

Unsupervised Machine Learning

A Crash Course

Prof. Akka Zemmari
Univ. Bordeaux

Machine Learning



Supervised Learning



Unsupervised Learning







Machine Learning



Machine Learning



Supervised Learning

{ (, face), (, face), (, face), ... ,
(, non-face), (, non-face), (, non-face),
... }



→ face or non-face?

Machine Learning



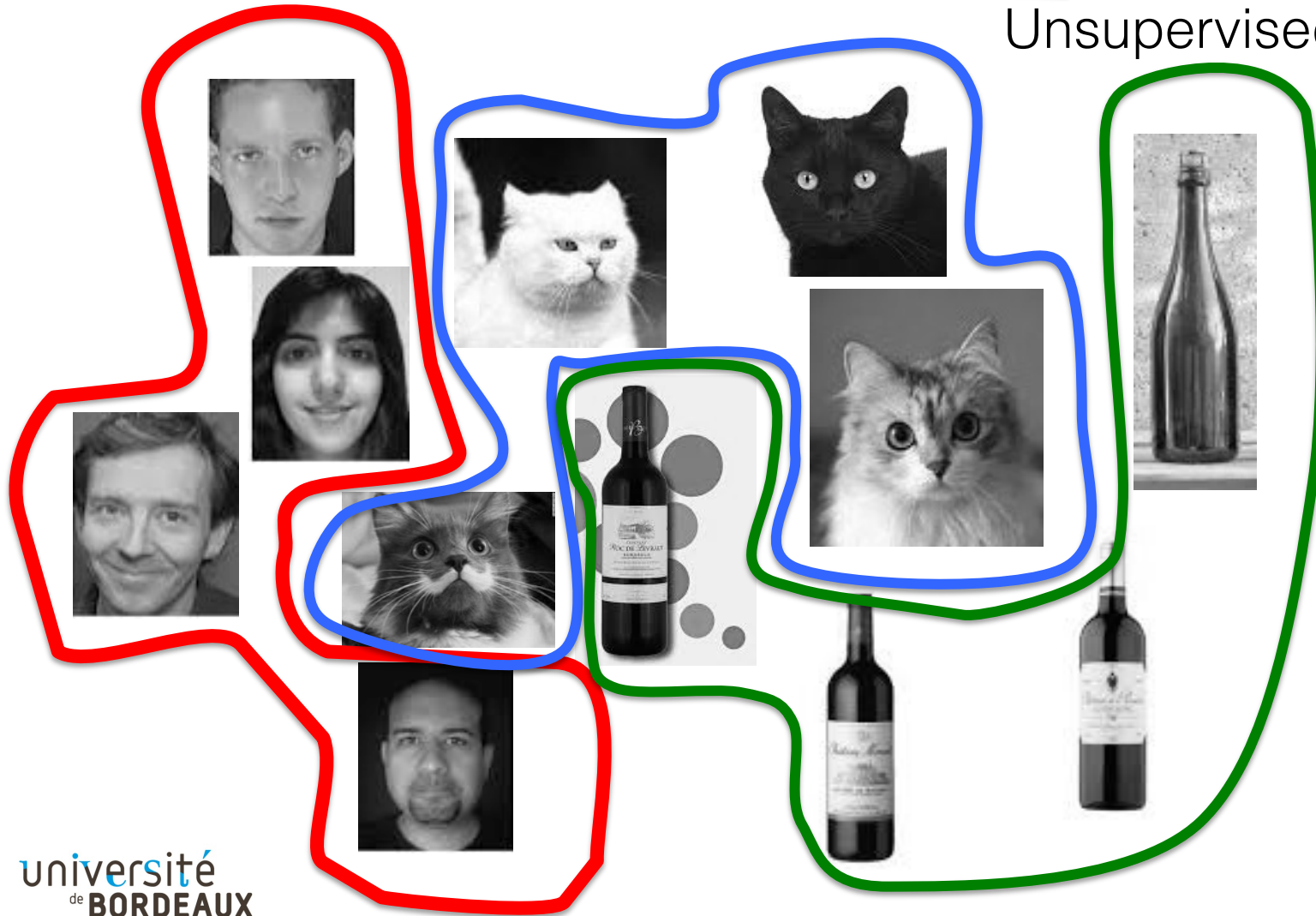
Unsupervised Learning



Machine Learning



Unsupervised Learning

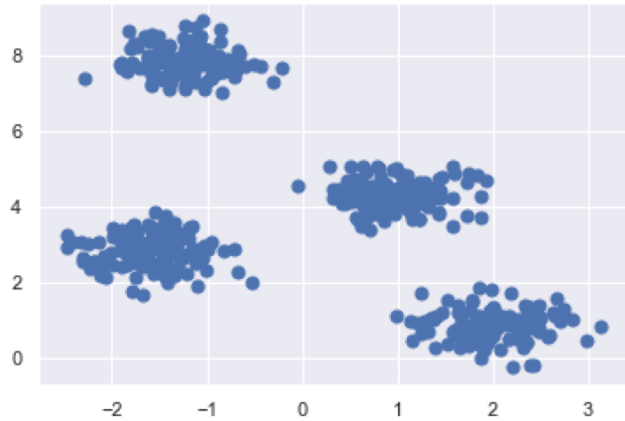


Unsupervised learning

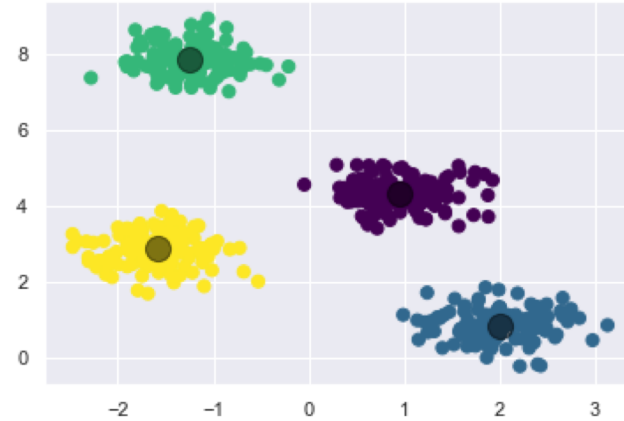
- Unsupervised learning finds hidden patterns or intrinsic structures in data.
- It is used to draw inferences from datasets consisting of input data *without* labelled responses.
- Clustering is the most common unsupervised learning technique.
- It is used for exploratory data analysis to find hidden patterns or groupings in data.
- Applications in computer vision : visual data summarization (collections of images, video)

Unsupervised learning

- Raw data ,



- 4 clusters



k-means clustering

J. MacQueen, "Some methods for classification and analysis of multivariate observations", Proc. Of the Fifth Berkley Symposium on Math. Stat. And Prob., pp. 281 – 296, 1967

- **Principle** : Unsupervised classification with a priori known number of clusters.
 - **Parameter** : the number k of clusters
 - **Input data** : a sample of M descriptor vectors x_1, \dots, x_M
- 1. Chose k initial centers c_1, \dots, c_k
- 2. For each of M vectors, assign it to the i -th cluster the center c_i of which is closest in the sense of chosen metrics
- 3. If none vector changes its class then stop.
- 4. Compute new centers: for each i , c_i is the mean of vectors of the class i
- 5. Go to 2

