Density Based Clustering algorithms and validation

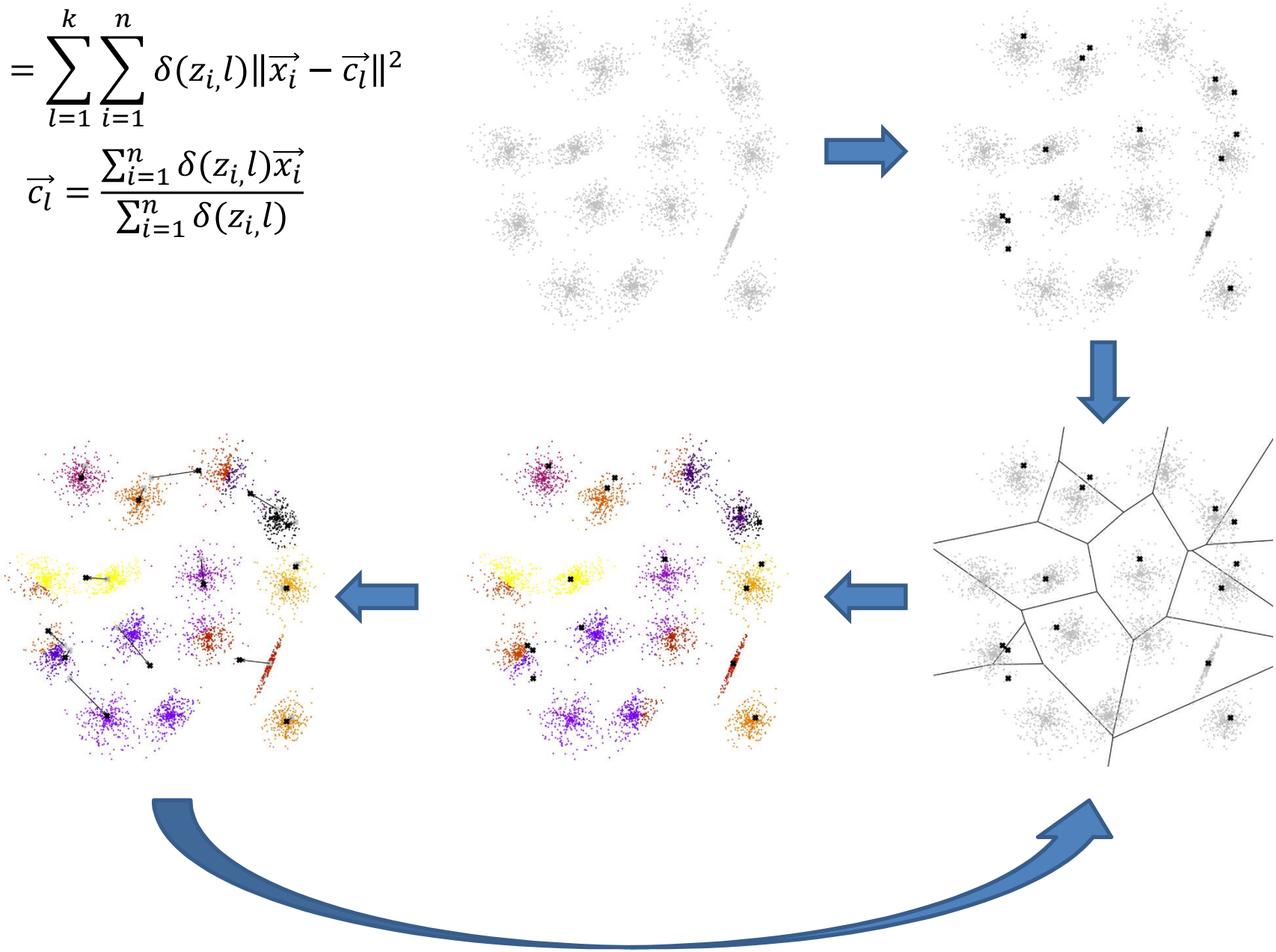
Unsupervised Machine Learning

Clustering: An overview

- K-means clustering:
  Hard partition.

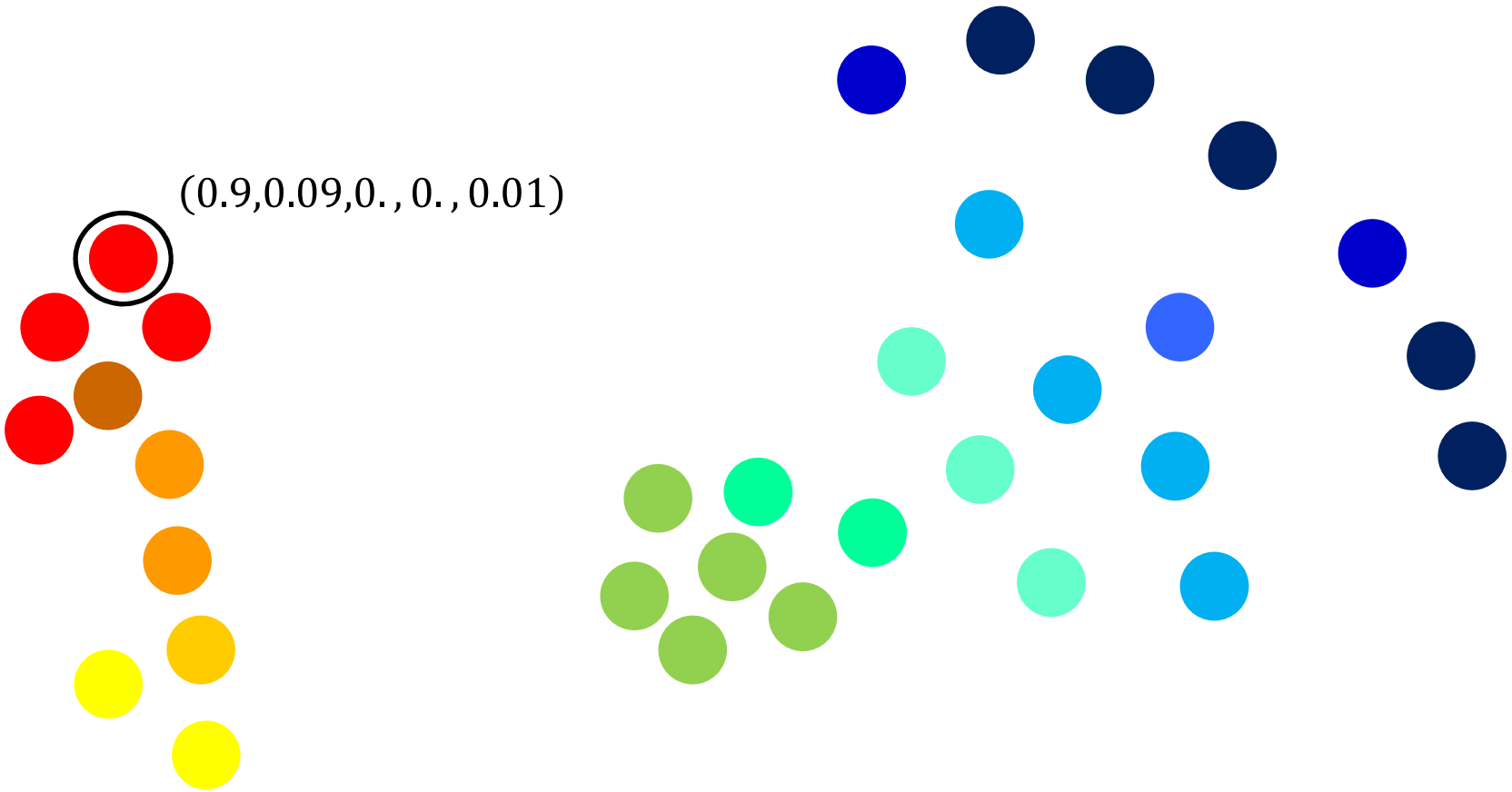
$$O(z) = \sum_{l=1}^{k} \sum_{i=1}^{n} \delta(z_{i,} l) \|\overrightarrow{x_i} - \overrightarrow{c_l}\|^2$$

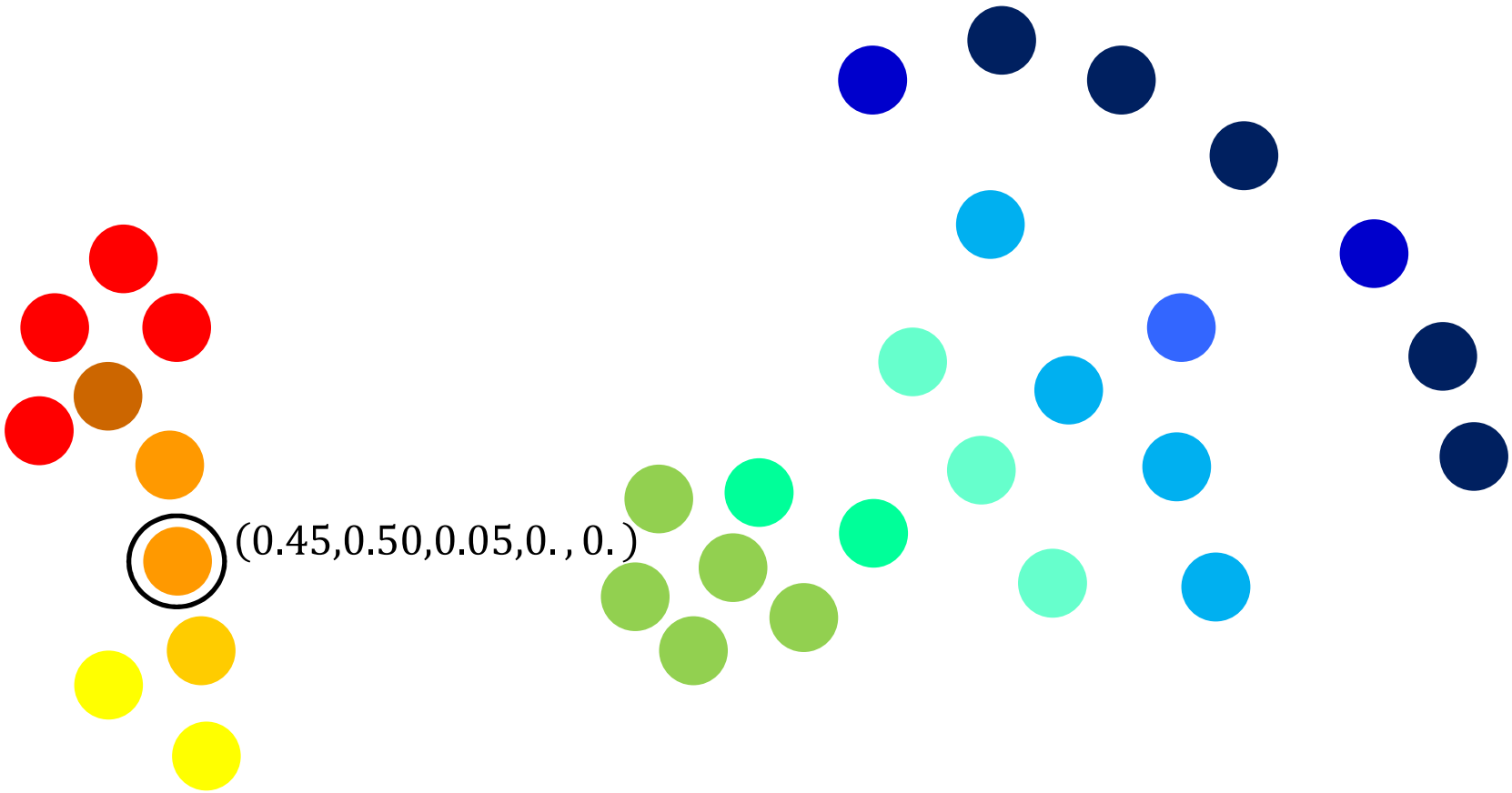
$$\overrightarrow{c_l} = \frac{\sum_{i=1}^{n} \delta(z_{i,} l) \overrightarrow{x_i}}{\sum_{i=1}^{n} \delta(z_{i,} l)}$$

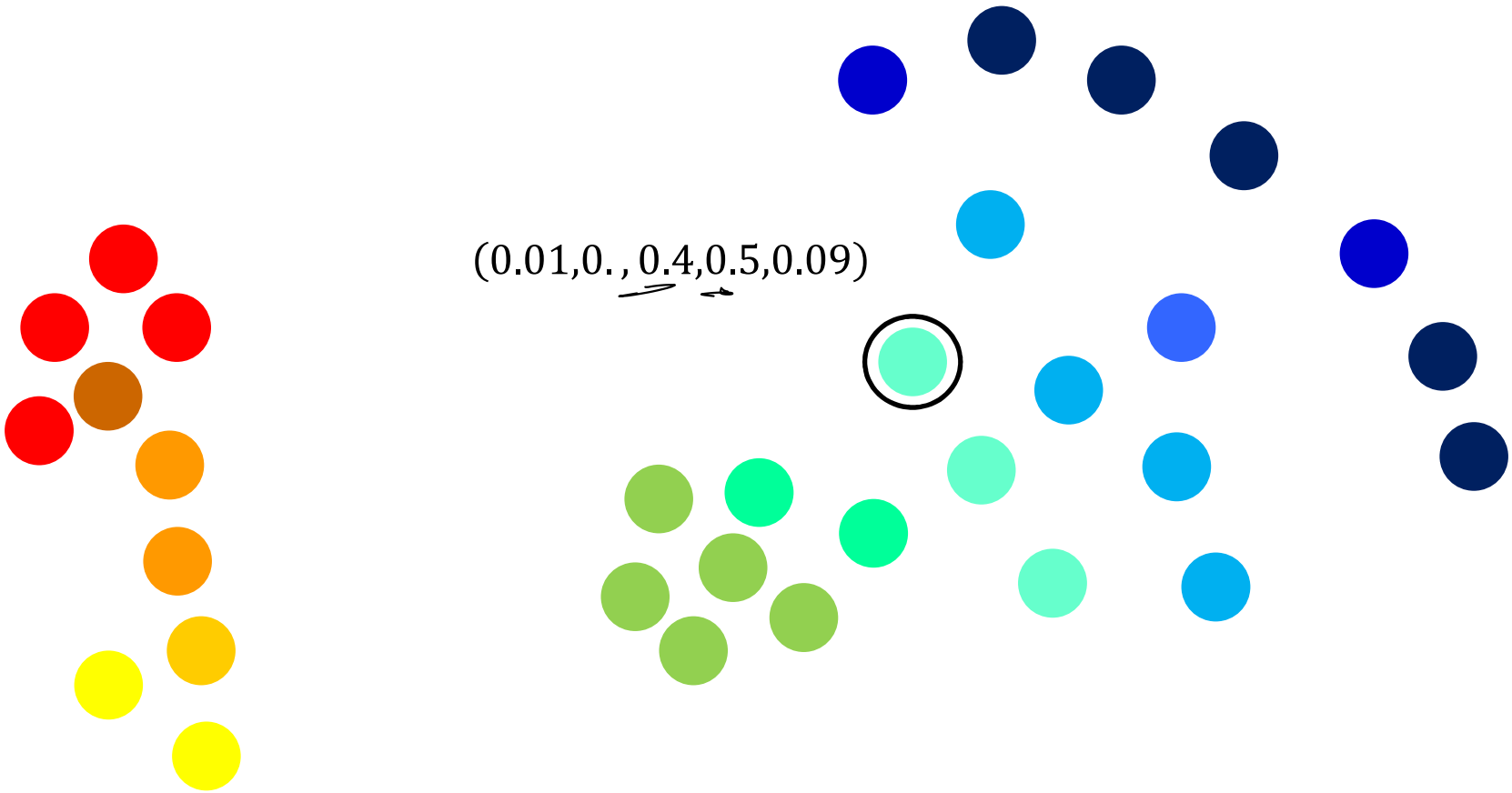


Clustering: An overview

- K-means clustering:
  Hard partition.
- c-means clustering:
  Fuzzy partition.

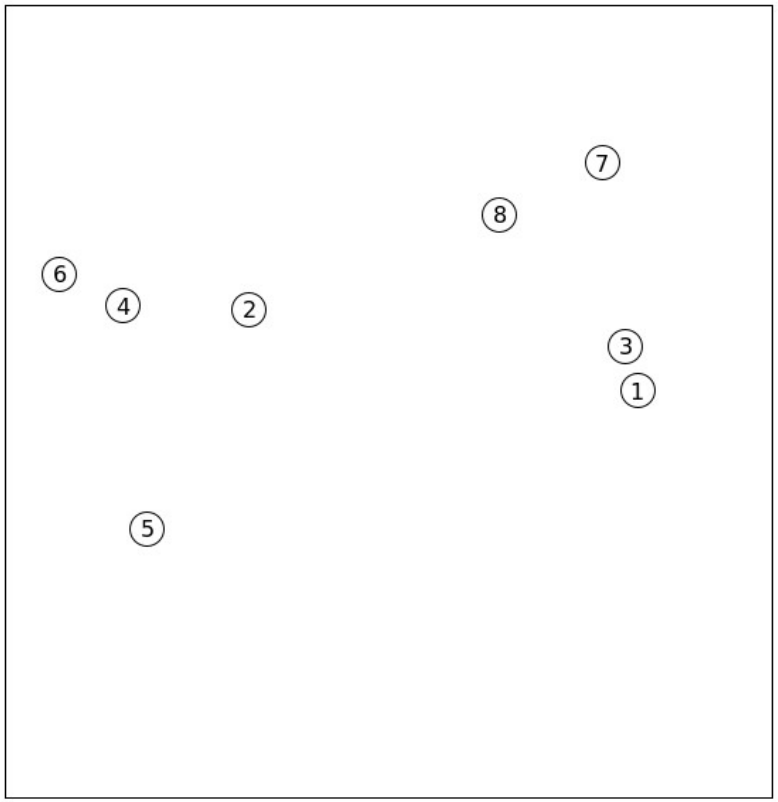

(0.9,0.09,0.,0.,0.01)

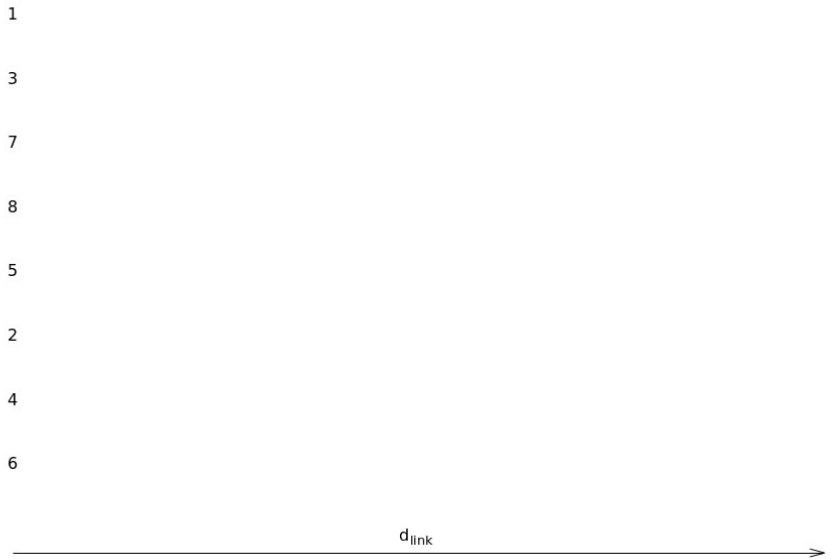

(0.45,0.50,0.05,0.,0.)

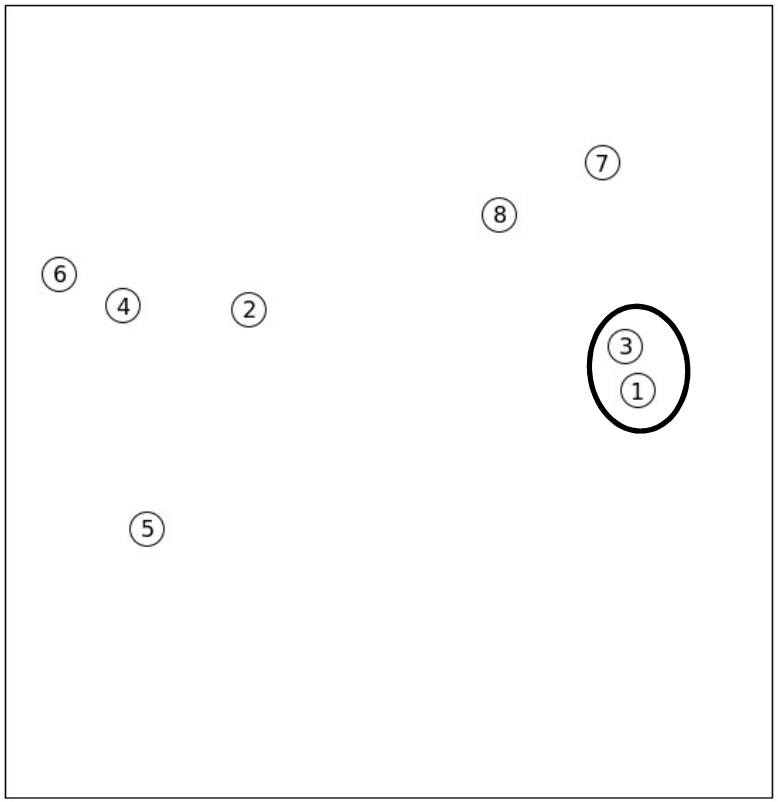

(0.01,0.,0.4,0.5,0.09)

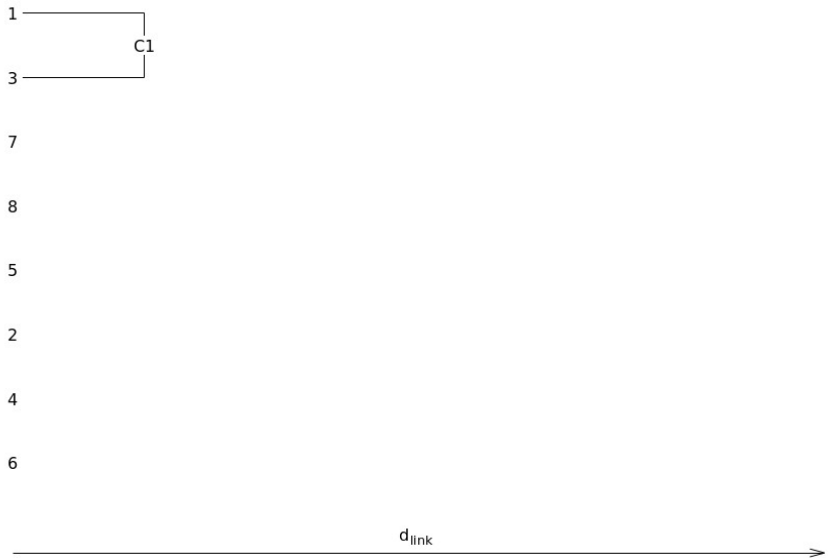
Clustering: An overview

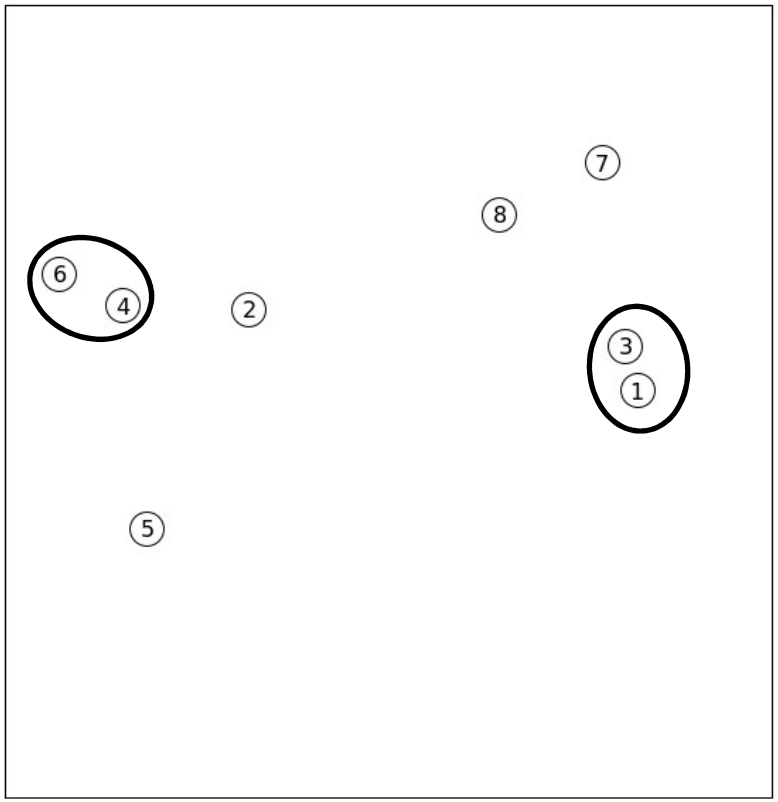
- K-means clustering:
  Hard partition.
- c-means clustering:
  Fuzzy partition.
- Hierarchical clustering:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Ward's method

