Algorithmics
Clustering and Seriation

Jaak Vilo
2013 Spring
Seriation?
Clustering

• Grouping objects by similarity

• Take all data and ask – what are typical examples, groups in data
Topics

• What is clustering
• Hierarchical clustering
• K-means
  – K-medoids
  – EM
• SOM
• Density based methods (DBSCAN)
• Conceptual clustering
• ...
• ...
Unsupervised vs. Supervised

Find groups inherent to data (clustering)

Find a “classifier” for known classes
Unsupervised vs. Supervised

Find groups inherent to data (clustering)

Find a “classifier” for known classes
Clustering

• An old problem
• Many methods

• No single best “suites all needs” method
## Vehicle Example

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Top speed km/h</th>
<th>Colour</th>
<th>Air resistance</th>
<th>Weight Kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>220</td>
<td>red</td>
<td>0.30</td>
<td>1300</td>
</tr>
<tr>
<td>V2</td>
<td>230</td>
<td>black</td>
<td>0.32</td>
<td>1400</td>
</tr>
<tr>
<td>V3</td>
<td>260</td>
<td>red</td>
<td>0.29</td>
<td>1500</td>
</tr>
<tr>
<td>V4</td>
<td>140</td>
<td>gray</td>
<td>0.35</td>
<td>800</td>
</tr>
<tr>
<td>V5</td>
<td>155</td>
<td>blue</td>
<td>0.33</td>
<td>950</td>
</tr>
<tr>
<td>V6</td>
<td>130</td>
<td>white</td>
<td>0.40</td>
<td>600</td>
</tr>
<tr>
<td>V7</td>
<td>100</td>
<td>black</td>
<td>0.50</td>
<td>3000</td>
</tr>
<tr>
<td>V8</td>
<td>105</td>
<td>red</td>
<td>0.60</td>
<td>2500</td>
</tr>
<tr>
<td>V9</td>
<td>110</td>
<td>gray</td>
<td>0.55</td>
<td>3500</td>
</tr>
</tbody>
</table>
Vehicle Clusters

![Graph showing clusters of vehicles based on weight and top speed. The graph includes clusters for Sports cars, Medium market cars, and Lorries.]
Terminology

Object or data point

label

cluster

feature space

Sports cars

Medium market cars

Lorries

feature

Top speed [km/h]

Weight [kg]

feature
Motivation: Why Clustering?

**Problem:** Identify (a small number of) groups of similar objects in a given (large) set of object.

**Goals:**

- Find representatives for homogeneous groups → **Data Compression**
- Find “natural” clusters and describe their properties → ”natural” **Data Types**
- Find suitable and useful grouping → ”useful” **Data Classes**
- Find unusual data object → **Outlier Detection**
What is Cluster Analysis?

• Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

Intra-cluster distances are minimized

Inter-cluster distances are maximized
Clustering – it’s “easy” (for humans)
- **Edge-Detection** (advantage to smooth contours)
- **Texture clustering**
Clustering cont...
Types

- Partitioning vs Hierarchical
- Exclusive vs Overlapping vs Fuzzy membership
- Complete vs partial
## 2-D data

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>
## 5-Dimensional

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>
5-dimensional...
<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>
Exercise: cluster 2-D data

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>
### 2-D data

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

![2-D plot](image)

**2-D plot**

- Series 1

Data points: (2, 5), (3, 7), (6, 8)
Ingredients

• Similarity/distance between objects

• Find “closest” elements
Distance measures: which two profiles are similar to each other?

- Euclidean, Manhattan etc.
- Correlation, angle, etc.
- Rank correlation
- Time warping
Distance measures

• How to formally describe which objects are “close” to each other, and which are not
• More than one way to define distances.
• Distance is a metric, if
  – $d(X,X) = 0$
  – $d(X,Y) = d(Y,X) \geq 0$
  – $d(A,B) \leq d(A,C) + d(C, B)$
Some standard distance measures

Euclidean distance
\[ d(f, g) = \sqrt{\sum_{i=1}^{c} (f_i - g_i)^2} \]

Euclidean squared
\[ d(f, g) = \sum_{i=1}^{c} (f_i - g_i)^2 \]

Manhattan (city-block)
\[ d(f, g) = \sum_{i=1}^{c} |f_i - g_i| \]

Average distance
\[ d(f, g) = \frac{1}{c} \sum_{i=1}^{c} (f_i - g_i)^2 \]
Pearson correlation

\[ d(f, g) = 1 - \frac{\sum_{i=1}^{c} (f_i - \bar{f})(g_i - \bar{g})}{\sqrt{\sum_{i=1}^{c} (f_i - \bar{f})^2 \sum_{i=1}^{c} (g_i - \bar{g})^2}} \]

If means of each column are 0, then it becomes:

\[ d(f, g) = 1 - \frac{\sum_{i=1}^{c} f_i g_i}{\sqrt{\sum_{i=1}^{c} f_i^2 \sum_{i=1}^{c} g_i^2}} = 1 - \cos \Theta \]
Chord distance

\[ d(f, g) = \sqrt{2(1 - \frac{\sum_{i=1}^{c} f_i g_i}{\sqrt{\sum_{i=1}^{c} f_i^2 \sum_{i=1}^{c} g_i^2}})} \]

\[ d(f, g) = \sqrt{2(1 - \cos \Theta)} \]

Euclidean distance between two vectors whose length has been normalized to 1.
Rank correlation

\[ d(f, g) = 1 - \frac{6 \sum_{i=1}^{c} (\text{rank}_{fi} - \text{rank}_{gi})}{c(c^2 - 1)} \]

Rank - smallest has rank 1, next 2, etc.
Equal values have rank that is average of the ranks

\[ f = \begin{array}{cccccc} 3 & 17 & 12 & 12 & 8 \\ \text{rank} = & 1 & 5 & 3.5 & 3.5 & 2 \end{array} \]
Hierarchical clustering
Hierarchical clustering

1. All against all distance matrix
2. Linkage strategy – identify closest clusters and merge
1. All against all distance matrix
2. Linkage strategy – identify closest clusters and merge
Hierarchical clustering

1. All against all distance matrix
2. Linkage strategy – identify closest clusters and merge

Performance: \( O(dn^2) \)
Hierarchical clustering

Keep joining together two closest clusters by using the:

- Minimum distance => Single linkage
- Maximum distance => Complete linkage
- Average distance => Average linkage (UPGMA, WPGMA)

\[ \min \{ d_{ai,bj} \mid ai \text{ in } A, bj \text{ in } B \} \]

\[ \max \{ d_{ai,bj} \mid ai \text{ in } A, bj \text{ in } B \} \]

\[ \text{avg}\{ d_{ai,bj} \mid ai \text{ in } A, bj \text{ in } B \} \]
Hierarchical clustering

• Calculate all pairwise distances
  – and assign each object into a singleton cluster

• While more than 1 cluster
  – select smallest distance
  – merge the two clusters
  – update the changed distances after merger
Update distances

- Merge $C_a$, $C_b$ into $C$
- Re-calculate all distances $D(C_i, C)$
Merge Ca, Cb into C

• \( D(C_i, C) = \min\{ D(C_i, Ca) , D(C_i, Cb) \} \)
  – Single link; Minimal distance

• \( D(C_i, C) = \max\{ D(C_i, Ca) , D(C_i, Cb) \} \)
  – Complete link; Maximum distance

• \( D(C_i, C) = \text{average}\{ D(C_i, Ca) , D(C_i, Cb) \} \)
  – \( \frac{n_a}{n_a+n_b} \times D(C_i, Ca) + \frac{n_b}{n_a+n_b} \times D(C_i, Cb) \)
  – UPGMA – Unweighted Pair Group Method Average
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix
How to Define Inter-Cluster Similarity

- **MIN**
- **MAX**
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></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>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></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>
<td></td>
</tr>
</tbody>
</table>

Proximity Matrix
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></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>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
UPGMA

\[ D(C_k, (C_i \cup C_j)) \leftarrow \frac{|C_i|}{|C_i| + |C_j|} \cdot D(C_k, C_i) + \frac{|C_j|}{|C_i| + |C_j|} \cdot D(C_k, C_j) \]

WPGMA: \[ D(C_k, (C_i \cup C_j)) \leftarrow \frac{1}{2} \left( D(C_k, C_i) + D(C_k, C_j) \right) \]

Single-Linkage: \[ D(C_k, (C_i \cup C_j)) \leftarrow \min \{ D(C_k, C_i), D(C_k, C_j) \} \]
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></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>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Input data

<table>
<thead>
<tr>
<th>$i$</th>
<th>$x_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(3.5, 0.9)</td>
</tr>
<tr>
<td>2</td>
<td>(1.5, 3.0)</td>
</tr>
<tr>
<td>3</td>
<td>(1.7, 5.5)</td>
</tr>
<tr>
<td>4</td>
<td>(2.1, 8.0)</td>
</tr>
<tr>
<td>5</td>
<td>(3.0, 8.4)</td>
</tr>
<tr>
<td>6</td>
<td>(5.1, 5.2)</td>
</tr>
<tr>
<td>7</td>
<td>(8.4, 6.5)</td>
</tr>
<tr>
<td>8</td>
<td>(8.3, 2.2)</td>
</tr>
<tr>
<td>9</td>
<td>(8.3, 1.4)</td>
</tr>
</tbody>
</table>

Input data visualized
Euclidean distances

<table>
<thead>
<tr>
<th>$d_{l_2}(\cdot, \cdot)$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0.0</td>
<td>2.9</td>
<td>4.9</td>
<td>7.2</td>
<td>7.5</td>
<td>4.6</td>
<td>7.4</td>
<td>5.0</td>
<td>4.8</td>
</tr>
<tr>
<td>$x_2$</td>
<td>2.9</td>
<td>0.0</td>
<td>2.5</td>
<td>5.0</td>
<td>5.6</td>
<td>4.2</td>
<td>7.7</td>
<td>6.8</td>
<td>7.0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>4.9</td>
<td>2.5</td>
<td>0.0</td>
<td>2.5</td>
<td>3.2</td>
<td>3.4</td>
<td>6.8</td>
<td>7.4</td>
<td>7.8</td>
</tr>
<tr>
<td>$x_4$</td>
<td>7.2</td>
<td>5.0</td>
<td>2.5</td>
<td>0.0</td>
<td>1.0</td>
<td>4.1</td>
<td>6.5</td>
<td>8.5</td>
<td>9.1</td>
</tr>
<tr>
<td>$x_5$</td>
<td>7.5</td>
<td>5.6</td>
<td>3.2</td>
<td>1.0</td>
<td>0.0</td>
<td>3.8</td>
<td>5.7</td>
<td>8.2</td>
<td>8.8</td>
</tr>
<tr>
<td>$x_6$</td>
<td>4.6</td>
<td>4.2</td>
<td>3.4</td>
<td>4.1</td>
<td>3.8</td>
<td>0.0</td>
<td>3.5</td>
<td>4.4</td>
<td>5.0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>7.4</td>
<td>7.7</td>
<td>6.8</td>
<td>6.5</td>
<td>5.7</td>
<td>3.5</td>
<td>0.0</td>
<td>4.3</td>
<td>5.1</td>
</tr>
<tr>
<td>$x_8$</td>
<td>5.0</td>
<td>6.8</td>
<td>7.4</td>
<td>8.5</td>
<td>8.2</td>
<td>4.4</td>
<td>4.3</td>
<td>0.0</td>
<td>0.8</td>
</tr>
<tr>
<td>$x_9$</td>
<td>4.8</td>
<td>7.0</td>
<td>7.8</td>
<td>9.1</td>
<td>8.8</td>
<td>5.1</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Average linkage hierarchical clustering
Hierarchical clustering
Running Cme for hierarchical clustering

Distances 10 attrib.
Distances 100 attrib.
Distances 10,100, 1000 dim

Time in seconds

Data size 10K 15K 20K

1 minute

5 minutes
Time

• $O( n^2 )$ distances

• $n-1$ times merge
  – select smallest distance
  – update all distances to new cluster
Design any heat-map coloring scheme

To create a customised colour scheme, you will need to first create a colour palette, and then, create the discretisation.

1. Click on the colour squares in the colour map on the left in the order of the colours in the desired palette, from left to right.
To create a familiar Green-Black-Red palette, click on a green square, followed by a black and a red square. You'll see the palette being formed below the colour cube map.

2. Pick a discretisation type (linear/exponential/harmonic/histogram), and enter the number of slots into which you'd like to discretise the palette.