k-means clustering

DEMO

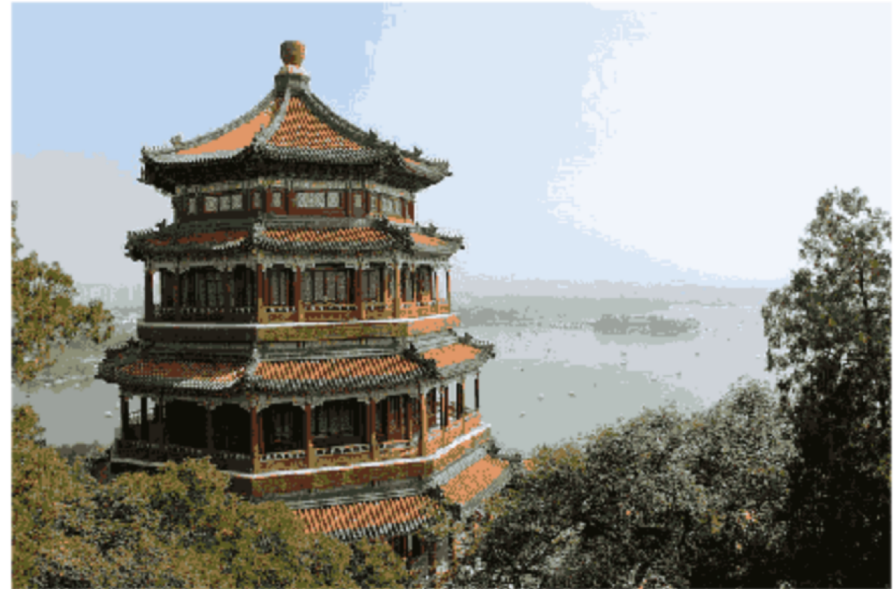
k-means clustering

Application: Color Compression

Original Image



16-color Image



DEMO

Hierarchical agglomerative clustering (HAG)

- Principle:

1. At the initialisation, each data point in the data sample forms a class
2. While the number of clusters is larger than k (limit $k=1$)
 - Groupe classes in the sense of a distance d - Distance between clusters

- Distances:

- Max-link

$$d_{max}(C_i, C_j) = \max \{d(x, y); x \in C_i, y \in C_j\}$$

- Min-link

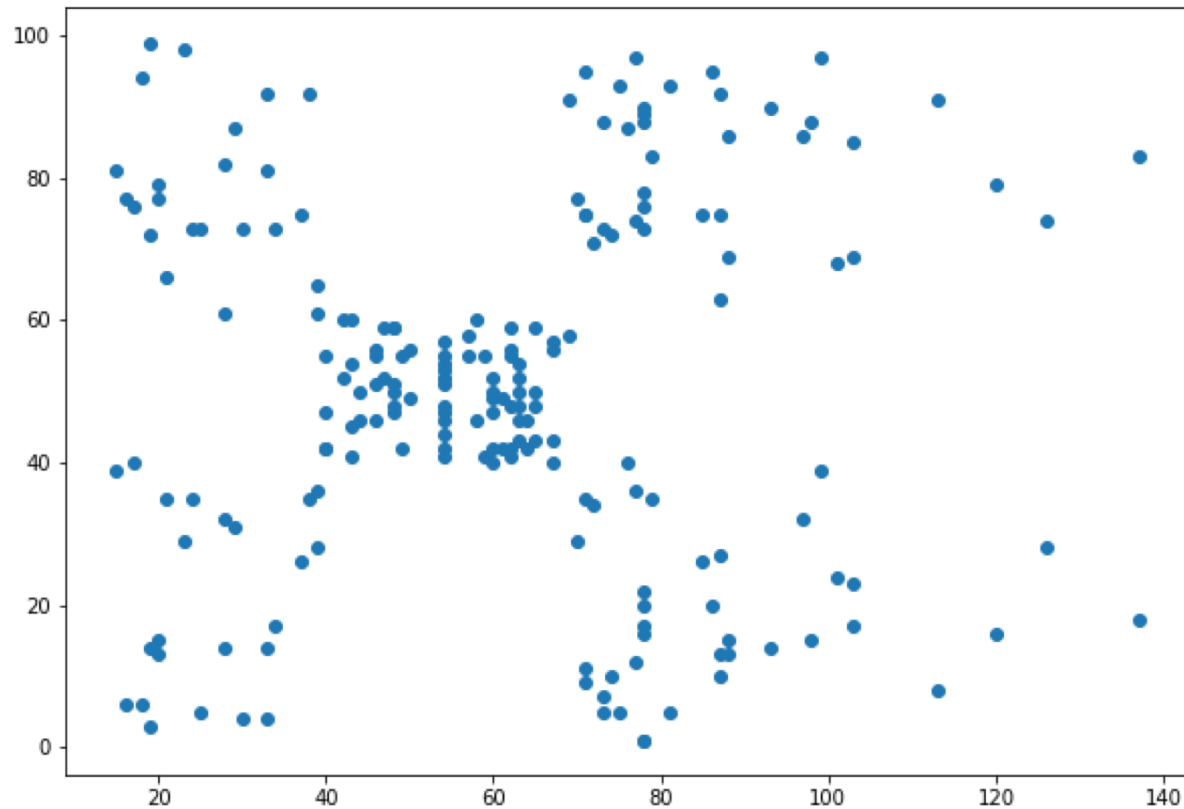
$$d_{min}(C_i, C_j) = \min \{d(x, y); x \in C_i, y \in C_j\}$$

