Machine Learning (DLMDSML01)

Nghia Duong-Trung[1]
[1] **German Research Center for Artificial Intelligence (DFKI GmbH)**
Internal Use @ IU International University of Applied Science, Berlin Campus

SPONSORED BY THE

Federal Ministry
of Education
and Research

Section 2: Clustering

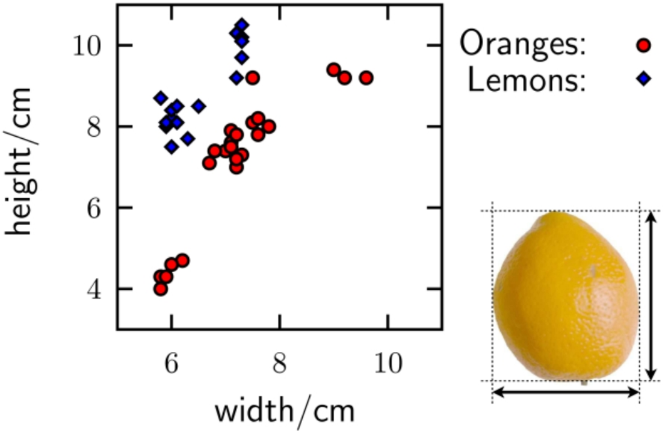
- Textbook: check it in MyCampus
- Clustering problem
- Centroid-Based Clustering
- Gaussian Mixture Models Clustering
- Hierarchical Clustering
- Density-Based Clustering

Clustering analysis

Clustering analysis

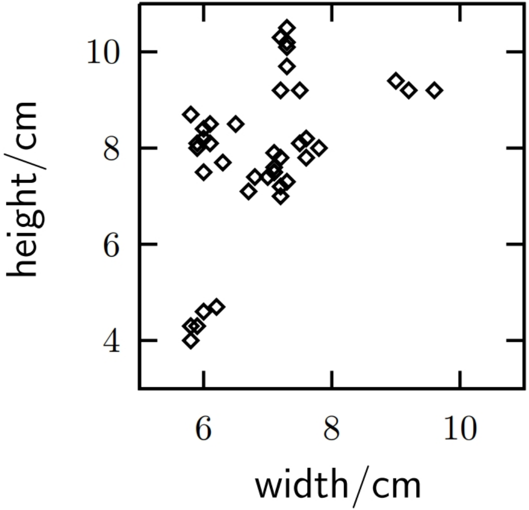
- Cluster analysis aims to partition a data set into meaningful or useful groups, based on distances between data points
- Clustering is an unsupervised process | the data items do not have class labels
- In some cases the aim of cluster analysis is to obtain greater understanding of the data, and it is hoped that the clusters capture the natural structure of the data
- In other cases cluster analysis does not necessarily add to understanding of the data, but enables it to be processed more efficiently

Clustering analysis

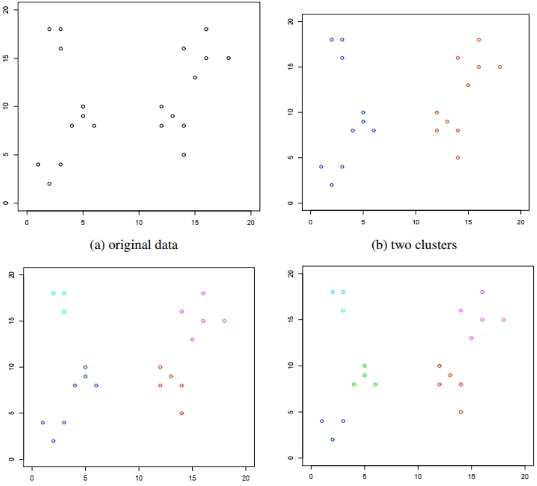


Clustering analysis

German Research Center for Artificial Intelligence



Clustering analysis



(a) original data

(b) two clusters

(c) four clusters

(d) five clusters

Clustering analysis

- There are many reasons to perform clustering. Most commonly it is done to better understand the data (data interpretation), or to efficiently code the data set (data compression)
    - Data interpretation: Automatically dividing a set of data items into groups is an important way to analyse and describe the world
    - Data compression: Clustering may be used to compress data by representing each data item in a cluster by a single cluster prototype, typically at the centre of the cluster

