Density-based Cluster Algorithms in Low-dimensional and High-dimensional Applications

Benno Stein
Bauhaus University Weimar
Web-based Information Systems

Michael Busch
IBM Silicon Valley Laboratory
(WebSphere II Project)
Introduction

Given a set of objects (documents) $D = \{d_1, \ldots, d_n\}$.

Clustering is the *unsupervised* classification of $d_i$ into groups.

Result is a partitioning $C$ of $D$. 
Introduction

Given a set of objects (documents) \( D = \{d_1, \ldots, d_n\} \).
Clustering is the *unsupervised* classification of \( d_i \) into groups.

Result is a partitioning \( C \) of \( D \).

Objective: Maximize intra-group similarity.

Minimize inter-group similarity.
Introduction

Cluster algorithms form the backbone of document categorization.

Example AIsearch [www.aisearch.de]:
Introduction

Documents → Feature extraction → Similarity computation → Cluster analysis → Categories

Indexing (includes parsing, stopword elimination, stemming):

Vector representation with weighting scheme:

\[ \text{tf} \cdot \log \frac{|X|}{\# \text{docs}} \]

\[ # \text{abs} \]

```
<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>chrysler</td>
<td>0.12</td>
</tr>
<tr>
<td>deal</td>
<td>0.2</td>
</tr>
<tr>
<td>leav</td>
<td>0.1</td>
</tr>
<tr>
<td>amc</td>
<td>0.01</td>
</tr>
<tr>
<td>cat</td>
<td>0.0</td>
</tr>
<tr>
<td>sal</td>
<td>0.01</td>
</tr>
<tr>
<td>dog</td>
<td>0.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```
Introduction

Documents → Feature extraction → Similarity computation → Cluster analysis → Categories

e.g. under the vector space model:

\[
\begin{bmatrix}
0.3 \\
0.5 \\
0.1 \\
0.0 \\
0.0 \\
0.0 \\
0.3 \\
0.5 \\
0.1 \\
0.0 \\
0.0 \\
0.0
\end{bmatrix}
\]

Document A

\[
\begin{bmatrix}
0.1 \\
0.3 \\
0.5 \\
0.0 \\
0.0 \\
0.1
\end{bmatrix}
\]

Document B

\[
\begin{bmatrix}
0.2 \\
0.0 \\
0.3 \\
0.0 \\
0.0 \\
0.0
\end{bmatrix}
\]

Document C

TIR’05 Sep. 9th, 2005 Stein/Busch
Introduction

Documents $\rightarrow$ Feature extraction $\rightarrow$ Similarity computation $\rightarrow$ Cluster analysis $\rightarrow$ Categories

- Document A
  - 0.3
  - 0.5
  - 0.1
  - 0.0
  - 0.0

- Document B
  - 0.1
  - 0.3
  - 0.5
  - 0.0
  - 0.1

- Document C
  - 0.2
  - 0.0
  - 0.0
  - 0.3
  - 0.0

Categories

TIR'05 Sep. 9th, 2005 Stein/Busch
Introduction

Documents \rightarrow \text{Feature extraction} \rightarrow \text{Similarity computation} \rightarrow \text{Cluster analysis} \rightarrow \text{Categories}

Cluster approach

- hierarchical
  - agglomerative
    - single-linkage, group average
    - min-cut-analysis
  - divisive
- iterative
  - exemplar-based
  - k-means, k-medoid
  - Kerninghan-Lin
- density-based
  - point concentration
  - DBSCAN
  - cumulative attraction
    - MajorClust
  - competitive
    - descend-methods
    - simulated annealing
    - genetic algorithm
- meta-search-controlled

TIR'05 Sep. 9th, 2005 Stein/Busch
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Hierarchical agglomerative: Single-linkage
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Single-linkage: Chaining Problem
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based algorithm: $k$-Means
Exemplar-based versus Linkage

Exemplar-based algorithms fail with large differences in size.
Exemplar-based versus Linkage

Exemplar-based algorithms fail with entwined clusters.
Exemplar-based versus Linkage

Exemplar-based algorithms fail with entwined clusters.
Exemplar-based algorithms fail with entwined clusters.
Density-based Cluster Analysis

Density-based algorithms try to separate the set $D$ into subsets of similar densities.

Density estimation can happen

- **parameter-based**: the underlying distribution is known
- **parameter-less**: histogramm, kernel function
  
  (construct barcharts, superimpose continuous functions)
Density-based Cluster Analysis

Density-based algorithms try to separate the set $D$ into subsets of similar densities.

Density estimation can happen

- parameter-based: the underlying distribution is known
- parameter-less: histogramm, kernel function
  (construct barcharts, superimpose continuous functions)

Example (Carribean Islands):
Density-based Cluster Analysis

Density estimation with Gaussian Kernel for the example.
Density-based Cluster Analysis

Density estimation with Gaussian Kernel for the example.
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
Puerto Rico
Density-based Algorithm: DBSCAN  [Ester et al. 1996]
Density-based Algorithm: DBSCAN  [Ester et al. 1996]

$p$ is core point: $|N_\varepsilon(p)| \geq \text{MinPts}$.
$p$ is noise point: $p$ is not density-reachable from a core point.
$p$ is border point: otherwise.
Density-based Algorithm: DBSCAN

- \( p \) is core point: \(|N_\varepsilon(p)| \geq \text{MinPts}\).
- \( p \) is noise point: \( p \) is not density-reachable from a core point.
- \( p \) is border point: otherwise.

\( p \) is density-reachable from \( q \):

1. \( p \in |N_\varepsilon(q)| \), where \( q \) is a corepoint
2. Transitive application of condition (a):
Density-based Algorithm: DBSCAN

A cluster $C \subseteq D$ satisfies the following conditions:

1. $\forall p, q : \text{If } p \in C \text{ and } q \text{ is density-reachable from } p \text{ then } q \in C.$
Density-based Algorithm: DBSCAN

A cluster $C \subseteq D$ satisfies the following conditions:

1. $\forall p, q :$ If $p \in C$ and $q$ is density-reachable from $p$ then $q \in C$.

2. $\forall p, q :$ $p$ is density-connected to $q$.
   There is a point $o$ such that both, $p$ and $q$ are density-reachable from $o$. 
Density-based Algorithm: DBSCAN

Overall cluster procedure:

1. Select unclassified point \( p \in D \).

2. Construct \( \varepsilon \)-neighborhood \( N_\varepsilon(p) \).

3. If \( p \) is a core point
   - Then Insert \( N_\varepsilon(p) \) into new cluster \( C \).
   - Recursively analyze the \( \varepsilon \)-neighborhoods of \( q \in N_\varepsilon(p) \) and insert all density-reachable points into \( C \).

   Else Classify \( p \) as noise.
Density-based Algorithm: DBSCAN
Density-based Algorithm: DBSCAN
Density-based Algorithm: DBSCAN
Density-based Algorithm: DBSCAN
Density-based Algorithm: DBSCAN
Density-based Algorithm: DBSCAN

- Core point
- Border point

TIR'05 Sep. 9th, 2005 Stein/Busch
Density-based Algorithm: DBSCAN

- Core point
- Border point
Density-based Algorithm: DBSCAN

- Core point
- Border point
- Noise point
Density-based Algorithm: DBSCAN

- Core point
- Border point
- Noise point
Density-based Algorithm: DBSCAN

- Noise point
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust

- **Definite majority decision (agglomeration):**

- **Definite majority decision (node changes cluster):**
Density-based Algorithm: MajorClust

- Definite majority decision (agglomeration):
  ![Definite Majority Decision Diagram]

- Indefinite majority decision:
  ![Indefinite Majority Decision Diagram]
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Density-based Algorithm: MajorClust
Analysis I (low-dimensional)

Geometrical Data—map of the Caribbean Islands (approx. 20,000 points) :
Analysis I (low-dimensional)

Geometrical Data—map of the Caribbean Islands (approx. 20,000 points):

DBSCAN:

$\varepsilon = 3.0$, $\text{MinPts} = 3$

$\varepsilon = 5.0$, $\text{MinPts} = 4$

$\varepsilon = 10.0$, $\text{MinPts} = 5$
Analysis I (low-dimensional)

The problem of choosing a good \( \varepsilon \)-value in DBSCAN.

\[ \varepsilon = 3.0, \text{MinPts} = 3 \]

Two separate clusters are found.

Clusters are merged.
Analysis I (low-dimensional)

Geometrical Data—map of the Caribbean Islands  (approx. 20,000 points) :

MajorClust:

Initialization
Analysis I (low-dimensional)

The problem of a global analysis (no $\varepsilon$-neighborhood restriction) in MajorClust.
Analysis II (high-dimensional)

Document categorization with the Reuters corpus.

- 1000 documents
- 10 categories: politics, culture, economics, etc.
- Uniformly distributed, exclusive membership
- >10,000 dimensions
Analysis II (high-dimensional)

DBSCAN requires embedding of data in low-dimensional space.

Classification results ($F$-Measure) over dimensionality:

![Graph showing classification results over dimensionality.](image)

- **MajorClust (original data)**
- **MajorClust (embedded data)**
- **DBSCAN (embedded data)**

Number of dimensions, (Stress)

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

0 2 (52.1) 3 (49.1) 4 (44.3) 5 (43.5) 6 (40.7) 7 (37.6) 8 (35.1) 9 (34.2) 10 (11.6) 11 (10.8) 12 (10.2) 13 (9.6)

F-Measure
Analysis (runtime)

Runtime-behavior on the geometrical data:

Note:

The embedding of data in a low-dimensional space (MDS) is computationally very expensive:

I. e., most cluster algorithms will be faster than DBSCAN + MDS.
Analysis (runtime)

DBSCAN employs the $R$-tree data structure for region queries, which constructs minimum bounding regions for inserted points:

"Existing methods are outperformed on on average by a simple sequential scan, if the number of dimensions exceeds around 10."

[Weber 99, Gionis/Indyk/Motwani 99-04]
## Summary

An alternative categorization scheme:

<table>
<thead>
<tr>
<th>Analysis strategy</th>
<th>Cluster approach</th>
<th>Recovery characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hierarchical</td>
<td>revocable</td>
</tr>
<tr>
<td>relative comparison based on two items</td>
<td>iterative</td>
<td>revocable</td>
</tr>
<tr>
<td>absolute comparison based on $k$ items</td>
<td>density-based</td>
<td>revocable</td>
</tr>
<tr>
<td>relative comparison based on $k$ items</td>
<td>meta-search controlled</td>
<td>revocable</td>
</tr>
</tbody>
</table>

Orthogonal to this scheme is the concept for similarity computation:

- distance (neighborhood) analysis in low-dimensional space
- similarity predicate in arbitrary (high-dimensional) space
Summary

The strengths and weaknesses of density-based cluster algorithms can be explained with the dimensionality of the data.

- DBSCAN usually outperforms other cluster algorithms on low-dimensional data.
- MajorClust usually outperforms other cluster algorithms on high-dimensional data, in particular in the document categorization field.

Current work:

How fingerprints can be utilized for efficient region queries in high-dimensional spaces.