- Mean-link

$$d_{mean}(C_i, C_j) = \frac{1}{n_i \times n_j} \sum_{l=1}^{n_i} \sum_{m=1}^{n_j} d(x_l, y_m)$$

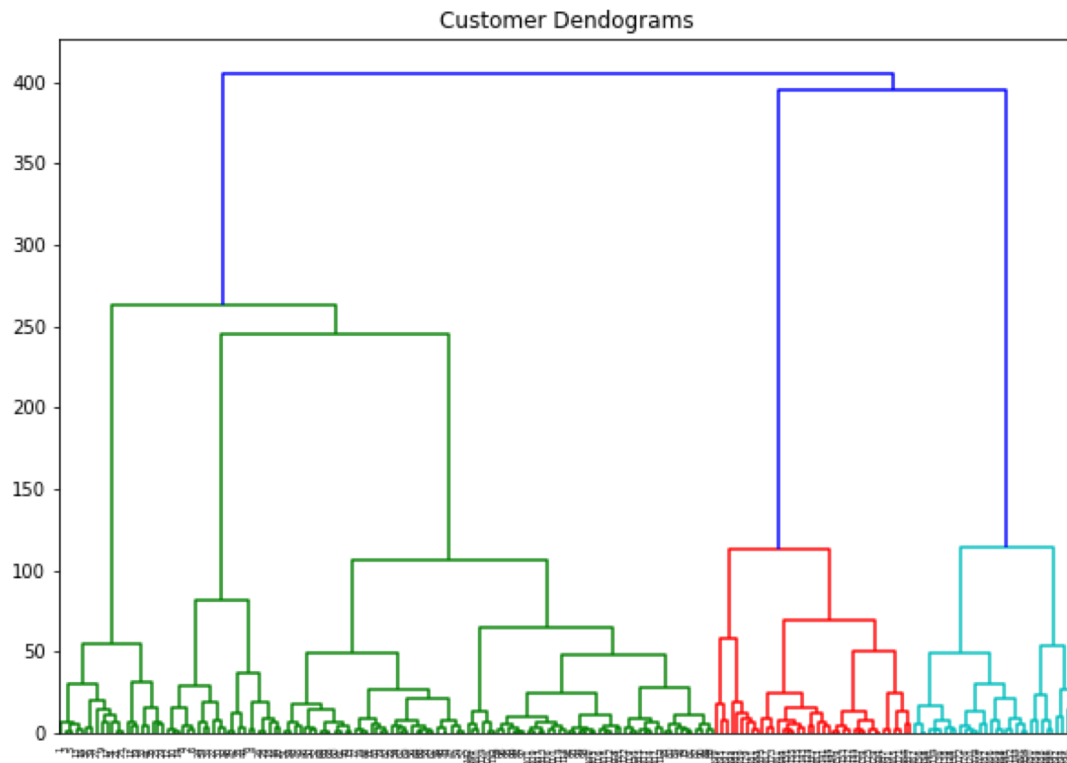
Hierarchical agglomerative clustering (HAG)

