Data Mining MTAT.03.183
(4AP = 6EAP)
Clustering and Seriation

Jaak Vilo
2011 Fall

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

[Diagram showing vehicle clusters]
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 → 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

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

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>7</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>9</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>7</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...
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
\[ d(f, g) = 1 - \frac{\sum_{i=1}^{c} (f_i - \overline{f})(g_i - \overline{g})}{\sqrt{\sum_{i=1}^{c} (f_i - \overline{f})^2 \sum_{i=1}^{c} (g_i - \overline{g})^2}} \]

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

Rank correlation
\[ d(f, g) = 1 - \frac{6 \sum_{i=1}^{c} (\text{rank}_f - \text{rank}_g)}{c(c^2 - 1)} \]

Legendre & Legendre: Numerical Ecology 2nd ed.
Hierarchical clustering

- All against all distance matrix
- Linkage strategy – identify closest clusters and merge

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

Keep joining together two closest clusters by using the:
- Minimum distance
  \[ \min \{ d_{ai,bj} | ai \in A, bj \in B \} \]
- Maximum distance
  \[ \max \{ d_{ai,bj} | ai \in A, bj \in B \} \]
- Average distance
  \[ \text{avg} \{ d_{ai,bj} | ai \in A, bj \in B \} \]
  (UPGMA, WPGMA)

Update distances

- Merge \( C_a, C_b \) into \( C \)
- Re-calculate all distances \( D(C_r, C) \)

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

- \( D(C, C) = \min \{ D(C, C_a), D(C, C_b) \} \)
  - Single link; Minimal distance
- \( D(C, C) = \max \{ D(C, C_a), D(C, C_b) \} \)
  - Complete link; Maximum distance
- \( D(C, C) = \text{average} \{ D(C, C_a), D(C, C_b) \} \)
  - 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

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:

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

Single-Linkage:

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

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.5</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>1.7</td>
<td>5.6</td>
</tr>
<tr>
<td>4</td>
<td>2.1</td>
<td>6.0</td>
</tr>
<tr>
<td>5</td>
<td>2.6</td>
<td>8.4</td>
</tr>
<tr>
<td>6</td>
<td>5.1</td>
<td>5.2</td>
</tr>
<tr>
<td>7</td>
<td>8.4</td>
<td>5.5</td>
</tr>
<tr>
<td>8</td>
<td>9.3</td>
<td>2.2</td>
</tr>
<tr>
<td>9</td>
<td>8.3</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Euclidean distances

Average linkage hierarchical clustering

Jaak Vilo and other authors
UT: Data Mining 2009
6.10.2011

Persistent Systems Pvt. Ltd.

hsp://www.persistent.co.in

Running Jme for hierarchical clustering

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

Time in seconds
10\text{K}
20\text{K}

1 minute

Data size
15\text{K}

Time
\cdot O(n^2) distances
\cdot n-1 times merge
  \quad \text{select smallest distance}
  \quad \text{update all distances to new cluster}

Running time for hierarchical clustering

Hierarchical clustering output

“Cut”

“Zoom”

GENOMES: Yeast

Cut
6.10.2011

Design any heat-map coloring scheme

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

“COLLAPSE” 75 subtrees

Developed and implemented in Expression Profiler in October 2000 by Jaak Vilo and other authors

UT: Data Mining 2009

Heat map, color schema design

6200 genes, 80 exp.

VisHC; 2009
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

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

[Pubmed]

Distances from one pivot
- \( d(x_1, x_2) \) = 0.0, \( d(x_3, x_4) \) = 4.6
- Euclidean distance

Distances from two pivots
- \( d(x_1, x_2) \) = 0.0, \( d(x_3, x_4) \) = 4.6
- Euclidean distance

\( \epsilon \)-grid

1) Datapoints sorted according to EGO ord
2) Each point is compared with the later points until one hypercube away

\( \epsilon \)-grid

1) Datapoints sorted according to EGO ord
2) Each point is compared with the later points until one hypercube away

Epsilon Grid Order (EGO)

Epsilon Grid Order (EGO)

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
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
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**

**Strength**
- Relatively efficient: $O(kn^2d)$, where $n$ is # objects, $k$ is # clusters, and $d$ is the # dimensions. Usually, $k$, $d$ >> $n$; 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 be in the cluster.
- Shape of K-means/K-mediods 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 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.

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

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.
Self Organising Maps (SOM)

A 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

Motivation: The Idea

- Text objects must be automatically produced with semantics relationships

Self-Organizing Maps : Origins

- 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'

Self-Organizing Maps

- 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)

**SOM – Result Example**

Helsinki University of Technology

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

---

**Initialisation**

(i) Randomly initialise the weight vectors \( w_j \) for all nodes \( j \)

**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 \( x \)

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

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

**Weight Update**

- SOM Weight Update Equation

\[
w_j(t+1) = w_j(t) + \mu(t) \lambda_{\omega_0(j,t)} [x - w_j(t)]
\]

- “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 \( \lambda_{\omega_0(j,t)} \)
Example: Self-Organizing Maps

A grouping according to similarity has emerged.

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

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 an optimal 'hyperplane'
  - Also used for classification of samples- 'expression fingerprinting' for disease classification

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

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

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

- Core
- Border
- Noise

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 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 kth nearest neighbors are at roughly the same distance
- Noise points have the kth nearest neighbor at farther distance
- So, plot sorted distance of every point to its kth 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

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.
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

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 vs clustering**

<table>
<thead>
<tr>
<th>CLUSTERING:</th>
<th>SERIATION:</th>
</tr>
</thead>
<tbody>
<tr>
<td>k=4</td>
<td></td>
</tr>
</tbody>
</table>

| 4 / 36%     | 3 / 27%     |
| 2 / 18%     |             |

---

**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
Farmers Co-op School
Veterinary
Hospital
No Water Supply
Police Station
Land Reallocation

Villages
Towns
Cities

Seriation

Urban

Rural

Example by Prof. Gilles Caraux (Permutmatrix software):

Figure 1: A dataset ordered in different permutations.

Can you see the pattern in data?
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

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;

“Optimal leaf ordering (OLO)”

- Bar-Joseph et al. 2001:
Similarity ("goodness") measures

- McCormick et al. (1969, 1972):
  \[
  \arg\max_{\Phi} \sum_{i,j} a_{ij} \left( \sum_{k=0}^{n-1} \left( a_{ik} + a_{kj} + a_{i(k+1)} + a_{(k+1)j} \right) \right)
  \]
- Cumulative Hamming (Verin/Grishin, 1986):
  \[
  \arg\min_{\Phi} \left[ \frac{1}{2} \sum_{i,j} d(x_{\Phi(i)}, x_{\Phi(j)+1}) \right] \times \left( \sum_{i=1}^{n-1} d(x_{\Phi(i)}, x_{\Phi(i+1)}) \right)^{-1}
  \]
- Can be generalized and written compactly:
  \[
  \arg\min_{\Phi} L(J \Phi)
  \]

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)

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:

Seriation: a unified view

- unidimensional seriation
- block diagonal seriation
- block checkboard seriation
- Pareto seriation

Seriation and Matrix Reordering

- How is it different from correlation?
- 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

EXAMPLE: QUERY2

School violence

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