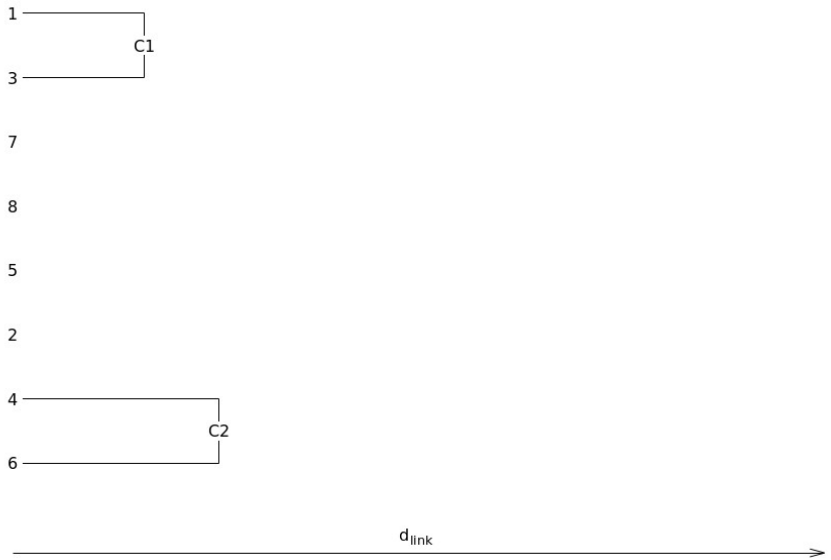

7
8
6
4
2
3
1
5

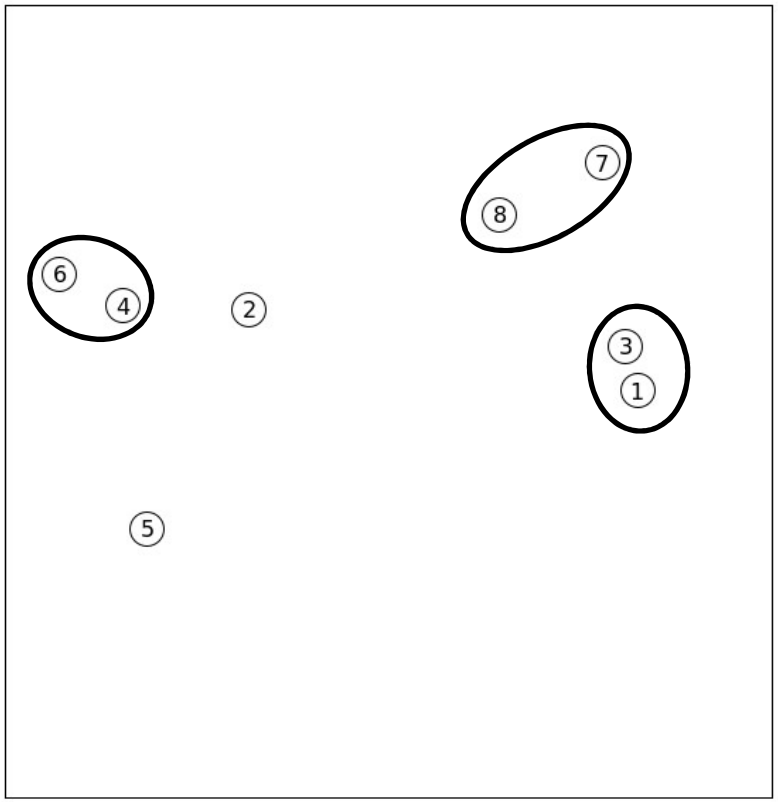

1
3
7
8
5
2
4
6
d_{link}

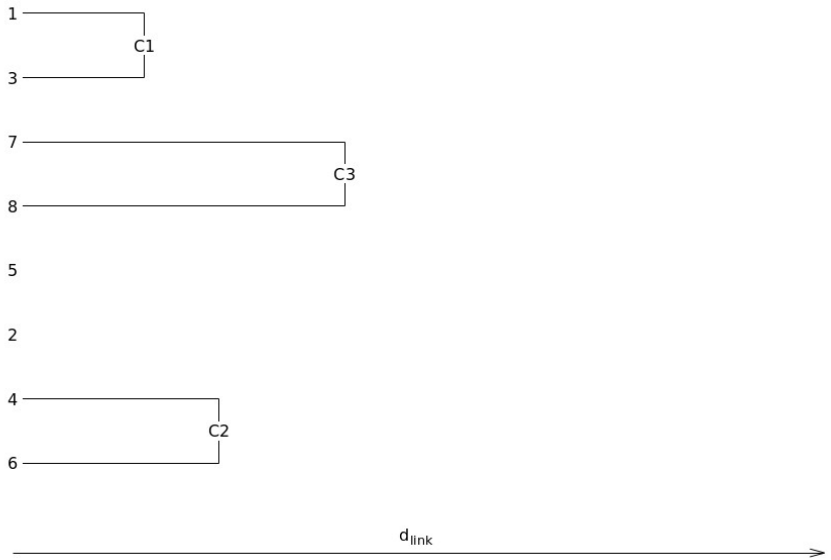

7
8
6
4
2
3
1
5

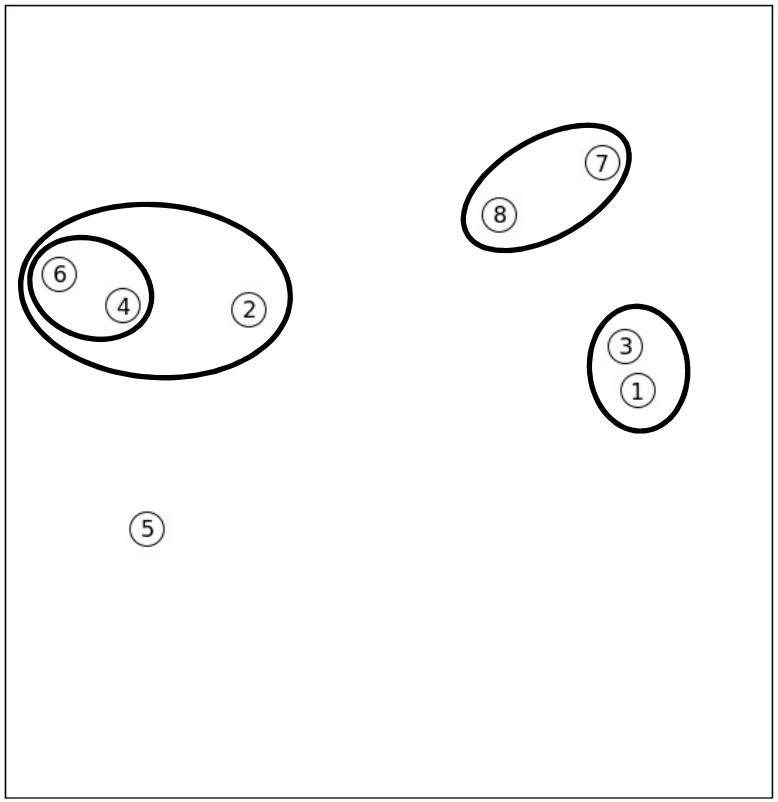

1
C1
3
7
8
5
2
4
6
d_{link}

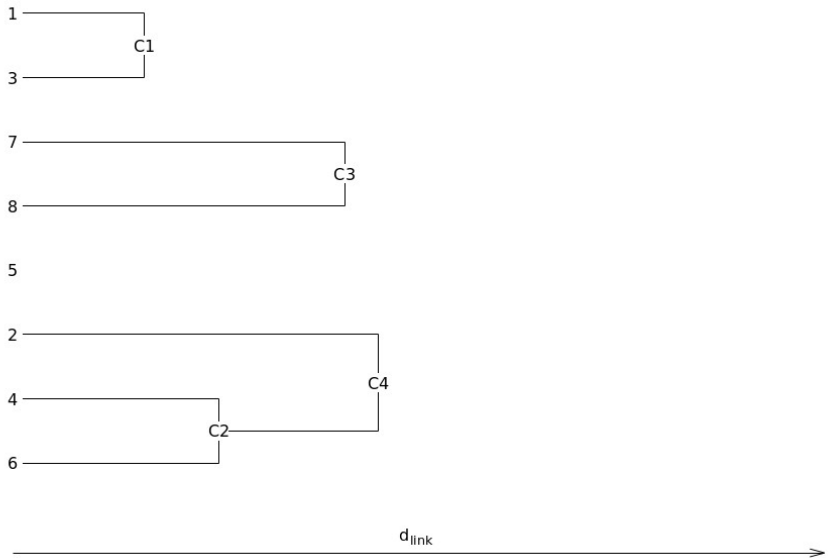

7
8
6
4
2
3
1
5

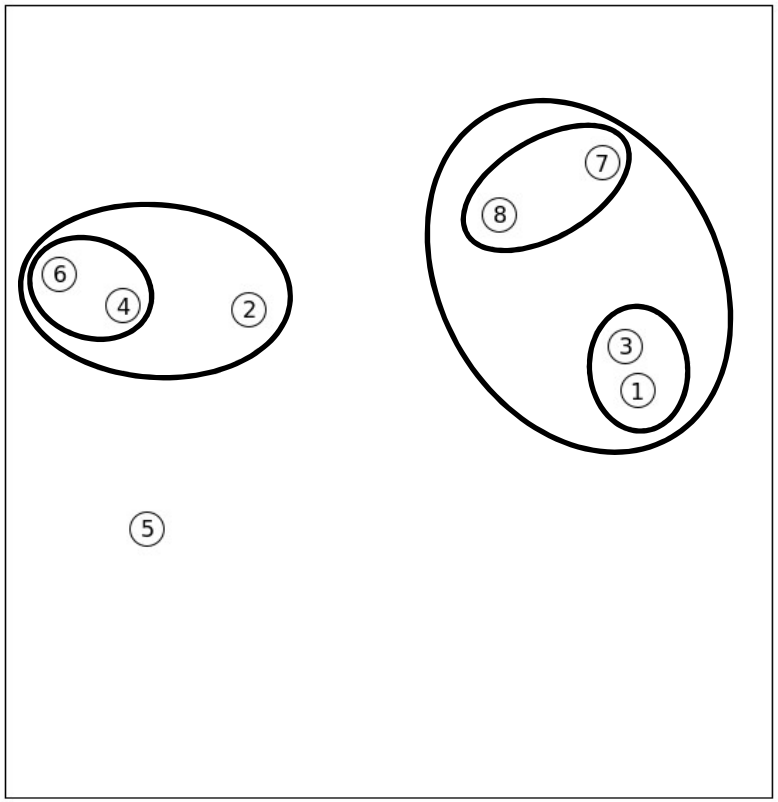

1
C1
3
7
8
5
2
4
C2
6
d_{link}

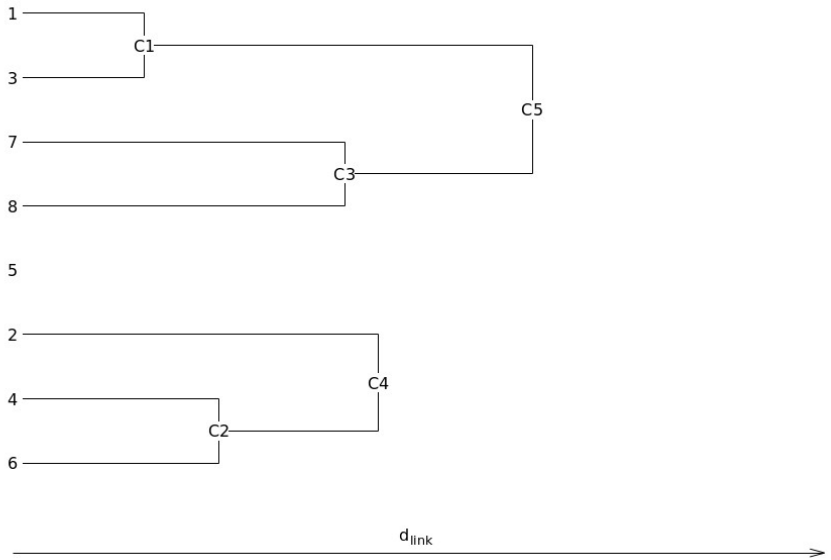

7
8
6
4
2
3
1
5

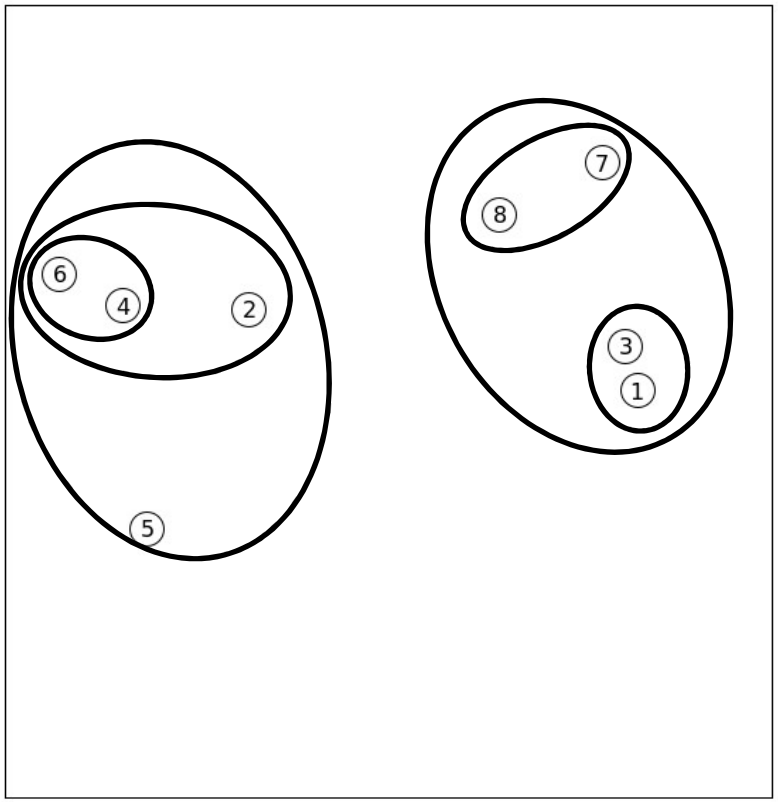

1
C1
3
7
C3
8
5
2
4
C2
6
d_{link}

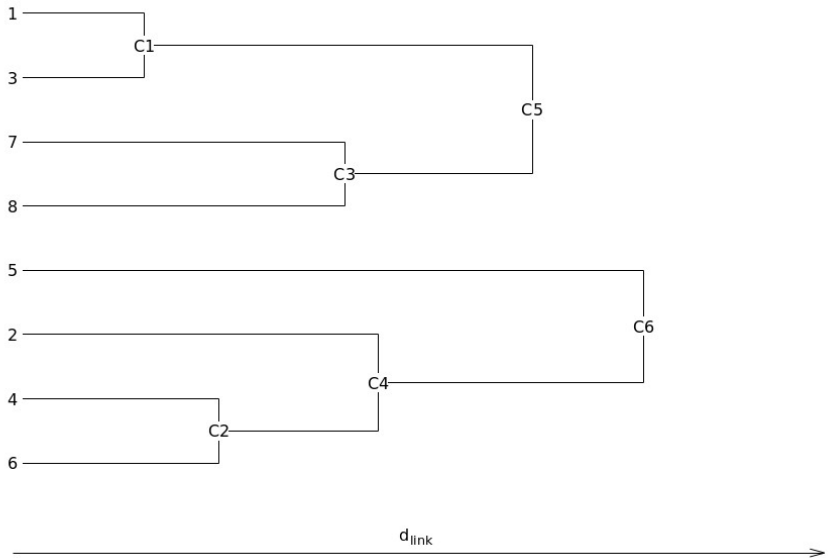

7
8
6
4
2
3
1
5

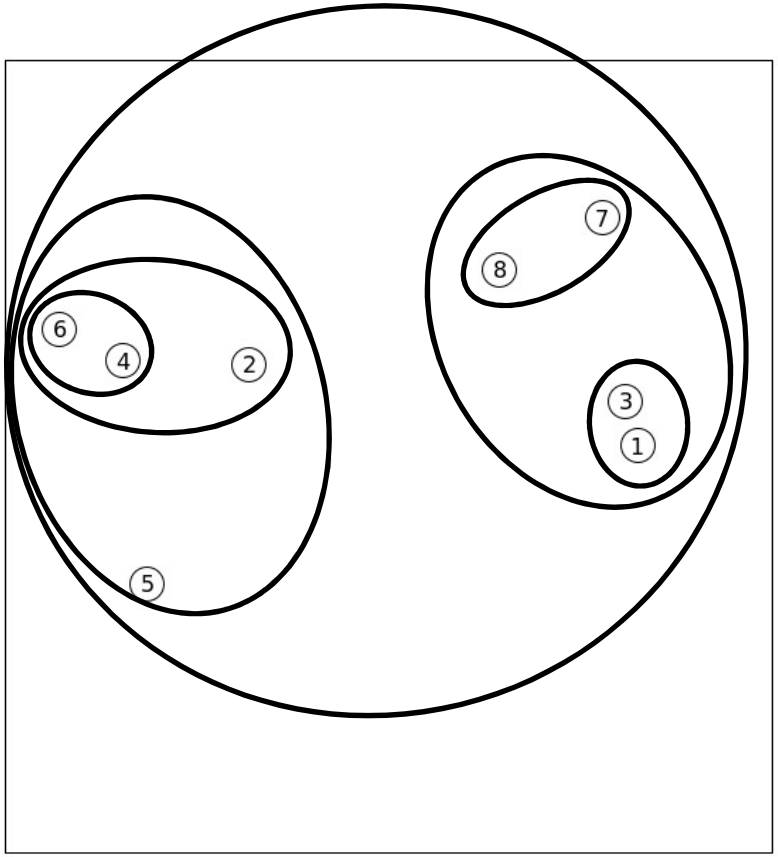

1
C1
3
7
C3
8
5
2
C4
4
C2
6
d_{link}

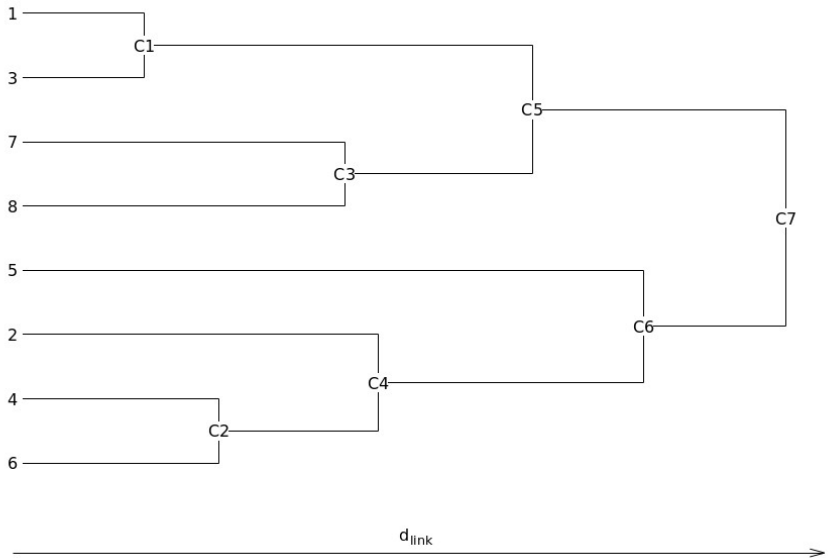

6
4
2
7
8
3
1
5


1
C1
3
C5
7
C3
8
5
2
C4
4
C2
6
d_{link}


6
4
2
5
7
8
3
1


1
C1
3
C5
7
C3
8
5
C6
2
C4
4
C2
6
d_{link}


6
4
2
7
8
3
1
5


1
C1
3
C5
7
C3
8
C7
5
C6
2
C4
4
C2
6
d_{link}

Clustering: An overview

- K-means clustering: Hard partition.
- c-means clustering: Fuzzy partition.
- Hierarchical clustering:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Ward's method

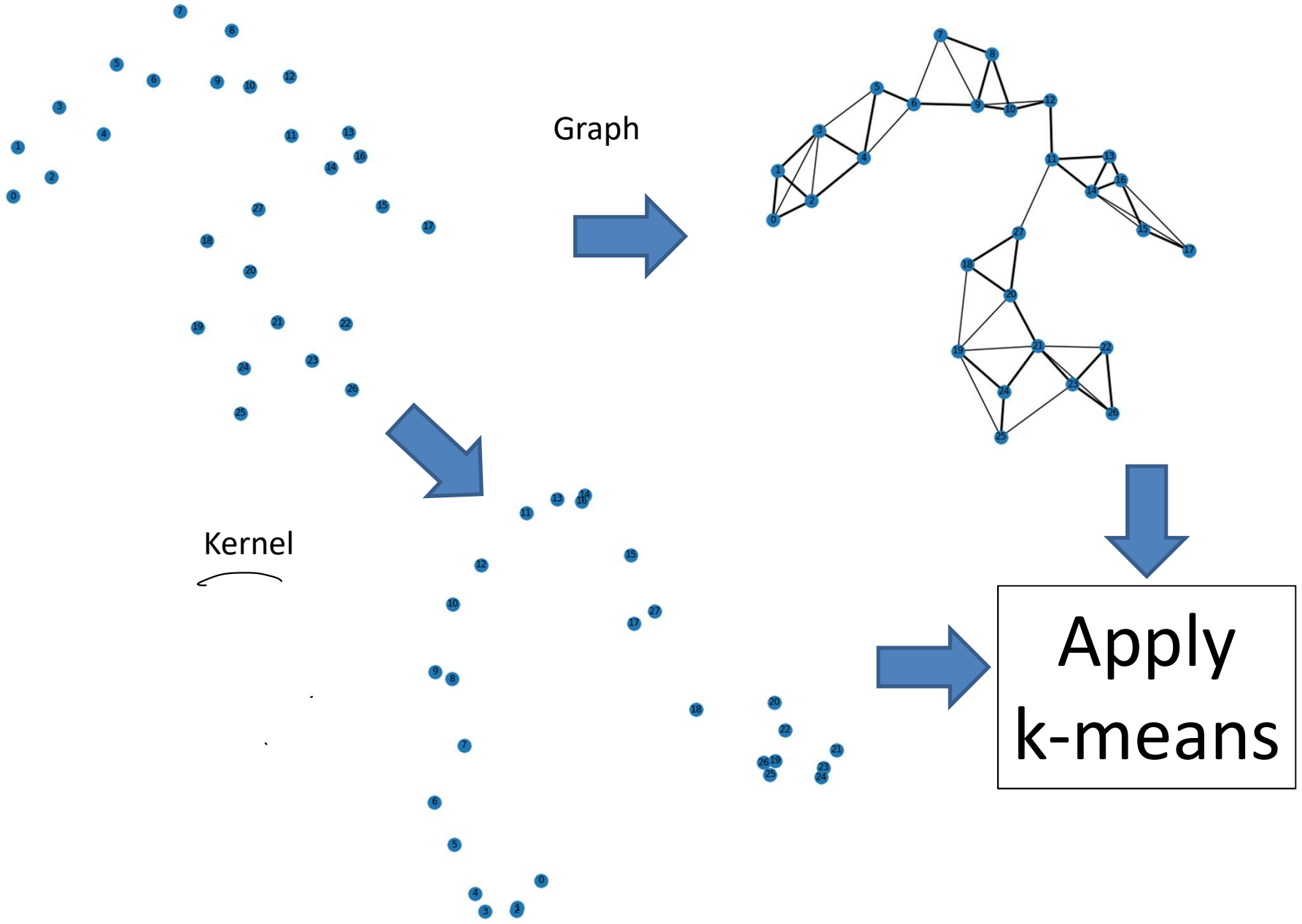
Classical methods

Clustering: An overview

- K-means clustering: Hard partition.
- c-means clustering: Fuzzy partition.
- Hierarchical clustering:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Ward's method

