XI. Cluster Analysis

- Data Mining Overview
- Cluster Analysis Basics
- Hierarchical Cluster Analysis
- Iterative Cluster Analysis
- Density-Based Cluster Analysis
- Cluster Evaluation
- Constrained Cluster Analysis
Density-Based Cluster Analysis

Density-based algorithms strive to partition the graph \( G = \langle V, E, w \rangle \), better: the set of points \( V \), into regions of equal density.

Approaches to density estimation:

- parameter-based: the type of the underlying data distribution is known
- parameterless: construction of histograms, superposition of kernel density estimators
Density-Based Cluster Analysis

Density-based algorithms strive to partition the graph $G = \langle V, E, w \rangle$, better: the set of points $V$, into regions of equal density.

Approaches to density estimation:

- parameter-based: the type of the underlying data distribution is known
- parameterless: construction of histograms, superposition of kernel density estimators

Example (Caribbean Islands):
Density-Based Cluster Analysis
Density Estimation with Gaussian Kernel for the Example

- Dominican Republic
- Cuba
- Puerto Rico
Density-Based Cluster Analysis
Density Estimation with Gaussian Kernel for the Example
Density-Based Cluster Analysis

Density Estimation with Gaussian Kernel for the Example

Dominican Republic
Cuba
Puerto Rico
Density-Based Cluster Analysis

Density Estimation with Gaussian Kernel for the Example
Density-Based Cluster Analysis

DBSCAN: Density Estimation Principle  [Ester et al. 1996]

Let $N_\varepsilon(v)$ denote the $\varepsilon$-neighborhood of some point $v \in V$. Distinguish between three kinds of points:

1. $v$ is a core point $\iff |N_\varepsilon(v)| \geq \text{MinPts}$
2. $v$ is a noise point $\iff$
   $v$ is not density-reachable from any core point
3. $v$ is a border point otherwise
A point $u$ is **density-reachable** from a point $v$, if either of the following conditions hold:

(a) $u \in N_\varepsilon(v)$, where $v$ is a core point.

(b) There exists a set of points $\{v_1, \ldots, v_l\}$, where $v_{i+1} \in N_\varepsilon(v_i)$ and $v_i$ is core point, $i = 1, \ldots, l - 1$, with $v_1 = v$, $v_l = u$.

Condition (b) can be considered as the transitive application of Condition (a).
Density-Based Cluster Analysis

DBSCAN: Cluster Interpretation

A cluster $C \subseteq V$ fulfills the following two conditions:

1. $\forall u, v: \text{If } v \in C \text{ and } u \text{ is density-reachable from } v, \text{ then } u \in C.$
A cluster $C \subseteq V$ fulfills the following two conditions:

1. $\forall u, v : \text{If } v \in C \text{ and } u \text{ is density-reachable from } v, \text{ then } u \in C$. 

2. $\forall u, v \in C : u \text{ is density-connected with } v$, which is defined as follows:

   There exists a point $t$ wherefrom $u$ and $v$ are density-reachable.
Remarks:

- Condition 1 (maximality) states a constraint between any two points. Condition 2 (connectivity) states an additional constraint with respect to a third point.

- The maximality condition is problematic if a border point lies in the \( \varepsilon \)-neighborhoods of two core points that belong to two different clusters. Such a border point would then belong to both clusters; however, the algorithm breaks this tie by assigning this point to the first cluster found.
Density-Based Cluster Analysis

DBSCAN: Algorithm

Input:

\( G = \langle V, E, w \rangle \). Weighted graph.
\( d \). Distance measure for two nodes in \( V \).
\( \varepsilon \). Neighborhood radius.
\( MinPts \). Lower bound for point number in \( \varepsilon \)-neighborhood.

Output:

\( \gamma : V \rightarrow \mathbb{Z} \). Cluster assignment function.

