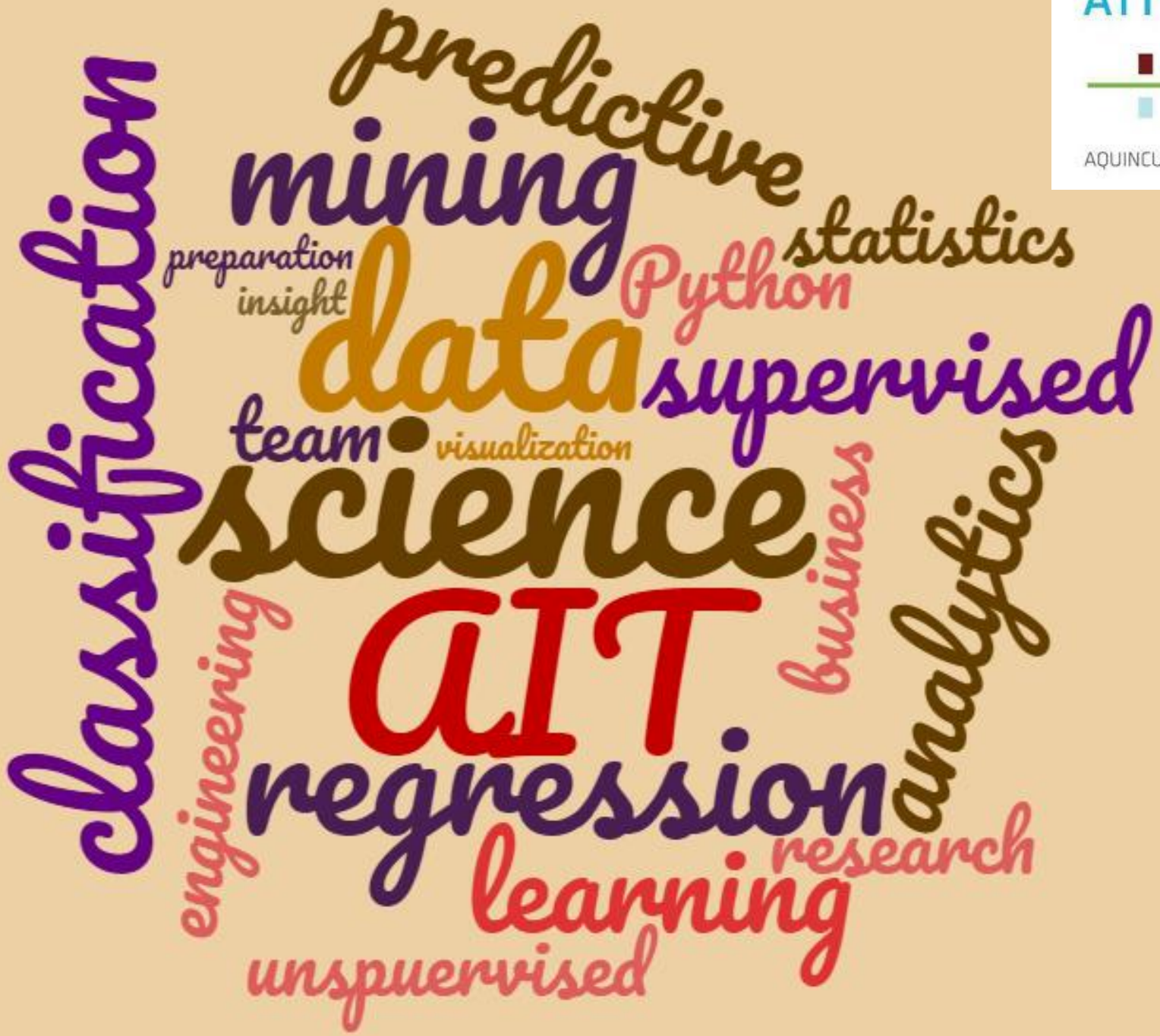
Data Science

May 5, 2020.
Clustering II.



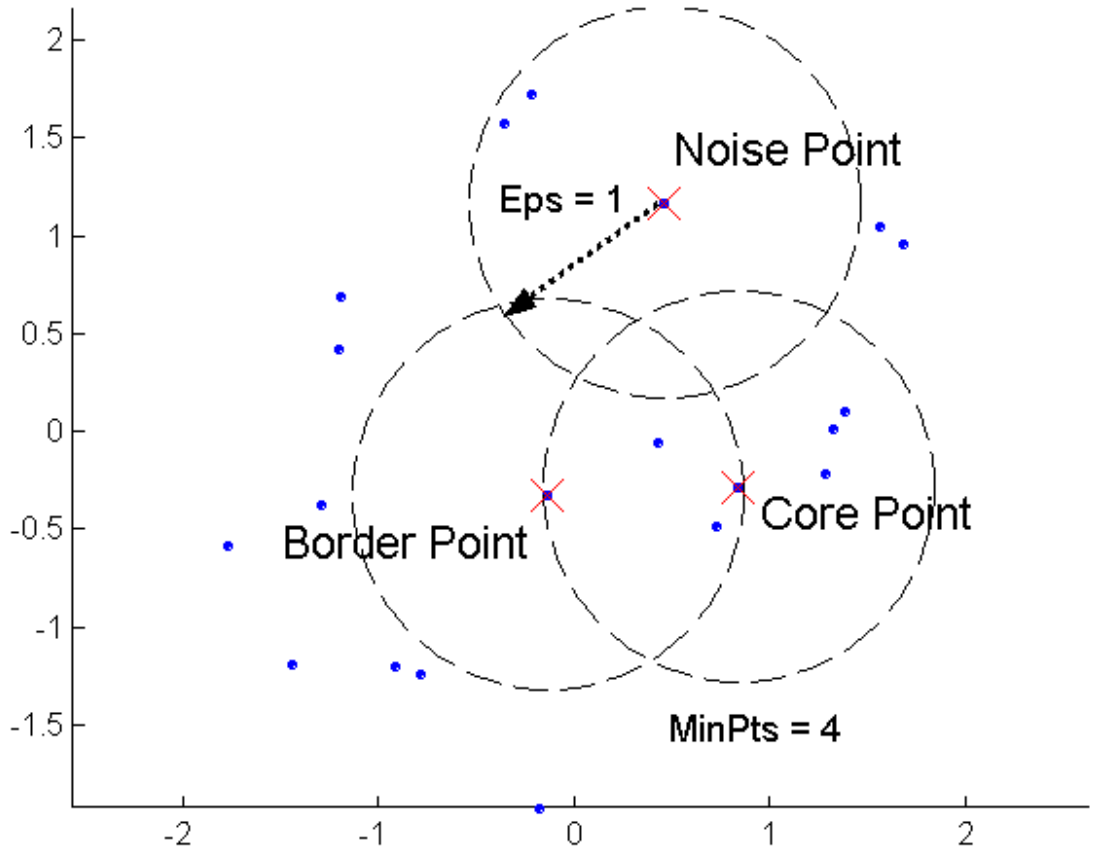
AIT-BUDAPEST
AQUINCUM INSTITUTE OF TECHNOLOGY

