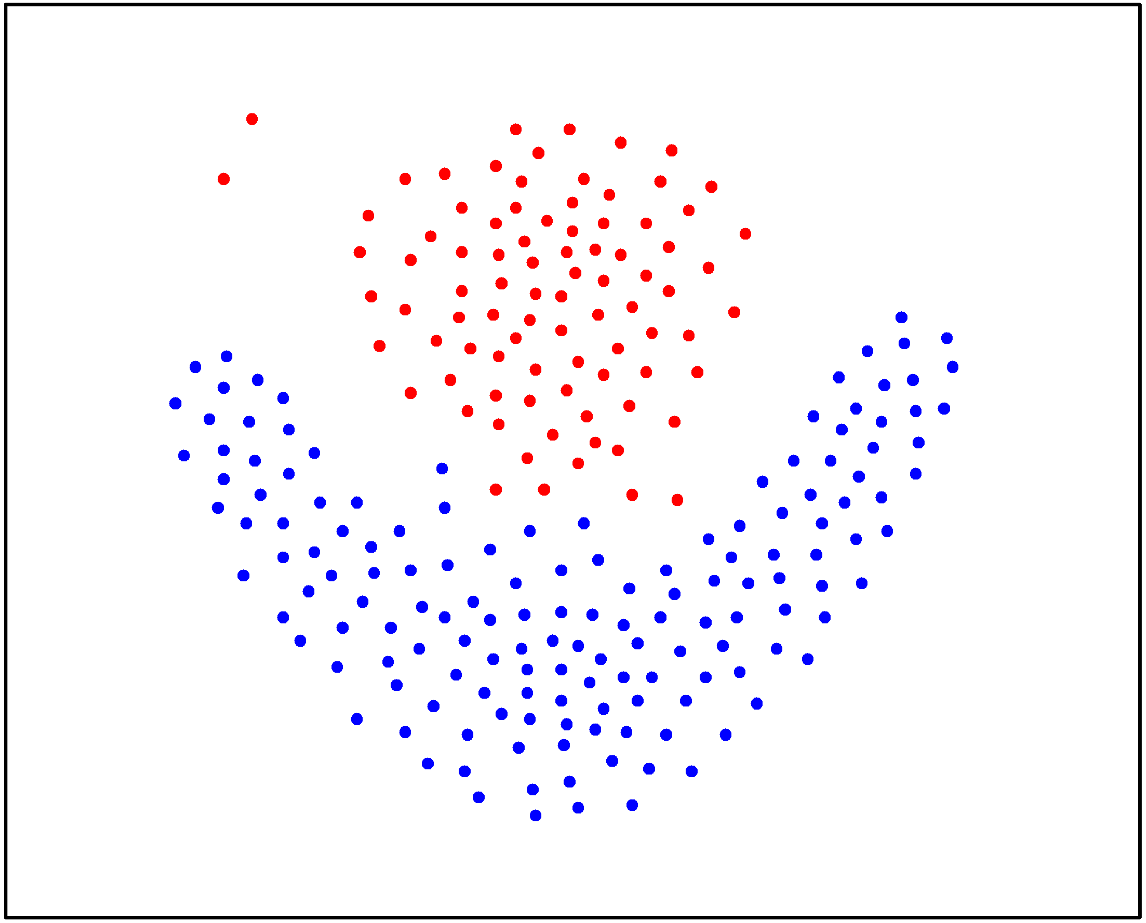
What is a cluster???



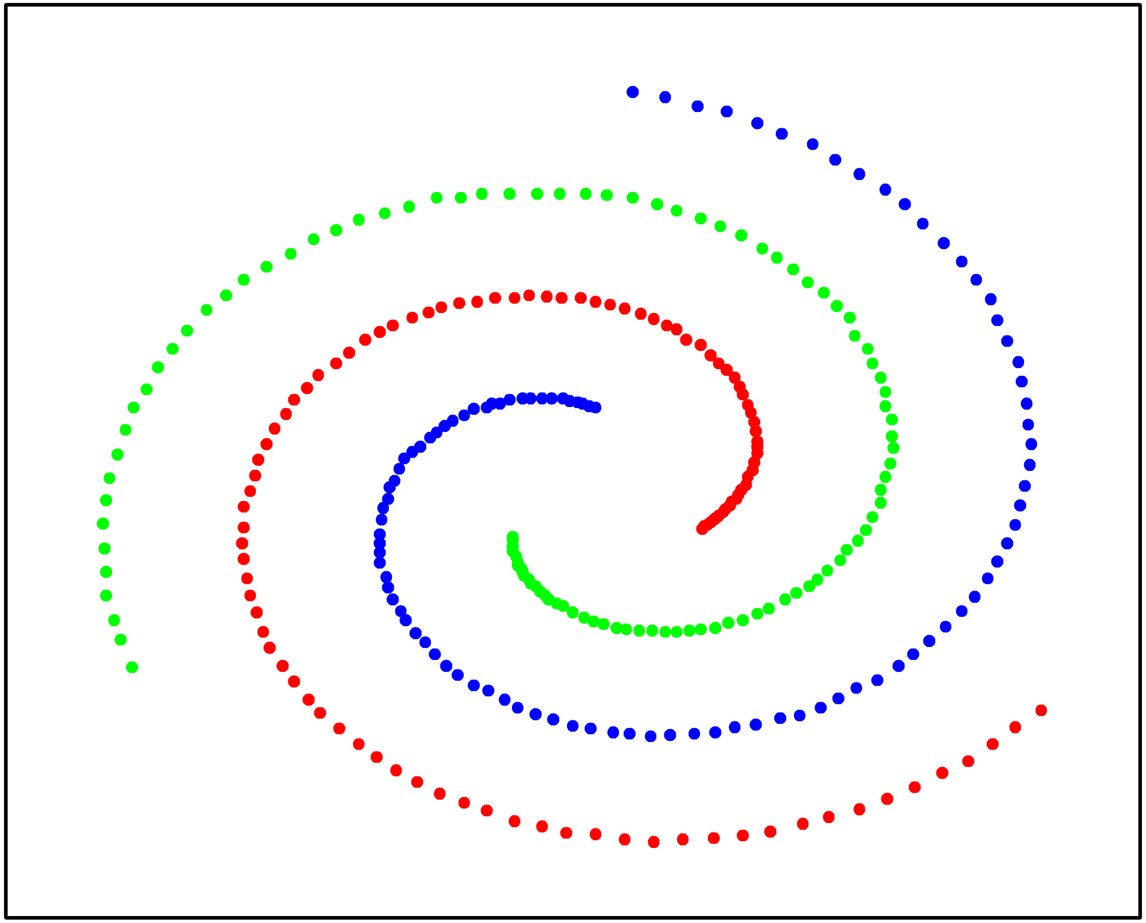
Other cluster examples…



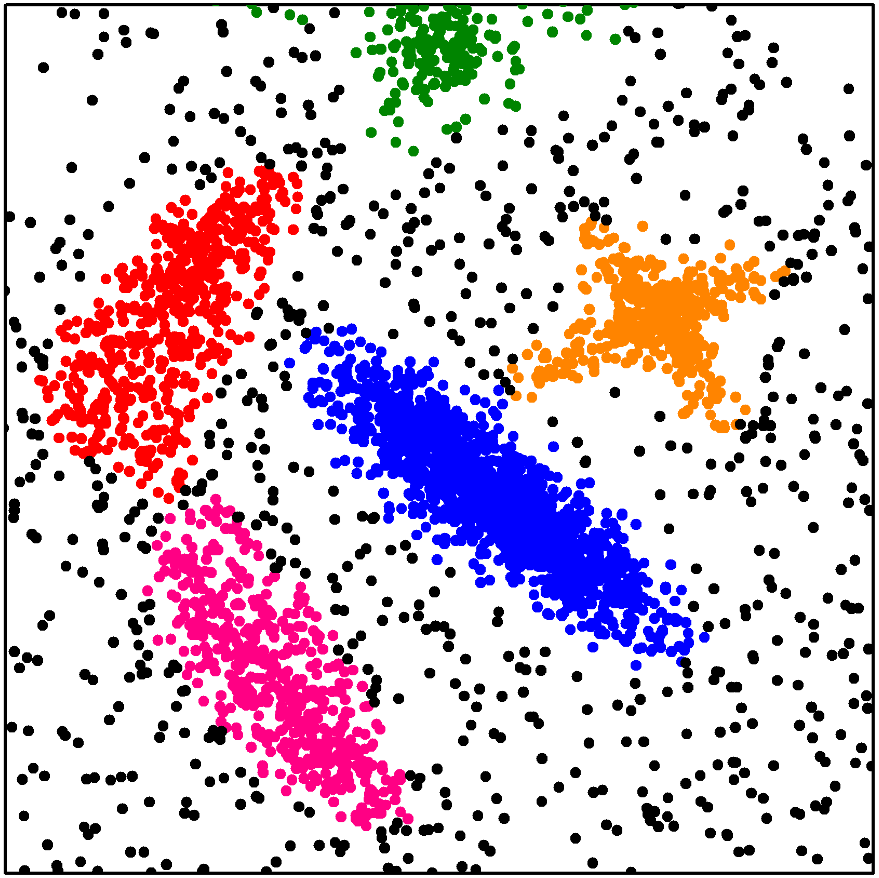
Other cluster examples…



Other cluster examples…

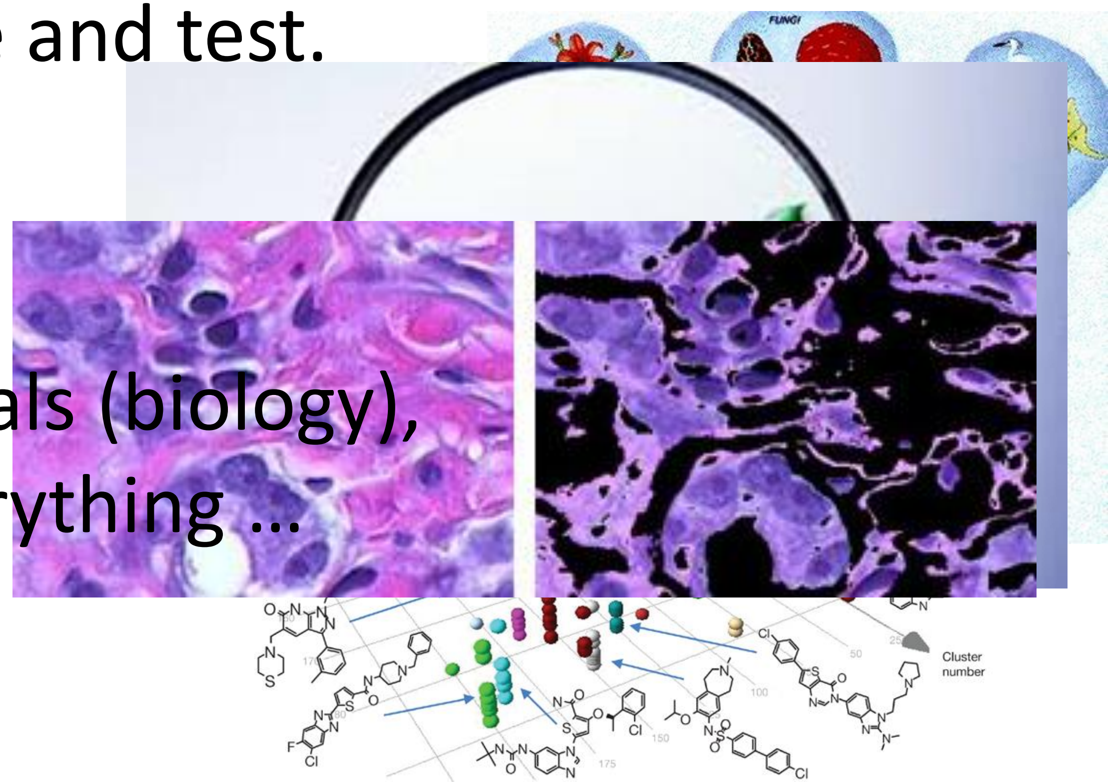


Other cluster examples…



Data mining technique that can be used for:

- Decide which set of drugs from a big library we should synthetize and test.
- WWW (googling…)
- Image recognition
- Classify plants, animals (biology), books(libraries), everything …



Why Clustering?

"Understanding our world requires conceptualizing the similarities and differences between the entities that compose it" (Tyron and Bailey, 1970).

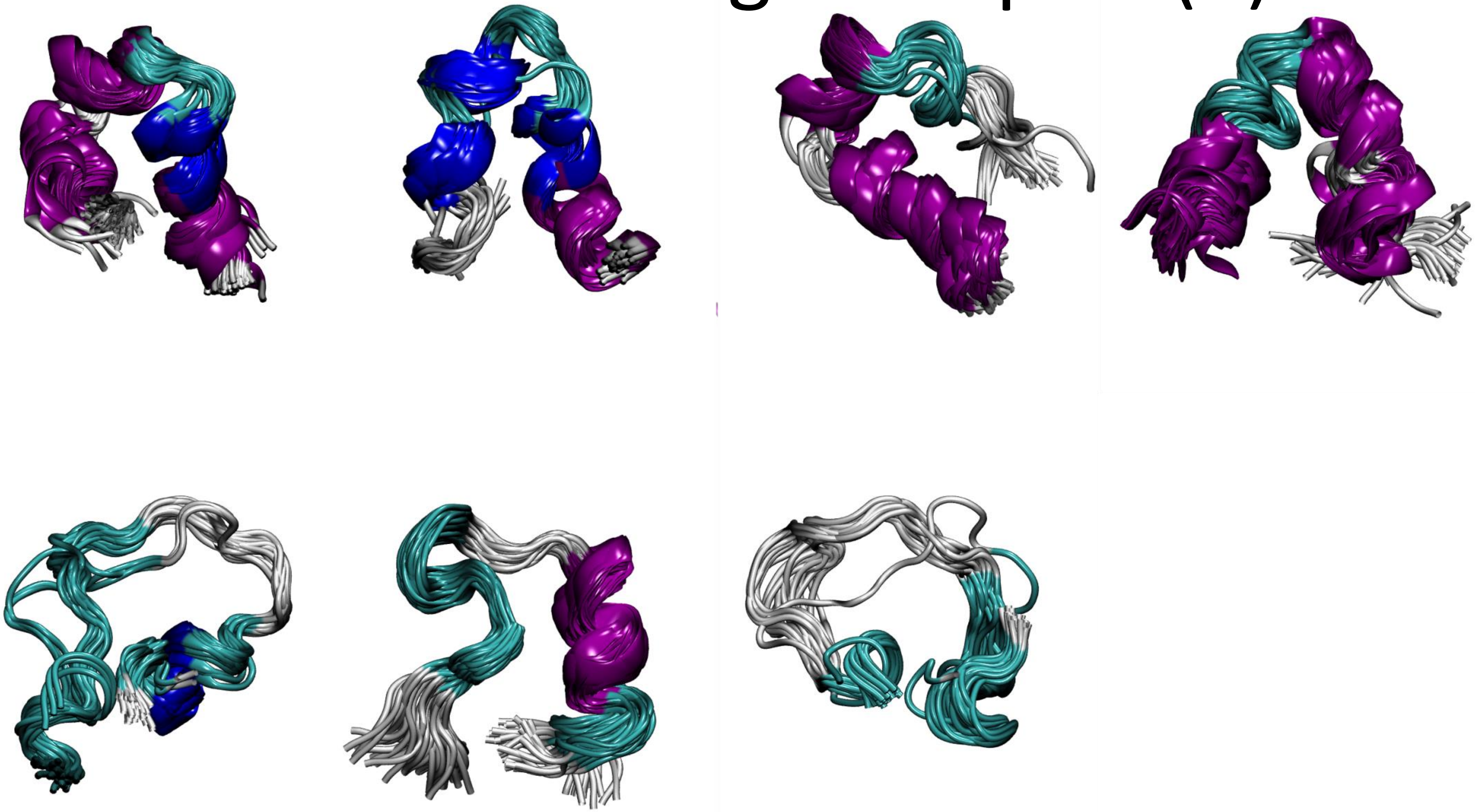
- Clustering ≠ Classification
  - Clustering generate groups
  - Classification generate groups <u>& labels</u>

My stone collection

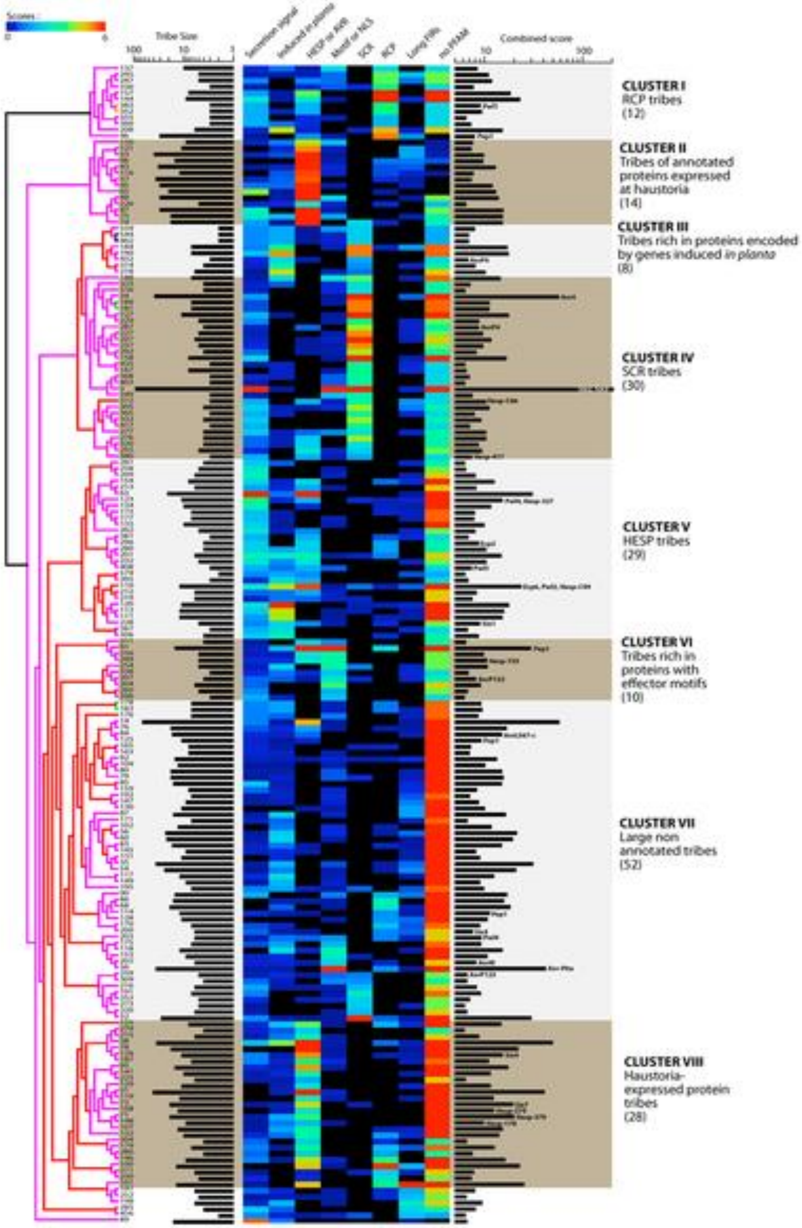
Cluster by light wavelength

Are they nice or ugly?
Color classification?

Some clustering examples (1)



Some clustering examples (2)



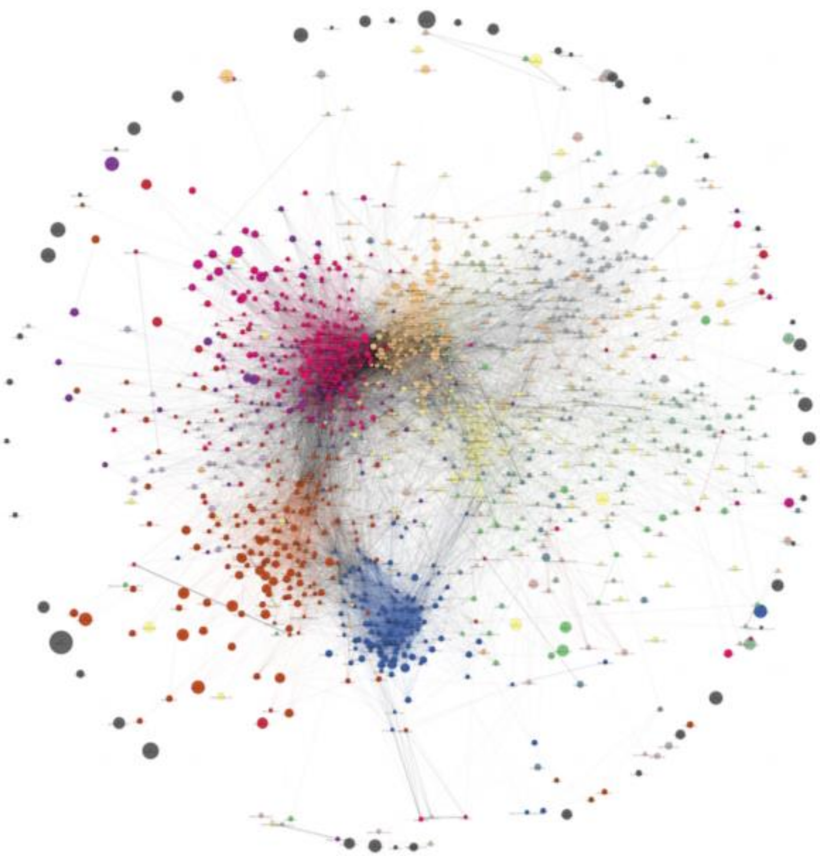
Hierarchical clustering of the secretome reveals clusters of secreted protein families as high priority effector candidates.

Saunders DGO, Win J, Cano LM, Szabo LJ, Kamoun S, et al. (2012) Using Hierarchical Clustering of Secreted Protein Families to Classify and Rank Candidate Effectors of Rust Fungi. PLoS ONE 7(1): e29847. doi:10.1371/journal.pone.0029847 http://journals.plos.org/plosone/article?id=info:doi/10.1371/journal.pone.0029847

By clustering and classifying plant proteins detect protein groups (families) that will probably act against mold

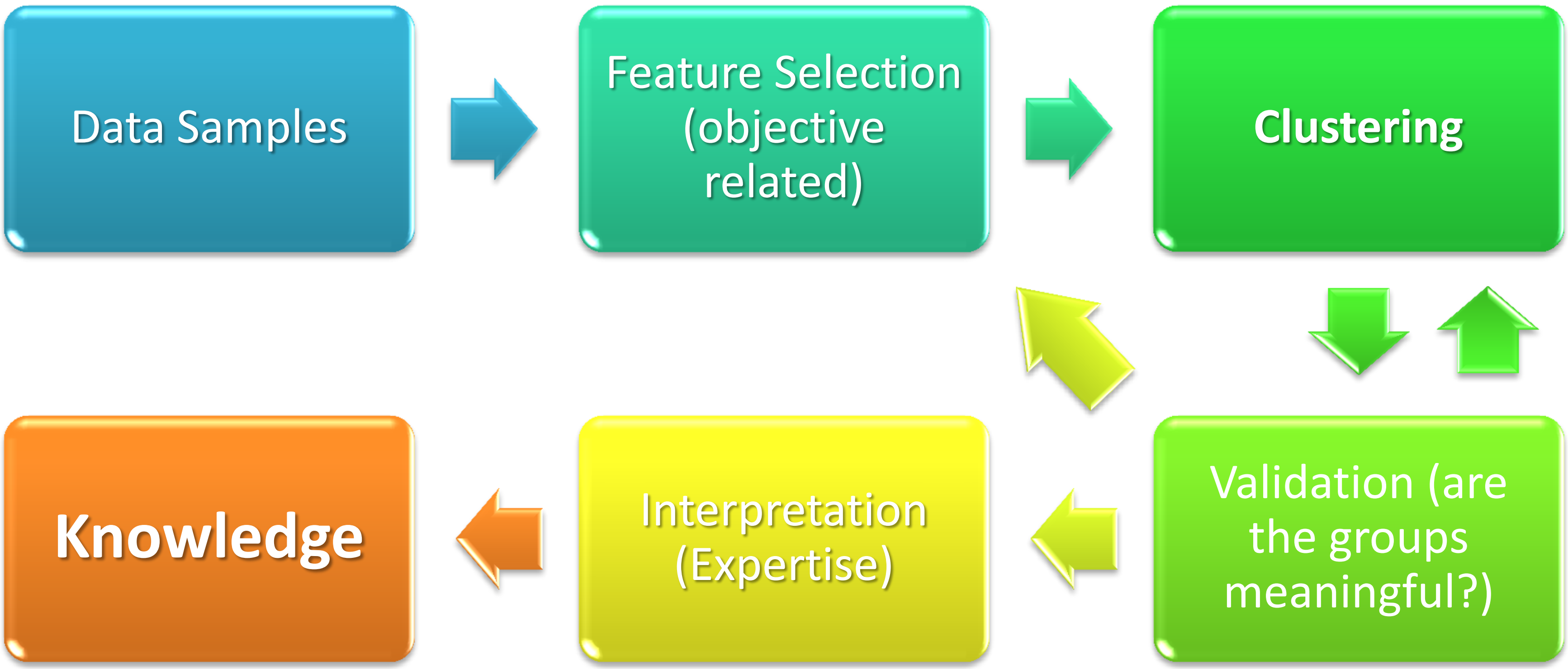
PLOS | ONE

Some clustering examples (3)



- Web search algorithms need to cluster.
- Not always known
- Cluster can be done in many ways (cluster the net by connections, cluster the results by text mining…)

Clustering procedure



Characteristics of the data sample

- Raw characteristics:
  - Number of features (Dimension)
  - Number of samples (Cardinality)
  - Type of features (Reals, integers, binary, qualitative)
- Learned characteristics:
  - Statistics (Variances, covariances, averages…)
  - Intrinsic dimension.
  - Clustering…

My stone collection (2)

- Weight
- Light wavelength
- Shape
- Volume
- Rugosity
- …

My stone collection (2)

Integer · Categorical · Real

Number	Weight (g)	Wavelenght (nm	Shape	Volume	Rugosity
1	20	450	spherical	0,2	1,04
2	33	698	cube	0,4	1,3
3	12	543	cube	0,2	0,8
4	70	691	spherical	1,1	1,9
5	120	465	cube	0,3	0,2
6	45	486	cube	0,8	0,3
7	136	504	cube	1,35	0,5
8	20	504	spherical	0,2	0,5
9	54	623	spherical	0,5	1
10	93	430	spherical	0,7	1
….					

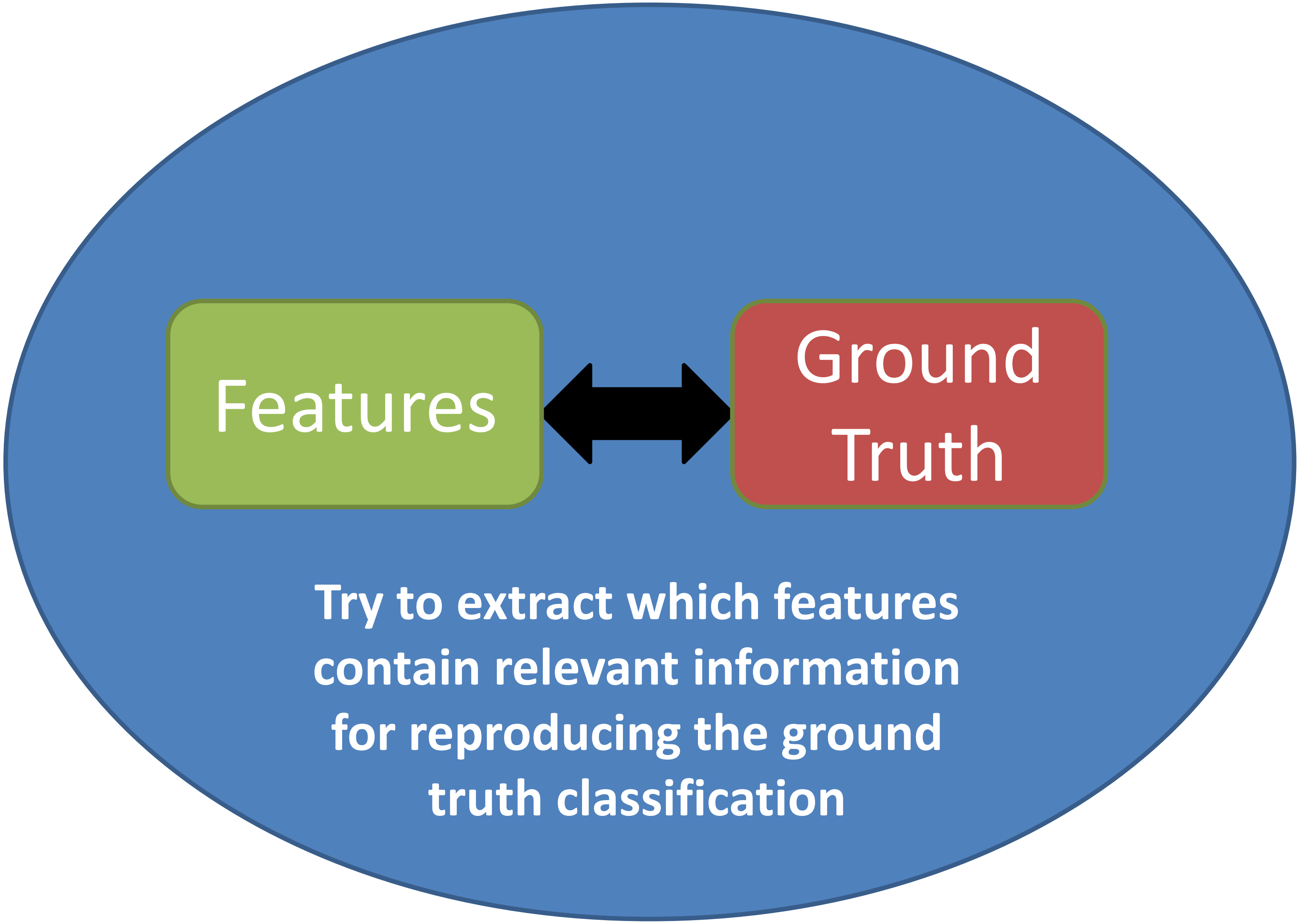
Feature Selection

Master in High Performance Computing

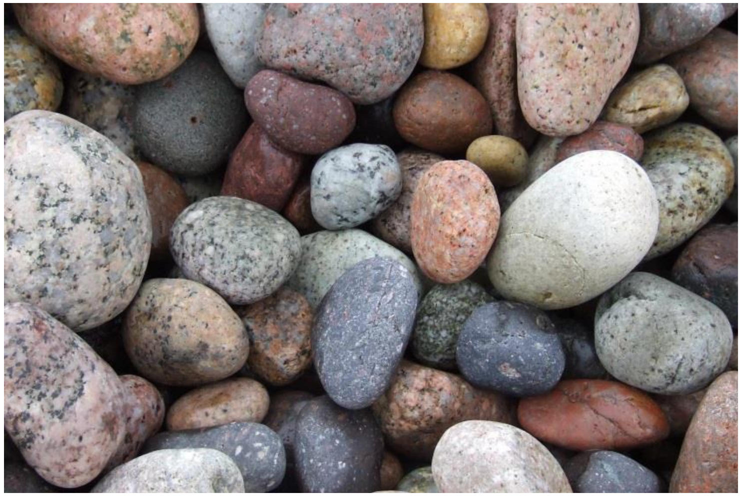
Feature Selection & Dimensionality reduction

- Highly related with the problem that we want to solve
- It may need feedback from the whole process
- Feature selection usually depends on labeled data while dimensionality reduction does not.
- Feature selection can be based on expertise…

Feature selection



My stone collection (3)



- Weight
- Light wavelength
- Shape
- Volume
- Rugosity
- …

Of course, if you are an expert, you already know which are the relevant features… but, I´m quite dummy…

Techniques for Feature Selection (1) Variable Ranking

- Which variables explain better the labels? Quantify it.

Techniques for Feature Selection (1)
Variable Ranking

- Which variables explain better the labels? Quantify it.
  - Linear correlation coefficient (R).

Techniques for Feature Selection (1)
Linear Correlation coefficient

$$R(i) = \frac{\sum_{k=1}^{m}\left(x_{k,i} - \overline{x_i}\right)(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{m}\left(x_{k,i} - \overline{x_i}\right)^2 \sum_{k=1}^{m}(y_k - \bar{y})^2}}$$

- For continuous variables and outputs.
- Goodness of linear fit
- Easy to extend to linear fit of functions of variables (i.e. take log of x).

Techniques for Feature Selection (1) Variable Ranking

- Which variables explain better the labels? Quantify it.
  - Linear correlation coefficient (R).
  - Single variable classifier. Jaccard index, F-score, etc.

Techniques for Feature Selection (1) Single variable classifier

Confusion matrix

R ↓ C →	Forest	Indust.	Urban	Water	Total
Forest	**68**	7	3	0	**78**
Indust.	12	**112**	15	10	**149**
Urban	3	9	**89**	0	**101**
Water	0	2	5	**56**	**63**
Total	**83**	**130**	**112**	**66**	**391**

- Based in the correspondence between the ground truth classification and the ones that comes from the single variable
- In some cases, requires labeling (assign the variable to a class).
- The confusion matrix can summarize some of them
- We will go in deeper detail when talking about external validation

Techniques for Feature Selection (1) Variable Ranking

- Which variables explain better the labels? Quantify it.
  - Linear correlation coefficient (R).
  - Single variable classifier. Jaccard index, F-score, etc.
  - Mutual information between variable and the target.

Techniques for Feature Selection (1) Information Theoretic Ranking

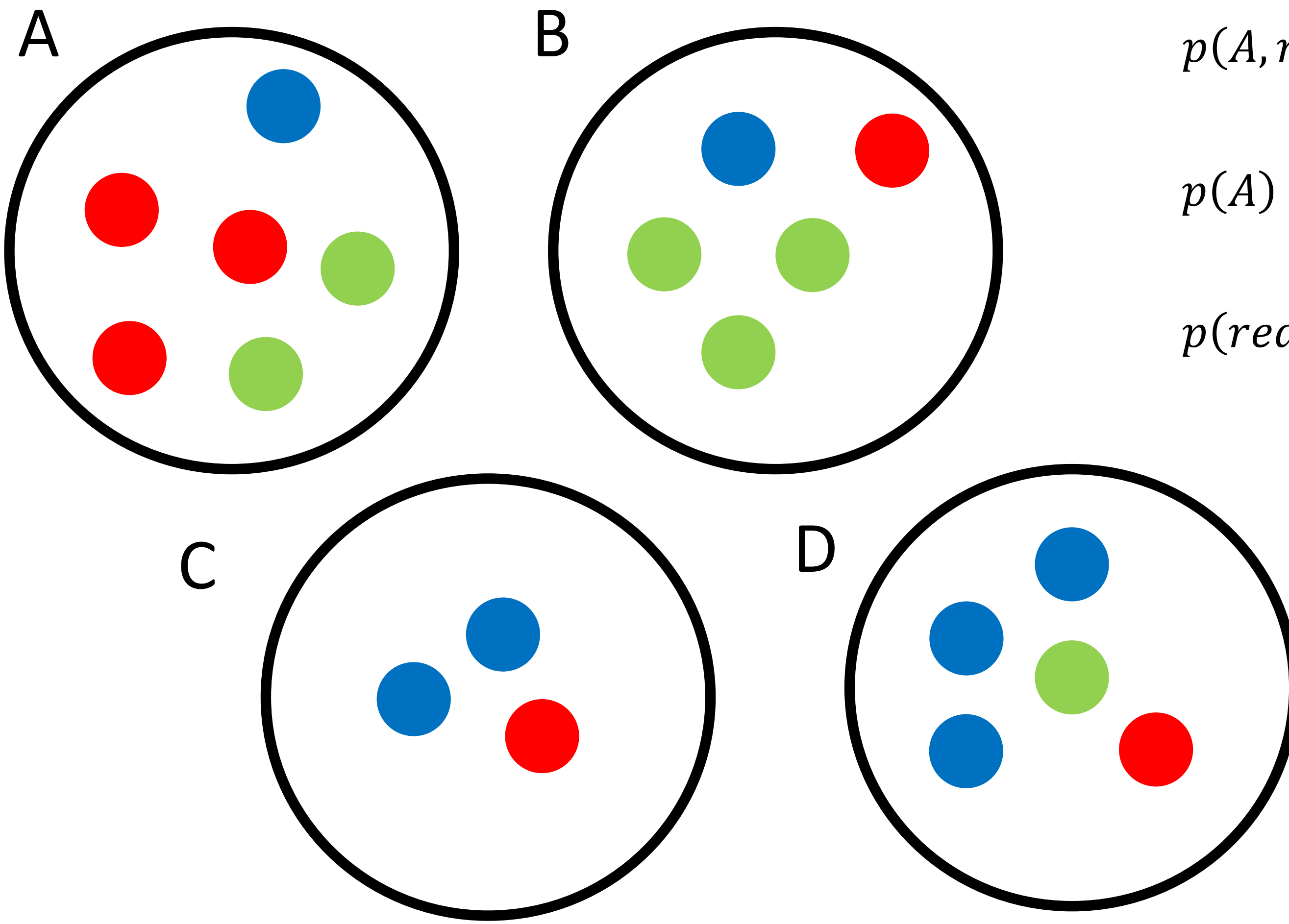
$$I(i) = \int_{x_i} \int_{y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)}$$

- A kind of single variable classifier.
- In non-continuous variables, the integrals become sums
- Extensible to continuous variables by non parametric density estimation
- Using a Gaussian distribution for estimating the density will lead to a similar criteria to the correlation coefficient.
- Is a formalization of the intuition that the higher the joint distribution, the higher the mutual information, i.e. the higher should it be in the rank.

