Clustering

What is Clustering?

- Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data.

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters

- Clustering is unsupervised classification: no predefined classes

Types of Data in Cluster Analysis

A Categorization of Major Clustering Methods

Partitioning Methods

Hierarchical Methods

Typical applications

- As a stand-alone tool to get insight into data distribution
- As a preprocessing step for other algorithms

Use cluster detection when you suspect that there are natural groupings that may represent groups of customers or products that have a lot in common.

When there are many competing patterns in the data, making it hard to spot a single pattern, creating clusters of similar records reduces the complexity within clusters so that other data mining techniques are more likely to succeed.
Examples of Clustering Applications

- **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.
- **Land use**: Identification of areas of similar land use in an earth observation database.
- **Insurance**: Identifying groups of motor insurance policy holders with a high average claim cost.
- **City-planning**: Identifying groups of houses according to their house type, value, and geographical location.
- **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults.

Clustering definition

- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that:
  - data points in one cluster are more similar to one another (high intra-class similarity).
  - data points in separate clusters are less similar to one another (low inter-class similarity).
- Similarity measures: e.g. Euclidean distance if attributes are continuous.

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Notion of a Cluster is Ambiguous

- Initial points.
- Two Clusters.
- Six Clusters.
- Four Clusters.
Clustering

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods

Data Matrix

- Represents n objects with p variables (attributes, measures)
- A relational table

\[
\begin{bmatrix}
  x_{11} & \ldots & x_{1f} & \ldots & x_{1p} \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  x_{i1} & \ldots & x_{if} & \ldots & x_{ip} \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  x_{n1} & \ldots & x_{nf} & \ldots & x_{np}
\end{bmatrix}
\]

Dissimilarity Matrix

- Proximities of pairs of objects
- \(d(i,j)\): dissimilarity between objects i and j
- Nonnegative
- Close to 0: similar

\[
\begin{bmatrix}
  0 & & & \\
  d(2,1) & 0 & & \\
  d(3,1) & d(3,2) & 0 & \\
  \vdots & \vdots & \ddots & \vdots \\
  d(n,1) & d(n,2) & \ldots & 0
\end{bmatrix}
\]

Type of data in clustering analysis

- Continuous variables
- Binary variables
- Nominal and ordinal variables
- Variables of mixed types
Continuous variables

- To avoid dependence on the choice of measurement units the data should be standardized.

Standardize data
- Calculate the mean absolute deviation:
  \[ s_f = \frac{1}{n} (|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|) \]
- where \( m_f = \frac{1}{n} (x_{1f} + x_{2f} + \ldots + x_{nf}) \)
- Calculate the standardized measurement (z-score)
  \[ z_{if} = \frac{x_{if} - m_f}{s_f} \]
- Using mean absolute deviation is more robust than using standard deviation. Since the deviations are not squared the effect of outliers is somewhat reduced but their z-scores do not become too small; therefore, the outliers remain detectable.

Similarity/Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Euclidean distance is probably the most commonly chosen type of distance. It is the geometric distance in the multidimensional space:
  \[ d(i,j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} \]
  - Properties
    - \( d(i,j) \geq 0 \)
    - \( d(i,i) = 0 \)
    - \( d(i,j) = d(j,i) \)
    - \( d(i,j) \leq d(i,k) + d(k,j) \)

*City-block (Manhattan) distance*. This distance is simply the sum of differences across dimensions. In most cases, this distance measure yields results similar to the Euclidean distance. However, note that in this measure, the effect of single large differences (outliers) is dampened (since they are not squared).

\[ d(i,j) = |x_{1i} - x_{1j}| + |x_{2i} - x_{2j}| + \ldots + |x_{pi} - x_{pj}| \]
- The properties stated for the Euclidean distance also hold for this measure.

*Manhattan distance* = distance if you had to travel along coordinates only.