Roland Molontay

DBSCAN algorithm

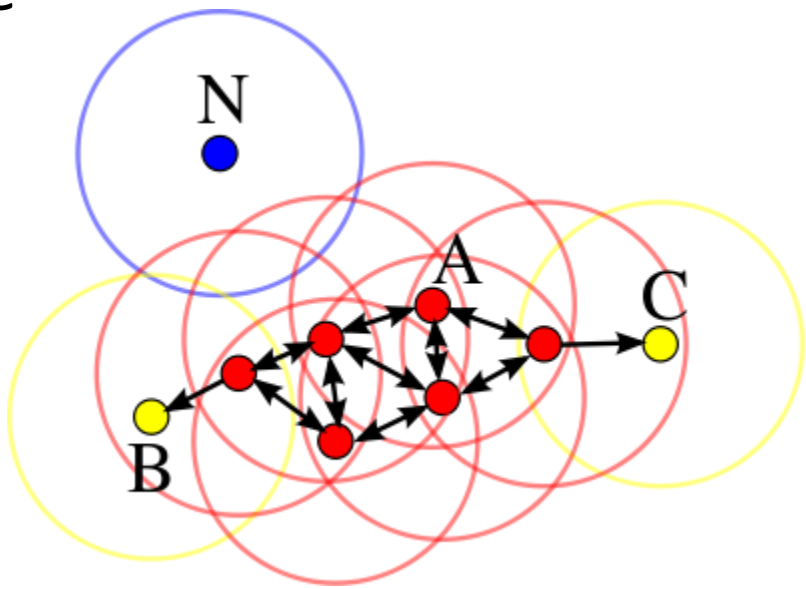
- DBSCAN: **D**ensity-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise
  - Density: number of data points within a certain radius (*Eps*)
  - **Core points**: data points that have at least *MinPts* points within distance *Eps* of them (in their *Eps*-neighborhood)
    - The core points form the interior of the clusters
  - **Border points**: data points that have fewer than *MinPts* points in their *Eps*-neighborhood, but they themselves are in the *Eps*-neighborhood of a core point
    - The border points form the borders of the clusters
  - **Noise points (outliers)**: data points that are neither core points nor border points
    - Noise points are not clustered

Core points, border points, noise points



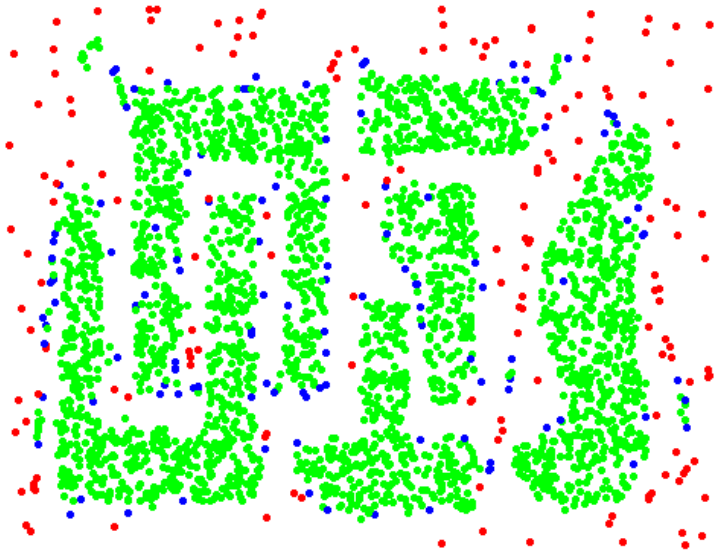
DBSCAN algorithm

- Group the data points: core points, border points, noise points
- Noise points won't be clustered (they are ignored)
- A point *q* is said to be **reachable** from *p* if there exists a path $p_1, p_2, \ldots, p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is in the *Eps*-neighborhood of core point p_i
  - All points on the path must be core points, with the possible exception of the last point
- If *p* is a core point, then it forms a **cluster** with all points that are reachable from it



