Advanced Algorithmics (6EAP)
Clustering and Seriation

Jaak Vilo
2011 Spring
# 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

Top speed [km/h]

Weight [kg]

Lorries

Sports cars

Medium market cars
Terminology

- Object or data point
- Feature
- Feature space
- Label
- Cluster
- Top speed [km/h]
- Weight [kg]
Machine learning

\[ \begin{array}{c|c}
 x & y \\
 0.50 & 0.21 \\
 0.49 & 0.42 \\
 0.67 & 0.34 \\
 1.02 & 0.55 \\
 1.53 & 0.71 \\
 0.31 & 0.33 \\
 0.35 & 0.70 \\
 0.21 & 1.20 \\
 \ldots & \ldots \\
\end{array} \]
Typical approaches

• Clustering ("Unsupervised learning")
Typical approaches

• Regression, classification ("Supervised learning")
Typical approaches

- Outlier detection
Unsupervised learning vs. Supervised

- **Find groups inherent to data (clustering)**
- **Find a “classifier” for known classes**
Motivation: Why Clustering?

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

**Goals:**
- Find representatives for homogeneous groups $\rightarrow$ **Data Compression**
- Find “natural” clusters and describe their properties $\rightarrow$ ”natural” **Data Types**
- Find suitable and useful grouping $\rightarrow$ ”useful” **Data Classes**
- Find unusual data object $\rightarrow$ **Outlier Detection**
Typical approaches

“Specific” pattern mining (e.g. graph clustering)
Clustering

• An old problem
• Many methods

• No single best “suites all needs” method
Clustering – it’s “easy” (for humans)
- *Edge-Detection* (advantage to smooth contours)

- *Texture clustering*
Clustering cont...
Group objects by similarity

• What means similarity?
  – Similar features

• Distance
  – Small distance ~ high similarity
  – Low similarity ~ large distance
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

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

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

\[ f = \begin{pmatrix} 3 & 17 & 12 & 12 & 8 \end{pmatrix} \]

\[ \text{rank=} \begin{pmatrix} 1 & 5 & 3.5 & 3.5 & 2 \end{pmatrix} \]
Clustering algorithms

• Similar objects should belong to same cluster
Hierarchical clustering

• Assign each object into a singleton cluster

• Calculate all pairwise distances

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

- **Merge** Ca, Cb **into** C

- **Re-calculate** all distances $D(C_i, C)$
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)
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

• \( D(C_i, C) = \text{average}\{ D(C_i, Ca) , D(C_i, Cb) \} \)
  – \( n_a/(n_a+n_b) \ast D(C_i, Ca) + n_b/(n_a+n_b) \ast D(C_i, Cb) \)
  – 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

<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>
<tr>
<td>.</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>
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>
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>
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
UPGMA

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

**WPGMA:**

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

**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>
<tr>
<td>...</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

[Graph showing points $x_1$ to $x_9$ on a 2D plane]
Euclidean distances

<table>
<thead>
<tr>
<th>$d_{L^2} \cdot \cdot \cdot$</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>5.0</td>
<td>5.1</td>
<td>0.8</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Average linkage hierarchical clustering
Running time for hierarchical clustering

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

Clustering 10,100, 1000 dim
Clustering 10 attrib.

Time in seconds

Data size

1 minute
5 minutes

0 2000 4000 6000 8000 10000 12000 14000 16000 18000 20000
0 50 100 150 200 250 300 350 400 450 500

Time

- $O(n^2)$ distances

- $n-1$ times merge
  - select smallest distance
  - update all distances to new cluster
Hierarchical clustering output

- Large tree with OTF icons
- Profile mode
- Output tree in DOT format
- Show the distribution of distances, subtitles and clusters
- Show value calculation
- Show histogram of all pairwise distances

Click close to the desired subtrees root to zoom in and study that subtree

It has been reported that clicking doesn’t work properly on Mac’s.
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
6200 genes, 80 exp.

Monitor size 1600x1200 pixels

Laptop: 800x600
6200 genes, 80 exp.