1. 
2. 
3. \( v = \text{choose_unclassified_point}(V) \)
4. \( N_\varepsilon(v) = \text{neighborhood}(G, d, v, \varepsilon) \)
5. \( \text{IF } |N_\varepsilon(v)| \geq \text{MinPts \ THEN } \text{ // v is core point} \)
6. 
7. \( C_i = \text{densityreachablehull}(G, d, N_\varepsilon(v)) \) // form a new cluster
8. \( \text{FOREACH } v \in C_i \text{ DO } \gamma(v) = i \) // label the cluster points
9. \( \text{ELSE } \gamma(v) = -1 \) // v is _tentatively_ classified as noise
10. 
11. 
Density-Based Cluster Analysis

DBSCAN: Algorithm

Input: \( G = \langle V, E, w \rangle \). Weighted graph.
\( d \). Distance measure for two nodes in \( V \).
\( \varepsilon \). Neighborhood radius.
\( MinPts \). Lower bound for point number in \( \varepsilon \)-neighborhood.

Output: \( \gamma : V \rightarrow \mathbb{Z} \). Cluster assignment function.

1. \( i = 0 \)
2. **WHILE** \( \exists v : (v \in V \text{ AND } \gamma(v) = \bot) \) **DO** // \( \bot = \text{unclassified} \)
3. \( v = \text{choose\_unclassified\_point}(V) \)
4. \( N_\varepsilon(v) = \text{neighborhood}(G, d, v, \varepsilon) \)
5. **IF** \( |N_\varepsilon(v)| \geq \text{MinPts} \) **THEN** // \( v \) is core point
6. \( i = i + 1 \)
7. \( C_i = \text{density\_reachable\_hull}(G, d, N_\varepsilon(v)) \) // form a new cluster
8. **FOREACH** \( v \in C_i \) **DO** \( \gamma(v) = i \) // label the cluster points
9. **ELSE** \( \gamma(v) = -1 \) // \( v \) is _tentatively_ classified as noise
10. **ENDDO**
11. **RETURN**(\( \gamma \))
Density-Based Cluster Analysis

DBSCAN
Density-Based Cluster Analysis

DBSCAN
Density-Based Cluster Analysis

DBSCAN
Density-Based Cluster Analysis

DBSCAN
Density-Based Cluster Analysis

DBSCAN
Density-Based Cluster Analysis

DBSCAN

- Core point
- Border point
- Noise point
Density-Based Cluster Analysis

DBSCAN

Core point
Border point
Noise point
Density-Based Cluster Analysis

DBSCAN

Core point
Border point
Noise point
Density-Based Cluster Analysis

DBSCAN

- Core point
- Border point
- Noise point
Density-Based Cluster Analysis

DBSCAN

- Core point
- Border point
- Noise point

Noise point
Remarks:

- Note that points that are labeled as noise can be re-labeled with a cluster number exactly once. I.e., a point will retain its tentative noise label only if it is not density-reachable from any other point.

- The construction of $C_i$ as the density-reachable hull of $N_\varepsilon(v)$ (Line 7) corresponds to a recursive analysis of the points in $N_\varepsilon(v)$ with regard to their density reachability.

- A slightly different and compact formulation of the algorithm is given in [Tan/Steinbach/Kumar 2005, p. 528].
Density-Based Cluster Analysis

Merging Principles

- Cluster analysis
  - hierarchical
    - agglomerative
      - single link, group average
    - divisive
  - iterative
    - exemplar-based
    - exchange-based
    - Kerninghan-Lin
  - density-based
    - point-density-based
      - DBSCAN
    - attraction-based
      - MajorClust
    - gradient-based
      - simulated annealing
    - competitive
      - evolutionary strategies
  - meta-search-controlled
  - stochastic
    - Gaussian mixtures
    - ...
Density-Based Cluster Analysis

MajorClust: Density Estimation Principle  [Stein/Niggemann 1999]

The weighted edges in a graph $G = \langle V, E, w \rangle$ are interpreted as attracting forces, whereas members of the same cluster combine their forces. Illustration:

Unique membership situation, leading to a merge of two clusters:
Density-Based Cluster Analysis

MajorClust: Density Estimation Principle  [Stein/Niggemann 1999]

The weighted edges in a graph $G = \langle V, E, w \rangle$ are interpreted as attracting forces, whereas members of the same cluster combine their forces. Illustration:

Unique membership situation, leading to a merge of two clusters:

![Diagram showing a merge of two clusters](image)