Classical methods

- Kernel k-means
- Spectral clustering

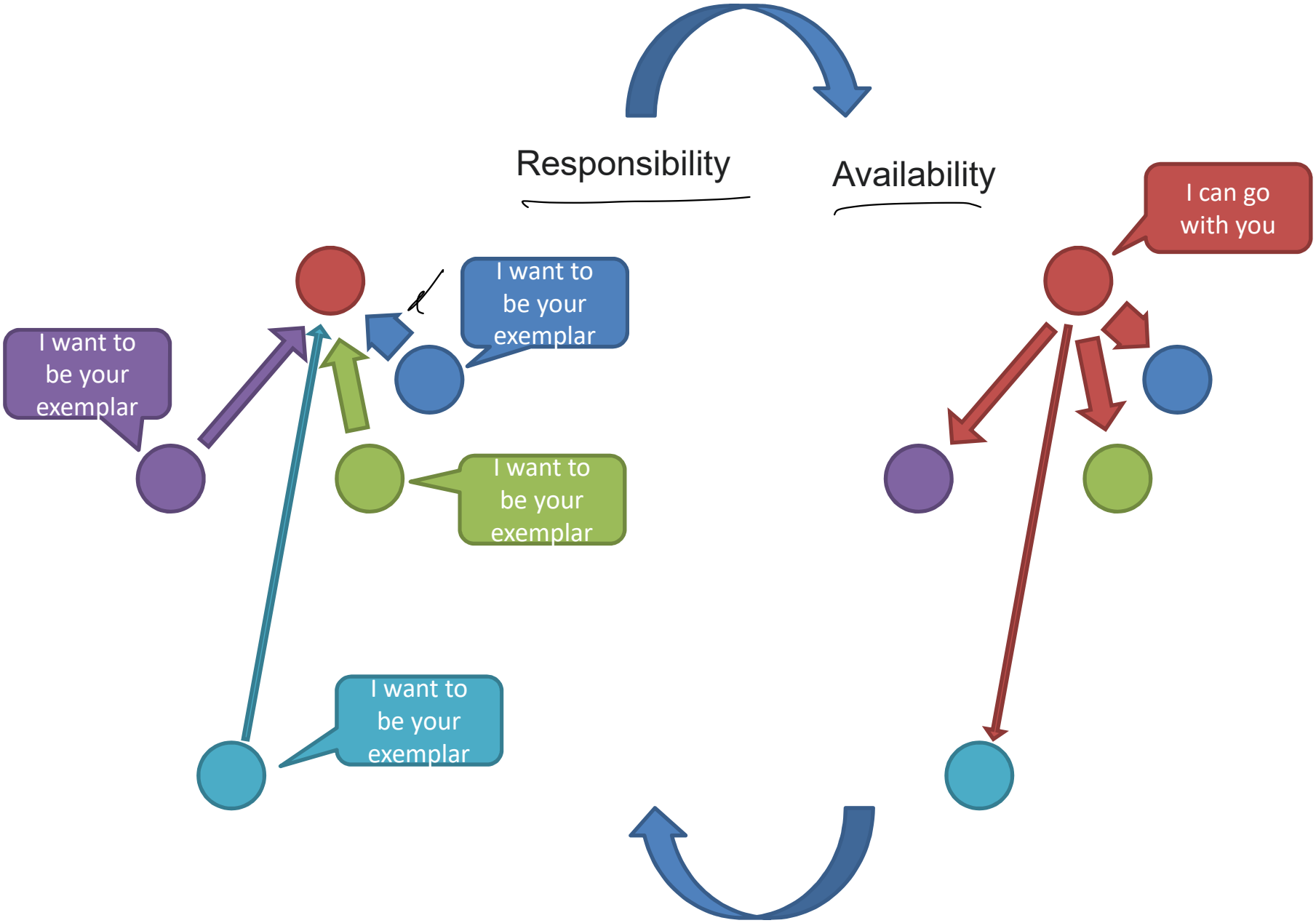

Graph
Kernel
Apply
k-means

Clustering: An overview

- K-means clustering: Hard partition.
- c-means clustering: Fuzzy partition.
- Hierarchical clustering:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Ward's method

Classical methods

- Kernel k-means
- Spectral clustering
- Affinity Propagation


Responsibility
Availability
I want to be your exemplar
I want to be your exemplar
I want to be your exemplar
I want to be your exemplar
I can go with you

Clustering: An overview

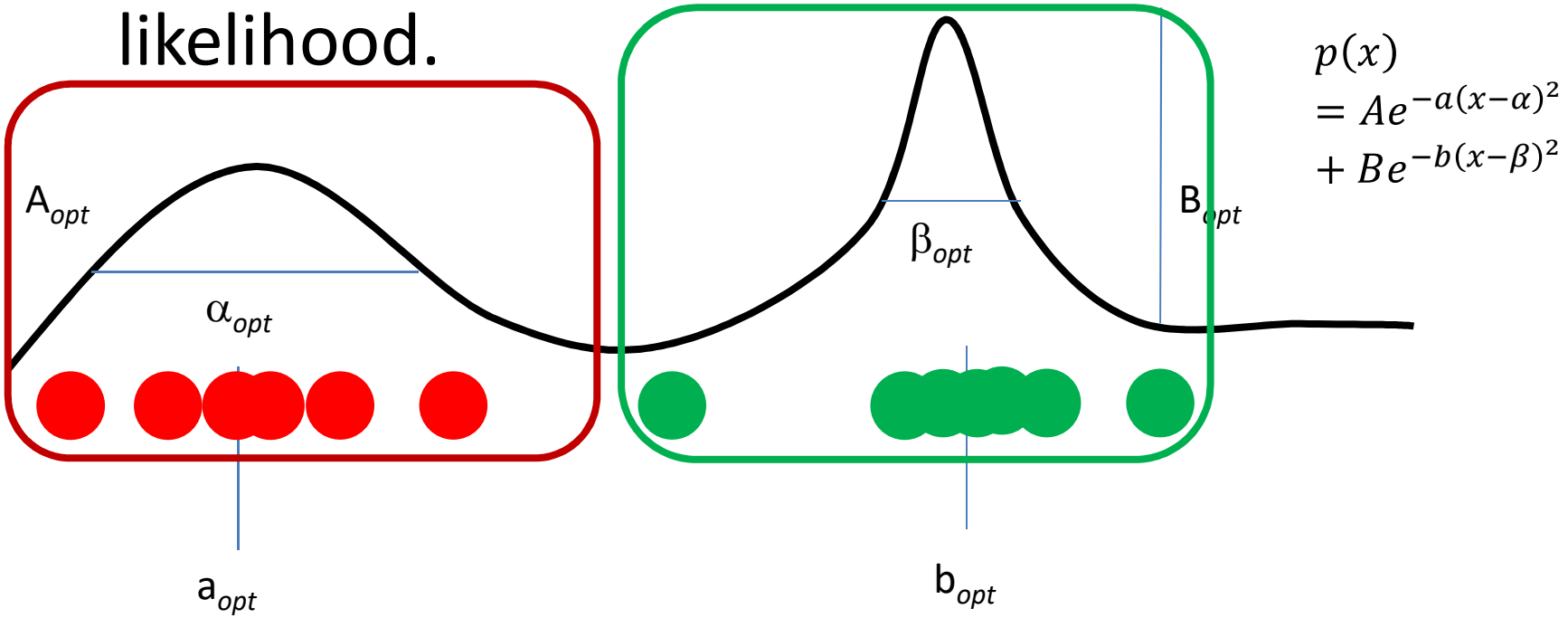
- K-means clustering: Hard partition.
- c-means clustering: Fuzzy partition.
- Hierarchical clustering:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Ward's method

Classical methods

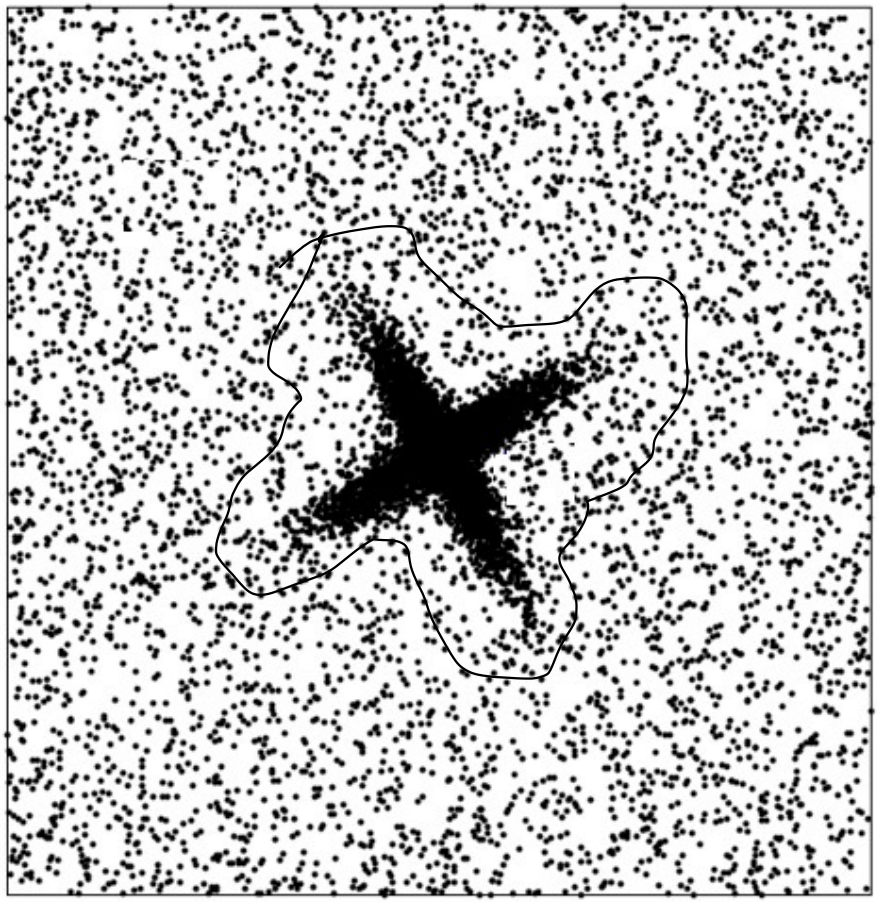
- Kernel k-means
- Spectral clustering
- Affinity Propagation
- Expectation-Maximization GMM

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.



What happens with generic shape clusters?



This distribution comes from the sum of two gaussian distributions, but… are there two clusters?

(NON-PARAMETRIC) DENSITY BASED CLUSTERING

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

Modern clustering algorithms (IV)

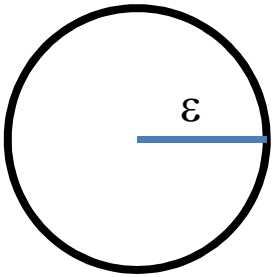
DBSCAN

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)

(ε)

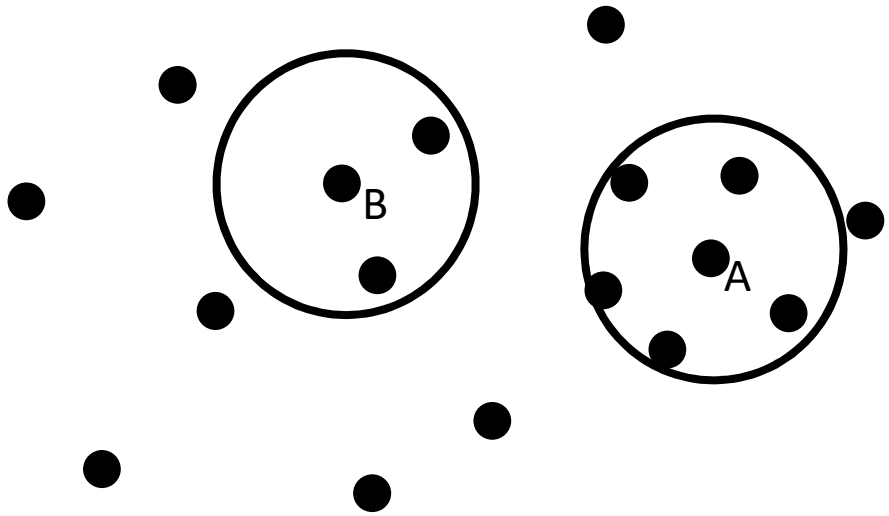
A route for this explanation



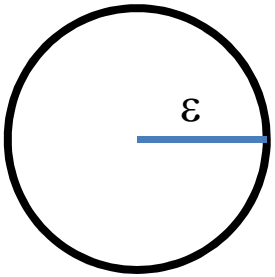
ε / Eps is a method parameter.

$\rho_A = 6$

$\rho_B = 3$



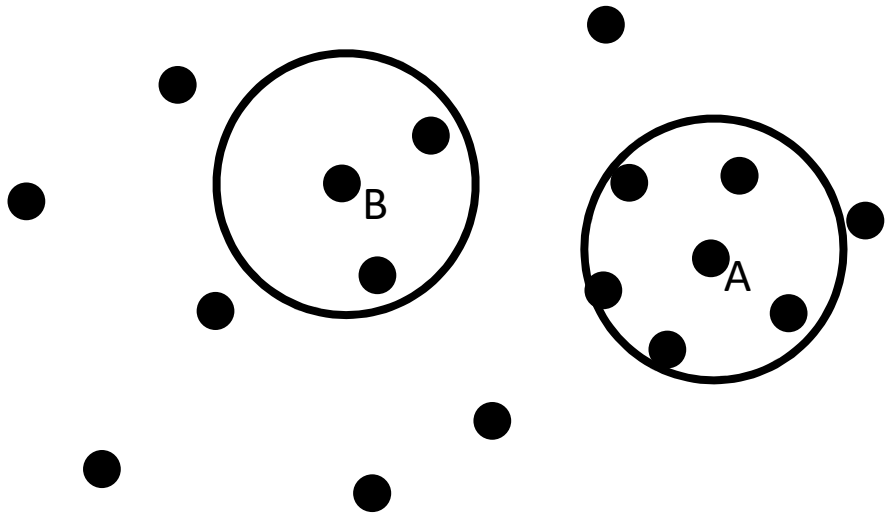
A route for this explanation



ε / Eps is a method parameter.

$\rho_A = 6$

$\rho_B = 3$



- We will classify the points according with their density.
- We will define a set of rules that exploit this classification in order to define the clusters.

DBSCAN: Point classes

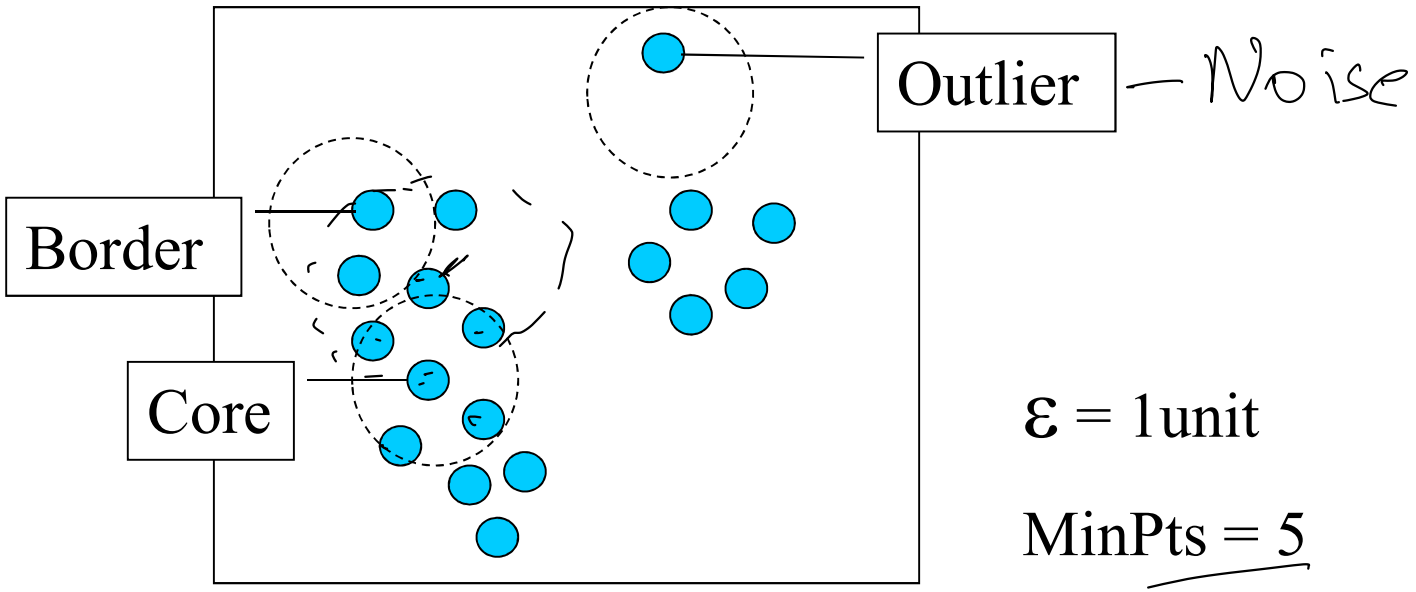
- A point is a **core point** if it has more than a specified number of points (MinPts) within Eps. These are points that are at the interior of a cluster
- A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point.

DBSCAN: Point classes

- 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
- A **noise point** is any point that is not a core point or a border point.

Note that we are adding a new method parameter: MinPts, ε

Border & Core

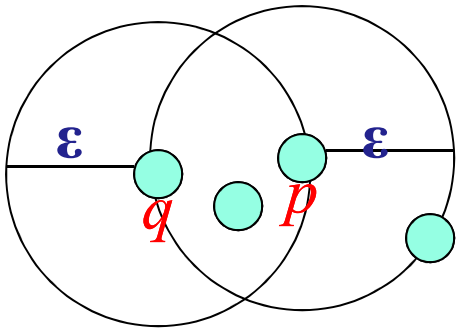


DBSCAN: Ideas behind the rules

- 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

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

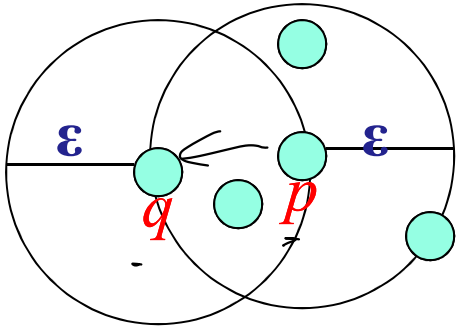
ε-Neighborhood of q

p is a core object (MinPts = 4)

q is not a core object

Concepts: Reachability

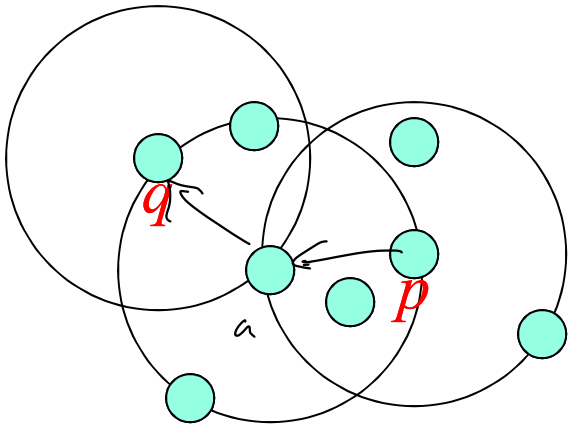
- **Directly density-reachable**
  - An object q is directly density-reachable from object p if q is within the ε-Neighborhood of p and p is a core object.



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

Concepts: Reachability

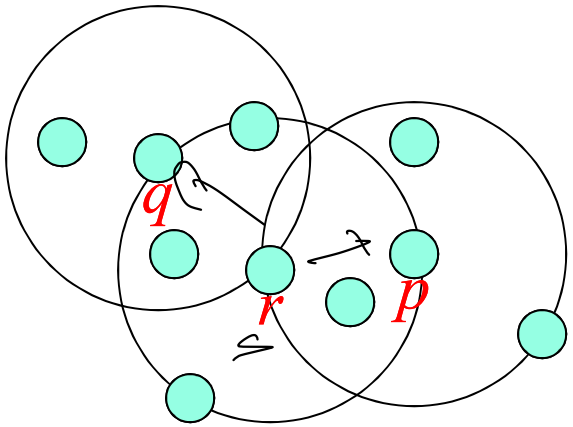
- **Density-reachable:**
  - An object *p* is density-reachable from *q* w.r.t ε and *MinPts* if there is a chain of objects $p_1,\ldots,p_n$, with $p_1=q$, $p_n=p$ such that $p_{i+1}$is directly density-reachable from p_i w.r.t ε and *MinPts* for all 1 <= *i* <= *n*



- *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*

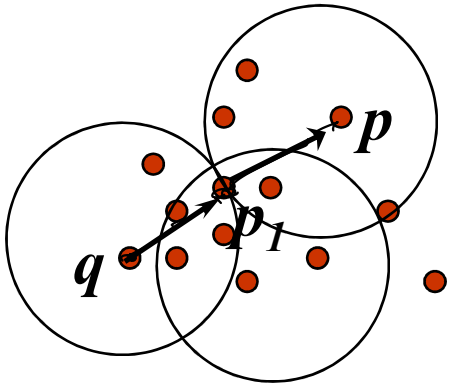


- *P* and *q* are density-connected to each other by *r*
- Density-connectivity is symmetric

Concepts: cluster & noise

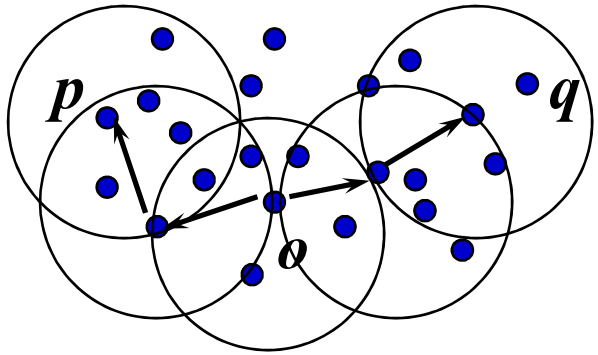
- **Cluster**: a cluster ***C*** in a set of objects ***D*** w.r.t ε and *MinPts* is a non empty subset of ***D*** satisfying
  - Maximality: For all *p*, *q* if $p \in \boldsymbol{C}$ and if *q* is density-reachable from *p* w.r.t ε and *MinPts*, then also $q \in \boldsymbol{C}$.
  - Connectivity: for all *p*, *q* $\in \boldsymbol{C}$, *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:



p is density reachable from q

Density-connected

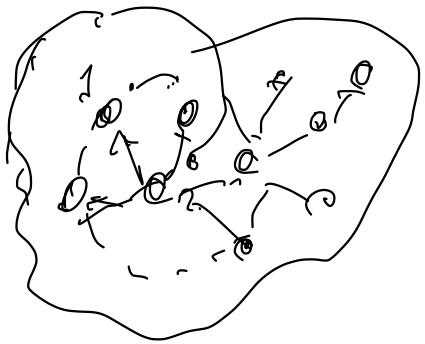


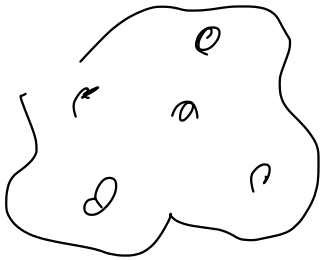
p is density reachable from o & q is density reachable from o then p & q Density connected

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

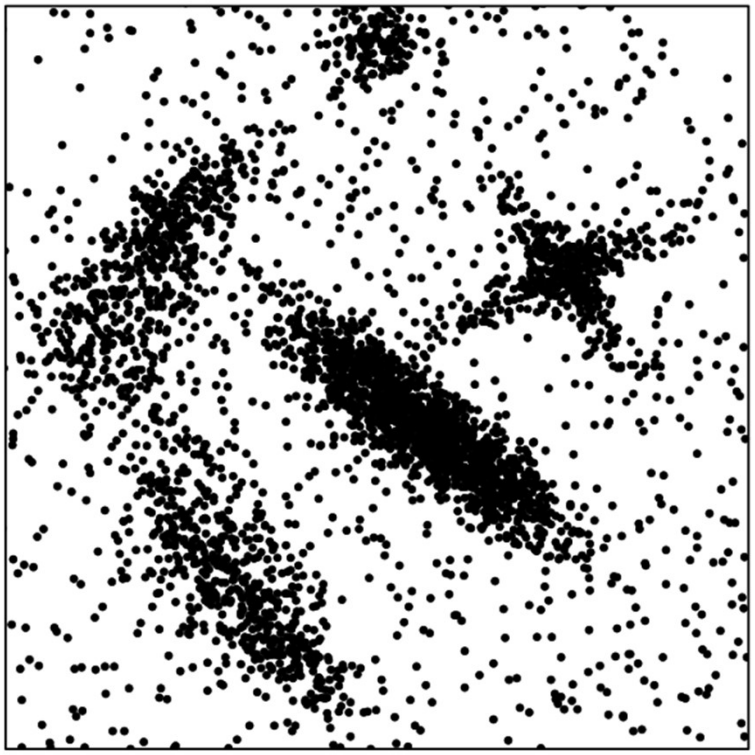




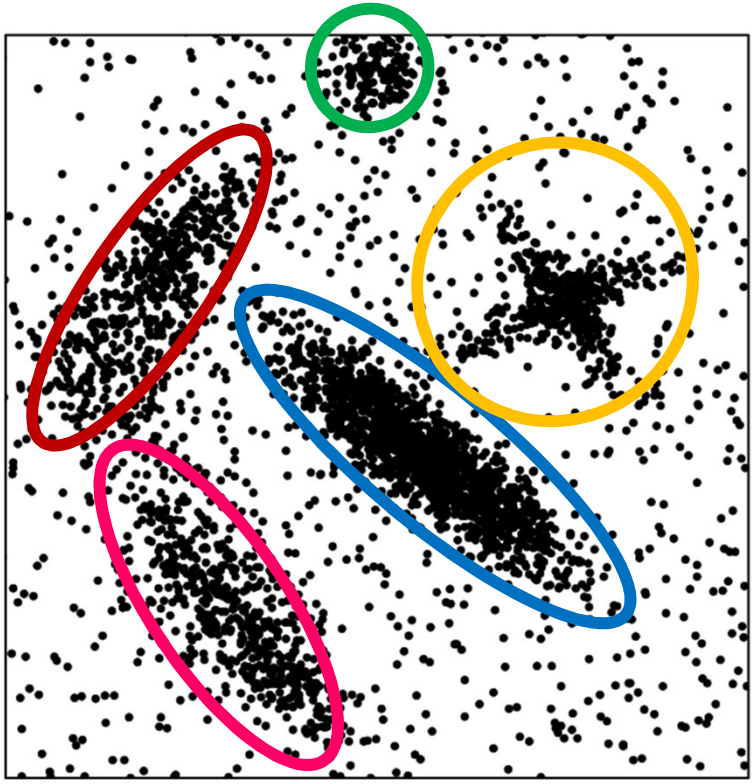
Fast Search and Find of Density Peaks

Modern clustering algorithms (V)