Techniques for Feature Selection (1) Information Theoretic Ranking

- The probabilities, in a discrete case are estimated from frequency counts.
- Imagine a three class problem (red, green, blue) with a discrete variable that can take 4 values (A,B,C,D).
  - P(y) are 3 frequency counts.
  - P(x) are 4 frequency counts.
  - P(x,y) are 12 frequency counts.

Techniques for Feature Selection (1)
Information Theoretic Ranking



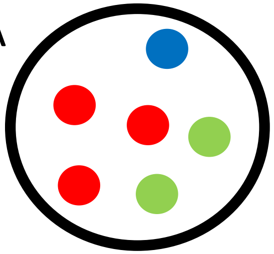
$$p(A, red) = \frac{3}{19}$$

$$p(A) = \frac{6}{19}$$

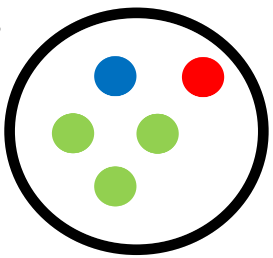
$$p(red) = \frac{6}{19}$$

Techniques for Feature Selection (1)
Information Theoretic Ranking

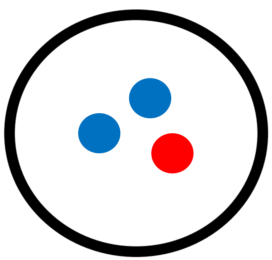
A



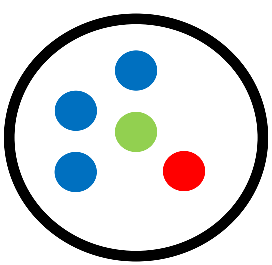
B



C



D



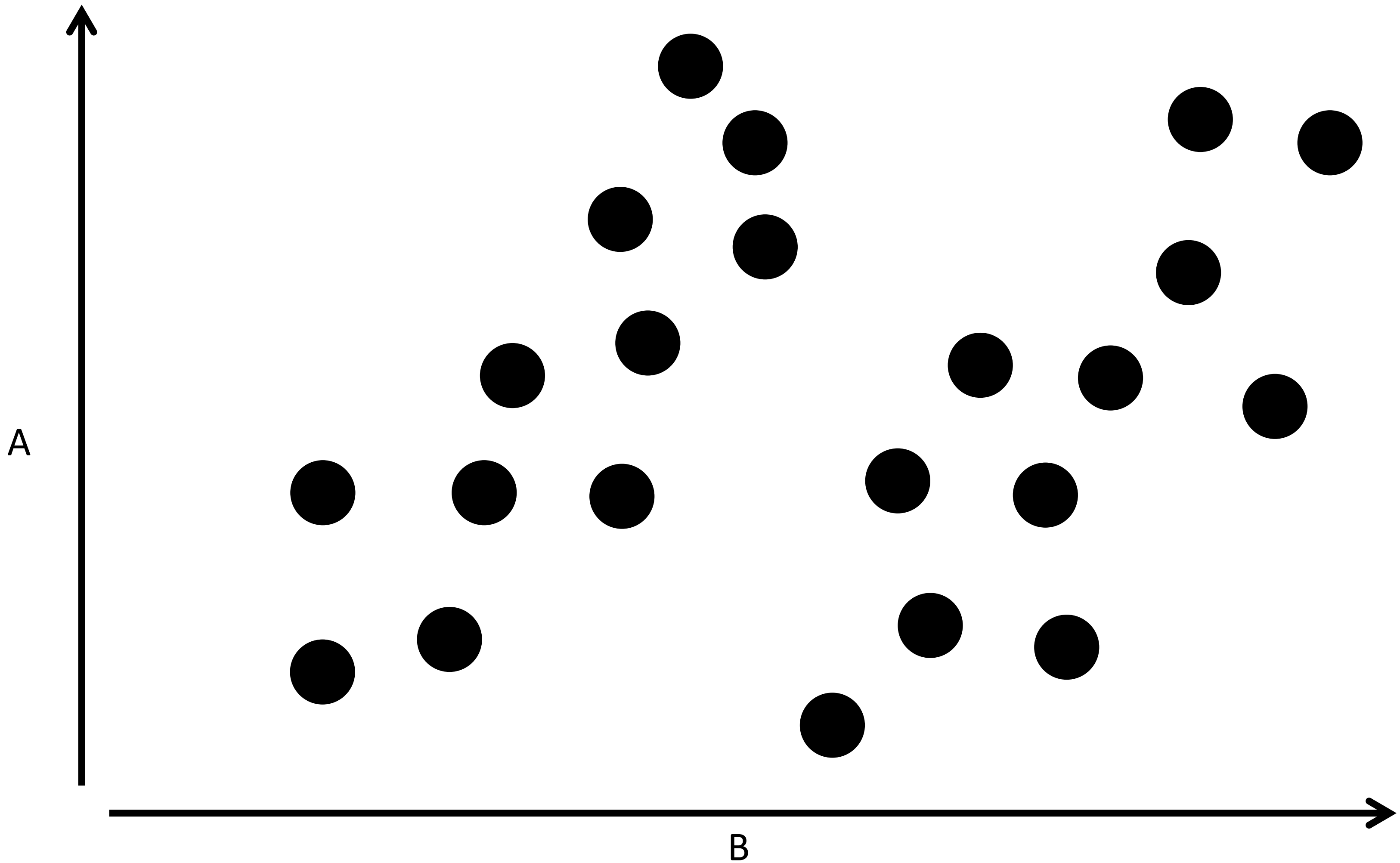
$$I = \frac{3}{19}\log\left(\frac{3/_{19}}{6/_{19}\,6/_{19}}\right) + \frac{1}{19}\log\left(\frac{1/_{19}}{6/_{19}\,7/_{19}}\right)$$
$$+ \frac{2}{19}\log\left(\frac{2/_{19}}{6/_{19}\,6/_{19}}\right) + \frac{1}{19}\log\left(\frac{1/_{19}}{5/_{19}\,6/_{19}}\right) +$$
$$\frac{1}{19}\log\left(\frac{1/_{19}}{5/_{19}\,7/_{19}}\right) + \frac{3}{19}\log\left(\frac{3/_{19}}{5/_{19}\,6/_{19}}\right) +$$
$$\frac{1}{19}\log\left(\frac{1/_{19}}{3/_{19}\,6/_{19}}\right) + \frac{0}{19}\log\left(\frac{0/_{19}}{3/_{19}\,7/_{19}}\right) +$$
$$\frac{2}{19}\log\left(\frac{2/_{19}}{3/_{19}\,6/_{19}}\right) + \frac{1}{19}\log\left(\frac{1/_{19}}{5/_{19}\,6/_{19}}\right) +$$
$$\frac{1}{19}\log\left(\frac{1/_{19}}{5/_{19}\,7/_{19}}\right) + \frac{3}{19}\log\left(\frac{3/_{19}}{5/_{19}\,6/_{19}}\right) \approx 0.17$$

Techniques for Feature Selection (1)

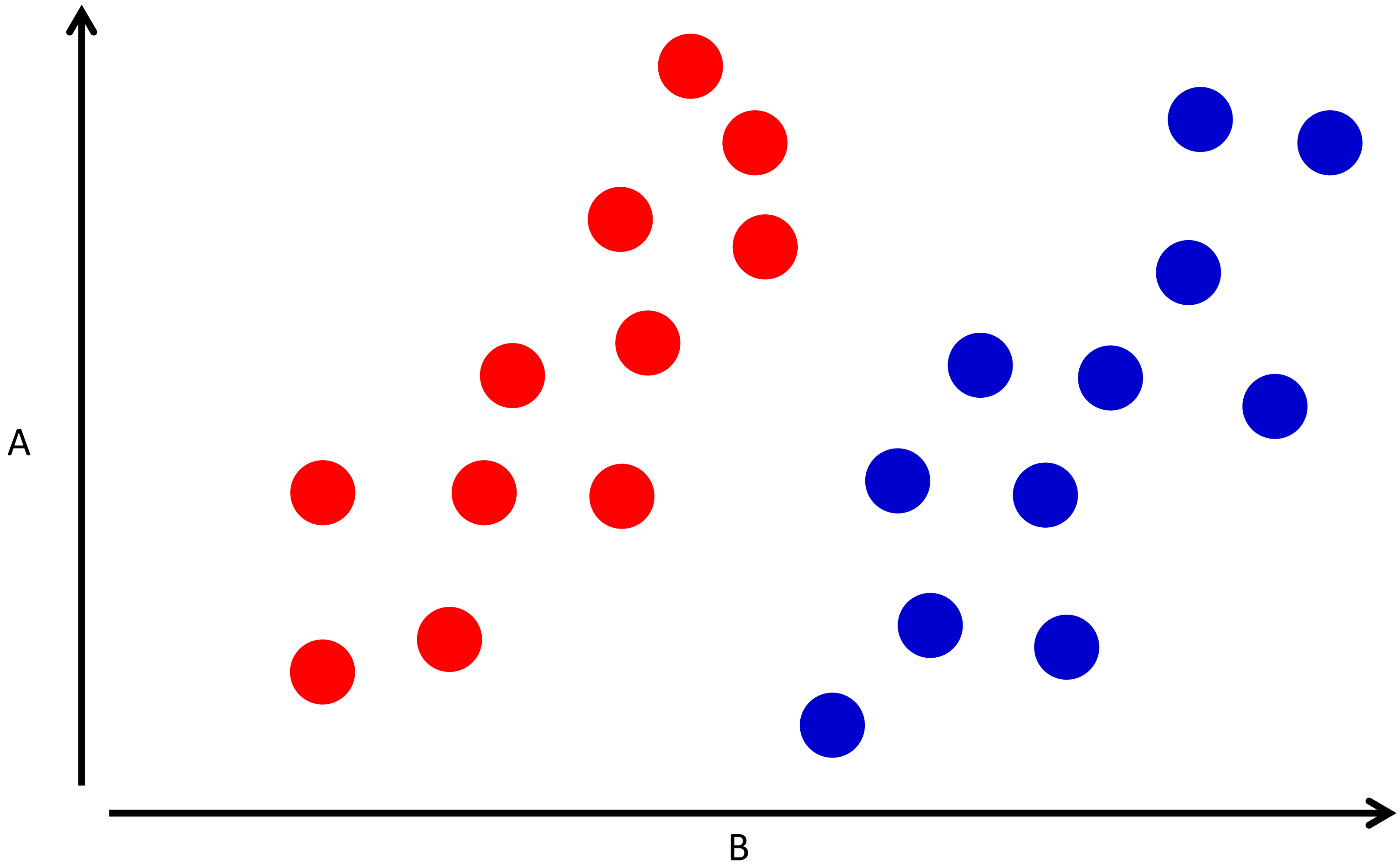
Some questions about Variable Ranking

- How to treat redundant variables?
  - Redundant variables get the same information but its combination can lead to noise reduction.
  - Correlation is a measure of redudancy
- A variable useless by itself can be useful together with others

Features useful only in combination



Features useful only in combination: which combination of A and B will be useful for separate the two clusters?



Techniques for Feature Selection (2) Subset selection

- Wrappers: Use the predicting power of a given learning machine to assess the usefulness of a given subset
  - How to search the space? (Brute force is NP hard.)
  - How to assess the performance of the prediction.
  - Which learning machine use.

What if we don't have a response function?

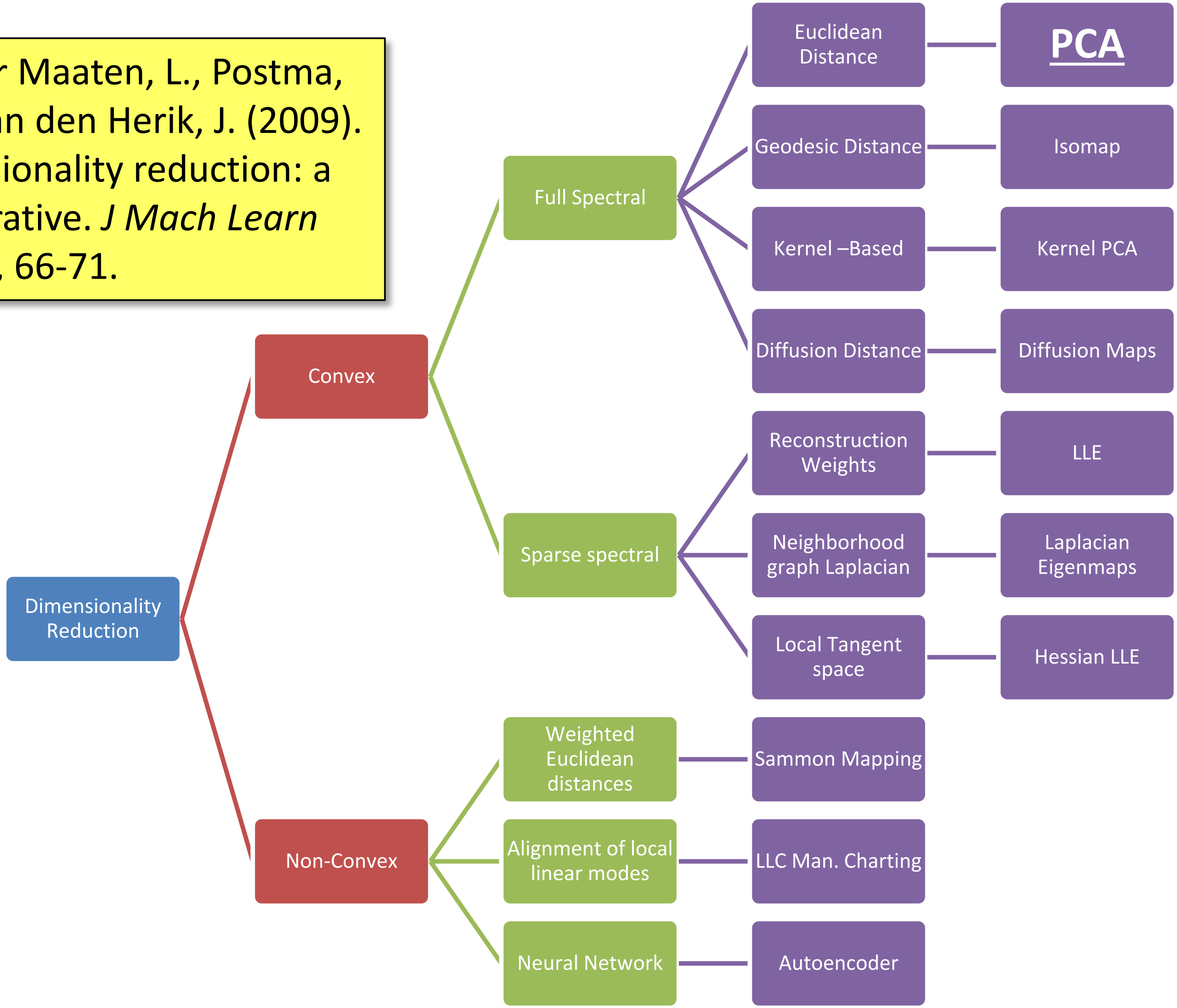
Dimensionality
Reduction

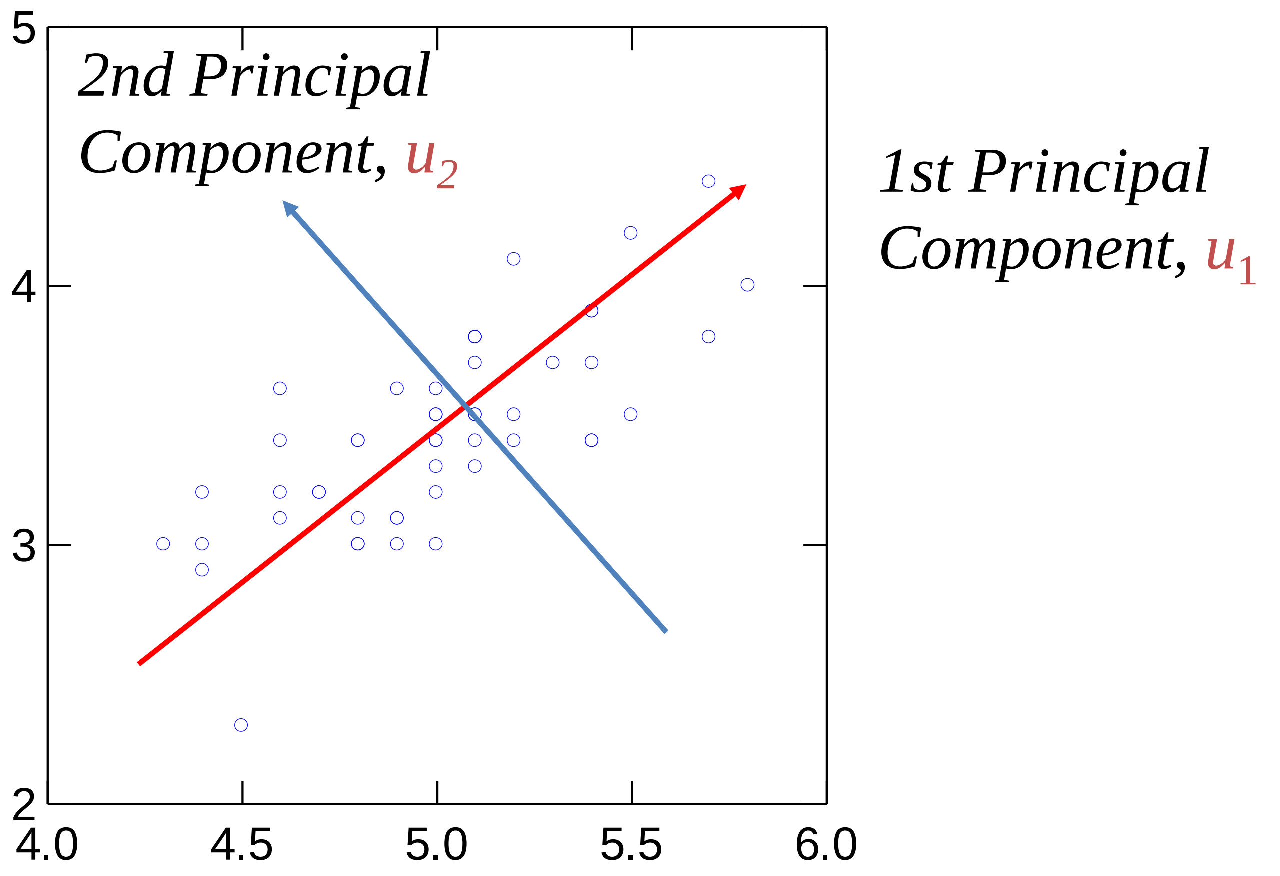
Dimensionality Reduction

Master in High Performance Computing

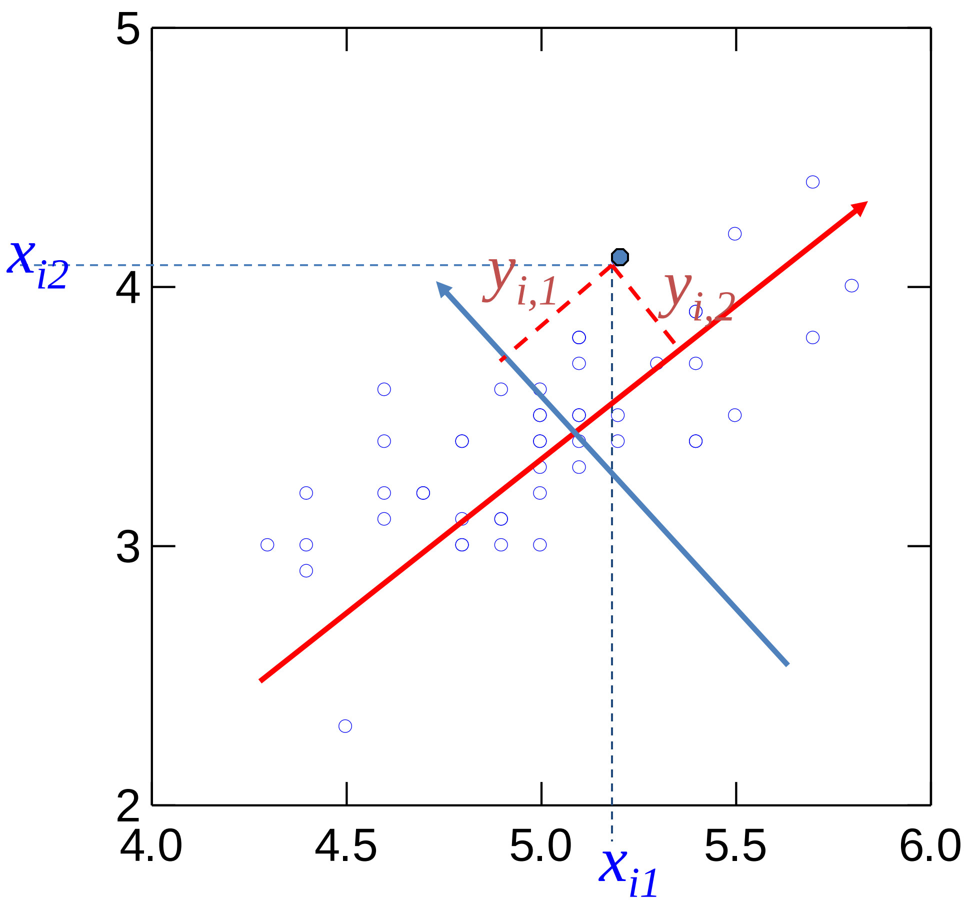
Techniques for Dim. reduction

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, *10*, 66-71.

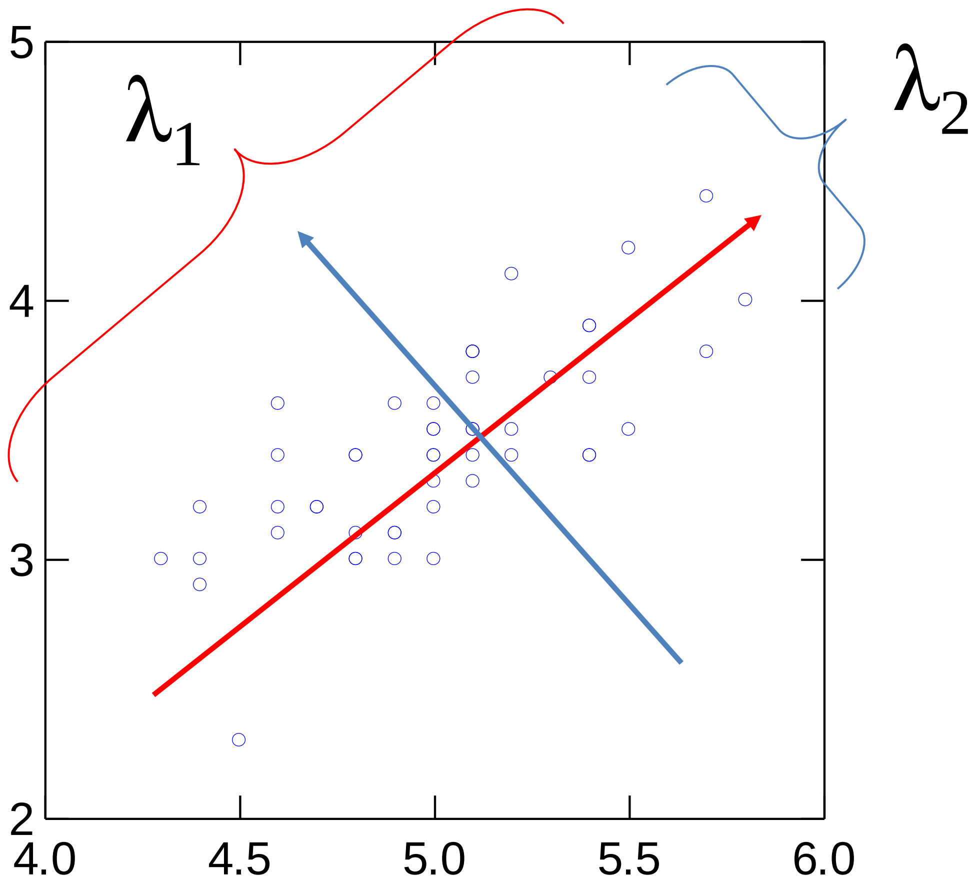



2nd Principal Component, u_2
1st Principal Component, u_1
5
4
3
2
4.0
4.5
5.0
5.5
6.0

PCA Scores



PCA Eigenvalues

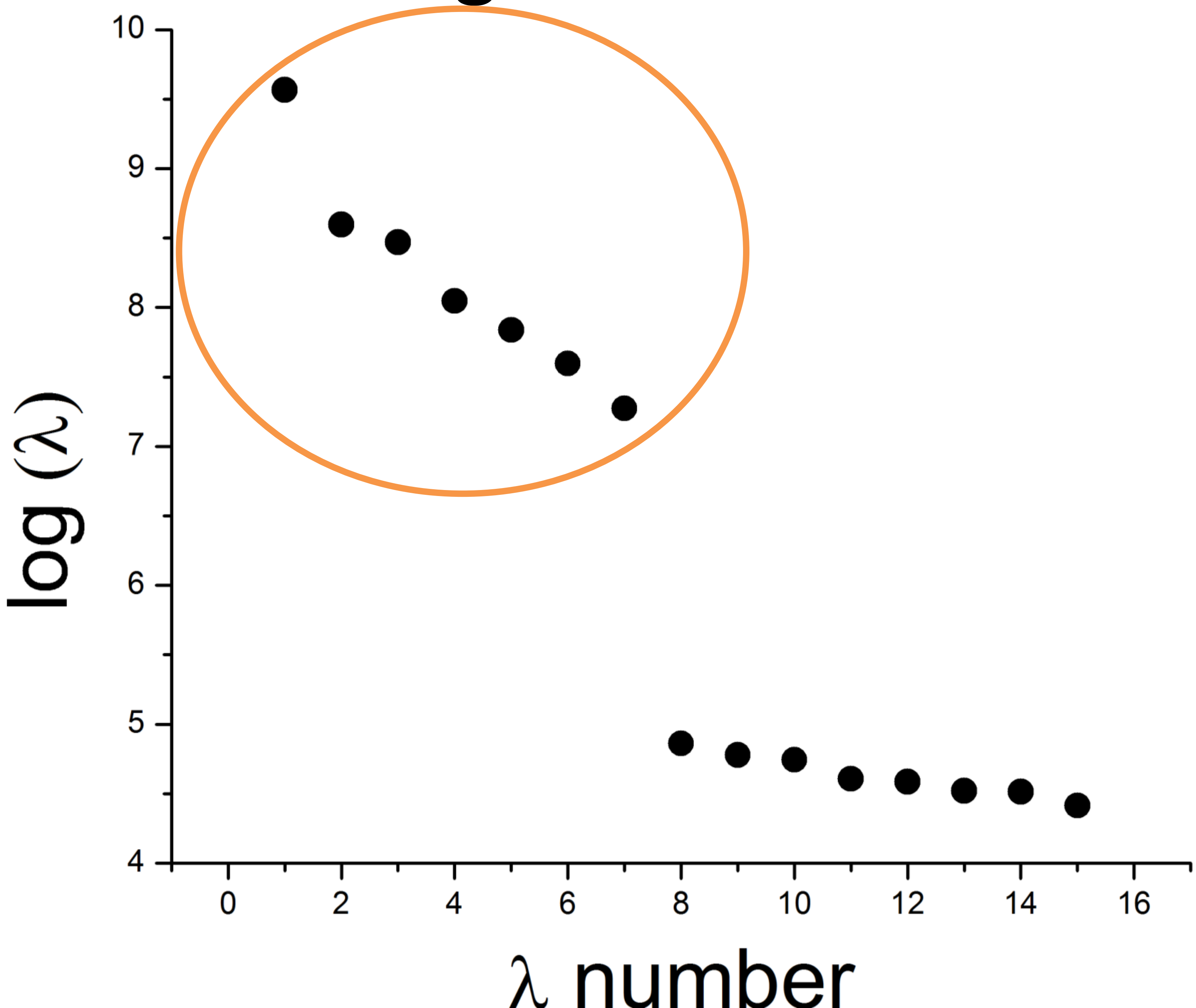


PCA. How to…

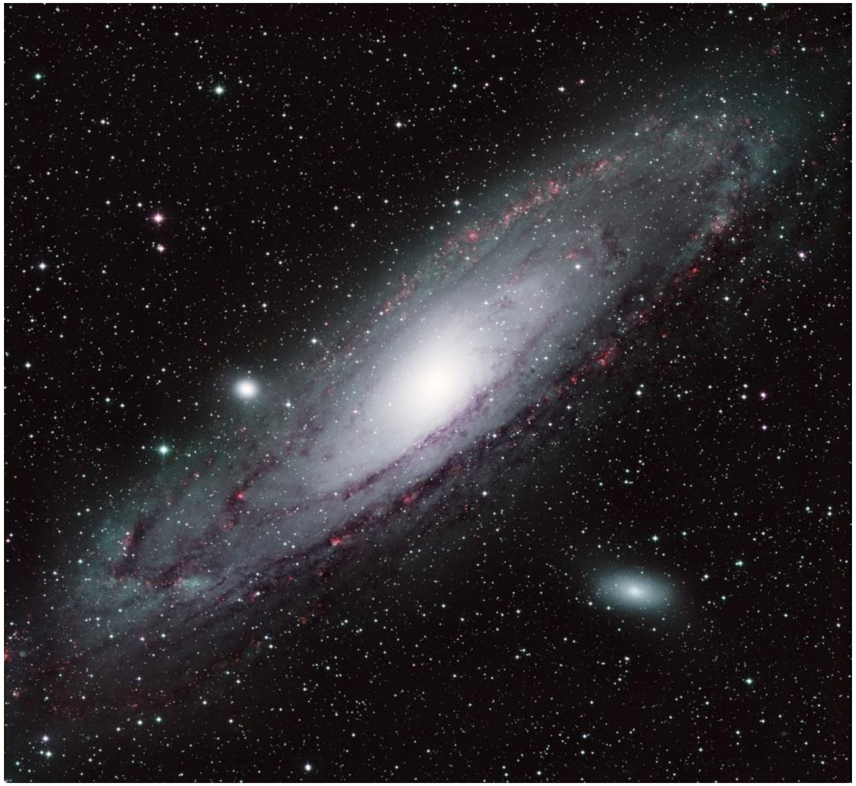
Given n observations $\{x_1, x_2, \cdots, x_n\}$ of m-dimensional column vectors

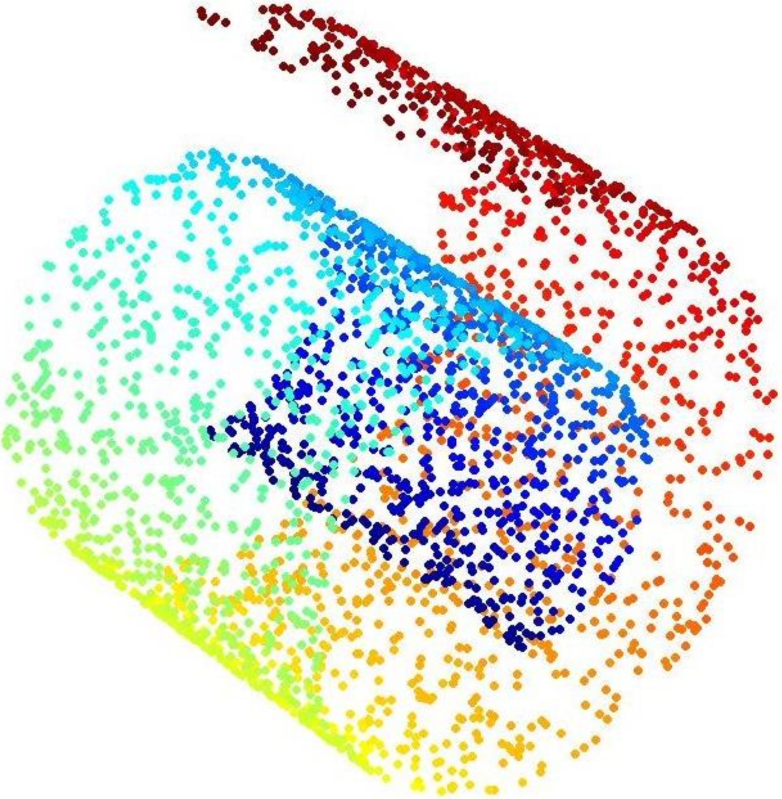
1. Compute the mean vector $\mu = \frac{1}{n}\sum_1^n x_i$
2. Compute the covariance matrix by MLE $\mathbb{C} = \frac{1}{n}\sum_1^n (x_i - \mu)(x_i - \mu)^T$
3. Compute the eigenvalue/eigenvector pairs (λ_i, u_i) of $\mathbb{C}$ with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_m$ (Diagonalize/SVD) and sort.
4. Choose the number of dimensions (d) in which project the data
5. Project by ${y_i}^{(j)} = {x_i}^T u_j$

Choosing the number of PC's



Intrinsic dimension example

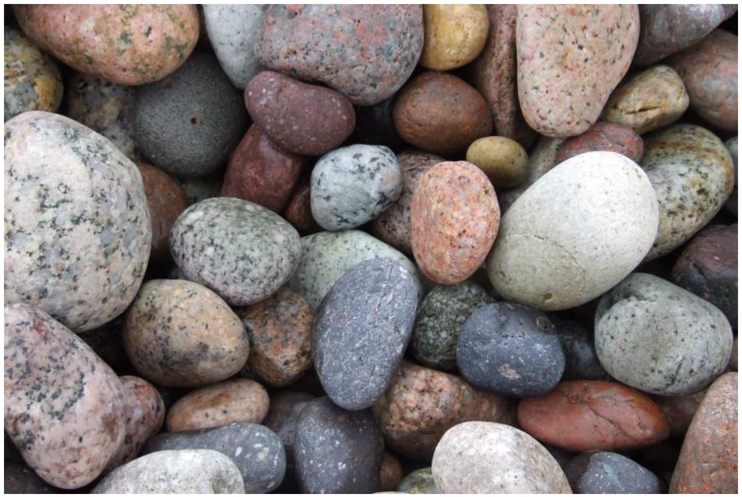




PCA problems

- Poorly suited for non-linear transformations
- Not able of capturing invariances
- By using the covariance along the whole dataset, is poor suited for problems not well described by this parameter.
- Not scale invariant
- Focused on large pairwise distances (!!!)

My stone collection (4)



- Weight
- Light wavelength
- Shape
- Volume
- Rugosity
- …

Will it recover the density as an important feature for mineral recognition?
Density=Weight/Volume

Similarities and distances

Master in High Performance Computing

Similarity and Distances

- Clustering tries to separate data "*naturally*", in such a way that *similar* elements lay in the same cluster while *dissimilar* elements belong to a different one
- *Similarity* (S_{ij}) is a pairwise function of the features of the elements i and j.
- In terms of space, it can be thought that similar elements are near while dissimilar are far. So many times it is useful to talk about "distances between elements" (D_{ij})
- Its definition depends on the nature of the features

Similarity and Distances almost the same but…

A (metric) distance must accomplish:

1. Symmetry: $d(x, y) = d(y, x)$
2. Non-negativity: $d(x, y) \geq 0$
3. Identity of indiscernibles: $d(x, y) = 0 \Leftrightarrow x = y$
4. Triangle inequality: $d(x, z) + d(z, y) \geq d(x, y)$

We have to take this into account for some clustering algorithms

Quantitative Features: *Metric Distances*

- Minkowski distance:

$$d_{ij} = \left(\sum_{l=1}^{d} \left|x_{il} - x_{jl}\right|^{p}\right)^{1/p}$$

  - Special cases are:
    - Euclidean (p=2)
    - City-block (p=1)
    - Sup (p→∞) . Eqv to $d_{ij} = \max_{l} \left|x_{il} - x_{jl}\right|$

- Mahalanobis distance

$$d_{ij} = \left(x_i - x_j\right)^T \mathbb{C}^{-1} \left(x_i - x_j\right)$$

Invariant with respect to any non-singular linear transformation of the coordinates. $\mathbb{C}$ is the covariance matrix.