L₂-norm:
\[ \text{dist}(x,y) = \sqrt{(4^2+3^2)} = 5 \]
L₁-norm:
\[ \text{dist}(x,y) = 4 + 3 = 7 \]
Similarity/Dissimilarity Between Objects

- **Minkowski distance.** Sometimes one may want to increase or decrease the progressive weight that is placed on dimensions on which the respective objects are very different. This measure enables to accomplish that and is computed as:

\[
d(i, j) = \left(\sum_{q=1}^{p} |x_{iq} - x_{jq}|^q\right)^{1/q}
\]

If we have some idea of the relative importance that should be assigned to each variable, then we can weight them and obtain a weighted distance measure.

\[
d(i, j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + \cdots + w_p(x_{ip} - x_{jp})^2}
\]

**Binary Variables**

- A binary variable has only two states: 0 or 1

- A binary variable is **symmetric** if both of its states are equally valuable, that is, there is no preference on which outcome should be coded as 1.

- A binary variable is **asymmetric** if the outcome of the states are not equally important, such as positive or negative outcomes of a disease test.

- Similarity that is based on symmetric binary variables is called **invariant similarity.**

**A contingency table for binary data**

<table>
<thead>
<tr>
<th></th>
<th>Object ( i )</th>
<th></th>
<th>Object ( j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>a+b</td>
</tr>
<tr>
<td>0</td>
<td>c+d</td>
<td></td>
<td>p</td>
</tr>
</tbody>
</table>

- Simple matching coefficient (invariant, if the binary variable is symmetric):

\[
d(i, j) = \frac{b + c}{a + b + c + d}
\]

- Jaccard coefficient (noninvariant if the binary variable is asymmetric):

\[
d(i, j) = \frac{b + c}{a + b + c}
\]
Dissimilarity between Binary Variables

Example

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
<th>Fever</th>
<th>Cough</th>
<th>Test-1</th>
<th>Test-2</th>
<th>Test-3</th>
<th>Test-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>M</td>
<td>Y</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Mary</td>
<td>F</td>
<td>Y</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td>Jim</td>
<td>M</td>
<td>Y</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

- gender is a symmetric attribute
- the remaining attributes are asymmetric binary
- let the values Y and P be set to 1, and the value N be set to 0

Jaccard coefficient

\[
\begin{align*}
  d(\text{jack,mary}) &= \frac{0+1}{2+0+1} = 0.33 \\
  d(\text{jack,jim}) &= \frac{1+1}{1+1+1} = 0.67 \\
  d(\text{jim,mary}) &= \frac{1+2}{1+1+2} = 0.75
\end{align*}
\]

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green

  Method 1: simple matching
  - \( m \): # of matches, \( p \): total # of variables

  \[
  d(i,j) = \frac{p-m}{p}
  \]

  Method 2: use a large number of binary variables
  - creating a new binary variable for each of the \( M \) nominal states

Ordinal Variables

- On ordinal variables order is important
  - e.g. Gold, Silver, Bronze

  Can be treated like continuous
  - the ordered states define the ranking 1,...,\( M_f \)
  - replacing \( x_{if} \) by their rank \( r_{if} \in \{1,...,M_f\} \)
  - map the range of each variable onto \([0, 1] \) by replacing i-th object in the f-th variable by

  \[
  z_{if} = \frac{r_{if} - 1}{M_f - 1}
  \]

  compute the dissimilarity using methods for continuous variables

Variables of Mixed Types

- A database may contain several/all types of variables
  - continuous, symmetric binary, asymmetric binary, nominal and ordinal.