Type of clustering

- Hierarchical clustering
  - form a tree of nested clusters in which, at each level in the tree, a cluster is the union of its children
  - top-down clustering, bottom-up approaches
- Partitional clustering
  - devide the dataset into a fixed number of non-overlapping clusters, with each data point assigned to exactly one cluster

Distance calculation

- The similarity/dissimilarity metric that is routinely utilized in clustering analysis is a form of distance function between each pair of data records, e.g., A and B
- Therefore, the distance measures how close A and B are to each other, and a decision is made whether to combine A and B in one cluster
- Simple forms of basic distance functions between two two-dimension data points A and B
  - Euclidean distance: $d_{A,B} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$
  - Manhattan distance: $d_{A,B} = |x_A - x_B| + |y_A - y_B|$
  - where (x_A, y_A) and (x_B, y_B) are the coordinates, e.g., the features of A and B, respectively

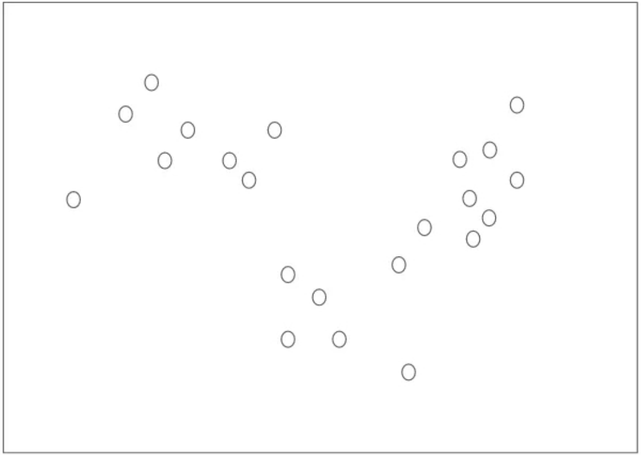
Centroid-Based Clustering

K-means clustering

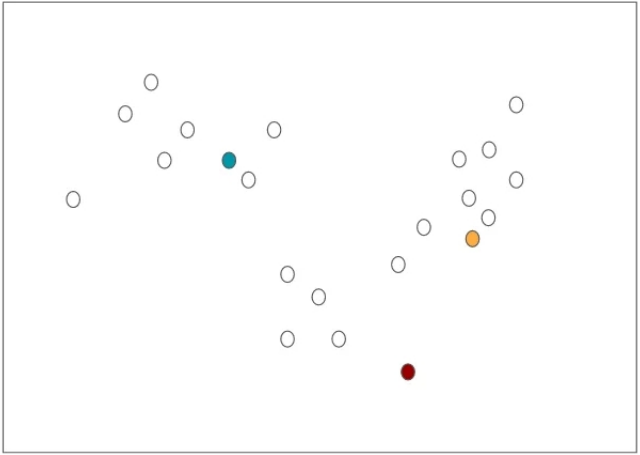
1. Pick K random points as cluster center positions
2. Assign each point to its nearest center*
3. Move each center to mean of its assigned points
4. IF centers moved, goto step 2

* In the unlikely event of a tie, break tie in some way. For example, assign to the centre with smallest index in memory.