Quantitative Features: *Not metric distances*

- Pearson correlation:

$$d_{ij} = \frac{1-r_{ij}}{2};\ r_{ij} = \frac{\sum_{k=1}^{m}\left(x_{k,i}-\overline{x_i}\right)\left(x_{k,j}-\overline{x_j}\right)}{\sqrt{\sum_{k=1}^{m}\left(x_{k,i}-\overline{x_i}\right)^2 \sum_{k=1}^{m}\left(x_{k,j}-\overline{x_j}\right)^2}}$$

- Point Symmetry distance:

$$d_{ij} = \min_{k \neq i} \frac{\left\|\left(x_i - x_j\right) + \left(x_k - x_j\right)\right\|}{\left\|\left(x_i - x_j\right)\right\| + \left\|\left(x_k - x_j\right)\right\|}$$

- Cosine similarity:

$$S_{ij} = \frac{{x_i}^T \cdot x_j}{\left\|x_j\right\| \left\|x_i\right\|}$$

Qualitative Features

- Jaccard similarity:

$$S_{ij} = \frac{|i \cap j|}{|i \cup j|}$$

$$i = \boxed{1}000\boxed{1}00\boxed{1} \; j = 0\boxed{1}00\boxed{1}0\boxed{1}\boxed{1}; S_{ij} = \frac{2}{5}$$

- Hamming distance:

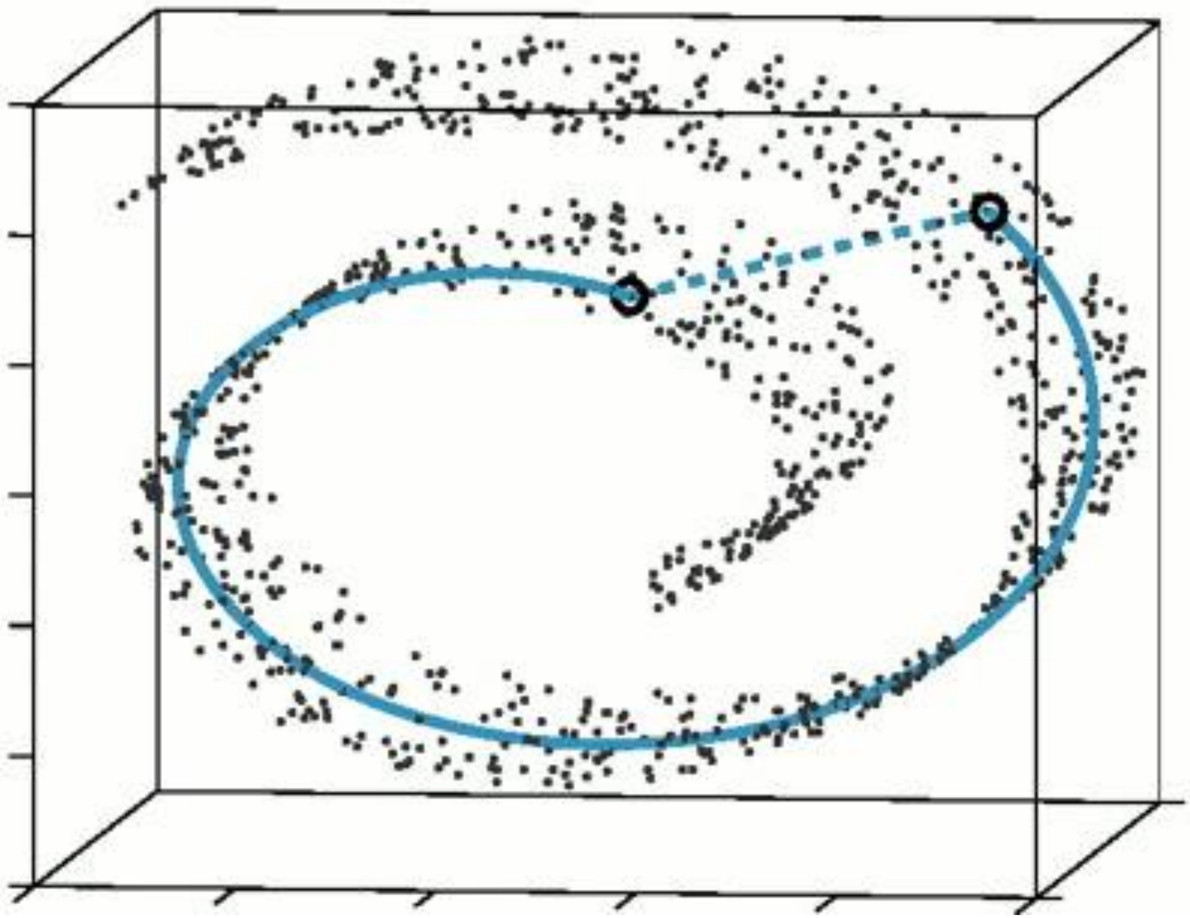
$$D_{ij} = |i \cup j| - |i \cap j|$$

i.e. minimum number of changes that you need to turn i in j.

More complicated distances

- Working in the metric can extremely simplify the clustering work.
- A good metric can dramatically improve the performance of an algorithm.
- However, usually they need to compute a simplest distance as starting point.
- Example: Geodesic distance

Geodesic distance



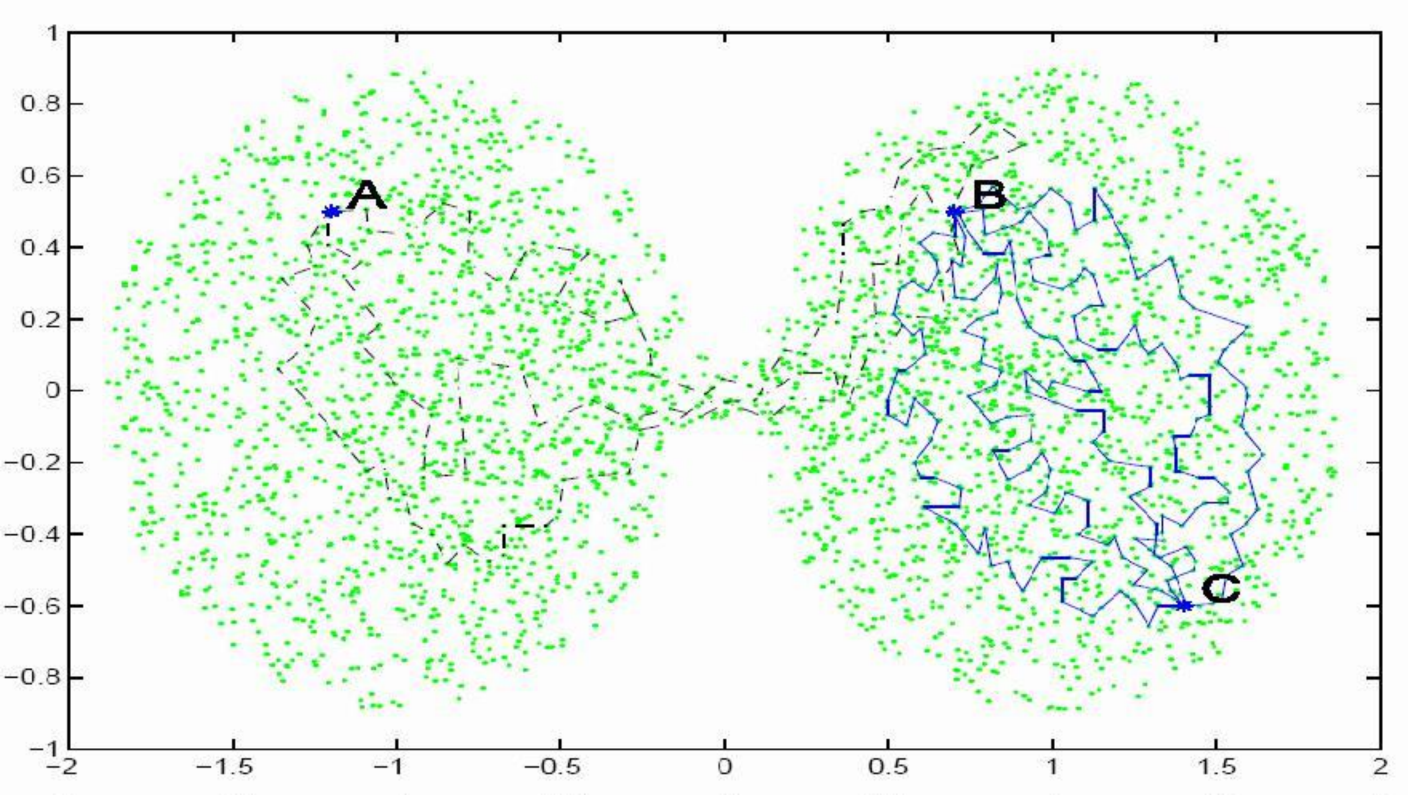
Geodesic distance

- Determine the neighbors.
  - All points in a fixed radius.
  - K nearest neighbors
- Construct a neighborhood graph.
  - Each point is connected to the other if it is a K nearest neighbor.
  - Edge Length equals the Euclidean distance
- Compute the shortest paths between two nodes
  - Floyd’s Algorithm
  - Dijkstra’s Algorithm

Geodesic distance

- Unstable?
- Only free parameter is : How many neighbors?
- How to choose neighborhoods?
- Susceptible to short-circuit errors if neighborhood is larger than the folds in the manifold.
- If small we get isolated patches.

Diffusion vs. Geodesic Distance



$D_{geod.}(A,B) \approx D_{geod.}(C,B)$ $\qquad$ $D_m(A,B) >> D_m(C,B)$

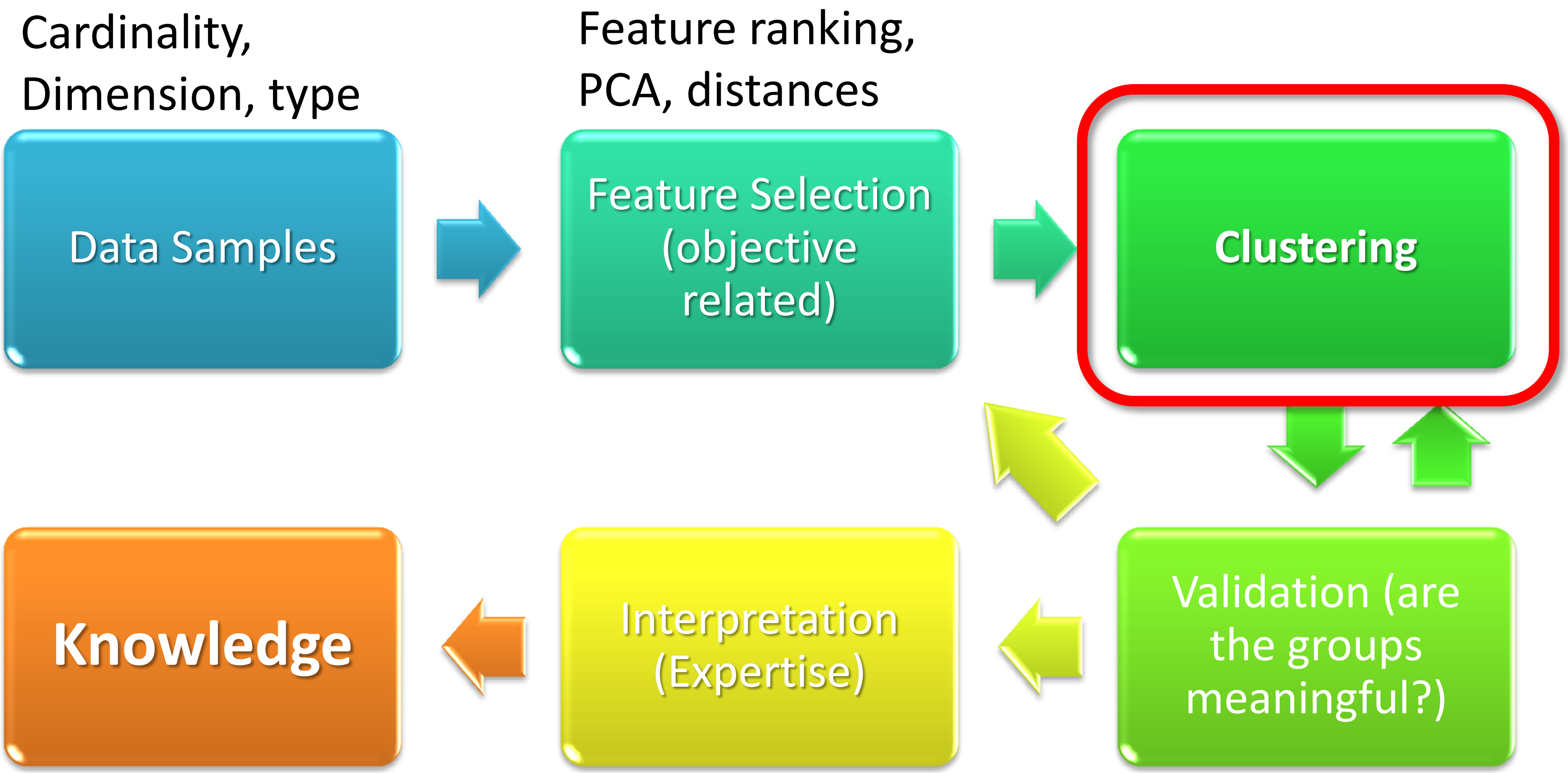
Flat, hierarchical and fuzzy clustering

Master in High Performance Computing

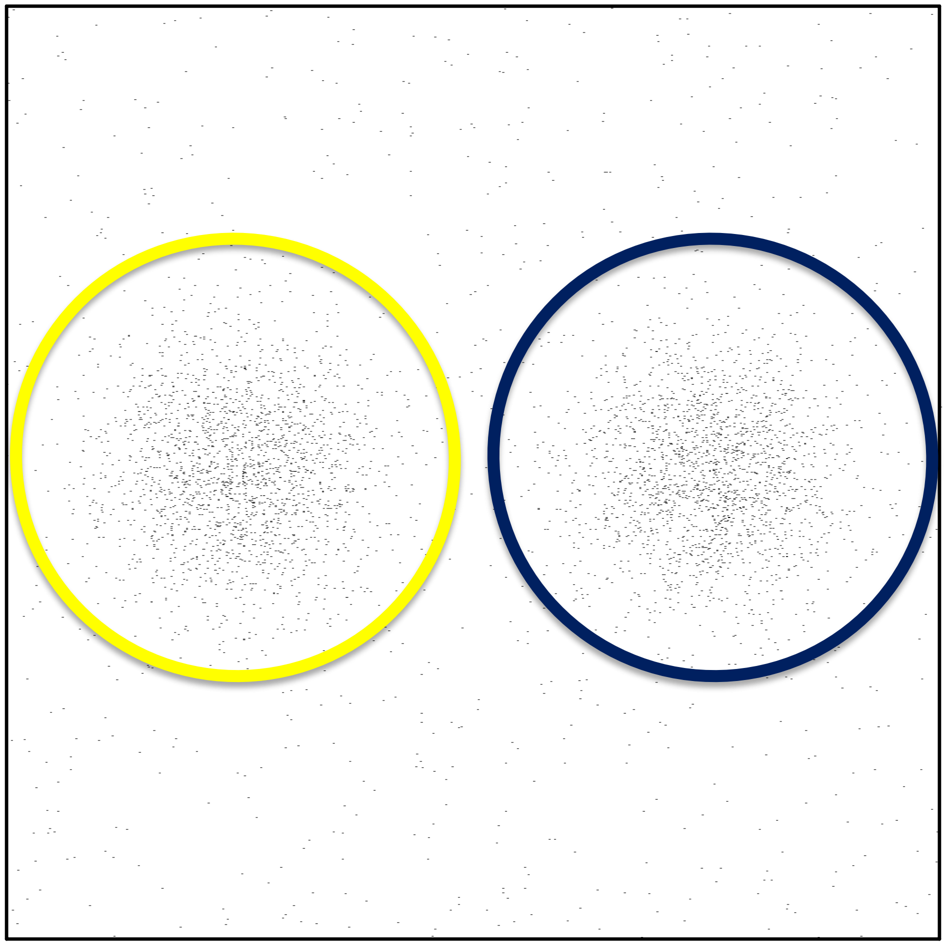
Previously…

- **Data samples:** Characterized by the number of elements (cardinality), the number of features (dimensionality) and the type of these features.
- **Feature selection:** We employ it when a ground truth is available. The method of choice depends on the nature of the ground truth and the nature of the features.
- **Dimensionality reduction:** Does not require a ground truth. It usually simplifies the data and allows a more efficient representation or treatment. Although we focused on PCA, there are many alternatives.
- **Similarities and distances:** a pairwise measure of how similar are the elements of the data set. Are the basis for many clustering methods and they should be selected according with the nature of the dataset. More complicated distances (like the geodesic distance) are available and can be useful in difficult analyses.

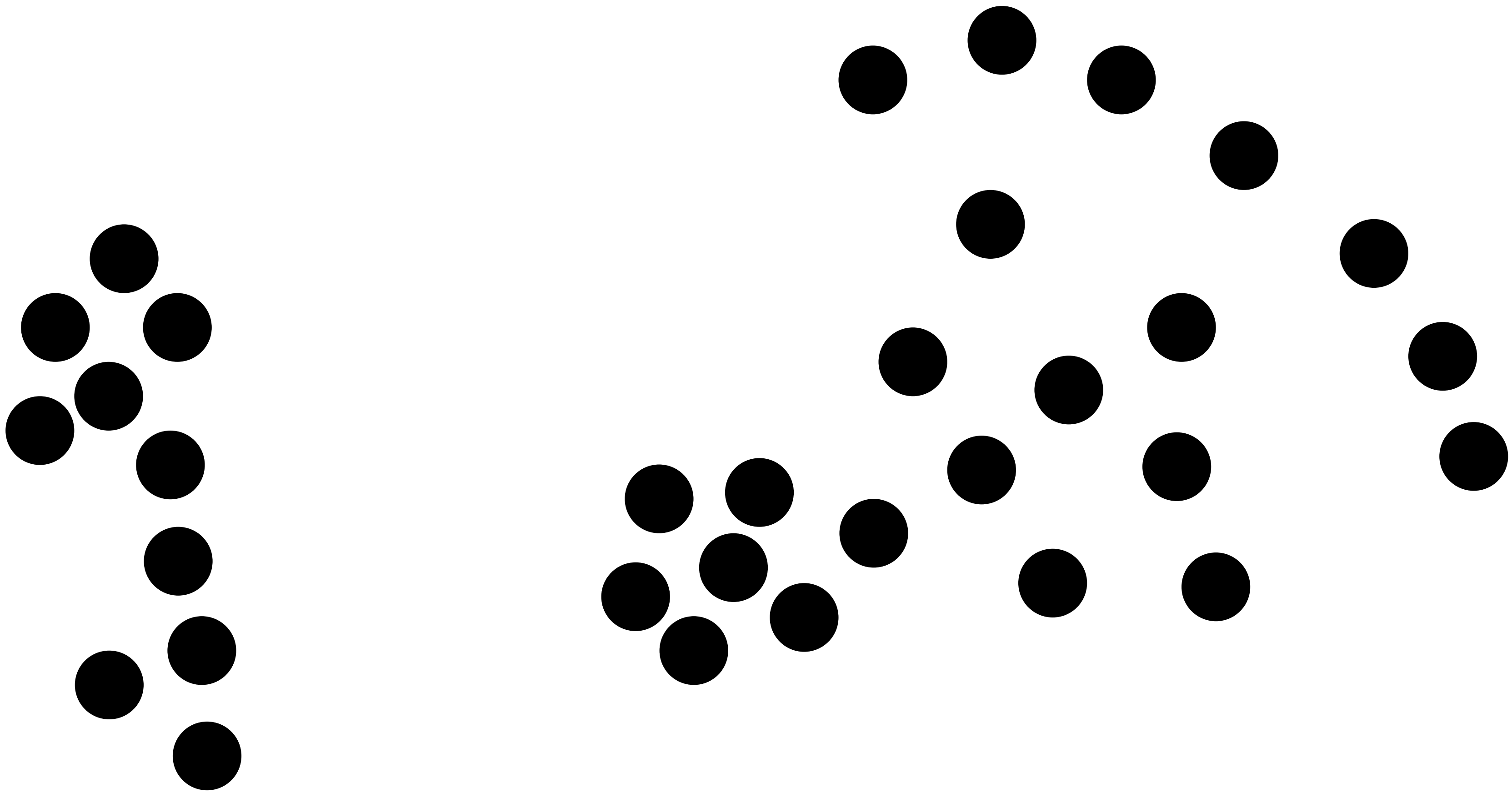
Clustering procedure



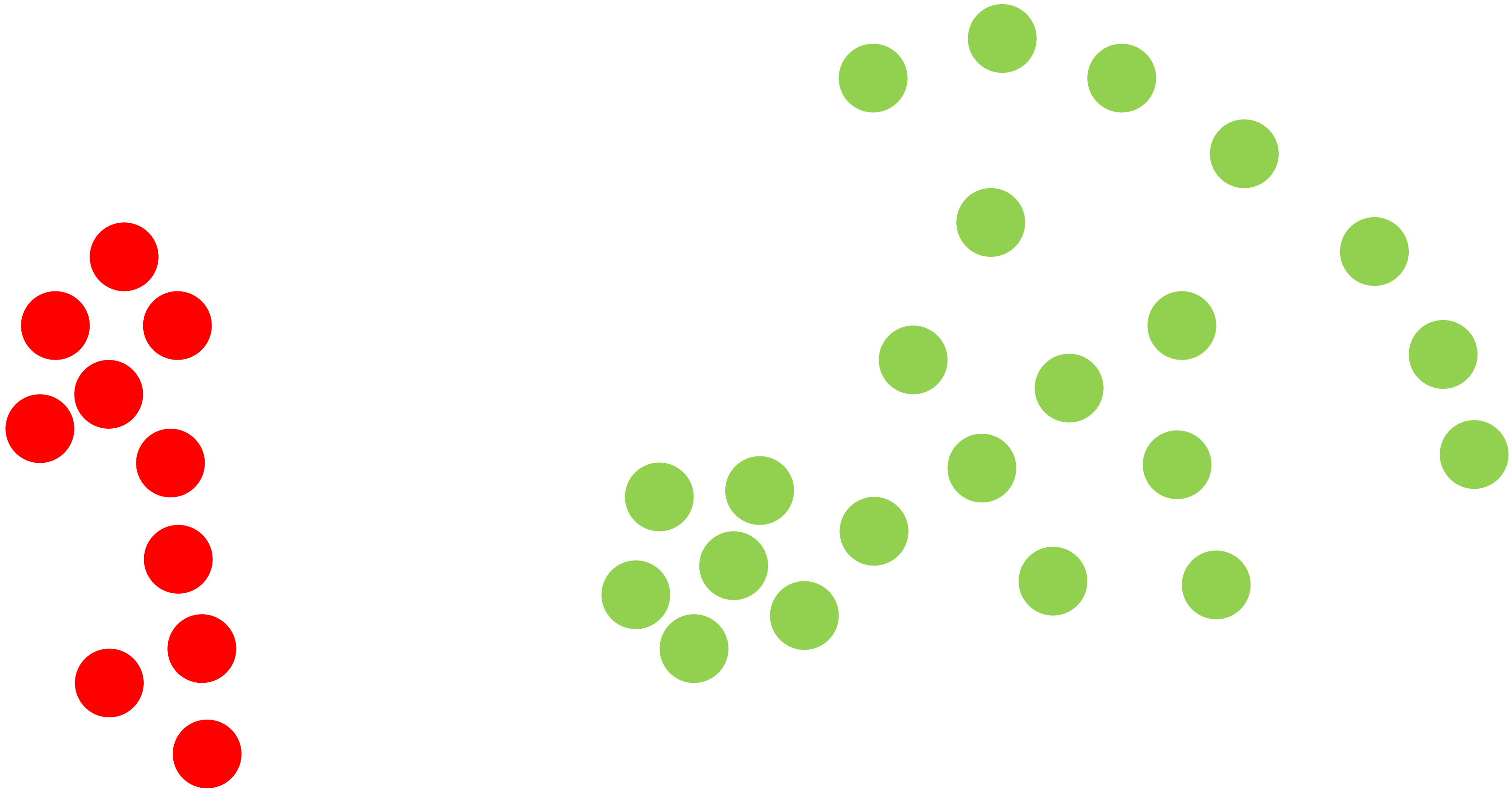
What is a cluster (revisited)



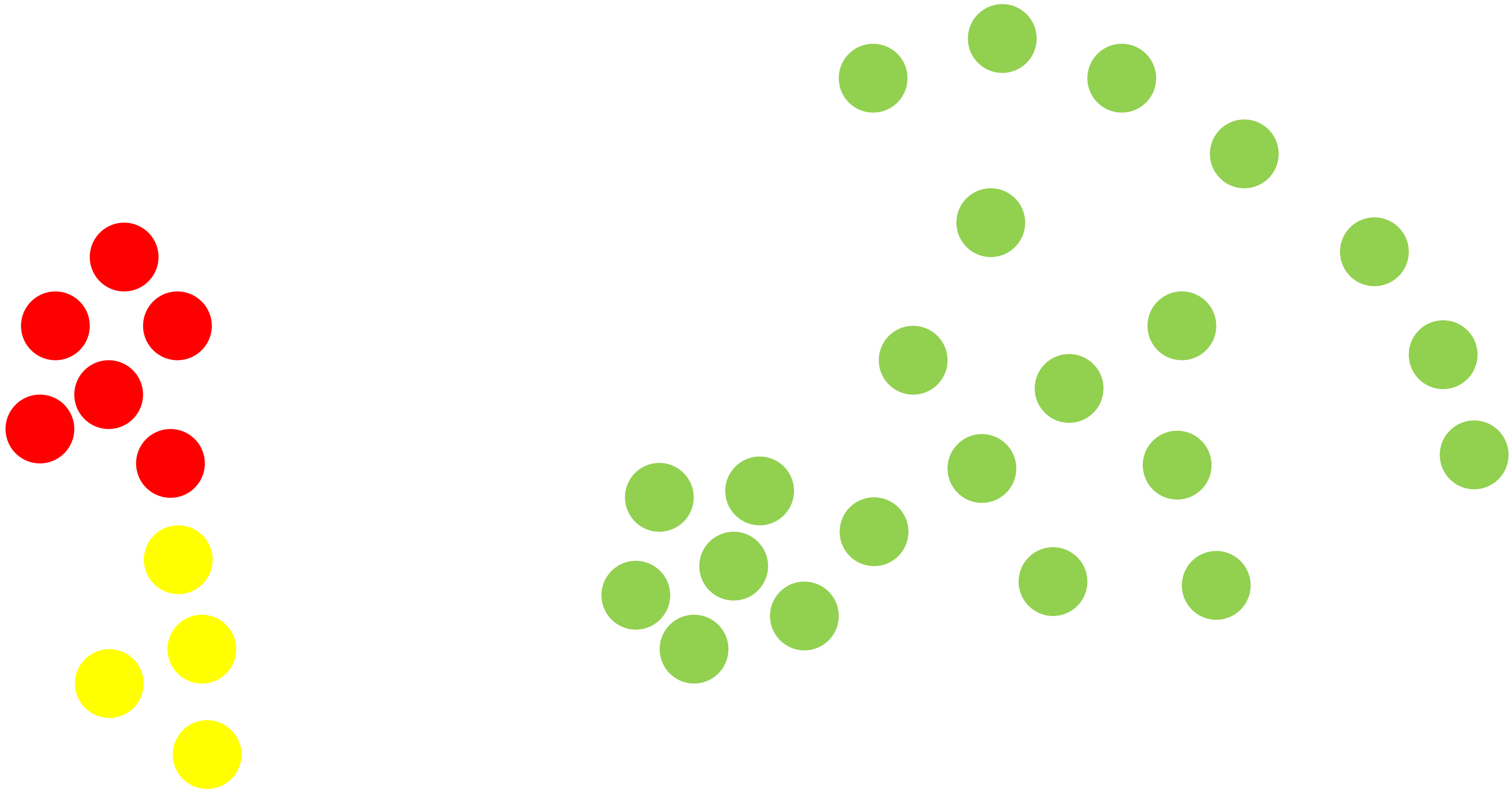
What is a cluster (revisited)



What is a cluster (revisited)



What is a cluster (revisited)



What is a cluster (revisited)

What is a cluster (revisited)

What is a cluster (revisited)

Some consideration about Clustering

- Tautology: the result of the clustering process depends on your cluster definition
- And it depends on the metric
- And it also depends on the features that you have chosen

Flat, fuzzy and hierarchical clustering

- Flat clustering performs a hard partition of the data
- Fuzzy clustering is a flat clustering algorithm with soft element assignation
- Hierarchical clustering generates a tree instead of a single partition. Can be agglomerative (joining elements) or divisive (dividing the dataset)

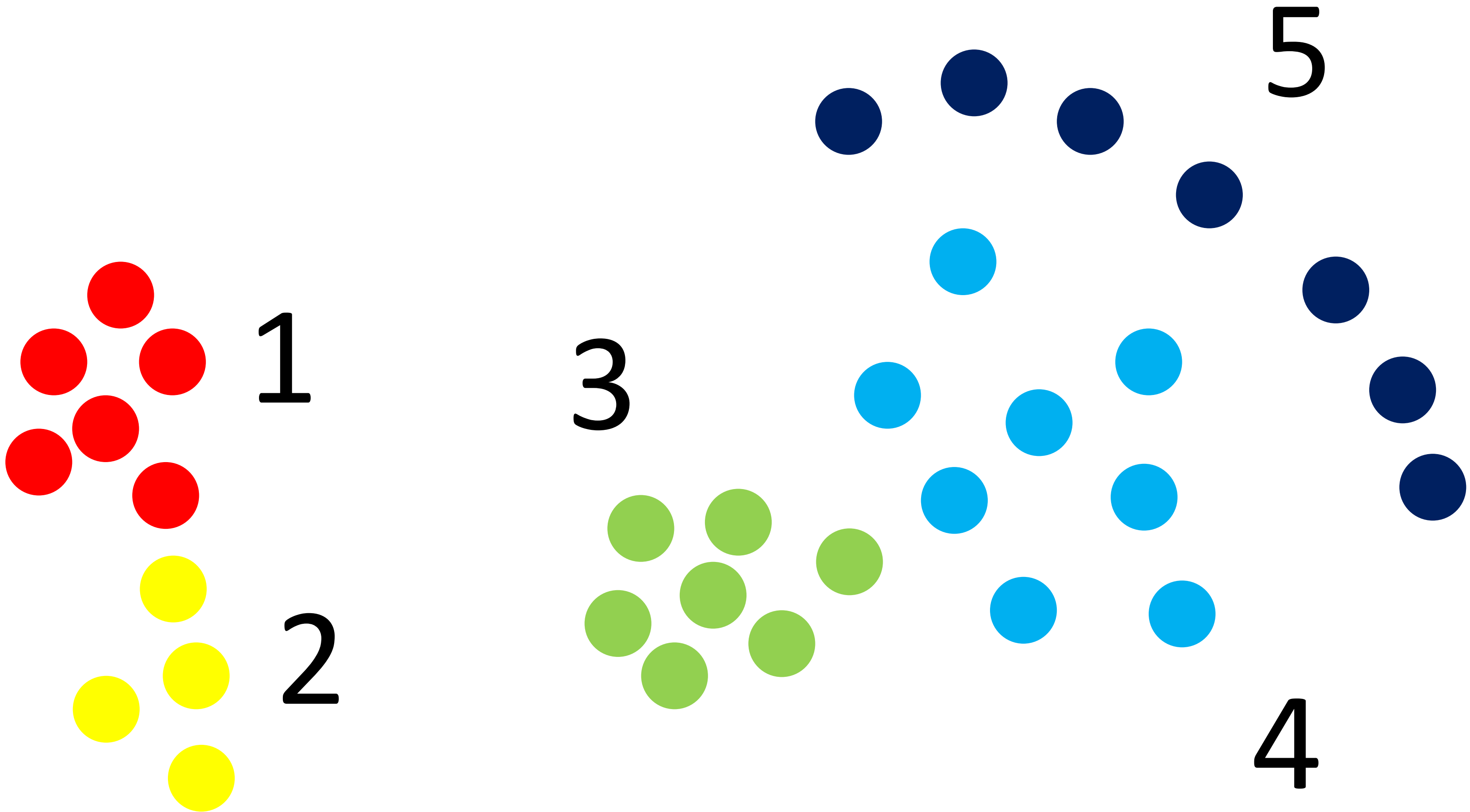
Flat clustering

- Each element is assigned to a single cluster.

$$Cl(i) = l$$

- Traditionally, the number of clusters (k) should be given to the algorithm as an external parameter. Nowadays, many clustering algorithms estimate internally this parameter.
- usually when one performs clustering one looks for a hard partition.
- Can not deal well with multilevel structures

Flat clustering



Fuzzy clustering

- Each element is assigned to *all* the clusters in the dataset with a given degree of membership.

