SMU | Bobby B. Lyle School of Engineering

Dissimilarity Plots: A Visual Exploration Tool for Partitional Clustering

CSE Colloquium

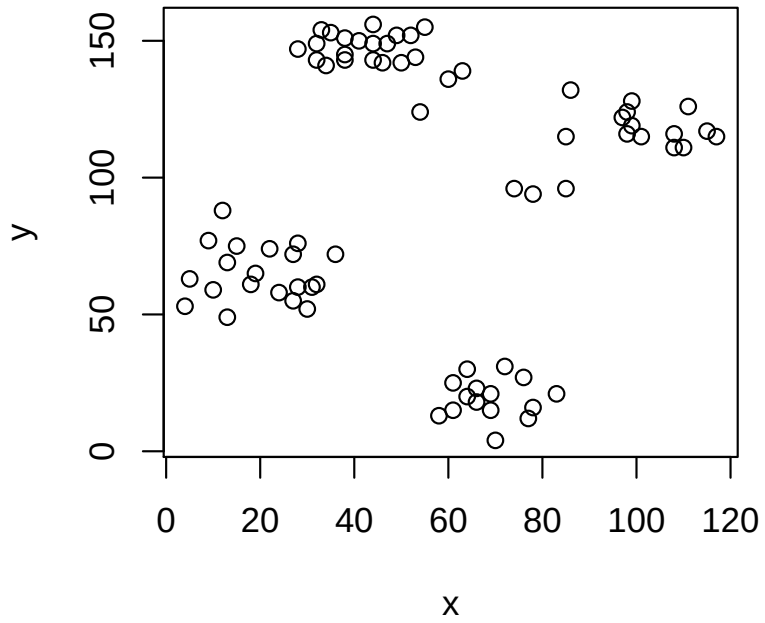
Dr. Michael Hahsler

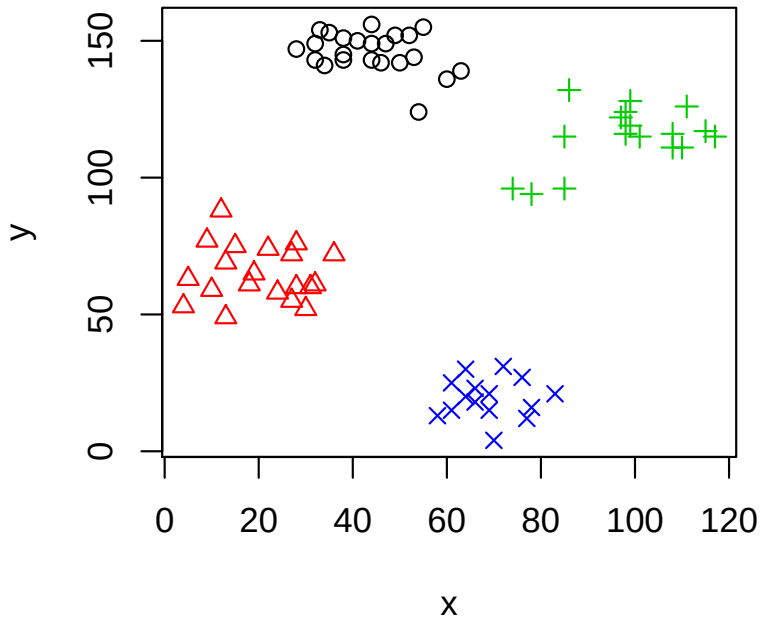
Department of Computer Science and Engineering,
Lyle School of Engineering,
Southern Methodist University.

Dallas, April 3, 2009.

Motivation

Clustering: assignment of objects to groups (clusters) so that objects from the same cluster are more similar to each other than objects from different clusters.



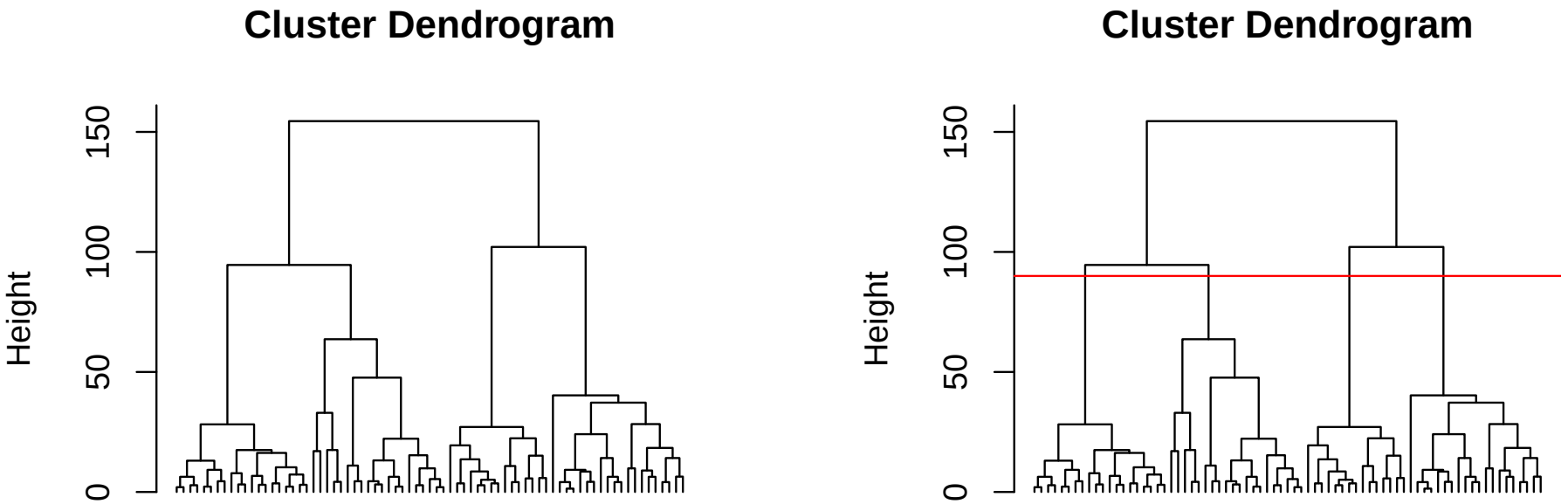


Assess the quality of a cluster solution:

- Typically judged by intra and inter-cluster similarities
- Visualization for judging the quality of a clustering and to explore the cluster structure

Motivation (cont'd)

Dendrograms (Hartigan, 1967) for hierarchical clustering:



$\rightarrow$ Unfortunately dendrograms are only possible for hierarchical/nested clusterings.

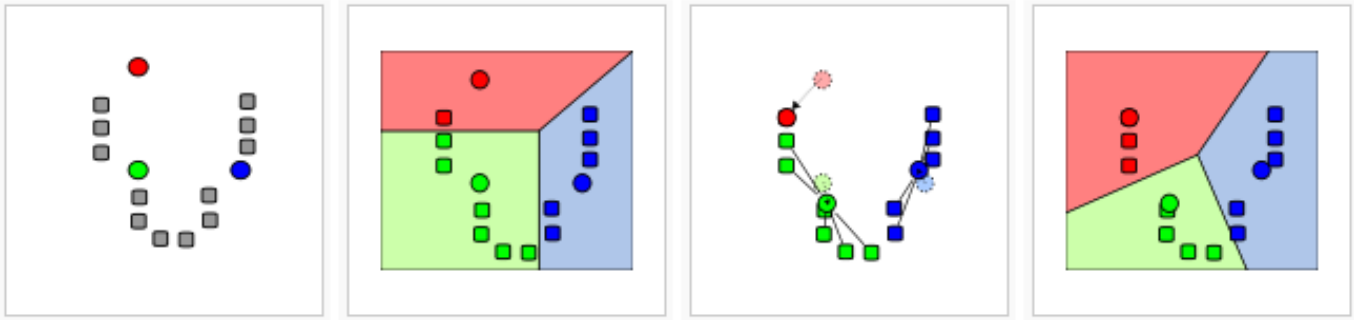
Outline

1. Clustering Basics
2. Existing Visualization Techniques
3. Matrix Shading
4. Seriation
5. Creating Dissimilarity Plots
6. Examples

Clustering Basics

- Partition: Each point is assigned to a (single) group.
  $\Gamma : \mathbb{R}^m \to \{1, 2, \ldots, k\}$