  One may use a weighted formula to combine their effects.

  \[
  d(i,j) = \frac{\sum_{f=1}^{M} \delta_{ij} (\epsilon_f)^2 \delta(f)}{\sum_{f=1}^{M} \delta(f)}
  \]

  \( \delta_{ij} = 0 \) if \( x_{if} \) is missing or \( x_{if}=x_{jf}=0 \) and the variable \( f \) is asymmetric binary

  \( \delta_{ij} = 1 \) otherwise

  continuous and ordinal variables \( d_{ij} \): normalized absolute distance

  binary and nominal variables \( d_{ij} = 0 \) if \( x_{if}=x_{jf} \); otherwise \( d_{ij} = 1 \)
Clustering

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods

Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: Based on connectivity and density functions. Able to find clusters of arbitrary shape. Continues growing a cluster as long as the density of points in the neighborhood exceeds a specified limit.
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Clustering

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database $D$ of $n$ objects into a set of $k$ clusters
- Given a $k$, find a partition of $k$ clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: $k$-means and $k$-medoids algorithms
    - $k$-means: Each cluster is represented by the center of the cluster
    - $k$-medoids or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster
The K-Means Clustering Method

- **Given** \( k \), the k-means algorithm is implemented in 4 steps:
  1. Partition objects into \( k \) nonempty subsets
  2. Compute centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each object to the cluster with the nearest seed point.
  4. Go back to Step 2; stop when no more new assignment.

Comments on the K-Means Method

- **Strengths & Weaknesses**
  - Relatively efficient: \( O(tk) \), where \( n \) is \# objects, \( k \) is \# clusters, and \( t \) is \# iterations. Normally, \( k, t \ll n \).
  - Often terminates at a local optimum
  - Applicable only when mean is defined
  - Need to specify \( k \), the number of clusters, in advance
  - Sensitive to noise and outliers as a small number of such points can influence the mean value
  - Not suitable to discover clusters with non-convex shapes

Importance of Choosing Initial Centroids
The K-Medoids Clustering Method

- Find *representative* objects, called *medoids*, in clusters.
- **PAM** (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering.
- **CLARA** (Kaufmann & Rousseeuw, 1990)
- **CLARANS** (Ng & Han, 1994): Randomized sampling

Getting $k$ Right

- Try different $k$, looking at the change in the average distance to centroid, as $k$ increases.
- Average falls rapidly until right $k$, then changes little.

PAM (Partitioning Around Medoids)

- **PAM** (Kaufman and Rousseeuw, 1987)
- Use real object to represent the cluster.
  - Select *$k$ representative objects* arbitrarily.
  - For each pair of a non-selected object $h$ and a selected object $i$, calculate the total swapping cost $TC_{ih}$.
  - Select pair of $i$ and $h$ which corresponds to the minimum $TC_{ih}$.
    - If min. $TC_{ih} < 0$, $i$ is replaced by $h$.
    - Then assign each non-selected object to the most similar representative object.
  - Repeat steps 2-3 until there is no change.
**PAM Clustering: Total swapping cost**  
\[ T_{C_{ih}} = \sum_j C_{jih} \]

- **i, t:** medoids
- **h:** medoid candidate
- **j:** a point
- \( C_{jih} \): swapping cost due to j

\[ C_{jih} = d(j, h) - d(j, i) \]

\[ C_{jih} = d(j, t) - d(j, i) \]

---

**Comments on PAM**

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean.
- **PAM** works effectively for small data sets, but does not scale well for large data sets.

---

**CLARA (Clustering LARge Applications)**

- **CLARA** (Kaufmann and Rousseeuw in 1990) draws a sample of the dataset and applies PAM on the sample in order to find the medoids.
- If the sample is representative the medoids of the sample should approximate the medoids of the entire dataset.
- To improve the approximation, multiple samples are drawn and the best clustering is returned as the output.
- The clustering accuracy is measured by the average dissimilarity of all objects in the entire dataset.
  - Experiments show that 5 samples of size 40+2k give satisfactory results.