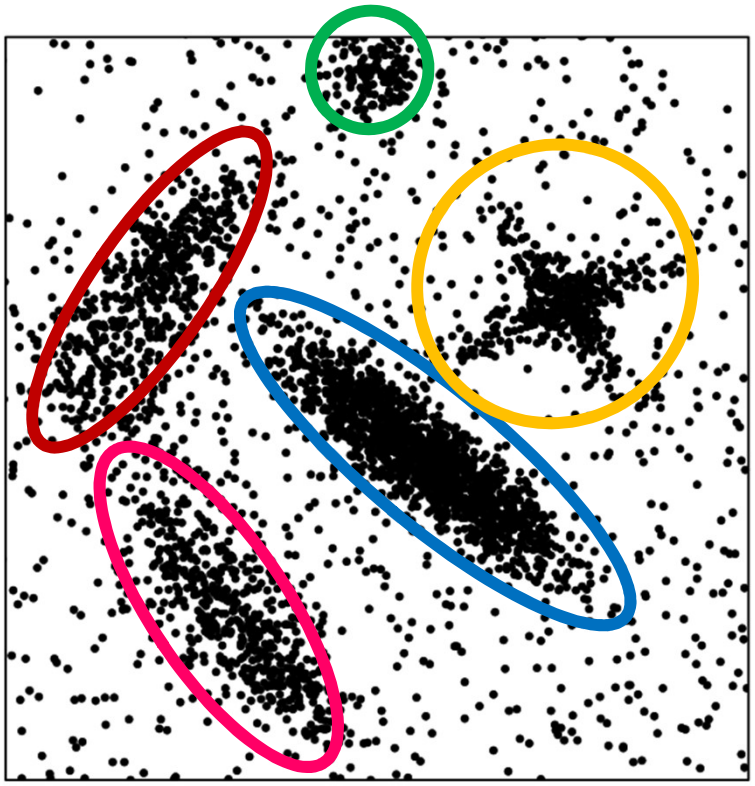
Density Peaks clustering



Density Peaks clustering



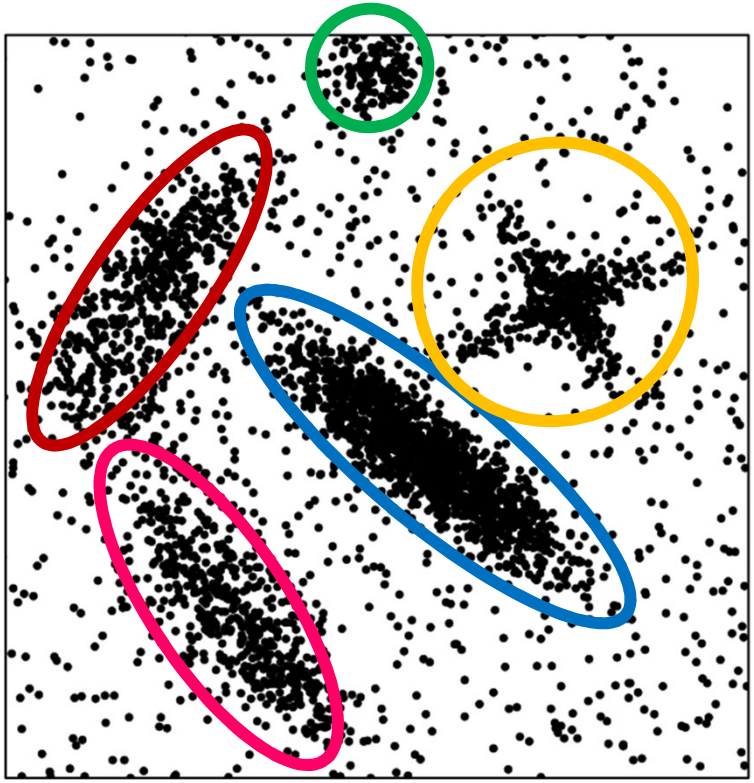
Density Peaks clustering

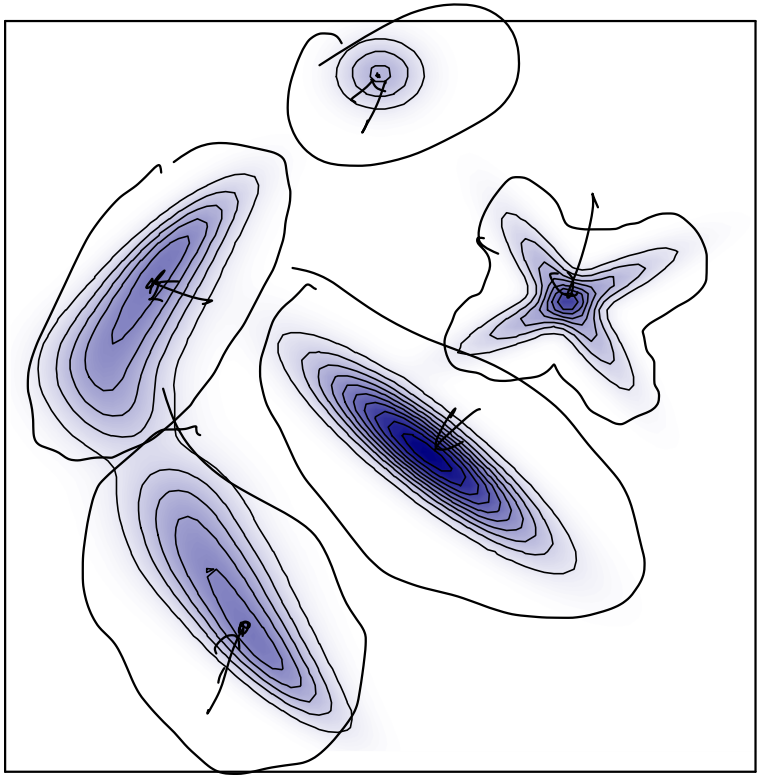


Clusters = peaks in the density of points* (new idea)

*Idea already present in mean-shift (this is seen later)

Density Peaks clustering





Clusters = peaks in the density of points*

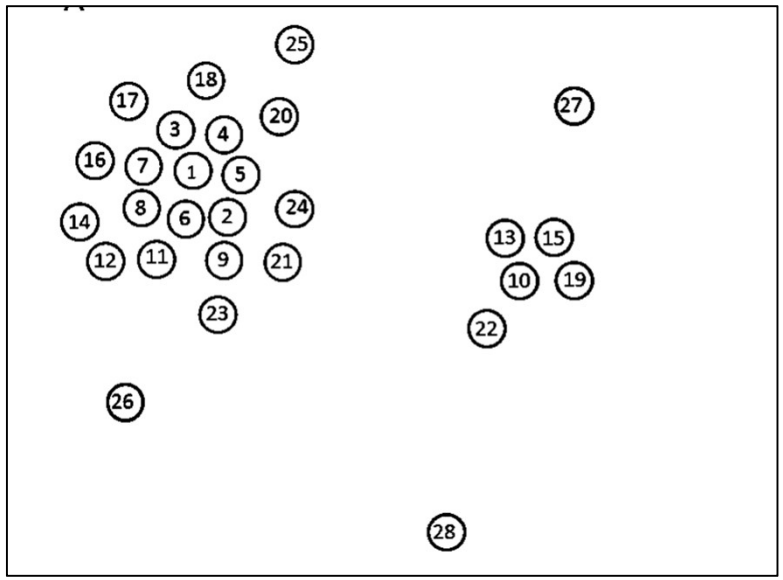
= peaks in the “mother” probability distribution

*Idea already present in mean-shift

Density Peaks: δ concept

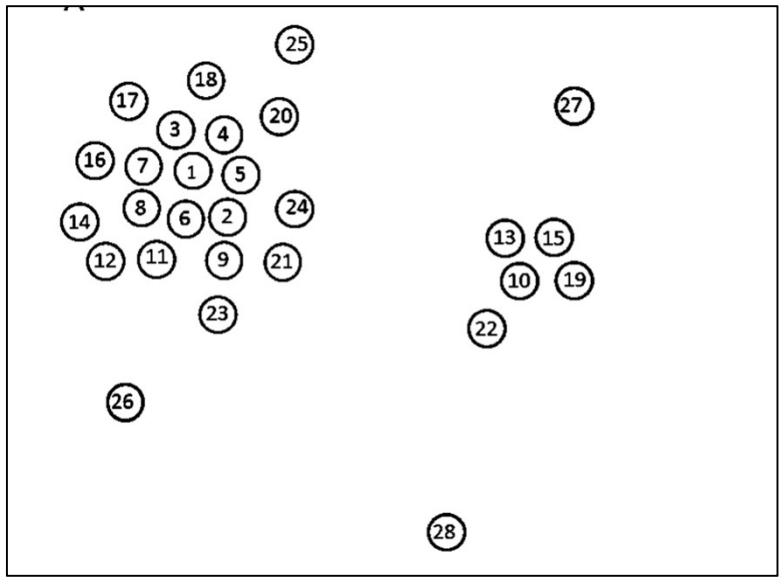
Density Peaks: δ concept

Clustering in a 2-dimensional space



Density Peaks: δ concept

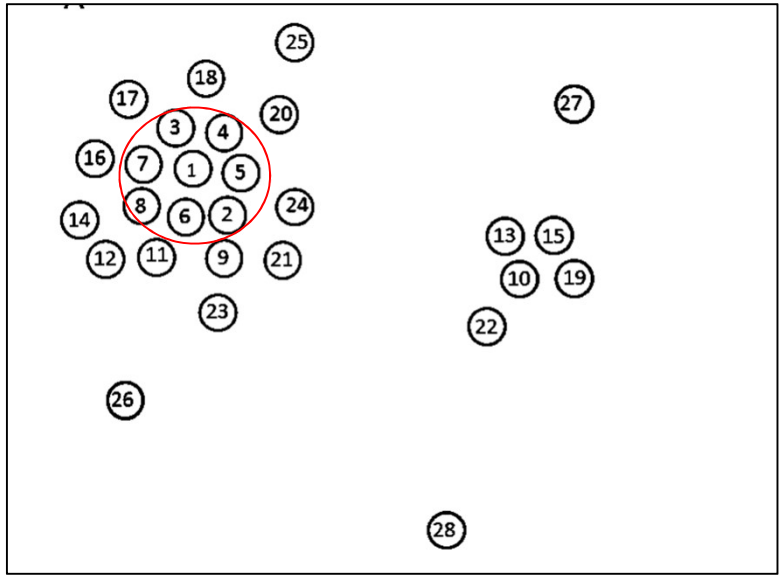
Clustering in a 2-dimensional space



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

Density Peaks: δ concept

Clustering in a 2-dimensional space

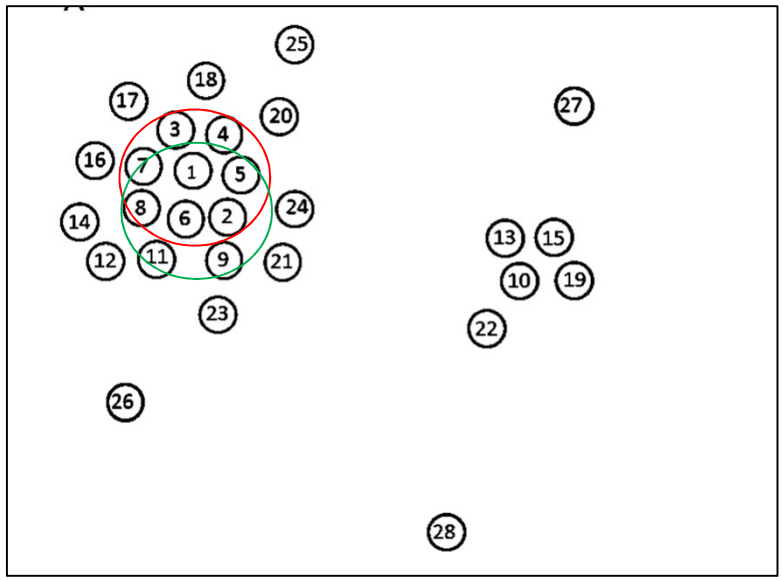


1) Compute the local density (ρ) around each point

$\rho(1)=7$

Density Peaks: δ concept

Clustering in a 2-dimensional space



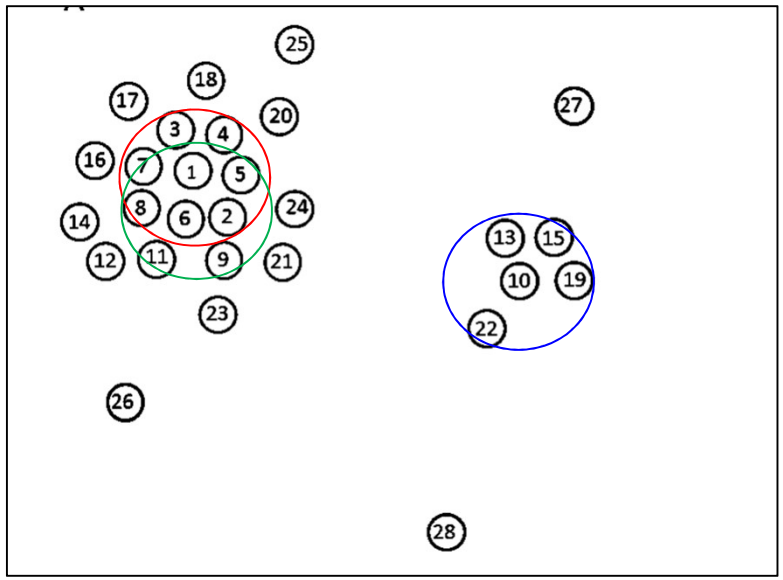
1) Compute the local density (ρ) around each point

$\rho(1)=7$

$\rho(6)=5$

Density Peaks: δ concept

Clustering in a 2-dimensional space



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

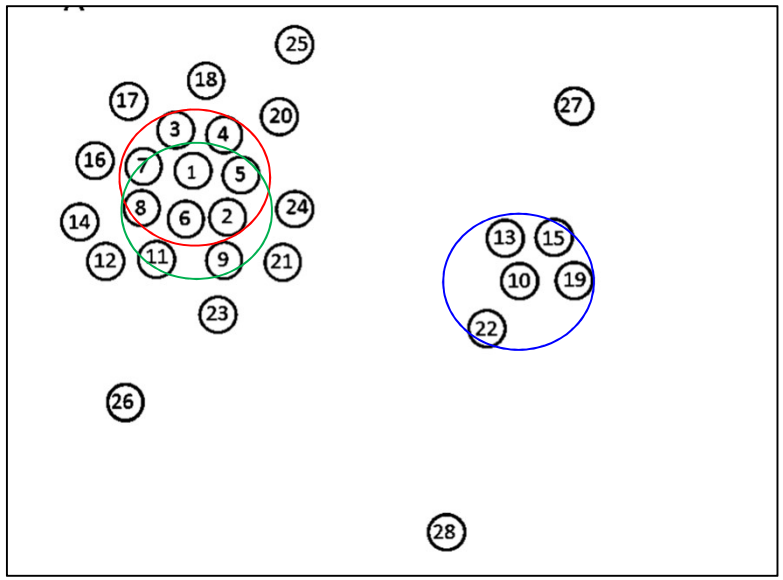
$\rho(1)=7$

$\rho(6)=5$

$\rho(10)=4$

Density Peaks: δ concept

Clustering in a 2-dimensional space



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

$\rho(1)=7$

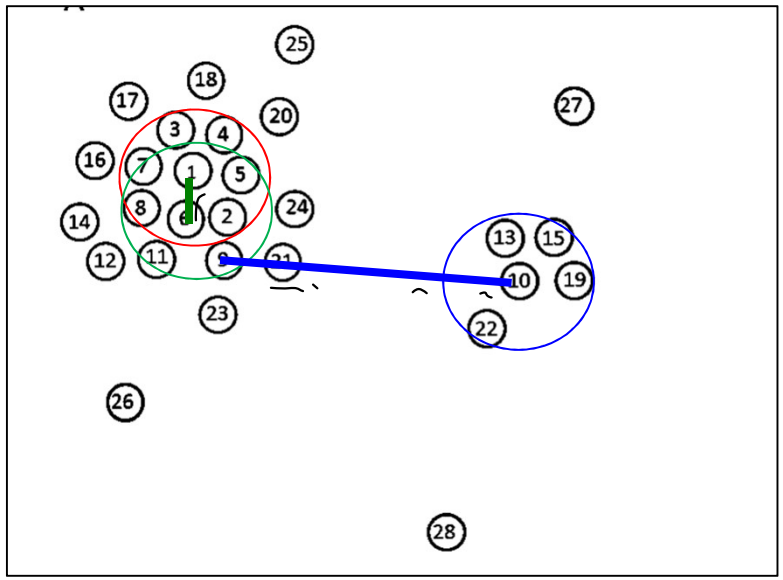
$\rho(6)=5$

$\rho(10)=4$

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

Density Peaks: δ concept

Clustering in a 2-dimensional space



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

$\rho(1)=7$

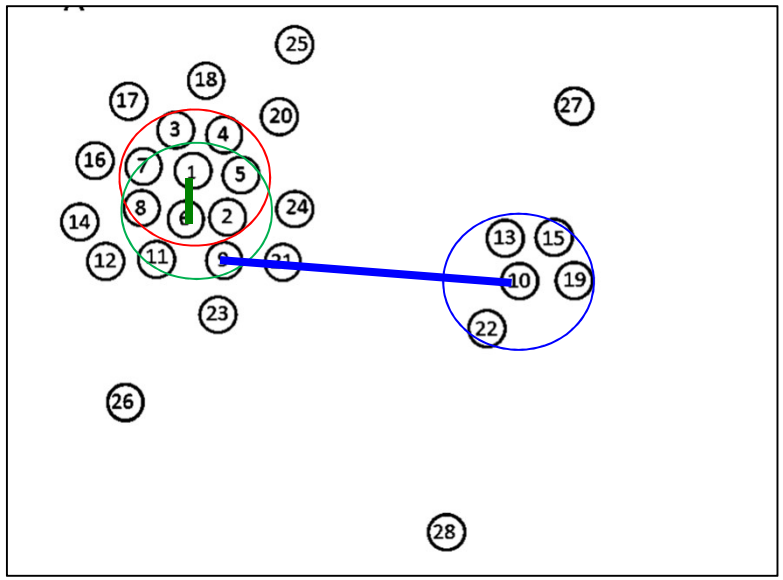
$\rho(6)=5$

$\rho(10)=4$

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

Density Peaks: δ concept

Clustering in a 2-dimensional space



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

$\rho(1)=7$

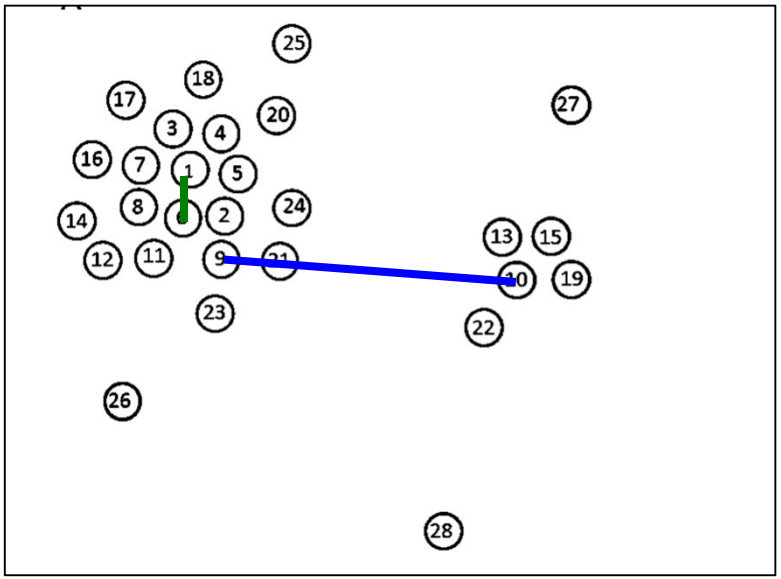
$\rho(6)=5$

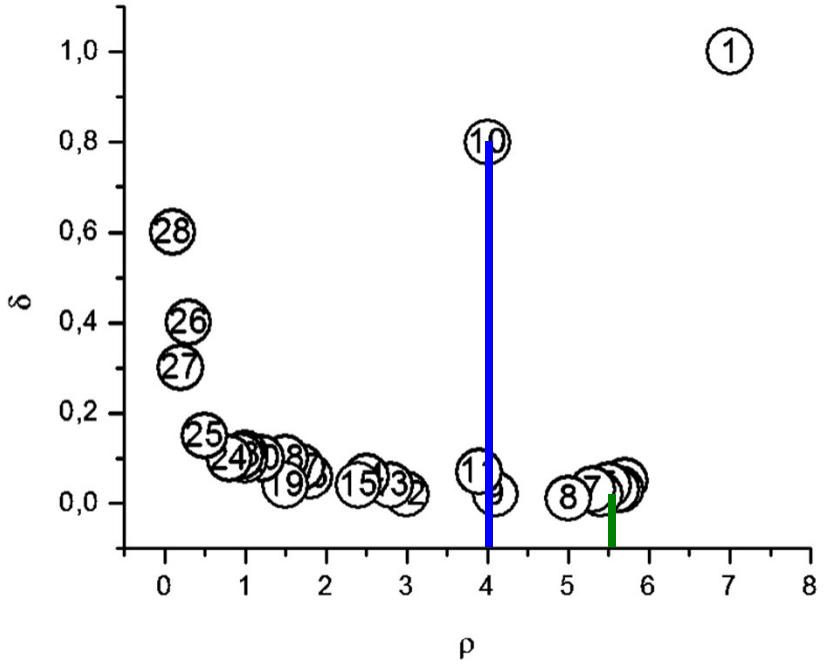
$\rho(10)=4$

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.*

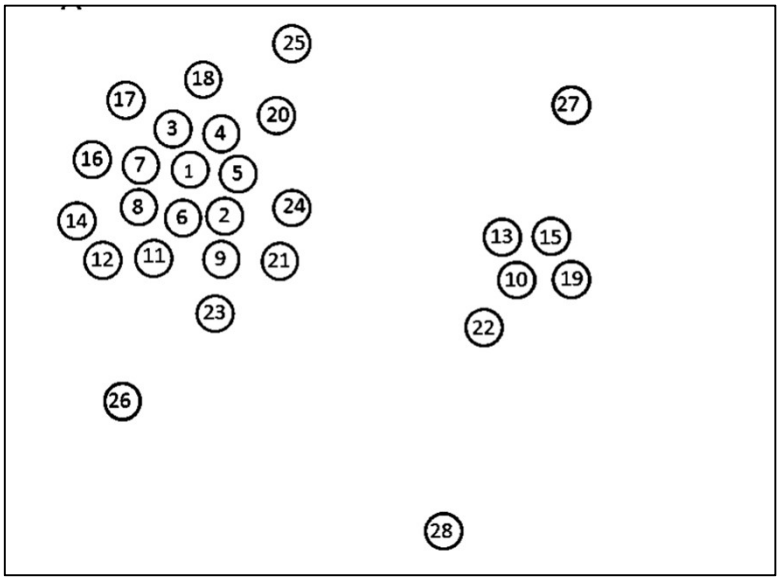


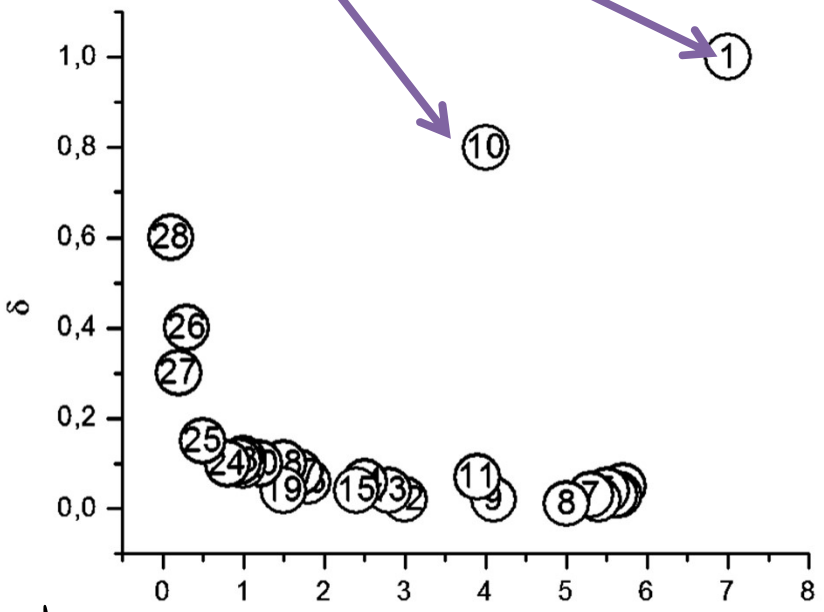


*The point with higher density has the largest delta value by convention

Density Peaks decision graph

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





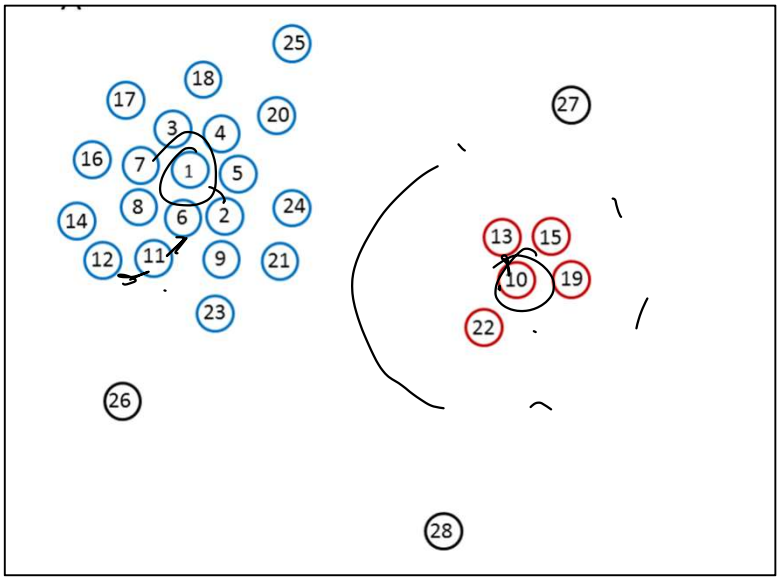
Density peaks: relatively dense & far away from denser point

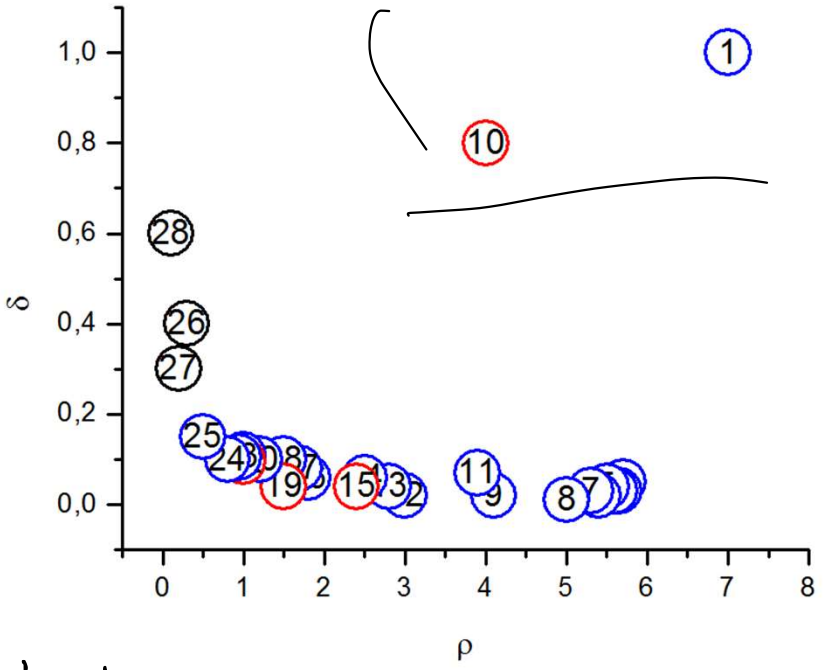
(d_{ij}^2)

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





(1) allows us to follow density profile

(2) prevents formation of convex clusters

Density Peaks algorithm so far…

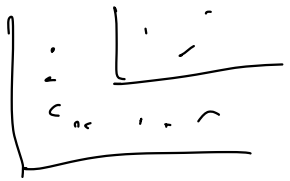
1. Given a distance matrix d_{ij}, for each data point *i* compute the associated density.

