Unsupervised Learning
Density-based Methods
Road Map

1. DBSCAN
2. OPTICS
3. Subspace Clustering
The Principle

- Regard clusters as dense regions in the data space separated by regions of low density

- Major features
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need of density parameters as termination condition

- Several interesting studies
  - **DBSCAN**: Ester, et al. (KDD’96)
  - **DENCLUE**: Hinneburg & D. Keim (KDD’98)
  - **CLIQUE**: Agrawal, et al. (SIGMOD’98) (more grid-based)
The neighborhood within a radius $\varepsilon$ of a given object is called the $\varepsilon$-neighborhood of the object.

If the $\varepsilon$-neighborhood of an object contains at least a minimum number, $\text{MinPts}$, of objects than the object is called a core object.

Example: $\varepsilon = 1 \text{ cm}$, $\text{MinPts}=3$.

$m$ and $p$ are core objects because each is in an $\varepsilon$-neighborhood containing at least 3 points.
An object $p$ is directly density-reachable from object $q$ if $p$ is within the $\varepsilon$-neighborhood of $q$ and $q$ is a core object.

**Example:**
- $q$ is directly density-reachable from $m$
- $m$ is directly density-reachable from $p$
- and vice versa
Density-Reachable Objects

- An object $p$ is **density-reachable** from object $q$ with respect to $\varepsilon$ and $\text{MinPts}$ if there is a chain of objects $p_1, \ldots, p_n$ where $p_1 = q$ and $p_n = p$ such that $p_{i+1}$ is directly reachable from $p_i$ with respect to $\varepsilon$ and $\text{MinPts}$.

- **Example:**
  - $q$ is density-reachable from $p$ because $q$ is directly density-reachable from $m$ and $m$ is directly density-reachable from $p$.
  - $p$ is not density-reachable from $q$ because $q$ is not a core object.
An object \( p \) is **density-connected** to object \( q \) with respect to \( \varepsilon \) and \( \text{MinPts} \) if there is an object \( O \) such as both \( p \) and \( q \) are density reachable from \( O \) with respect to \( \varepsilon \) and \( \text{MinPts} \).

**Example:**

\( p, q \) and \( m \) are all density connected.
DBSCAN Algorithm

- Searches for clusters by checking the $\varepsilon$-neighborhood of each point in the database.

- If the $\varepsilon$-neighborhood of a point $p$ contains more than $\text{MinPts}$, a new cluster with $p$ as a core object is created.

- DBSCAN iteratively collects directly density reachable objects from these core objects. Which may involve the merge of a few density-reachable clusters.

- The process terminates when no new point can be added to any cluster.
DBSCAN Example

MinPts=4
DBSCAN Example
Exercise
Distance

<table>
<thead>
<tr>
<th>( \text{dist}(d_i, d_j) )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>8.5</td>
<td>3.6</td>
<td>7.1</td>
<td>7.2</td>
<td>8.1</td>
<td>2.2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>6.1</td>
<td>4.2</td>
<td>5</td>
<td>4.1</td>
<td>3.2</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5</td>
<td>1.4</td>
<td>2</td>
<td>7.3</td>
<td>6.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>3.6</td>
<td>4.1</td>
<td>7.2</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1.4</td>
<td>6.7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>5.4</td>
<td>5.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>7.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.
Road Map

1. DBSCAN

2. OPTICS

3. Subspace Clustering
Why DBSCAN is not enough?

- Very different local densities may be needed to reveal clusters in different regions.
- Clusters $A, B, C_1, C_2,$ and $C_3$ cannot be detected using one global density parameter.
- A global density parameter can detect either $A, B, C$ or $C_1, C_2, C_3$.

**Solutions**
- Use hierarchical clustering, but
  - Single link effect
  - Hard to interpret
- Use OPTICS.
OPTICS Principle

- Produce a special order of the database
  - with respect to its density-based clustering structure
  - containing information about every clustering level of the data set (up to a generating distance $\varepsilon$)

- Which information to use?
The **core-distance** of an object \( p \) is the smallest \( \epsilon' \) that makes \( p \) a core object.

- If \( p \) is not a core object, the core distance of \( p \) is **undefined**.

**Example** (\( \epsilon, \text{MinPts}=5 \))
- \( \epsilon' \) is the core distance of \( p \)
- It is the distance between \( p \) and the fourth closest object.