- Typical partitional clustering algorithm: k-means



Source: Wikipedia (`http://en.wikipedia.org/wiki/K-means_algorithm`)

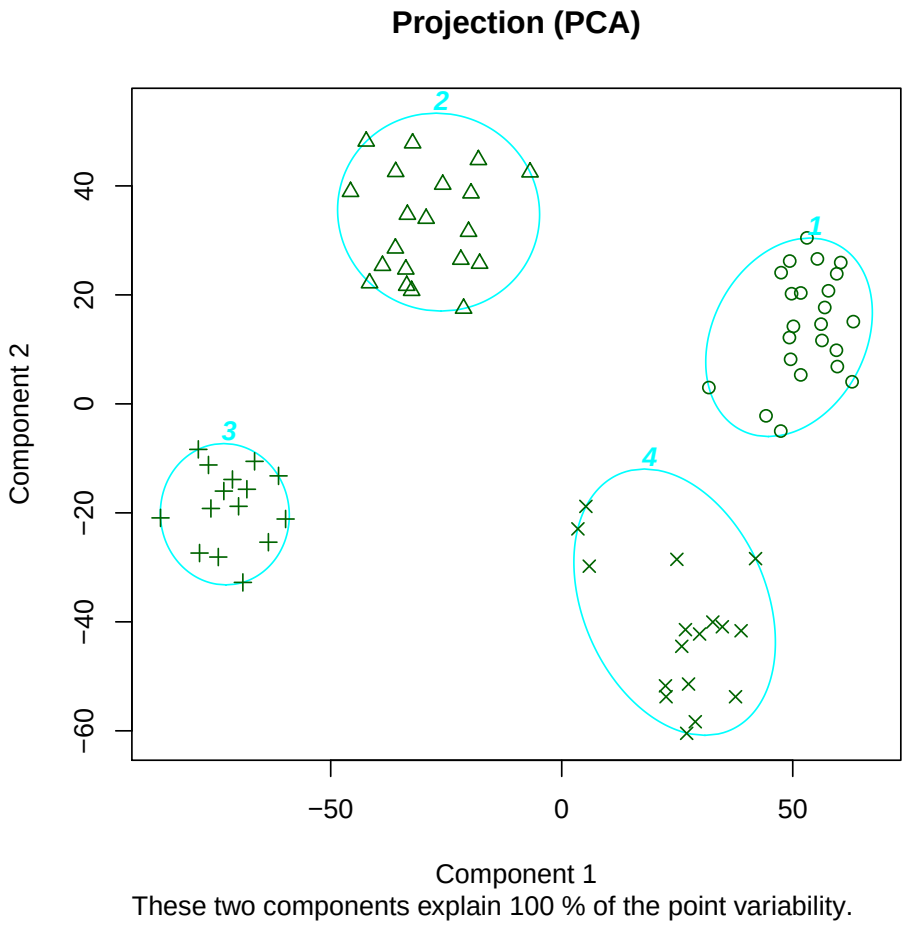
- Dissimilarity (distance) matrix: $d : \mathcal{O} \times \mathcal{O} \to \mathbb{R}$

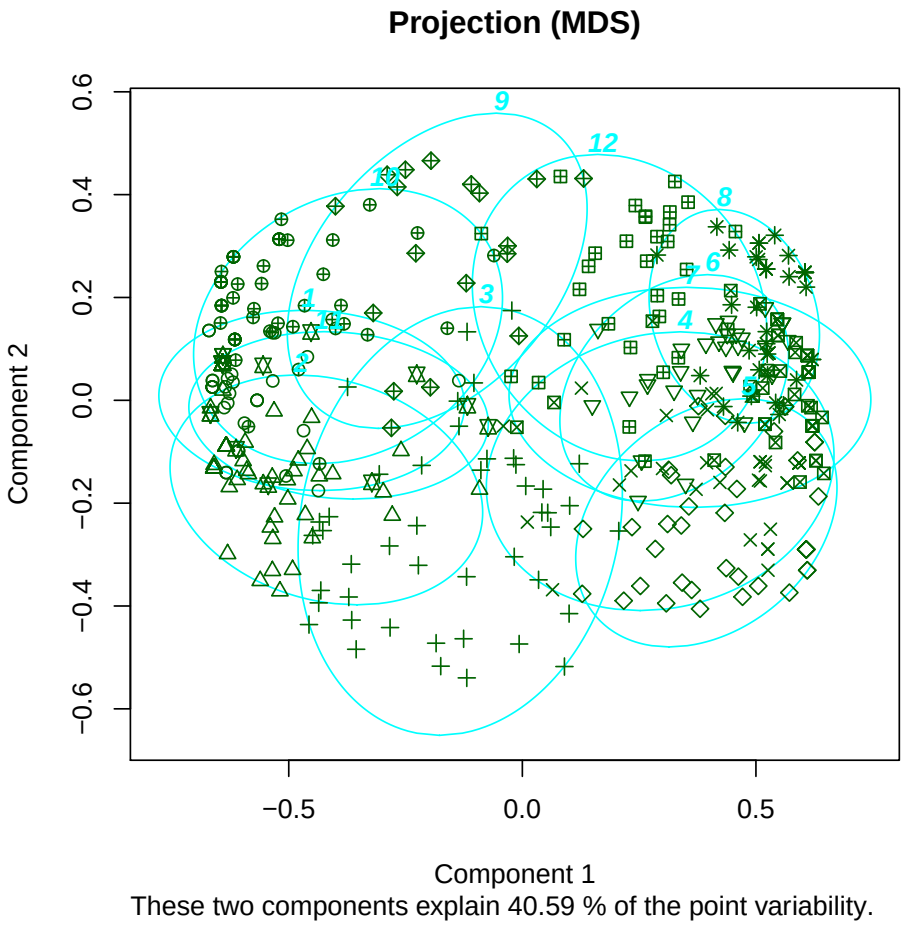
D

	O_1	O_2	O_3	O_4
O_1	0	4	1	8
O_2	4	0	2	2
O_3	1	2	0	3
O_4	8	2	3	0

Visualization Techniques for Partitions

Project objects into 2-dimensional space (dimensionality reduction techniques, e.g., PCA, MDS; Pison *et al.*, 1999).

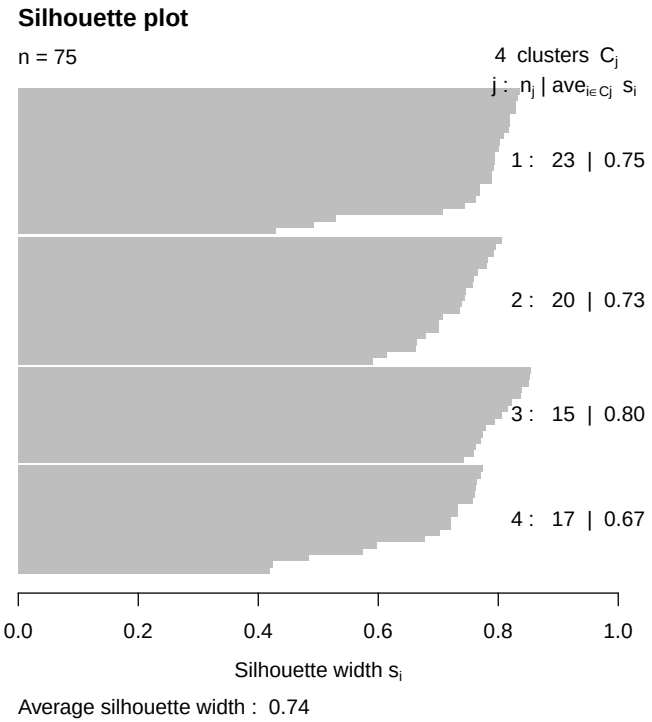




→ Problems with dimensionality (figure to the right: 16 dimensional data)

Visualization Techniques for Partitions (cont'd)

- Visualize metrics calculated from inter and intra-cluster similarities to judge cluster quality. For example, silhouette width (Rousseeuw, 1987; Kaufman and Rousseeuw, 1990).