3. Click on the "CREATE DISCRETISATION" link. You'll see the palette broken up into chunks of colour. Linear discretisation will contain equal size chunks, and others will have proportionately varied colour stretches.

4. Click on the "Save" button to save this palette in the main colour options screen.
Heat map color schema design
Limits of standard clustering

• Hierarchical clustering is (very) good for visualization (first impression) and browsing
• Speed for modern data sets remains relatively slow (minutes or even hours)
• ArrayExpress database needs some faster analytical tools
• Hard to predict number of clusters
  (=>Unsupervised)
6200 genes, 80 exp.

Monitor size 1600x1200 pixels

Laptop: 800x600
6200 genes, 80 exp.

“COLLAPSE”

Monitor size 1600x1200

Laptop: 800x600

75 subtrees

Developed and implemented in Expression Profiler in October 2000 by Jaak Vilo
Figure 1: A biological case study with VisHiC. (A) gene expression matrix and annotated dendrogram with significant clusters; (B) mitochondrion cluster (ID:30189) (C) muscle cluster (ID:25531) (D) annotation box of the mitochondrion cluster, appears when moving the mouse over the dendrogram (E) detailed view of the muscle cluster with heatmap, dendrogram and lineplot (F) table with functional enrichments, including clusters 30189 and 25531. The data presented in the figure comprises microarray measurements of the heart tissue of cardiovascular patients with left ventricular assist device. VisHiC reveals clusters with expected relevant annotations, e.g. mitochondrion, muscle tissue and extracellular matrix (see Results section).
Fast Approximate Hierarchical Clustering using Similarity Heuristics

Hierarchical clustering is applied in gene expression data analysis, number of genes can be 20000+

Hierarchical clustering:
Each subtree is a cluster.
Hierarchy is built by iteratively joining two most similar clusters into a larger one.
Fast Hierarchical Clustering

Avoid calculating all $O(n^2)$ distances:

- Estimate distances
- Use pivots
- Find close objects
- Cluster with partial information

Distances from one pivot

Distances from two pivots

Distances from two pivots

<table>
<thead>
<tr>
<th>$i$</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>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(x_i, x_1)$</td>
<td>0.0</td>
<td>2.9</td>
<td>4.9</td>
<td>7.2</td>
<td>7.5</td>
<td>4.6</td>
<td>7.4</td>
<td>5.0</td>
<td>4.8</td>
</tr>
<tr>
<td>$d(x_i, x_6)$</td>
<td>4.6</td>
<td>4.2</td>
<td>3.4</td>
<td>4.1</td>
<td>3.8</td>
<td>0.0</td>
<td>3.5</td>
<td>4.4</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Distances from two pivots

<table>
<thead>
<tr>
<th>$i$</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>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(x_i, x_1)$</td>
<td>0.0</td>
<td>2.9</td>
<td>4.9</td>
<td>7.2</td>
<td>7.5</td>
<td>4.6</td>
<td>7.4</td>
<td>5.0</td>
<td>4.8</td>
</tr>
<tr>
<td>$d(x_i, x_6)$</td>
<td>4.6</td>
<td>4.2</td>
<td>3.4</td>
<td>4.1</td>
<td>3.8</td>
<td>0.0</td>
<td>3.5</td>
<td>4.4</td>
<td>5.0</td>
</tr>
</tbody>
</table>

$\varepsilon$-grid $\varepsilon = 1.5$

Here we use Chebyshev distance (maximum of differences)

By triangle inequality we get:
Euclidean distance in original plot cannot be smaller than Chebyshev distance here
Epsilon Grid Order (EGO)

1) Datapoints sorted according to EGO order
2) Each point is compared with the later points until one hypercube away
Epsilon Grid Order (EGO)

1) Datapoints sorted according to EGO order
2) Each point is compared with the later points until one hypercube away
   e.g. $\bar{x}_2$ is compared with the points in the marked hypercubes
Representative-Based Clustering

• Aims at finding a set of objects among all objects (called representatives) in the data set that best represent the objects in the data set. Each representative corresponds to a cluster.

• The remaining objects in the data set are then clustered around these representatives by assigning objects to the cluster of the closest representative.

Remarks:

1. The popular $k$-medoid algorithm, also called PAM, is a representative-based clustering algorithm; K-means also shares the characteristics of representative-based clustering, except that the representatives used by k-means not necessarily have to belong to the data set.

2. If the representative do not need to belong to the dataset we call the algorithms prototype-based clustering. K-means is a prototype-based clustering algorithm
K-means, K-medoids, ...

- Partition the data points into K groups

- Each group is centered around it’s mean or medoid

- Mean is an “abstract point”
- Medoid: most central object
K-means

1. Guess K centres
K-means

1. Guess K centres

2. Assign obj to clusters
K-means

1. Guess K centres

2. Assign obj to clusters

3. Move C to gravity centres
K-means
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 seed points as the 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.
New centers -

Center of gravity for a cluster

Cluster - objects closest to a center

- Start clustering by choosing K centers randomly
- Most distant centers
- Iterate clustering step until no cluster changes
- Deterministic, might get "stuck" in local minimum

Only the graph--lines:

Out of 15 requested clusters there were 15 non-empty clusters
K-means

• Finds local optimum
  – vary many times with random start
  – make an “educated guess” to start with
    • e.g. sample the data, perform hierarchical clustering, select K “centers”.
Comments on *K-Means*

**Strength**

- *Relatively efficient*: $O(t*k*n*d)$, where $n$ is # objects, $k$ is # clusters, and $t$ is # iterations, $d$ is the # dimensions. Usually, $d, k, t \ll n$; in this case, K-Mean’s runtime is $O(n)$.
- Storage only $O(n)$—in contrast to other representative-based algorithms, only computes distances between centroids and objects in the dataset, and not between objects in the dataset; therefore, the distance matrix does not need to be stored.
- Easy to use; well studied; we know what to expect
- Finds *local optimum of the SSE fitness function*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Implicitly uses a fitness function (finds a local minimum for SSE see later) --- does not waste time computing fitness values

**Weakness**

- Applicable only when *mean* is defined --- what about categorical data?
- Need to specify $k$, the *number* of clusters, in advance
- Sensitive to *outliers*
- Not suitable to discover clusters with non-convex shapes
- Sensitive to initialization; bad initialization might lead to bad results.
K-medoids

• Choose the cluster center to be one of the existing objects.

