Algorithmics
Clustering and Seriation

Jaak Vilo
2013 Spring

Jaak Vilo and other authors
UT: Data Mining 2009

Jaak Vilo and other authors
UT: Data Mining 2009
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) vs. 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

Terminology

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

### Types
- Partitioning vs Hierarchical
- Exclusive vs Overlapping vs Fuzzy membership
- Complete vs partial

---

Clustering cont...

Jaak Vilo and other authors
2-D data

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

5-Dimensional

<table>
<thead>
<tr>
<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>3</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>5</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>I</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

5-dimensional...

Exercise: cluster 2-D data

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

2-D data
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

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

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

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

Legendre & Legendre: Numerical Ecology 2nd ed.
Rank correlation

\[ d(f, g) = \frac{2}{c^2 - 1} \sum_{i=1}^{c} (\text{rank}_i - \text{rank}_g) \]

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

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

Performance:
\[ O(dn^2) \]

Hierarchical clustering

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

Keep joining together two closest clusters by using the:
- Minimum distance
- Maximum distance
- Average distance

\[ \min \{ d_{ab} | a \in A, b \in B \} \]
\[ \max \{ d_{ab} | a \in A, b \in B \} \]
\[ \text{avg}(d_{ab}) | a \in A, b \in B \]
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 \( D(C_a, C) \)

Merge \( C_a, C_b \) into \( C \)

- \( D(C_i, C) = \min\{ D(C_i, C_a), D(C_i, C_b) \} \)
  - Single link; Minimal distance
- \( D(C_i, C) = \max\{ D(C_i, C_a), D(C_i, C_b) \} \)
  - Complete link; Maximum distance
- \( D(C_i, C) = \text{average}\{ D(C_i, C_a), D(C_i, C_b) \} \)
  - \( n_i/(n_a+n_b) * D(C_a, C_i) + n_i/(n_a+n_b) * D(C_b, C_i) \)
  - UPGMA – Unweighted Pair Group Method Average

How to Define Inter-Cluster Similarity

- \( \text{MIN} \)
- \( \text{MAX} \)
- 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

\[
\begin{array}{cccc}
& p_1 & p_2 & p_3 & p_4 & \ldots \\
\hline
p_1 & & & & & \\
p_2 & & & & & \\
p_3 & & & & & \\
p_4 & & & & & \\
& & & & & \\
\end{array}
\]

UPGMA

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

WPGMA:

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

Single-Linkage:

\[
D(C_k, (C_i \cup C_j)) = \min \{ D(C_k, C_i), D(C_k, C_j) \}
\]

Other methods driven by an objective function

- Ward’s Method uses squared error

Euclidean distances

Average linkage hierarchical clustering

Input data visualized
Running time for hierarchical clustering

- \( O(n^2) \) distances
- \( n-1 \) times merge
  - select smallest distance
  - update all distances to new cluster

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

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 representatives 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

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**

- 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**

**Strengths**
- Relatively efficient: $O(k*n^2*d)$, where $n$ is # objects, $k$ is # clusters, and $d$ 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

**Weaknesses**
- 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 initialisation; bad initialisation 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 ⊆ 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 a 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!
Representative-Based Clustering ... (Continued)

Attribute 1
1 2 3 4
Attribute 2

Representative-Based Supervised Clustering ... (continued)

Attribute 1
1 2 3 4
Attribute 2

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 $C_k$ with the following
  \[ P(X_i \in C_k) = \frac{p(C_k)p(X_i|C_k)}{p(X_i)} \]
  - Maximization step:
    - Estimation of model parameters
    \[ m_k = \frac{1}{N} \sum_{i=1}^{N} X_i P(X_i \in C_k) \]


- Mldemos - http://mldemos.epfl.ch/

MEM – visual output

QUERY: POU5F1

Red: similar to query
Blue: distant / not similar to query

Query of OCT4 (POU5F1) (210265_X_AT) Robust Rank Aggregation
StdDev = 0.28

MEM – visual output

STACC
E-Health

SOM

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

MoN matrix of neurons, each representing “a cluster”
Object X is put to W₁, to which it is most similar.
W₁ and its near surrounding is changed to resemble X more
Train, train, train…

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
- 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’

Motivation: The Idea

- 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

Input vector

(ii) Choose an input vector \( \mathbf{x} \) from the training set

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

A Test 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 required to help us understand this high-dimensional data.