$\rightarrow$ Only a diagnostic tool for cluster quality

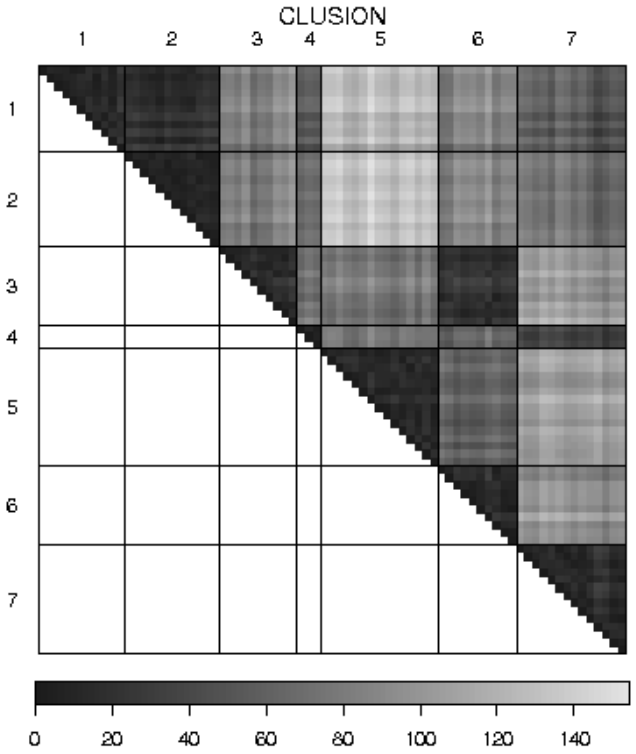
- Several other visualization methods (e.g., based on self-organizing maps and neighborhood graphs) are reviewed in Leisch (2008).
  $\rightarrow$ Typically hide structure within clusters or are limited by the number of clusters and dimensionality of data.

Matrix Shading

Each cell of the matrix (typically a dissimilarity matrix) is represented by a gray value (see, e.g., Sneath and Sokal, 1973; Ling, 1973; Gale *et al.*, 1984).

Initially matrix shading was used with hierarchical clustering $\rightarrow$ heatmaps.

For graph-based partitional clustering: ***CLUSION*** (Strehl and Ghosh, 2003). Uses ***coarse seriation*** such that "good" clusters from blocks around the main diagonal.



CLUSION allows to judge cluster quality but does not reveal the structure of the data

$\rightarrow$ ***Dissimilarity plots:*** improve matrix shading/CLUSION with (near) optimal placement of clusters and objects using ***seriation***

Seriation

Part of ***combinatorial data analysis*** (Arabie and Hubert, 1996)

- **Aim:** arrange objects in a linear order given available data and some loss function in order to reveal structural information.

- **Problem:** Requires to solve a discrete optimization problem
  $\rightarrow$ solution space grows by the order of $O(n!)$

Techniques:

1. Partial enumeration methods (currently solve problems with $n \leq 40$)
   - dynamic programming (Hubert *et al.*, 1987)
   - branch-and-bound (Brusco and Stahl, 2005)

2. Heuristics for larger problems

Seriation (cont'd)

Set of n objects

$$\mathcal{O} = \{O_1, O_2, \ldots, O_n\} \tag{1}$$

Symmetric dissimilarity matrix

$$\mathbf{D} = (d_{ij}) \tag{2}$$

where d_{ij} for $1 \leq i, j \leq n$ represents the dissimilarity between O_i and O_j, and $d_{ii} = 0$ for all i.

Permutation function Ψ reorders the objects in $\mathbf{D}$ by simultaneously permuting rows and columns

Define a loss function L to evaluate a given permutation

Seriation is the optimization problem:

$$\Psi^* = \underset{\Psi}{\operatorname{argmin}}\, L(\Psi(\mathbf{D})) \tag{3}$$

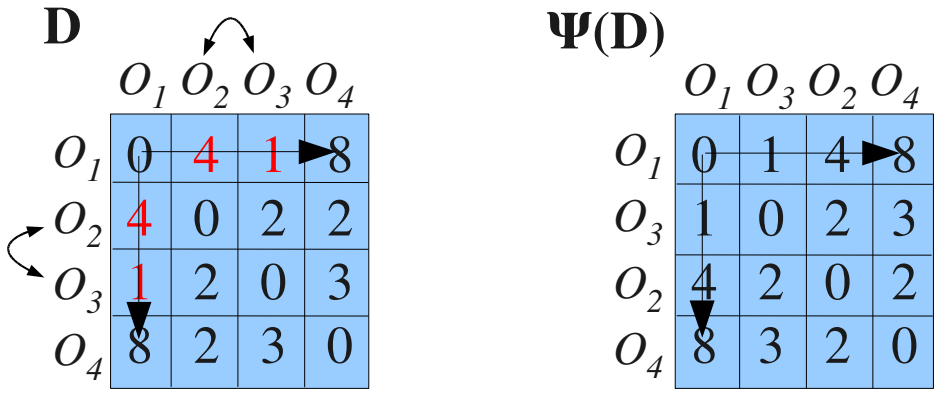
Column/row gradient measures

Perfect anti-Robinson matrix (Robinson, 1951): A symmetric matrix where the values in all rows and columns only increase when moving away from the main diagonal

Gradient conditions (Hubert *et al.*, 1987):

$$\text{within rows:} \quad d_{ik} \leq d_{ij} \quad \text{for} \quad 1 \leq i < k < j \leq n; \tag{4}$$

$$\text{within columns:} \quad d_{kj} \leq d_{ij} \quad \text{for} \quad 1 \leq i < k < j \leq n. \tag{5}$$



In an anti-Robinson matrix the smallest dissimilarity values appear close to the main diagonal, therefore, the closer objects are together in the order of the matrix, the higher their similarity.

Note: Most matrices can only be brought into a near anti-Robinson form.

Column/row gradient measures (cont'd)

Loss measure (quantifies the divergence from anti-Robinson form):

$$L(\mathbf{D}) = \sum_{i<k<j} f(d_{ik}, d_{ij}) + \sum_{i<k<j} f(d_{kj}, d_{ij}) \tag{6}$$

where $f(\cdot, \cdot)$ is a function which defines how a violation or satisfaction of a gradient condition for an object triple (O_i, O_k and O_j) is counted.

Raw number of violations minus satisfactions:

$$f(z, y) = \text{sign}(y - z) = \begin{cases} -1 & \text{if} \quad z > y; \\ 0 & \text{if} \quad z = y; \\ +1 & \text{if} \quad z < y. \end{cases} \tag{7}$$

Weight each satisfaction or violation by its magnitude (absolute difference between the values):

$$f(z, y) = |y - z|\text{sign}(y - z) = y - z \tag{8}$$

Anti-Robinson events

An even simpler loss function can be created in the same way as the gradient measures above by concentrating on violations only.

$$L(\mathbf{D}) = \sum_{i<k<j} f(d_{ik}, d_{ij}) + \sum_{i<k<j} f(d_{kj}, d_{ij}) \tag{9}$$

To only count the violations we use

$$f(z, y) = I(z, y) = \begin{cases} 1 & \text{if} \quad z < y \quad \text{and} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

$I(\cdot)$ is an indicator function returning 1 only for violations.

Chen (2002) also introduced a weighted versions of this loss function by using the absolute deviations as weights:

$$f(z, y) = |y - z| I(z, y) \tag{11}$$

Hamiltonian path length

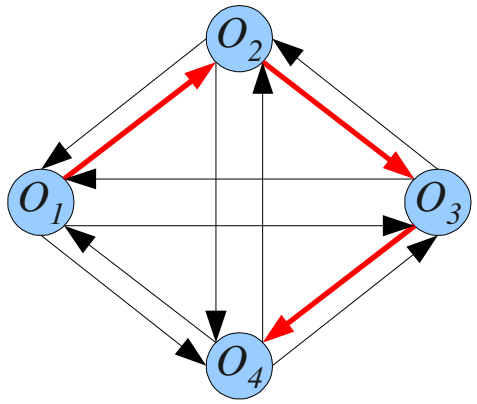
The dissimilarity matrix $\mathbf{D}$ can be represented as a finite weighted graph $G = (\Omega, E)$ where the set of objects constitute the vertices $\Omega = \{O_1, O_2, \ldots, O_n\}$ and each edge $e_{ij} \in E$ between the objects $O_i, O_j \in \Omega$ has a weight w_{ij} associated which represents the dissimilarity d_{ij}.