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

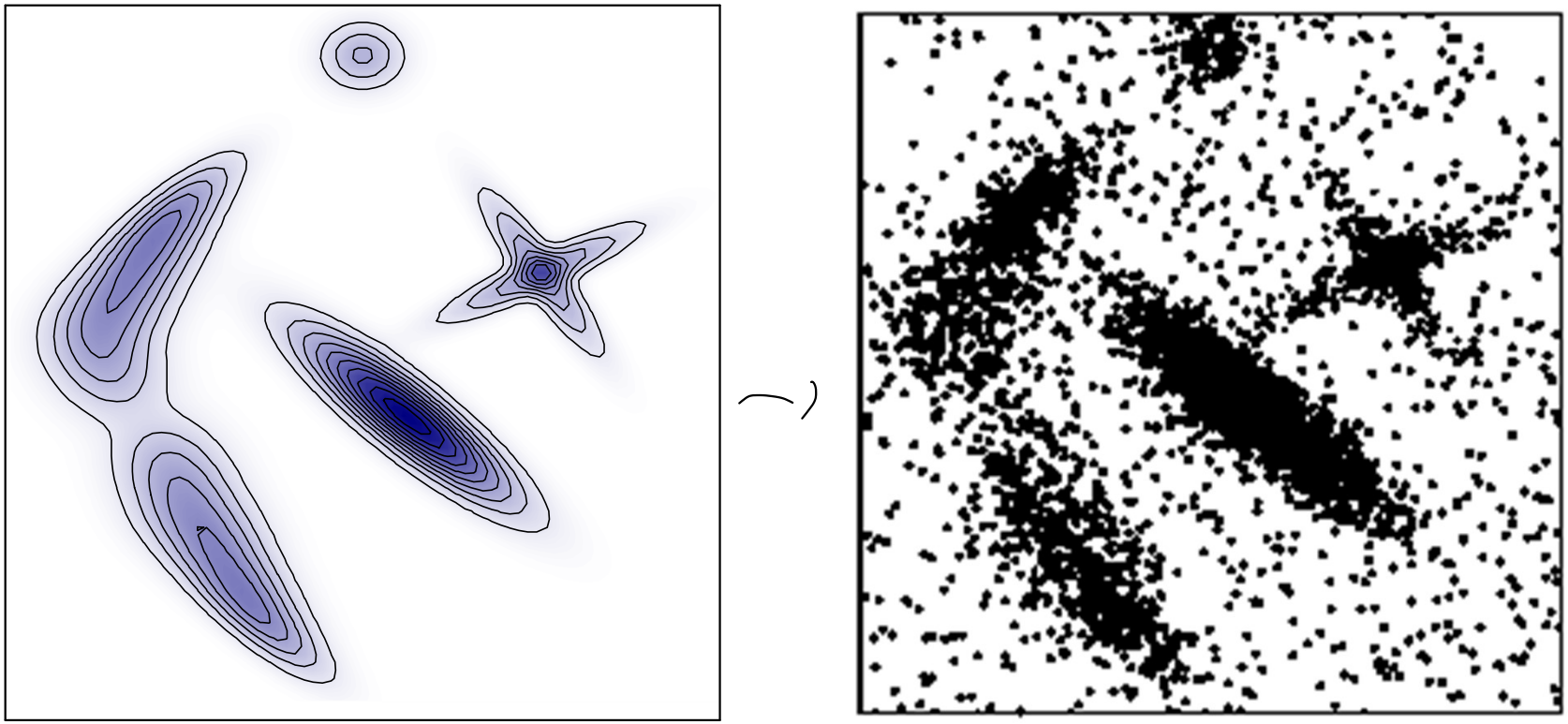
2. Sort data points in order of decreasing ρ_i and compute the minimum distance from a point with higher density (that is, those before in the sorted array).

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

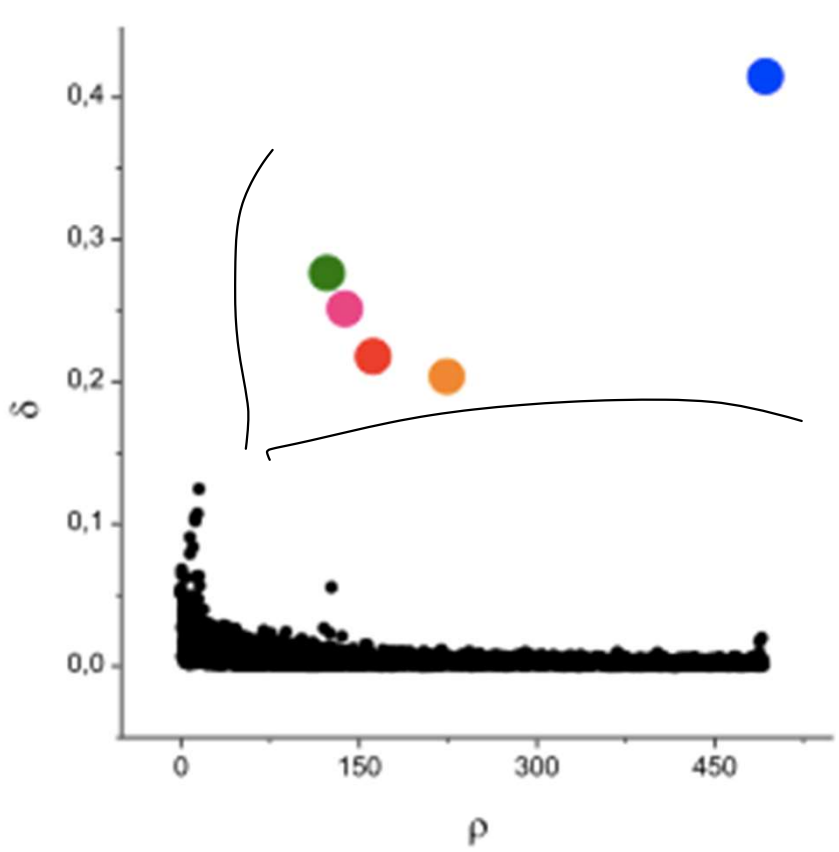
3. The delta for the first element can be arbitrarily assigned at the end as 5% higher than the maximum computed one.
4. Generate the decision graph (scatter plot ρ_i vs δ_i) and identify the outliers. Each of them is assigned to a different cluster.
5. Assign each point to the same cluster of its nearest neighbor of higher density. These can be done efficiently by storing the *argmin* in step 2.

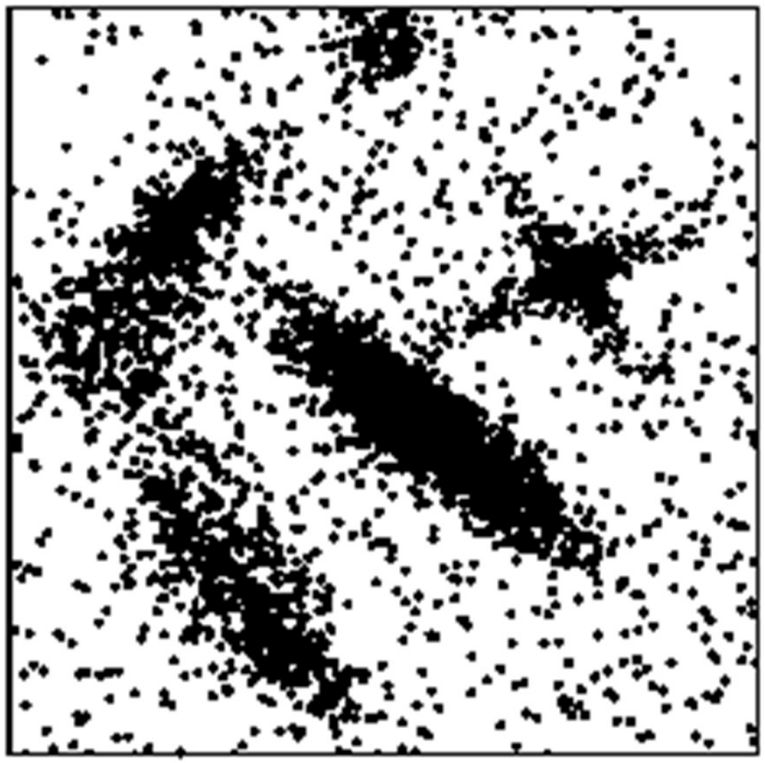


Density Peaks clustering example

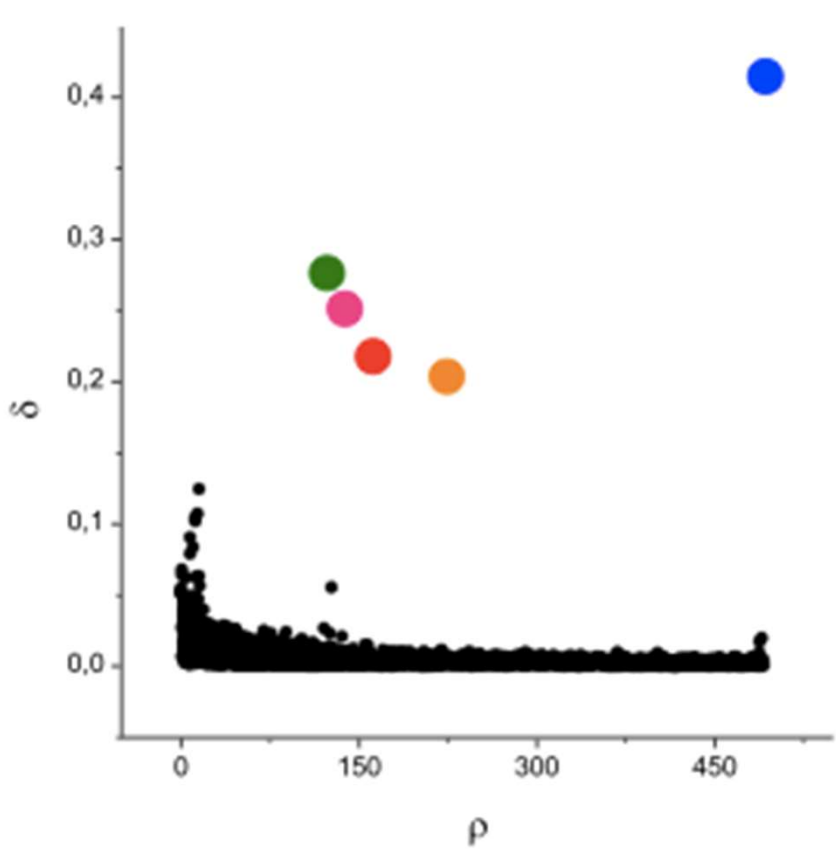


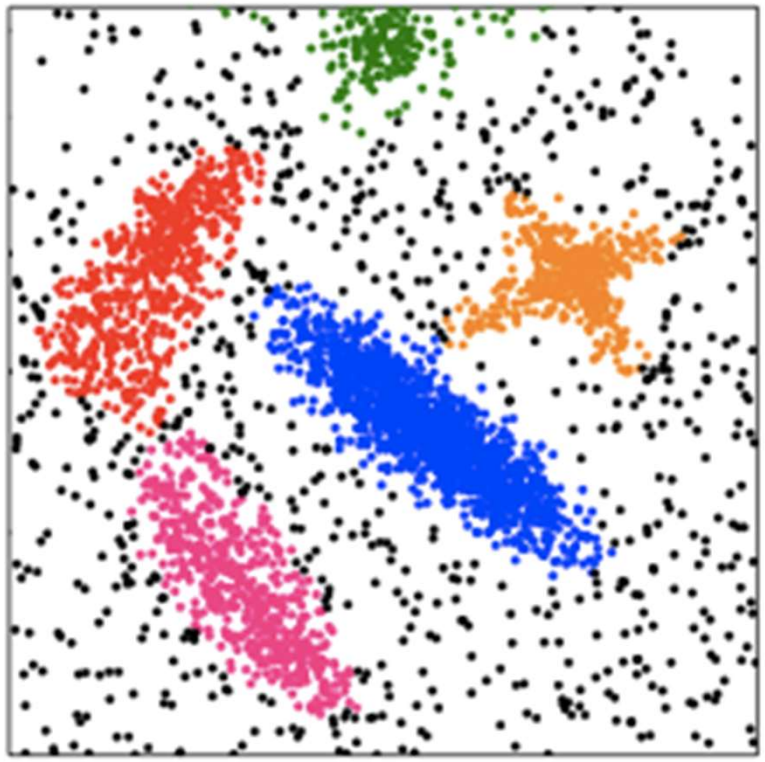
Density Peaks clustering example



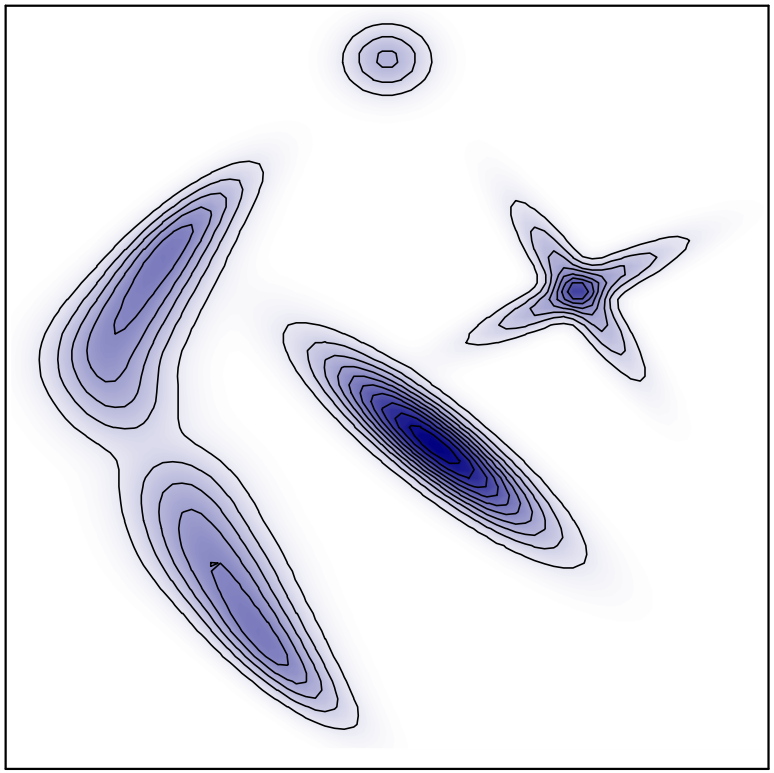


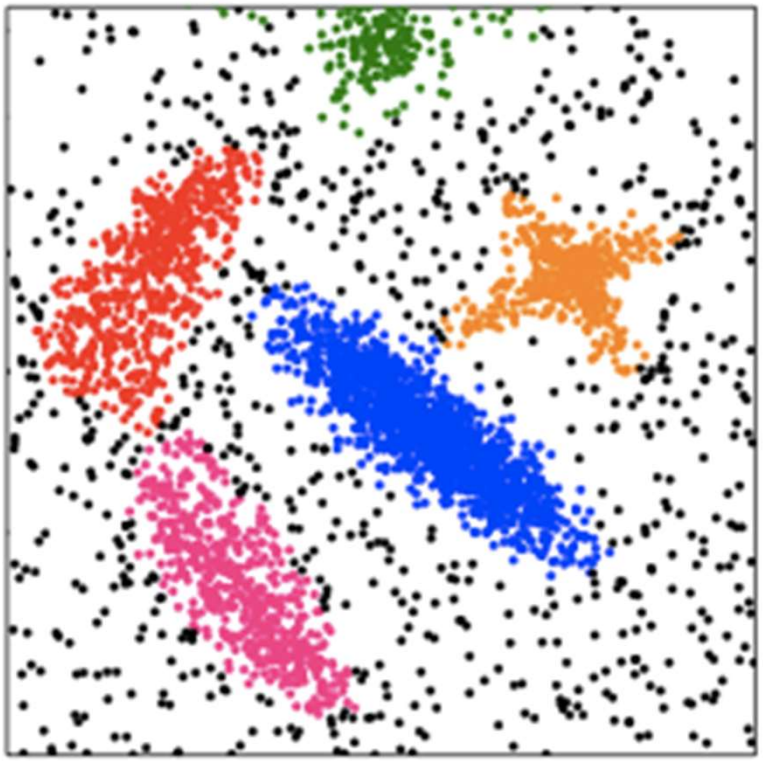
Density Peaks clustering example



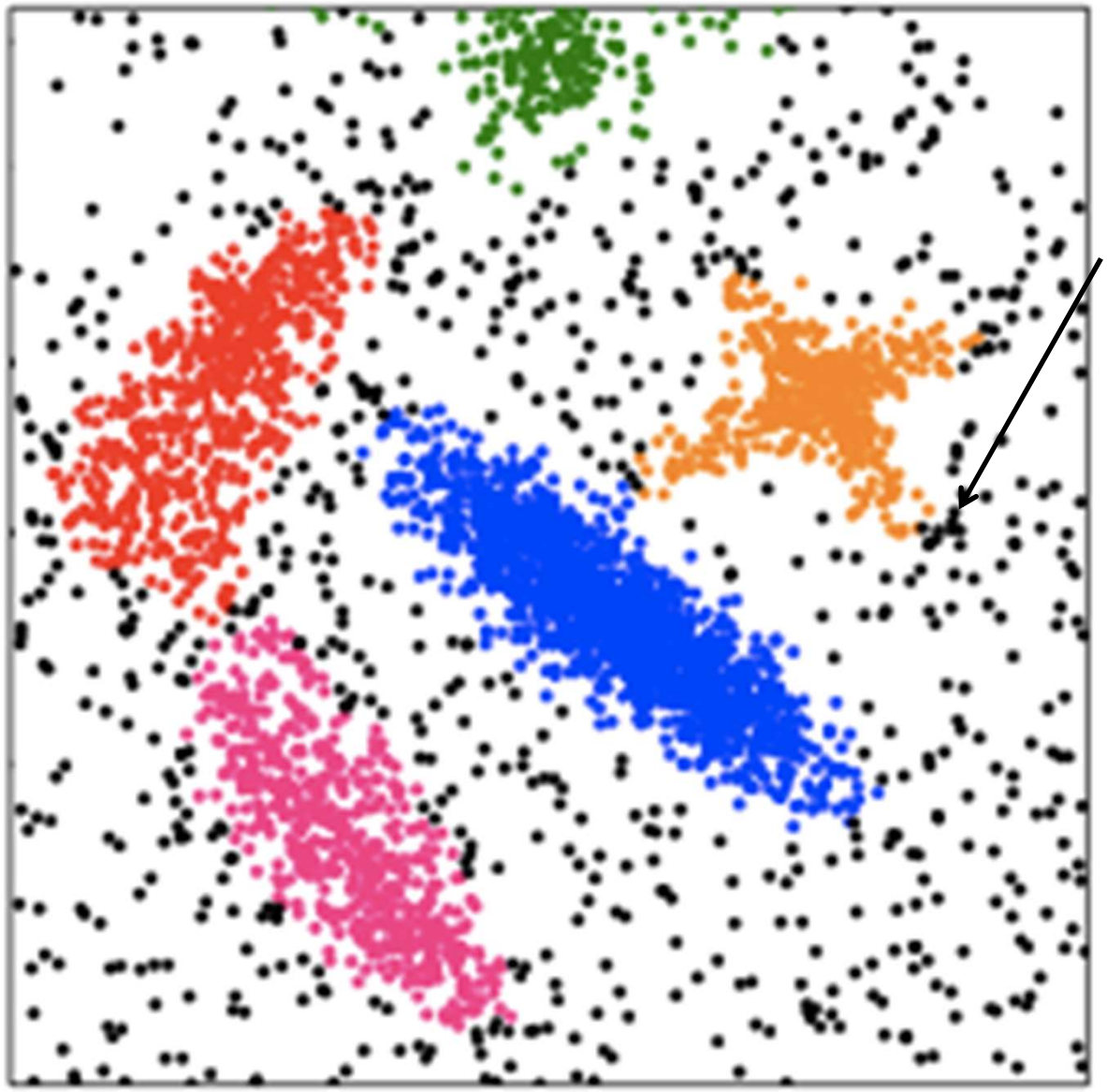


Density Peaks clustering example





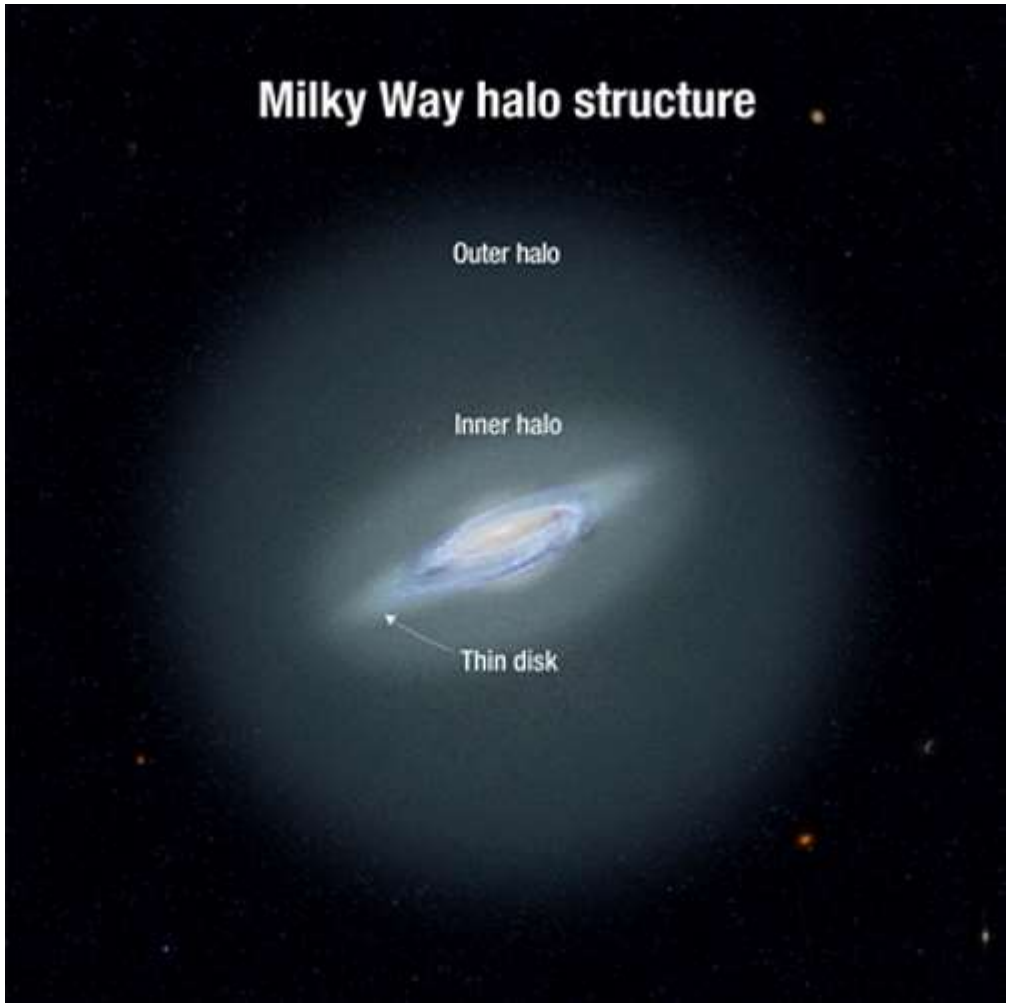
Density Peaks: Halo



What are these black points?