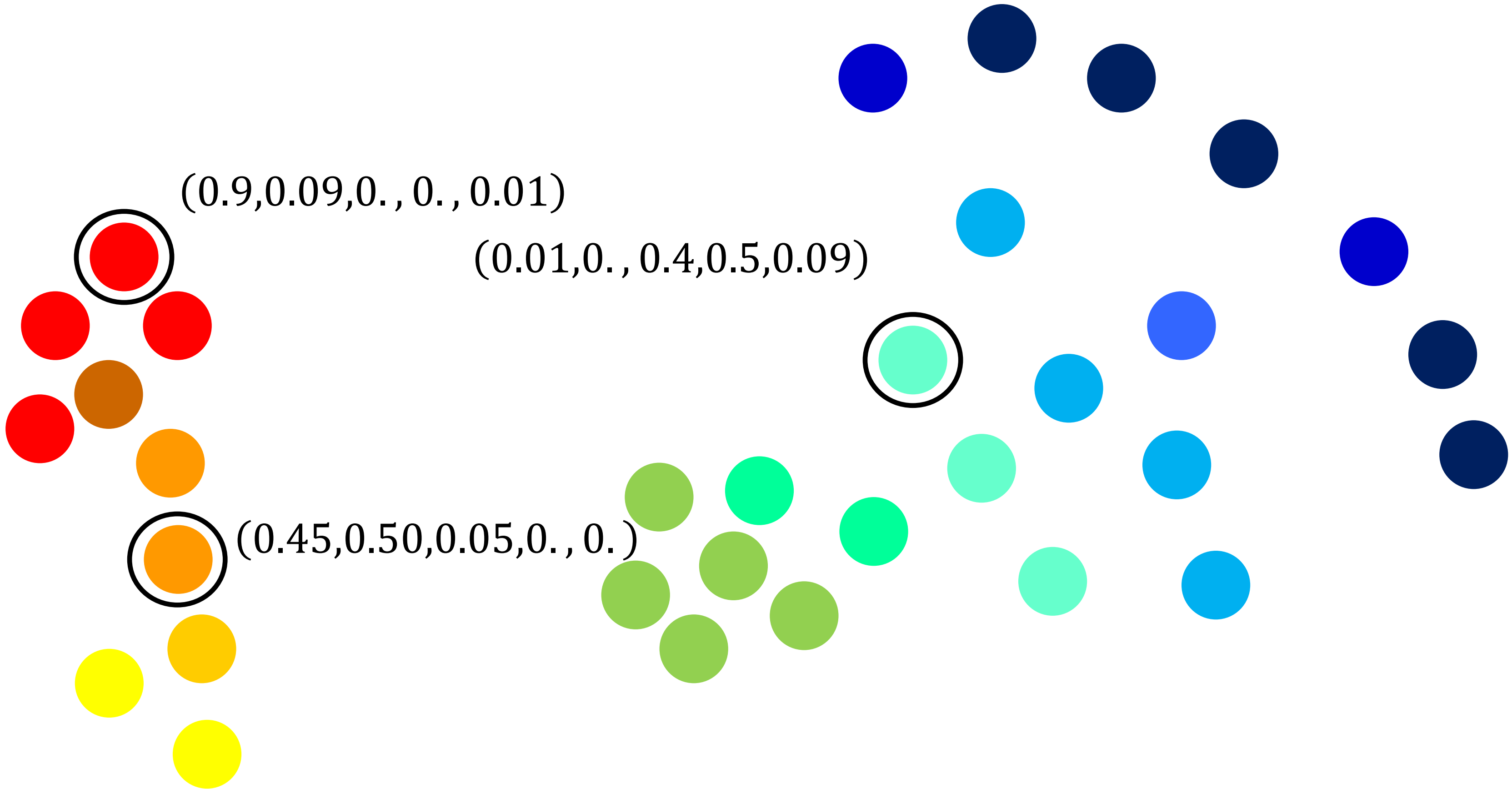
$$\overrightarrow{Cl}(i) = (u_1, u_2, \ldots u_l \ldots, u_k)$$

- The membership vector is normalized

$$\sum_{l=1}^{k} u_l = 1$$

- Again, the number of clusters should be provided as an external parameter.
- It may be difficult to transform in a hard partition

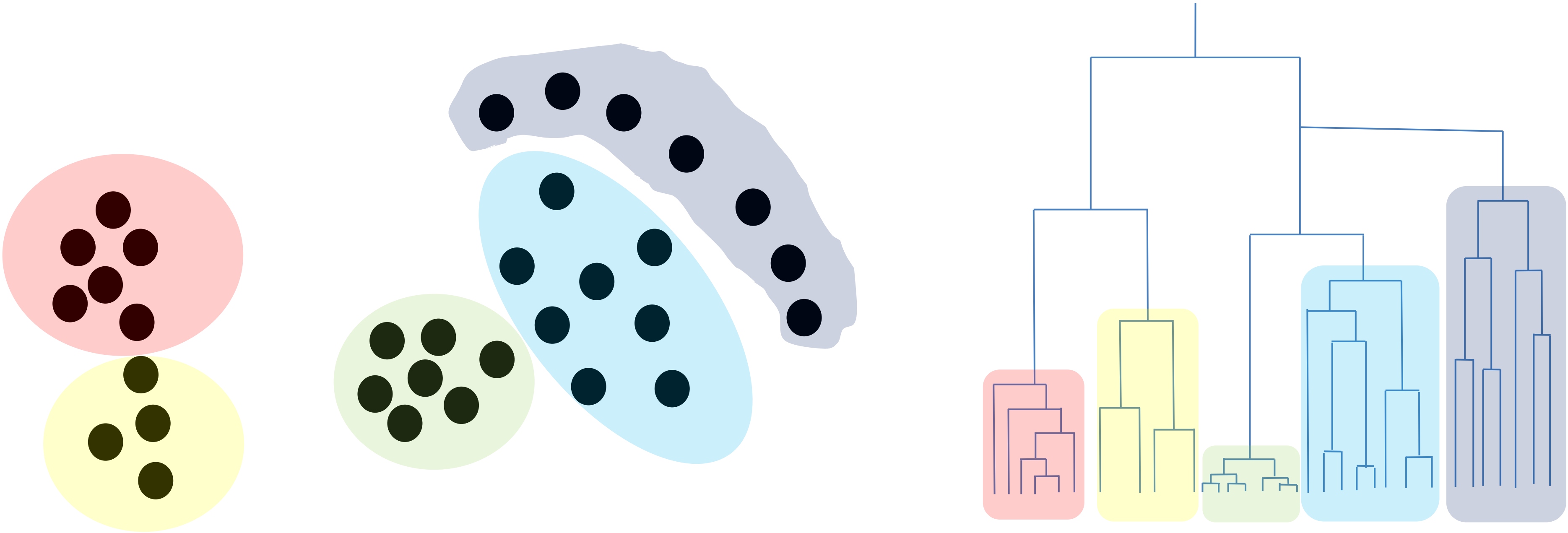
Fuzzy clustering



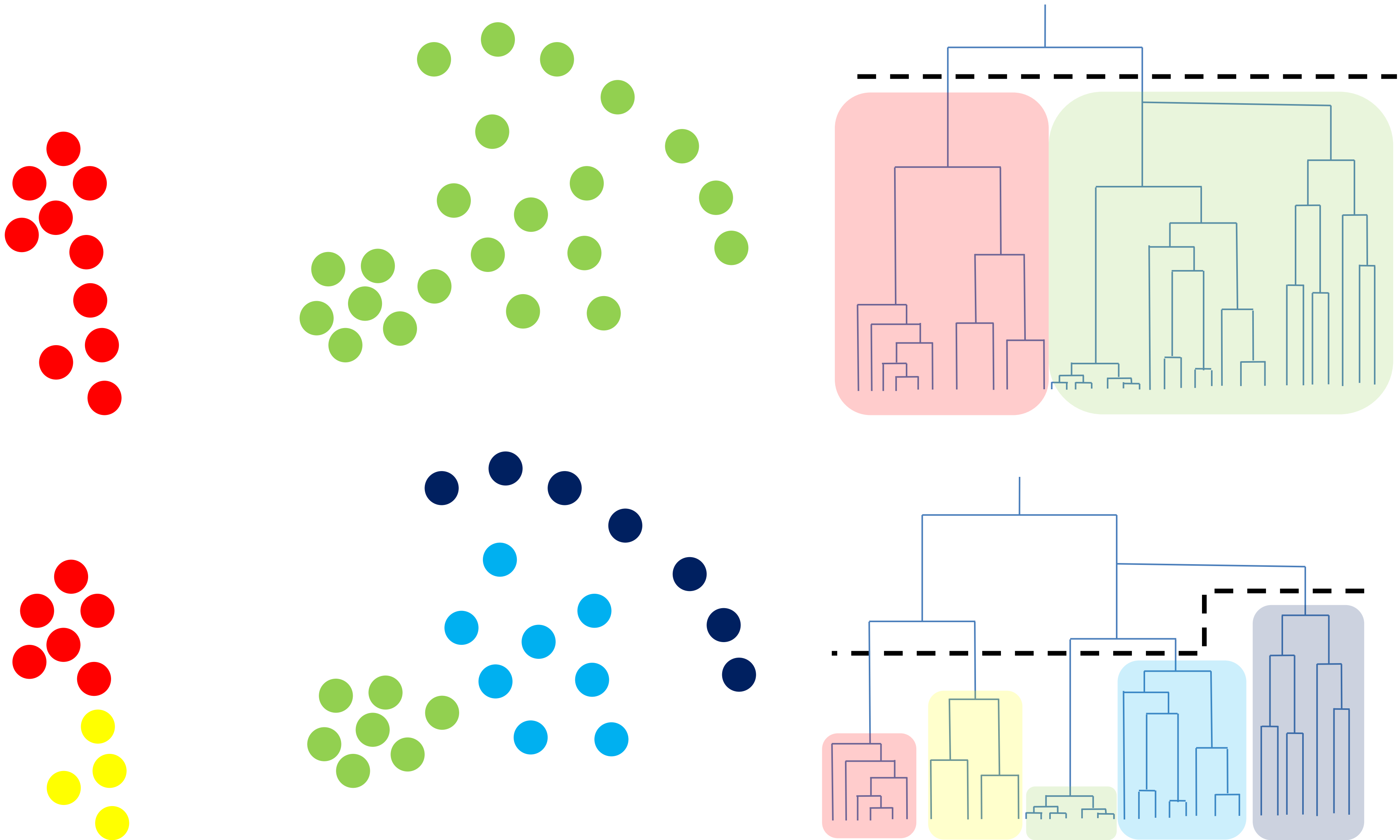
Hierarchical clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram (A tree-like diagram that records the sequences of merges or splits)
- No assumptions on the number of clusters
- Hierarchical clusterings may correspond to meaningful taxonomies
- It can be transformed in many hard partitions

Hierarchical clustering



Flattening the hierarchical clustering



K-means: A flat clustering algorithm

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

K-means clustering

- Attempts to minimize the intracluster distance while maximizing the intercluster distance.
- It is based on the concept of cluster centroid, i.e. the average position of the cluster elements.
- Still widely used.
- It can be easily parallelized and linearized.
- User must provide k

K-means objective function

- Objective function:

$$O(z) = \sum_{l=1}^{k} \sum_{i=1}^{n} \delta_{z_i l} \|\overrightarrow{x_i} - \overrightarrow{c_l}\|^2$$

- z is an array with n components that reflects the assignation in clusters.
- $\overrightarrow{c_l}$ is the vector with the coordinates of the l cluster center. $\overrightarrow{c_l} = \frac{\sum_{i=1}^{n} \delta_{z_i l} \overrightarrow{x_i}}{\sum_{i=1}^{n} \delta_{z_i l}}$

k-means algorithm

Input: cluster size k, instances $\{\vec{x_i}\}_{i=1}^{n}$, distance metric $d(\cdot,\cdot)$
Output: cluster membership assignments $\{z_i\}_{i=1}^{n}$

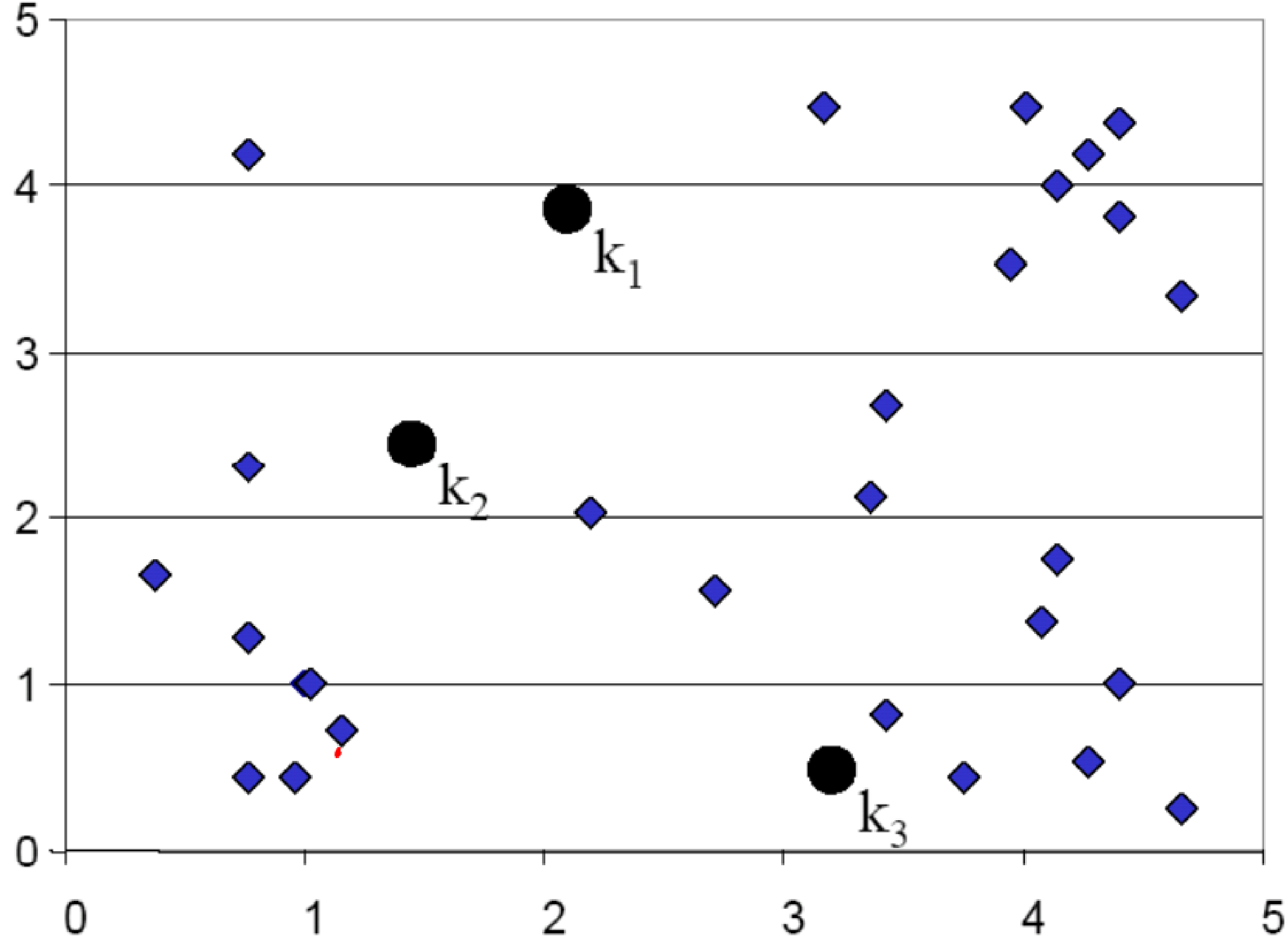
1. Initialize k cluster centroids $\{\vec{c_l}\}_{l=1}^{k}$ (randomly if no domain knowledge available)
2. Repeat until no instance changes its cluster membership:
   - Decide the cluster membership of instances by assigning them to the nearest cluster centroid

     $$z_i = argmin_l d(\vec{c_l}, \vec{x_i})$$ *Minimize intra distance*

   - Update the k cluster centroids based on the assigned cluster membership

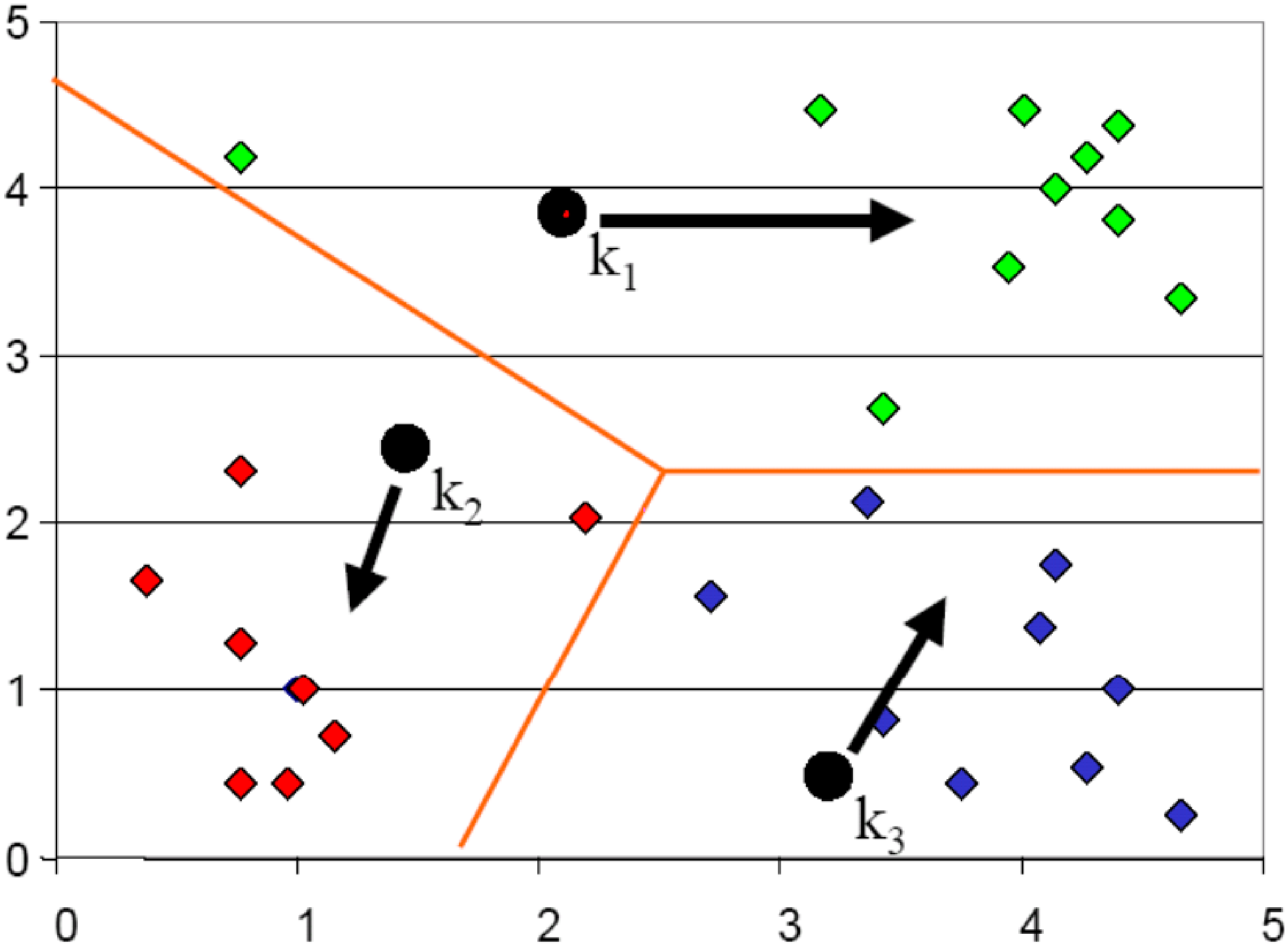
     $$\vec{c_l} = \frac{\sum_{i=1}^{n} \delta_{z_i l} \vec{x_i}}{\sum_{i=1}^{n} \delta_{z_i l}}$$ *Maximize inter distance*

k-means illustration

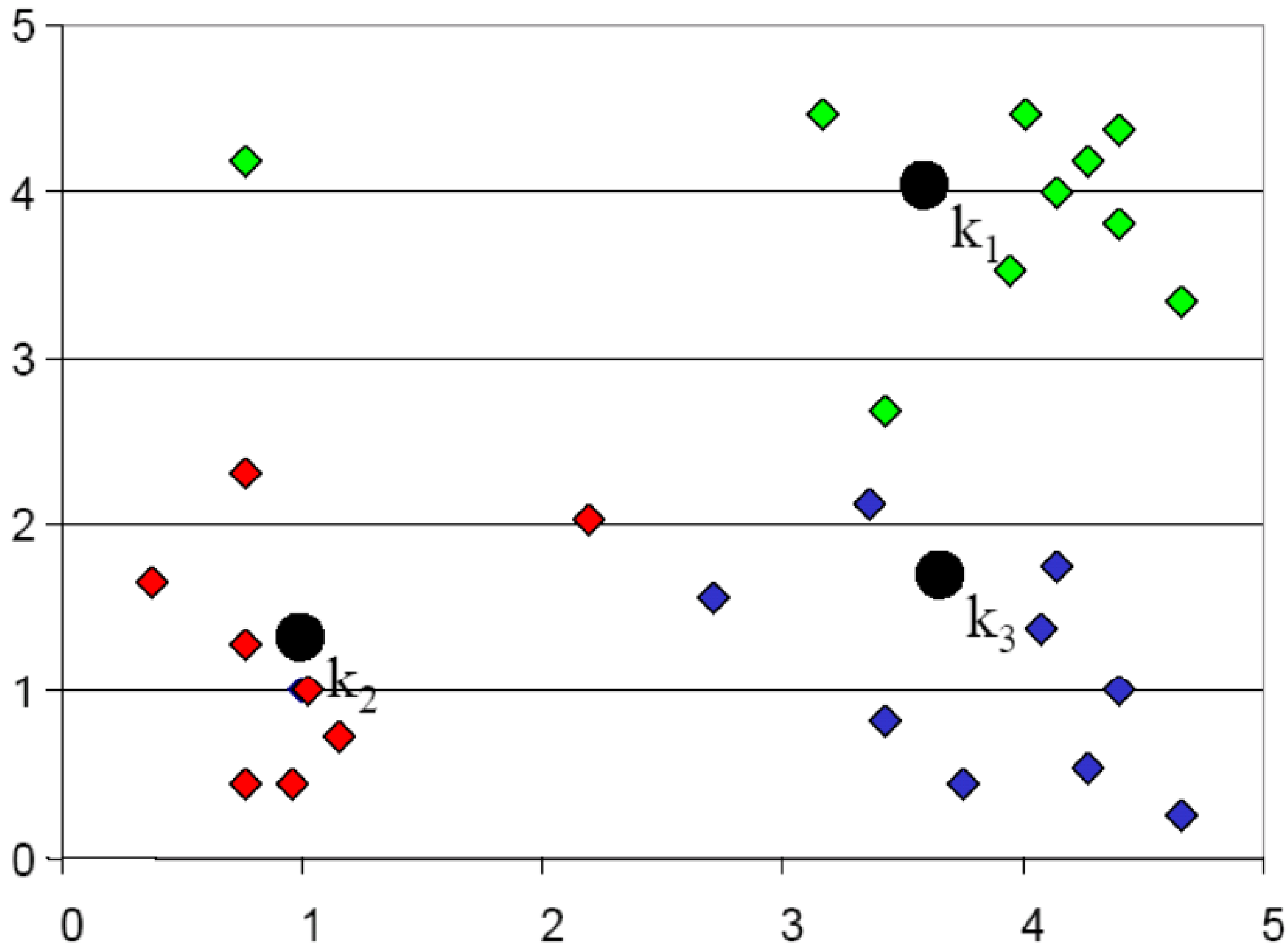


k-means illustration

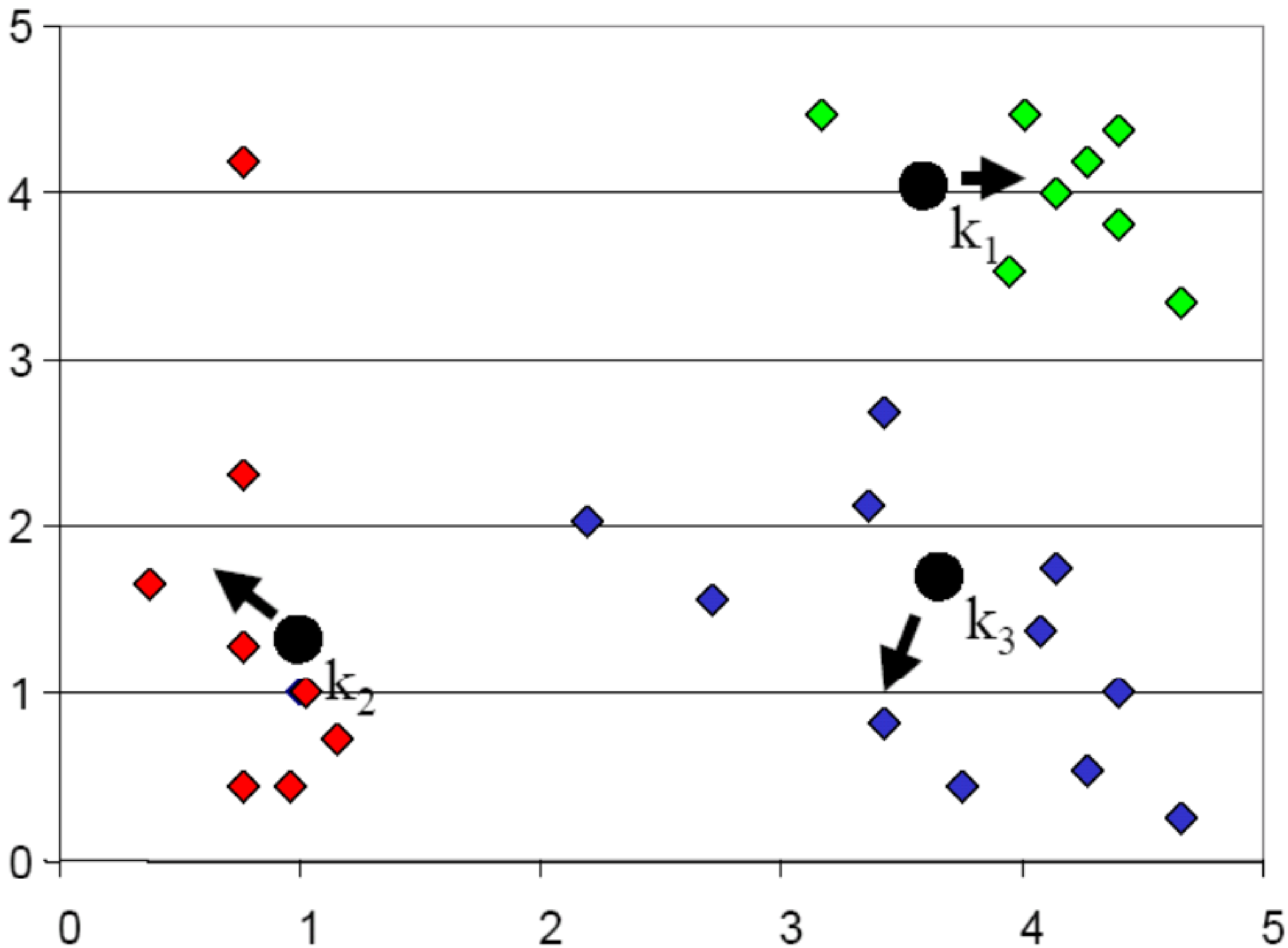
Voronoi diagram



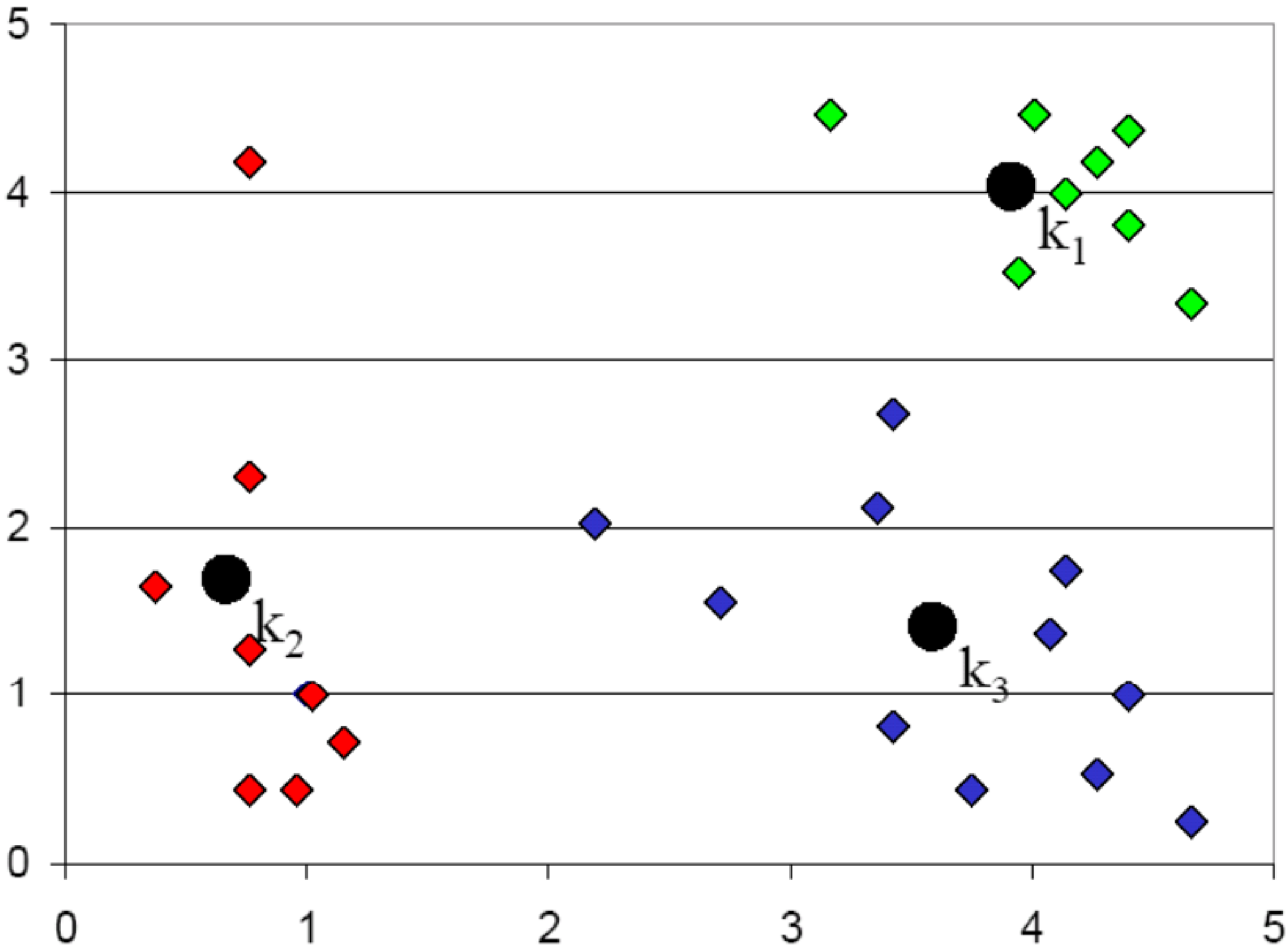
k-means illustration



k-means illustration



k-means illustration



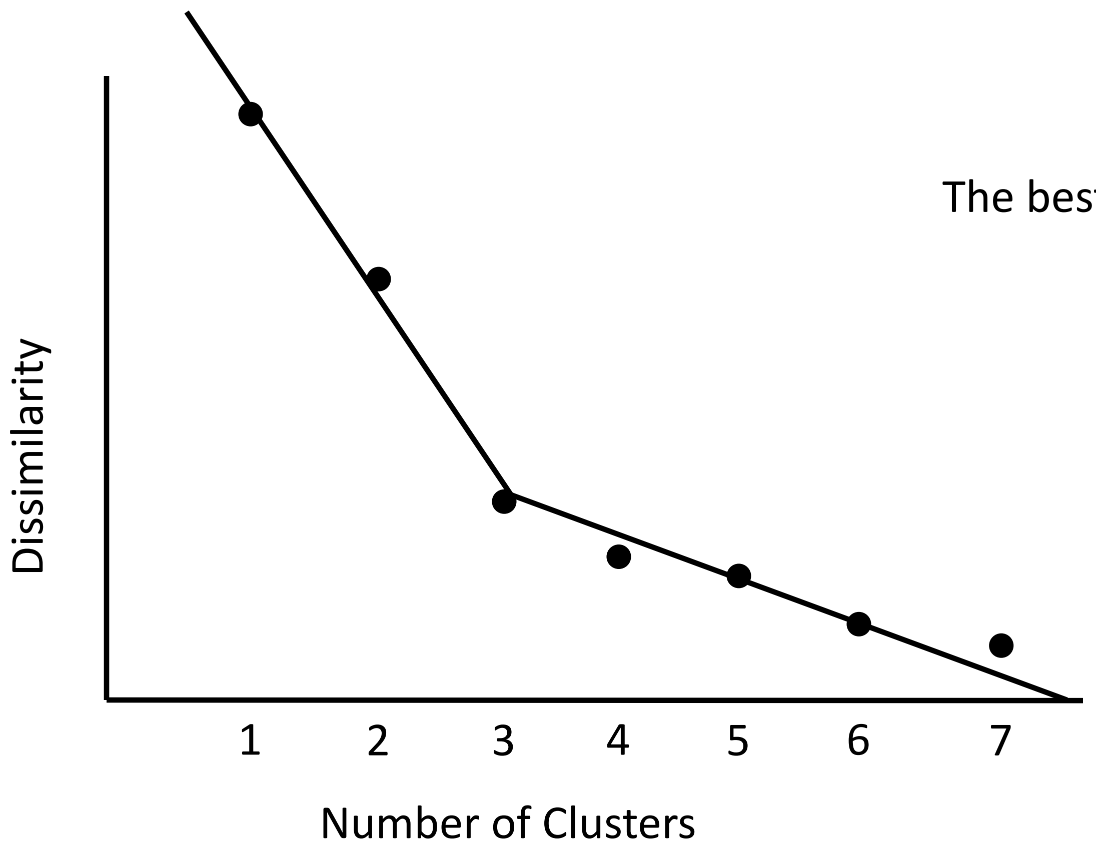
K-means weakness

- Initialization sensitive (local optimization) →k-means++
- Which k-employing →Scree test
- Sensitive to outliers →k-medoids
- employs Euclidean distance →k-medoids
- Only for spherical clusters →kernel k-means (advanced)

Better initialization: *k*-means++

1. Choose the first cluster center at random
2. Repeat until all *k* centers have been found
   - For each instance compute $\mathrm{D_x} = \min_{\mathrm{k}} d(x, c_k)$
   - Choose a new cluster center with probability $p(x) \propto D_x^2$ ← *new center should be far away from existing centers*
3. Run *k*-means with selected centers as initialization

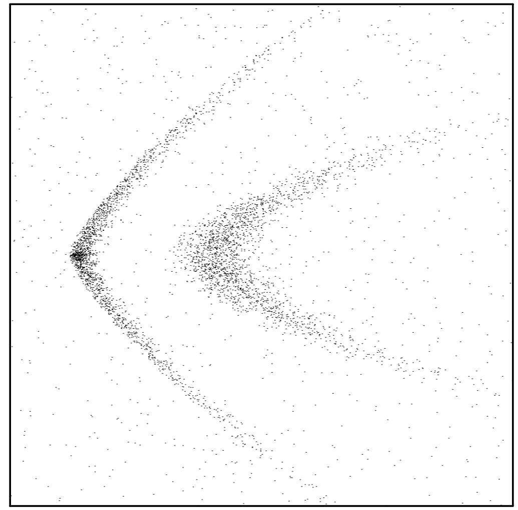
Scree test for k-means



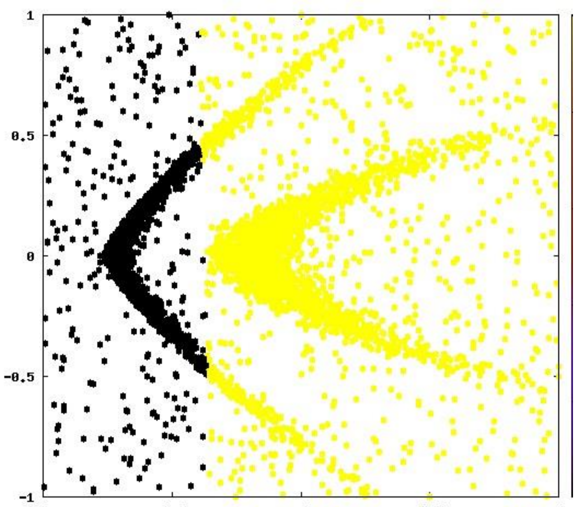
The best K is 3.

K-medoids

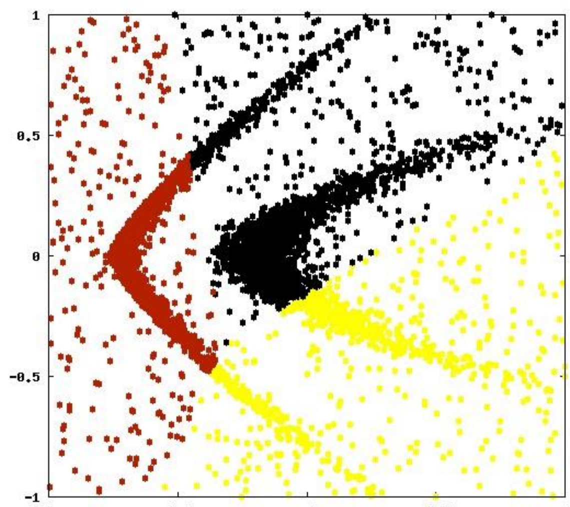
- It can be used with all kind of distances.
- Reduces the impact of outliers.
- Instead of working with centroids, it works with medoids, i.e. the most central element of the cluster.
- The cluster medoid is the element with minimum sum of distances to the rest of the elements of the cluster



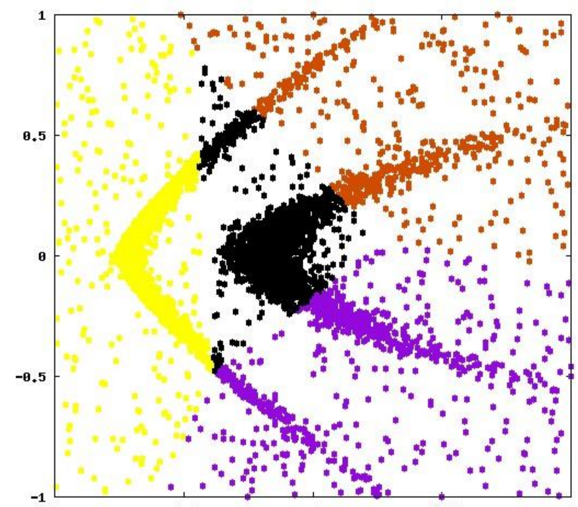
K=2



K=3



K=4



Fuzzy c-means: A fuzzy clustering algorithm

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2), 191-203.

Fuzzy c-means

- It can be considered a version of k-means with a soft assignation criteria.

$$\overrightarrow{Cl}(i) = (u_1, u_2, \dots u_l \dots, u_k)$$

$$\sum_{l=1}^{k} u_l = 1$$

- Therefore, the objective function and the optimization algorithm must be adapted to fulfill these conditions.