Monitor size 1600x1200

Laptop: 800x600

“COLLAPSE”

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

### 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>5.0</td>
<td>5.1</td>
<td>0.8</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Average linkage hierarchical clustering
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></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>

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. $x_2$ is compared with the points in the marked hypercubes
Major Clustering Approaches

- **Partitioning algorithms/Representative-based/Prototype-based Clustering Algorithm**: Construct various partitions and then evaluate them by some criterion or fitness function \(\rightarrow\) Kmeans
- **Hierarchical algorithms**: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- **Density-based**: based on connectivity and density functions \(\rightarrow\) DBSCAN, DENCLUE,…
- **Grid-based**: based on a multiple-level granularity structure
- **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 \(\rightarrow\) EM
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 its mean or medoid
• Re-calculate the “center” and repeat

• Mean is an “abstract point”
• Medoid: most central object
The *K-Means* Clustering Method

- Example
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.
K-means

1. Guess K centres
2. Assign obj to clusters
3. Move C to gravity centres
K-means
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, $d, k, t << n$; 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.
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.
We assume that the k-means initialization assigns the green, blue, and brown points to a single cluster; after centroids are computed and objects are reassigned, it can easily be seen that the brown cluster becomes empty.
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 be in the cluster.

• Shape of K-means/K-mediods 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.
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!
New centers - center of gravity for a cluster
Cluster - objects closest to a center

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

K-means clustering

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

Only the graph-lines:

Seeds: Final centers:
K-means clustering output

Cluster nr. 3 (size=13) contents:

Cluster nr. 4 (size=87) contents:
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”.

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.
Clustering – 2 dimensions
Bi-clustering

• Traditional clustering will be driven by “global” signals and ignore “local” signals
• Bi-clustering identifies groups of genes and conditions rather than just genes

Traditional clustering

Bi-clustering
Bi-clustering goals/issues

• Better capture biological reality
  – Genes only cooperate in certain conditions
  – Genes can have multiple functions
  – Datasets have functional biases

• Computationally difficult problem
  – Reducible to bi-clique finding
    • NP-complete

• Heuristics, simplifications, approximations
  – e.g. δ-biclusters, SAMBA, PISA
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-dimesnional 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
Example: Self-Organizing Maps

Animal names and their attributes

A grouping according to similarity has emerged

[Teuvo Kohonen 2001] Self-Organizing Maps; Springer;
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

2d array of neurons

Set of input signals (connected to all neurons in lattice)

Weighted synapses

\[ w_{j1}, w_{j2}, w_{j3}, \ldots, w_{jn} \]
Self-Organizing Maps

SOM – Result Example
Classifying World Poverty

Helsinki University of Technology

‘Poverty map’ based on 39 indicators from World Bank statistics (1992)
Motivation: The Idea

- Text objects must be automatically produced with semantics relationships
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

May 4, 2011
SOM - algorithm

• Iterate through data points (examples)
  – Find closest point in SOM grid
  – Adjust the representative vector and local neighbourhood
(iii) Find the best-matching neuron $\omega(x)$, usually the neuron whose weight vector has **smallest Euclidean distance** from the input vector $x$

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

‘Euclidean distance’ is the straight line distance between the data points, if they were plotted on a (multi-dimensional) graph

Euclidean distance between two vectors $a$ and $b$, $a = (a_1,a_2,…,a_n)$, $b = (b_1,b_2,…,b_n)$, is calculated as:

$$d_{a,b} = \sqrt{\sum_{i}(a_i - b_i)^2}$$
Weight Update

SOM Weight Update Equation

\[ w_j(t+1) = w_j(t) + \mu(t) \lambda_{\omega(x)}(j,t) [x - w_j(t)] \]

“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

\[ \begin{array}{c}
\text{L. rate} \\
\text{No. of cycles}
\end{array} \]

\( x \)-axis shows distance from winning node
\( y \)-axis shows ‘degree of neighbourhood’ (max. 1)
Clustering 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
Similarity searches
Similarity searches

Query: "cyc1" (cyc1, activator for cyc1, repressor for cyc1)
=> 3 genes + 10 most similar ones for each
= 3 “clusters”
Similarity searches

Expand a tight cluster by other most similar genes:
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
    \[
    m_k = \frac{1}{N} \sum_{i=1}^{N} \frac{X_i P(X_i \in C_k)}{\sum_j P(X_i \in C_j)}.
    \]
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
Genes in the dataset: 9211
Genes clustered on this page (correlation with patterns more than 0.8): 6813
Genes not clustered on this page (correlation with patterns less than 0.8): 1390

Global Clusters
- Time Series
- Global Clusters
- Day of rise

Specific Gene Classes
- Transcription regulators (CGR8, E14TG2a)
- Unknown ESTs (CGR8, E14TG2a)