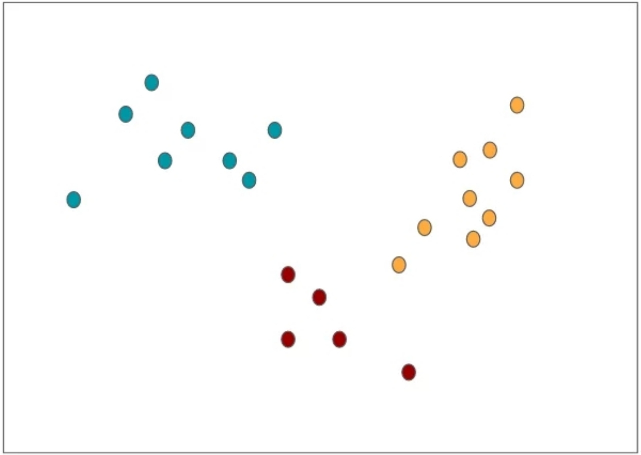
K-means clustering



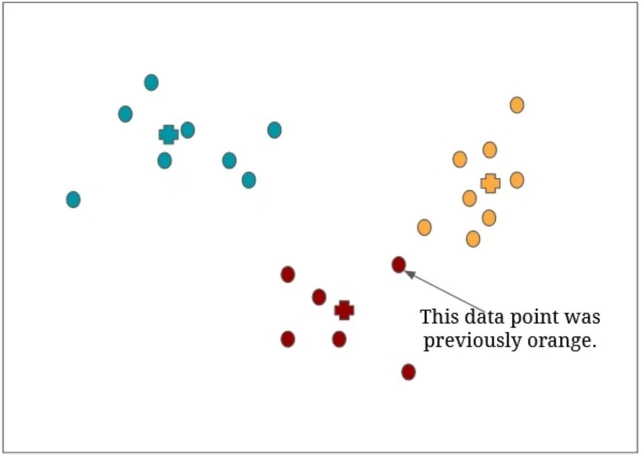
K-means clustering



K-means clustering



K-means clustering



This data point was previously orange.

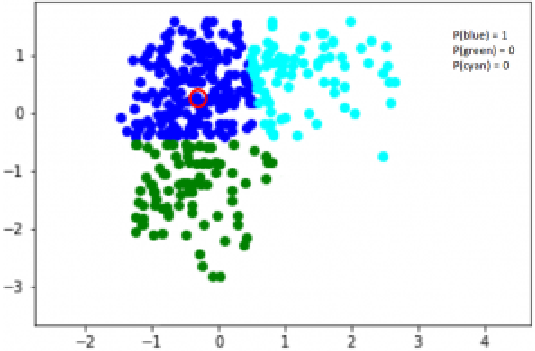
Gaussian Mixture Models Clustering

Gaussian Mixture Models (GMMs)

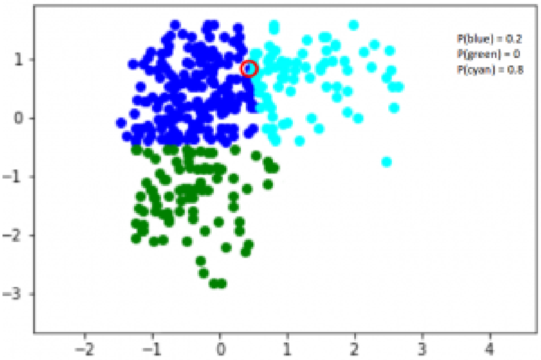
- GMMs assume that there are a certain number of Gaussian distributions, and each of these distributions represent a cluster. Hence, a Gaussian Mixture Model tends to group the data points belonging to a single distribution together.

- Let's say we have three Gaussian distributions – GD1, GD2, and GD3. These have a certain mean (μ1, μ2, μ3) and variance (σ1, σ2, σ3) value respectively. For a given set of data points, our GMM would identify the probability of each data point belonging to each of these distributions.
- probabilistic generative model
  - A data record has a probability for belonging to each cluster, and it is assigned to the cluster returning the highest probability. Thus, it is a way of performing a "soft" clustering.