• **Why?**

• If more complex data or distance measure the “Real” center could not be found easily

• What is the mean of categorical data?
  – yellow, red, pink?

• Instead of trying to “invent” – use one of the existing objects, whatever the distance measure
Convex Shape Cluster

• Convex Shape: if we take two points belonging to a cluster then all the points on a direct line connecting these two points must also in the cluster.

• Shape of K-means/K-medoids clusters are convex polygons $\subseteq$ Convex Shape.

• Shapes of clusters of a representative-based clustering algorithm can be computed as a Voronoi diagram for the set of cluster representatives.

• Voronoi cells are always convex, but there are convex shapes that are different from those of Voronoi cells.
Voronoi Diagram for a Representative-based Clustering

Each cell contains one representatives, and every location within the cell is closer to that sample than to any other sample.

A Voronoi diagram divides the space into such cells.

Voronoi cells define cluster boundary!

Cluster Representative (e.g. medoid/centroid)
Representative-Based Clustering ... (Continued)
Objective of RSC: Find a subset $O_R$ of $O$ such that the clustering $X$ obtained by using the objects in $O_R$ as representatives minimizes $q(X)$; $q$ is an objective/fitness function.
EM — Expectation Maximization

- EM — A popular iterative refinement algorithm
- An extension to k-means
  - Assign each object to a cluster according to a weight (prob. distribution)
  - New means/covariances are computed based on weighted measures
- General idea
  - Starts with an initial estimate of the parameter vector
  - Iteratively rescores the patterns against the mixture density produced by the parameter vector
  - The rescored patterns are used to update the parameter updates
  - Patterns belonging to the same cluster, if they are placed by their scores in a particular component
- Algorithm converges fast but may not be in global optima
The EM (Expectation Maximization) Algorithm

- Initially, randomly assign $k$ cluster centers
- Iteratively refine the clusters based on two steps
  - Expectation step: assign each data point $X_i$ to cluster $C_i$ with the following:
    \[
    P(X_i \in C_k) = p(C_k|X_i) = \frac{p(C_k)p(X_i|C_k)}{p(X_i)},
    \]
  - Maximization step:
    - Estimation of model parameters
    \[
    m_k = \frac{1}{N} \sum_{i=1}^{N} \frac{X_i P(X_i \in C_k)}{\sum_j P(X_i \in C_j)}.
    \]

• Mldemos - [http://mldemos.epfl.ch/](http://mldemos.epfl.ch/)

• MeV - [http://www.tm4.org/mev/](http://www.tm4.org/mev/)
12488 genes total divided into 40 clusters. Used datasets:

1. Transcriptional profiling of mouse inner cell mass of the blastocyst, primordial germ cells and cultured pluripotent stem cells (E-GEOD-35416)
101 genes total. Used datasets:

1. Transcriptional profiling of mouse inner cell mass of the blastocyst, primordial germ cells and cultured pluripotent stem cells (E-GEOD-35416)
MEM – visual output

Robust Rank Aggregation

QUERY: POU5F1

Red: similar to query
Blue: distant / not similar to query
MEM – visual output

QUERY: POU5F1

Red: similar to query
Blue: distant / not similar to query
Query of OCT4 (POU5F1)

Robust Rank Aggregation

StdDev < 0.29
• “Neural network map”
Self Organising Maps (SOM)

MxN matrix of neurons, each representing “a cluster”
Object X is put to Wi, to which it is most similar.
Wi and its near surrounding is changed to resemble X more
Train, train, train…

Problem - there is no clear objective function to map
d-dimensional data to 2 dimensions…
Motivation: The Problem Statement

The problem is how to find out semantics relationship among lots of information without manual labor

- How do I know, where to put my new data in, if I know nothing about information’s topology?
- When I have a topic, how can I get all the information about it, if I don’t know the place to search them?
Motivation: The Idea

- Computer know automatically information classification and put them together

  Input Pattern 1

  Input Pattern 2

  Input Pattern 3
Motivation: The Idea

- Text objects must be automatically produced with semantics relationships
Self-Organizing Maps: Origins

Self-Organizing Maps

- Ideas first introduced by C. von der Malsburg (1973), developed and refined by T. Kohonen (1982)
- Neural network algorithm using unsupervised competitive learning
- Primarily used for organization and visualization of complex data
- Biological basis: ‘brain maps’

Teuvo Kohonen
Self-Organizing Maps

SOM - Architecture

- Lattice of neurons (‘nodes’) accepts and responds to set of input signals
- Responses compared; ‘winning’ neuron selected from lattice
- Selected neuron activated together with ‘neighbourhood’ neurons
- Adaptive process changes weights to more closely resemble inputs
Self-Organizing Maps

SOM - Architecture

2d array of neurons

Weighted synapses

Set of input signals

(connected to all neurons in lattice)
Self-Organizing Maps

SOM – Result Example

Classifying World Poverty

Helsinki University of Technology

‘Poverty map’ based on 39 indicators from World Bank statistics (1992)
Choose an input vector $x$ from the training set

In computer texts are shown as a frequency distribution of one word.

**A Text Example:**

Self-organizing maps (SOMs) are a data visualization technique invented by Professor Teuvo Kohonen which reduce the dimensions of data through the use of self-organizing neural networks. The problem that data visualization attempts to solve is that humans simply cannot visualize high dimensional data as is so technique are created to help us understand this high dimensional data.
Finding a Winner

• (iii) Find the best-matching neuron \( \omega(x) \), usually the neuron whose weight vector has smallest Euclidean distance from the input vector \( x \)

\[
d_{a,b} = \sqrt{\sum_{i} (a_i - b_i)^2}
\]

• The winning node is that which is in some sense ‘closest’ to the input vector
Weight Update

- SOM Weight Update Equation
  \[ w_j(t + 1) = w_j(t) + \mu(t) \lambda_{\omega(x)}(j,t) \left[ x - w_j(t) \right] \]

- “The weights of every node are updated at each cycle by adding

- Current learning rate \times Degree of neighbourhood with respect to winner \times Difference between current weights and input vector

- to the current weights”

- Example of \( \mu(t) \)
- Example of \( \lambda_{\omega(x)}(j,t) \)

L. rate

\[ \text{No. of cycles} \]