DBSCAN – types of data points

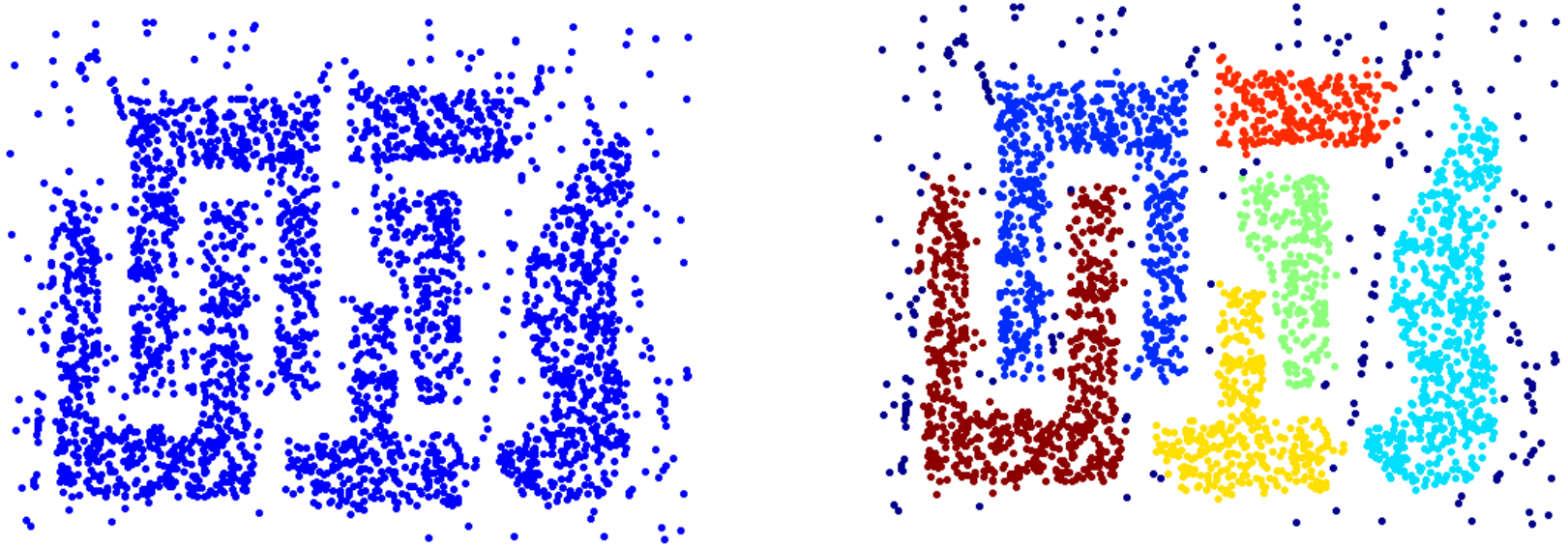
The data



Types of points: core, border and noise

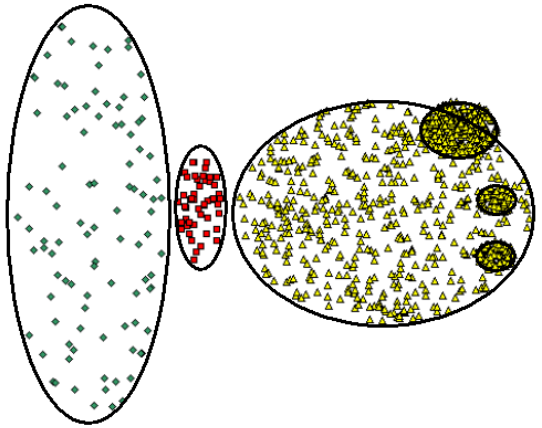
Eps = 10, MinPts = 4

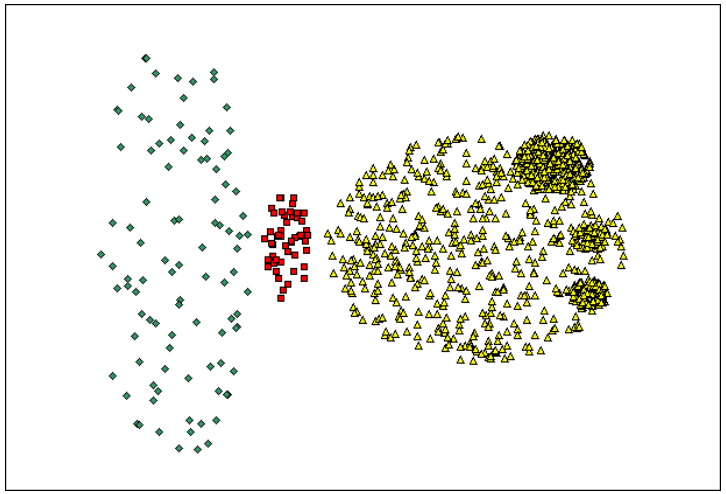
DBSCAN - example



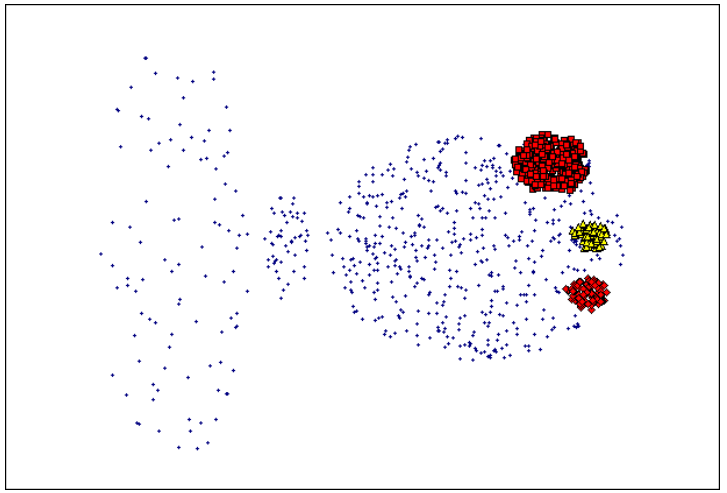
Clusters

DBSCAN - example





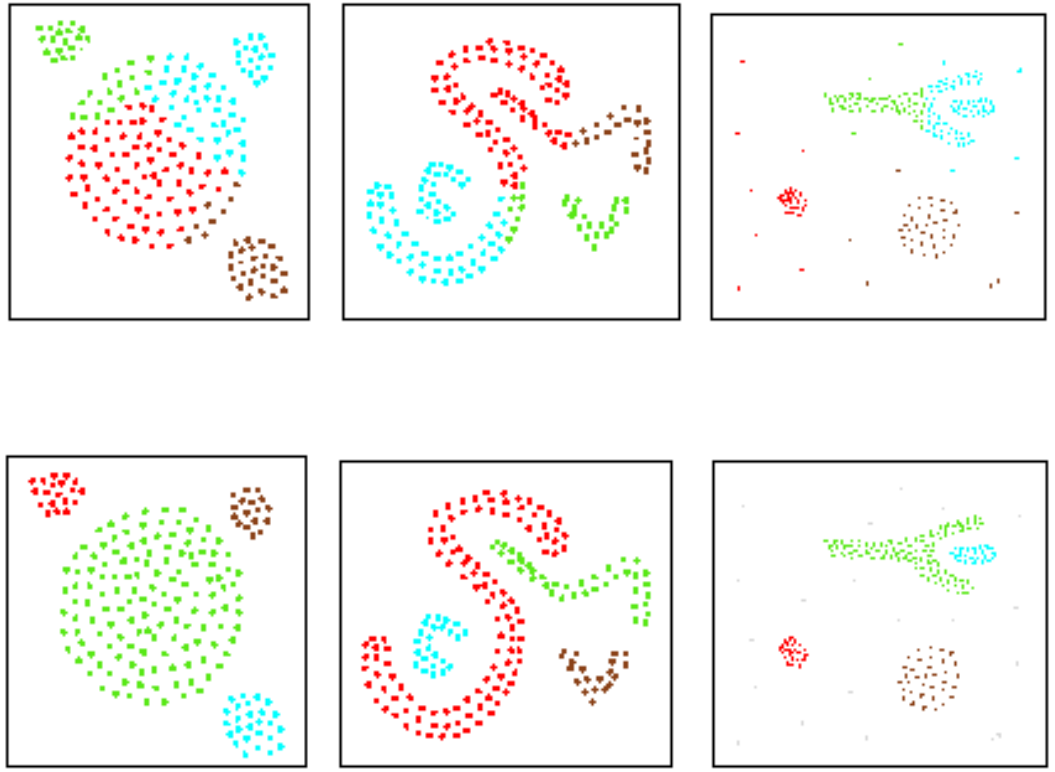
(MinPts=4, Eps=9.75)