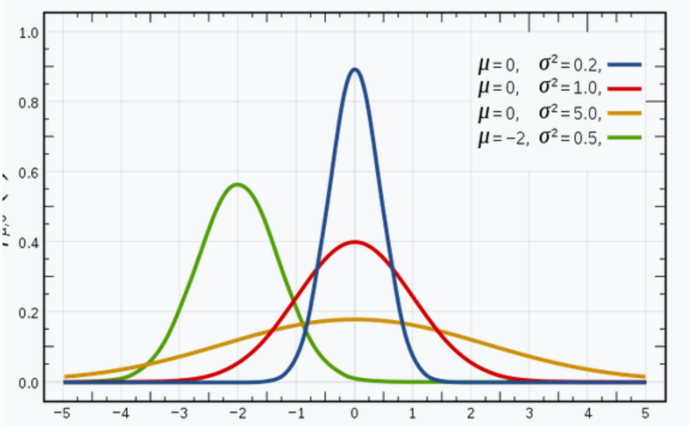
GMMs



GMMs



The Gaussian Distribution



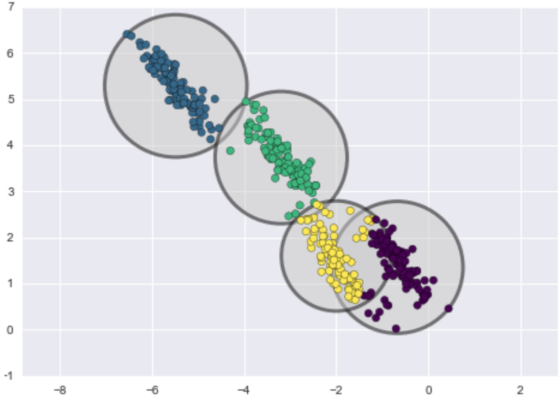
Expectation Maximization Algorithm (EM)

- The EM algorithm is implemented when there is an analytic model for the dataset (i.e., the probabilistic Gaussian mixtures model), the model's shape (i.e., Gaussian distribution) is known, but the parameters μ and σ of this model are unknown
- We typically use EM when the data has missing values, or in other words, when the data is incomplete.
    - These missing variables are called latent variables
- Since we do not have the values for the latent variables, EM tries to use the existing data to determine the optimum values for these variables and then finds the model parameters.

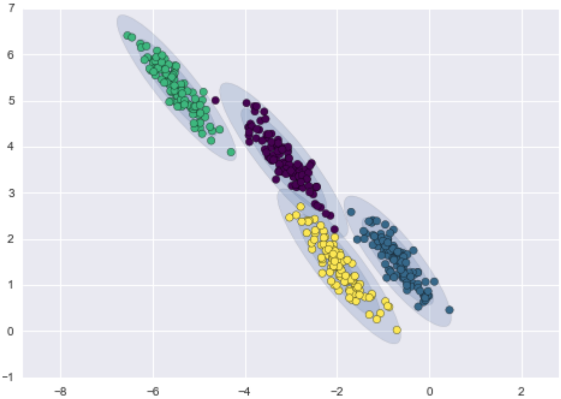
EM

- The EM algorithm has two steps:
  - E-step: In this step, the available data is used to estimate (guess) the values of the missing variables
  - M-step: Based on the estimated values generated in the E-step, the complete data is used to update the parameters