Fuzzy c-means algorithm

- Objective function:

$$O(\mathbb{U}) = \sum_{l=1}^{k} \sum_{i=1}^{n} (u_{il})^m \|\overrightarrow{x_i} - \overrightarrow{c_l}\|^2$$

- $\mathbb{U}$ is a $n \times k$ matrix with the membership of the n elements in the k clusters.
- m is the fuzzification parameter. Usually m=2.
- $\overrightarrow{c_l}$ is the vector with the coordinates of the l cluster center. As in the case of k-means, it is the average value of the cluster coordinates. $\overrightarrow{c_l} = \frac{\sum_{i=1}^{n} (u_{il})^m \overrightarrow{x_i}}{\sum_{i=1}^{n} (u_{il})^m}$

Fuzzy c-means algorithm

- Random initialization of $\mathbb{U}$
- Iterative optimization:
  - Given $\mathbb{U}$ compute the centers:

$$\vec{c_l} = \frac{\sum_{i=1}^{n} (u_{il})^m \vec{x_i}}{\sum_{i=1}^{n} (u_{il})^m}$$

  - Given the centers, update $\mathbb{U}$:

$$u_{ij} = \frac{1}{\sum_{l=1}^{k} \left(\frac{\left\| \vec{x_i} - \vec{c_j} \right\|}{\left\| \vec{x_i} - \vec{c_l} \right\|} \right)^{2/_{m-1}}}$$

- Repeat until the change in $\mathbb{U}$ is smaller than a given threshold

Fuzzy c-means algorithm

- Similar problems to k-means:
  - Initialization
  - Number of clusters
  - Sensitive to outliers
  - Spherical clusters

Hierarchical clustering algorithms

Based on slides by Evimaria Terzi
(http://www.cs.bu.edu/~evimaria/)

Hierarchical Clustering

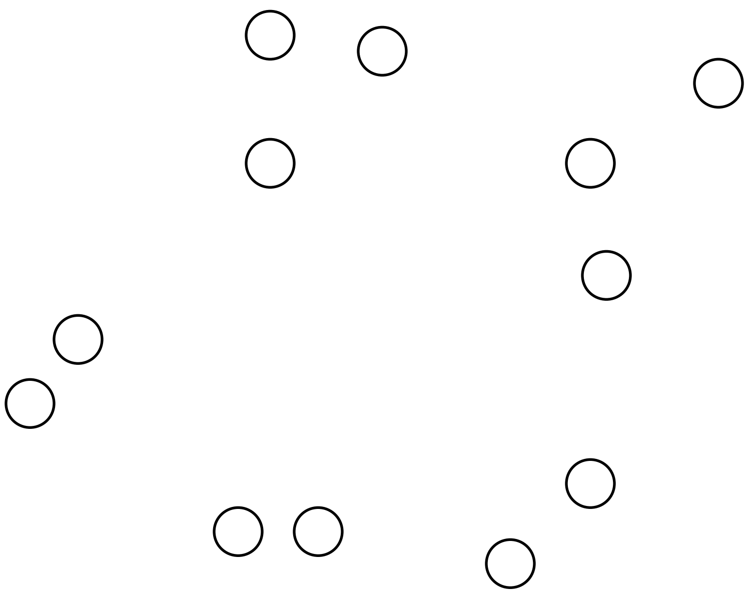
- Two main types of hierarchical clustering
  - **Agglomerative:**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or **k** clusters) are left
  - **Divisive:**
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are **k** clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative clustering algorithm

- Most popular hierarchical clustering technique
- Basic algorithm
  1. Compute the distance matrix between the input data points
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the distance matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the distance between two clusters
  - Different definitions of the distance between clusters lead to different algorithms

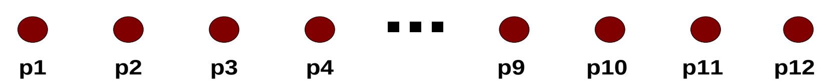
Input/ Initial setting

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



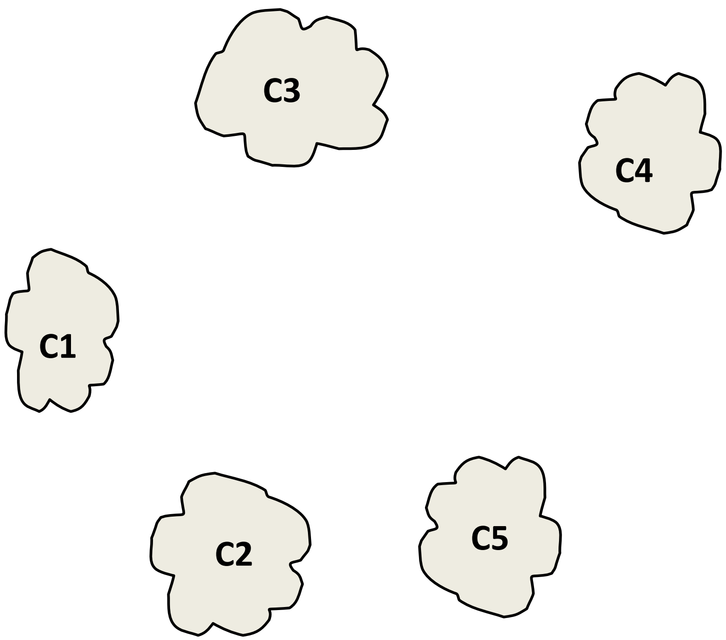
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance/Proximity Matrix



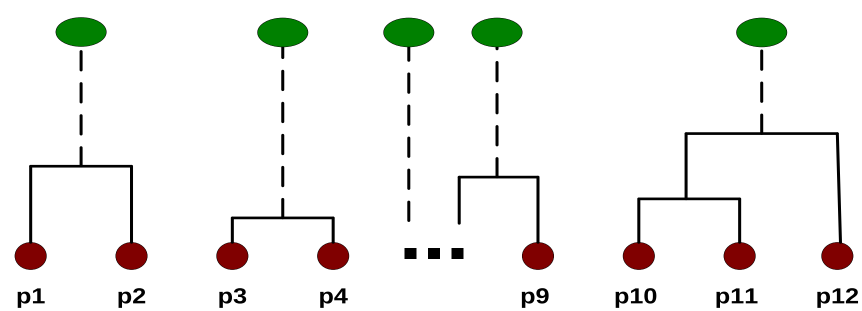
Intermediate State

- After some merging steps, we have some clusters



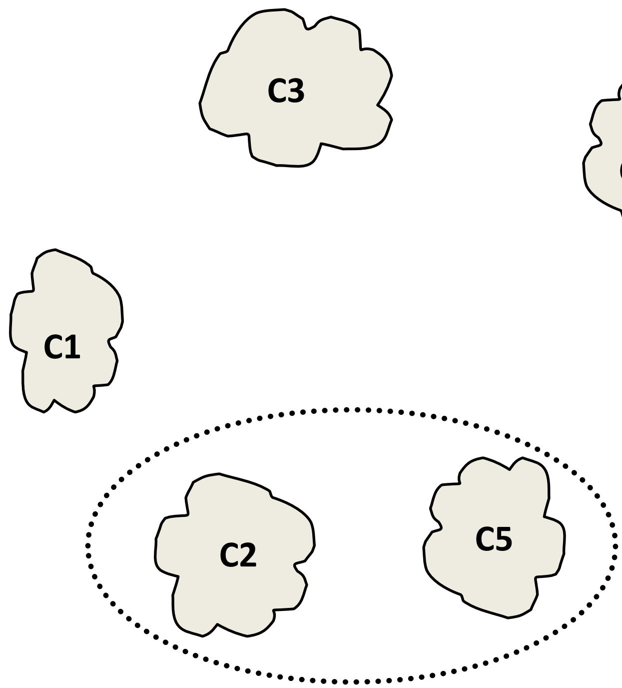
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

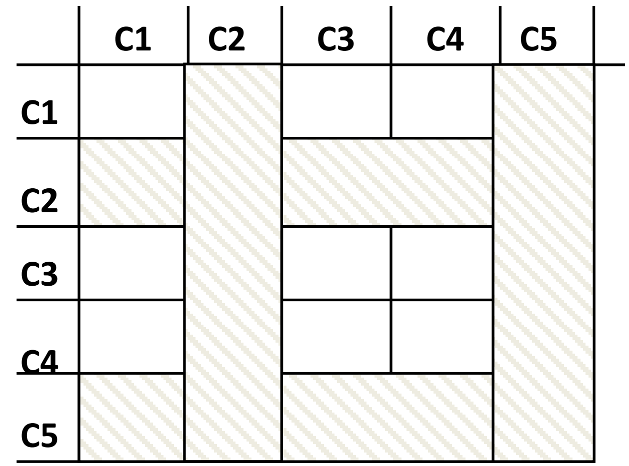
Distance/Proximity Matrix



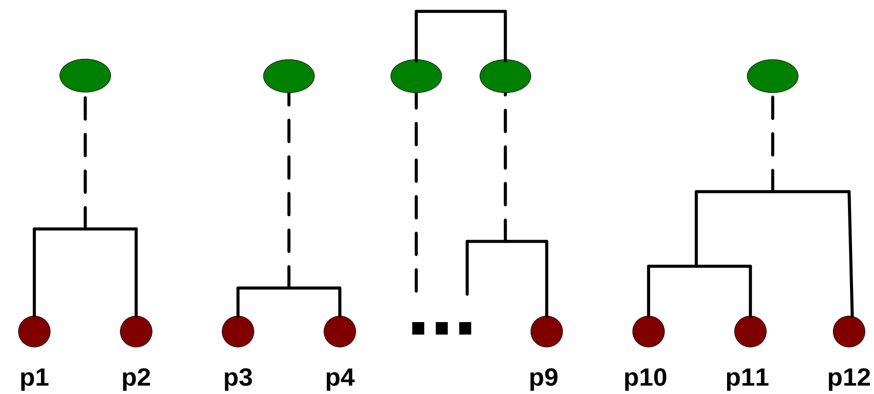
Intermediate State

- Merge the two closest clusters (C2 and C5) and update the distance matrix.



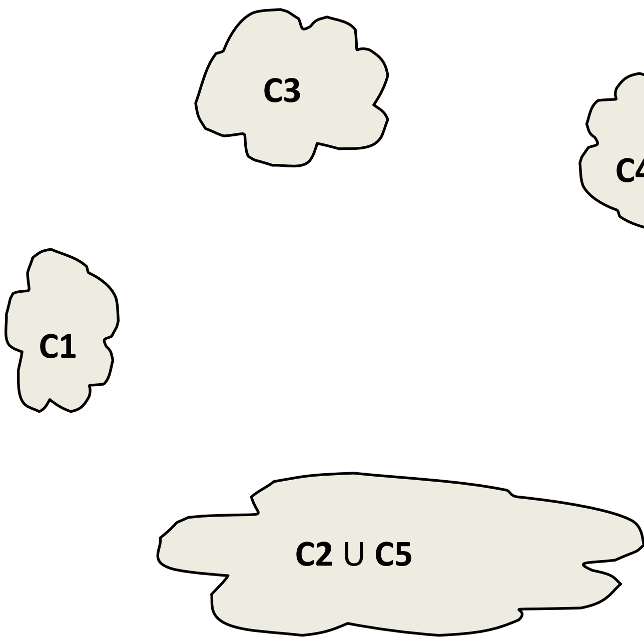


Distance/Proximity Matrix

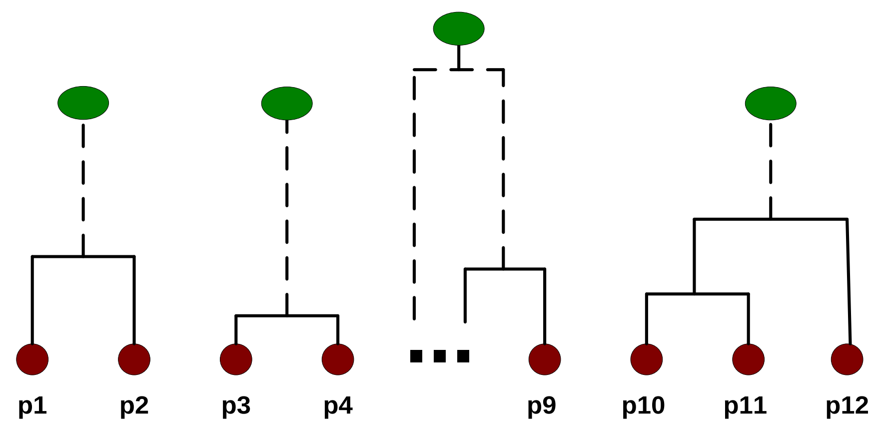


After Merging

- "How do we update the distance matrix?"



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		



Distance between two clusters

- Each cluster is a set of points

- How do we define distance between two sets of points
  - Lots of alternatives
  - Not an easy task

Distance between two clusters

- **Single-link distance** between clusters $\mathbf{C_i}$ and $\mathbf{C_j}$ is the ***minimum distance*** between any object in $\mathbf{C_i}$ and any object in $\mathbf{C_j}$

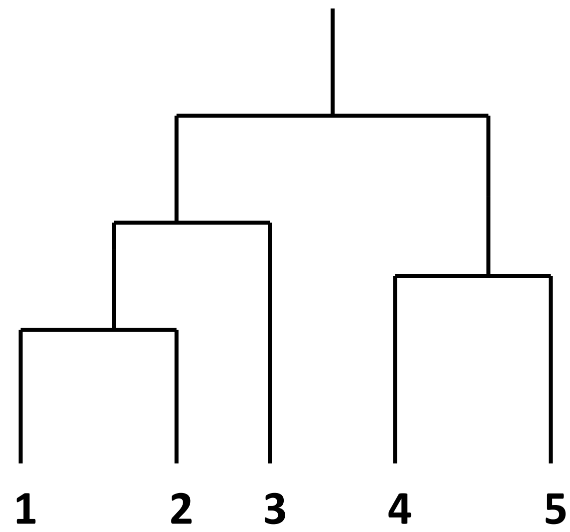
- The distance is **defined by the two most similar (i.e nearest) objects**

$$D_{sl}\left(C_i, C_j\right) = \min_{x,y} \left\{ d(x, y) \middle| x \in C_i, y \in C_j \right\}$$

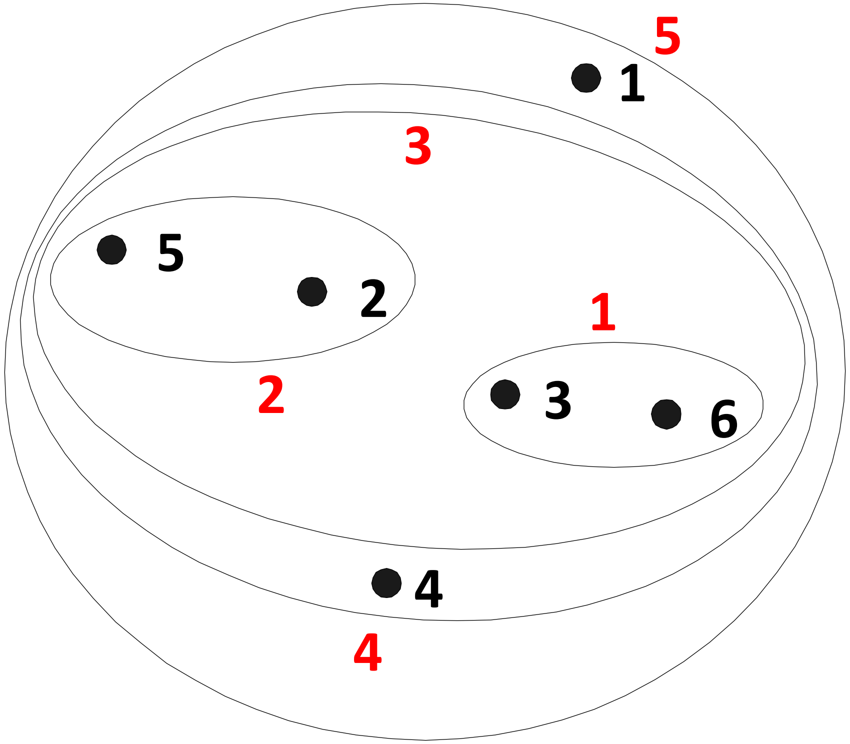
Single-link clustering: example

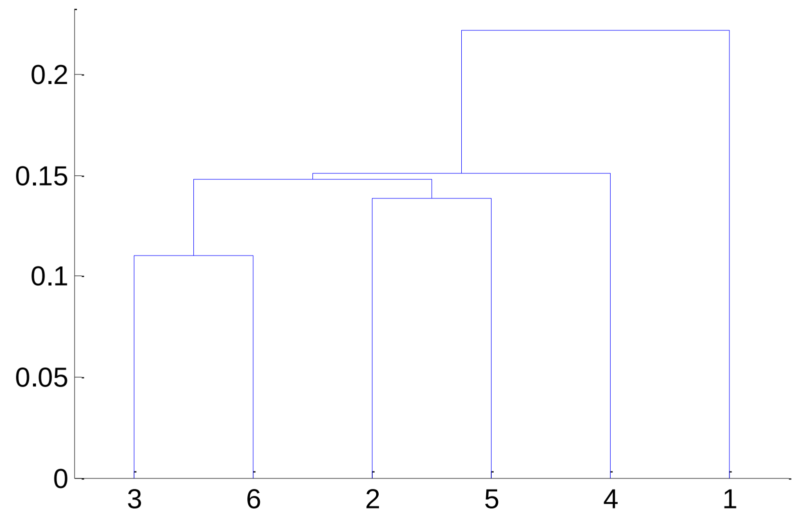
- Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Single-link clustering: example

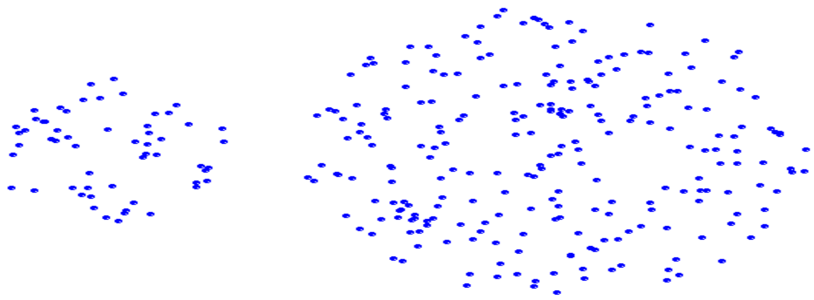




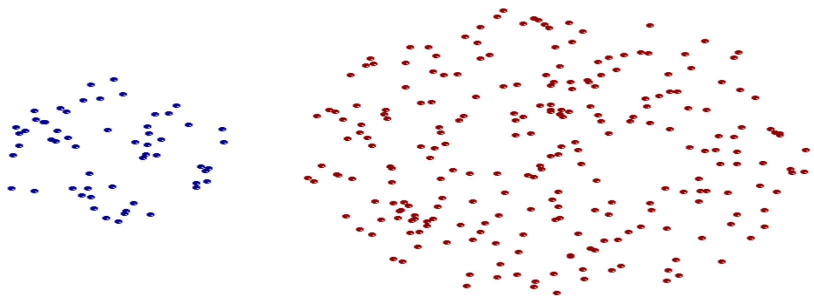
Nested Clusters

Dendrogram

Strengths of single-link clustering



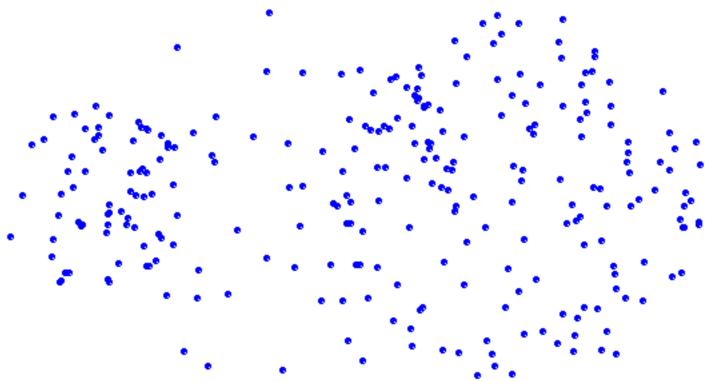
Original Points



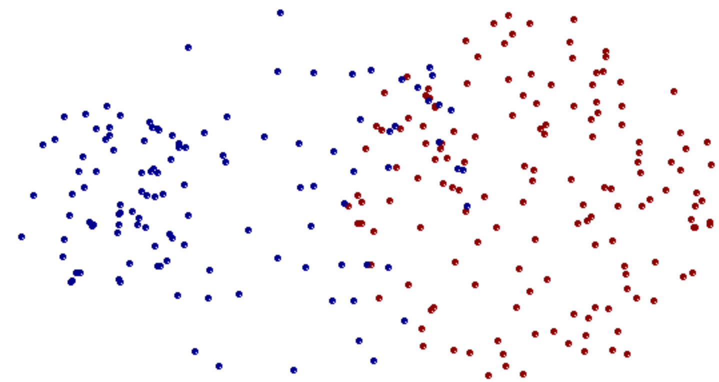
Two Clusters

- **Can handle non-elliptical shapes**

Limitations of single-link clustering



Original Points



Two Clusters

- Sensitive to noise and outliers
- It produces long, elongated clusters

Distance between two clusters

- **Complete-link distance** between clusters C_i and C_j is the ***maximum distance*** between any object in C_i and any object in C_j

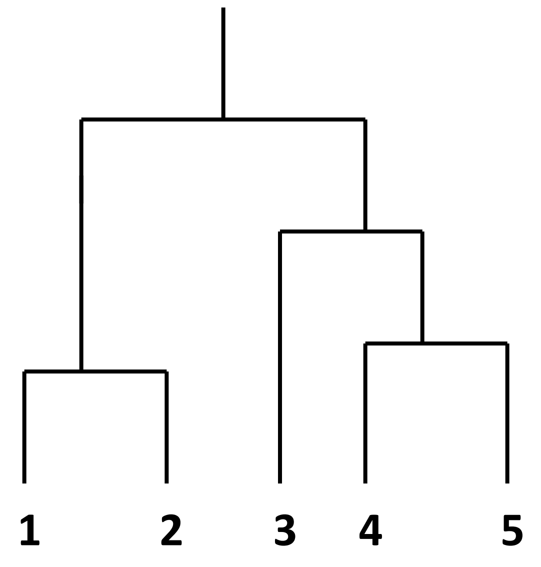
- The distance is **defined by the two most dissimilar (i.e. furthest) objects**

$$D_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

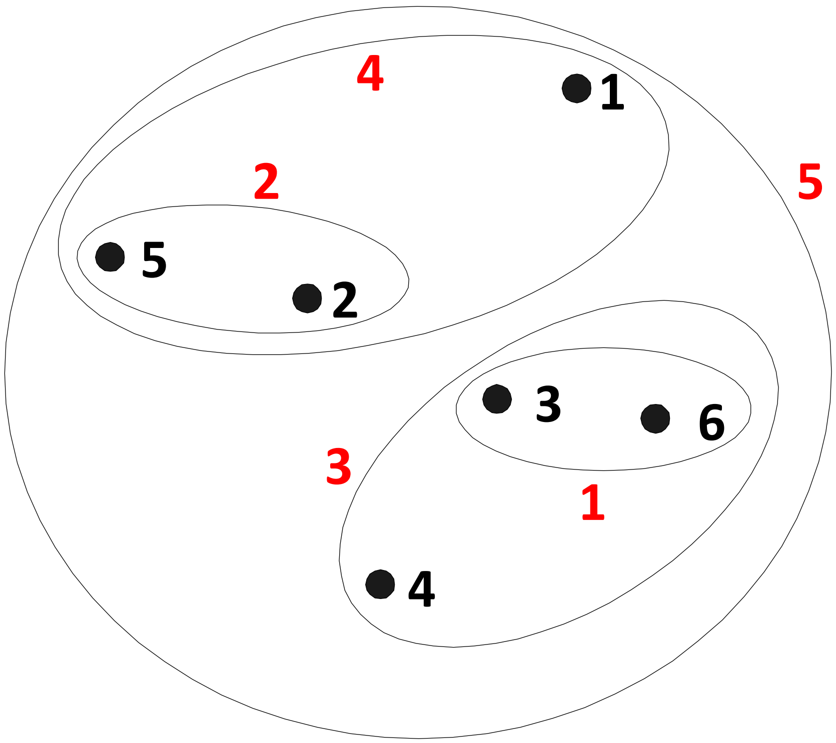
Complete-link clustering: example

- Distance between clusters is determined by the two most distant points in the different clusters

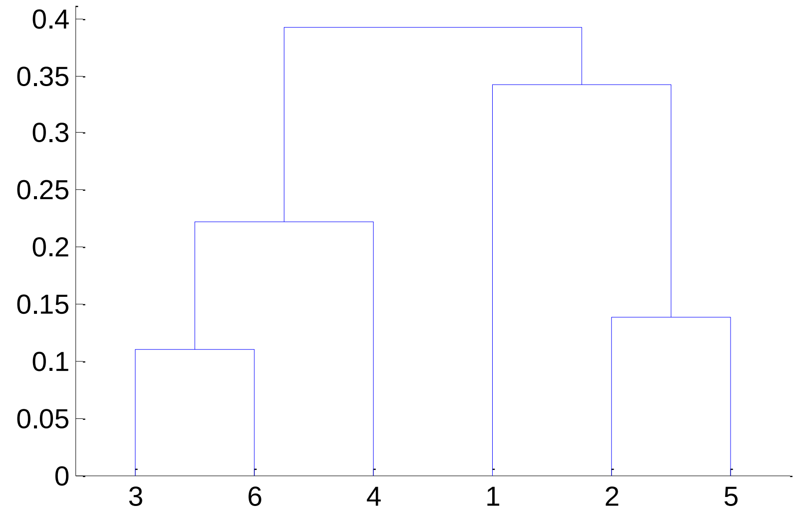
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Complete-link clustering: example

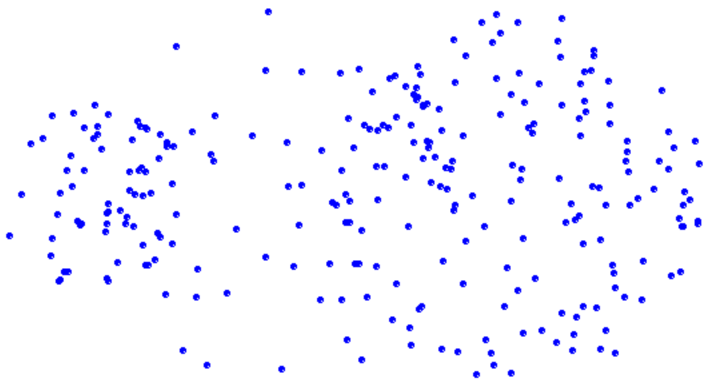


Nested Clusters

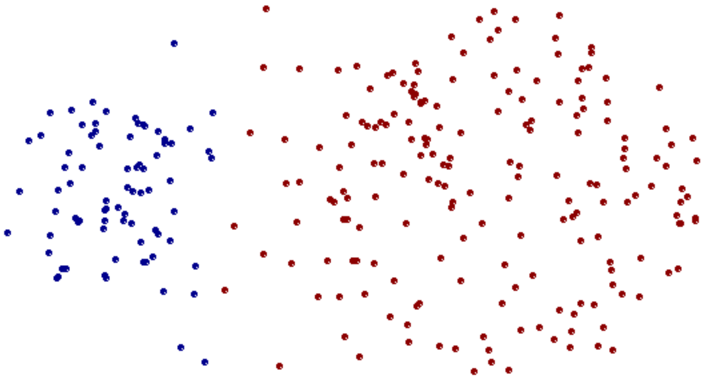


Dendrogram

Strengths of complete-link clustering



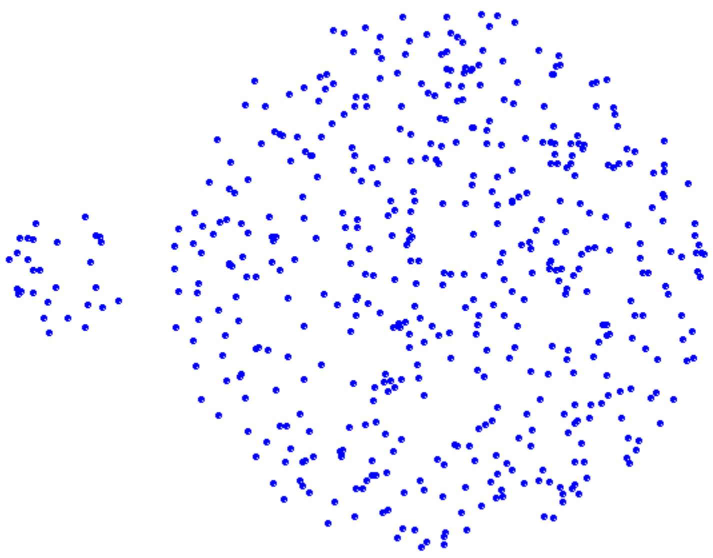
Original Points



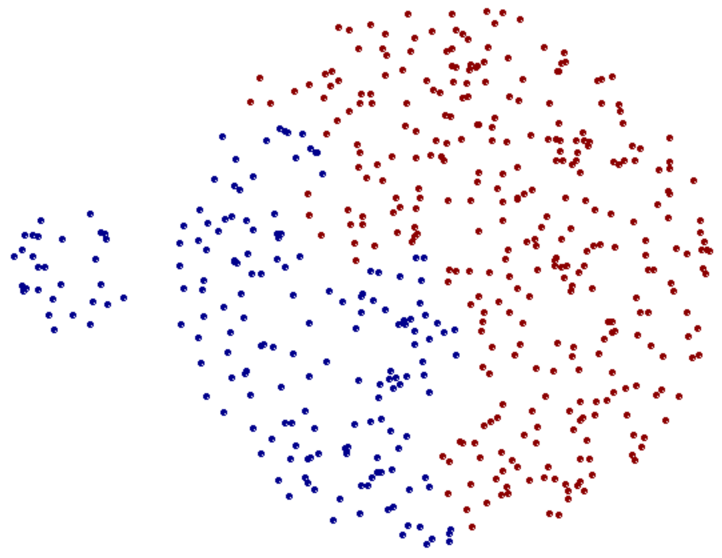
Two Clusters

- **More balanced clusters (with equal diameter)**
- **Less susceptible 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**

Distance between two clusters

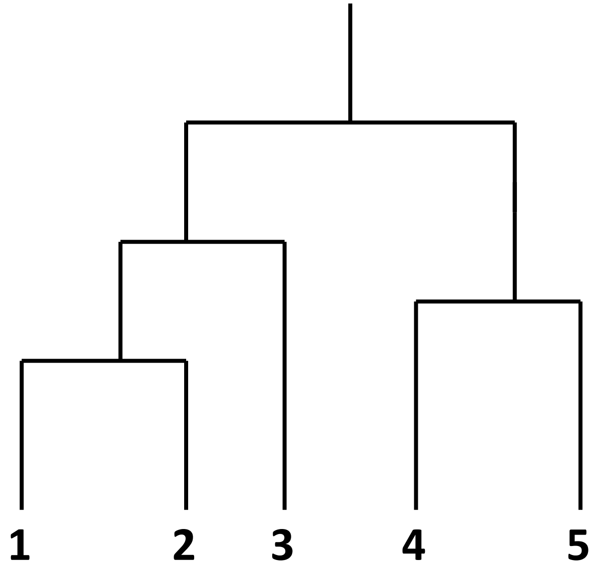
- **Group average distance** between clusters C_i and C_j is the ***average distance*** between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

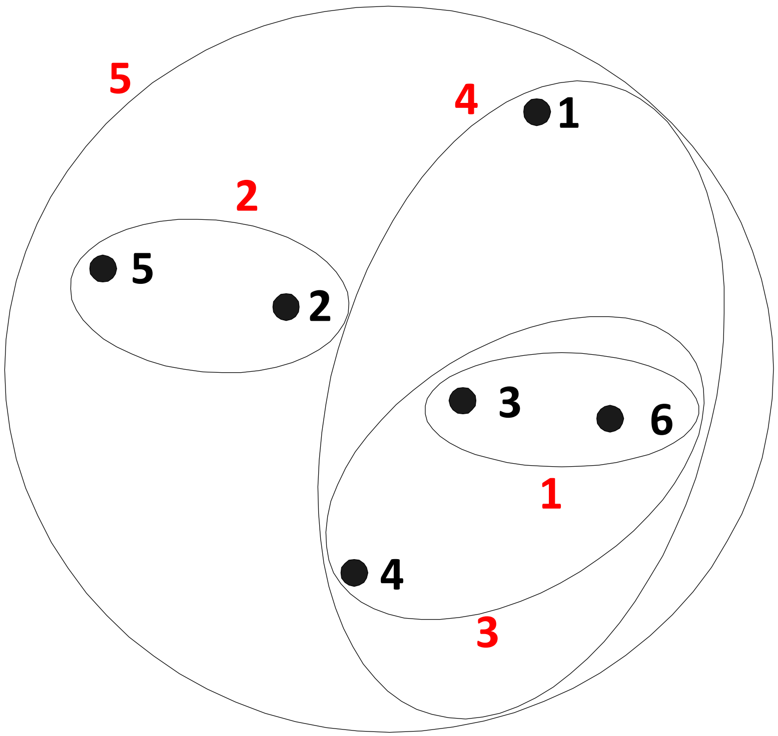
Average-link clustering: example

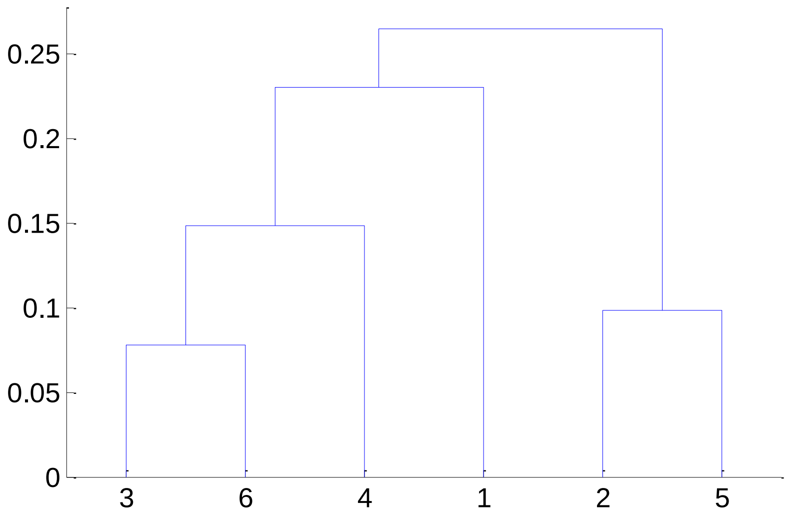
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



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
  - Biased towards globular clusters

Distance between two clusters

- **Centroid distance** between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}\left(C_i, C_j\right) = d(r_i, r_j)$$

Distance between two clusters

- **Ward's distance** between clusters C_i and C_j is the ***difference*** between the ***total within cluster sum of squares for the two clusters separately***, and the ***within cluster sum of squares resulting from merging the two clusters*** in cluster C_{ij}

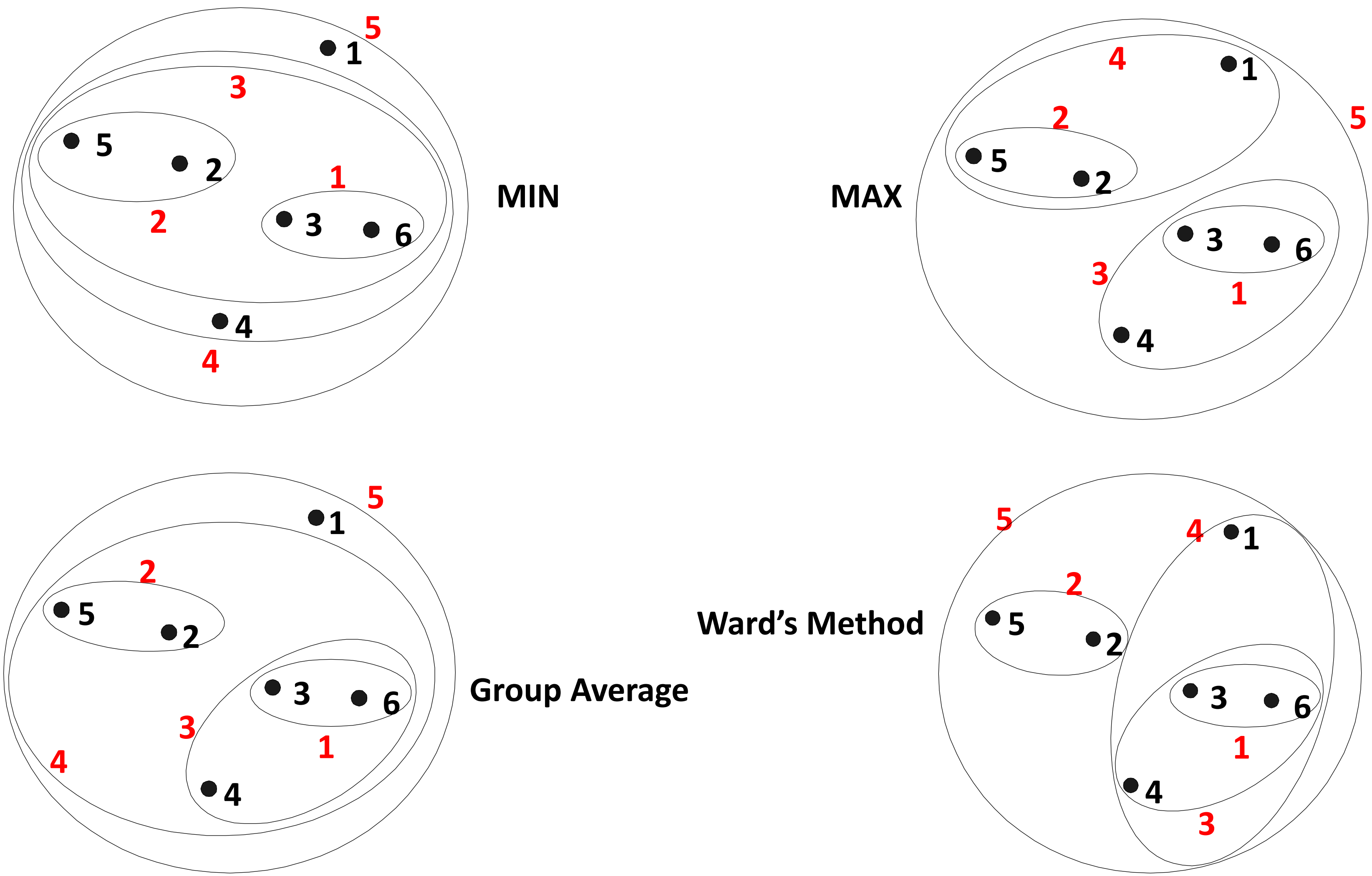
$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