(MinPts=4, Eps=9.62)

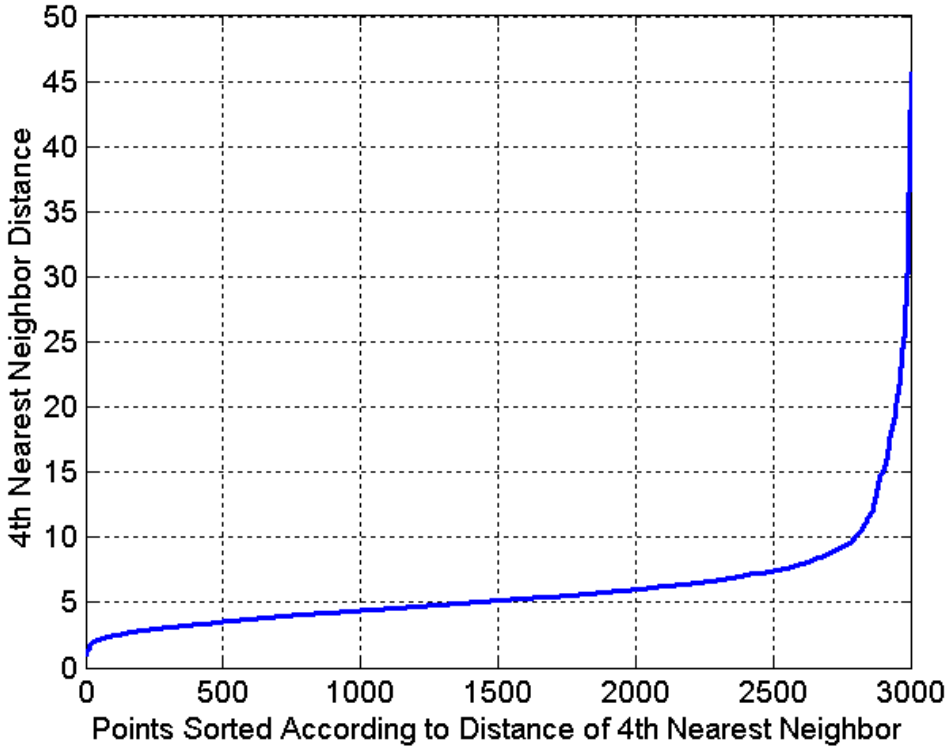
Evaluation of DBSCAN

- Insensitive to noise
- Not all the data points are clustered
- It can automatically handle outliers (by ignoring them)
- Treats clusters with differing size and shapes well
- Can’t handle clusters with differing density
- Sensitive for the choice of hyperparameters (*MinPts*, *Eps*)