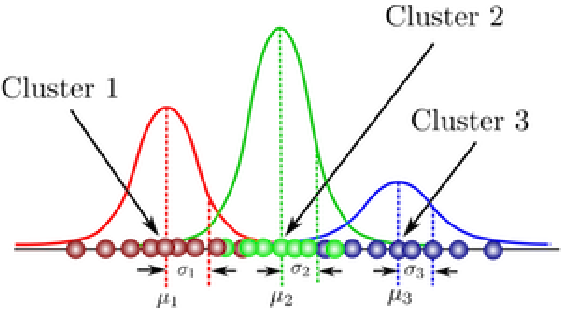
EM vs K-means



EM vs K-means



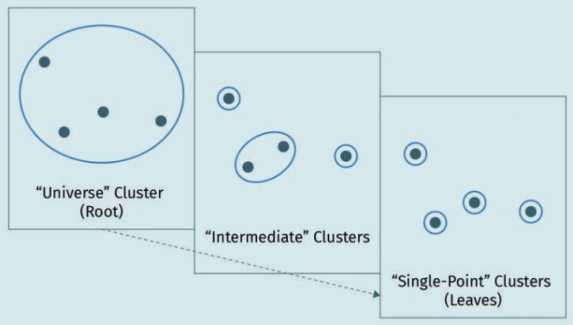
EM vs K-means


Cluster 2
Cluster 1
Cluster 3
σ_1
σ_2
σ_3
μ_1
μ_2
μ_3

Hierarchical Clustering

Hierarchical Clustering

- The clusters are built in a hierarchy. This hierarchy of clusters is represented as a tree (or dendrogram)
- The root of this tree is the “universe” cluster that includes all the data records, while the leaves form “singlepoint” clusters, where they include an individual data record for each leaf



Type of Hierarchical Clustering

- agglomerative
  - bottom-up approach that starts at the "single-point" clusters and moves up by merging similar clusters until it reaches the "universe" cluster.
- divisive
  - as a top-down approach

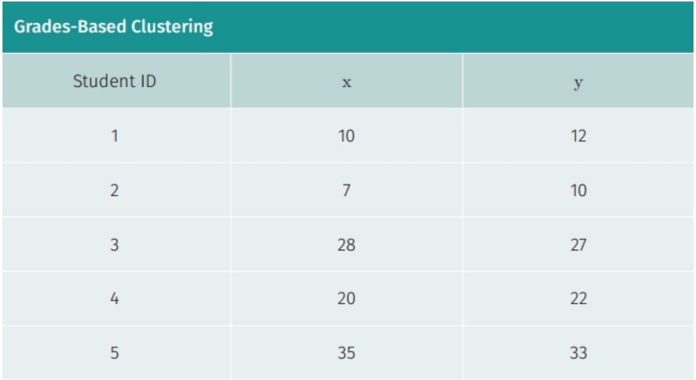
Agglomerative clustering

1. Consider each data record as a cluster (i.e., "single-point" cluster). The number of clusters is equal to n, which is the number of data records within the input dataset

2. Merge the two closest clusters into one bigger cluster. The number of clusters will become $(n - 1)$

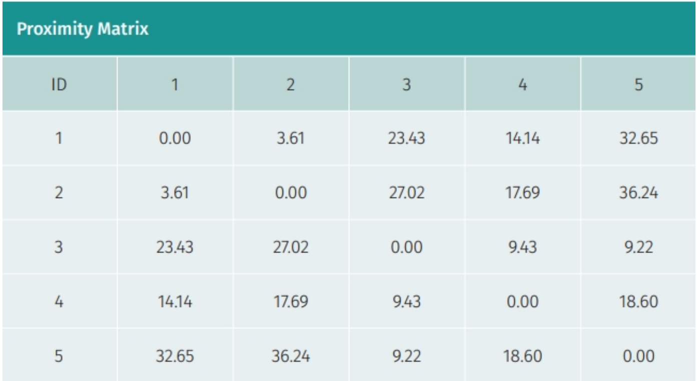
3. Repeat step two until a single cluster is formed: the "universe" cluster