The **reachability-distance** of an object \( q \) with respect to object \( p \) is:

\[
\text{Max}(\text{core-distance}(p), \text{Euclidian}(p,q))
\]

**Example**
- Reachability-distance \( (q_1,p) = \text{core-distance}(p) = \epsilon' \)
- Reachability-distance \( (q_2,p) = \text{Euclidian}(q_2,p) \)
OPTICS Algorithm

- Creates an ordering of the objects in the database and stores for each object its:
  - Core-distance
  - Distance reachability from the closest core object from which an object have been directly density-reachable

- This information is sufficient for the extraction of all density-based clustering with respect to any distance $\epsilon'$ that is smaller than $\epsilon$ used in generating the order
Illustration of Cluster Ordering

Reachability-distance

undefined

\( \varepsilon \)

\( \varepsilon' \)

Cluster-order of the objects
Road Map

1. DBSCAN
2. OPTICS
3. Subspace Clustering
The Curse of Dimensionality

Let’s take an example of one dimensional data

We add a second dimension

We add a third dimension
The Curse of Dimensionality

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless (equidistance)
Subspace Clustering

- Extension to attribute selection
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces

(a) Dimension a

(b) Dimension b

(c) Dimension c

(a) Dims a & b

(b) Dims b & c

(c) Dims a & c
Subspace Clustering

- How to find subspace clusters effectively and efficiently?

- We are going to see two approaches
  - Dimension-growth subspace clustering
  - Frequent pattern-based clustering
CLIQUE (CLustering in QUEst) was the first algorithm proposed for dimension growth subspace clustering in high-dimensional space.

- Start at single-dimensional subspaces and grow upward to higher dimensional ones.

- CLIQUE partitions each dimension like a grid structure and determines whether a cell is dense based on the number of points it contains.

- CLIQUE is an integration of grid-based and density-based methods.
CLIQUE

- Partition the d-dimensional data space into non overlapping rectangular units (done in 1-D for each partition)
- Identify dense units
- A unit is dense if the fraction of total data points contained in it exceeds an input model parameter
CLIQUE

- The subspaces representing dense regions are intersected to form a candidate search space in which dense units of higher dimensionality may exist.

- Why does CLIQUE confine its search for dense units of higher dimensionality to the intersection of the dense units in the subspaces?
The property adapted by CLIQUE states:
- If a k-dimensional unit is dense, then so are its projections in (k-1) dimensional space.

Generate potential or candidate dense units in k-dimensional space from dense units found in (k-1) dimensional space.

The resulting space searched is much smaller than the original space.

The dense units are then examined to determine clusters.
CLIQUE

**Strength**
- Automatically finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
- Insensitive to the order of records in input and does not presume some canonical data distribution
- Scales linearly with the size of input and has good scalability as the number of dimensions in the data increases

**Weakness**
- The accuracy of the clustering result may be degraded at the expense of simplicity of the method
Frequent Pattern-based Clustering

- Frequent pattern mining leads to the discovery of interesting associations and correlations among data objects.

- The frequent patterns discovered may also indicate clusters.

- Well suited for high dimensional data.
  - Rather than growing clusters dimension by dimension, we grow sets of frequent items.
  - Lead to clusters descriptions.
Example: Frequent term-based text

- Documents contain terms
- Extract terms
  - Parsing
  - Stemming
- Each document can be represented as a set of terms
- Consider each term as a dimension
- The dimension space will be very high

The dimension space can be referred as: **term vector space**
Documents are clustered based on the frequent terms they contain.

Consider only the low-dimensional frequent term sets as "cluster candidates".

Frequent term set is not a cluster but a description of a cluster.

A cluster consists of documents containing all the terms of the frequent term set.

Example: Frequent term-based text

- News, education, sport

- Science, Computer

Cluster 1

Cluster 2
Example: Frequent term-based text

- How to select a good subset of the set of all frequent term sets?

- Let
  - \( F_i \) be a set of frequent term sets
  - \( \text{Cov}(F_i) \) be the set of documents covered by \( F_i \)

- Find a well-selected subset \( F_1, F_2, \ldots, F_k \), of all frequent term sets

- **Principle**
  - (1) the selected subset should cover all the documents to be clustered
    \[
    \sum_{i=1}^{k} \text{Cov}(F_i) = D
    \]
  - (2) the overlap between any two partitions \( F_i \) and \( F_j \) for \( i \neq j \) should be minimized (e.g., using entropy)

- This approach automatically generates cluster description. In traditional methods, an additional step is required to describe the resulting clusters.
Summary

- Density-based Clustering find clusters with arbitrary shapes
- Handles noise
- Handles High Dimensional Data