An order Ψ of the objects can be seen as a path through the graph where each node is visited exactly once, i.e., a Hamiltonian path. Minimizing the Hamiltonian path length results in a seriation optimal with respect to dissimilarities between neighboring objects (see, e.g., Hubert, 1974; Caraux and Pinloche, 2005).

The loss function based on the Hamiltonian path length is:

$$L(\mathbf{D}) = \sum_{i=1}^{n-1} d_{i,i+1} \qquad (12)$$

$\mathbf{D}$	O_1	O_2	O_3	O_4
O_1	0	**4**	1	8
O_2	4	0	**2**	2
O_3	1	2	0	**3**
O_4	8	2	3	0

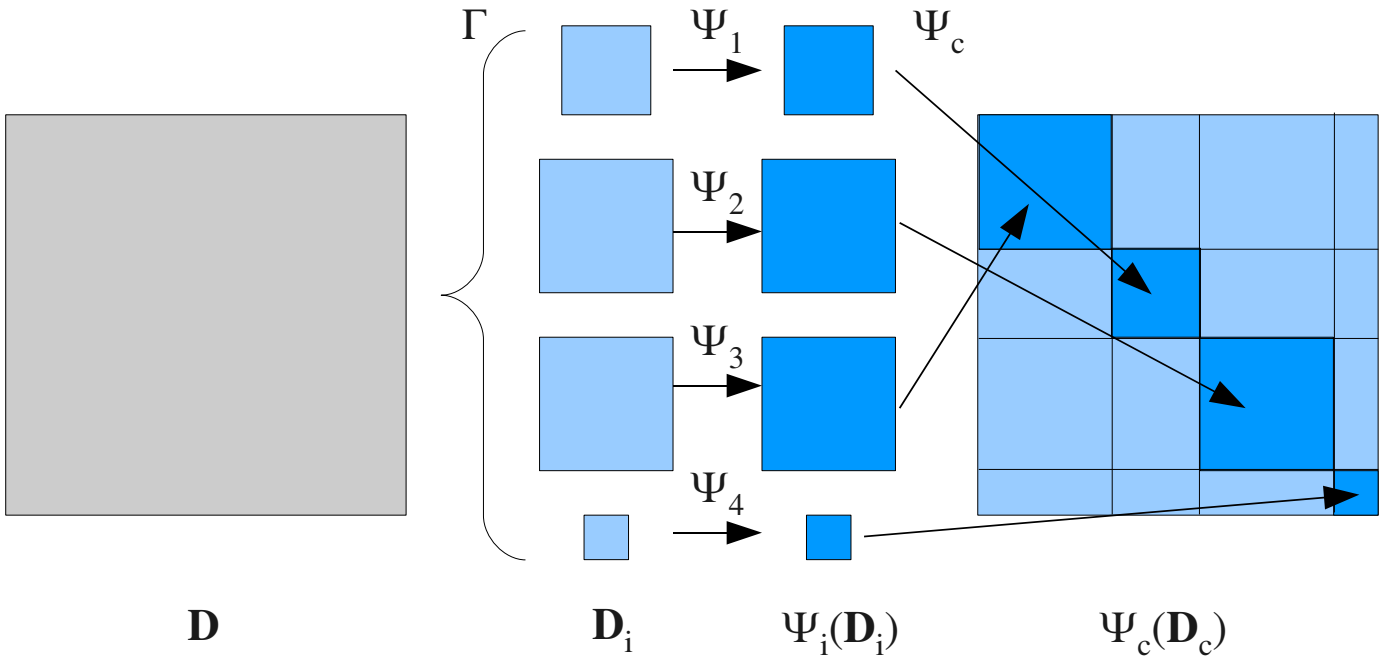


This optimization problem is related to the ***traveling salesperson problem*** (Gutin and Punnen, 2002) for which good solvers and efficient heuristics exist.

Creating dissimilarity plots

We use matrix shading with two improvements:

1. Rearrange clusters: more similar clusters are placed closer together (macro-structure).
2. Rearrange objects: show micro-structure



The assignment function Γ assigns a cluster membership to each object (provided by a partitional clustering algorithm)

Examples

We use the column/row gradient measure as the loss function for seriation.

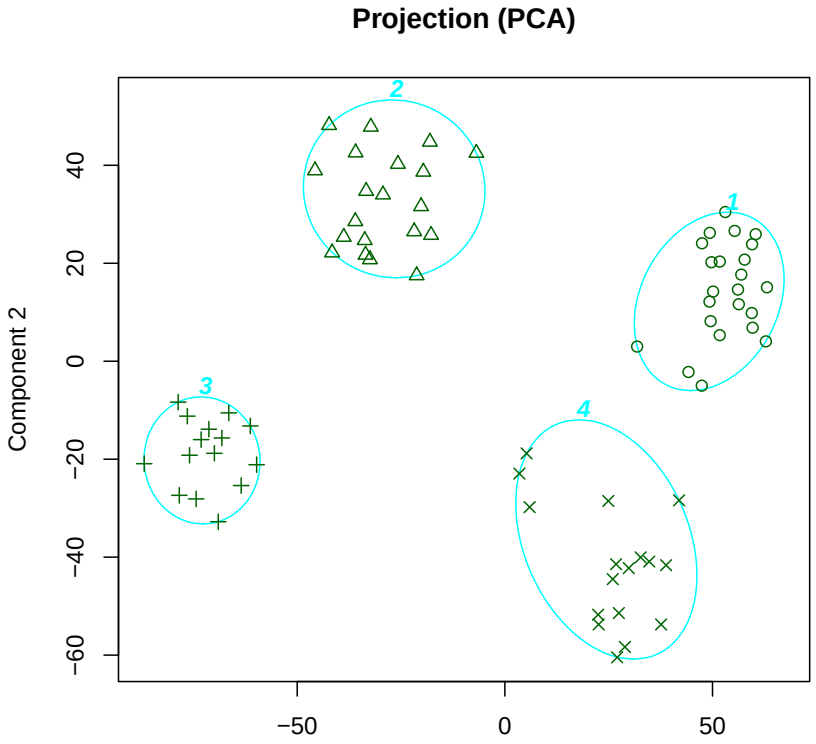
- Placement (seriation) of clusters is done using branch-and-bound to find the optimal solution
- Placement (seriation) of objects within its cluster uses a simulated annealing heuristic

Seriation algorithms are provides by Brusco and Stahl (2005) and are available in the R extension package **seriation** (Hahsler *et al.*, 2008).

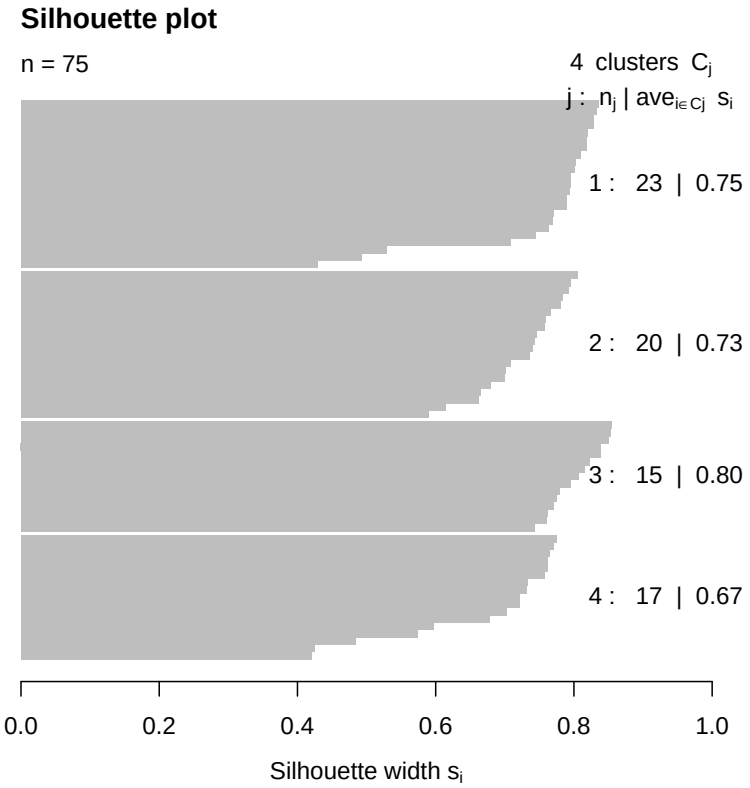
Easily distinguishable groups

Ruspini data set (Ruspini, 1970) with 75 points in two-dimensional space with four clearly distinguishable groups.

