Geometric Data Analysis

Density-based Clustering

MAT 6480W / STT 6705V

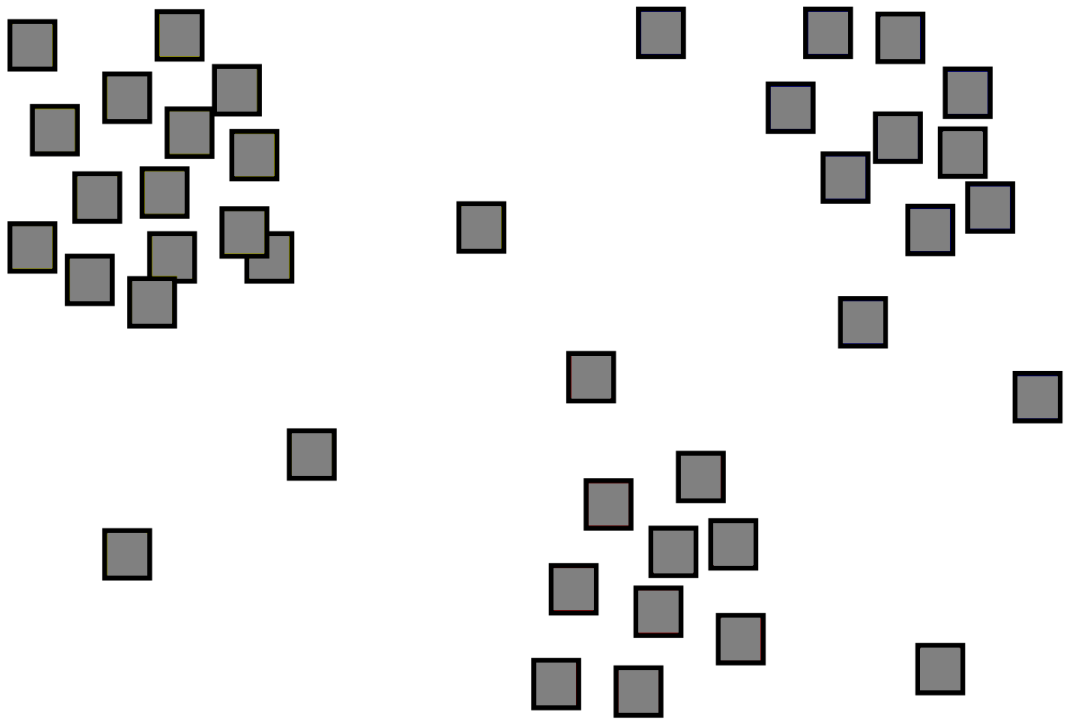
Guy Wolf
guy.wolf@umontreal.ca

Université de Montréal
Fall 2019

Outline

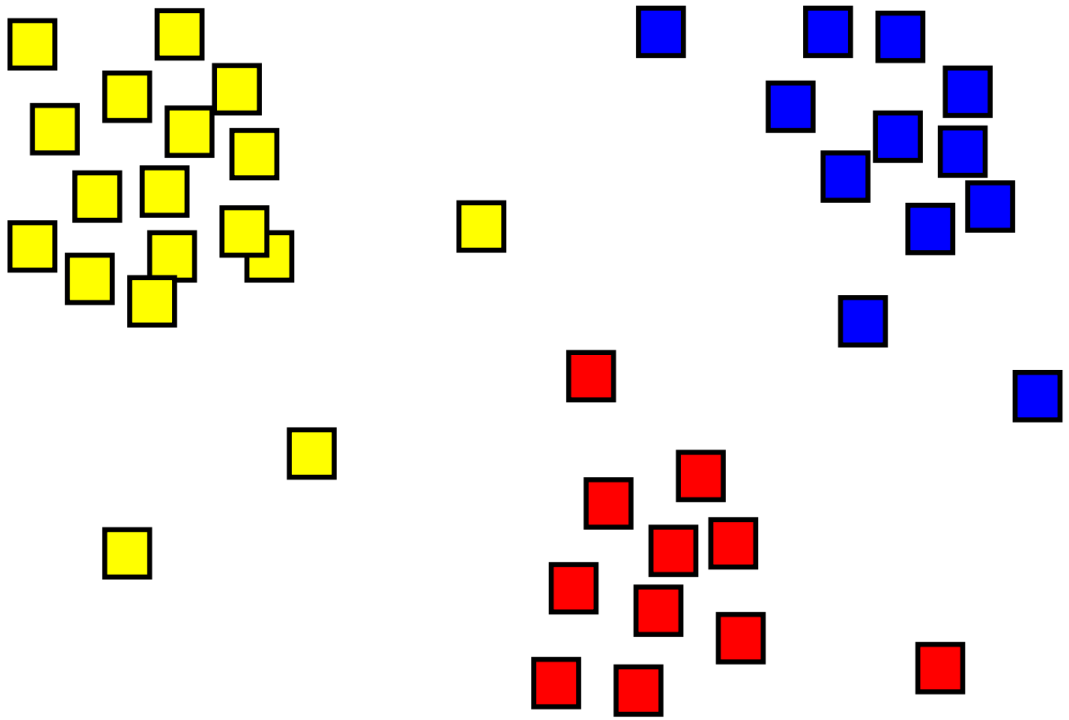
Clustering

Group together similar “items” while separating ones that are different from each other.