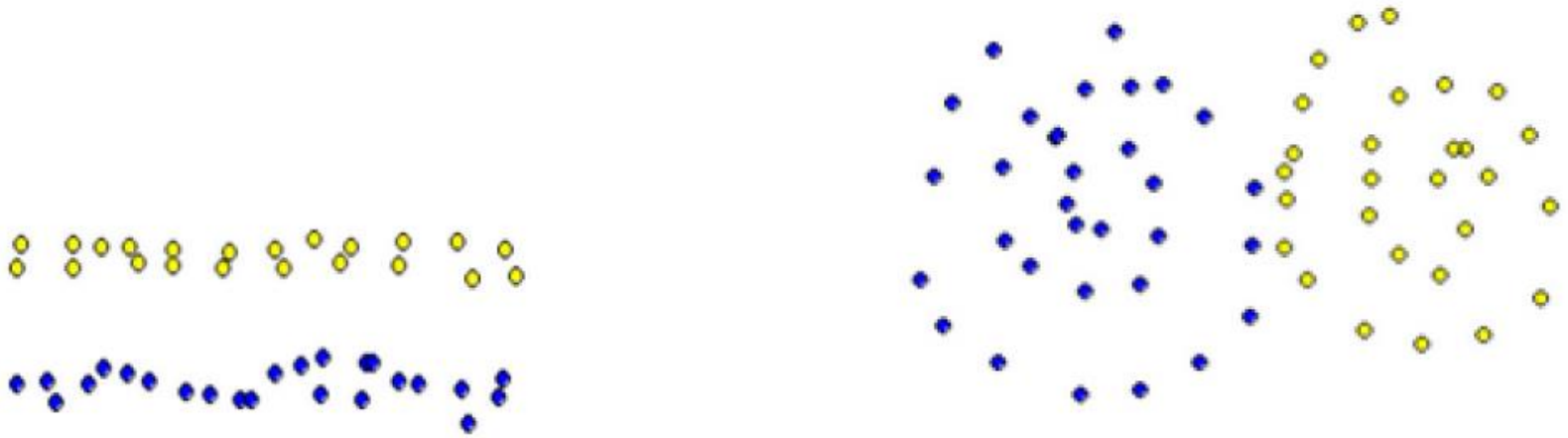
Determining the hyperparameters

- Plot the distance of the *k*th nearest neighbor for the data points
  - Idea: in dense areas the *k*th nearest neighbors are almost constants but the *k*th nearest neighbor of a noise point is much further away
  - Can we observe an angle in the graph?
    - The corresponding distance value is a reasonable choice for *Eps*



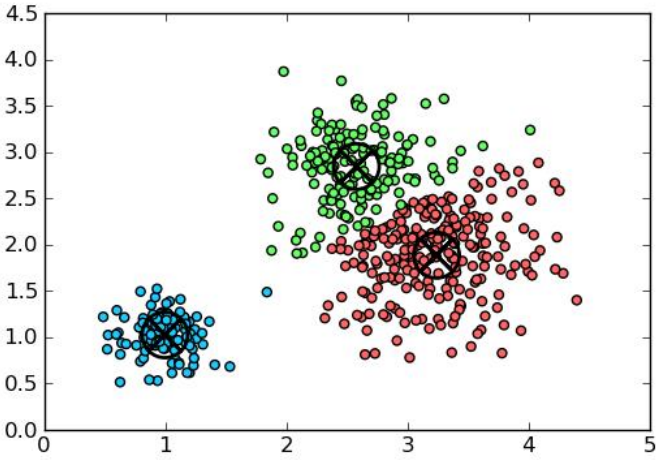
Problem

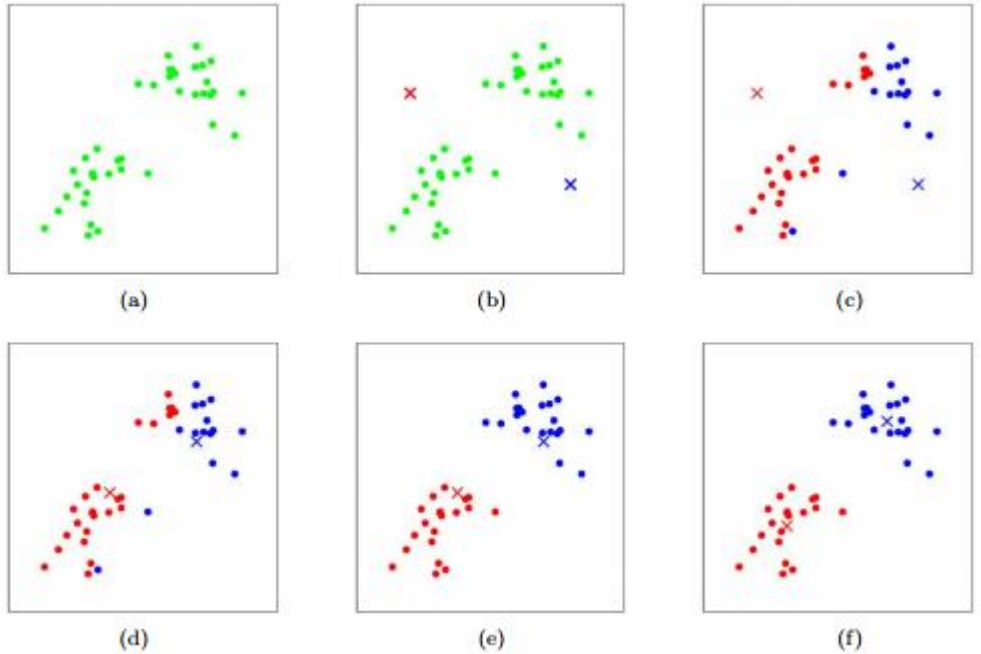
Which clustering algorithm would you use if the goal was to find the two natural clusters (marked by blue and yellow colors)? Consider the following algorithms: k-means, hierarchical clustering (both single and complete linkage) and DBSCAN.