Euclidean distances and k-medoids clustering algorithm (partitioning around medoids (PAM); Kaufman and Rousseeuw, 1990) to produce a partition with $k = 4$

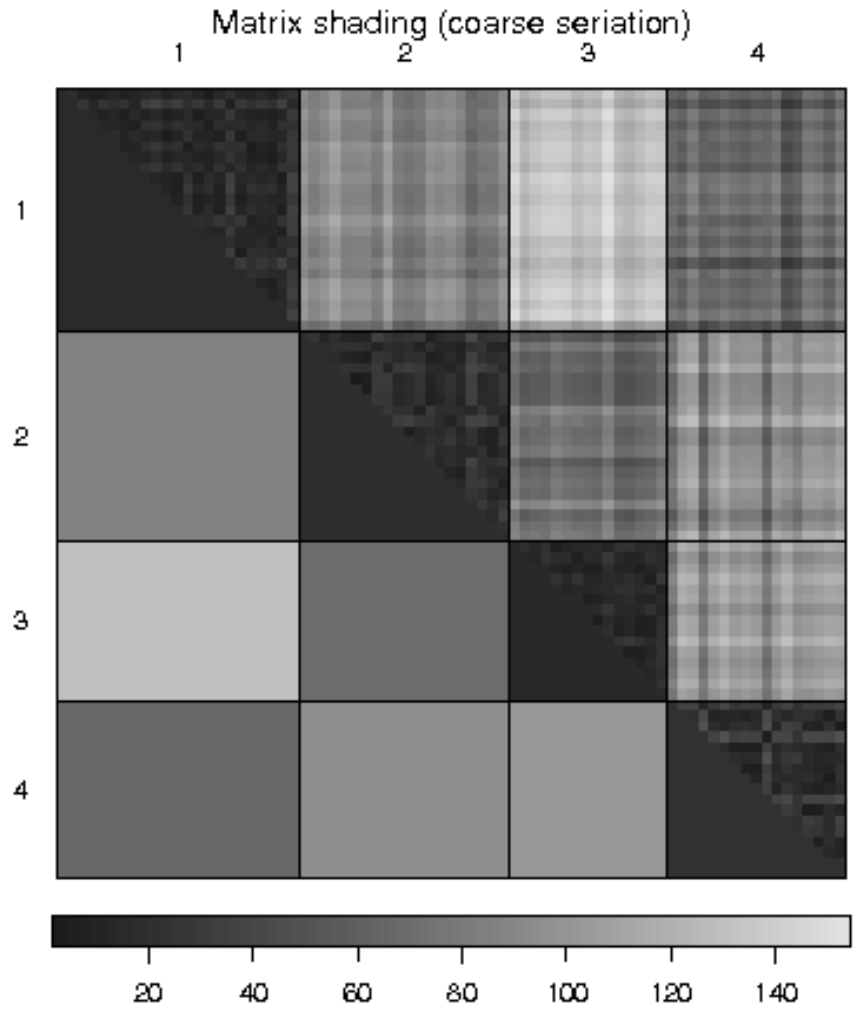


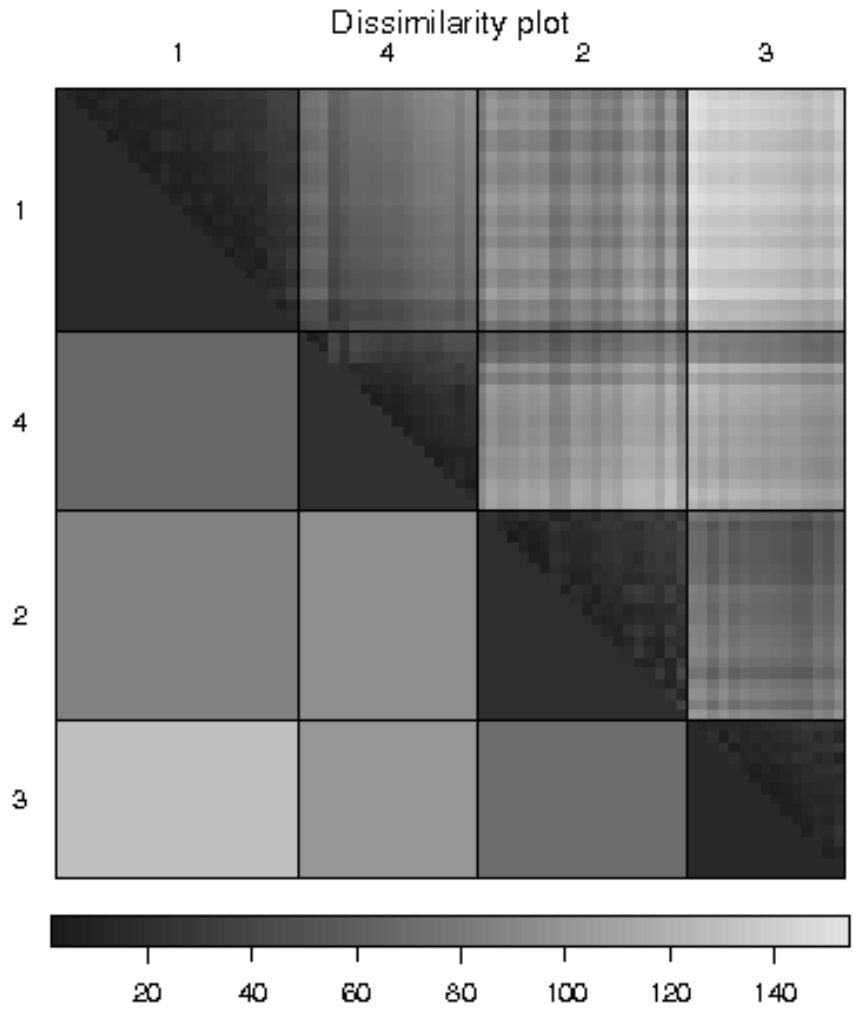
These two components explain 100 % of the point variability.



Average silhouette width : 0.74

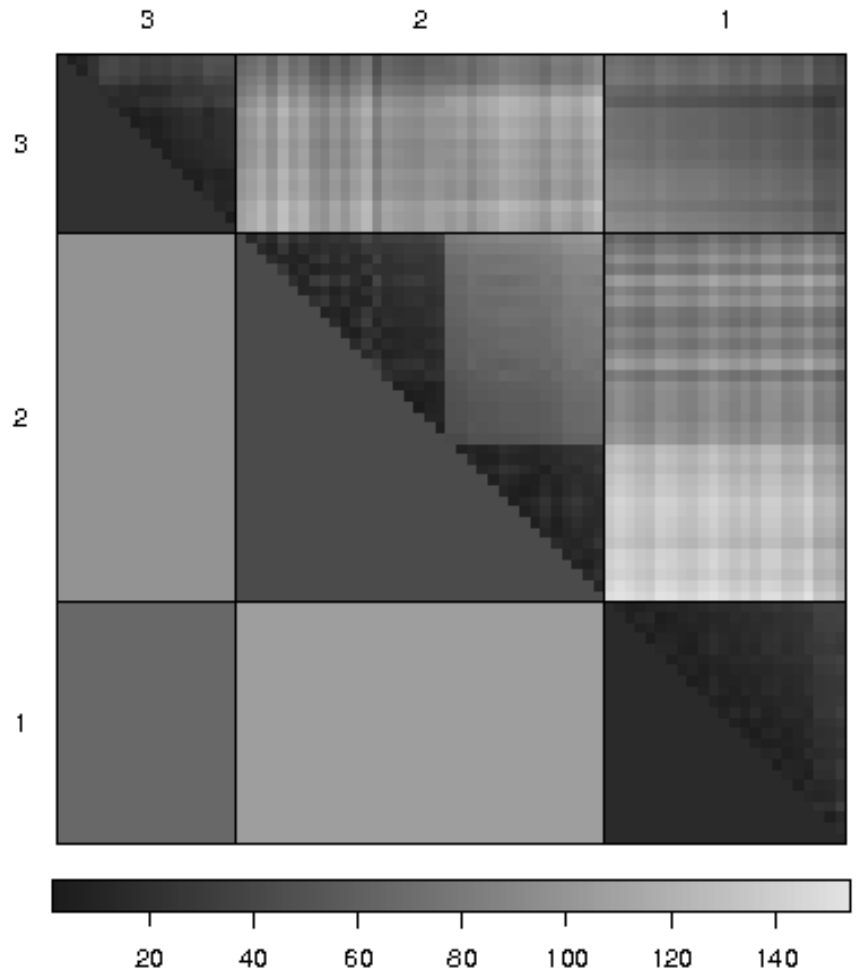
Easily distinguishable groups (cont'd)