- r_i: centroid of C_i
- r_j: centroid of C_j
- r_{ij}: centroid of C_{ij}

Ward's distance for clusters

- Similar to group average and centroid distance
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
  - Can be used to initialize k-means

Hierarchical Clustering: Comparison



Modern Clustering algorithms

Master in High Performance Computing

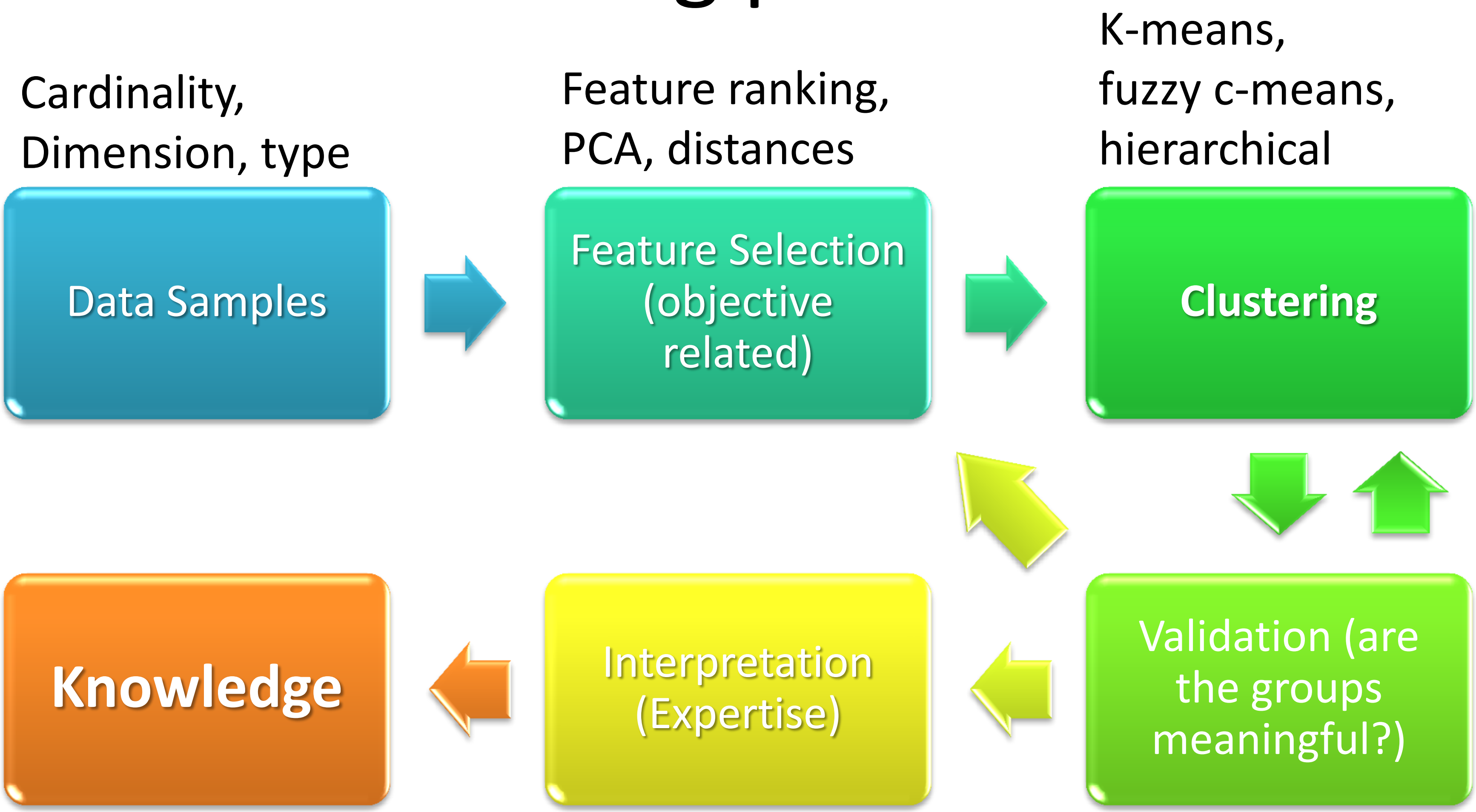
Previously at lesson one…

- **Data samples:** Characterized by the number of elements (cardinality), the number of features (dimensionality) and the type of these features.
- **Feature selection:** We employ it when a ground truth is available. The method of choice depends on the nature of the ground truth and the nature of the features.
- **Dimensionality reduction:** Does not require a ground truth. It usually simplifies the data and allows a more efficient representation or treatment. Although we focused on PCA, there are many alternatives.
- **Similarities and distances:** a pairwise measure of how similar are the elements of the data set. Are the basis for many clustering methods and they should be selected according with the nature of the dataset. More complicated distances (like the geodesic distance) are available and can be useful in difficult analyses.

Previously at lesson two…

- **Flat clustering:** Provides a hard partition of your dataset in k groups. (k-means)
- **Fuzzy clustering:** Provides a soft partition of your dataset in k groups. A soft partition implies that every point belongs to a cluster with a certain degree of membership. (fuzzy c-means)
- **Hierarchical clustering:** It can be done in an agglomerative or in a divisive way. Its output is a dendrogram: a tree-like diagram that records the sequences of merges or splits. Agglomerative cluster algorithms differ in the way they compute the inter-cluster distance (Single-link, Complete-link, Ward's method...).

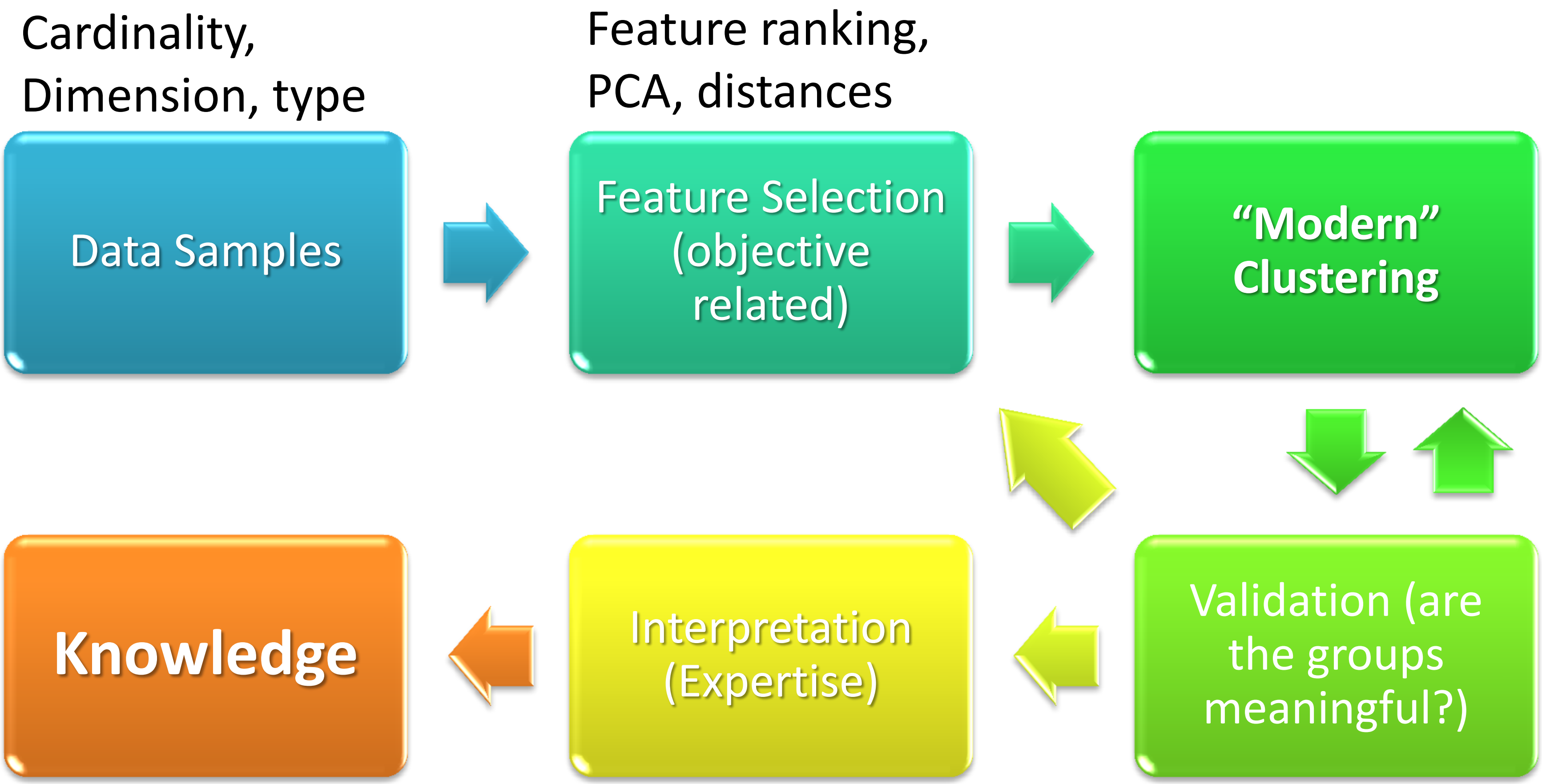
Clustering procedure



Some problems with the discussed algorithms

- K-means and fuzzy c-means are not well suited to deal with arbitrary shape (non-convex) clusters.
- Which intercluster distance shall we use for hierarchical clustering?
- What happens with noisy data sets?
- Which should be the final number of clusters?

Clustering procedure



Reviews on clustering algorithms

- There are hundreds of clustering algorithms, each of them with their advantages and withdrawals.
  - Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.
  - Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, *16*(3), 645-678.
  - Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), 651-666.
  - Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms.*Annals of Data Science*, *2*(2), 165-193.

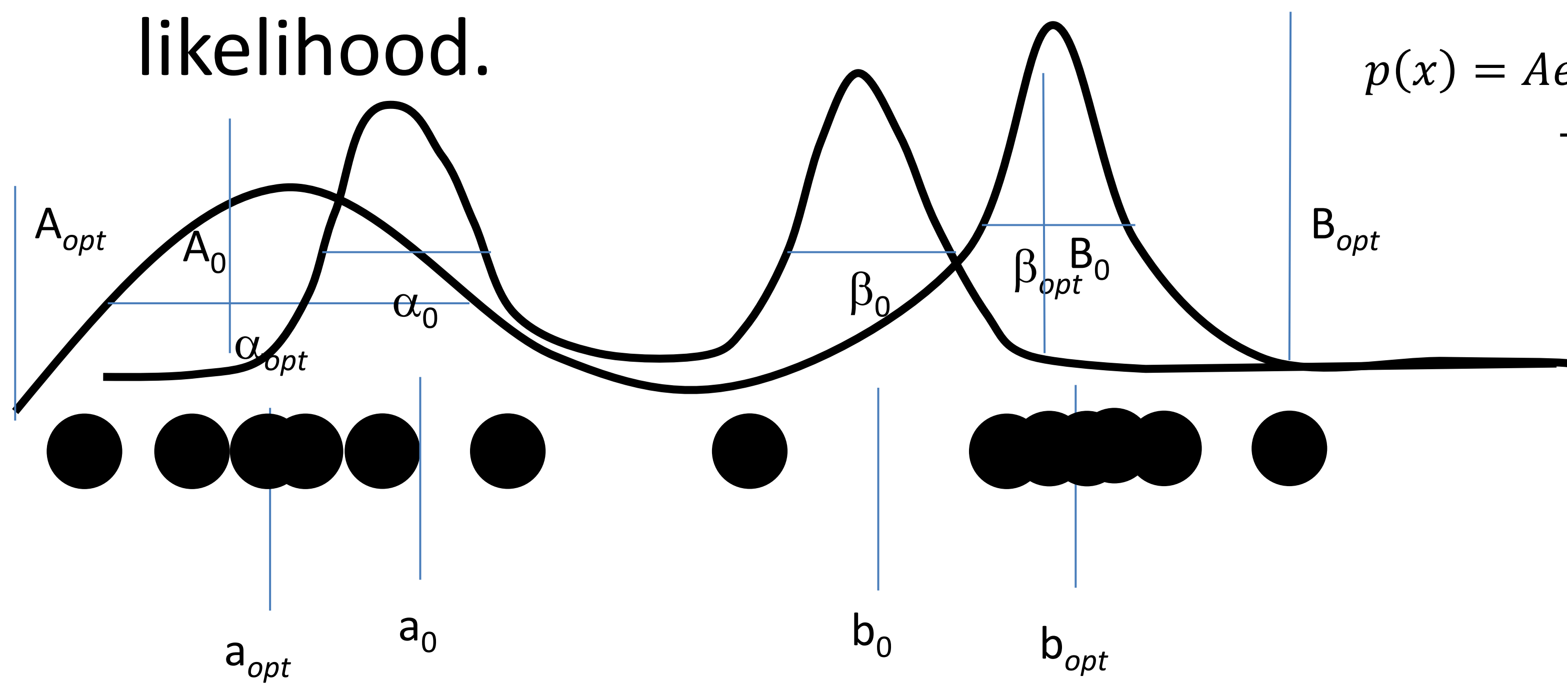
Expectation-Maximization clustering

Modern clustering algorithms (I)

Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.

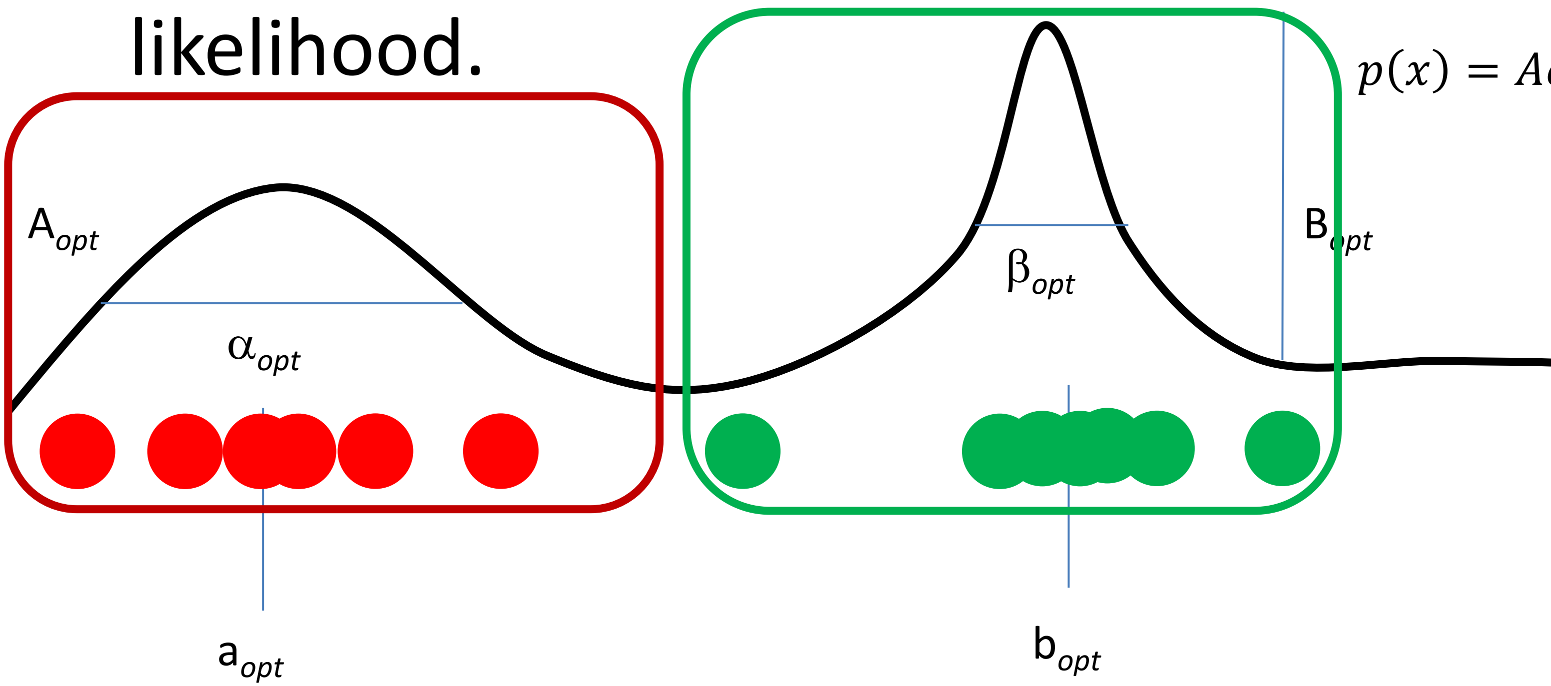
$$p(x) = Ae^{-a(x-\alpha)^2} + Be^{-b(x-\beta)^2}$$



Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.

$$p(x) = Ae^{-a(x-\alpha)^2} + Be^{-b(x-\beta)^2}$$



Expectation-maximization algorithm

- Iterative procedure to compute the ***Maximum Likelihood (ML)*** estimate – even in the presence of missing or hidden data
- **EM** consists of two steps:
  - **Expectation step:** the (missing) data are estimated given the observed data and current estimates of model parameters
  - **Maximization step:** The likelihood function is maximized under the assumption that the (missing) data are known

EM algorithm for mixture of Gaussians

- What is a mixture of **K** Gaussians?

$$p(x) = \sum_{k=1}^{K} \pi_k F(x \mid \Theta_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1$$

and **F(x|Θ)** is the Gaussian distribution with parameters **Θ = {μ,Σ}**

EM algorithm for mixture of Gaussians

- If all points **x϶X** are mixtures of **K** Gaussians then

$$p(X) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k F(x_i \mid \Theta_k)$$

- **Goal:** Find **$\pi_1, \ldots, \pi_k$** and **$\Theta_1, \ldots, \Theta_k$** such that **P(X)** is maximized
- Or, **ln(P(X))** is maximized:

$$L(\Theta) = \sum_{i=1}^{n} \ln \left\{ \sum_{k=1}^{K} \pi_k F(x_i \mid \Theta_k) \right\}$$

Mixtures of Gaussians -- notes

- Every point $\mathbf{x_i}$ is ***probabilistically*** assigned (generated) to (by) the $\mathbf{k}$-th Gaussian

- Probability that point $\mathbf{x_i}$ is generated by the $\mathbf{k}$-th Gaussian is

$$w_{ik} = \frac{\pi_k F(x_i \mid \Theta_k)}{\sum_{j=1}^{K} \pi_j F(x_i \mid \Theta_j)}$$

Mixtures of Gaussians -- notes

- Every Gaussian (cluster) C_k has an effective number of points assigned to it N_k

$$N_k = \sum_{i=1}^{n} w_{ik}$$

- With mean

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} x_i$$

- And variance

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} \left(x_i - \mu_k\right) x_i \left(x_i - \mu_k\right)^T$$

EM for Gaussian Mixtures

- Initialize the means μ_k, variances Σ_k ($\Theta_k=(\mu_k,\Sigma_k)$) and mixing coefficients π_k, and evaluate the initial value of the loglikelihood

- **Expectation step**: Evaluate weights

$$w_{ik} = \frac{\pi_k F(x_i \mid \Theta_k)}{\sum_{j=1}^{K} \pi_j F(x_i \mid \Theta_j)}$$

EM for Gaussian Mixtures

- **Maximization step**: Re-evaluate parameters

$$\mu_k^{new} = \frac{1}{N_k}\sum_{i=1}^{n} w_{ik} x_i$$

$$\Sigma_k^{new} = \frac{1}{N_k}\sum_{i=1}^{n} w_{ik}\left(x_i - \mu_k^{new}\right) x_i \left(x_i - \mu_k^{new}\right)^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

- Evaluate **L(Θ$^{\mathbf{new}}$)** and stop if converged

EM characteristics

- Notice the similarity between EM for Normal mixtures and K-means: The expectation step is the assignment, while the maximization step is the update of the centers.
- If the model (k, functional form) is not realistic, neither will the results.
- It is not guaranteed to reach the global optimum.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Modern clustering algorithms (II)

DBSCAN

Density-based Clustering locates regions of high density that are separated from one another by regions of low density.

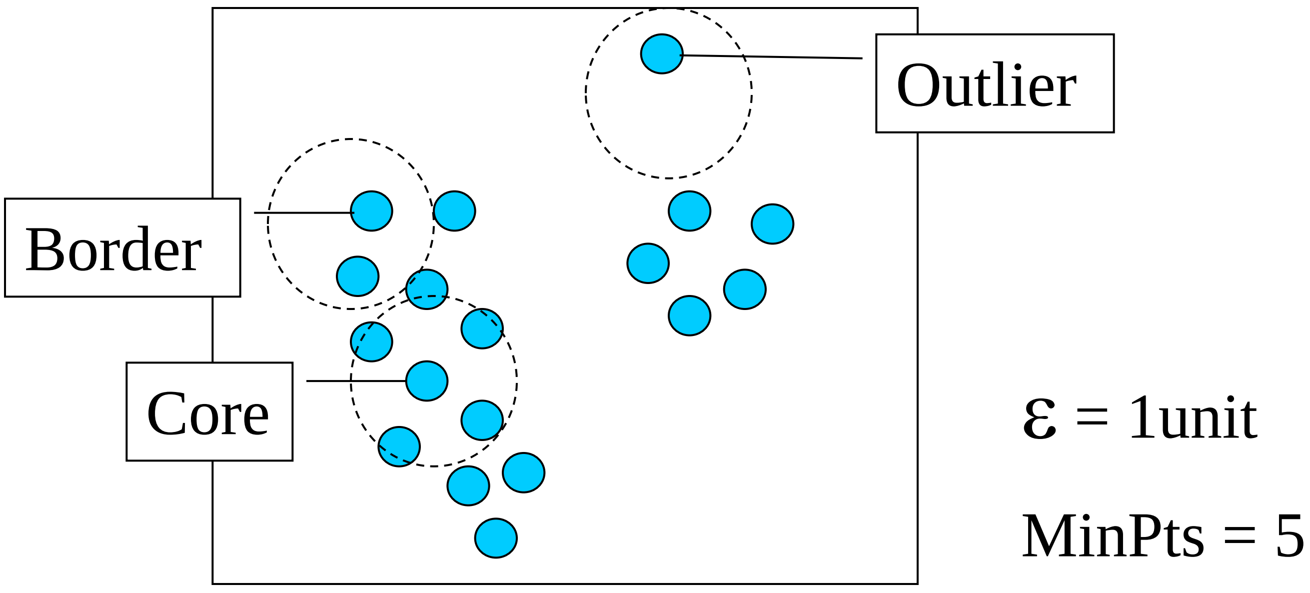
- Density = number of points within a specified radius (Eps)

- DBSCAN is a density-based algorithm.
- A point is a core point if it has more than a specified number of points (MinPts) within Eps. These are points that are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

DBSCAN

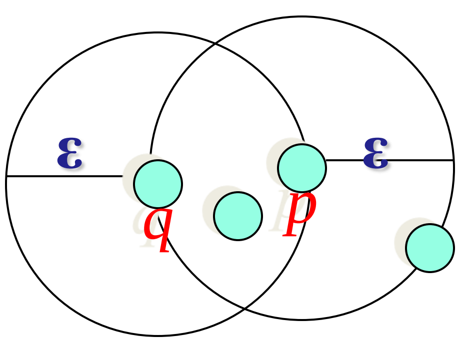
- A noise point is any point that is not a core point or a border point.
- Any two core points are close enough– within a distance *Eps* of one another – are put in the same cluster
- Any border point that is close enough to a core point is put in the same cluster as the core point
- Noise points are discarded

Border & Core



Concepts: ε-Neighborhood

- **ε-Neighborhood** - Objects within a radius of ε from an object. (epsilon-neighborhood)
- **Core objects** - ε-Neighborhood of an object contains at least **MinPts** of objects



ε-Neighborhood of *p*

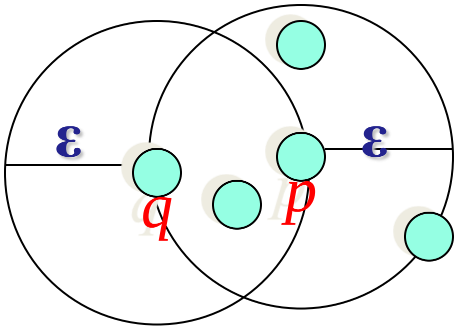
ε-Neighborhood of *q*

p is a core object (MinPts = 4)

q is not a core object

Concepts: Reachability

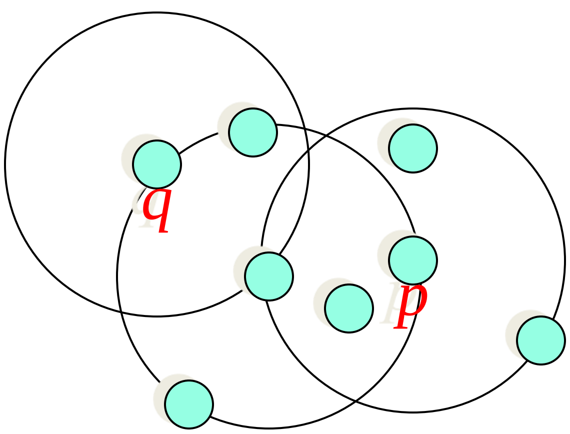
- **Directly density-reachable**
  - An object q is directly density-reachable from object p if q is within the ε-Neighborhood of p and p is a core object.



- q is directly density-reachable from p
- p is not directly density-reachable from q?

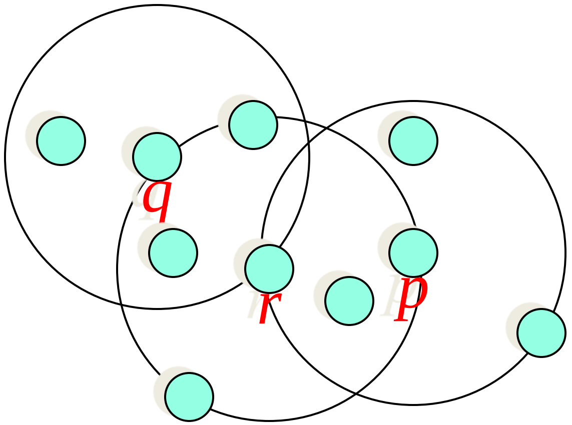
Concepts: Reachability

- **Density-reachable:**
  - An object p is density-reachable from q w.r.t ε and *MinPts* if there is a chain of objects $p_1,\ldots,p_n$, with $p_1=q$, $p_n=p$ such that p_{i+1} is directly density-reachable from p_i w.r.t ε and *MinPts* for all $1 <= i <= n$
    - q is density-reachable from p
    - p is not density- reachable from q?
    - Transitive closure of direct density-Reachability, asymmetric



Concepts: Connectivity

- **Density-connectivity**
  - Object *p* is density-connected to object *q* w.r.t ε and *MinPts* if there is an object *o* such that both *p* and *q* are density-reachable from *o* w.r.t ε and *MinPts*

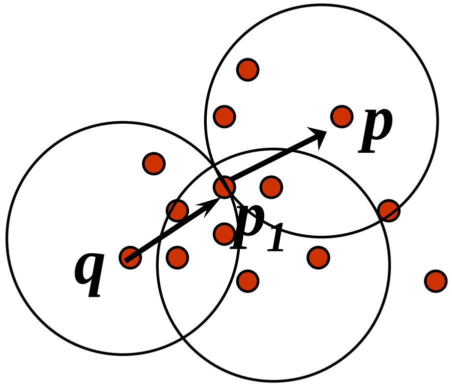


- *P* and *q* are density-connected to each other by *r*
- Density-connectivity is symmetric

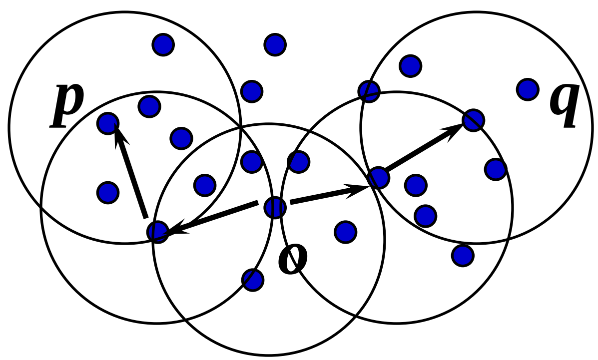
Concepts: cluster & noise

- **Cluster**: a cluster ***C*** in a set of objects ***D*** w.r.t ε and *MinPts* is a non empty subset of ***D*** satisfying
  - Maximality: For all *p*, *q* if $p \in$ ***C*** and if *q* is density-reachable from *p* w.r.t ε and *MinPts*, then also $q \in$ ***C.***
  - Connectivity: for all $p, q \in$ ***C***, *p* is density-connected to *q* w.r.t ε and *MinPts* in ***D.***
  - ***Note:*** cluster contains *core objects* as well as *border objects*
- **Noise:** objects which are not directly density-reachable from at least one core object.

(Indirectly) Density-reachable:



Density-connected

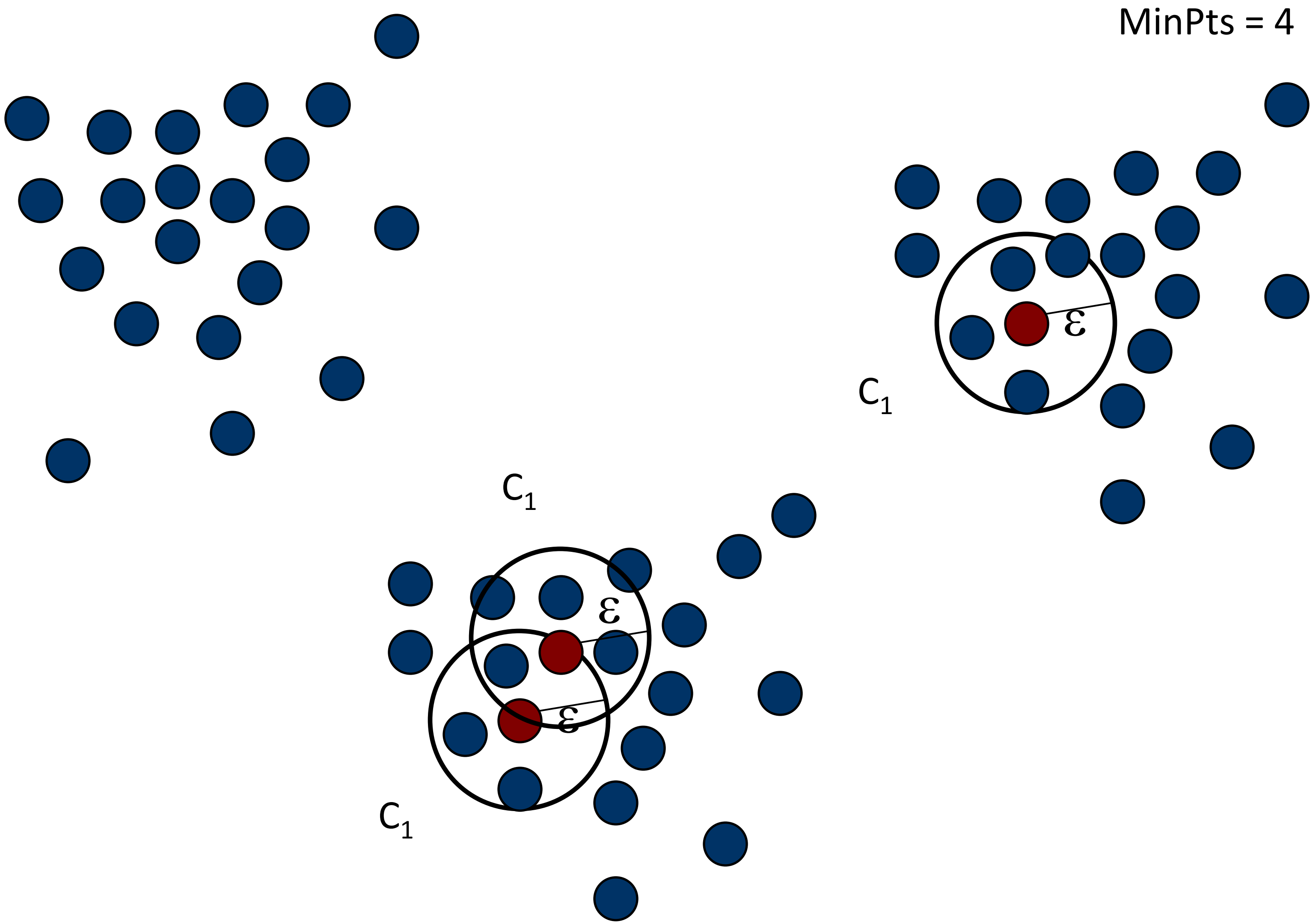


DBSCAN: The Algorithm

- select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ε and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Result is independent of the order of processing the points

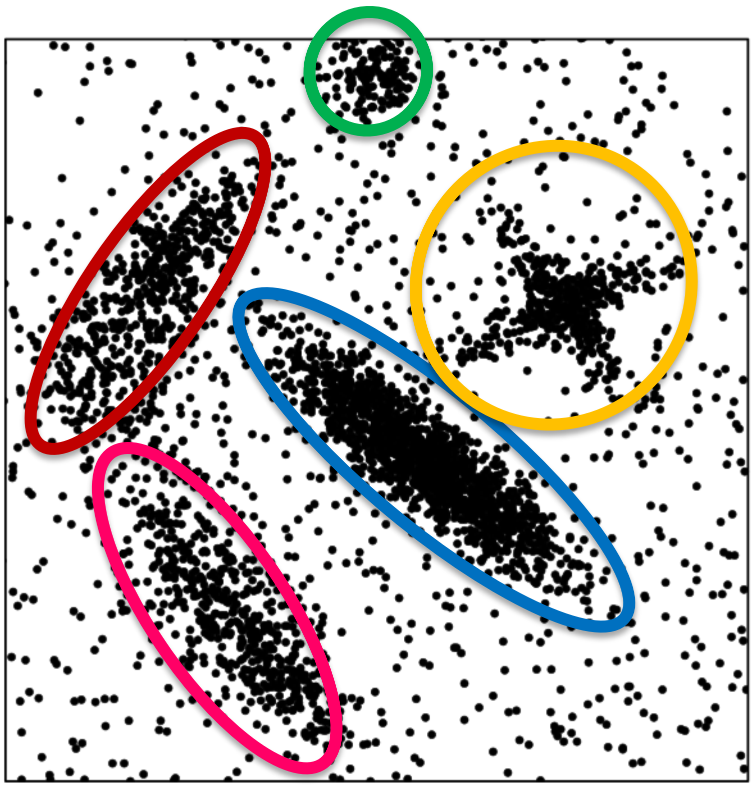
An Example

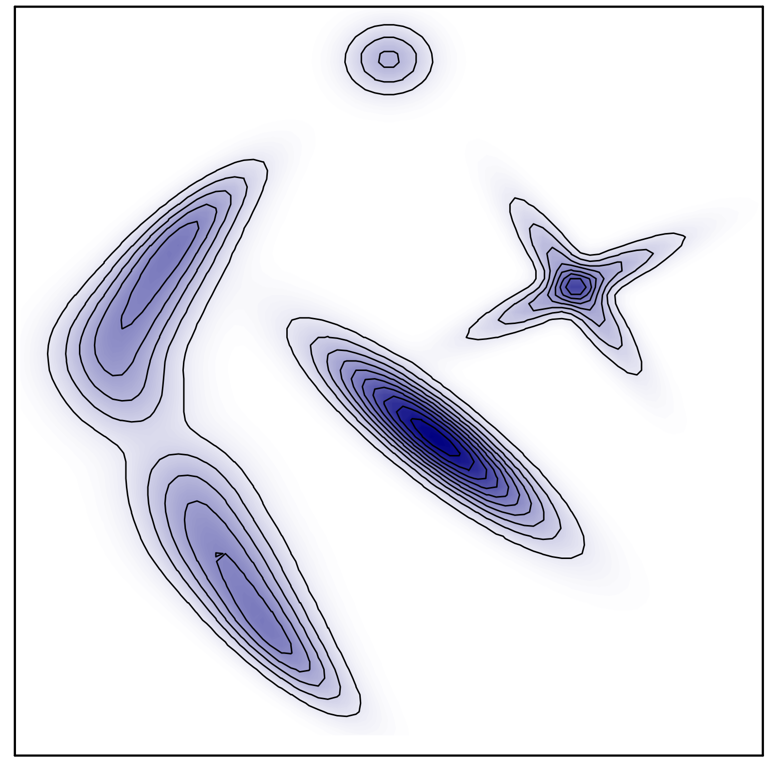


Fast Search and Find of Density Peaks

Modern clustering algorithms (III)