Revisiting *K*-means

- A variant: K-medoid algorithm
- Applying *K*-means for image compression





K-means for image compression

- Pixels: three-dimensional vectors (RGB color codes)
  - Let's cluster the pixels
  - Pixels are substituted with the centroid of their corresponding cluster

2 means

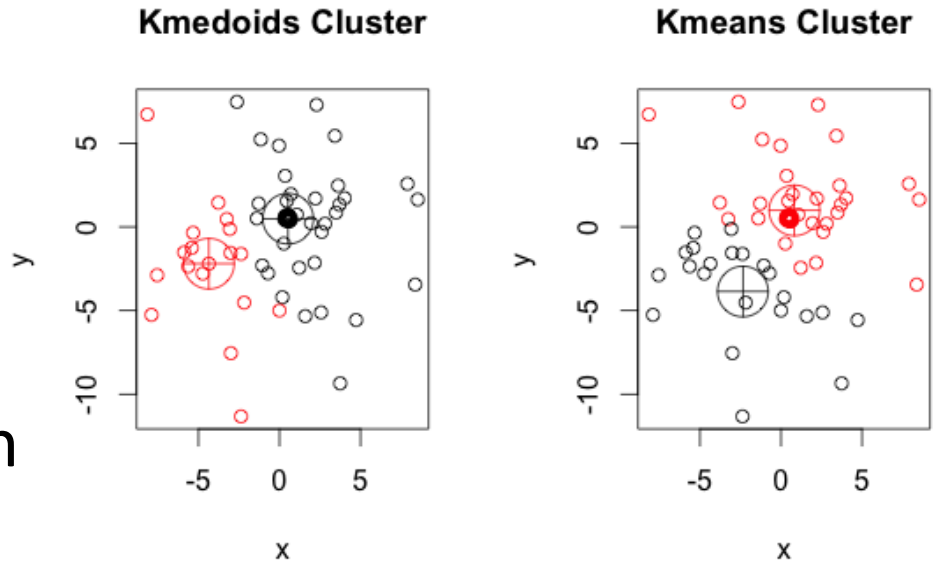
4 means

8 means

16 means

K-medoids

- Very similar to K-means algorithm but in contrast to K-means, K-medoids choose data points (medoids) as centers
  - Medoid: The representative object of the cluster whose average dissimilarity to all the objects in the cluster is minimal
- An advantage that it can be used with any arbitrary distance defined between data points
  - E.g. it can be used for discrete attributes as well

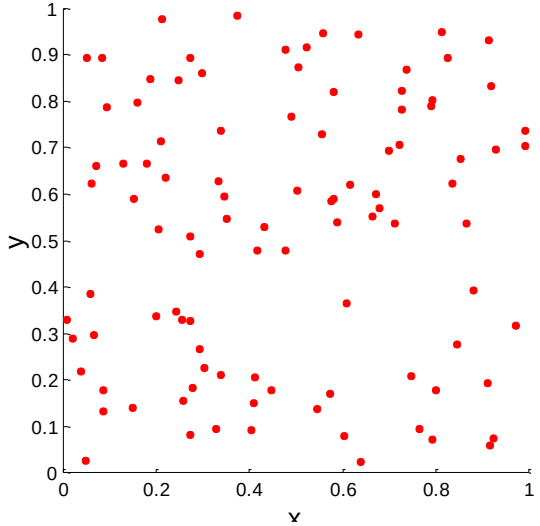


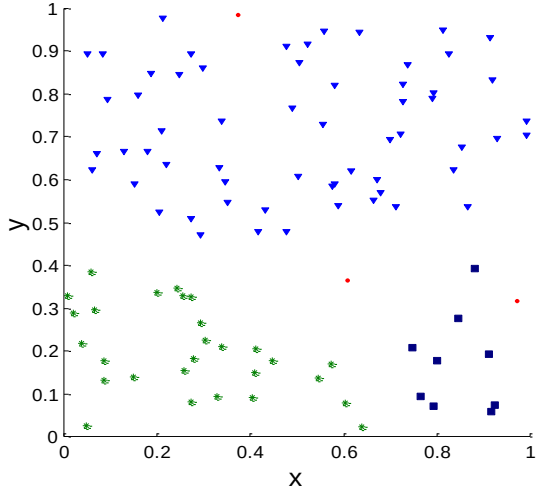
Validation of clustering

- The aim is to quantify how good a clustering is
- We aim to compare two clustering results
- Are there „natural” clusters in the data? Or does the algorithm just find clusters that are not really present in the data?
- This is a challenging task since no ground truth is known (no true labels) – unsupervised learning

Clusters in randomly generated data

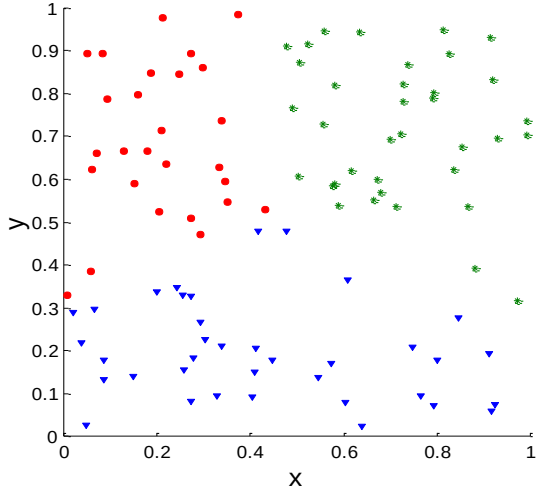
Data points generated uniformly at random on the unit square