-x-axis shows distance from winning node
-y-axis shows ‘degree of neighbourhood’ (max. 1)
Example: Self-Organizing Maps

Animal names and their attributes

<table>
<thead>
<tr>
<th></th>
<th>Dove</th>
<th>Hen</th>
<th>Duck</th>
<th>Goose</th>
<th>Owl</th>
<th>Hawk</th>
<th>Eagle</th>
<th>Fox</th>
<th>Dog</th>
<th>Wolf</th>
<th>Cat</th>
<th>Tiger</th>
<th>Lion</th>
<th>Horse</th>
<th>Zebra</th>
<th>Cow</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>has</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>likes to</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

A grouping according to similarity has emerged

[Teuvo Kohonen 2001] Self-Organizing Maps; Springer;
Web Document Clustering Using SOM

- The result of SOM clustering of 12088 Web articles
- The picture on the right: drilling down on the keyword “mining”
- Based on websom.hut.fi Web page
Cluster	
ing etc. algorithms

• Hierarchical clustering methods + visualisation
• K-means, Self Organising Maps (SOM)
• SOTA trees (Self Organising Maps + Tree)
• Fuzzy, EM (object can belong to several clusters)
• Graph theory (cliques, strongly connected components)
• Similarity search: X -> Y s.t. d(X,Y)< 0.3
• Model based (rediscover distributions)
• Planar embeddings, Multidimensional scaling
• Principal Component Analysis
• Correspondence analysis
• Independent Component Analysis
Other Clustering Methods

• PCA (Principal Component Analysis)
  – Also called SVD (Singular Value Decomposition)
  – Reduces dimensionality of gene expression space
  – Finds best view that helps separate data into groups

• Supervised Methods
  – SVM (Support Vector Machine)
  – Previous knowledge of which genes expected to cluster is used for training
  – Binary classifier uses ‘feature space’ and ‘kernel function’ to define a optimal ‘hyperplane’
  – Also used for classification of samples- ‘expression fingerprinting’ for disease classification
Dynamic changes in the transcriptome of mouse embryonic stem cells
Gene expression profiling results of the FunGenES Consortium

<table>
<thead>
<tr>
<th>Global Clusters</th>
<th>Description Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arranged by:</td>
<td></td>
</tr>
<tr>
<td>Experimental condition</td>
<td></td>
</tr>
<tr>
<td>Time series</td>
<td></td>
</tr>
<tr>
<td>2815 genes organized in 113 clusters across 67 distinct experimental conditions.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time Series</th>
<th>Description Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concise (50)</td>
<td></td>
</tr>
<tr>
<td>Analytical (200)</td>
<td></td>
</tr>
<tr>
<td>CGR8</td>
<td></td>
</tr>
<tr>
<td>E14T2a</td>
<td></td>
</tr>
<tr>
<td>Links to raw data</td>
<td></td>
</tr>
<tr>
<td>Genes organized in clusters according to time of expression during differentiation of ES cells.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specific Gene Classes</th>
<th>Description Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcription regulators (CGR8, E14T2a)</td>
<td></td>
</tr>
<tr>
<td>Unknown ESTs (CGR8, E14T2a)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression Waves</th>
<th>Description Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similarity cutoff 0.8</td>
<td></td>
</tr>
<tr>
<td>Similarity cutoff 0.85</td>
<td></td>
</tr>
<tr>
<td>Expression clusters of up or down-regulated genes during ES cell differentiation and their corresponding mirror image clusters.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pathway Animations</th>
<th>Description Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Pathway</td>
<td></td>
</tr>
<tr>
<td>FunGenES profiling data organized as articulated gene expression changes in KEGG pathways.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Study your Gene(s) of Interest</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter the gene names!</td>
<td></td>
</tr>
<tr>
<td>Study your genes on FunGenES data and other resources.</td>
<td></td>
</tr>
</tbody>
</table>
Genes in the dataset: 8211
Genes clustered on this page (correlation with patterns more than 0.8): 7178
Genes not clustered on this page (correlation with patterns less than 0.8): 1033

Global Clusters
Timeseries
All experiments

Time Series
Concise
Analytical
CGR8
EL4TG2a
Links to raw data

Specific Gene Classes
Transcription regulators (CGR8, EL4TG2a)
Unknown ESTs (CGR8, EL4TG2a)

Expression Waves
Similarity cutoff 0.8
Similarity cutoff 0.85

Pathway Animations
Select Pathway

Study your genelist
Visualize expression
giftviewer

Fungenes Internal Data
Fungenes data analysis (private)

Additional Links
Official Fungenes page
<table>
<thead>
<tr>
<th>Global Clusters</th>
<th>Time Series</th>
<th>Specific Gene Classes</th>
<th>Expression Waves</th>
<th>Pathway Animations</th>
<th>Study your genelist</th>
<th>Fungenes Internal Data</th>
<th>Additional Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeseries</td>
<td>Analytical</td>
<td>CGR8, E14TQ2a</td>
<td>Similarity cutoff 0.85</td>
<td>Select Pathway</td>
<td>Visualize expression</td>
<td>Fungenes data analysis (private)</td>
<td></td>
</tr>
<tr>
<td>All Experiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Genes in the dataset: 8211
Genes clustered on this page (correlation with patterns more than 0.8): 6013
Genes not clustered on this page (correlation with patterns less than 0.8): 1398

<table>
<thead>
<tr>
<th>Up</th>
<th>Down</th>
<th>Days up</th>
<th>Merge</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 day</td>
<td></td>
<td>2 days</td>
</tr>
<tr>
<td>ES expressed</td>
<td>(1610)</td>
<td>47</td>
<td></td>
<td>124</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 days</td>
<td></td>
<td>3 days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>540</td>
<td></td>
<td>213</td>
</tr>
<tr>
<td></td>
<td></td>
<td>363</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>207</td>
<td></td>
<td>6 days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td></td>
<td>7 days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>207</td>
<td></td>
<td>8 days</td>
</tr>
</tbody>
</table>

**Day of rise**: 214
The Curse of Dimensionality

(graphs adapted from Parsons et al. KDD Explorations 2004)

- 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—due to equi-distance

May 2, 2013

Data Mining: Concepts and Techniques 124
Why Subspace Clustering?
(adapted from Parsons et al. SIGKDD Explorations 2004)

- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces
Density based clustering
DBSCAN

- **DBSCAN** is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a **core point** if it has more than a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
    - A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
    - A **noise point** is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

\[ \text{current\_cluster\_label} \leftarrow 1 \]

\[ \text{for all core points do} \]

\[ \text{if the core point has no cluster label then} \]

\[ \text{current\_cluster\_label} \leftarrow \text{current\_cluster\_label} + 1 \]

\[ \text{Label the current core point with cluster label } \text{current\_cluster\_label} \]

\[ \text{end if} \]

\[ \text{for all points in the Eps-neighborhood, except } i^{th} \text{ the point itself do} \]

\[ \text{if the point does not have a cluster label then} \]

\[ \text{Label the point with cluster label } \text{current\_cluster\_label} \]

\[ \text{end if} \]

\[ \text{end for} \]

\[ \text{end for} \]
DBSCAN: Core, Border and Noise Points

Original Points

Point types: core, border and noise

Eps = 10, MinPts = 4
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
ELKI

From Wikipedia, the free encyclopedia

**ELKI** (for *Environment for DeveLoping KDD-Applications Supported by Index-Structures*) is a knowledge discovery in databases (KDD, "data mining") software framework developed for use in research and teaching by the database systems research unit of Professor Hans-Peter Kriegel at the Ludwig Maximilian University of Munich, Germany. It aims at allowing the development and evaluation of advanced data mining algorithms and their interaction with database index structures.

### Contents [hide]

1 Description
2 Awards
3 Included algorithms
4 Version history
5 Related applications
6 External links
7 References

### Description

The ELKI framework is written in [Java](https://java.com/) and built around a modular architecture. Most currently included algorithms belong to clustering, outlier detection[1] and database indexes. A key

### Environment for DeveLoping KDD-Applications Supported by Index-Structures

![Screenshot of ELKI 0.4 visualizing OPTICS cluster analysis.](https://example.com/screenshot.png)

**Developer(s)** Ludwig Maximilian University of Munich

**Stable release** 0.5.0 / June 30, 2012; 3 months ago

**Written in** Java

**Operating system** Microsoft Windows, Linux, Mac OS

**Platform** Java platform

**Type** Data mining

**License** AGPL (since version 2.4)
Summary

- **Cluster analysis** groups objects based on their similarity and has wide applications.
- Measure of similarity can be computed for various types of data.
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods.
- **Outlier detection** and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches.
- There are still lots of research issues on cluster analysis.
Summary: Goals of clustering

- Scalability
- Deal with different types of attributes
- Clusters of arbitrary shape
- Minimal expert knowledge on parameters
- Deal with noise
- Incremental and not dependent on order
- High dimensionality
- Constraints
- Interpretability and usability
- ...
Quality: What Is Good Clustering?

• A good clustering method will produce high quality clusters with
  – high intra-class similarity
  – low inter-class similarity

• The quality of a clustering result depends on both the similarity measure used by the method and its implementation

• The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns
Measure the Quality of Clustering

- **Dissimilarity/Similarity metric**: Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate “quality” function that measures the “goodness” of a cluster.
- The definitions of **distance functions** are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
  - the answer is typically highly subjective.
Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

DBSCAN
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

K-means

© Tan, Steinbach, Kumar  Introduction to Data Mining  4/18/2004  91
Using Similarity Matrix for Cluster Validation

DBSCAN
Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

*Algorithms for Clustering Data*, Jain and Dubes
Finding the modules

Public datasets for *H. sapiens*
- IntAct: Protein interactions (PPI), 18773 interactions
- IntAct: PPI via orthologs from IntAct, 6705 interactions
- MEM: gene expression similarity over 89 tumor datasets, 46286 interactions
- Transfac: gene regulation data, 5183 interactions
Finding the modules

Public datasets for *H.sapiens*

- **IntAct:** Protein interactions (PPI), *18773* interactions
- **IntAct:** PPI via orthologs from IntAct, *6705* interactions
- **MEM:** Gene expression similarity over 89 tumor datasets, *46286* interactions
- **Transfac:** Gene regulation data, *5183* interactions
Module evaluation

- GO: Transforming growth factor beta signaling pathway, embryonic development, gastrulation
  - KEGG: Cell cycle, cancers, WNT pathway

- GO: JAK-STAT cascade, Kinase inhibitor activity
  - Insulin receptor signaling pathway
  - KEGG: Type II diabetes mellitus

- GO: Brain development
  - Pigment granule
  - Melanin metabolic process
MCL clustering algorithm

• Markov (Chain Monte Carlo) Clustering

• Random walks according to edge weights

• Follow the different paths according to their probability

• Regions that are traversed “often” form clusters
Stijn van Dongen
http://www.micans.org/mcl/intro.html

With this, the MCL algorithm can be written as

$$G$$ is a graph
add loops to $$G$$ $\$ see below
set $$\Gamma$$ to some value $\$ affects granularity
set $$M_1$$ to be the matrix of random walks on $$G$$

while (change) {
    $$M_2 = M_1 \times M_1$$ $\$ expansion
    $$M_1 = \Gamma(M_2)$$ $\$ inflation
    change = difference($$M_1, M_2$$)
}

set CLUSTERING as the components of $$M_1$$ $\$ see below
Seriation, Clustering and Matrix Reordering: Towards the Encyclopedia of Structures

INNAR LIIV
innar.liiv@ttu.ee

10:15am, Oct 8, 2009
Tartu
Guest lecture@MTAT.03.183
Perception of/and experience

• Simple example of 5 entities (persons) and their relationships
• Who would you prefer to be?
  • Who wouldn’t you want to be?
• And what if the relationship means “company A sells to company B”?
• What if relationship means “love”?
Outline

• Micro-tutorial on Seriation:
  – What is it? Who cares?
  – Clustering versus seriation;
  – Related work and background
  – Recent advances

• Similarity (“goodness”) measures

• Where to go from here?
  – The Encyclopedia (Gallery, DB) of Structures?
Matrix representation of a graph
Definiton: Seriation

• **Seriation** is an exploratory combinatorial data analysis technique to reorder objects into a sequence along a one-dimensional continuum so that it best reveals regularity and patterning among the whole series.
Seriation?
A simple example with 11 objects

“Raw data”
Seriation vs clustering

CLUSTERING:

k=4

4 / 36%

3 / 27%

2 / 18%

SERIATION:
Seriation and Matrix Reordering

- Seriation is typically applied for matrix reordering (two-way one/two-mode seriation);
- Every matrix is two-way*, N x N matrix is one-mode and N x M matrix is two-mode.