Clustering

Group together similar “items” while separating ones that are different from each other.



Clustering

Group together similar "items" while separating ones that are different from each other.

Clustering is a common examples of an unsupervised task.

Typically, clustering (or cluster analysis) is used as:

1. A stand-alone tool descriptive tool to reveal data distribution and relations
2. A preprocessing tool (e.g., discretization) for other algorithms
3. A preliminary step for outlier and anomaly detection (e.g, identifying normal behavior patterns).

Clustering can be extended to underlying distribution inference (e.g., Gaussian mixture model).

Clustering is often considered as an ill-posed problem. Unlike classification validation methods (e.g., cross-validation), there is no general application-independent validation approach for clustering.

In general, good clusters are always expected to be:

- Cohesive: high intra-class similarity
- Distinctive: low inter-class similarity

However, these criteria are vague and depend on the considered cluster types.

In practice, clusters are usually evaluated by their interpretability using specific domain knowledge.

If we have some labeled reference data, we can evaluate the clustering quality with **RandIndex**:

RandIndex

Given a dataset $X = \{x_1, \ldots, x_N\}$, corresponding labels $L = \{l_1, \ldots, l_N\}$, and a clustering function $C : X \to \{1, \ldots, k\}$, define $\text{RandIndex}(X, L, C) = \binom{N}{2}^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \text{correct}(x_i, x_j)$ where

$$\text{correct}(x_i, x_j) = \begin{cases} 1 & l_i = l_j \& C(x_i) = C(x_j) \\ 1 & l_i \neq l_j \& C(x_i) \neq C(x_j) \\ 0 & \text{otherwise} \end{cases}$$

If we have some labeled reference data, we can evaluate the clustering quality with **RandIndex**:

RandIndex

Given a dataset $X = \{x_1, \ldots, x_N\}$, corresponding labels $L = \{l_1, \ldots, l_N\}$, and a clustering function $C : X \to \{1, \ldots, k\}$, define $\text{RandIndex}(X, L, C) = \binom{N}{2}^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \text{correct}(x_i, x_j)$.

Notice that RandIndex does not require correspondence (in type/number) or mapping between labels and cluster indices.

Also, unlike classification validation, RanIndex doesn't quantify prediction quality, but suitability to detect clustering patterns in similar data, which may be shifted, rotated or otherwise deformed.

Clustering

Types of clusters

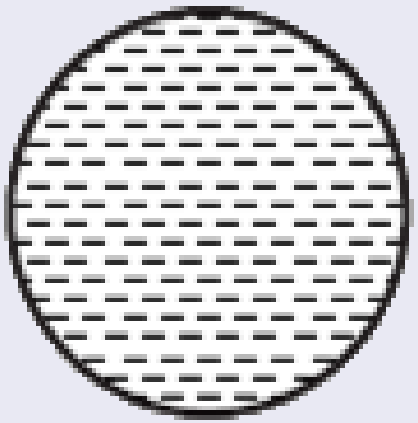
Cluster types can be characterized in several ways:

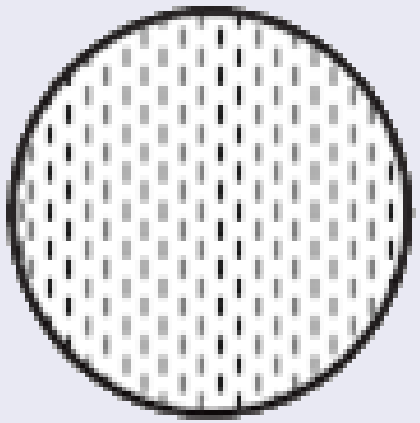
- Exclusive vs. nonexclusive - can a data point belong to two clusters?
- Fuzzy vs. non-fuzzy - is cluster membership binary, or quantifiable?
- Heterogeneous vs. homogeneous - are all clusters the same size/shape/density?
- Partial vs. complete - does every data point have to be in a cluster?

Beyond these general characterizations, the shape of the considered clusters is crucial for formulating a clustering strategy.