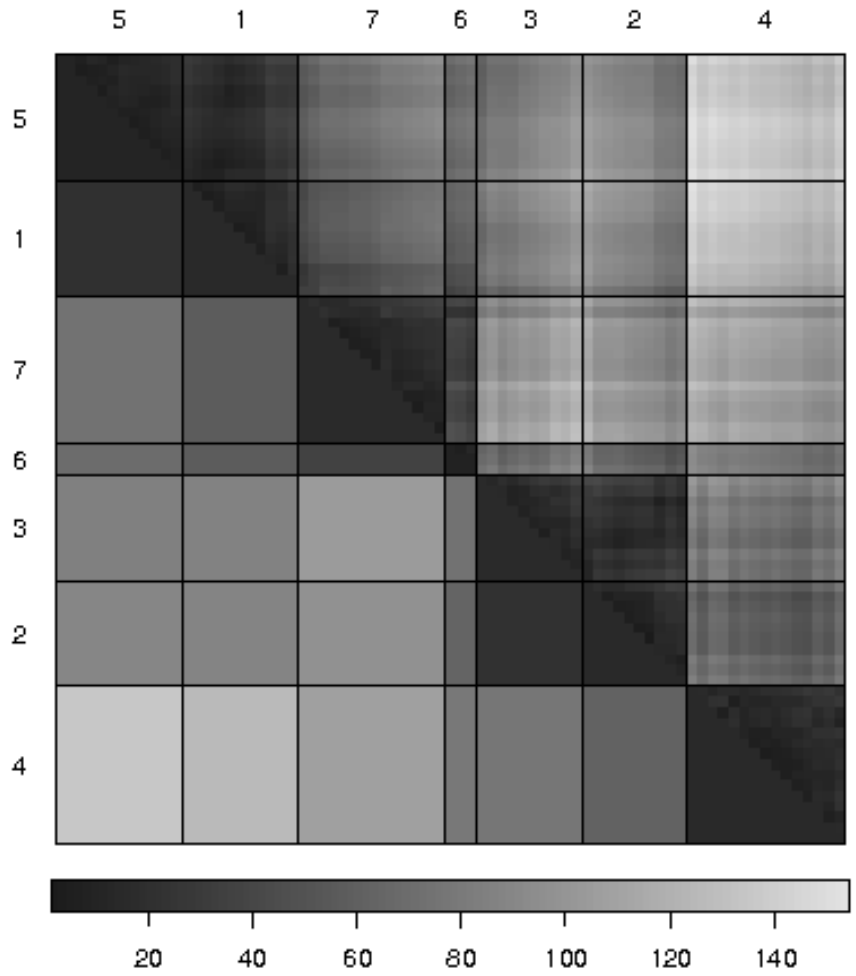


Misspecification of the number of clusters

Ruspini data set with 4 groups.