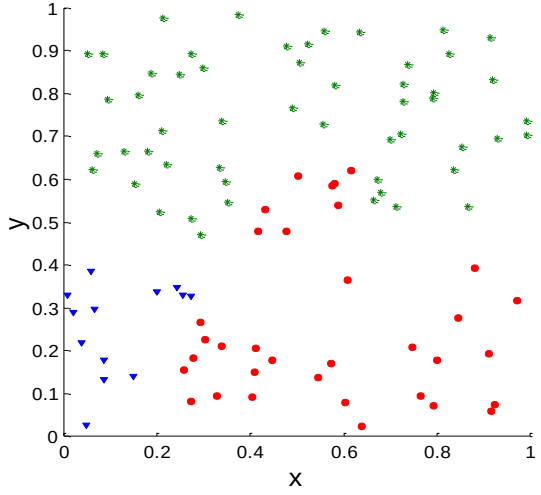




DBSCAN

K-means (K=3)

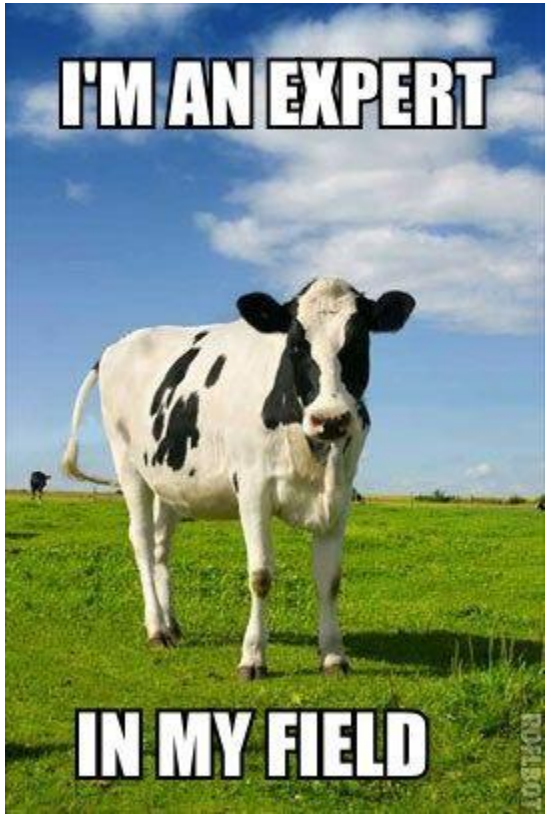




Hierarchical (complete linkage)

Validation using expert knowledge

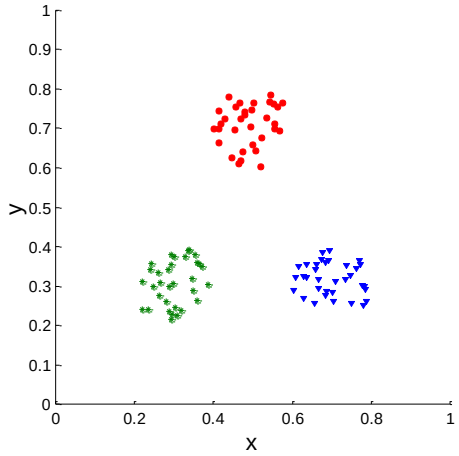
- We make the problem supervised by adding labels to the records with the help of a domain expert
  - Then the evaluation is similar to the usual evaluation techniques of classification problems
    - Do the clusters that were given by the expert agree with the result of the clustering algorithm? To what extent?

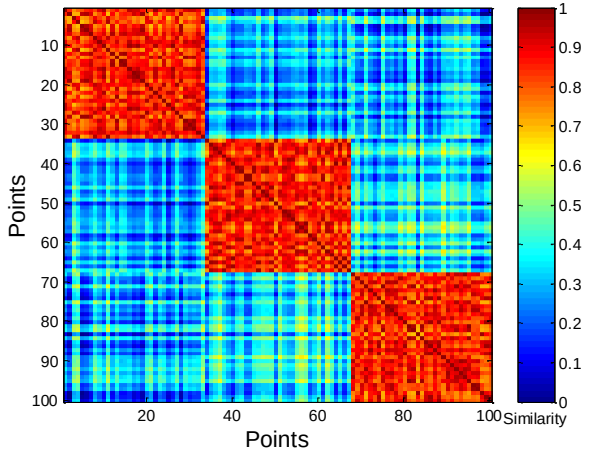


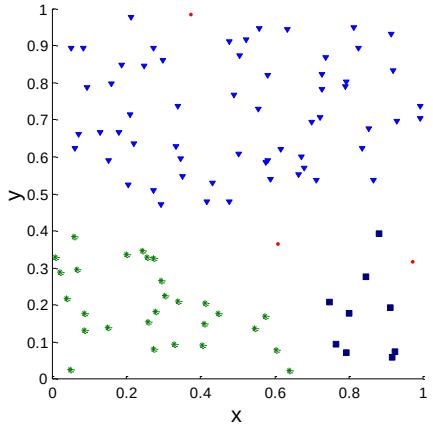
Comparing the distance matrix and the clustering

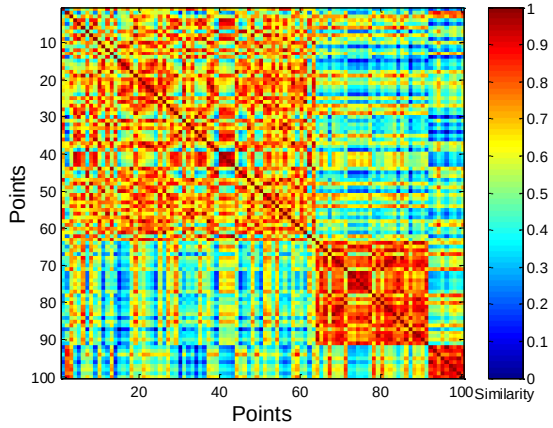
- Compare the distance matrix (or similarity matrix) with the incidence matrix
  - Incidence matrix: $A_{ij} = 1$, if the *i*th and *j*th records belong to the same cluster, and 0 otherwise
- What is the „correlation” between the two matrices?
- Visual inspection: reorder the distance matrix in such a way that the records that belong to the same clusters be next to each other
  - Does the distance matrix have a block structure?