Density Peaks clustering





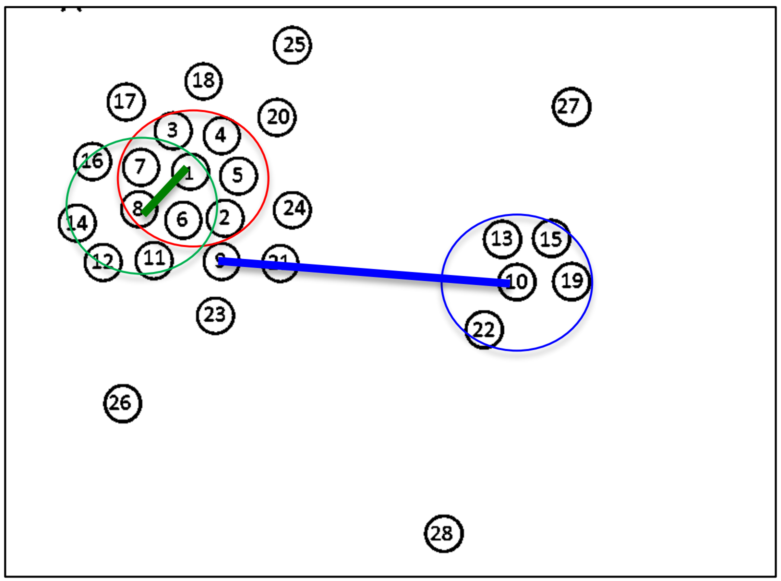
Clusters = peaks in the density of points*

= peaks in the "mother" probability distribution

*Idea already present in DB-SCAN

Density Peaks: δ concept

Clustering in a 2-dimensional space



1) Compute the local density (ρ) around each point

ρ(1)=7

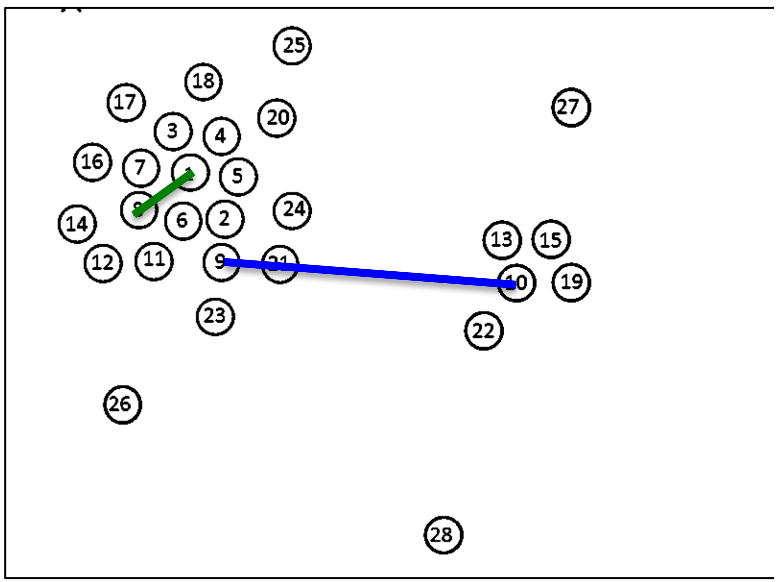
ρ(8)=5

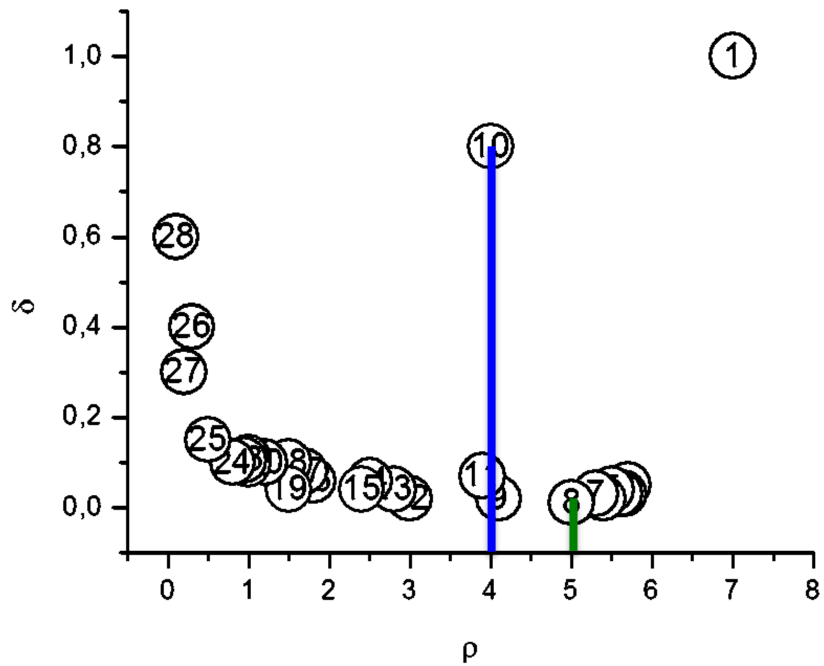
ρ(10)=4

2) For each point compute the distance from all the points with higher density. Take the minimum value.

Density Peaks decision graph

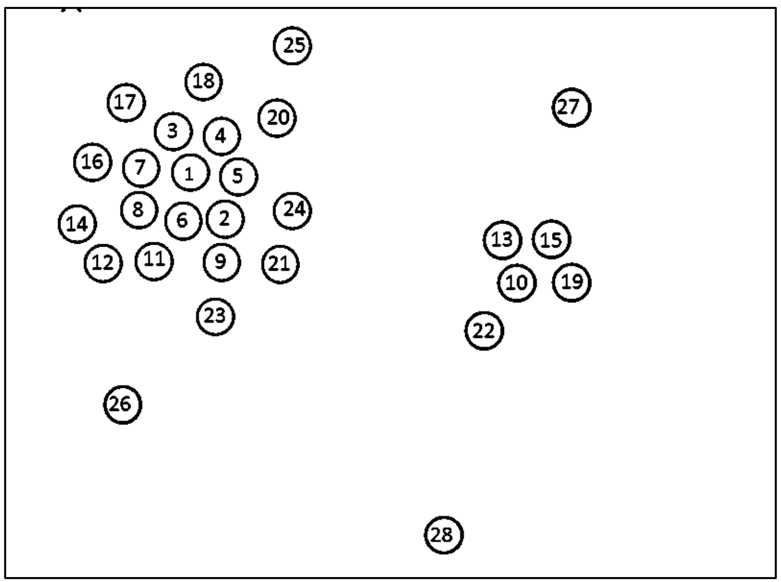
3) For each point, plot the minimum distance as a function of the density.

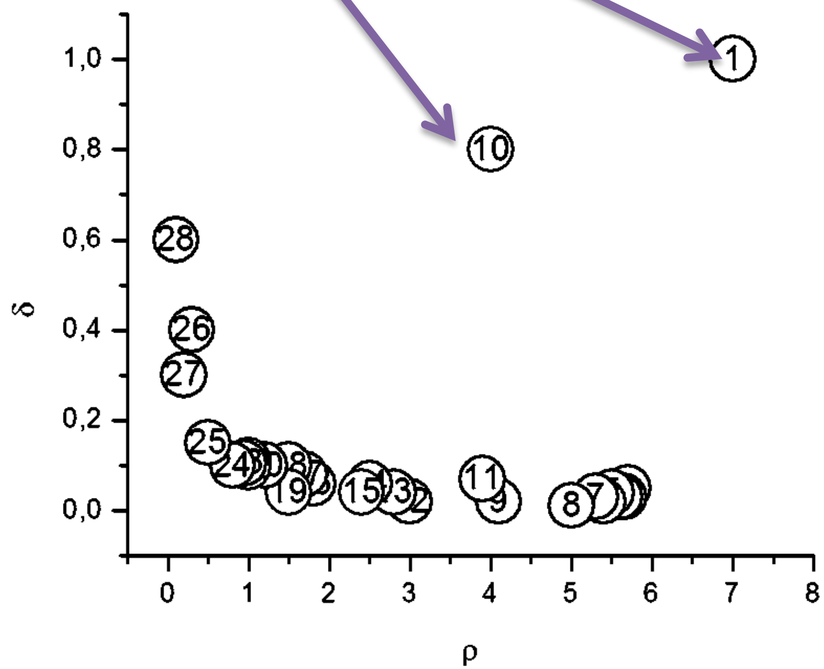




Density Peaks decision graph

4) the “outliers” in this graph are the cluster centers

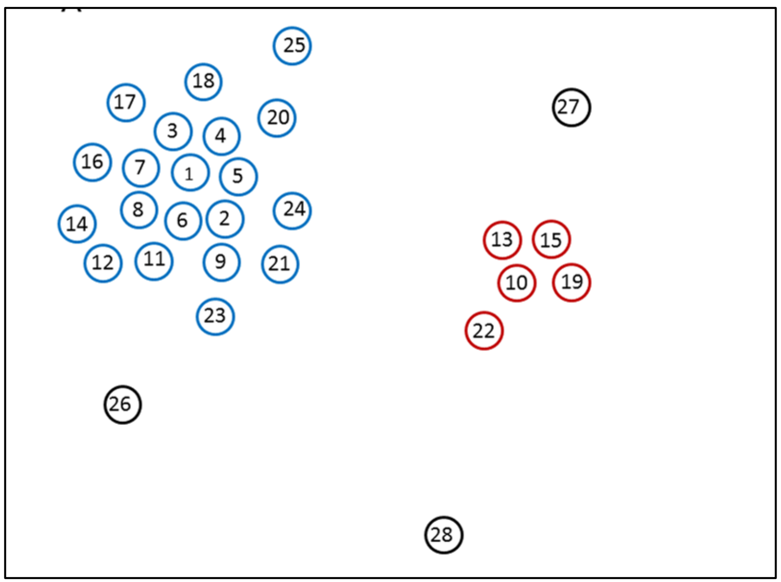


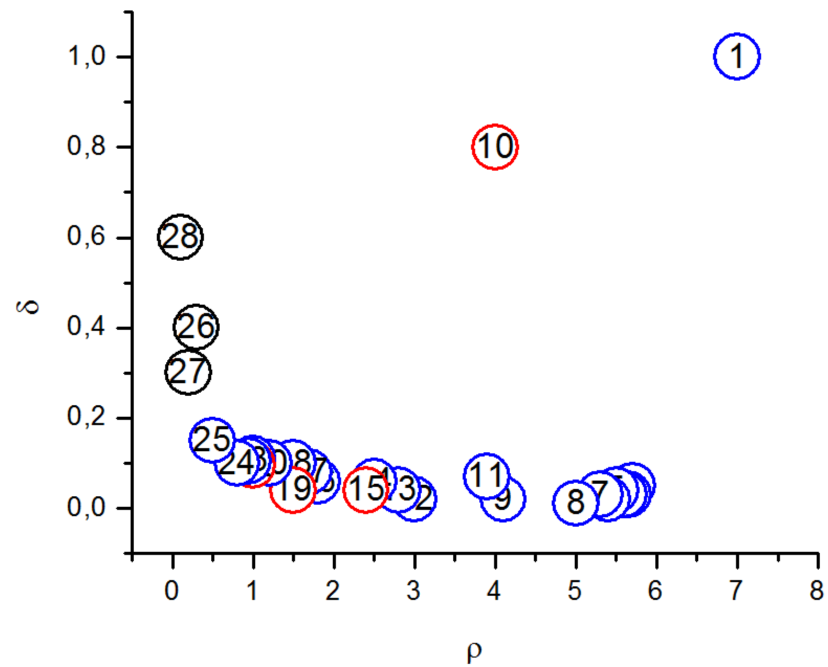


Density Peaks decision graph

4)) the “outliers” in this graph are the cluster centers

5) Assign each point to the same cluster of its nearest neighbor of higher density





Density Peaks algorithm

Given a distance matrix d_{ij}, for each data point i compute:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \sim \sum_j e^{-\left(d_{ij}/d_c\right)^2}$$

(number of data points within a distance d_c or density estimation)

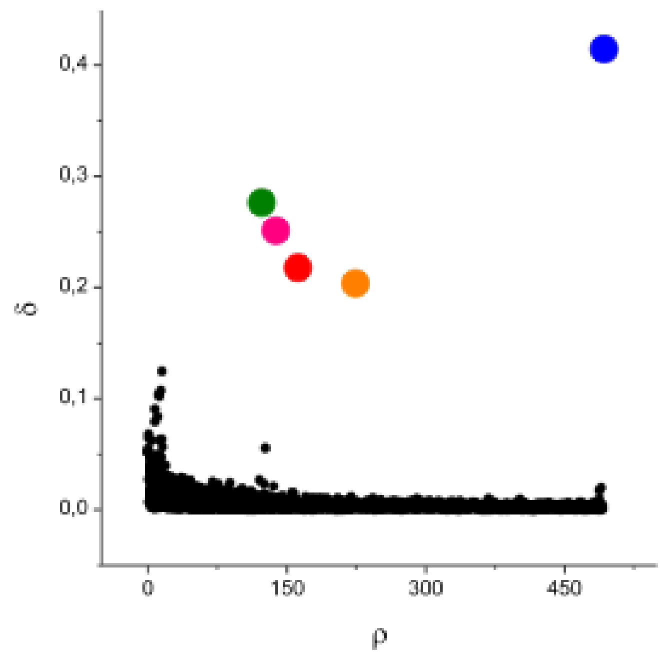
$$\delta_i = \min_{\rho_i < \rho_j} (d_{ij})$$

(distance of the closest data point of higher density)

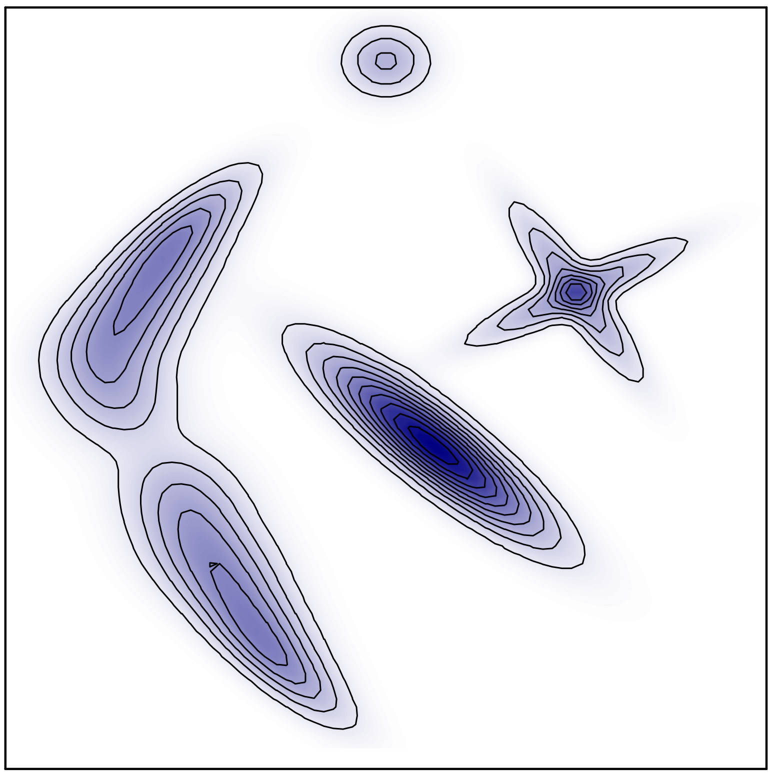
Make the Decision graph:
Plot δ as a function of ρ

One free parameter: the cutoff distance d_c

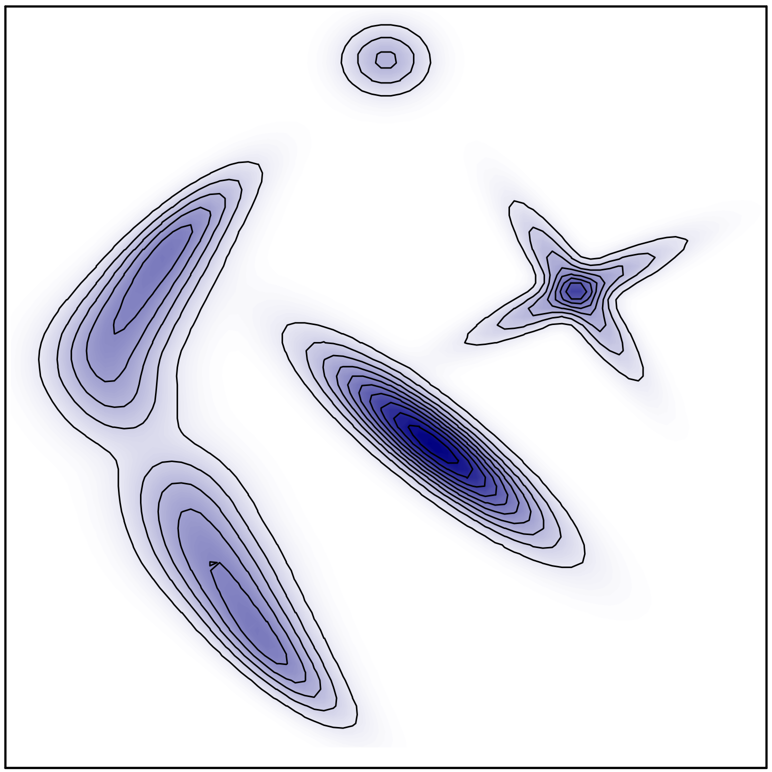
However: the method is only sensitive to the relative density of two data points, not to the absolute value of ρ

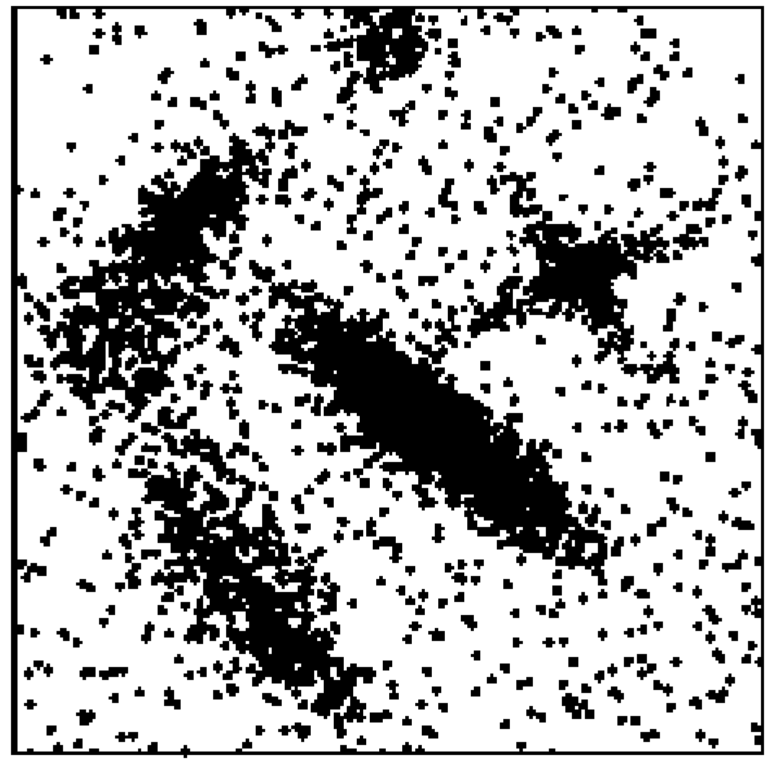


Density Peaks clustering example

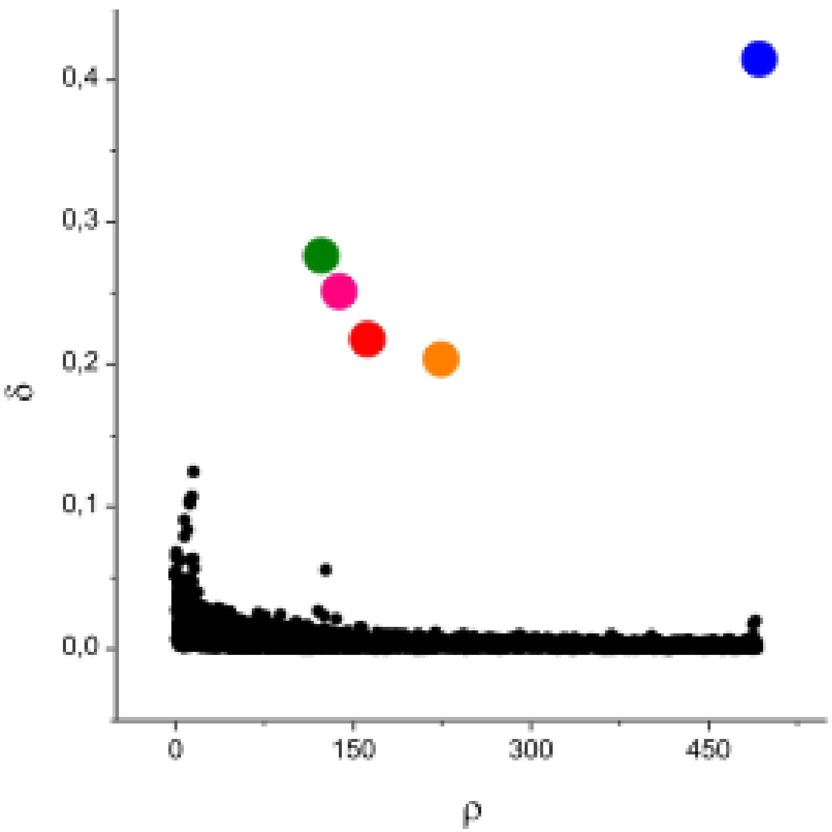


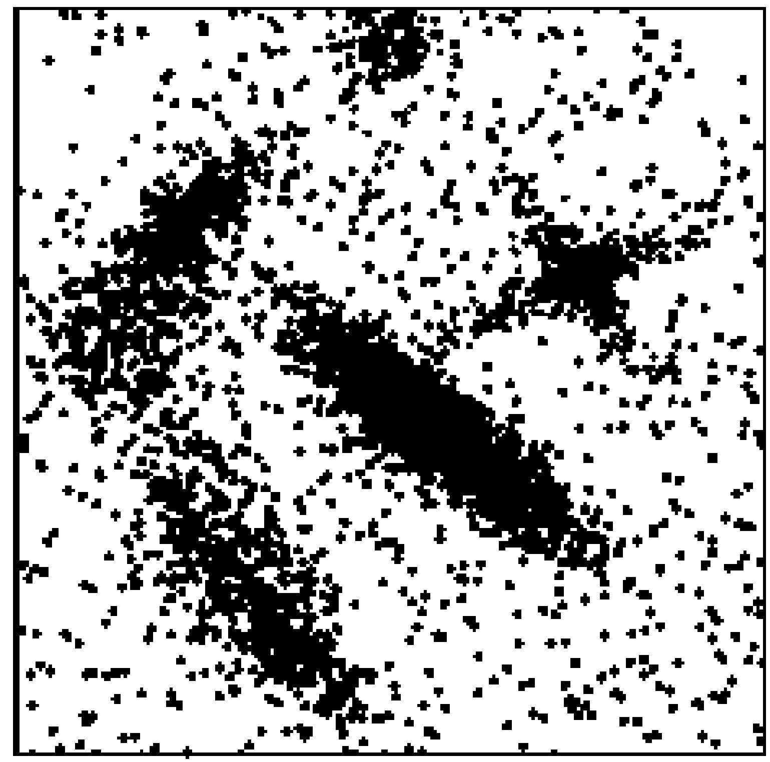
Density Peaks clustering example



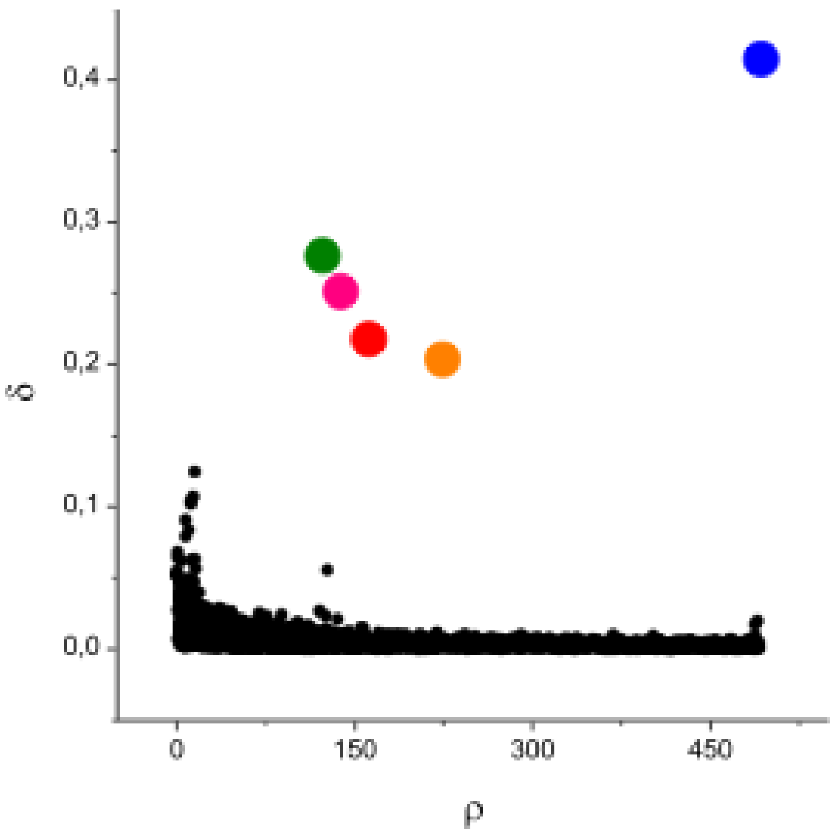


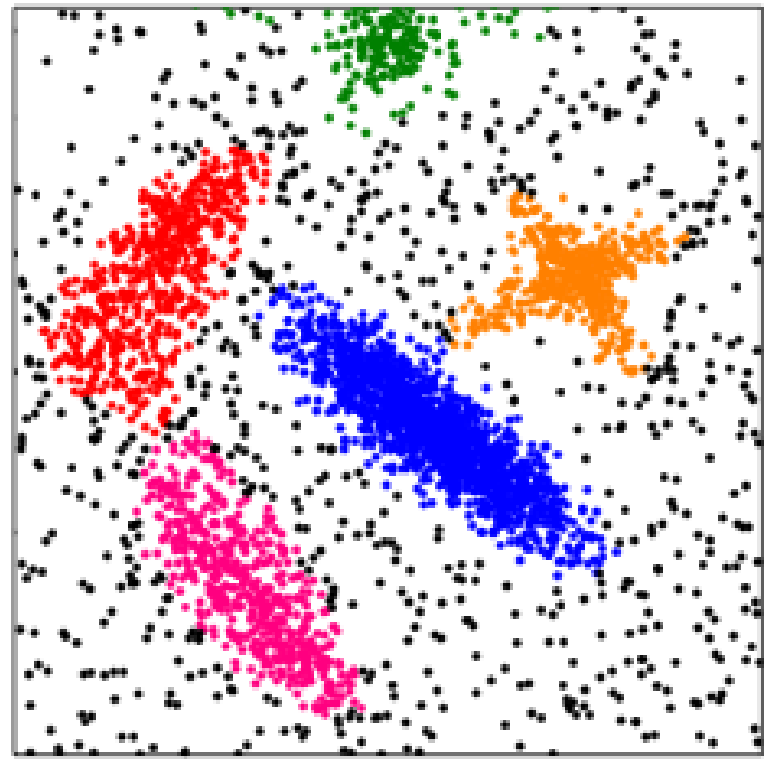
Density Peaks clustering example



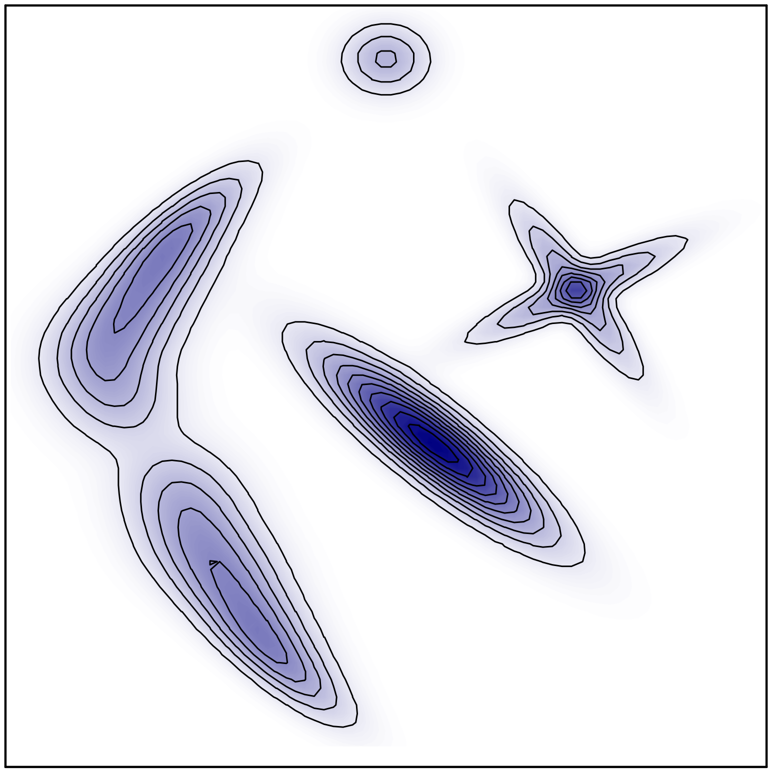


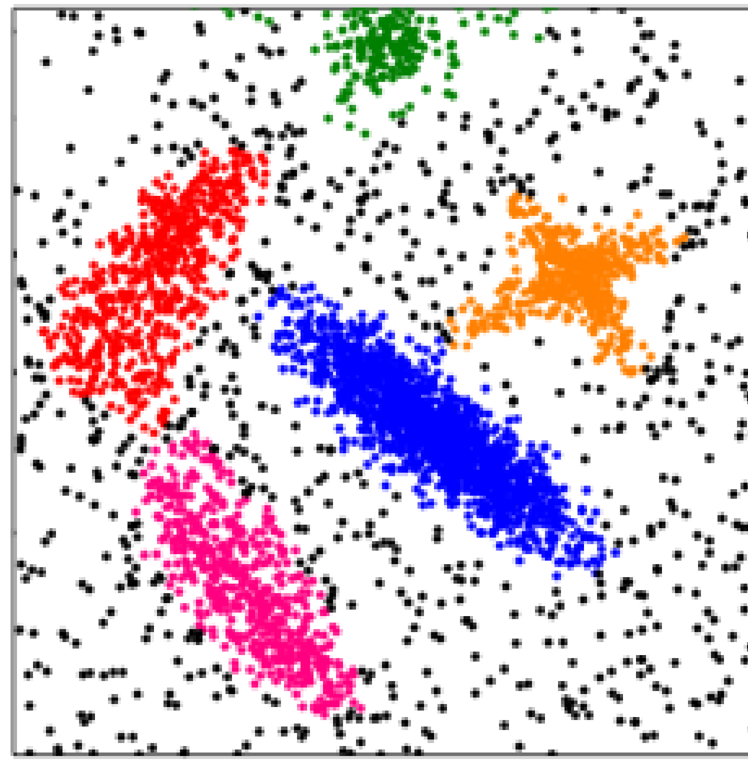
Density Peaks clustering example





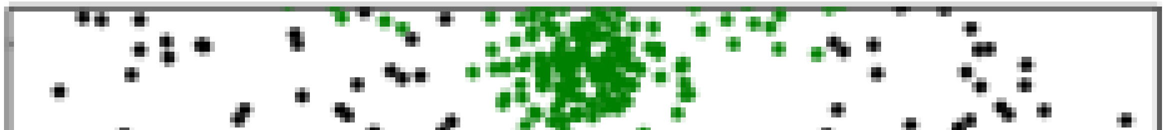
Density Peaks clustering example

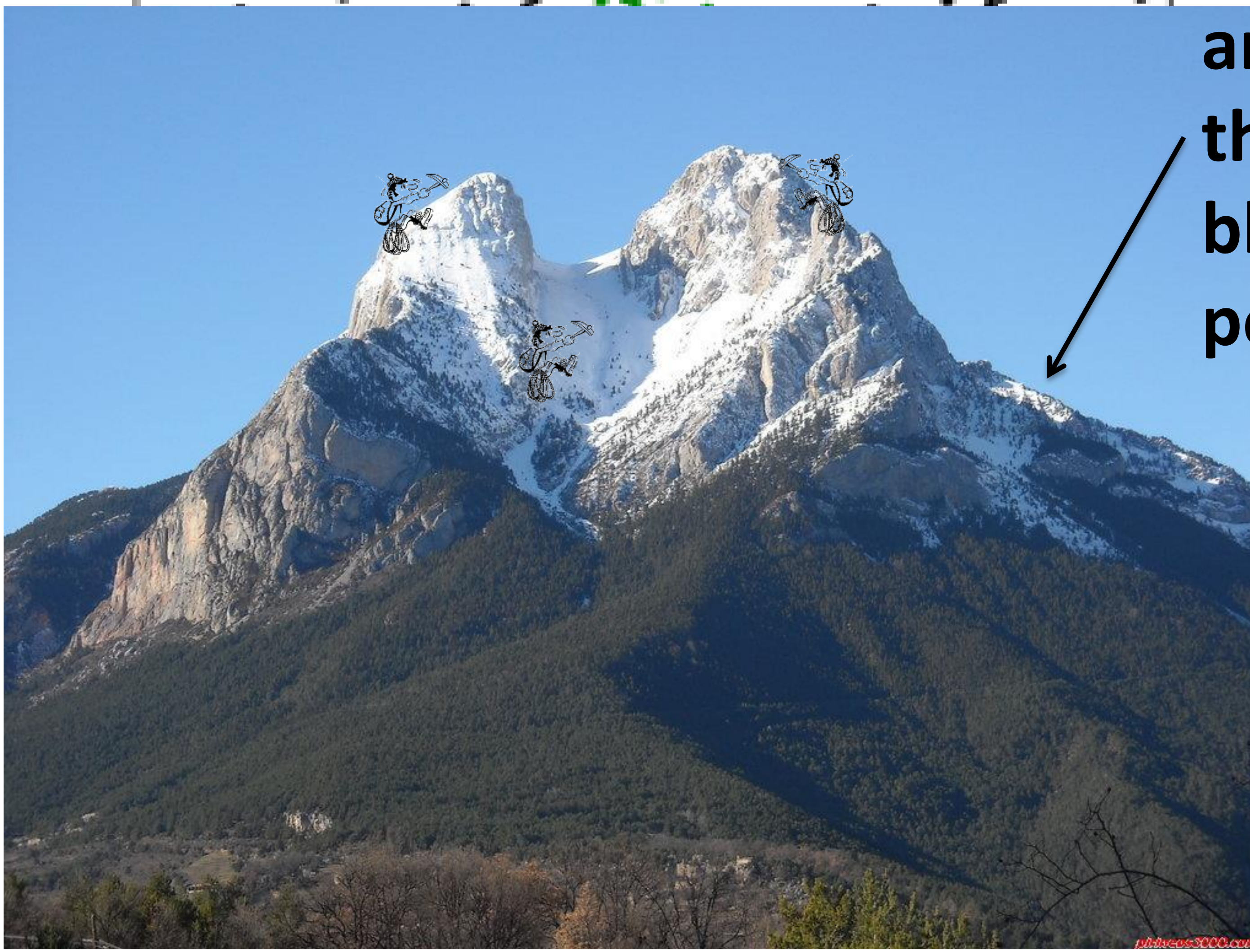




Density Peaks: Halo

Halo





What are these black points?

Border definition

- A point i of cluster A is border point if it has, within d_c, a point j belonging to another cluster.
- The border density of A is, then, the higher density among all the border points belonging to A.
- The border point definition is strongly dependent on d_c, but it can be easily correct if we add the condition that j does not have any nearest point belonging to A.