*Using Carroll-Arabie taxonomy of scaling methods and terminology of Tucker
Seriation and Matrix Reordering

HIGH SCHOOL
AGRICULTURAL COOP.
RAILWAY STATION
ONE-ROOM SCHOOL
VETERINARY
NO DOCTOR
NO WATER SUPPLY
POLICE STATION
LAND REALLOCATION

VILLAGES  TOWNS  CITIES

URBAN
RURAL
**Figure 2** | A dataset rendered in different permutations.
Natural “ordering”

• Archaeological samples
  – Location 1: pottery, bones, spear
  – Location 2: pottery, copper, spear
  – Location 3: glass, copper
  – Location 4: steel, copper, pottery

• Each location should have a correct date
• Ordering – gives you a natural “timeline”
Visual matrix explorer for collaborative seriation

Innar Liiv, Rain Opik, Jaan Ubi and John Stasko

In this article, we present a web-based open source tool to support cross-disciplinary collaborative seriation with the following goals: to compare different matrix permutations, to discover patterns from the data, annotate it, and accumulate knowledge. Seriation is an unsupervised data mining technique that reorders objects into a sequence along a one-dimensional continuum to make sense of the whole series. Clustering assigns objects to groups, whereas seriation assigns objects to a position within a sequence. Seriation has been applied to a variety of disciplines including archaeology and anthropology; cartography, graphics, and information visualization; sociology and sociometry; psychology and psychometrics; ecology; biology and bioinformatics; cellular manufacturing; and operations research. Interestingly, across those different disciplines, there are several commonly emerging similar structural patterns. Visual Matrix Explorer allows users to explore and link those patterns, share an online workplace and instantly transmit changes in the system to other users. © 2011 John Wiley & Sons, Inc. WIREs Comp Stat 2011 DOI: 10.1002/wics.193

Keywords: seriation; matrix reordering; collaboration; information visualization
Seriation vs Clustering

- Example by Prof. Gilles Caraux (Permutmatrix software):

  - ORIGINAL DATA MATRIX
  - AFTER CLUSTER ANALYSIS OF ROWS
  - AFTER SERIATION
Gower & Digby (1981)

dendrogram for the variates

doubly reordered data matrix

shaded similarity matrix

shaded similarity matrix

dendrogram for the units
Can you see the pattern in data?
Did you see it?
Seriation: who cares (nowadays)?

- **Information visualization & HCI community;**
  - Ghoniem & Fekete & Castagliola: A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations + MatrixExplorer (Henry, Fekete)

- **Data mining and statistics community;**
- **Bioinformatics community;**
- **Social Network Analysis community;**
- **Operations research and combinatorial optimization community.**
Recent advances

• Niermann (2005) presented a GA approach for seriation in *The American Statistician*


“Optimal leaf ordering (OLO)”

• Bar-Joseph et al. 2001:

Unclassed Matrix Shading and Optimal Ordering in Hierarchical Cluster Analysis (Gale et al., Journal of Classification, 1:75-92, 1984)
No name consensus

• Czekanowski diagram, Robinson matrix, Reorderable matrix, Matrix reordering, Matrix visualization, Matrix analysis, Matrix permutation, Permutable matrix, Array-based clustering, Block clustering, Biclustering (two(n)-mode clustering), Co-clustering, Product Flow Analysis, Group Technology, Part/Machine group formation, Manufacturing cell formation, Cellular manufacturing, Seriation, cleaned up differential shading of the similarity matrix, Matrix tile analysis, Rearrangement clustering, Generalized Association Plots (GAP), non-destructive data analysis, optimal order of matrices, Optimal leaf ordering, band form, banded structure; Matrice ordonnable; Differentialdiagnose; метод групповой технологии и организации группового производства;
Seriation: a unified view

unidimensional seriation

block diagonal seriation

block checkerboard seriation

Pareto seriation
Similarity ("goodness") measures

- McCormick et al. (1969, 1972):

$$\arg\max_{\Pi, \Phi} \sum_{i=1}^{i=M} \sum_{j=1}^{j=N} a_{\pi(i), \phi(j)} \left[ a_{\pi(i), \phi(j+1)} + a_{\pi(i), \phi(j-1)} + a_{\pi(i+1), \phi(j)} + a_{\pi(i-1), \phi(j)} \right]$$

- Cumulative Hamming (Verin/Grishin, 1986):

$$\arg\min_{\Pi, \Phi} \left[ \sum_{j=1}^{L-1} d(x_{\phi(j)}, x_{\phi(j+1)}) \right]^{-1} \times \left[ \sum_{i=1}^{N-1} d(x_{\pi(i)}, x_{\pi(i+1)}) \right]^{-1}$$

- Can be generalized and written compactly:

$$\arg\min_{\Pi, \Phi} L(\Pi\Phi\Phi)$$
4 Important questions

• I’m not buying that matrix representation is better than graph layout! (actually, it’s worse!!!)
• How is it different from correlation? (why can’t I just calculate corr coef for everything and sort as a list?)
• How is it different from clustering? (there’s lots of tools for clustering – Why can’t I just pick one of those?)
• What is the added value to InfoVis community from this approach?
I’m not buying that matrix representation is better than graph layout!

• Such discussion is older than us (Forsyth-Katz vs Moreno 1940s, recent user studies by Fekete, Henry, Ghoniem)
• “Cliques”, clusters, hubs, chains harder to detect in graphs with different entity types (“bipartite” and n-partite graphs)
• Hard to read if $n$ gets bigger
• With graphs we encode only positive connections (existing relationships)
Seriation and Matrix Reordering

High School
Agricultural Coop.
Railway Station
One-Room School
Veterinary
No Doctor

Urban
Rural

Villages Towns Cities
How is it different from correlation?
(why can’t I just calculate corr coef for everything and sort as a list?)

• We don’t know the **two** attributes!
• We want to find multiple correlations (corr between >> more than two attr.) (2n list)
• We don’t always know what “level” of correlation provides the most information (highest corr != “best” corr)
• We are interested in chained corr
• Not to mention that there are some fundamental issues already with std corr:
How is it different from **clustering**?
(there’s lots of tools for clustering – Why can’t I just pick one of those?)