---

**CLARA (Clustering LARge Applications)**

- **Strengths** and **Weaknesses:**
  - Deals with larger data sets than **PAM**
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased.
CLARANS ("Randomized" CLARA)

- CLARANS (A Clustering Algorithm based on Randomized Search) (Ng and Han’94)
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of $k$ medoids.
- Two nodes are neighbours if their sets differ by only one medoid.
- Each node can be assigned a cost that is defined to be the total dissimilarity between every object and the medoid of its cluster.
- The problem corresponds to search for a minimum on the graph.
- At each step, all neighbours of current node are searched; the neighbour which corresponds to the deepest descent in cost is chosen as the next solution.

For large values of $n$ and $k$, examining $k(n-k)$ neighbours is time consuming.

At each step, CLARANS draws a sample of neighbours to examine.

Note that CLARA draws a sample of nodes at the beginning of search; therefore, CLARANS has the benefit of not confining the search to a restricted area.

If the local optimum is found, CLARANS starts with a new randomly selected node in search for a new local optimum. The number of local optima to search for is a parameter.

It is more efficient and scalable than both PAM and CLARA; returns higher quality clusters.

---

Clustering

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods

Hierarchical Clustering

- Use distance matrix as clustering criteria.
- These methods work by grouping data into a tree of clusters.
- There are two types of hierarchical clustering:
  - Agglomerative: bottom-up strategy
  - Divisive: top-down strategy
- Does not require the number of clusters as an input, but needs a termination condition, e. g., could be the desired number of clusters or a distance threshold for merging.
Hierarchical Clustering

Agglomerative hierarchical clustering

Clustering result: dendrogram

Linkage rules (1)

- **Single link (nearest neighbour).** The distance between two clusters is determined by the distance of the two closest objects (nearest neighbours) in the different clusters.
  - This rule will, in a sense, string objects together to form clusters, and the resulting clusters tend to represent long "chains."

\[ d_{24} \]
**Linkage rules (2)**

- **Complete link (furthest neighbour).** The distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbours").
- This method usually performs quite well in cases when the objects actually form naturally distinct "clumps." If the clusters tend to be somehow elongated or of a "chain" type nature, then this method is inappropriate.

![Complete link diagram](image)

**Linkage rules (3)**

- **Pair-group average.** The distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters. This method is also very efficient when the objects form natural distinct "clumps," however, it performs equally well with elongated, "chain" type clusters.

\[
\text{Distance} = \frac{d_{14} + d_{15} + d_{24} + d_{25} + d_{34} + d_{35}}{6}
\]

**Linkage rules (4)**

- **Pair-group centroid.** The distance between two clusters is determined as the distance between centroids.

![Pair-group centroid diagram](image)

**AGNES (Agglomerative Nesting)**

- Use the Single-Link method and the dissimilarity matrix.
- Repeatedly merge nodes that have the least dissimilarity
  - merge C1 and C2 if objects from C1 and C2 give the minimum Euclidean distance between any two objects from different clusters.
- Eventually all nodes belong to the same cluster
DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Inverse order of AGNES
  - All objects are used to form one initial cluster
  - The cluster is split according to some principle
    - the maximum Euclidean distance between the closest neighbouring objects in different clusters
  - Eventually each node forms a cluster on its own

More on Hierarchical Clustering

- Do not scale well: time complexity of at least $O(n^2)$, where $n$ is the number of total objects
- Can never undo what was done previously
- It's nice that you get a hierarchy instead of an amorphous collection of groups
- Don't need to specify $k$. If you want $k$ groups, just cut the $(k-1)$ longest links
- In general give better quality clusters than k-means like methods

More on Hierarchical Clustering

- Integration of hierarchical with distance-based clustering
- **BIRCH**: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD '96)
  - A tree is built that captures needed information to perform clustering
  - Introduces two new concepts
    - **Clustering Feature** (contains info about a cluster)
    - **Clustering Feature Tree** which are used to summarize cluster representation

BIRCH algorithm

- **BIRCH**: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD '96)
  - A tree is built that captures needed information to perform clustering
  - Introduces two new concepts
    - **Clustering Feature** (contains info about a cluster)
    - **Clustering Feature Tree** which are used to summarize cluster representation
BIRCH - Clustering Feature Vector

A clustering feature is a triplet summarizing information about sub-clusters of objects. It registers crucial measurements for computing clusters in a compact form.