Self-organizing maps

Label

- Self-organizing
- Input vector
- Weight
- Update Equation

Finding a Winner

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

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

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

Weight Update

- SOM Weight Update Equation
  \[
  \mathbf{w}(t+1) = \mathbf{w}(t) + \eta(t) \lambda_{w_0}(l,t) [\mathbf{x} - \mathbf{w}(t)]
  \]

- "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 \( \eta(t) \)

Example of \( \lambda_{w_0}(l,t) \)

Self-Organizing Maps

- SOM – Result Example

Classifying World Poverty

Helsinki University of Technology

‘Poverty map’ based on 39 indicators from World Bank statistics (1992)

Example: Self-Organizing Maps

Animal names and their attributes

A grouping according to similarity has emerged

- peaceful
- Data
- hunters
- Fox
- Wolf
- Lion
- Eagle
- Owl
- Hawk
- Zebra
- Horse
- Cat
- Tiger
- Dog
- Wolf
- Tiger
- Zebra
- Fox
- Dog
- Cat
- Lion
- Eagle
- Owl
- Hawk
- Zebra
- Horse
- Peaceful
- Hunters

Example of \( \lambda_{\omega_0}(l,t) \)
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

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: $X \rightarrow Y \text{s.t. } d(X,Y) < 0.3$
- 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

Persistent Systems Pvt. Ltd.
http://www.persistent.co.in

Jaak Vilo and other authors
UT: Data Mining 2009

http://biit.cs.ut.ee/fungenes/
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

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 Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

```
current_cluster_label = 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label = current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
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

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

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
- ...
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

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
Module evaluation

GO: Transforming growth factor beta signaling via endoplasmic reticulum, gene regulator
Kegg: Cell cycle, cancer, Wnt
GO: JAK-STAT cascade, kinase inhibitor activity, insulin receptor signaling path.
Kegg: Type II diabetes mellitus
GO: Transforming growth factor beta signaling path.
Embryonic development, gastrulation.
Kegg: Cell cycle, cancers, Wnt pathway.

Jüri Reimand: GraphWeb.
Genome Informatics, CSHL. Nov 1 2007

MCL clustering algorithm
Stijn van Dongen

- Markov (Chain Monte Carlo) Clustering
  - http://www.micans.org/mcl/
- Random walks according to edge weights
- Follow the different paths according to their probability
- Regions that are traversed “often” form clusters

http://www.micans.org/mcl/intro.html

Visualization and human computation “brain exercise”

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”?

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
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?

---

**Definition: 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.

---

A simple example with 11 objects

“Raw data”

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
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”

Seriation vs Clustering

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

Can you see the pattern in data?
Did you see it?

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 (Galt 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; Differendaldiaignose; метод групповой технологии и организации группового производства;
Similarity (“goodness”) measures

- McCormick et al. (1969, 1972):
  \[
  \arg\max_{\phi,\Theta} \sum_{i<j} a_{\phi(i),\phi(j)} + a_{\phi(i),\Theta(j)} + a_{\Theta(i),\phi(j)} + a_{\Theta(i),\Theta(j)}
  \]
- Cumulative Hamming (Verin/Grishin, 1986):
  \[
  \arg\min_{\Phi,\Phi'} \left[ \sum_{i<j} d(x_{\Phi(i)}, x_{\Phi'(j)}) \right]^{1/2} \times \left[ \sum_{i<j} d(x_{\Phi(i)}, x_{\Phi(i+1)}) \right]^{1/2}
  \]
- Can be generalized and written compactly:
  \[
  \arg\min_{\Phi,\Phi'} L(\Pi,\Phi)
  \]

---

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 coeff 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

How is it different from correlation?

(why can’t I just calculate corr coeff 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 unefficient extra work for CPU/GPU

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>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
<th>a₅</th>
<th>conformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>O₂</td>
<td>0₂</td>
<td>1₂</td>
<td>0₂</td>
<td>1₃</td>
<td>1₃</td>
<td>20</td>
</tr>
<tr>
<td>O₃</td>
<td>0₂</td>
<td>1₂</td>
<td>0₂</td>
<td>1₃</td>
<td>1₃</td>
<td>20</td>
</tr>
<tr>
<td>O₄</td>
<td>1₃</td>
<td>1₃</td>
<td>0₂</td>
<td>1₃</td>
<td>1₃</td>
<td>16</td>
</tr>
<tr>
<td>O₅</td>
<td>0₂</td>
<td>0₂</td>
<td>1₂</td>
<td>0₃</td>
<td>1₂</td>
<td>14</td>
</tr>
<tr>
<td>O₆</td>
<td>0₁</td>
<td>1₄</td>
<td>1₄</td>
<td>1₄</td>
<td>1₄</td>
<td>12</td>
</tr>
</tbody>
</table>

Numerical example (cont.)

<table>
<thead>
<tr>
<th></th>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
<th>a₅</th>
<th>conformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>O₂</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>O₃</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>O₄</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>O₅</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>O₆</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

Numerical example (result)

<table>
<thead>
<tr>
<th></th>
<th>a₂</th>
<th>a₄</th>
<th>a₅</th>
<th>a₆</th>
<th>a₁</th>
<th>conformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>O₁</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>O₂</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>O₃</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>O₄</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>O₅</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>O₆</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

Conformity: 20 20 18 16 12
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_o_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 AS ver, tmp_freq.h 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 AS ver, tmp_freq.h 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))
GROUP BY tmp_frequency.a)
```

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

**EXAMPLE: QUERY2**

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