The shape of the considered clusters is crucial for formulating a clustering strategy:

Well-separated clusters

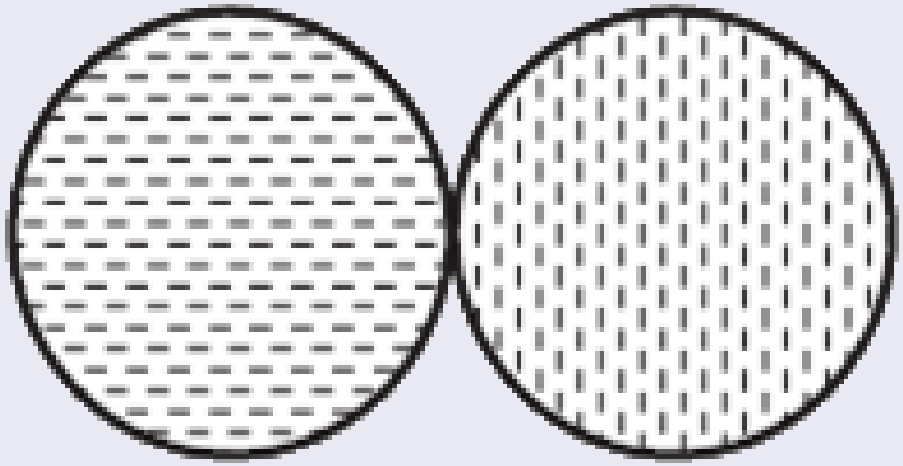




Convex clusters, where each point is closer to all other points in its cluster than to any other point in the data.

The shape of the considered clusters is crucial for formulating a clustering strategy:

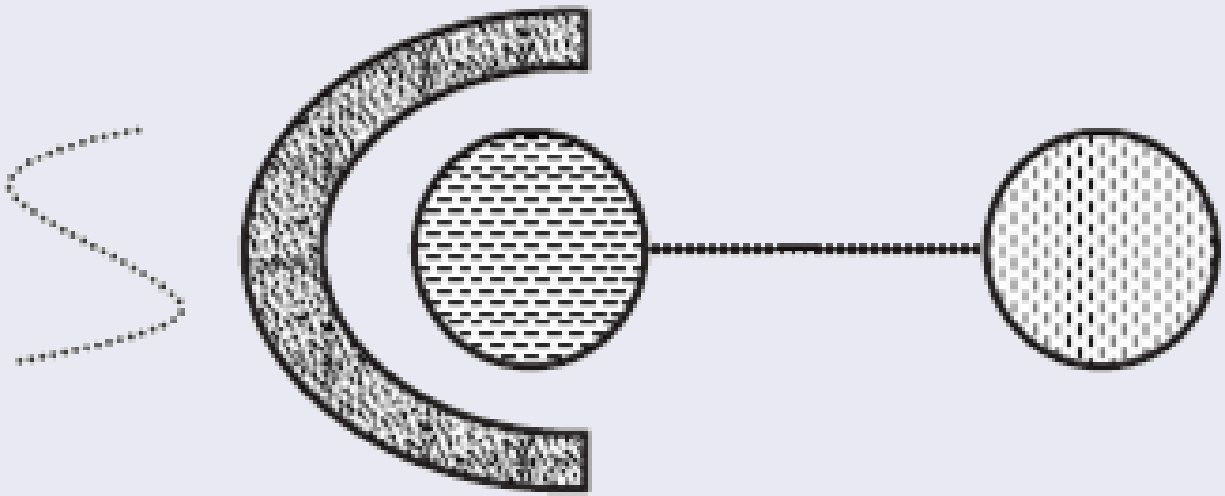
Center-based clusters



Convex clusters, where each cluster is identified by a centroid s.t. every point in the cluster is closer to its cluster-centroid than to any other cluster-centroid.

The shape of the considered clusters is crucial for formulating a clustering strategy:

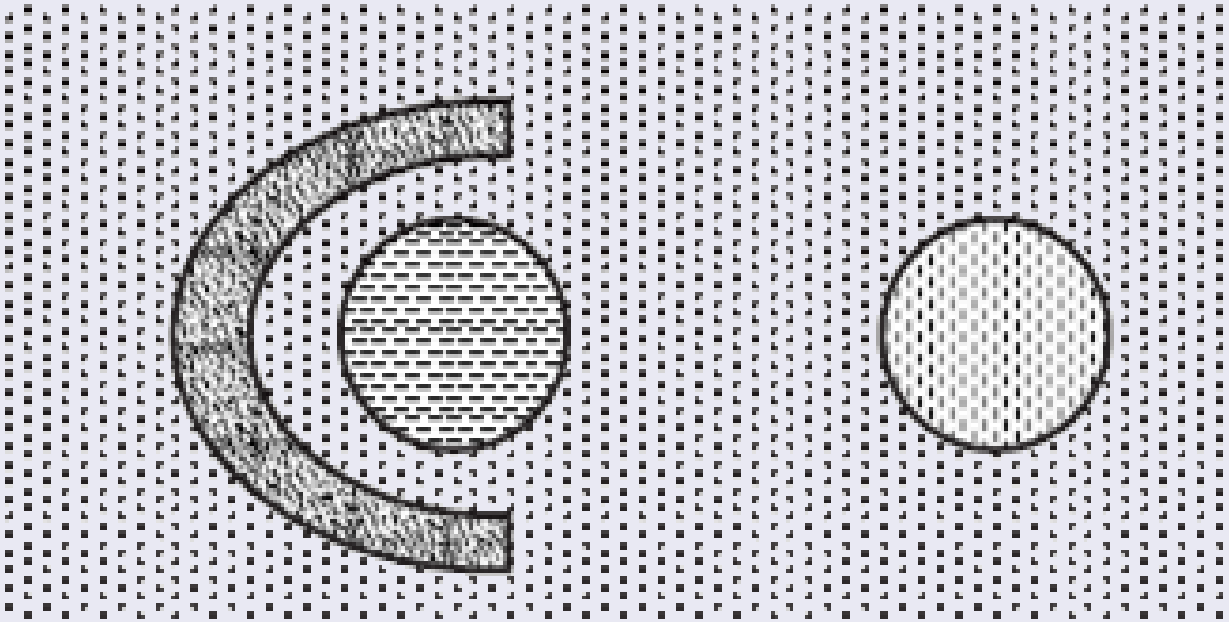
Contiguity-based clusters



Each cluster is a contiguous set of data points s.t. every point in the cluster is closer to at least one other point in it than to any point outside the cluster.

The shape of the considered clusters is crucial for formulating a clustering strategy:

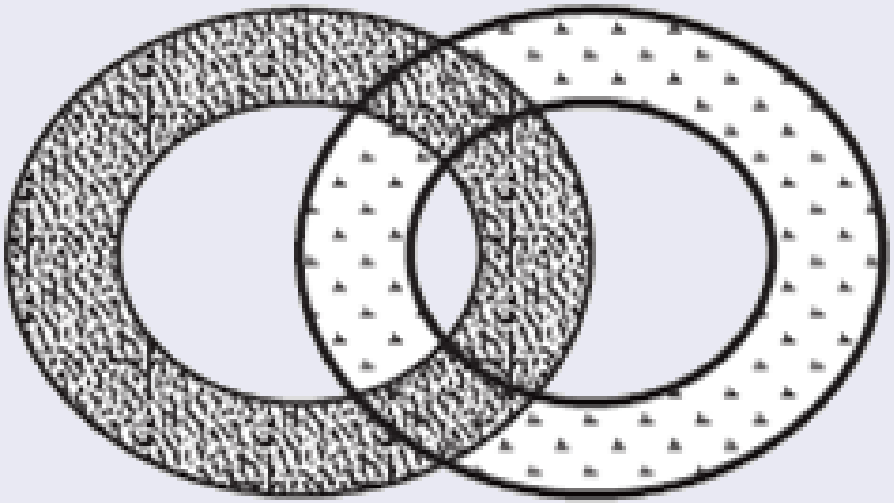
Density-based clusters



Clusters are regions of high density separated by regions of low density.

The shape of the considered clusters is crucial for formulating a clustering strategy:

Conceptual clusters



Clusters are defined by shared properties satisfied by all points in the cluster and not satisfied outside of the cluster.