Visual inspection of distance matrix





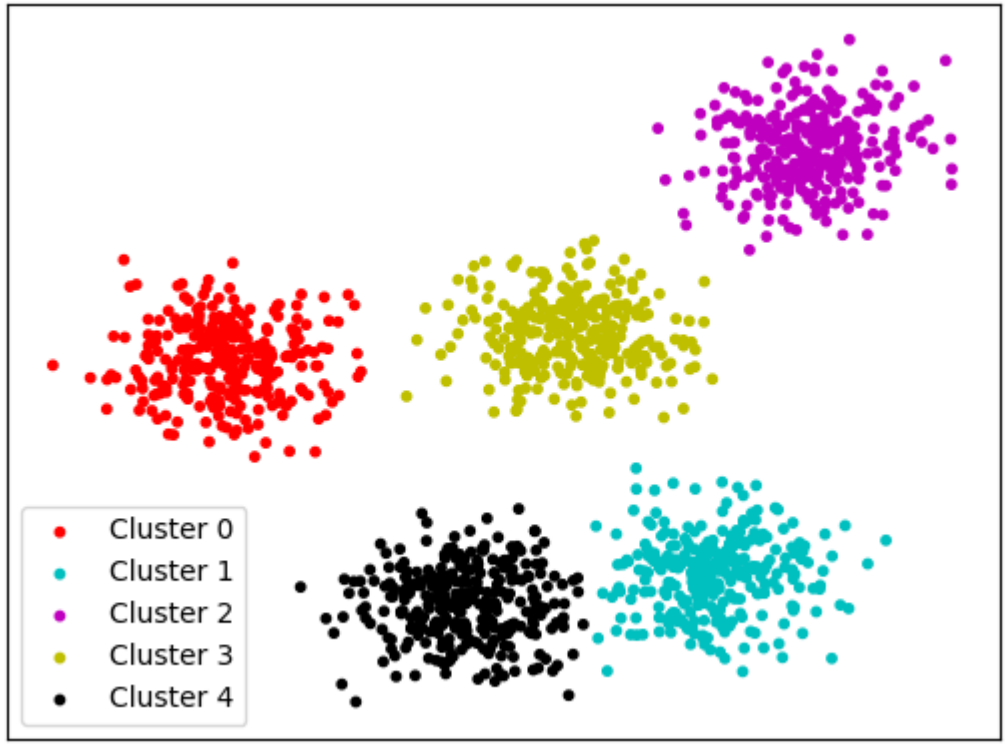




Using SSE

- We can compare two clustering outcomes with the same number of clusters
  - Which has the smaller SSE?

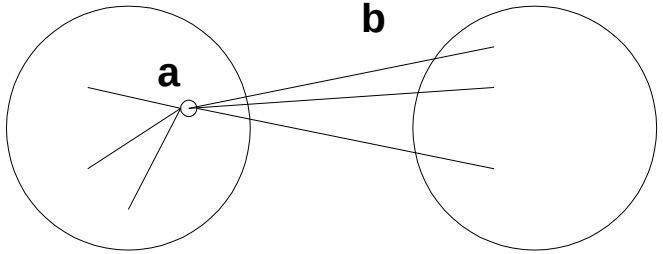
$$\text{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)^2$$



Silhouette coefficient

- Silhouette coefficient measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation)
- Let *i* be a record
  - Let ***a(i)*** be the average distance between *i* and all other data points in the same cluster
  - Let be ***b(i)*** be the smallest average distance of *i* to all points in any other cluster (of which *i* is not a member)
  - The silhouette coefficient of data point *i* :

    $s(i) = 1 - a(i)/b(i)$ if $a(i) < b(i)$ or
    $s(i) = b(i)/a(i) - 1$ if $a(i) > b(i)$ (usually that is NOT the case)



  - Usually the value is between 0 and 1
  - High value indicates that the record is well matched to its own cluster and poorly matched to neighboring clusters
- The average of *s(i)* over all points of a cluster measures how tightly grouped all points in the cluster are
- The average *s(i)* over the entire dataset measures how well the data have been clustered

The validation of clustering – closing thought

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

Acknowledgement

- András Benczúr, Róbert Pálovics, SZTAKI-AIT, DM1-2
- Krisztián Buza, MTA-BME, VISZJV68
- Bálint Daróczy, SZTAKI-BME, VISZAMA01
- Judit Csima, BME, VISZM185
- Gábor Horváth, Péter Antal, BME, VIMMD294, VIMIA313
- Lukács András, ELTE, MM1C1AB6E
- Tim Kraska, Brown University, CS195
- Dan Potter, Carsten Binnig, Eli Upfal, Brown University, CS1951A
- Erik Sudderth, Brown University, CS142
- Joe Blitzstein, Hanspeter Pfister, Verena Kaynig-Fittkau, Harvard University, CS109
- Rajan Patel, Stanford University, STAT202
- Andrew Ng, John Duchi, Stanford University, CS229