Unique membership situation, leading to a change of cluster membership:

![Diagram showing a change of cluster membership](image)
Density-Based Cluster Analysis

MajorClust: Density Estimation Principle [Stein/Niggemann 1999]

The weighted edges in a graph \( G = \langle V, E, w \rangle \) are interpreted as attracting forces, whereas members of the same cluster combine their forces. Illustration:

Unique membership situation, leading to a merge of two clusters:

Unique membership situation, leading to a change of cluster membership:

Ambiguous membership situation:
Density-Based Cluster Analysis

MajorClust: Algorithm

Input: \( G = (V, E, w) \). Weighted graph.
\[ d \] Distance measure for two nodes in \( V \).

Output: \( \gamma : V \rightarrow \mathbb{N} \). Cluster assignment function.

1. \[ i = 0, \ t = \text{False} \]
2. \[ \text{FOREACH } v \in V \ \text{DO} \]
3. \[ i = i + 1, \ \gamma(v) = i \]
4. \[ \text{ENDDO} \]
5. \[ \text{UNLESS } t \text{ DO} \]
6. \[ \gamma^* = \arg\max_{i \in \{1, \ldots, |V|\}} \sum_{\{u,v\} \in E \land \gamma(u) = i} w(u,v) \]
7. \[ \text{IF } \gamma(v) \neq \gamma^* \ \text{THEN } \gamma(v) = \gamma^*, \ t = \text{False} \ \text{ENDIF} \ // \text{relabeling} \]
8. \[ \text{ENDDO} \]
9.
10.
Density-Based Cluster Analysis
MajorClust: Algorithm

Input: \[ G = \langle V, E, w \rangle \]. Weighted graph.
      \( d \). Distance measure for two nodes in \( V \).

Output: \( \gamma : V \to \mathbb{N} \). Cluster assignment function.

1. \( i = 0, \ t = False \)
2. FOREACH \( v \in V \) DO \( i = i + 1, \ \gamma(v) = i \) ENDDO
3. UNLESS \( t \) DO
4. \( t = True \)
5. FOREACH \( v \in V \) DO
6. \( \gamma^* = \arg\max_{i: i \in \{1, \ldots, |V|\}} \sum_{\{u,v\}: \{u,v\} \in E \land \gamma(u)=i} w(u,v) \)
7. IF \( \gamma(v) \neq \gamma^* \) THEN \( \gamma(v) = \gamma^*, \ t = False \) ENDDO // relabeling
8. ENDDO
9. ENDDO
10. RETURN(\( \gamma \))
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust
Density-Based Cluster Analysis

MajorClust

[Diagram of clustered data points]
Each clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ induces $k$ subgraphs within $G = \langle V, E, w \rangle$. MajorClust is a heuristic to maximize the weighted partial edge connectivity, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$
Each clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ induces $k$ subgraphs within $G = \langle V, E, w \rangle$. MajorClust is a heuristic to maximize the weighted partial edge connectivity, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$
Density-Based Cluster Analysis

MajorClust: Density Estimation Principle (continued)

Each clustering $C = \{C_1, \ldots, C_k\}$ induces $k$ subgraphs within $G = \langle V, E, w \rangle$.

MajorClust is a heuristic to maximize the *weighted partial edge connectivity*, $\Lambda(C)$.

$$\Lambda(C) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$
Each clustering $C = \{C_1, \ldots, C_k\}$ induces $k$ subgraphs within $G = \langle V, E, w \rangle$. MajorClust is a heuristic to maximize the weighted partial edge connectivity, $\Lambda(C)$.

$$\Lambda(C) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$

\[
\begin{align*}
\lambda_1 &= 1 \\
\Lambda &= 5 \cdot 1 + 3 \cdot 2 = 11 \\
\lambda_2 &= 2 \\
\lambda_3 &= 2 \\
\Lambda &= 3 \cdot 2 + 2 \cdot 1 + 3 \cdot 2 = 14 \\
\lambda_1 &= 2 \\
\lambda_2 &= 3 \\
\Lambda &= \Lambda^* = 4 \cdot 2 + 4 \cdot 3 = 20
\end{align*}
\]
Density-Based Cluster Analysis
MajorClust: Density Estimation Principle (continued)

minimization of cut weight

Λ maximization
Theorem 5 (Strong Splitting Condition [Stein/Niggemann 1999])

Let $C = \{C_1, \ldots, C_k\}$ be a partitioning of a graph $G = \langle V, E, w \rangle$. Moreover, let $\lambda(G)$ denote the edge connectivity of $G$, and let $\lambda_1, \ldots, \lambda_k$ denote the edge connectivity values of the $k$ subgraphs that are induced by $C_1, \ldots, C_k$.