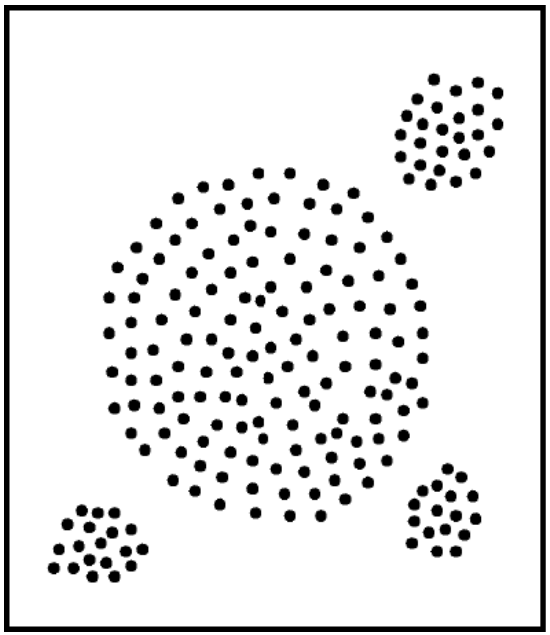
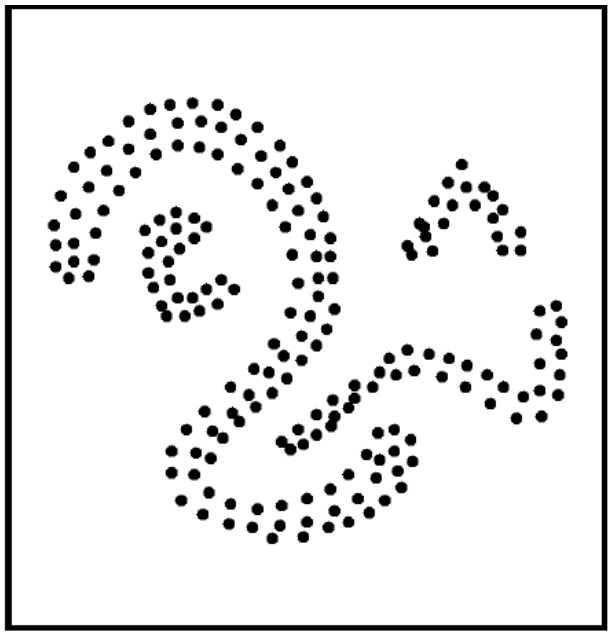
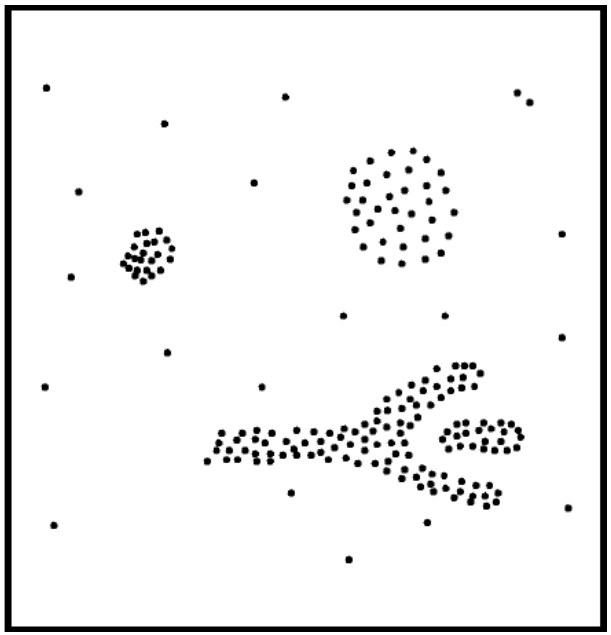
Clustering

Clustering approaches

Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	– Can find arbitrarily shaped clusters – Clusters are dense regions of objects in space that are separated by low-density regions – Cluster density: Each point must have a minimum number of points within its "neighborhood" – May filter out outliers
Grid-based methods	– Use a multiresolution grid data structure – Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

Density-based clustering

Density-based clustering methods consider **clusters as dense (or locally dense) regions** separated by sparse regions.

Such methods work via **density estimation** and thresholding to recover contiguous clusters of various shapes and sizes.

DBScan

Core, border, and noise points

DBScan performs a density-based scan of the data to progressively uncover clusters based on the following terminology:

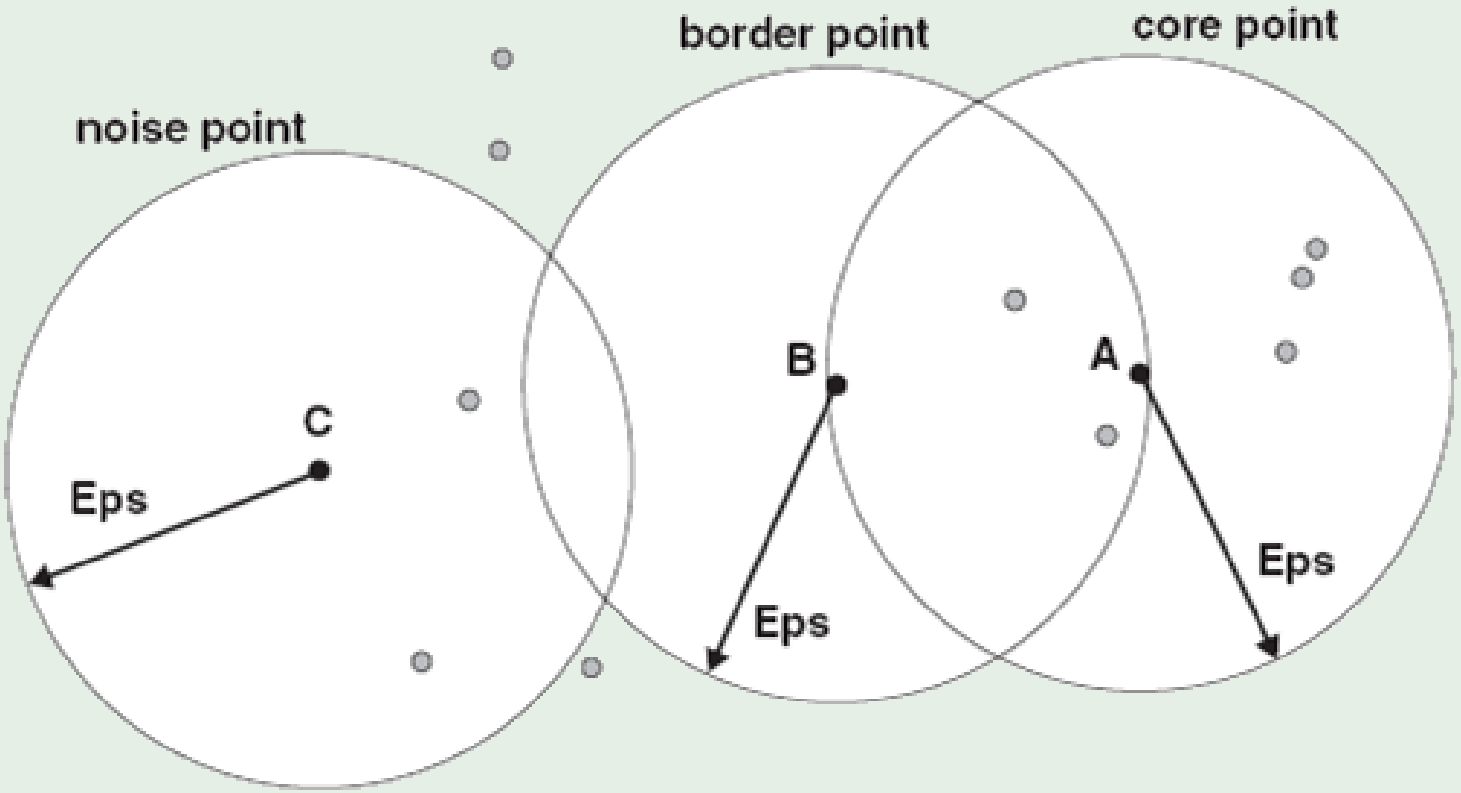
Configuration: **Input:** dataset X and distance $d(\cdot, \cdot)$

- ε (epsilon): radius for defining neighborhoods $N_\varepsilon(x) = \{y \in X \mid d(x, y) \leq \varepsilon\}$ for any data point $x \in X$.
- min_pts: threshold for defining dense neighborhoods as $|N_\varepsilon(x)| \geq$ min_pts.

Point types:

- Core point: a data point with dense neighborhood.
- Border point: a non-core point in a neighborhood of a core-point.
- Noise point: any point that is not a core- or border-point.

Example (point types)



min_pts = 5

DBScan

Density reachability and connectivity

Using this terminology, DBScan defines the following relations between data points:

Density reachability

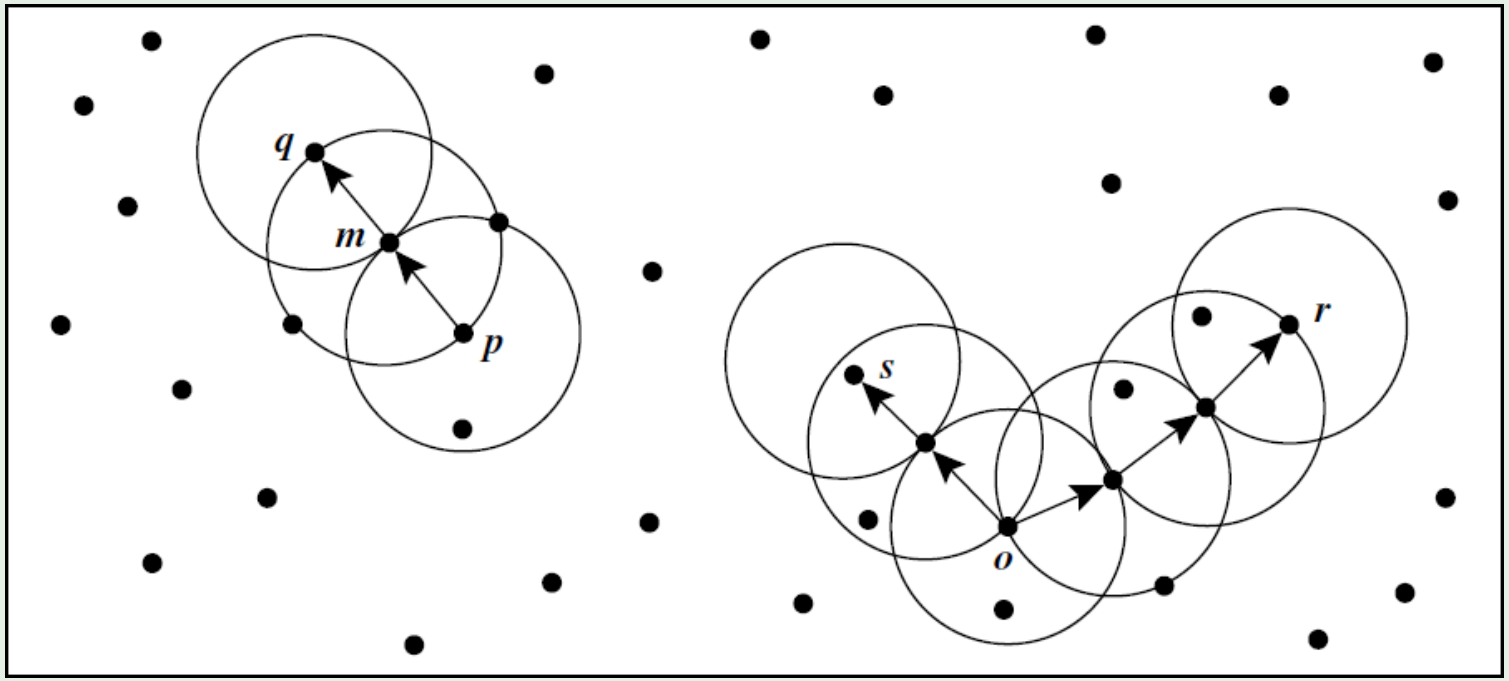
A data point $x \in X$ is density-reachable from a core-point c if there exists a path $c = p_1 \to \cdots \to p_\ell \to p_{\ell+1} = x$ (of arbitrary length $\ell > 0$) such that p_i is a core point and $p_{i+1} \in N_\varepsilon(p_i)$ for $i = 1, \ldots, \ell$.