- Algorithmic problem: \( k \) # of clusters unknown

- Goal of clustering is to assign similar entities to groups, not to identify or describe **similarities/affinities** between entities!

- Clustering of attributes aka/\( \sim \) factor analysis

- It is not a clustering’s “fault”, because if the goal is not to find all similarities between entities and between clusters, it would be unefficient extra work for CPU/GPU
unidimensional seriation
block diagonal seriation
block checkboard seriation
Pareto seriation
What is the added value to InfoVis community from this approach?

- Important to distinguish that learning to see different structural patterns from the overview is not just moving along the learning curve to get the technique, but to accumulate knowledge from all your previous works.
- Not just learning how to read the display, but how to connect and combine with past experience, background information, memories from previous investigations, not on entity level, but @ abstract metalevel
Conformity analysis with structured query language

Innar Liiv, Rein Kuusik, Leo Võhandu
Tallinn Univ. of Technology, ESTONIA
Contact: innar.liiv@ttu.ee
Outline

- What is conformity?
- Conformity analysis, scale of conformity?
- Numerical example;
- CA and data mining paradigm?
- Implementation with SQL;
- Conclusions and future work.
What is conformity?

- Conformity can be defined as adjusting one's behavior or thinking to match those of other people or a group standard.

- CRM motivation & challenge: When people are free to do as they please, they usually imitate each other.

*Eric Hoffer*
Conformity analysis

- Aligns the objects and attributes according to nearest-neighbour similarity and therefore establishes a scale of typicality in the data.
- Describes the transformation between the classes and clusters;
- **Goal: data “morphing”**
  * *Morphing* is a special effect in motion pictures and animations that changes (or morphs) one image into another through a seamless transition.*
### Numerical example

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$O_2$</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>$O_3$</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>$O_4$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>$O_5$</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$O_6$</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

| conformity | 12 | 20 | 20 | 16 | 14 | 18 |

| $f_1$ | 2 | 4 | 2 | 4 | 4 | 4 |
| $f_0$ | 4 | 2 | 4 | 2 | 2 | 2 |
### Numerical example (cont.)

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$f_1$</th>
<th>$f_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>$O_2$</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$O_3$</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$O_4$</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$O_5$</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$O_6$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

| conformity | 12 | 20 | 16 | 20 | 18 |
### Numerical example (result)

<table>
<thead>
<tr>
<th></th>
<th>$a_2$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_3$</th>
<th>$a_1$</th>
<th>conformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_2$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>$O_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>$O_6$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>$O_4$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>$O_5$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>$O_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td><strong>conformity</strong></td>
<td><strong>20</strong></td>
<td><strong>20</strong></td>
<td><strong>18</strong></td>
<td><strong>16</strong></td>
<td><strong>12</strong></td>
<td></td>
</tr>
</tbody>
</table>
CA and data mining paradigm

- Outlier research;
- Relations and connections mostly with unsupervised approach:
  - Self-organization & Kohonen;
  - Association rules;
  - Clustering;
- Value(knowledge)-adding also to supervised learning approaches (classification).
Implementation with SQL

- **Advantages**: eliminates data movement, speeds data mining, simplifies model deployment, and delivers security and scalability.
- **Motivation**: to delegate all the enumeration and calculation steps to database system;
- Possible to exploit the underlying SQL parallelization.
- Unoptimized $\rightarrow$ **compatible** with MySQL 4.1.1, MS-SQL 2000, MS Access 2000, PostgreSQL 8.1.0, Oracle 10g.
SELECT tmp_o_sum.o, tmp_a_sum.a, DATA_TABLE.v, tmp_o_sum.tmp_o_sum as o_sum, tmp_a_sum.tmp_a_sum as a_sum
FROM ((SELECT tmp_frequency.o, Sum(tmp_frequency.ver) AS tmp_o_sum
FROM (SELECT DATA_TABLE.o, DATA_TABLE.a, tmp_freq_v.s AS ver, tmp_freq_h.s AS hor
FROM (SELECT o, v, count(*) AS s
FROM DATA_TABLE
GROUP BY o, v) tmp_freq_h INNER JOIN (DATA_TABLE INNER JOIN (SELECT a, v, count(*) AS s
FROM DATA_TABLE
GROUP BY a, v) tmp_freq_v ON (DATA_TABLE.v = tmp_freq_v.v) AND (DATA_TABLE.a = tmp_freq_v.a)) ON (tmp_freq_h.v = DATA_TABLE.v) AND (tmp_freq_h.o = DATA_TABLE.o))
tmp_frequency
GROUP BY tmp_frequency.o) tmp_o_sum INNER JOIN DATA_TABLE ON tmp_o_sum.o = DATA_TABLE.o) INNER JOIN (SELECT tmp_frequency.a, Sum(tmp_frequency.hor) AS tmp_a_sum
FROM (SELECT DATA_TABLE.o, DATA_TABLE.a, tmp_freq_v.s AS ver, tmp_freq_h.s AS hor
FROM (SELECT o, v, count(*) AS s
FROM DATA_TABLE
GROUP BY o, v) tmp_freq_h INNER JOIN (DATA_TABLE INNER JOIN (SELECT a, v, count(*) AS s
FROM DATA_TABLE
GROUP BY a, v) tmp_freq_v ON (DATA_TABLE.v = tmp_freq_v.v) AND (DATA_TABLE.a = tmp_freq_v.a)) ON (tmp_freq_h.v = DATA_TABLE.v) AND (tmp_freq_h.o = DATA_TABLE.o))
tmp_frequency
GROUP BY tmp_frequency.a) tmp_a_sum ON DATA_TABLE.a = tmp_a_sum.a
ORDER BY tmp_o_sum.tmp_o_sum DESC, tmp_a_sum.tmp_a_sum DESC;
Conclusions

- Conformity analysis: an efficient data-driven exploratory data analysis method;
- Enables to gain insight to the object (customer) behaviour real-time as the data changes;
- Instant deployment;
- Future work: optimization (length+speed).
Thank you for your attention!

Questions?

Contact: innar.liiv@ttu.ee
School violence

1) Bottleneck machine (in manufacturing)
2) Excellent position (in supply chain)
3) Miserable love (Psychology)