If the inequality $\lambda(G) < \min\{\lambda_1, \ldots, \lambda_k\}$ holds, then the partitioning defined by $\Lambda$-maximization corresponds to the minimum cut splitting of $G$. The inequality is denoted as “Strong Splitting Condition”.

Density-Based Cluster Analysis

DBSCAN versus MajorClust: Low-Dimensional Data

Caribbean Islands, about 20,000 points:
Density-Based Cluster Analysis
DBSCAN versus MajorClust: Low-Dimensional Data (continued)

Caribbean Islands, about 20,000 points:

Cluster analysis by DBSCAN:

- $\varepsilon = 3.0$, $\text{MinPts} = 3$
- $\varepsilon = 5.0$, $\text{MinPts} = 4$
- $\varepsilon = 10.0$, $\text{MinPts} = 5$
Density-Based Cluster Analysis

DBSCAN versus MajorClust: Low-Dimensional Data  (continued)

The problem of finding useful $\varepsilon$-values for DBSCAN:

- $\varepsilon = 3.0$, MinPts = 3

Two separate clusters were detected.

- $\varepsilon = 3$

The clusters are merged.

- $\varepsilon = 8$
Density-Based Cluster Analysis

DBSCAN versus MajorClust: Low-Dimensional Data  (continued)

Caribbean Islands, about 20,000 points:

Cluster analysis by MajorClust:
The problem of the global analysis approach (no restriction by means of an $\varepsilon$-neighborhood) in MajorClust:
Remarks:

- MajorClust is superior to DBSCAN with regard to the identification of differently dense clusters within the same clustering. DBSCAN is more flexible (= can be better adapted) than MajorClust with regard to point densities in different clusterings.

- MajorClust considers always all points of $V$, while DBSCAN works locally, i.e., on small subsets of $V$. 
Density-Based Cluster Analysis

DBSCAN versus MajorClust: High-Dimensional Data

Document categorization setting using the Reuters corpus:

- 1000 documents
- 10 categories: politics, culture, economics, etc.
- the documents are equally distributed and belong to exactly one category
- dimension of the feature space: > 10,000

DBSCAN:

- degenerates with increasing number of dimensions
- the degeneration is rooted in the computation of the $\varepsilon$-neighborhood
- dimension reduction provides a way out, e.g. by embedding the data with multi-dimensional scaling, MDS
Density-Based Cluster Analysis
DBSCAN versus MajorClust: High-Dimensional Data (continued)

Classification effectiveness ($F$ measure) over dimension number:
Remarks:

- Usually, the neighborhood search in high-dimensional spaces cannot be solved efficiently. Given \( p \) dimensions with \( p \) about 10 or larger, an exhaustive search, i.e., a linear scan of all feature vectors will be more efficient than the application of a space partitioning data structure (quad-tree, k-d tree, etc.) or a data partitioning data structure (\( R \)-tree, \( Rf \)-tree, \( X \)-tree, etc.).

- DBSCAN employs the \( R \)-tree data structure to compute \( \varepsilon \)-neighborhoods. This data structure accomplishes the major part of the DBSCAN cluster analysis approach and is ideally suited for treating low-dimensional data efficiently. The application of DBSCAN to high-dimensional data either requires an embedding into a low-dimensional space or to accept the runtime for a naive construction of \( \varepsilon \)-neighborhoods.

- Neighborhood search in high-dimensional spaces can be addressed with approximate methods such as locality sensitive hashing (LSH), or Fuzzy fingerprinting. [Weber 1999] [Gionis/Indyk/Motwani 1999-2004] [Stein 2005-2007] [Andoni 2009]