Clustering Feature: \( \overline{CF} = (N, LS, SS) \)

- **N**: Number of data points
- **LS**: Linear sum \( \sum_{i=1}^{N} X_i \)
- **SS**: Square sum \( \sum_{i=1}^{N} X_i^2 \)

\( \overline{CF} = (5, (16,30),(54,190)) \)

(3,4) (2,6) (4,5) (4,7) (3,8)

Notes on Birch

- A leaf node represents a cluster.
- A sub-cluster in a leaf node must have a diameter no greater than a given threshold \( T \).
- A point is inserted into the leaf node (cluster) to which is closer.
- When one item is inserted into a cluster at the leaf node, the restriction \( T \) (for the corresponding sub-cluster) must be satisfied. The corresponding CF must be updated.
- If there is no space on the node the node is split.

BIRCH algorithm

- Incrementally construct a CF tree, a hierarchical data structure for multiphase clustering.
- **Phase 1**: scan DB to build an initial in-memory CF tree.
  - If threshold condition is violated
    - If there is room to insert – Insert point as a single cluster
    - If not
      - Leaf node split: take two farthest CFs and create two leaf nodes, put the remaining CFs (including the new one) into the closest node.
      - Update CF for non-leaves. Insert new non-leaf entry into parent node.
      - We may have to split the parent as well. Split the root increases tree height by one.
  - If not
    - Insert point into the closest cluster.
- **Phase 2**: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree.
Some Comments on Birch

- It can be shown that CF vectors can be stored and calculated incrementally and accurately as clusters are merged.
- Experiments have shown that scales linearly with the number of objects.
- Finds a good clustering with a single scan and improves the quality with a few additional scans.
- Handles only numeric data, and sensitive to the order of the data record.
- Better suited to find spherical clusters.

CURE (Clustering Using REpresentatives )

- Proposed by Guha, Rastogi & Shim, 1998
  - Uses multiple representative points to evaluate distance between clusters.
  - Representative points are well-scattered objects for the cluster and are shrunk towards the centre of the cluster.
    - (adjusts well to arbitrary shaped clusters; and avoids single-link effect)
  - At each step, the two clusters with the closest pair of representative points are merged.

Cure: The Algorithm

- Draw random sample \( s \) (to ensure data fits memory)
- Partition sample to \( p \) partitions with size \( s/p \) (to speed up algorithm)
- Partially cluster partitions into \( s/pq \) clusters (using hierarchical alg.)
- Eliminate outliers
  - If a cluster grows too slowly or if is very small at the end, eliminate it.
- Cluster partial clusters.
- Label data (cluster the entire database using \( c \) representative points for each cluster)

Partial Clustering

- Each cluster is represented by \( c \) representative points
- The r. p. are chosen to be far from each other
- The r.p. are shrunk toward the mean (the centroid) of the cluster (for \( \alpha = 1 \) all r.p. are shrunk to the centroid)
- The two clusters with the closest pair of r.p. are merged to form a new cluster and new r.p. are chosen (Hierarchical clustering)
CURE: Data Partitioning and Clustering

- \( s = 50 \)
- \( p = 2 \)
- \( s/p = 25 \)
- \( s/pq = 5 \)

Partitioning the sample data

Cure: Shrinking Representative Points

Further cluster the partial clustering

- Shrink the multiple representative points towards the gravity center by a fraction of \( \alpha \). (helps dampen the effects of outliers)
- Multiple representatives capture the shape of the cluster

CURE Approach

- Having several representative points per cluster allows CURE to adjust well to the geometry of nonspherical shapes.
- Shrinking the scattered points toward the mean by a factor of \( \alpha \) gets rid of surface abnormalities and mitigates the effect of outliers.
- Results with large datasets indicate that CURE scales well.
- Time complexity is \( O(n^2 \log n) \)
DBSCAN algorithm

- **Density-based Alg**: based on local connectivity and density functions

- **Major features**:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan

DBSCAN: Density-Based Clustering

- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerable higher density of points than outside of the cluster

DBSCAN: Density Concepts (1)

- **Density**: the minimum number of points within a certain distance of each other.