- Example:
Customer(Annual Income (k\$)', 'Spending Score (1-100))



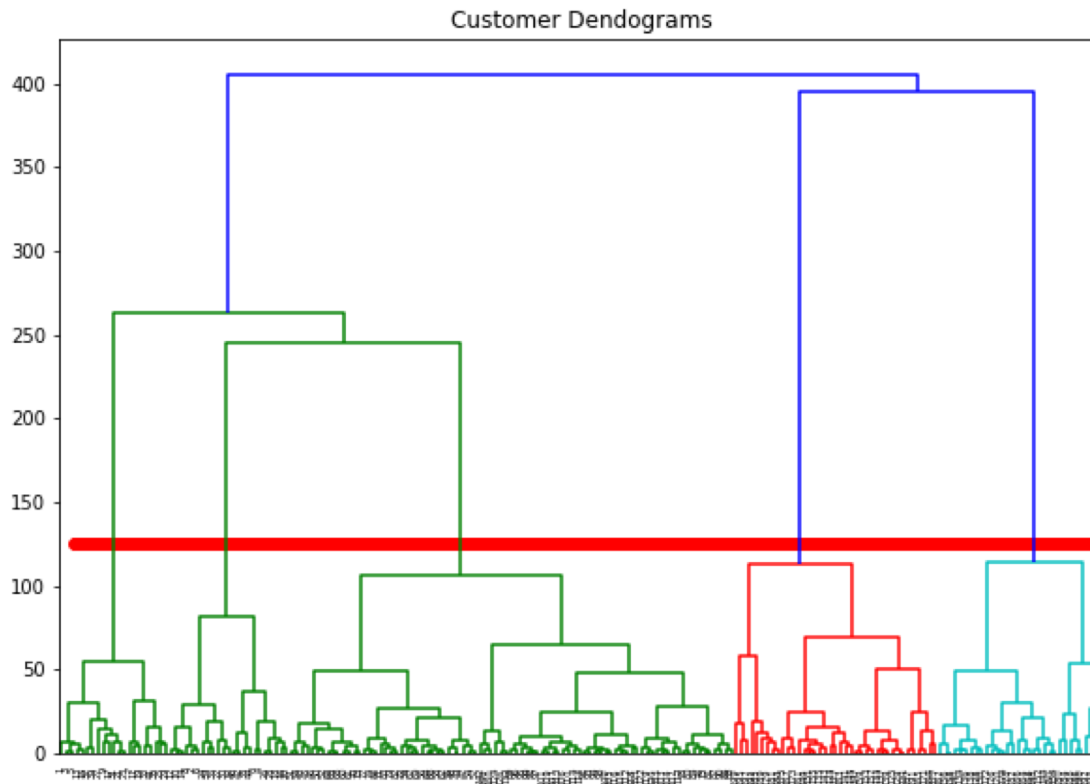
Dendrogramm

- Each data point is a cluster, Euclidean Distance



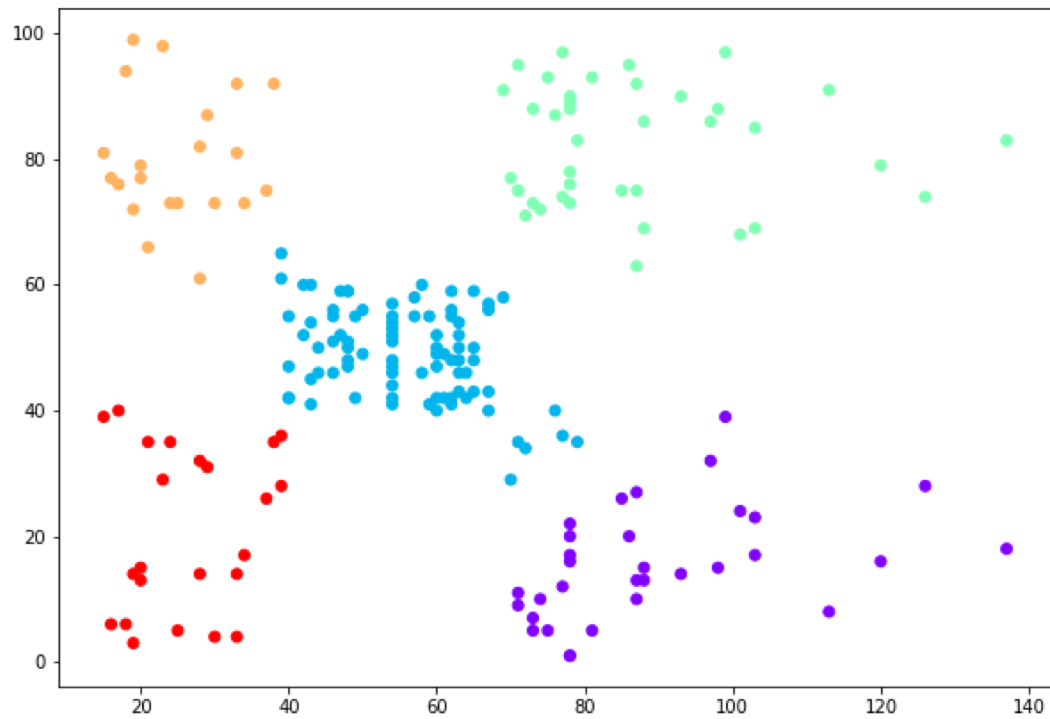
Dendrogramm

- Each data point is a cluster, Euclidean Distance



Dendrogramm

Obtained Partition:



Hierarchical agglomerative clustering (HAG)

DEMO

Clustering DB Scan

Ester, M. et al. A density- based algorithm for discovering clusters in large spatial databases with noise. (KDD-96).