Core ? Noise ? NO !

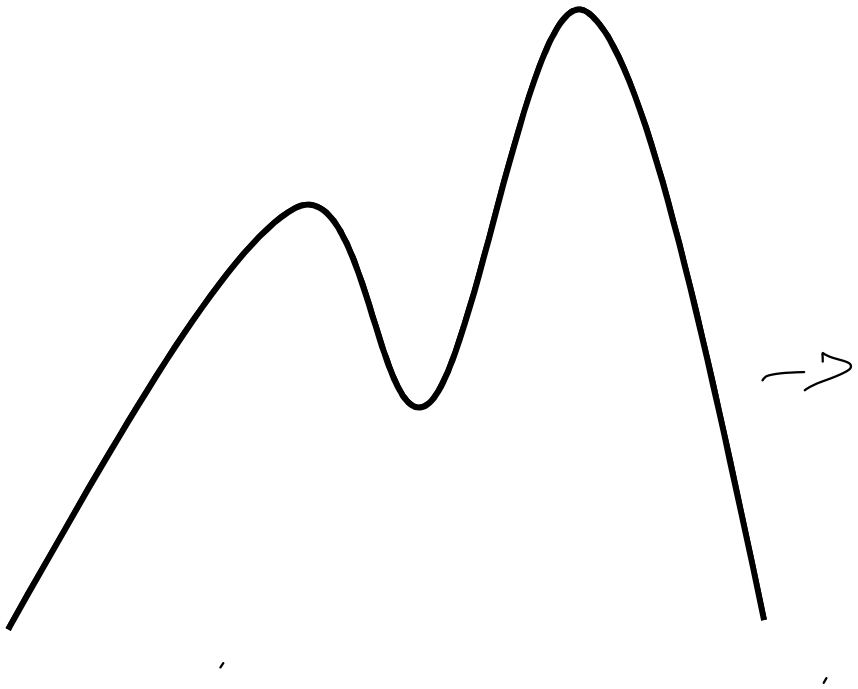
Density Peaks: Halo

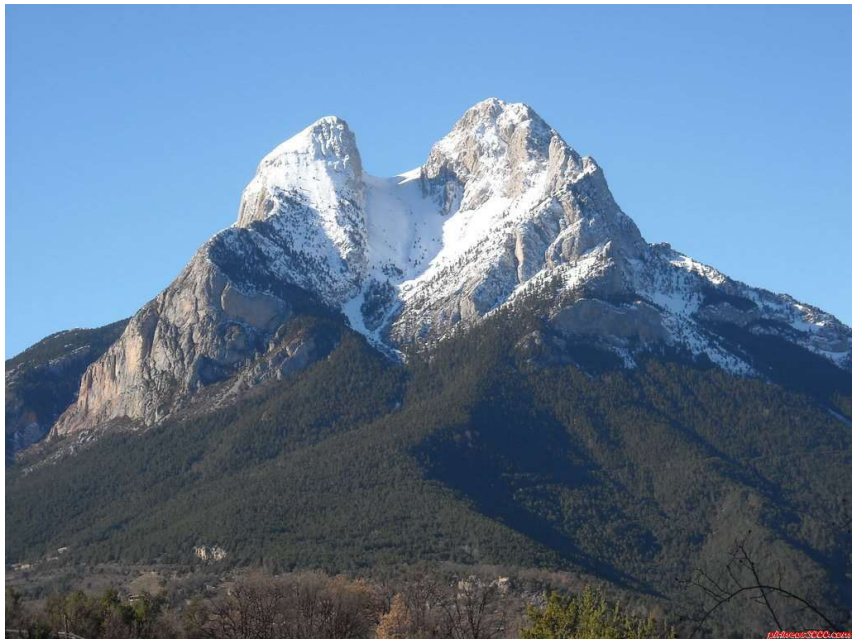


→ Points that are far away from the denser parts of the galaxy.

Density Peaks: Halo

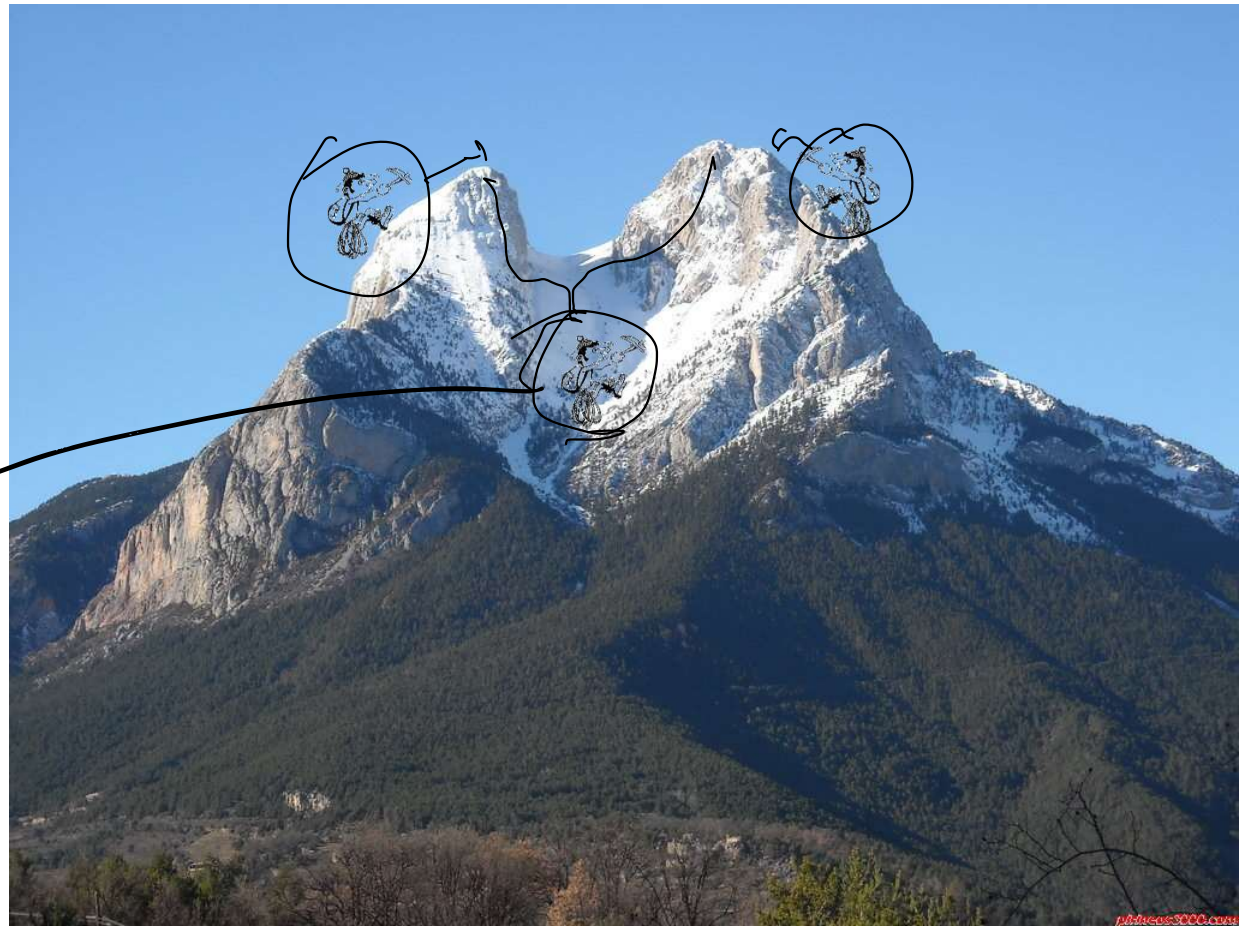
Halo concept: mountain analogy





Density Peaks: Halo

Could you tell which peak are these people climbing?

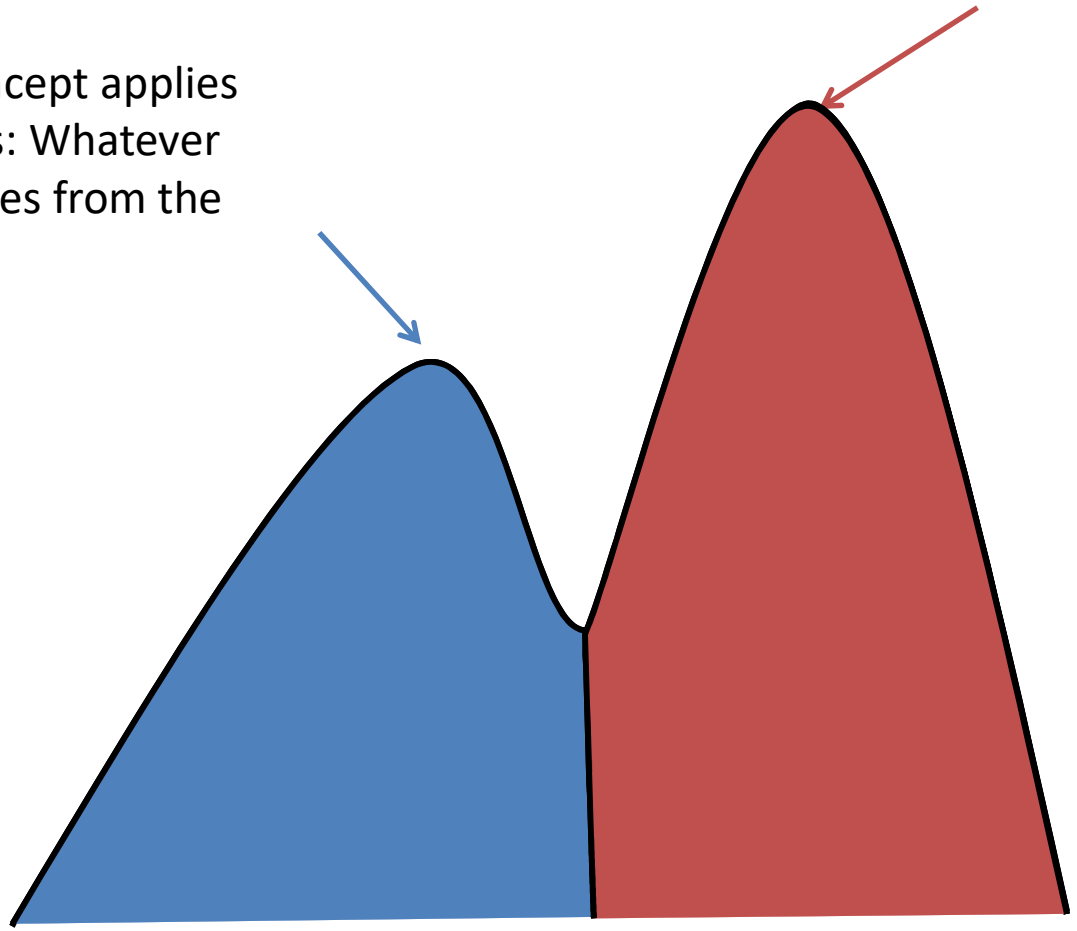


which peak in this climber climbing?

Hard to say.

Density Peaks: Halo

The same concept applies to data points: Whatever partition comes from the clustering…

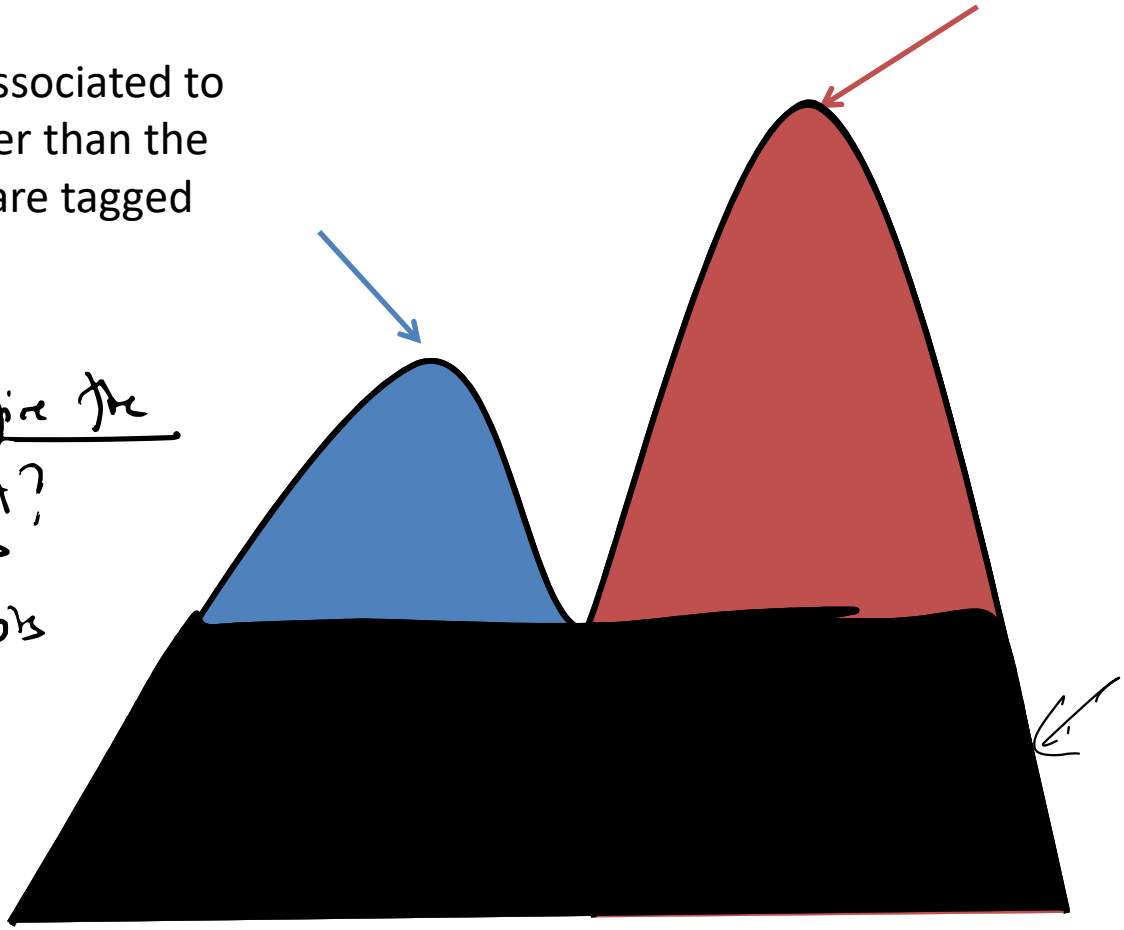


Density Peaks: Halo

Data points associated to a density lower than the saddle point are tagged as "halo"

How to define the saddle point?

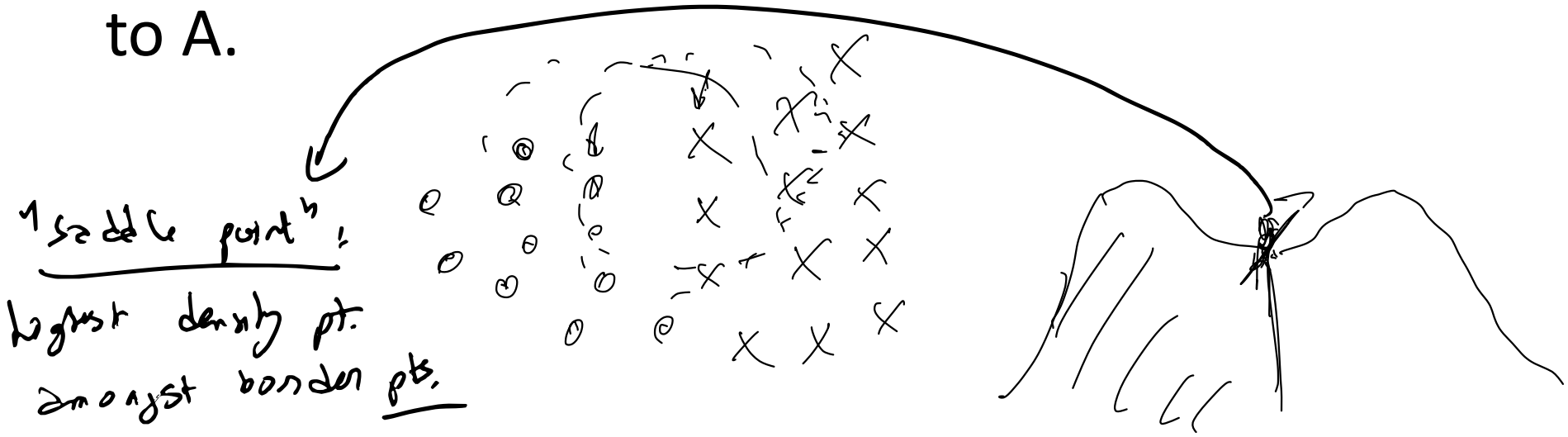
→ find data pts that are at the border btw clusters. "Saddle" is data pt. at the border with higher density.



Border definition

(used to define saddle point)

- 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.



Density Peaks algorithm (finally)

1. Given a distance matrix d_{ij}, for each data point *i* compute the associated density.

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

2. Sort data points in order of decreasing ρ_i and compute the minimum distance from a point with higher density (that is, those before in the sorted array).

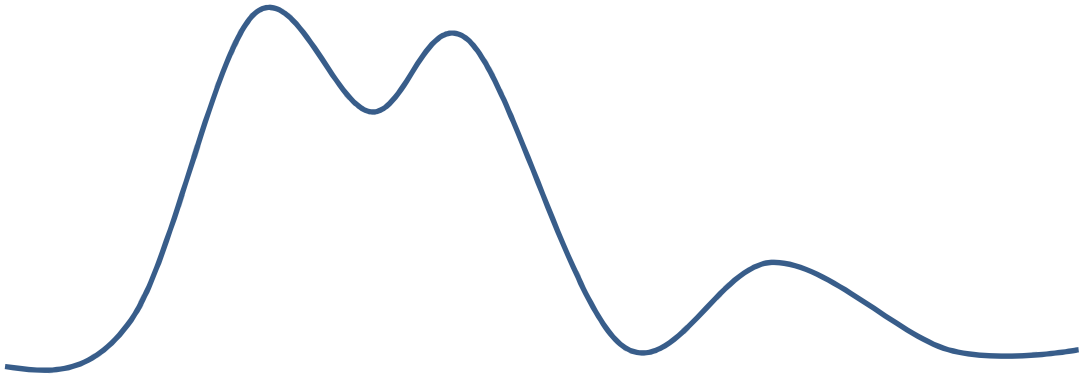
$$\delta_i = \min_{\rho_i < \rho_j} (d_{ij})$$

3. The delta for the first element can be arbitrarily assigned at the end as 5% higher than the maximum computed one.
4. Generate the decision graph (scatter plot ρ_i vs δ_i) and identify the outliers. Each of them is assigned to a different cluster.
5. Assign each point to the same cluster of its nearest neighbor of higher density. These can be done efficiently by storing the *argmin* in step 2.
6. Check the neighborhoods of each data points and assign as border points those that have an element of other cluster within d_c.
7. Find the border density of a cluster j as the higher density among the border points belonging to this cluster. All the points that belong to cluster j and have density lower than the border density are tagged as halo.

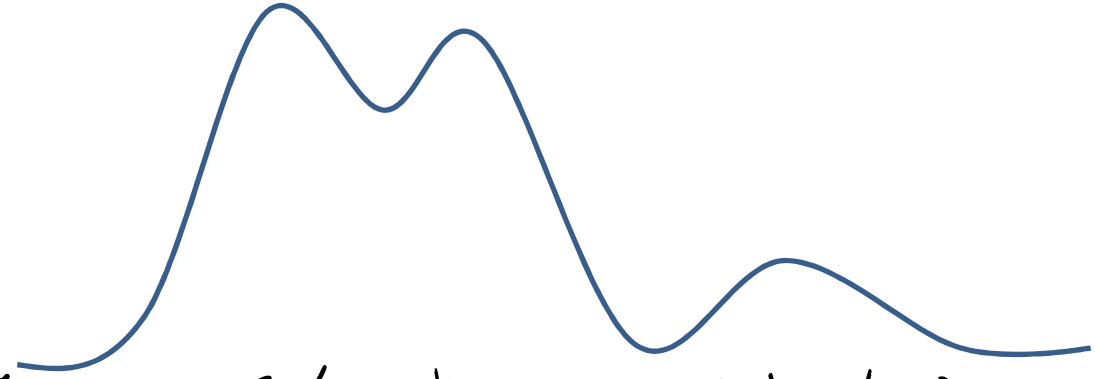
"saddle point"

Differences between DBSCAN and Density Peaks clustering

DBSCAN



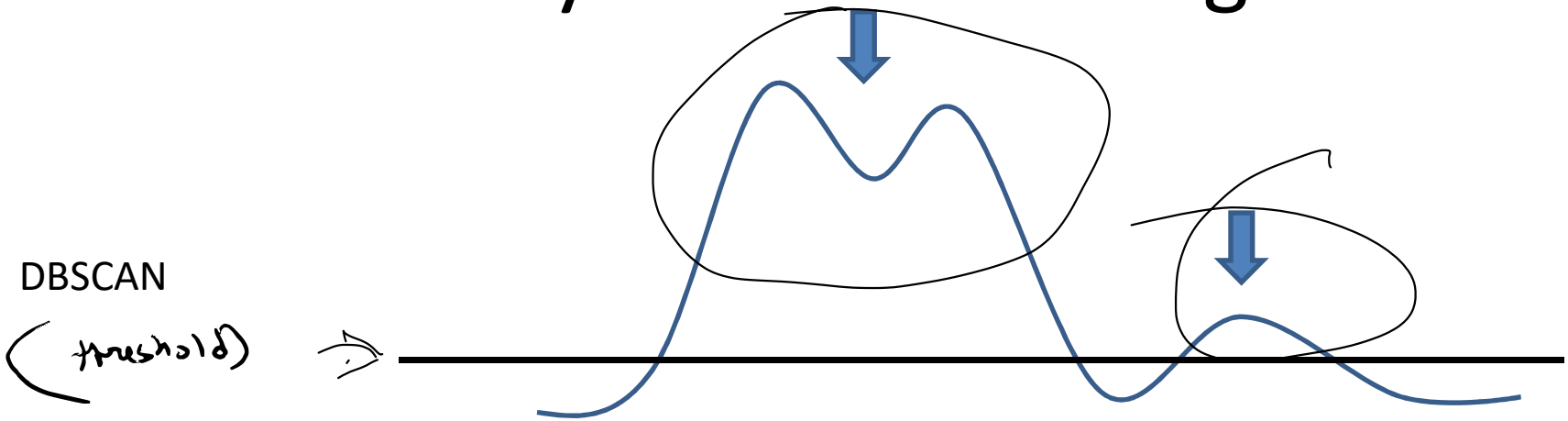
Density Peaks



→ comparison of defining "k" ols ε/minpts :- defining "k" in a strong assumption, i.e we give a certain shape to our data. But dependence on ε/minpts is weaker assumption so density based clustering is better.

