Lecture notes

Knowledge Discovery in Databases
Summer Semester 2012

Lecture 8: Clustering II

Lecture: Dr. Eirini Ntoutsi
Tutorials: Erich Schubert

http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_(KDD_I)
Sources

• Previous KDD I lectures on LMU (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Jörg Sander, Matthias Schubert, Arthur Zimek)

• Tan P.-N., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison-Wesley, 2006

• Jiawei Han, Micheline Kamber and Jian Pei, *Data Mining: Concepts and Techniques, 3rd ed.*, Morgan Kaufmann, 2011.
Outline

- Introduction
- A categorization of major clustering methods
- Hierarchical methods
- Density based methods
- Grid based methods (next lecture)
- Model-based methods (next lecture)
- Things you should know
- Homework/tutorial
Major clustering methods I

- **Partitioning approaches:**
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS

- **Hierarchical approaches:**
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CHAMELEON

- **Density-based approaches:**
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue
Major clustering methods II

- **Grid-based approaches:**
  - Based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

- **Model-based approaches:**
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB

- **Frequent pattern-based approaches:**
  - Based on the analysis of frequent patterns
  - Typical methods: pCluster

- **User-guided or constraint-based approaches:**
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
Outline

• Introduction

• A categorization of major clustering methods
  • Hierarchical methods
  • Density based methods
  • Grid based methods (next lecture)
  • Model-based methods (next lecture)
• Things you should know
• Homework/tutorial
Hierarchical methods idea

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree-like diagram that records the sequences of merges or splits
  - The height at which two clusters are merged in the dendrogram reflects their distance
Strengths of Hierarchical Clustering

• Do not have to assume any particular number of clusters
  – Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

• They may correspond to meaningful taxonomies
  – Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Knowledge Discovery in Databases I: Clustering II
Hierarchical vs Partitioning

Partitioning clustering

Partitioning algorithms typically have global objectives

Hierarchical clustering algorithms typically have local objectives

Nested clusters

Dendrogram
Hierarchical clustering methods

- Two main types of hierarchical clustering
  - **Agglomerative:**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or \( k \) clusters) left
    - e.g., AGNES
  - **Divisive:**
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are \( k \) clusters)
    - e.g., DIANA

- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time
Agglomerative clustering algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward

1. Compute the proximity matrix
2. Let each data point be a cluster
3. Repeat
4. Merge the two closest clusters
5. Update the proximity matrix
6. Until only a single cluster remains

- Key points:
  - the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms (single link, complete link, .....)
  - the update of the proximity matrix due to cluster merges
Starting situation