Density connectivity

Two data points $x, y \in X$ are density connected if there exists some core point c such that both x and y are density reachable from c.

DBScan clusters are defined as sets of density-connected data points.

Example (density-reachability & density-connectivity)



- q is density-reachable from core-point p (via core-point m)
- s and r are density-connected since both are density-reachable from core-point o

DBScan

Cluster construction

The DBScan algorithm builds a clusters from core points using the following steps:

DBScan algorithm

Mark all data points as unvisited

Repeat the following steps for each data point $x \in X$:

- **If** x has been visited, **then** skip it.
- **If** $|N_\varepsilon(x)| < \text{min\_pts}$, **then** skip it.
- **Mark** x as a core point **and** as visited.
- **Start** a new cluster $C_x \leftarrow \{x\}$:
  - **Add** all unvisited density-reachable points from x to C_x.

Mark all unvisited points as noise points with no cluster.

DBScan

Cluster construction

The DBScan algorithm builds a clusters from core points using the following steps:

Add all unvisited density-reachable points from x to C_x

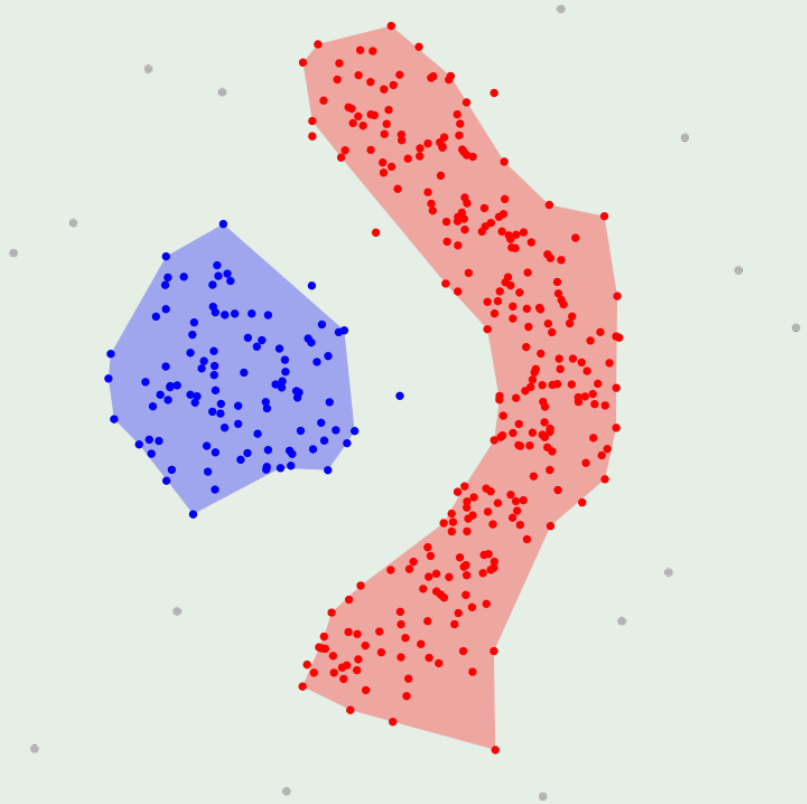
Initialize: $Q \leftarrow N_\varepsilon(x)$

Repeat the following steps for each data point $y \in Q$:

- **If** y has been visited, **then** skip it.
- **Add** y to C_x and **mark** it as visited.
- **If** $|N_\varepsilon(y)| < \text{min\_pts}$, **then**:
  - **Mark** it as border point and **move on**.
- **Mark** y as a core point and **set** $Q \leftarrow Q \cup N_\varepsilon(y)$.