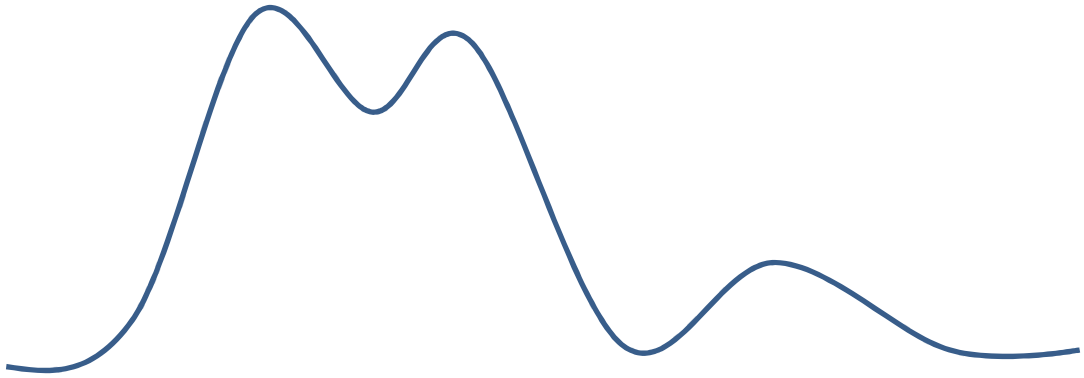
Differences between DBSCAN and Density Peaks clustering

DBSCAN

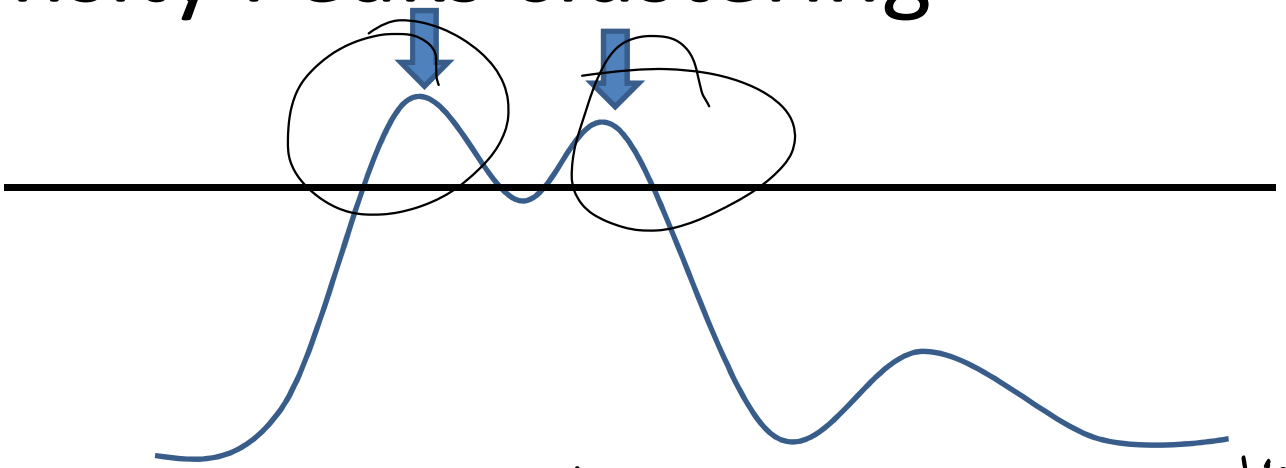


Threshold in the density: minPts/ε

Density Peaks



Differences between DBSCAN and Density Peaks clustering

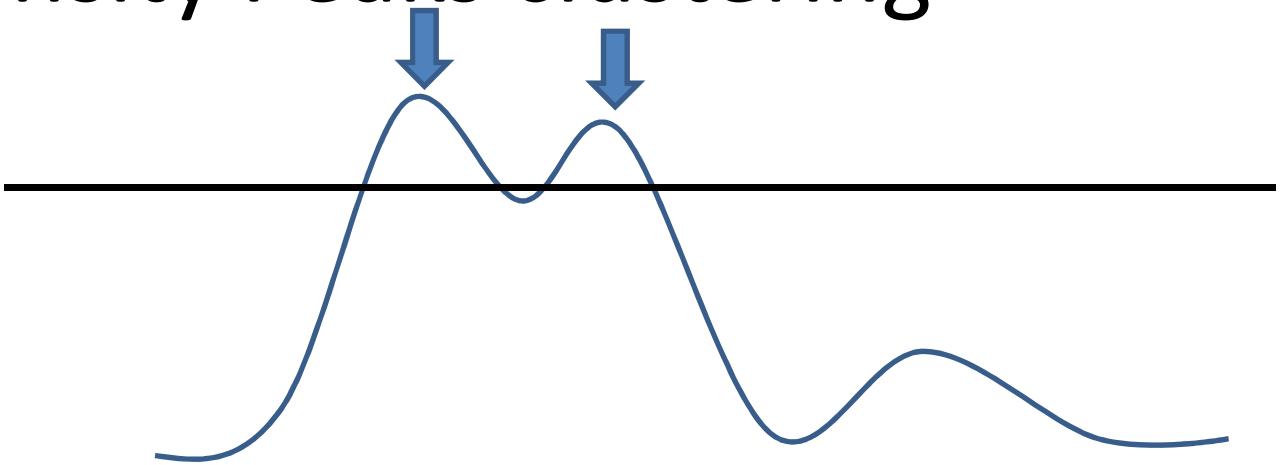


DBSCAN

Threshold in the density: minPts/ε → ⁄, DBSCAN can never find the 3 peaks. It has problems when density of clusters is really different → setting a common threshold for very diff. densities is very hard.

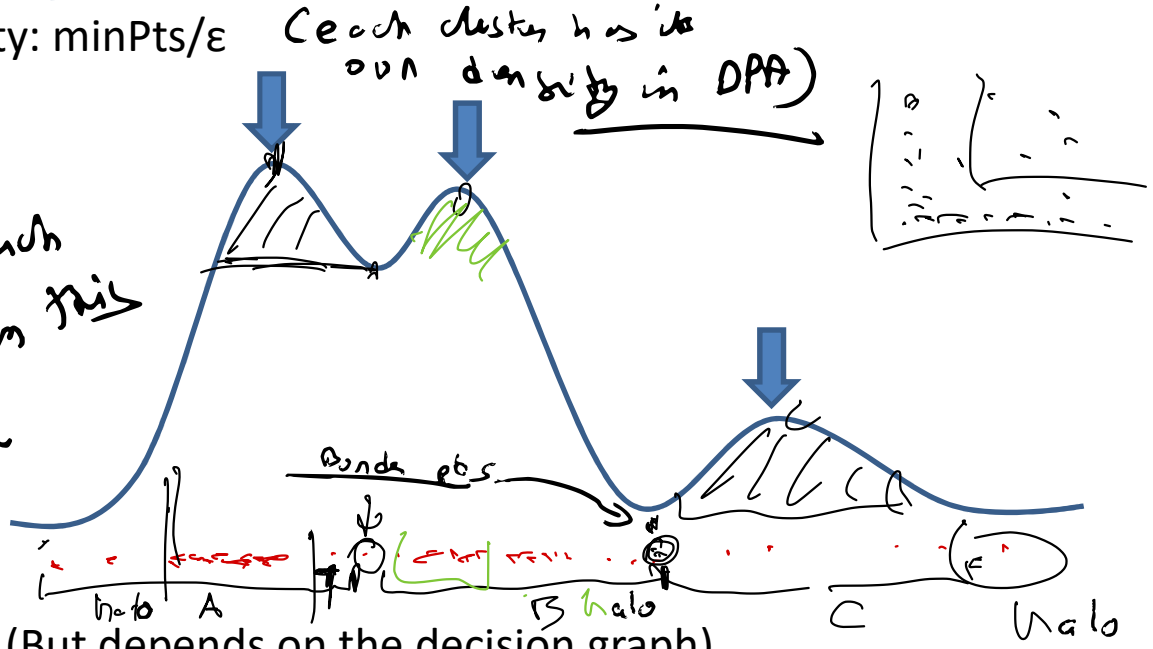
Density Peaks

Differences between DBSCAN and Density Peaks clustering



DBSCAN

Threshold in the density: minPts/ε



Density Peaks (works much better in this case)

not always clear how to define outliers.

Directly find the peaks (But depends on the decision graph)

Mean Shift cluster

Modern clustering algorithms (VI)

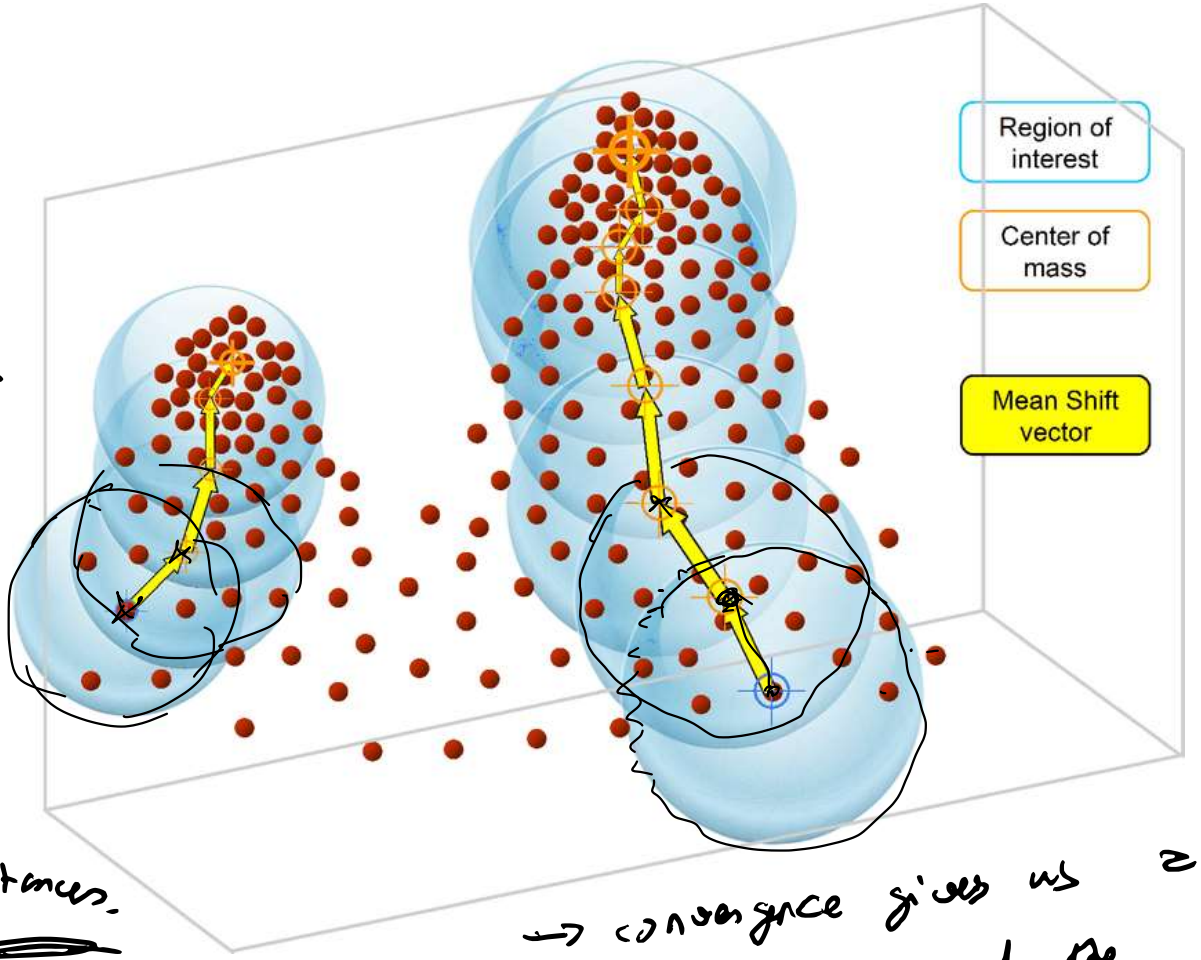
Mean-Shift algorithm

- The idea is similar to Density-Peaks: Find the density maxima and assign the data points to its correspondent maxima.
- The density is estimated with kernel density estimation.
- For each point, shifts its coordinates towards the weighted mean of the density $m(x) = \frac{\sum_{x_i \in N} x_i K(x_i - x)}{\sum_{x_i \in N} K(x_i - x)}$ in a region of interest until it converges.
- If the kernel is flat (Parzen windows), $m(x)$ is the average position of the points within the region of interest.

Compute weighted avg. pos. of all pts.

K is a kernel

Other important clustering algorithms:
Mean-Shift



Some ideas about Cluster Validation

Cluster validation

Supervised classification:

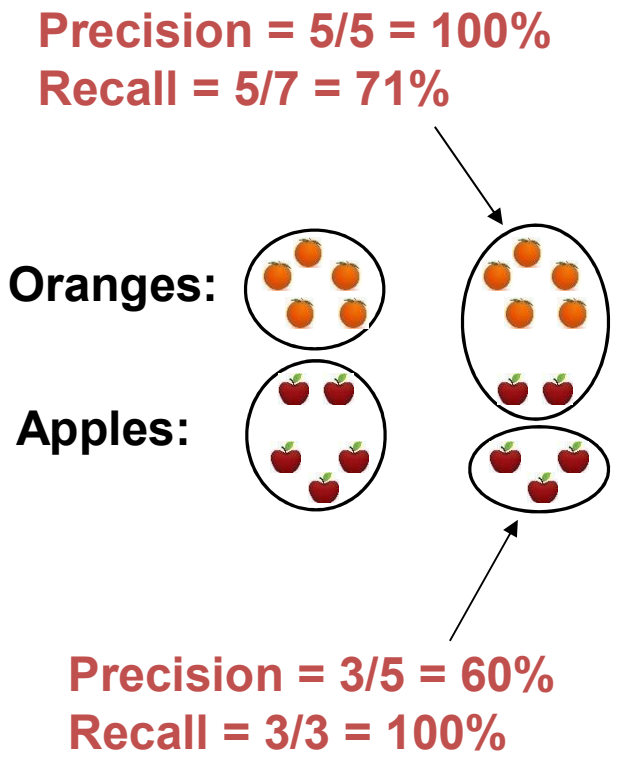
- Class labels known for ground truth
- Accuracy, precision, recall

Cluster analysis

- No class labels

Validation need to:

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



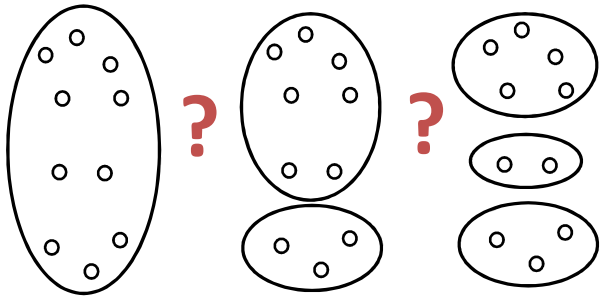
→ There is no rule that we'll have same # of clusters as class labels.

→ ∴ validating clustering is much harder.

Measuring clustering validity

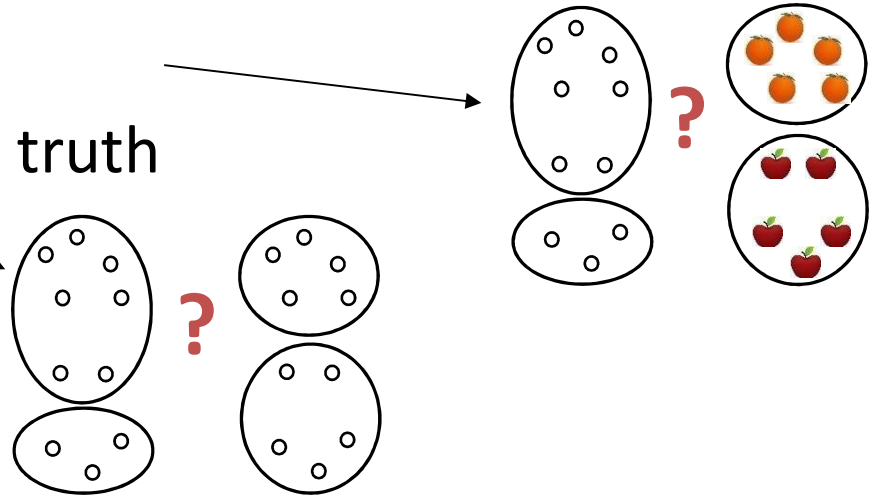
Internal Index:

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



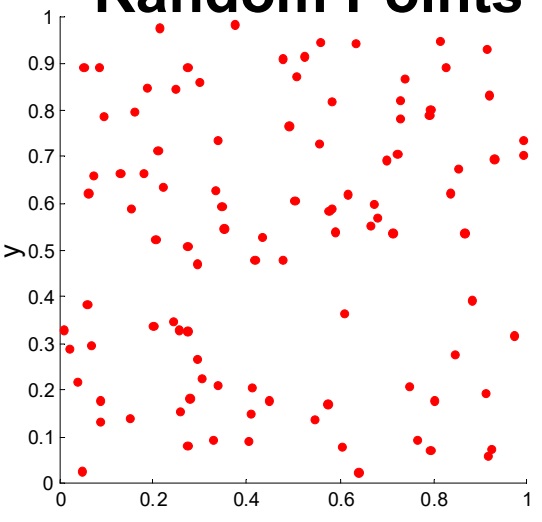
External Index

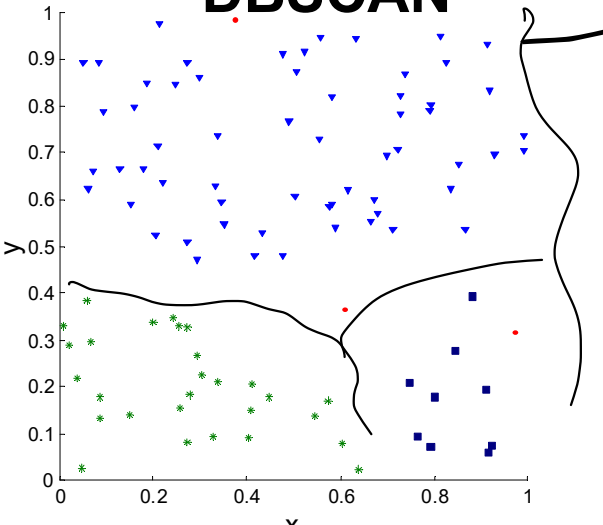
- Validate against ground truth
- Compare two clusters: (how similar)



Clustering of random data

(useful test)





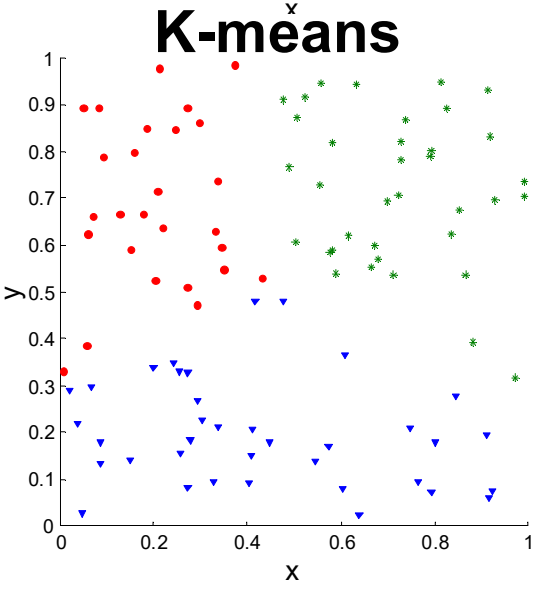
might find density fluctuations

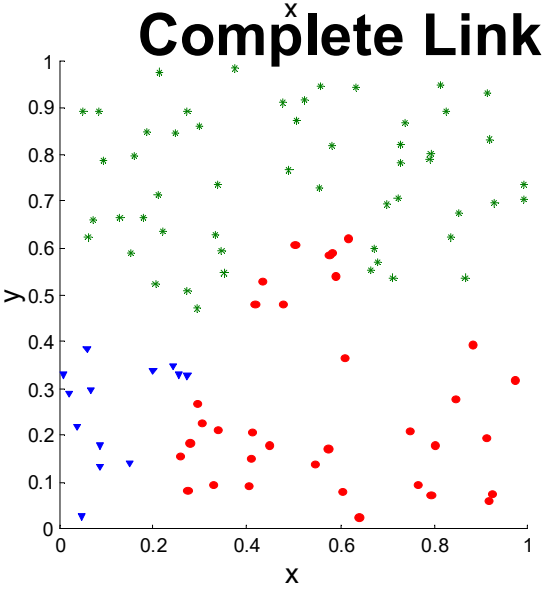
depends on params.

↓

density methods work better in NOT clustering random data.

by def? will give us hard partitions



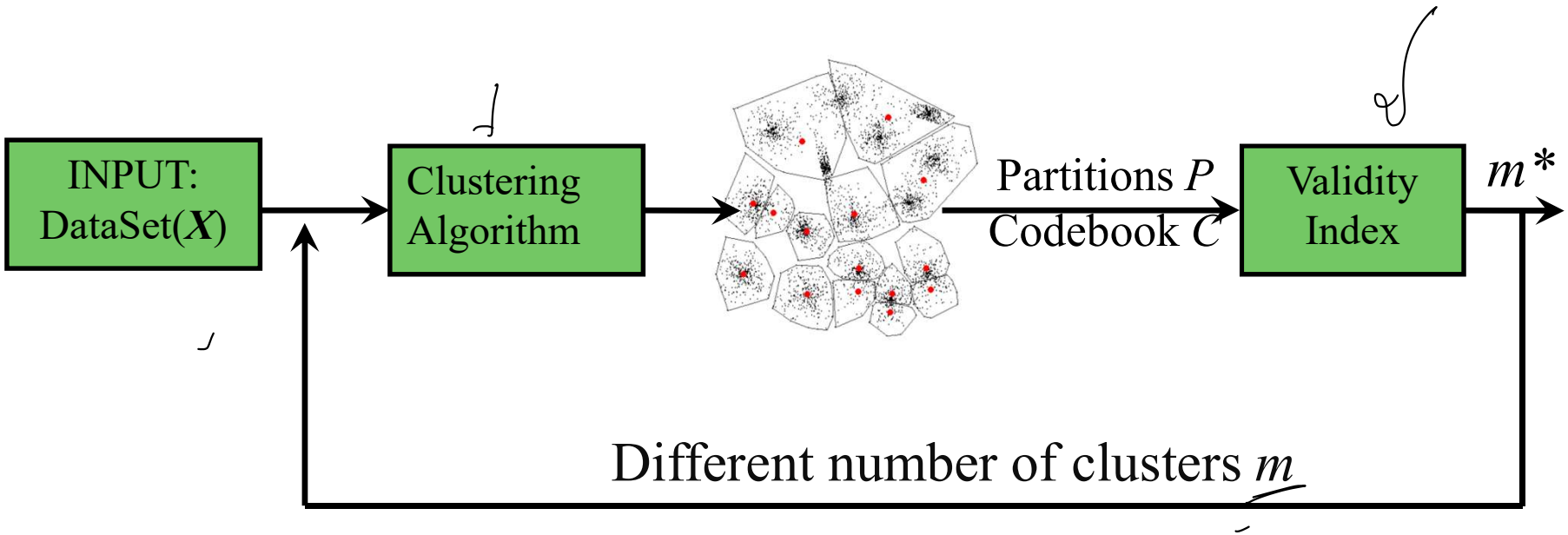


Cluster validation process

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.

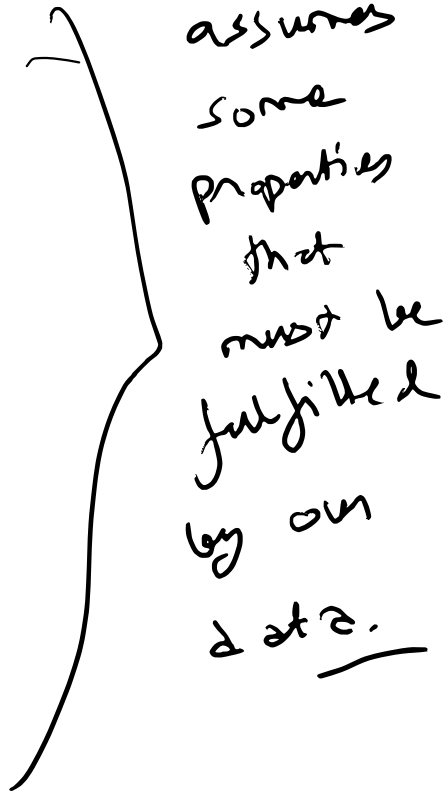
Cluster validation process

- **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!