• Start with clusters of individual points and a proximity matrix

```
<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>...</th>
<th>p12</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Proximity matrix
Intermediate situation I

- After some merging steps, we have some clusters

Proximity matrix

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Knowledge Discovery in Databases I: Clustering II
Intermediate situation II

- We want to merge the two closest clusters (C₂ and C₅) and update the proximity matrix.

![Diagram of clusters and proximity matrix]

Proximity matrix:

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Cluster diagram with points p1 to p12]

Knowledge Discovery in Databases I: Clustering II
The question is “How do we update the proximity matrix?” Or, in other words, what is the similarity between two clusters?
Measures of inter-cluster similarity I

A variety of different measures:

- Single link (or MIN)
- Complete link (or MAX)
- Group average
- Distance between centroids
- Distance between medoids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity matrix
Measures of inter-cluster similarity II

- Single link (or MIN): smallest distance between an element in one cluster and an element in the other, i.e.,

\[
dis_{sl}(C_i, C_j) = \min_{x,y} \left\{ d(x, y) \mid x \in C_i, y \in C_j \right\}
\]
Complete link (or MAX): largest distance between an element in one cluster and an element in the other, i.e.,

\[
\text{dis}_{cl}(C_i, C_j) = \max_{x,y} \left\{ d(x, y) \mid x \in C_i, y \in C_j \right\}
\]
Measures of inter-cluster similarity IV

- Group average: average distance between an element in one cluster and an element in the other, i.e.,

$$\text{dis}_{\text{avg}}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x, y)}{|C_i||C_j|}$$

\[
\begin{array}{ccccccc}
  & p1 & p2 & p3 & \ldots & p12 \\
p1 &   &   &   &   &   \\
p2 &   &   &   &   &   \\
p3 &   &   &   &   &   \\
\ldots &   &   &   &   &   \\
p12 &   &   &   &   &   \\
\end{array}
\]

Proximity matrix
Measures of inter-cluster similarity V

- Centroid: distance between the centroids of two clusters, i.e.,

\[
d_{\text{centroids}}(C_i, C_j) = d(c_i, c_j)
\]

\[
c_m = \frac{\sum_{i=1}^{n} p_i}{n}
\]

Proximity matrix
Example

Dataset (6 2D points)

<table>
<thead>
<tr>
<th>Point</th>
<th>x Coordinate</th>
<th>y Coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>0.40</td>
<td>0.53</td>
</tr>
<tr>
<td>p2</td>
<td>0.22</td>
<td>0.38</td>
</tr>
<tr>
<td>p3</td>
<td>0.35</td>
<td>0.32</td>
</tr>
<tr>
<td>p4</td>
<td>0.26</td>
<td>0.19</td>
</tr>
<tr>
<td>p5</td>
<td>0.08</td>
<td>0.41</td>
</tr>
<tr>
<td>p6</td>
<td>0.45</td>
<td>0.30</td>
</tr>
</tbody>
</table>

Distance matrix (Euclidean distance)

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>p6</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>0.00</td>
<td>0.24</td>
<td>0.22</td>
<td>0.37</td>
<td>0.34</td>
<td>0.23</td>
</tr>
<tr>
<td>p2</td>
<td>0.24</td>
<td>0.00</td>
<td>0.15</td>
<td>0.20</td>
<td>0.14</td>
<td>0.25</td>
</tr>
<tr>
<td>p3</td>
<td>0.22</td>
<td>0.15</td>
<td>0.00</td>
<td>0.15</td>
<td>0.28</td>
<td>0.11</td>
</tr>
<tr>
<td>p4</td>
<td>0.37</td>
<td>0.20</td>
<td>0.15</td>
<td>0.00</td>
<td>0.29</td>
<td>0.22</td>
</tr>
<tr>
<td>p5</td>
<td>0.34</td>
<td>0.14</td>
<td>0.28</td>
<td>0.29</td>
<td>0.00</td>
<td>0.39</td>
</tr>
<tr>
<td>p6</td>
<td>0.23</td>
<td>0.25</td>
<td>0.11</td>
<td>0.22</td>
<td>0.39</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Single link distance (MIN): discussion

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.
Single link distance (MIN): strengths

- Can handle non-elliptical shapes
Single link distance (MIN): limitations

- Sensitive to noise and outliers
- Chain like clusters
Complete link distance (MAX): discussion

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>p6</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>0.00</td>
<td>0.24</td>
<td>0.22</td>
<td>0.37</td>
<td>0.34</td>
<td>0.23</td>
</tr>
<tr>
<td>p2</td>
<td>0.24</td>
<td>0.00</td>
<td>0.15</td>
<td>0.20</td>
<td>0.14</td>
<td>0.25</td>
</tr>
<tr>
<td>p3</td>
<td>0.22</td>
<td>0.15</td>
<td>0.00</td>
<td>0.15</td>
<td>0.28</td>
<td>0.11</td>
</tr>
<tr>
<td>p4</td>
<td>0.37</td>
<td>0.20</td>
<td>0.15</td>
<td>0.00</td>
<td>0.29</td>
<td>0.22</td>
</tr>
<tr>
<td>p5</td>
<td>0.34</td>
<td>0.14</td>
<td>0.28</td>
<td>0.29</td>
<td>0.00</td>
<td>0.39</td>
</tr>
<tr>
<td>p6</td>
<td>0.23</td>
<td>0.25</td>
<td>0.11</td>
<td>0.22</td>
<td>0.39</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Nested clusters

Dendrogram
Complete link distance (MAX): strengths

- Less susceptible to noise and outliers
Complete link distance (MAX): limitations

- Tends to break large clusters
- Biased towards spherical clusters
Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

- Determined by all pairs of points in the two clusters

![Nested clusters Dendrogram](image)
Group average: strengths and limitations

- Compromise between Single and Complete Link

- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards spherical clusters
Ward’s method

- Ward’s method or Ward's minimum variance method
- The proximity between two clusters is measured in terms of the increase in SSE that results from merging the two clusters
  - At each step, merge the pair of clusters that leads to minimum increase in total inter-cluster variance after merging.
  - Similarly to k-Means, tries to minimize the sum of square distances of points from their cluster centroids
- Similar to group average if distance between points is distance squared

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>p6</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>0.00</td>
<td>0.24</td>
<td>0.22</td>
<td>0.37</td>
<td>0.34</td>
<td>0.23</td>
</tr>
<tr>
<td>p2</td>
<td>0.24</td>
<td>0.00</td>
<td>0.15</td>
<td>0.20</td>
<td>0.14</td>
<td>0.25</td>
</tr>
<tr>
<td>p3</td>
<td>0.22</td>
<td>0.15</td>
<td>0.00</td>
<td>0.15</td>
<td>0.28</td>