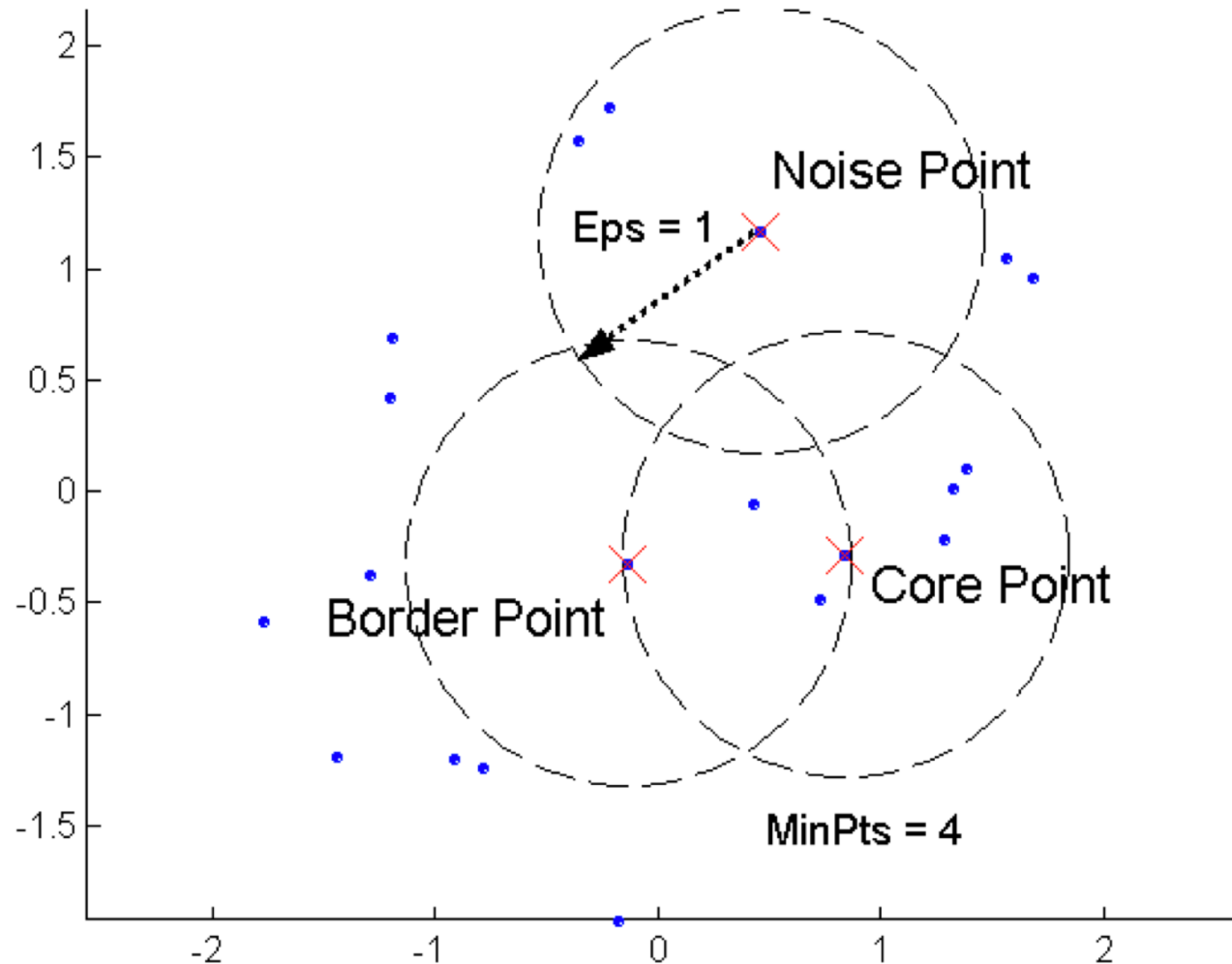
- **Density-based Spatial Clustering of Applications with Noise (DBSCAN)**
- **Principle:**
 - density – based clustering non-parametric algorithm
 - given a set of points in some space, it groups together points that are closely packed together
 - **Notion of outliers:** the points which are « far » from others, isolated, will be considered as noise

Clustering DB Scan

- DBSCAN is a density-based algorithm.
 - Density = number of points within a specified radius ε (Eps)
 - 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: Core, Border, and Noise points

(<https://csc.csudh.edu/btang/seminar/slides/DBSCAN.pdf>)



DBSCAN

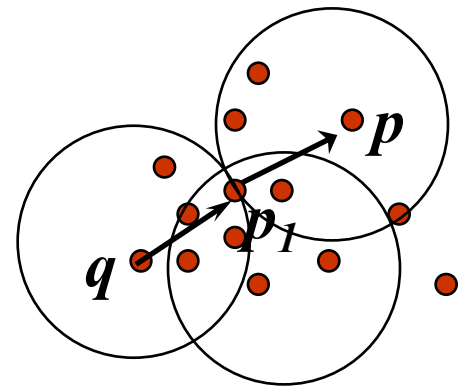
- Two parameters (Eps and MinPts):
 - Eps: Maximum radius of the neighbourhood
 - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point

$$N_{Eps}(p) = \{q \mid \mathit{dist}(p, q) \leq Eps\}$$

- **Directly density-reachable**: A point p is directly density-reachable from a point q wrt. ε , *MinPts* if
 1. p belongs to $N_{Eps}(q)$
 2. $|N_{Eps}(q)| \geq \mathit{MinPts}$, i.e., q is a core point.