$k = 3$

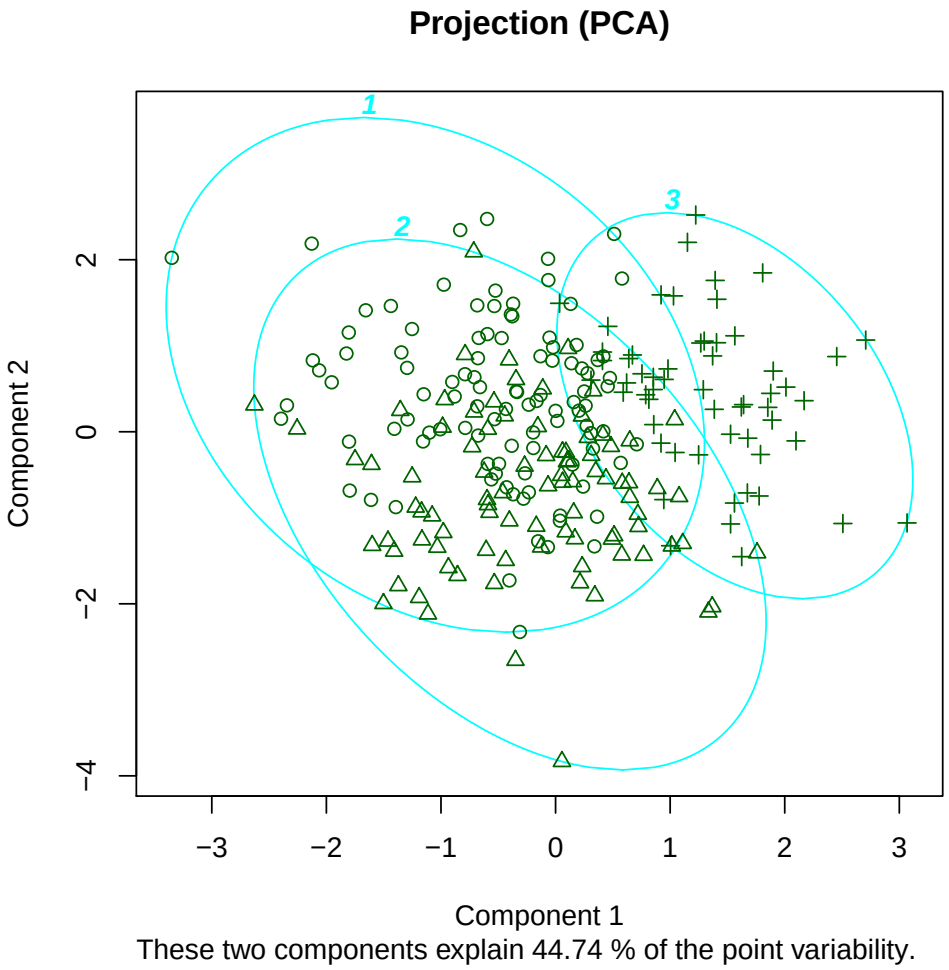


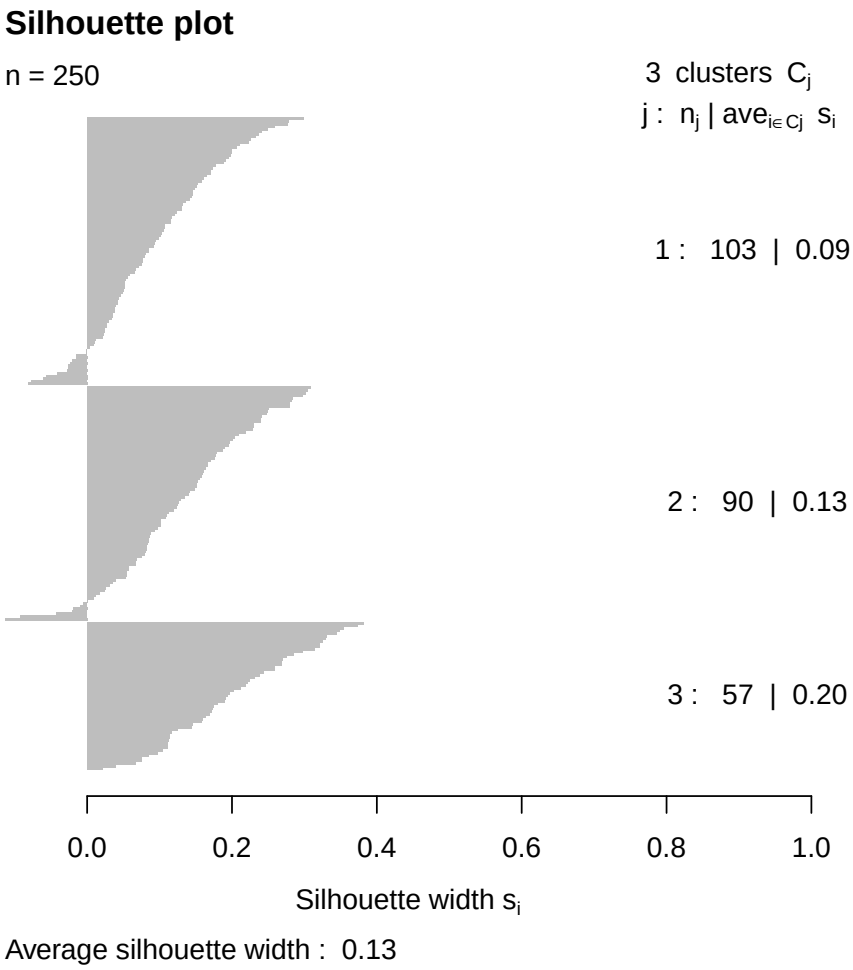
$k = 7$

No structure

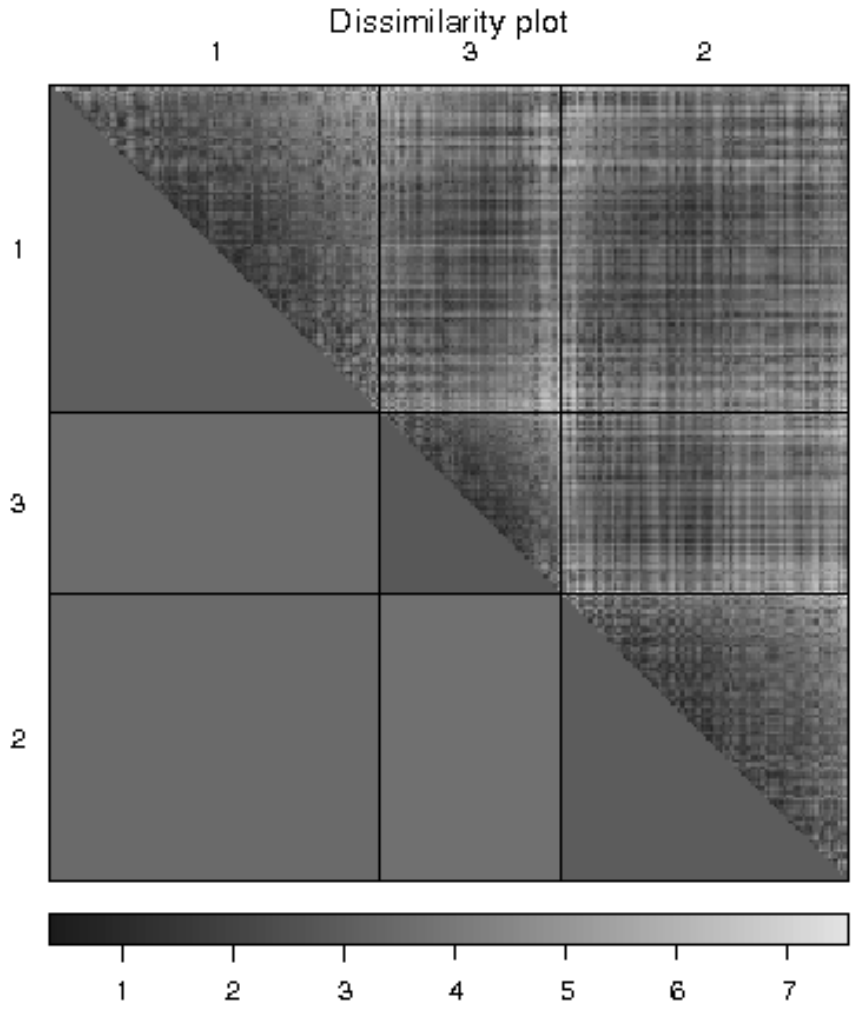
Random data for 250 objects in $\mathbb{R}^5$: $X_1, X_2, \ldots, X_5 \sim N(0, 1)$

Euclidean distance and PAM with $k = 3$





No structure (cont'd)



High-dimensional data

Votes data set (UCI Repository of Machine Learning Databases (Blake and Merz, 1998)). Votes for each of the U.S. House of Representatives congressmen on the 16 key votes during the second session of 1984.

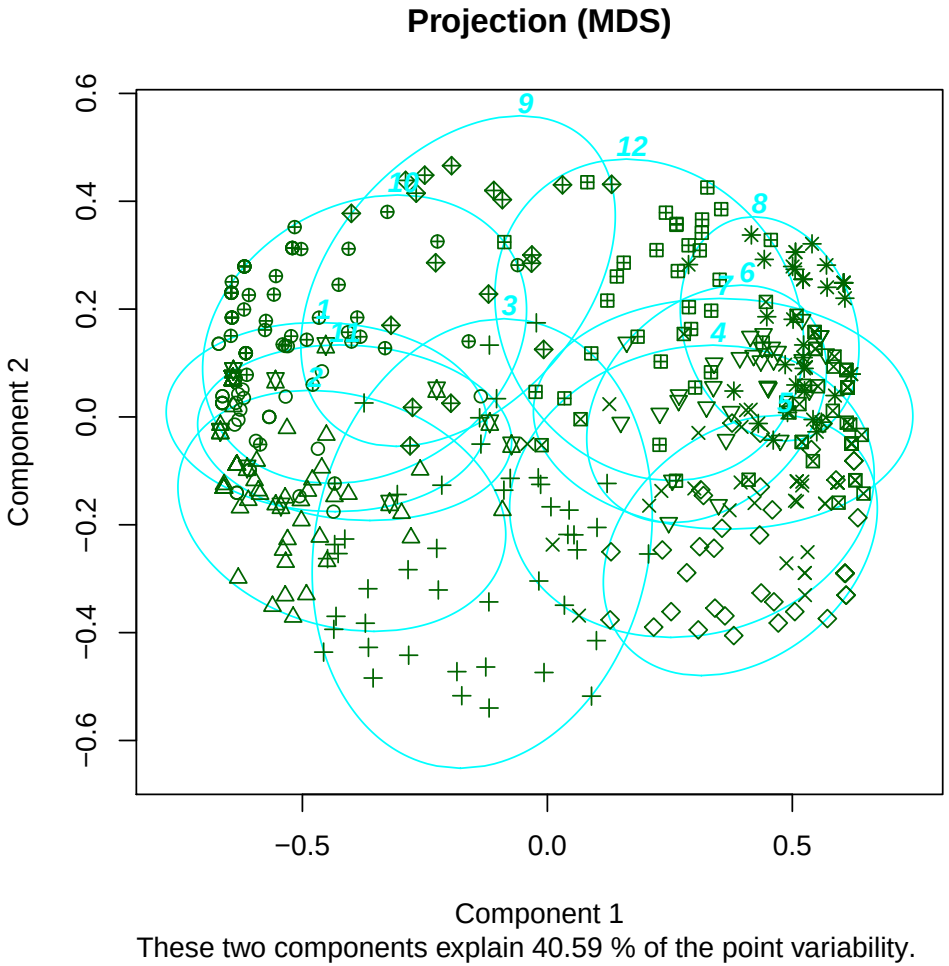
- **Coding:** 1 for in favor and 0 for vote against and unknown
  $\rightarrow$ Each congressman is represented by a vector in $\{0, 1\}^{16}$

- **Dissimilarity measure:** ***Jaccard dissimilarity*** (Sneath and Sokal, 1973) between congressmen. Let S_i and S_j be the sets of votes two congressmen voted for in favor. Then the Jaccard dissimilarity