- **Two parameters**:
  - 
  - **Eps**: Maximum radius of the neighborhood
  - **MinPts**: Minimum number of points in an Eps-neighborhood of that point

- **Core Point**: object with at least MinPts objects within a radius ‘Eps-neighborhood’

DBSCAN: Density Concepts (2)

- **Directly Density-Reachable**: A point \( p \) is directly density-reachable from a point \( q \) with respect to \( Eps, MinPts \) if
  1) \( p \) belongs to \( NEps(q) \)
  2) core point condition: \( |NEps (q)| >= MinPts \)

(a DDR point needs to be close to a core point but it does not need to be a core point itself, if not it is a border point)

\( p \)
\( q \)

MinPts = 5
Eps = 1 cm
**DBSCAN: Density Concepts (2)**

- **Density-reachable:**
  - A point $p$ is density-reachable from a point $q$ wrt. $\varepsilon$, $\text{MinPts}$ if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that $p_{i+1}$ is directly density-reachable from $p_i$.

- **Density-connected:**
  - A point $p$ is density-connected to a point $q$ wrt. $\varepsilon$, $\text{MinPts}$ if there is a point $o$ such that both, $p$ and $q$ are density-reachable from $o$ wrt. $\varepsilon$ and $\text{MinPts}$.

**DBSCAN: Cluster definition**

- A cluster is defined as a **maximal set of density-connected points**

- A cluster has a core set of points very close to a large number of other points (core points) and then some other points (border points) that are sufficiently close to at least one core point.

**DBSCAN: The Algorithm**

- **Arbitrary select** a point $p$

  - If $p$ is not a core point, no points are density-reachable from $p$ and DBSCAN visits the next point of the database.

  - If $p$ is a core point, a cluster is formed.
    - Retrieve all points density-reachable from $p$ wrt $\varepsilon$ and $\text{MinPts}$.
    - Continue the process until all of the points have been processed.

(\textit{it is possible that a border point could belong to two clusters. Such point will be assigned to whichever cluster is generated first})

**Comments on DBSCAN**

- For each core point which is not in a cluster
  - Explore its neighbourhood in search for every density reachable point
  - For each neighbourhood point explored
    - If it is a core point $\rightarrow$ further explore it
    - If it is not a core point $\rightarrow$ assign to the cluster and do not explore it

- The fact that a cluster is composed by the maximal set of points that are density-connect it is a property (and therefore a consequence) of the method
**DBScan**

- Experiments have shown DBScan to be faster and more precise than CLARANS
- Expected time complexity $O(n \log n)$

**Clustering Summary**

- Unsupervised method two find groups of instances
- Many approaches
  - partitioning
  - hierarchical
- Solution evaluation is difficult
  - Manual inspection by experts
  - Benchmarking on existing labels
  - Cluster quality measures
    - (measure the “tightness” or “purity” of clusters)
    - distance measures
    - high similarity within a cluster, low across clusters

**References**

- *Data Mining: Concepts and Techniques*, Jiawei Han, Micheline Kamber (Morgan Kaufmann - 2000)
- *Data Mining: Introductory and Advanced Topics*, Margaret Dunham (Prentice Hall, 2002)
- *Clustering Web Search Results*, Iwona Bialynicka-Birula, [http://www.di.unipi.it/~iwona/Clustering.ppt](http://www.di.unipi.it/~iwona/Clustering.ppt)

Solutions nearly always come from the direction you least expect, which means there’s no point in trying to look in that direction because it won’t be coming from there.

Douglas Adams