4. Construct a tree (i.e., dendrogram) to visualize the progression of the formed clusters at each step



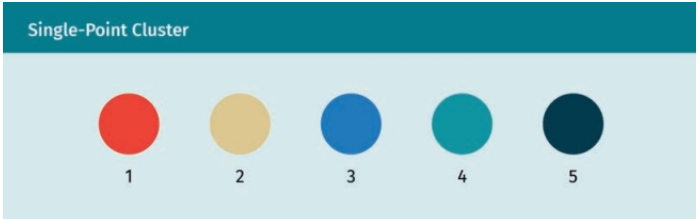
Grades-Based Clustering

Student ID	x	y
1	10	12
2	7	10
3	28	27
4	20	22
5	35	33

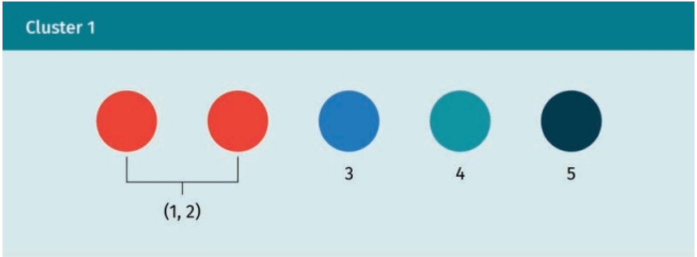


Proximity Matrix

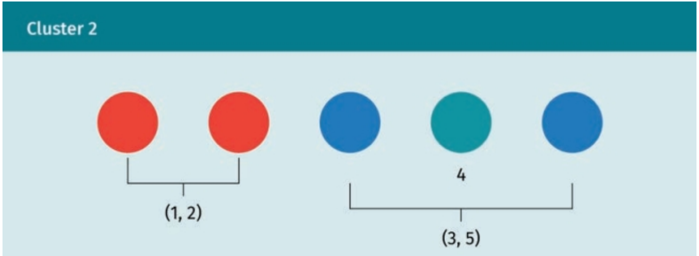
ID	1	2	3	4	5
1	0.00	3.61	23.43	14.14	32.65
2	3.61	0.00	27.02	17.69	36.24
3	23.43	27.02	0.00	9.43	9.22
4	14.14	17.69	9.43	0.00	18.60
5	32.65	36.24	9.22	18.60	0.00

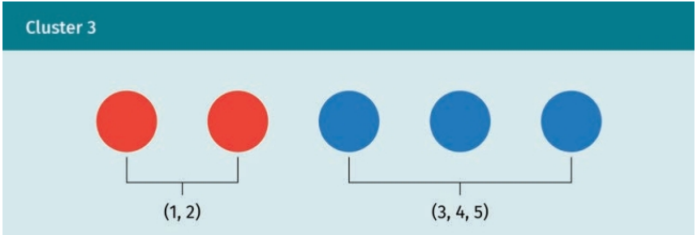


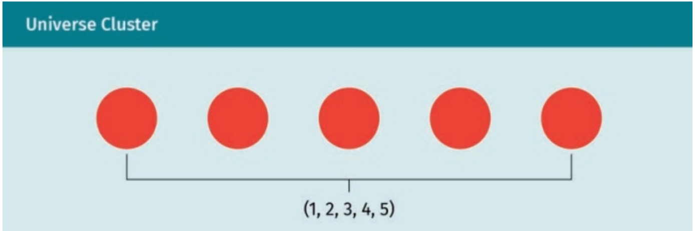
Single-Point Cluster

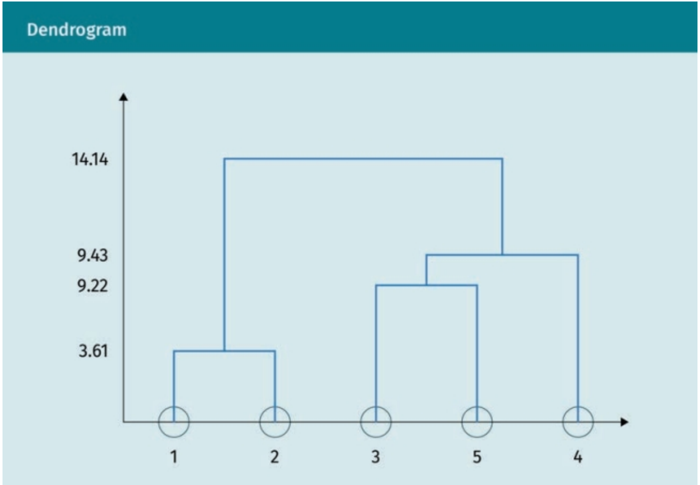


Cluster 1


Cluster 2
4
(1, 2)
(3, 5)