Expression Waves
- Similarity cutoff 0.8
- Similarity cutoff 0.82

Pathway Animations
- Select pathway

Study your genelist
- Visualize expression
- gFinder

FunGenes Internal Data
- FunGenes data analysis (private)

Additional Links
- Official FunGenES page
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
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

(a) Dimension a
(b) Dimension b
(c) Dimension c

(a) Dims a & b
(b) Dims b & c
(c) Dims a & c
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.
Density – no particular “shape”

Original Points

Clusters
DBSCAN: Core, Border, and Noise Points
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

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

\textbf{for} all core points \textbf{do}

\hspace{1em} \textbf{if} the core point has no cluster label \textbf{then}

\hspace{2em} \text{current\_cluster\_label} \leftarrow \text{current\_cluster\_label} + 1

\hspace{2em} Label the current core point with cluster label \text{current\_cluster\_label}

\hspace{1em} \textbf{end if}

\hspace{1em} \textbf{for} all points in the \textit{Eps}-neighborhood, except \textit{i}th the point itself \textbf{do}

\hspace{2em} \textbf{if} the point does not have a cluster label \textbf{then}

\hspace{3em} Label the point with cluster label \text{current\_cluster\_label}

\hspace{2em} \textbf{end if}

\hspace{1em} \textbf{end for}

\hspace{1em} \textbf{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

Original Points

- Varying densities
- High-dimensional data

(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.

![Graph showing the sorted distance of points to their $k^{th}$ nearest neighbor. The x-axis represents points sorted according to the distance, and the y-axis represents the distance.](image)
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
- ...

Jaak Vilo and other authors  UT: Data Mining 2009
Problems and Challenges

• Considerable progress has been made in scalable clustering methods
  – Partitioning: k-means, k-medoids, CLARANS
  – Hierarchical: BIRCH, ROCK, CHAMELEON
  – Density-based: DBSCAN, OPTICS, DenClue
  – Grid-based: STING, WaveCluster, CLIQUE
  – Model-based: EM, Cobweb, SOM
  – Frequent pattern-based: pCluster
  – Constraint-based: COD, constrained-clustering

• Current clustering techniques do not address all the requirements adequately, still an active area of research
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
<table>
<thead>
<tr>
<th>Numéro des Centres</th>
<th>Nom du Arrondissement</th>
<th>Population</th>
<th>Nettoyage</th>
<th>Eau potable</th>
<th>Gaz</th>
<th>Électricité</th>
<th>École</th>
<th>Hospitalisation</th>
<th>Réformés</th>
<th>Catholiques</th>
<th>Musulmans</th>
<th>Non-conformistes</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Le Marais</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>Le Bas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>Temple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>1er arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>2nd arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VI</td>
<td>3rd arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td>4th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIII</td>
<td>5th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IX</td>
<td>6th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>7th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XI</td>
<td>8th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XII</td>
<td>9th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XIII</td>
<td>10th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XIV</td>
<td>11th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XV</td>
<td>12th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XVI</td>
<td>13th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XVII</td>
<td>14th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XVIII</td>
<td>15th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XIX</td>
<td>16th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>17th arrondissement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• **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.
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

Seriation
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

shaded similarity matrix

doubly-reordered data matrix

shaded similarity matrix

dendrogram for the units
Can you see the pattern in data?
Did you see it?
Related work

- Robinson (Archaeology, 1951)
- Sokal (Biology, 1963)
- Burbidge (Manufacturing, 1969, 1972)
- Hartigan (Statistics, 1971)

Petrie (Anthropology, 1909)
Czekanowski (Anthropology, 1946)
Forsyth & Katz (Sociology, 1967)
Bertin (Cartography, 1967, 1979)
McCormick (Op. research, 1969, 1972)
Mullat (Survey DA, 1979)
Marcotorchino (Unified approach, 1991)

Biclustering (2004)
Liiv (Unified view, Algorithms Applications, 1999)
Chen et al. (GAP, 1992)
Siirtola (Visualization, 1991)
Seriation: who cares (nowadays)?

- **Information visualization & HCI community;**

- **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 checkboard seriation
- Pareto seriation
Similarity ("goodness") measures

- McCormick et al. (1969, 1972):

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

- 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:

$$\arg\min_{\Pi, \Phi} L(\Pi \Delta \Phi)$$
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)
- “Clique”, 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

VILLAGES TOWNS CITIES

URBAN RURAL
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/~ 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 inefficient 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
School violence

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