Some ideas about Cluster Validation

Master in High Performance Computing

Cluster validation

Supervised classification:

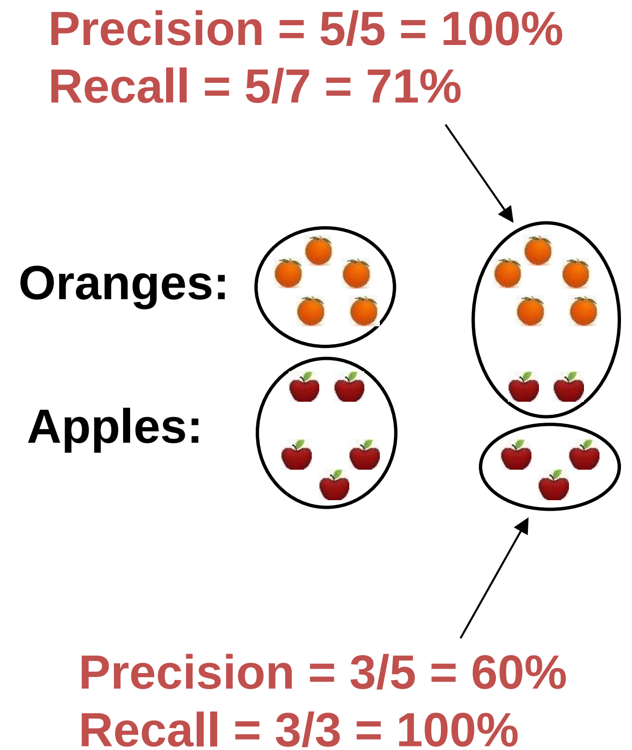
- Class labels known for ground truth
- Accuracy, precision, recall

Cluster analysis

- No class labels

Validation need to:

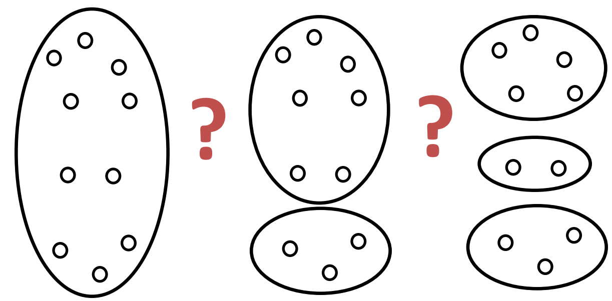
- Compare clustering algorithms
- Solve number of clusters
- Avoid finding patterns in noise



Measuring clustering validity

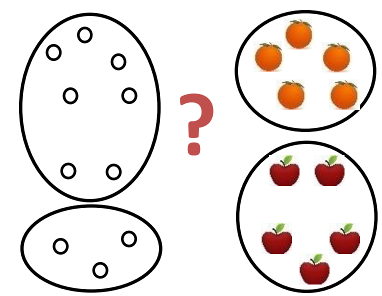
Internal Index:

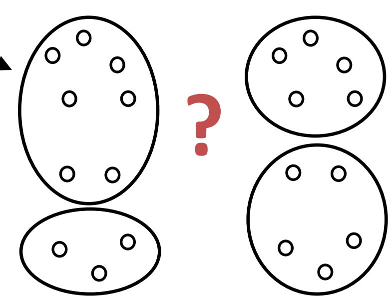
- Validate *without* external info
- With different number of clusters
- Solve the number of clusters



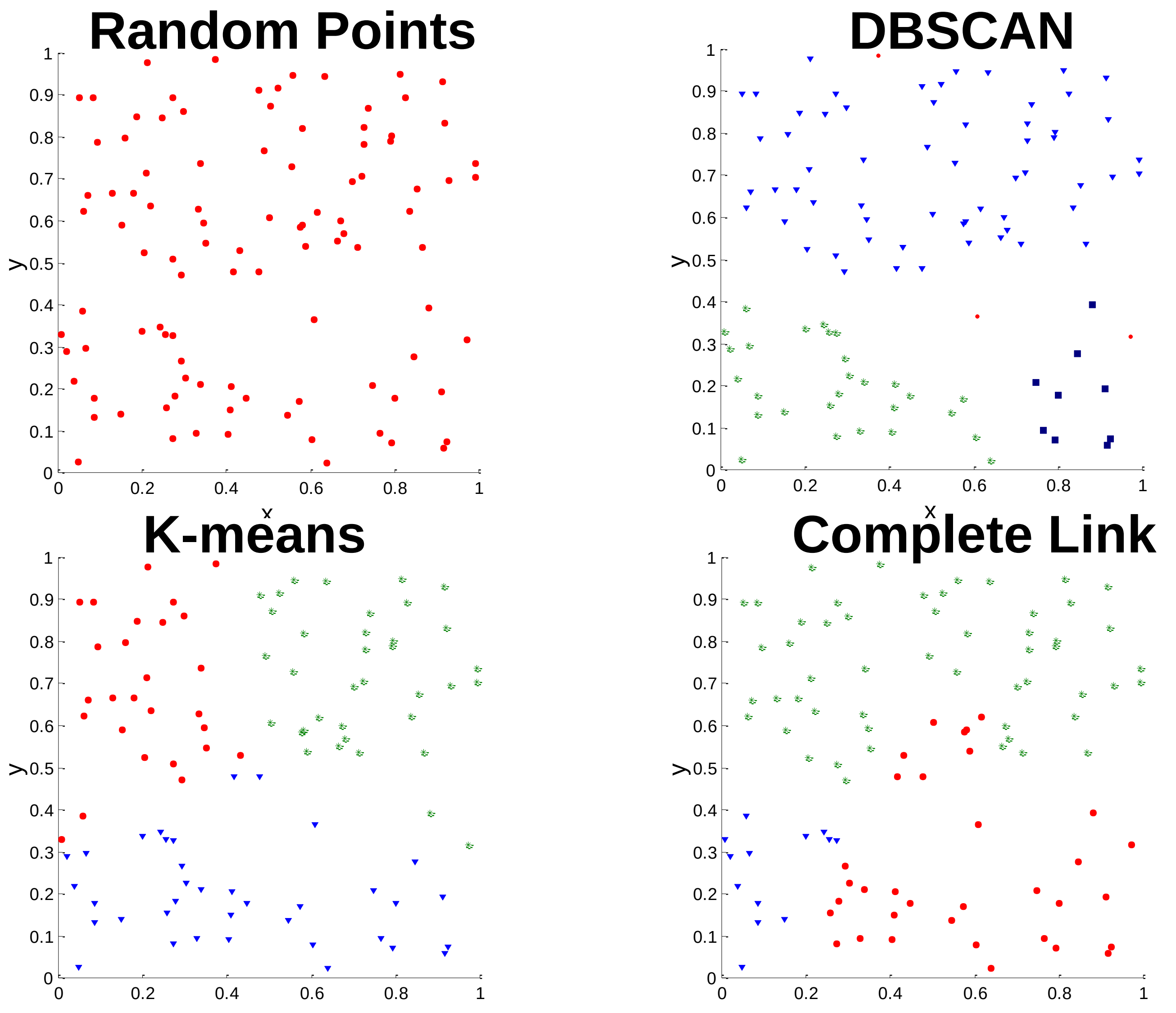
External Index

- Validate against ground truth
- Compare two clusters: (how similar)





Clustering of random data

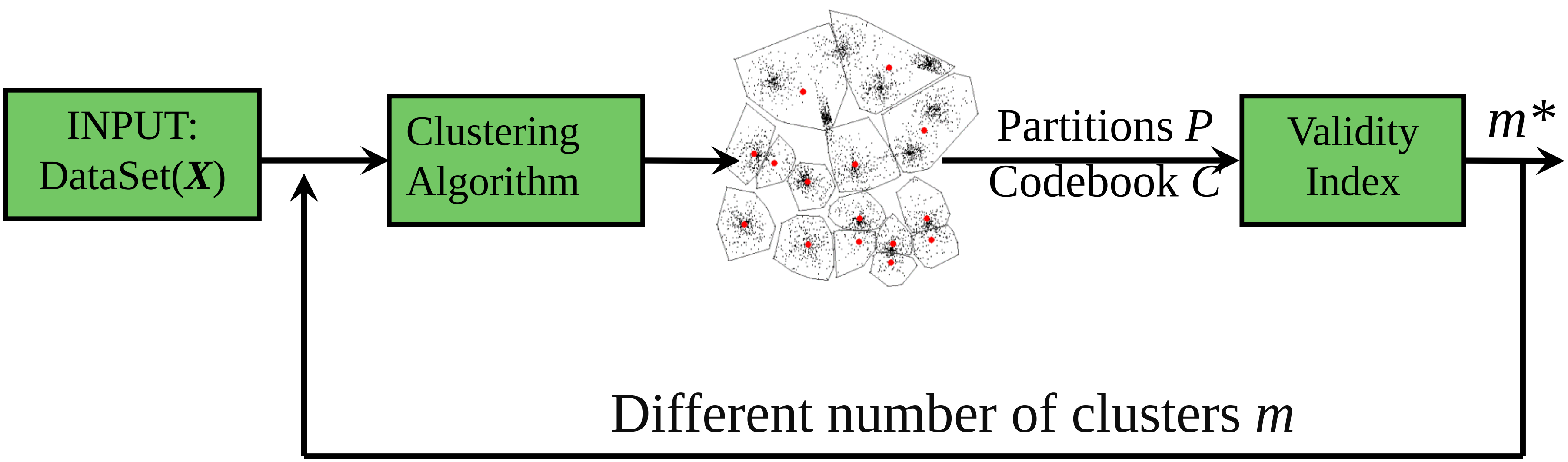


Cluster validation process

1. Distinguishing whether non-random structure actually exists in the data (one cluster).
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the number of clusters.

Cluster validation process

- **Cluster validation** refers to procedures that evaluate the results of clustering in a **quantitative** and **objective** fashion. [Jain & Dubes, 1988]
  - How to be “quantitative”: To employ the measures.
  - How to be “objective”: To validate the measures!

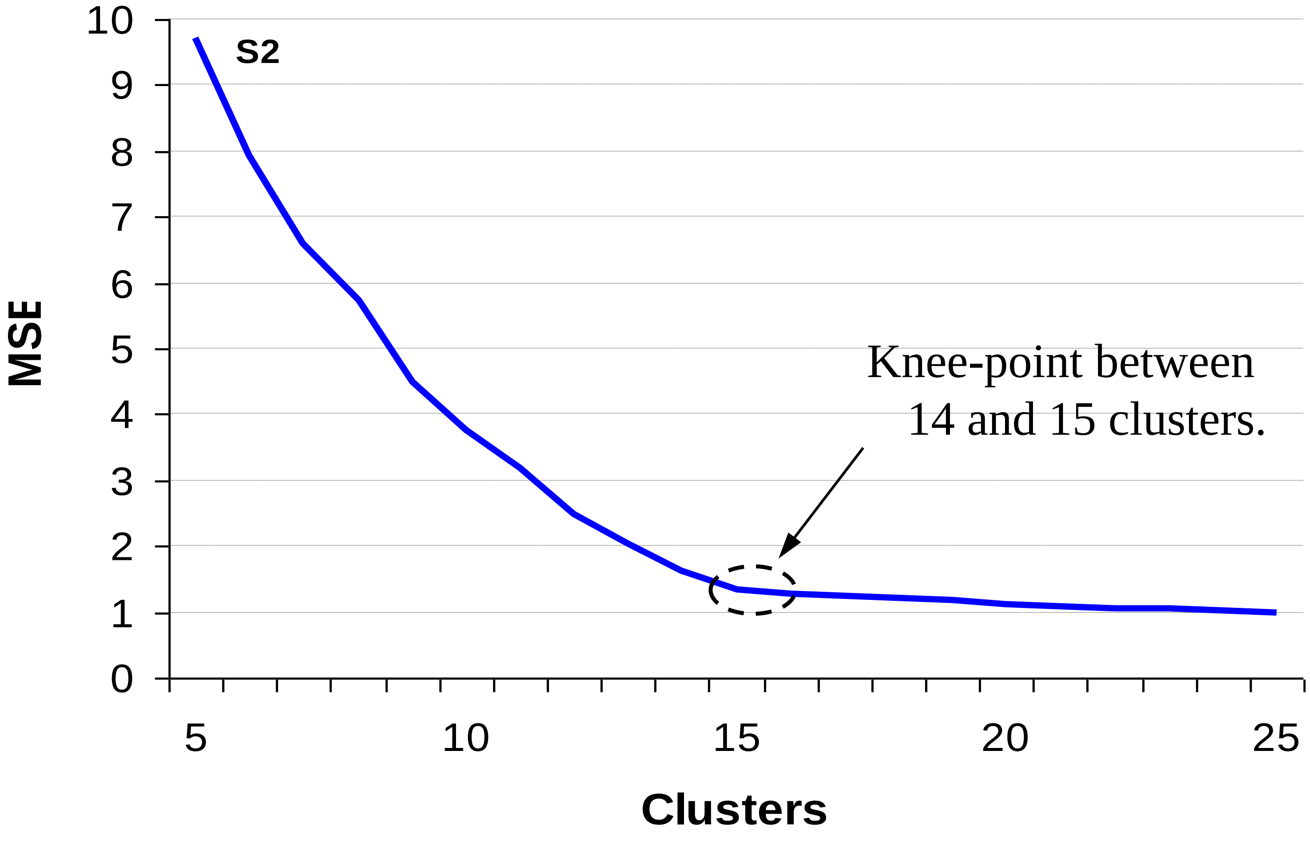


Internal indexes

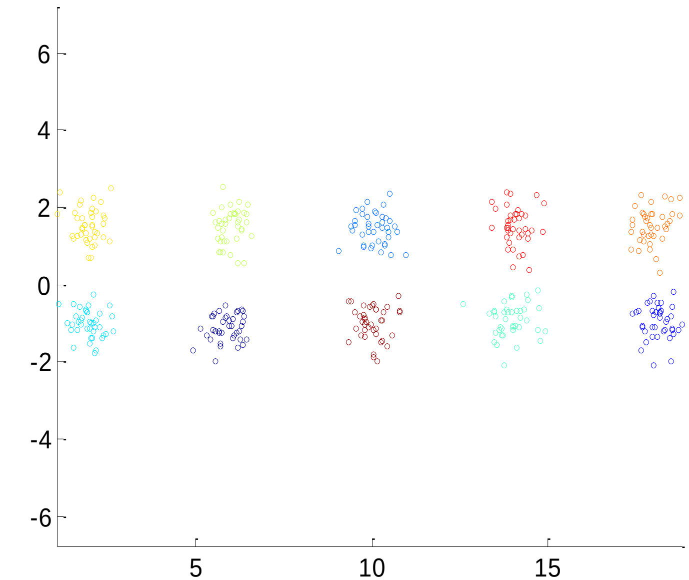
- Ground truth is rarely available but unsupervised validation must be done.
- Minimizes (or maximizes) internal index:
  - Variances of within cluster and between clusters
  - Rate-distortion method
  - F-ratio
  - Davies-Bouldin index (DBI)
  - Bayesian Information Criterion (BIC)
  - Silhouette Coefficient
  - Minimum description principle (MDL)
  - Stochastic complexity (SC)

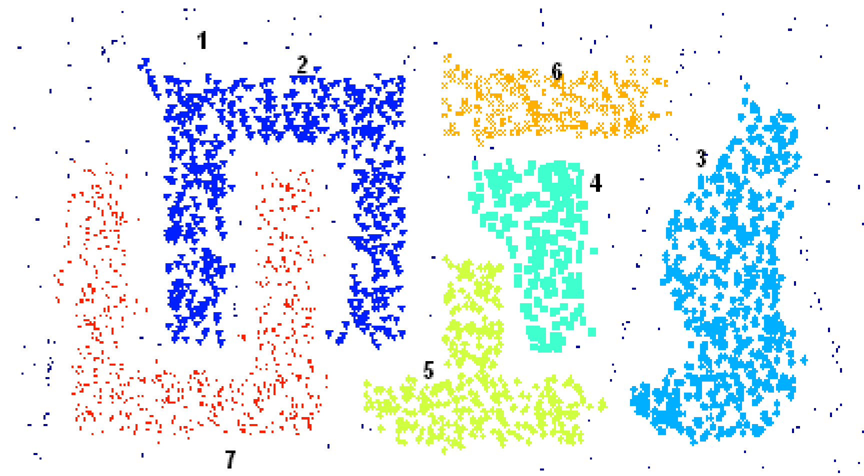
Mean square error (MSE)

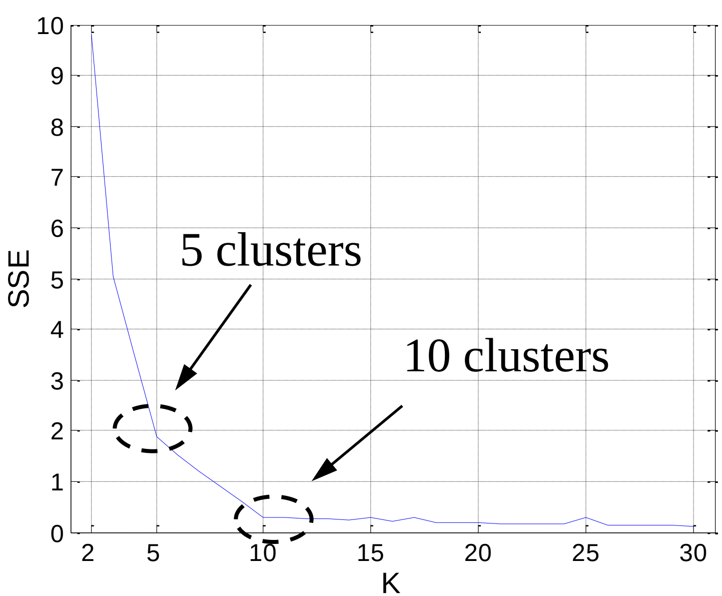
- The more clusters the smaller the MSE.
- Small knee-point near the correct value.
- But how to detect?

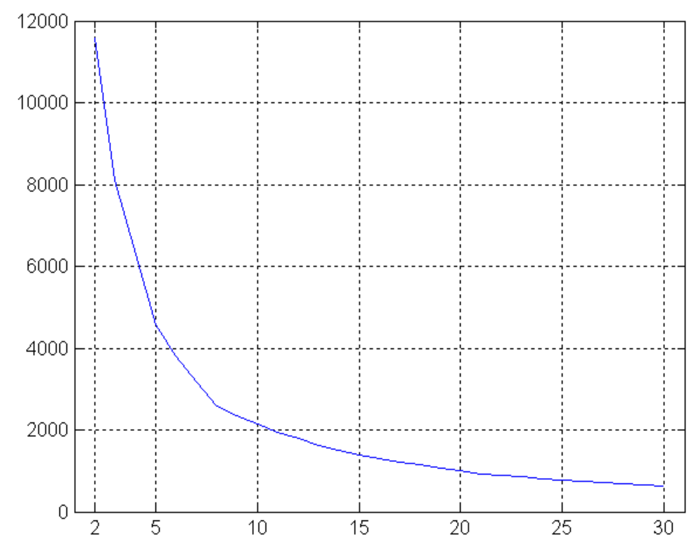


Mean square error (MSE)



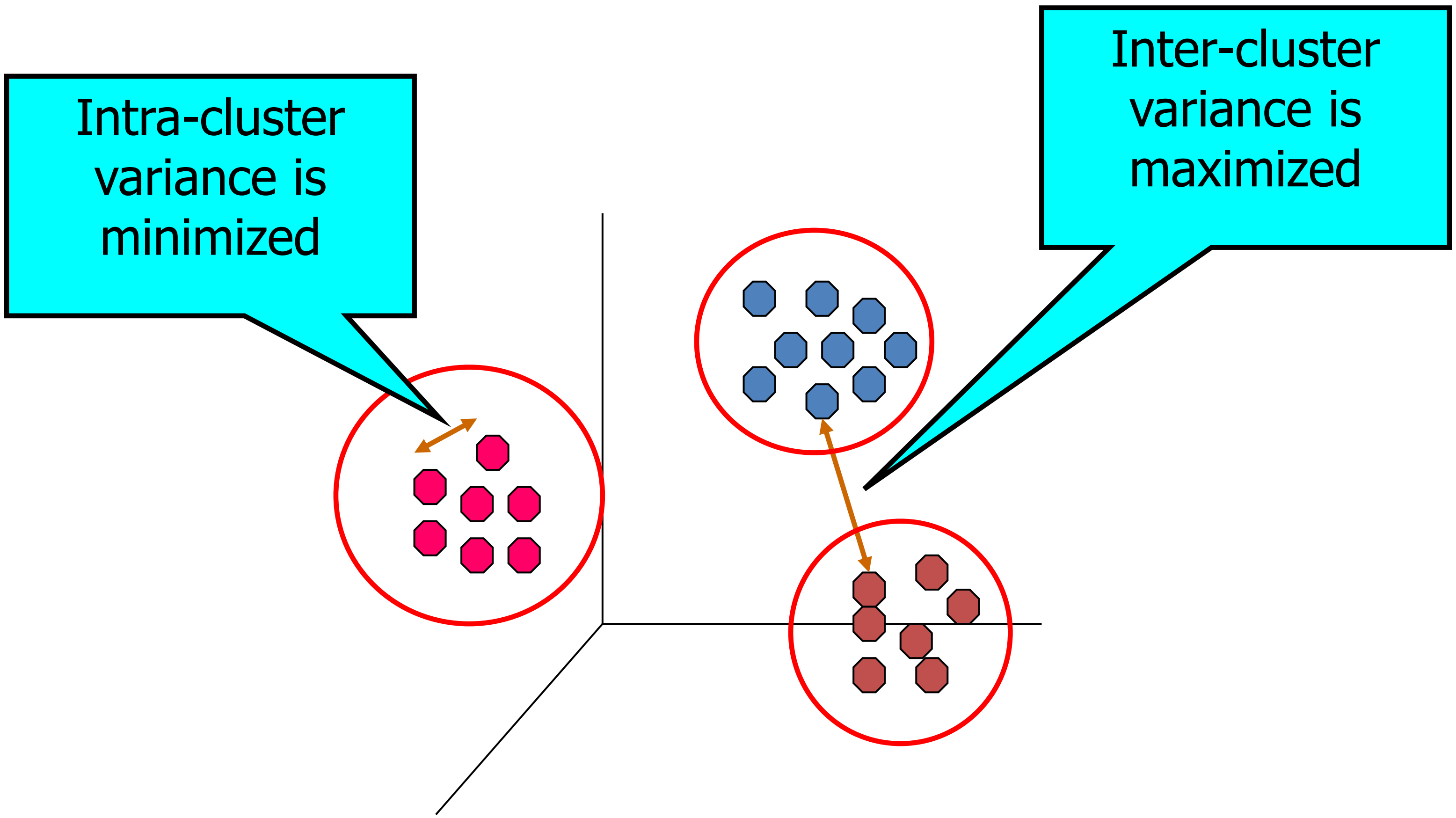






From MSE to cluster validity

- Minimize within cluster variance (MSE)
- Maximize between cluster variance

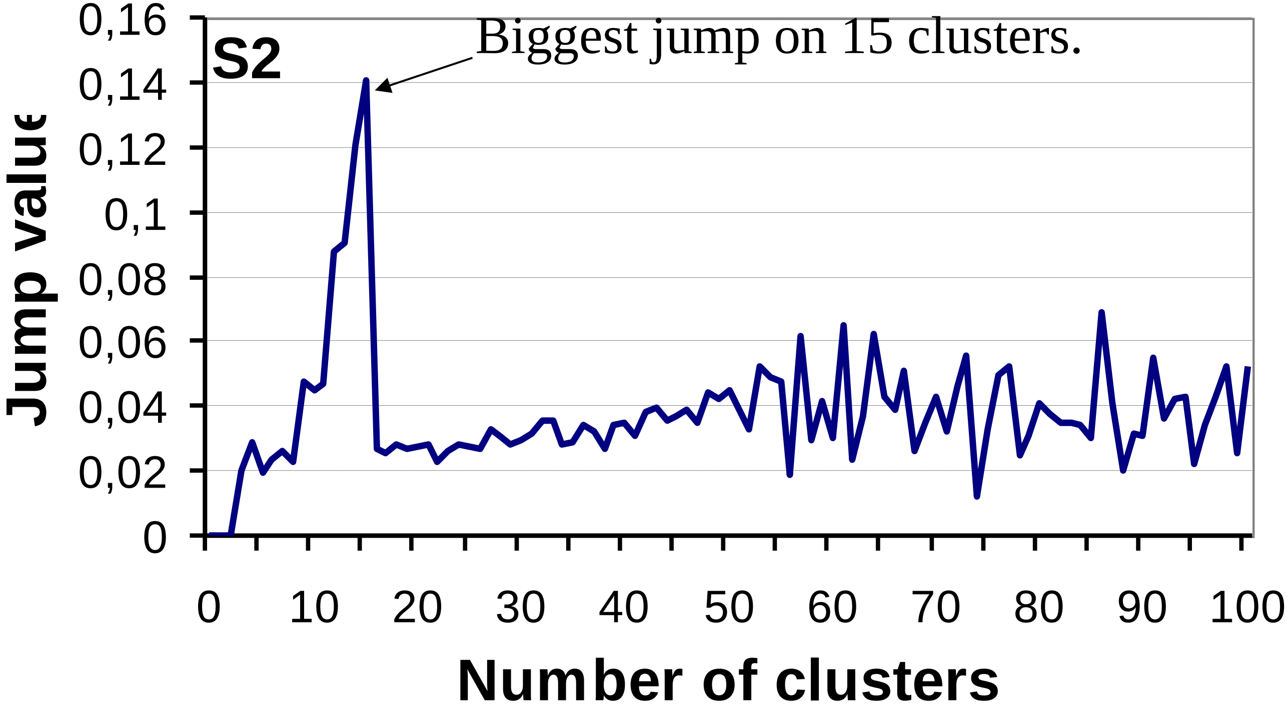


Jump point of MSE

(rate-distortion approach)

First derivative of powered MSE values:

$$J(k) = MSE(k)^{-d/2} - MSE(k-1)^{-d/2}$$



Variances

Within cluster: $$SSW(C,k)=\sum_{i=1}^{N}\| x_i - c_{p(i)} \|^2$$

Between clusters: $$SSB(C,k)=\sum_{j=1}^{k} n_j \| c_j - \overline{x} \|^2$$

Total Variance of data set:

$$\sigma(X)=\underbrace{\sum_{i=1}^{N}\| x_i - c_{p(i)} \|^2}_{\text{SSW}}+\underbrace{\sum_{j=1}^{k} n_j \| c_j - \overline{x} \|^2}_{\text{SSB}}$$

F-ratio variance test

- Variance-ratio F-test
- Measures ratio of between-groups variance against the within-groups variance (original f-test)
- F-ratio (WB-index):

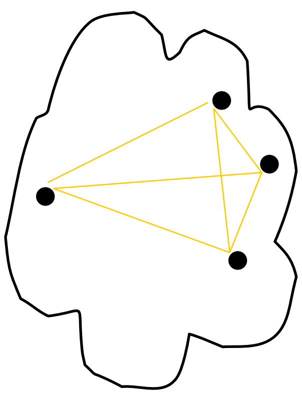
$$F = \frac{k \cdot \sum_{i=1}^{N} \| x_i - c_{p(i)} \|^2}{\sum_{j=1}^{k} n_j \| c_j - \overline{x} \|^2} = \frac{k \cdot SSW}{\sigma(X) - SSW}$$

SSB

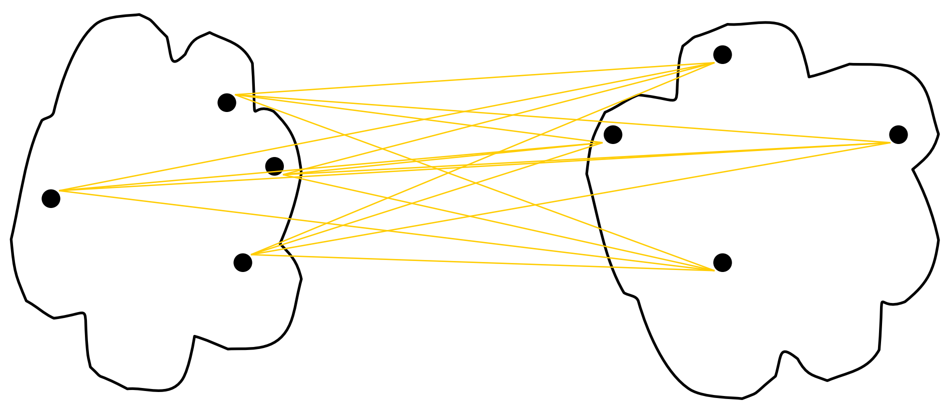
Silhouette coefficient

[Kaufman&Rousseeuw, 1990]

- Cohesion: measures how closely related are objects in a cluster
- Separation: measure how distinct or well-separated a cluster is from other clusters



cohesion



separation

Silhouette coefficient

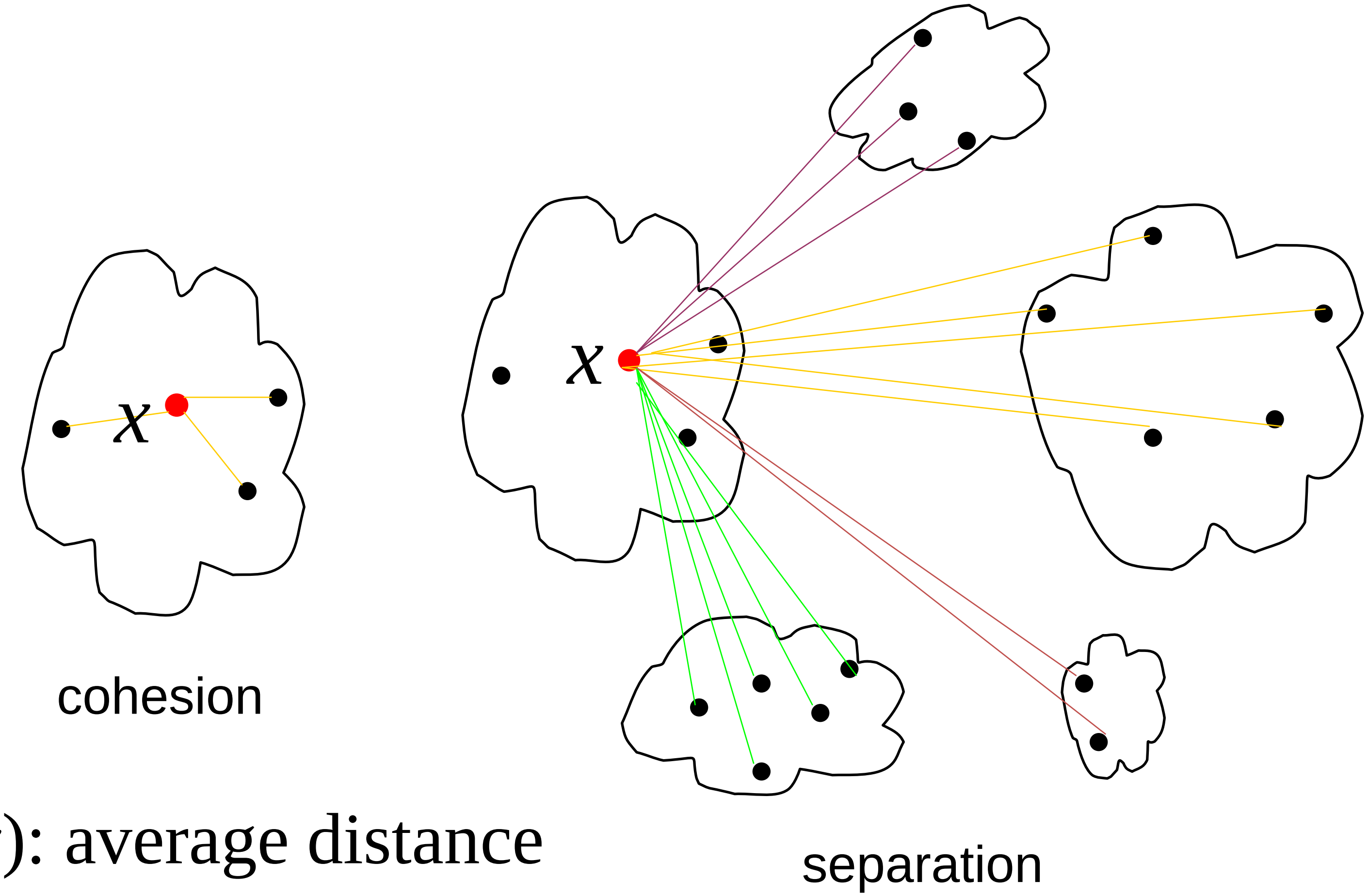
- *Cohesion* $a(x)$: average distance of x to all other vectors in the same cluster.
- *Separation* $b(x)$: average distance of x to the vectors in other clusters. Find the minimum among the clusters.
- *silhouette* $s(x)$:

$$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$$

- $s(x)$ = [-1, +1]: -1=bad, 0=indifferent, 1=good
- Silhouette coefficient (SC):

$$SC = \frac{1}{N}\sum_{i=1}^{N} s(x)$$

Silhouette coefficient



a(*x*): average distance in the cluster

b(x): average distances to others clusters, find minimal

External indexes

- Pair counting
- Information theoretic
- Set matching

External indexes

If true class labels (*ground truth*) are known, the validity of a clustering can be verified by comparing the class labels and clustering labels.

$$
\begin{array}{c|c} \boxed{N} & \cdot \\ \hline \cdot & n_{\cdot\cdot} \end{array}
=
\begin{array}{|cccc|c} \hline n_{11} & n_{12} & \cdots & n_{1l} & n_{1\cdot} \\ n_{21} & n_{22} & \cdots & n_{2l} & n_{2\cdot} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ n_{k1} & n_{k2} & \cdots & n_{kl} & n_{k\cdot} \\ \hline n_{\cdot 1} & n_{\cdot 2} & \cdots & n_{\cdot l} & n_{\cdot\cdot} \end{array}
$$

n_{ij} = number of objects in class i and cluster j

Pair-counting measures

Measure the number of pairs that are in:

Same class **both** in *P* and *G*.

$$a = \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}(n_{ij}-1)$$

Same class in *P* but different in *G*.

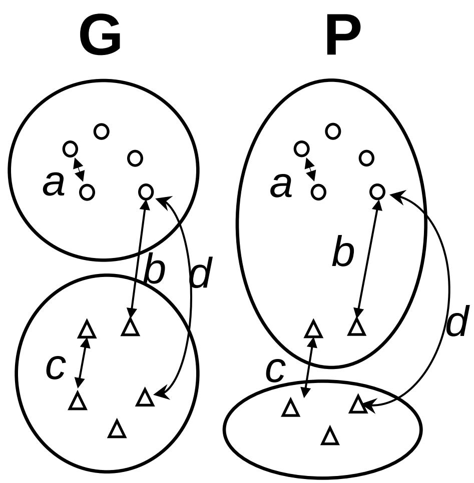
$$b = \frac{1}{2}\left(\sum_{j=1}^{K'} n_{.j}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\right)$$

Different classes in *P* but same in *G*.

$$c = \frac{1}{2}\left(\sum_{i=1}^{K} n_{i.}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\right)$$

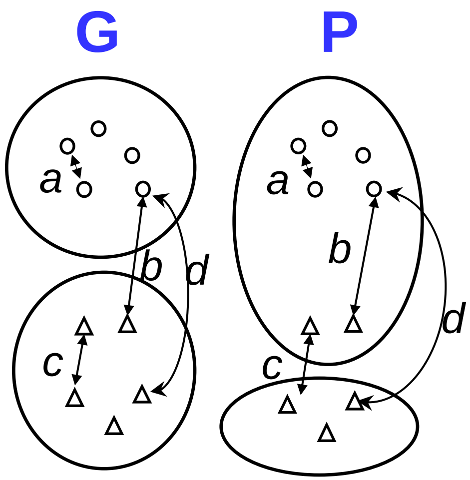
Different classes **both** in *P* and *G*.

$$d = \frac{1}{2}\left(N^2 + \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2 - \left(\sum_{i=1}^{K} n_{i.}^2 + \sum_{j=1}^{K'} n_{.j}^2\right)\right)$$



Rand and Adjusted Rand index

[Rand, 1971] [Hubert and Arabie, 1985]



Agreement: *a*, *d*
Disagreement: *b*, *c*

$$RI(P,G)=\frac{a+d}{a+b+c+d}$$

$$ARI=\frac{RI-E(RI)}{1-E(RI)}$$

Rand index
(example)

Vectors assigned to:	Same cluster	Different clusters
Same cluster in ground truth	20	24
Different clusters in ground truth	20	72

Rand index = (20+72) / (20+24+20+72) = 92/136 = **0.68**

Adjusted Rand = (to be calculated) = **0.xx**

Normalized Mutual information

$$MI(k,G) = \sum_{l=1}^{k} \sum_{j=1}^{G} p(l,j) \log \frac{p(l,j)}{p(l)p(j)}$$

However, it does not take into account our intuitive preference for few clusters, so we normalized it:

$$NMI(k,G) = \frac{2MI}{H(k) + H(G)}$$

$$H(k) = \sum_{l=1}^{k} p(l) \log[p(l)]$$