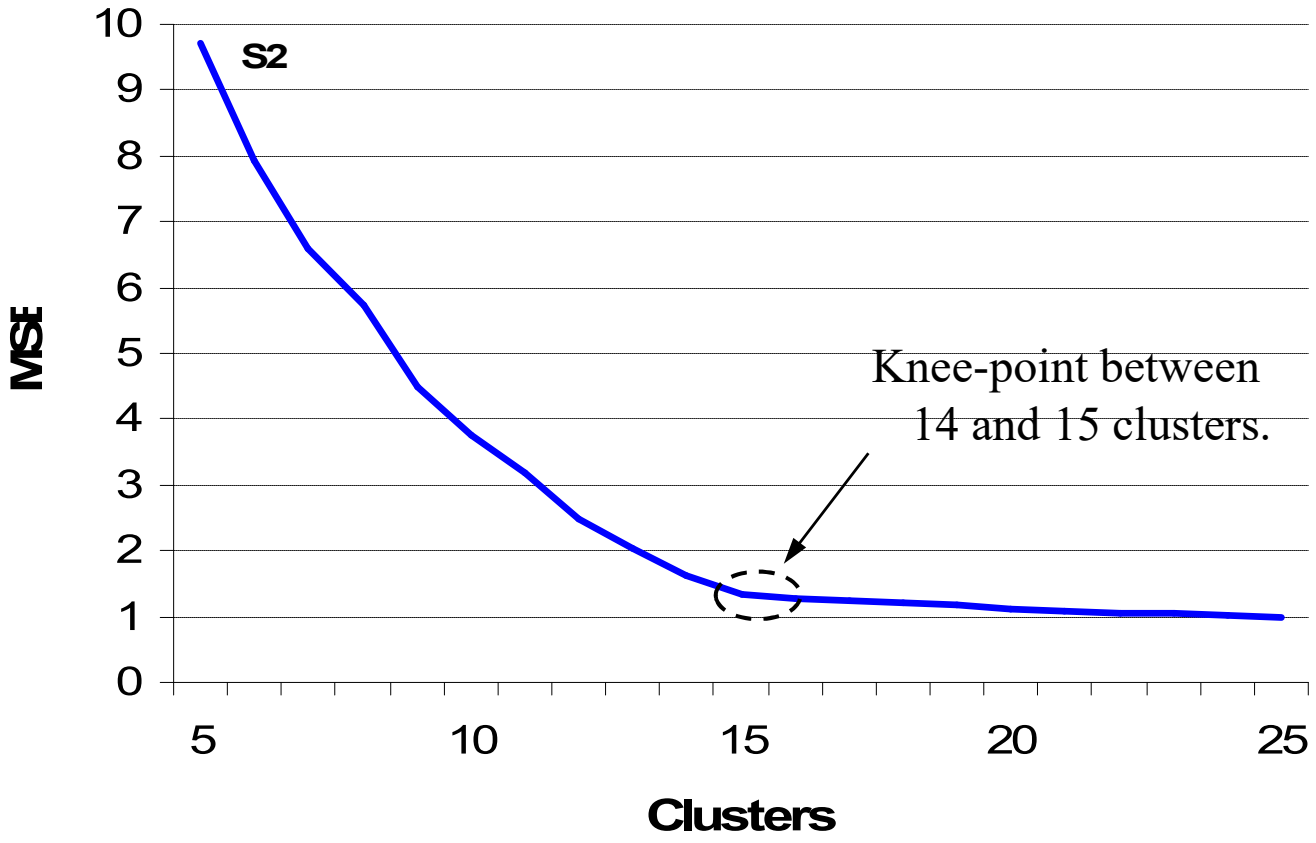
Internal indexes

- 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)

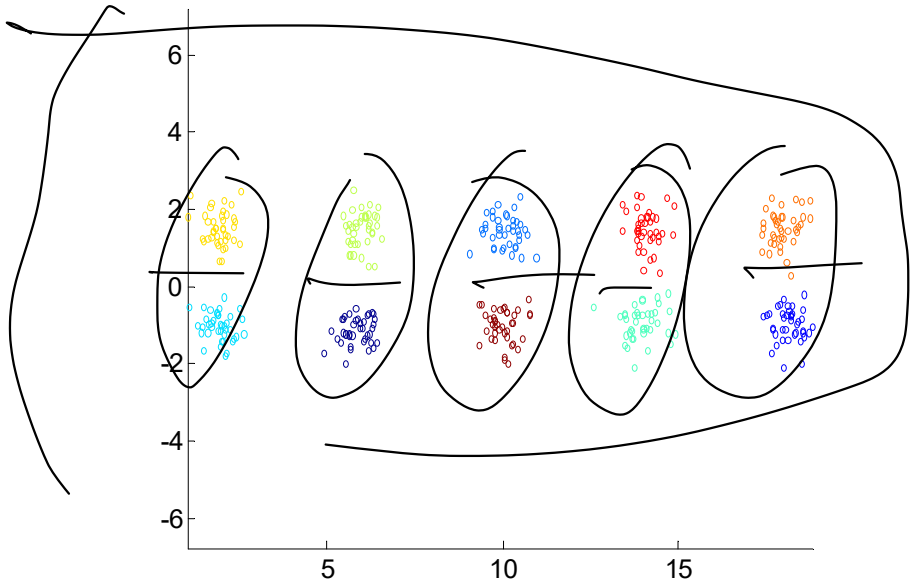


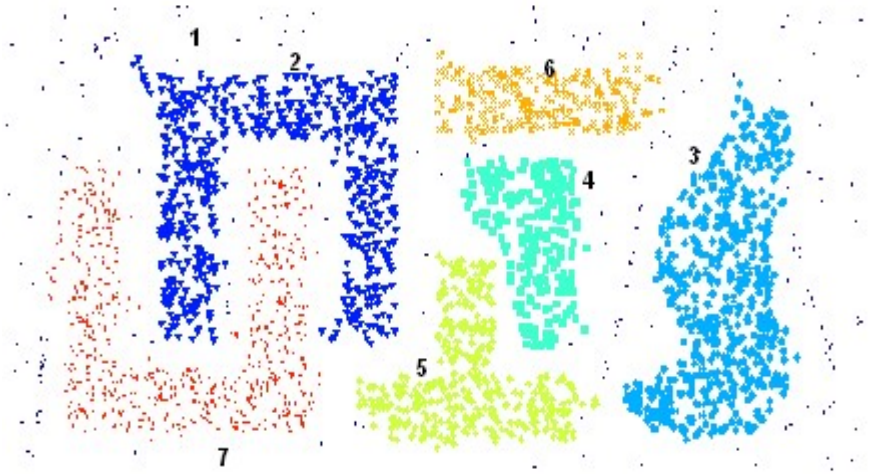
Mean square error (MSE)

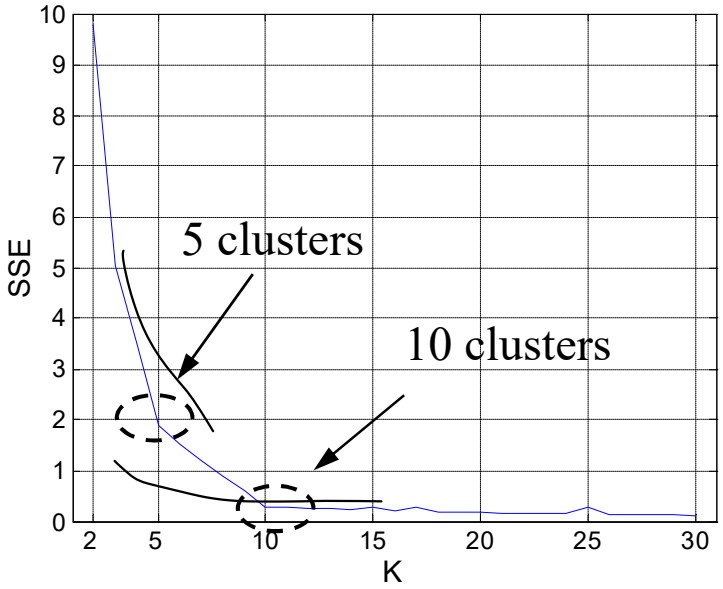
- The more clusters the smaller the MSE.
- Small knee-point near the correct value.
- But how to detect?

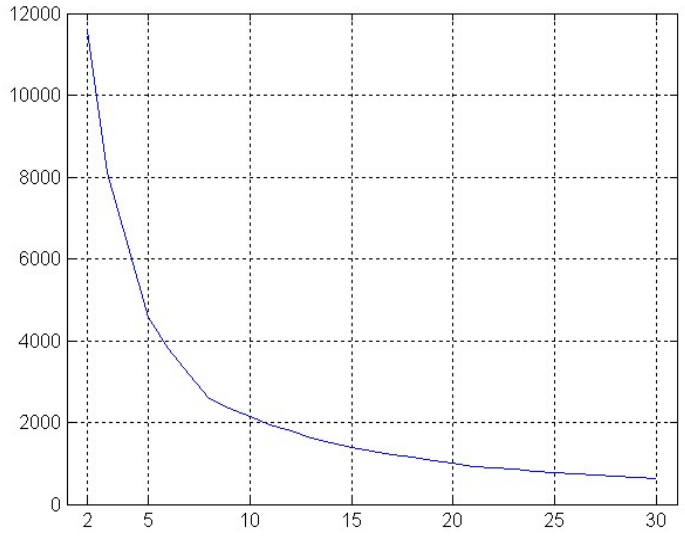


Mean square error (MSE)



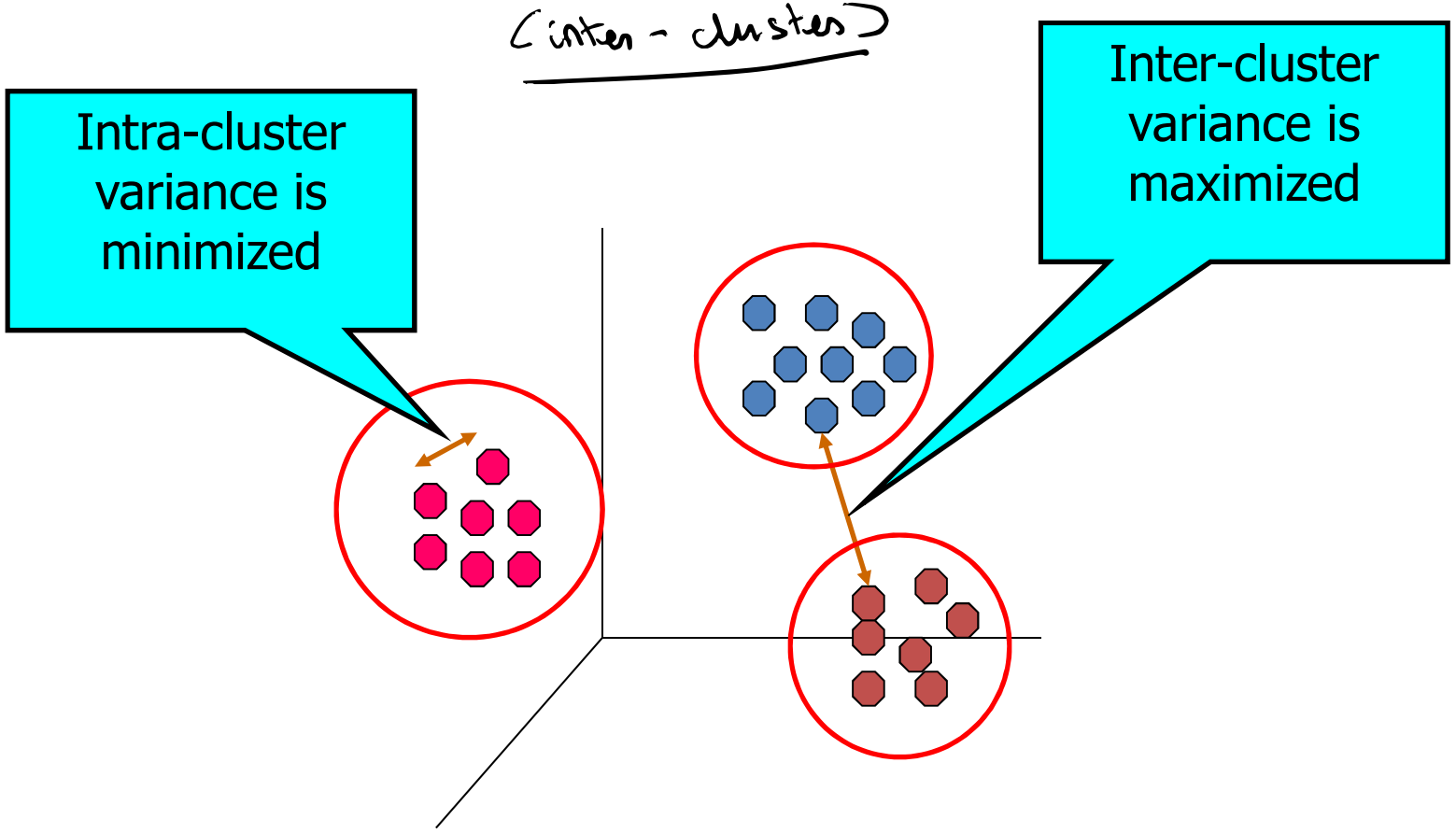






From MSE to cluster validity

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

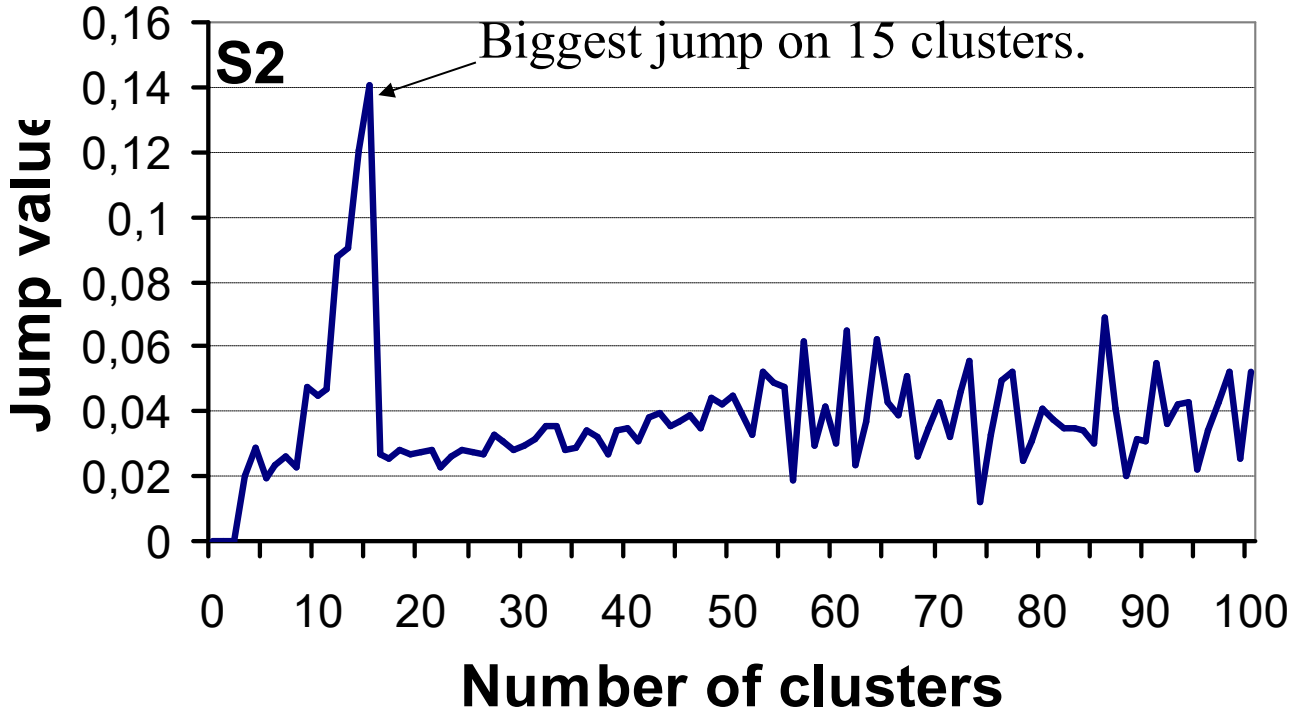


Jump point of MSE

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

Total Variance of data set:

$$\sigma(X) = \sum_{i=1}^{N} \| x_i - c_{p(i)} \|^2 + \sum_{j=1}^{k} n_j \| c_j - \bar{x} \|^2$$

SSW

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):

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

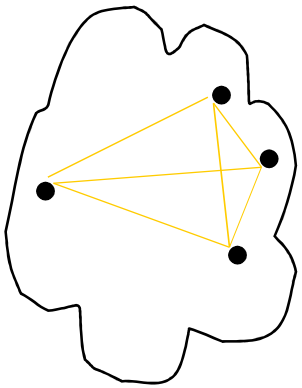
SSB

→ lower the F-ratio, better the clustering.

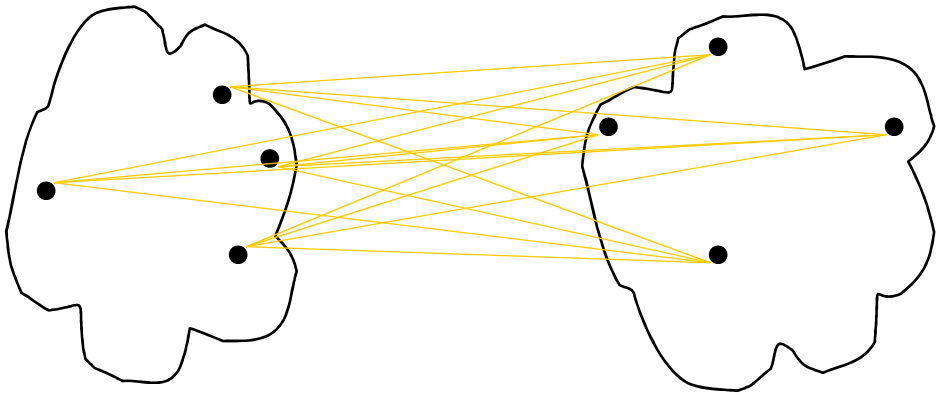
Silhouette coefficient

[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

→ work with distances while F-ratio only works with coords.
→ But much more computationally expensive than F-ratio.

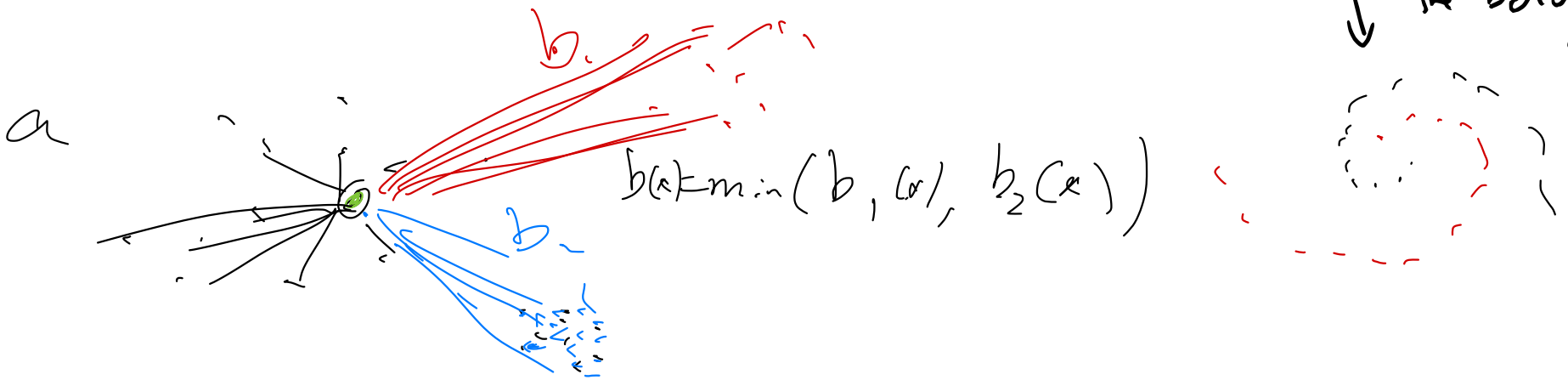
- *Cohesion* $a(x)$: average distance of x to all other vectors in the same cluster.
- *Separation* $b(x)$: average distance of x to the vectors in other clusters. Find the minimum among the clusters.
- *silhouette* $s(x)$:

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

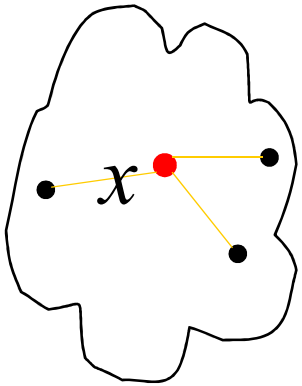
$S = \overline{s(x)}$ (avg. of $s(x)$)

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

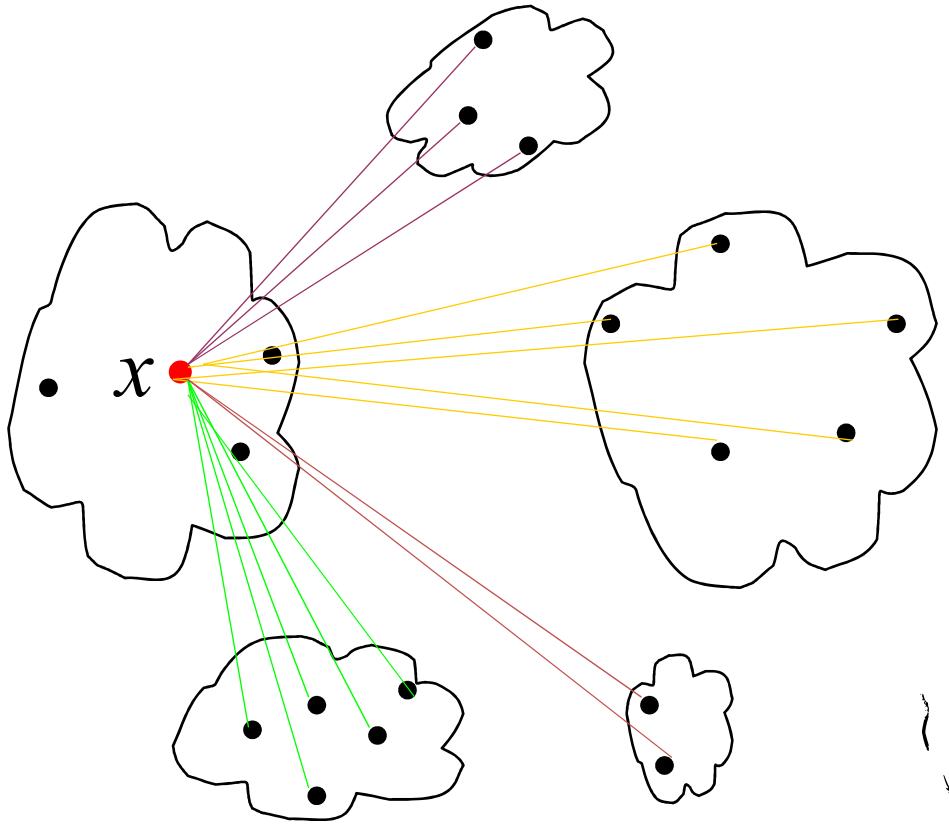
neither works → Silhouette F-ratio for clusters like the below one.



Silhouette coefficient



cohesion



separation

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

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

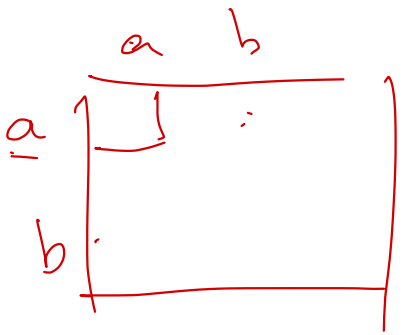
External indexes

allows us to compare clustering partition with a ground truth

- 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.



N	$\cdot$
$\cdot$	$n_{..}$

$=$

n_{11}	n_{12}	$\cdots$	n_{1l}	$n_{1.}$
n_{21}	n_{22}	$\cdots$	n_{2l}	$n_{2.}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n_{k1}	n_{k2}	$\cdots$	n_{kl}	$n_{k.}$
$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.l}$	$n_{..}$

→ NOT a confusion matrix ∵ it's not a square matrix.

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*.

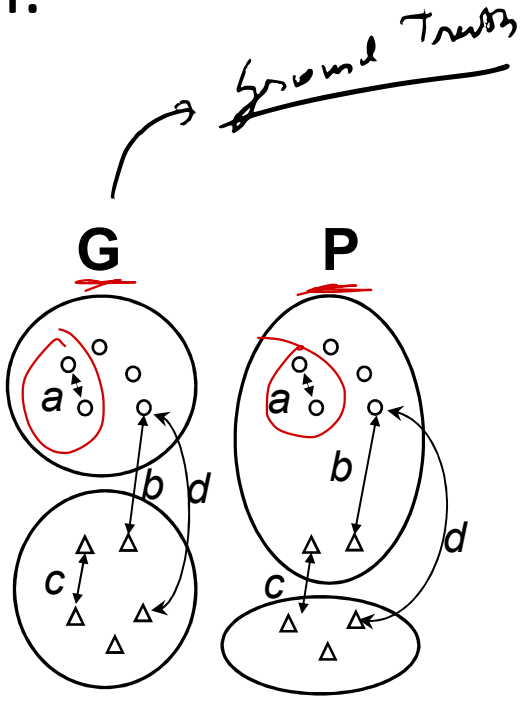
$$b = \frac{1}{2}\left(\sum_{j=1}^{K'} n_{.j}^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\right)$$

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

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

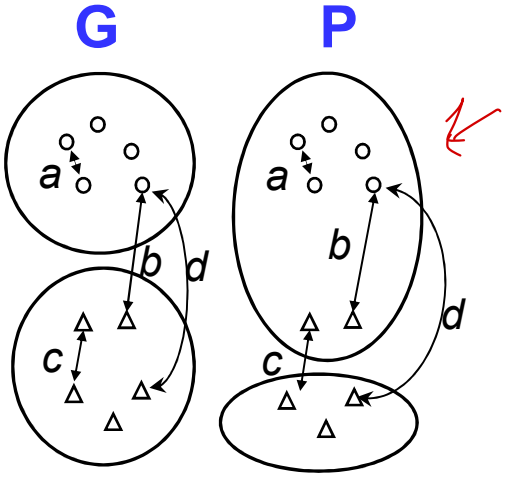
Different classes **both** in *P* and *G*.

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



Rand and Adjusted Rand index (ARI)

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



Agreement: a, d

Disagreement: b, c

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

$\frac{N^2 - N}{2}$

Corrects for taking into account that clusters are assigned randomly.

$$ARI = \frac{RI - E(RI)}{1 - E(RI)} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i\bullet}}{2}\sum_j \binom{n_{\bullet j}}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i\bullet}}{2} + \sum_j \binom{n_{\bullet j}}{2}\right] - \left[\sum_i \binom{n_{i\bullet}}{2}\sum_j \binom{n_{\bullet j}}{2}\right] \Big/ \binom{n}{2}}$$

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**

→ Find validation method :-

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:

another way of normalization.

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

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

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

(divide M.I. by the entropy)

→ Feature Selection

→ Clustering methods

→ Validation