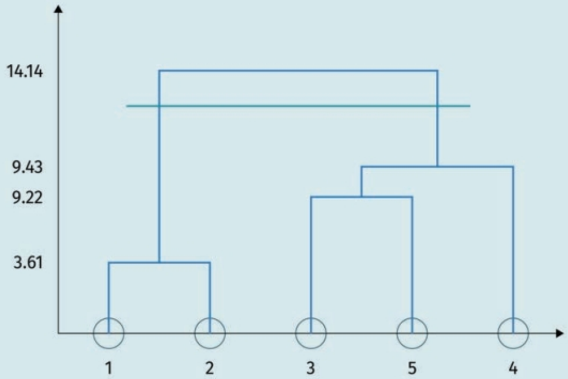

Cluster 3
(1, 2)
(3, 4, 5)


Universe Cluster
(1, 2, 3, 4, 5)


Dendrogram

- The optimum number of clusters, which ensures the largest intra-distances, can be determined heuristically through the following steps:
  - Determine the largest vertical line in the dendrogram that does not intersect any of the other clusters. In our example, it is the vertical line from 9.43 to 14.14 with a length of 4.71
  - Draw a horizontal line along this line
  - The optimal number of clusters is equal to the number of intersections this horizontal line has. In our example, there are two intersections

Optimal Clusters



Density-Based Clustering

Density-Based Clustering

- Density-based clustering approaches work to identify clusters by grouping "dense" data records together, which permits the representation of arbitrarily shaped clusters, as well as the learning of outliers within the data

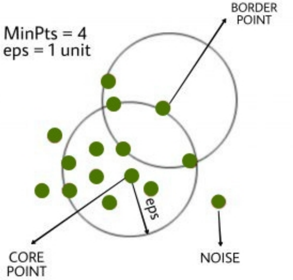
$$N(p) = \{q \in D | dist(p, q) \leq \epsilon\} \tag{1}$$

DBSCAN algorithm

- ϵ: It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered neighbors.
  - If the ϵ value is chosen too small then large part of the data will be considered as outliers.
  - If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters.
- MinPts: Minimum number of neighbors (data points) within ϵ radius. Larger the dataset, the larger value of MinPts must be chosen.
  - As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1

Type of points in DBSCAN

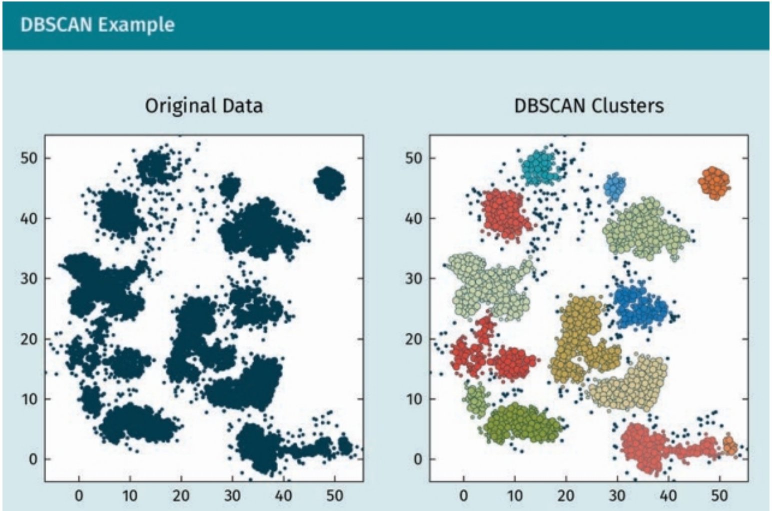
- Core Point: A point is a core point if it has more than MinPts points within ϵ.
- Border Point: A point which has fewer than MinPts within ϵ but it is in the neighborhood of a core point.
- Noise or outlier: A point which is not a core point or border point.



DBSCAN algorithm

1. Find all the neighbor points within ϵ and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.
   - A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the ϵ distance.
   - This is a chaining process. So, if b is neighbor of c, c is neighbor of d, d is neighbor of e, which in turn is neighbor of a implies that b is neighbor of a.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

DBSCAN example



DBSCAN Example

Questions?

German
Research Center
for Artificial
Intelligence