<td>0.11</td>
</tr>
<tr>
<td>p4</td>
<td>0.37</td>
<td>0.20</td>
<td>0.15</td>
<td>0.00</td>
<td>0.29</td>
<td>0.22</td>
</tr>
<tr>
<td>p5</td>
<td>0.34</td>
<td>0.14</td>
<td>0.28</td>
<td>0.29</td>
<td>0.00</td>
<td>0.39</td>
</tr>
<tr>
<td>p6</td>
<td>0.23</td>
<td>0.25</td>
<td>0.11</td>
<td>0.22</td>
<td>0.39</td>
<td>0.00</td>
</tr>
</tbody>
</table>

- Less susceptible to noise and outliers
- Biased towards spherical clusters

Nested clusters
Comparison of the different methods

- Single link (MIN)
- Complete link (MAX)
- Group average
- Ward’s method
Hierarchical methods: complexity

• $O(N^2)$ space since it uses the proximity matrix.
  – $N$ is the number of points.

• $O(N^3)$ time in many cases
  – There are $N$ steps and at each step the size, $N^2$, proximity matrix must be updated and searched
  – Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches
How to get a clustering from a dendrogram

- A dendrogram is a tree of clusters.
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.
Hierarchical clustering: overview

- No knowledge on the number of clusters
- Produces a hierarchy of clusters, not a flat clustering
- A single clustering can be obtained from the dendrogram

- Merging decisions are final
  - Once a decision is made to combine two clusters, it cannot be undone
- Lack of a global objective function
  - Decisions are local, at each step
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Breaking large clusters
  - Difficulty handling different sized clusters and convex shapes
- Inefficiency, especially for large datasets
Bisecting k-Means

- Hybrid methods: k-Means and hierarchical clustering
- Idea: first split the set of points into two clusters, select one of these clusters for further splitting, and so on, until k clusters.
- Pseudocode:

1. All data constitute one cluster ROOT.
2. The ROOT is partitioned in two clusters, its children, using K-Means for K=2.
3. In each subsequent iteration
   2.1. Choose among the leaf clusters the most inhomogeneous one,
   2.2. Partition it into two clusters with K-Means, K=2, until K leaf clusters are built.

Which cluster to split?
- e.g., the one with the largest SSE
- e.g., based on SSE and size

- Example:
Outline

• Introduction
• A categorization of major clustering methods
• Hierarchical methods

• Density based methods

• Grid based methods (next lecture)
• Model-based methods (next lecture)
• Things you should know
• Homework/tutorial
Density based clustering

• Clusters are regions of high density surrounded by regions of low density (noise)
• Clustering based on density (local cluster criterion), such as density-connected points
• Major features:
  – Discover clusters of arbitrary shape
  – Handle noise
  – One scan
  – Need 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 notion of density

- **Density:**
  - Density is measured locally in the Eps-neighborhood (or $\varepsilon$-neighborhood) of each point.
  - Density = number of points within a specified radius $\text{Eps}$ (point itself included).

- **Density depends on the specified radius**
  - In an extreme small radius, all points will have a density of 1 (only themselves).
  - In an extreme large radius, all points will have a density of $N$ (the size of the dataset).
• Consider a dataset $D$ of objects to be clustered

• Two parameters:
  – $\epsilon$ (or $\varepsilon$): Maximum radius of the neighbourhood
  – MinPts: Minimum number of points in an $\epsilon$-neighbourhood of that point

• $\epsilon$-neighborhood of a point $p$ in $D$
  – $N_{\epsilon}(p)$: $\{q \in D \mid \text{dist}(p,q) \leq \epsilon\}$

The $\epsilon$-neighborhood of $p$
Core points vs border points vs noise points

• Let D be a dataset. Given a radius parameter \( \text{Eps} \) and a density parameter \( \text{MinPts} \) we can distinguish between:

  - **Core points**
    A point is a core point if it has more than a specified number of points (\( \text{MinPts} \)) within a specified radius \( \text{Eps} \), i.e.,:
    \[
    |N_{\text{Eps}}(p) = \{q \mid \text{dist}(p,q) \leq \text{Eps}\}| \geq \text{MinPts}
    \]
    - These are points that are at the interior of a cluster.

  - **Border points**
    A border point has fewer than \( \text{MinPts} \) within \( \text{Eps} \), but it is in the neighborhood of a core point.

  - **Noise points**
    not a core point nor a border point.
Example

Noise Point

Border Point

Core Point

Eps = 1

MinPts = 4
Core, Border and Noise points

Original points

Point types: core, border and noise

Eps = 10, MinPts = 4
Direct reachability

- Directly density-reachable: A point $p$ is directly density-reachable from a point $q$ w.r.t. $Eps$, $MinPts$ if
  - $p$ belongs to $N_{Eps}(q)$
  - $q$ is a core point, i.e., $|N_{Eps}(q)| \geq MinPts$
Reachability

- Density-reachable:
  - A point $p$ is density-reachable from a point $q$ w.r.t. $Eps$, $MinPts$ if there is a chain of points $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that $p_{i+1}$ is directly density-reachable from $p_i$.
Connectivity

- Density-connected

  A point $p$ is density-connected to a point $q$ w.r.t. $Eps$, $MinPts$ if there is a point $o$ such that both, $p$ and $q$ are density-reachable from $o$ w.r.t. $Eps$ and $MinPts$.
• A cluster is a maximal set of density-connected points
DBSCAN algorithm

- Arbitrary select a point $p$
- Retrieve all points density-reachable from $p$ w.r.t. $Eps$ and $MinPts$.
- If $p$ is a core point, a cluster is formed.
- If $p$ is a border point, no points are density-reachable from $p$ and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.
DBSCAN pseudocode I

DBSCAN(Dataset DB, Real Eps, Integer MinPts)

// initially all objects are unclassified, // o.ClId = unclassified for all o ∈ DB

ClusterId := nextId(NOISE);
for i from 1 to |DB| do
    Object := DB.get(i);
    if Object.ClId = unclassified then
        if ExpandCluster(DB, Object, ClusterId, Eps, MinPts) then
            ClusterId:=nextId(ClusterId);
DBSCAN pseudocode II