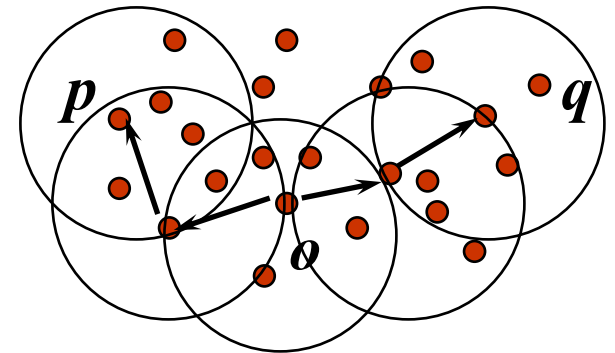
DBSCAN

- A point p is **density reachable** from a point q with respect to Eps and $MinPts$, if
 - $\exists p_1, p_2, \dots, p_n$, where $p_1 = q$, $p_n = p$ and p_{i+1} is directly density reachable from p_i



DBSCAN

- A point p is **density-connected** to a point q if there is a point o such that both, p and q are density-reachable from o .

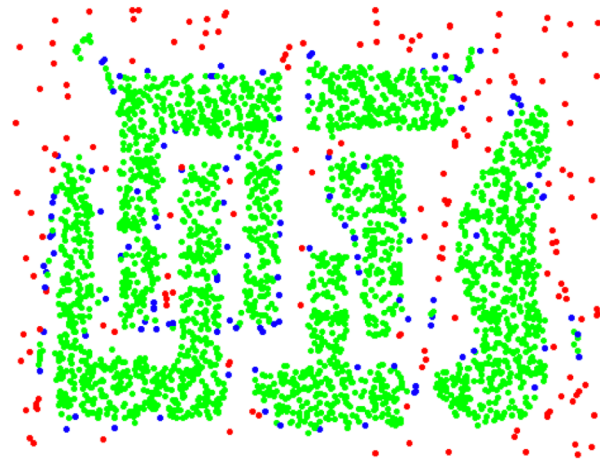


DBSCAN

- Large Eps



Original Points



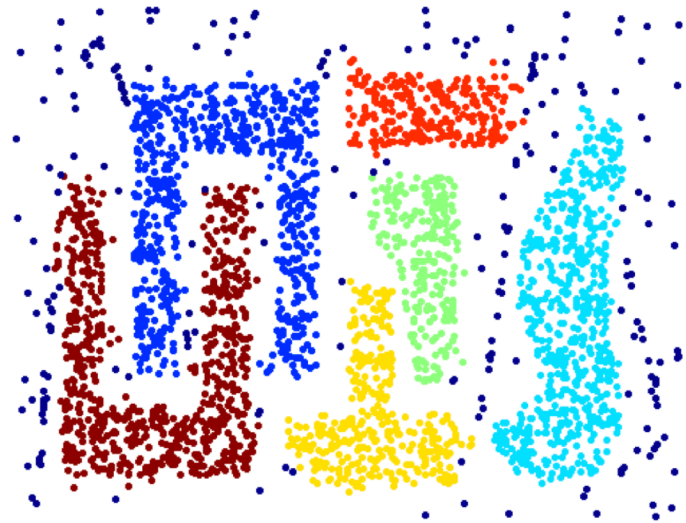
Point types: **core**,
border and **noise**

DBSCAN

- Optimal Eps



Original Points



Clusters

DBSCAN: Algorithm

Let ClusterCount=0.

For every point p :

1. If p it is not a core point, assign a null label to it [e.g., zero]
2. If p is a core point, a new cluster is formed [with label ClusterCount:= ClusterCount+1] Then find all points density-reachable from p and classify them in the cluster.
[Reassign the zero labels but not the others]

Repeat this process until all of the points have been visited.

Since all the zero labels of border points have been reassigned in 2, the remaining points with zero label are noise.

DBSCAN

DEMO