Until $Q = \emptyset$

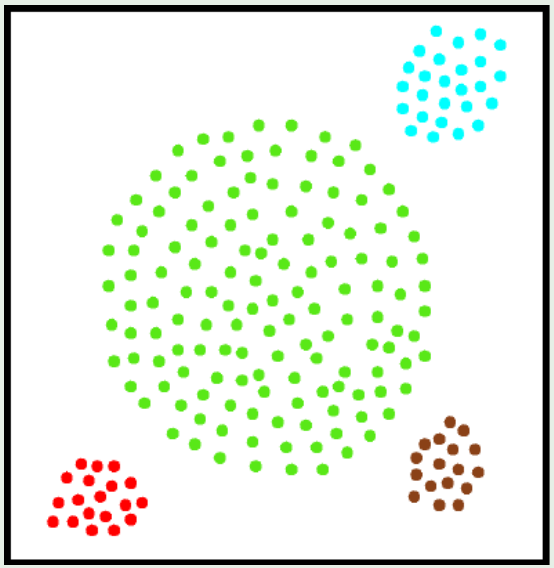
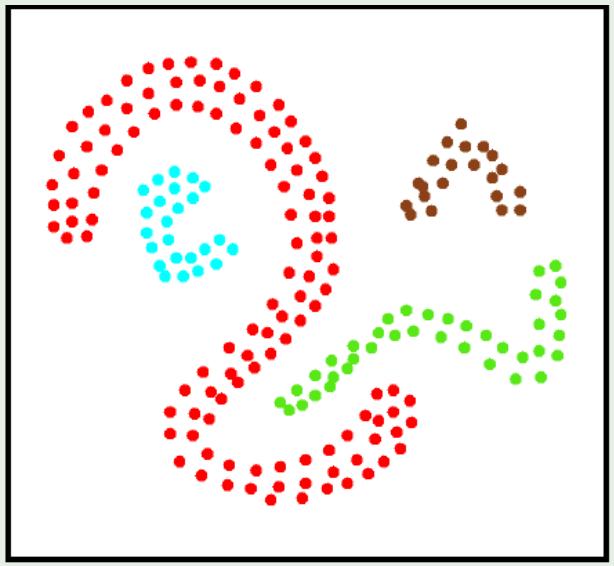
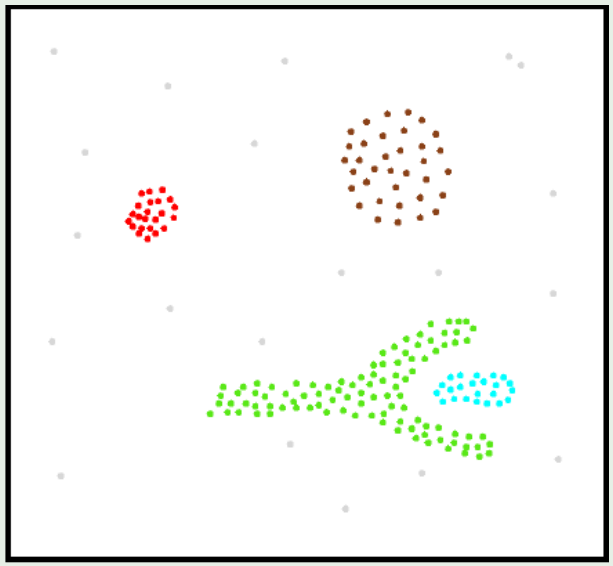
Example



Adapted from Wikipedia

This approach can capture a wide variety of cluster shapes.

Example

However, it is very sensitive to the configuration parameters, and suffers greatly from the curse of dimensionality.

Summary

Cluster analysis aims to detect clustering patterns for descriptive and preprocessing tasks.

- Generally, it is an ill-defined problem, since clustering patterns are not a coherent task-independent concept.
- While there are numerous cluster evaluation measures, their quality ultimately depends on task-dependent interpretability.

Density-based approaches consider clusters as dense regions separated by sparse regions.

- Such approaches rely on density estimation methods.
- DBScan and its variations (e.g., OPTICS) are popular examples of such an approach.