```
ExpandCluster(DB, StartObject, ClusterId, Eps, MinPts): Boolean

seeds := RQ(StartObject, Eps);
if |seeds| < MinPts then // StartObject is not a core object
    StartObject.ClId := NOISE;
    return false;
else // else: StartObject is a core object
    forall o ∈ seeds do o.ClId := ClusterId;
    remove StartObject from seeds;
    while seeds ≠ Empty do
        select an object o from the set of seeds;
        Neighborhood := RQ(o, Eps);
        if |Neighborhood| ≥ MinPts then // o is a core object
            for i from 1 to |Neighborhood| do
                p := Neighborhood.get(i);
                if p.ClId in {UNCLASSIFIED, NOISE} then
                    if p.ClId = UNCLASSIFIED then
                        add p to the seeds;
                        p.ClId := ClusterId;
                    end if;
                end if;
            end for;
            remove o from the seeds;
        end if;
    end while;
end if
return true;
```
Complexity

• For a dataset D consisting of n points, the time complexity of DBSCAN is \( O(n \times \text{time to find points in the Eps-neighborhood}) \)

• Worst case \( O(n^2) \)

• In low-dimensional spaces \( O(n\log n) \);
  – efficient data structures (e.g., \( kd \)-trees) allow for efficient retrieval of all points within a given distance of a specified point
When DBSCAN works well?

- Resistant to noise
- Can handle clusters of different shapes and sizes
When DBSCAN does not work well?

- Varying densities
- High-dimensional data

Original points

(MinPts=4, Eps=9.75).

(MinPts=4, Eps=9.92)
DBSCAN: determining Eps and MinPts

- Idea is that for points in a cluster, their $k^{th}$ nearest neighbors are at roughly the same distance.
- Noise points have the $k^{th}$ nearest neighbor at farther distance.
- So, plot sorted distance of every point to its $k^{th}$ nearest neighbor.
We will discuss OPTICS next time ....
Outline

• Introduction

• A categorization of major clustering methods

• Hierarchical methods

• Density based methods

• Grid based methods (next lecture)

• Model-based methods (next lecture)

• Things you should know

• Homework/tutorial
Things you should know

• Hierarchical methods
  – Agglomerative, divisive
  – Cluster comparison measures

• Bisecting k-Means

• Density based methods
  – DBSCAN
Homework/ Tutorial

**Tutorial:** Tutorial this Thursday on clustering

**Homework:**
- Try hierarchical clustering in Weka, Elki
- Implement your own hierarchical clusterer
  - Try the different cluster similarity measures
- Try density based clustering in Elki, Weka
- Implement your own DBSCAN
  - Experiment with different Eps, MinPts parameters

**Suggested reading:**
- Tan P.-N., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison-Wesley, 2006 (Chapter 8).