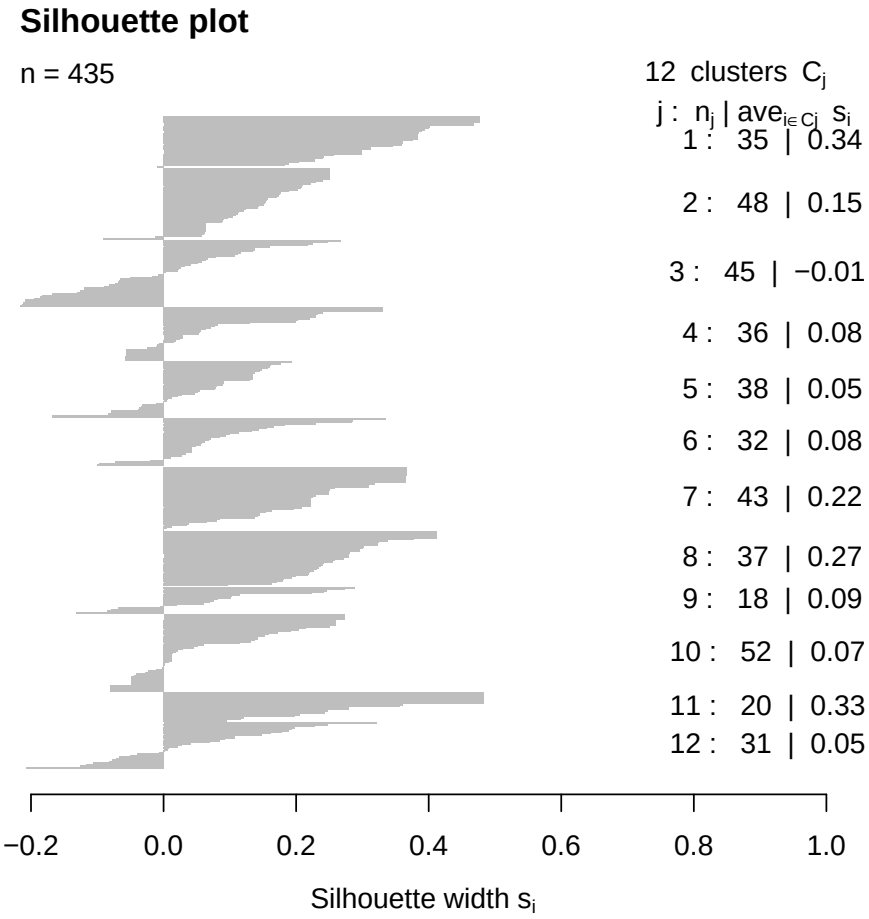
$$d_{ij} = 1 - \frac{S_i \cap S_j}{S_i \cup S_j}. \tag{13}$$

- **Cluster algorithm:** PAM with $k = 12$
  (the first bump of average silhouette for $k = 2, 3, \ldots, 30$)

High-dimensional data (cont'd)

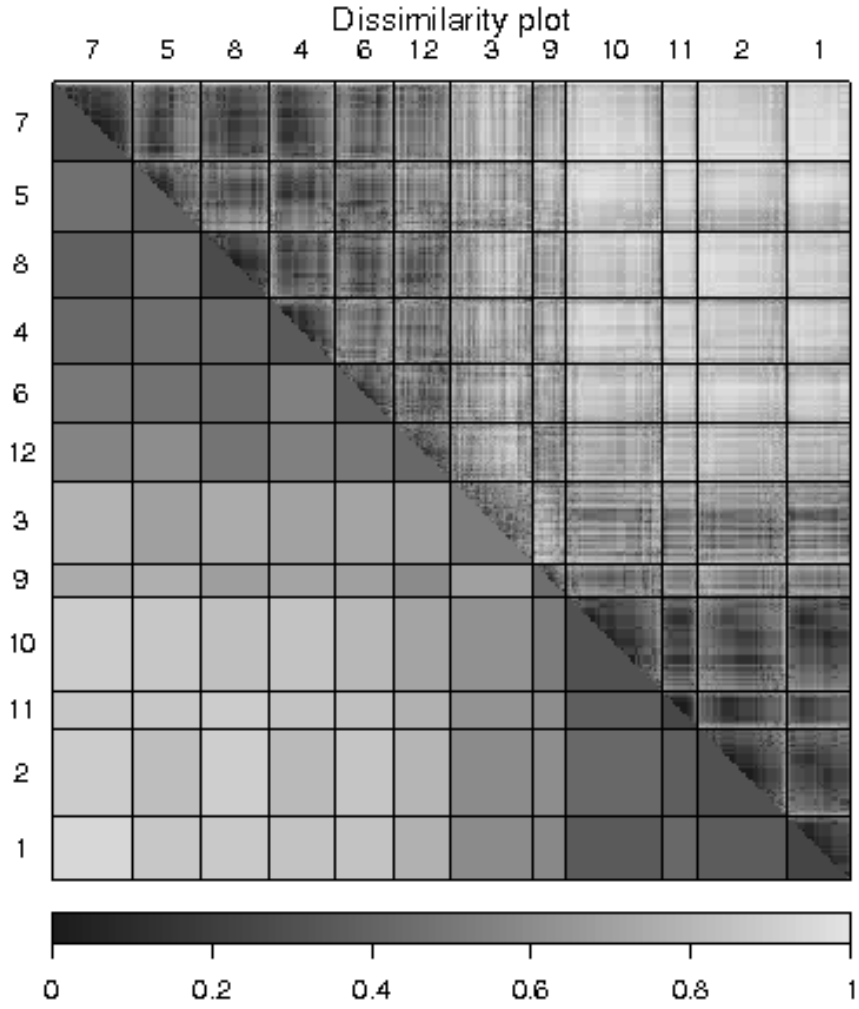


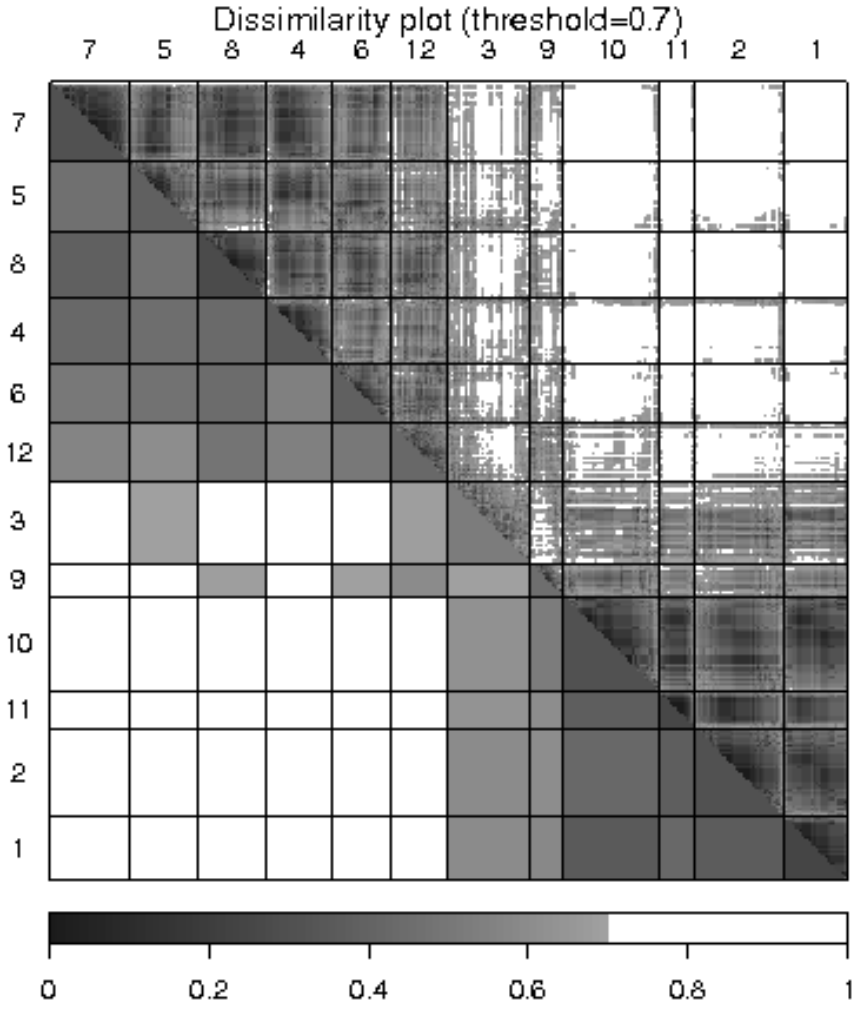
These two components explain 40.59 % of the point variability.



Average silhouette width : 0.14

High-dimensional data (cont'd)





High-dimensional data (cont'd)

	Cluster	Democrats	Republicans
1	7	42	1
2	5	38	0
3	8	36	1
4	4	34	2
5	6	32	0
6	12	26	5
7	3	41	4
8	9	2	16
9	10	3	49
10	11	5	15
11	2	7	41
12	1	1	34

Table 1: Cluster composition

Conclusion

Advantages of dissimilarity plots

- Independent of dimensionality of data (visualizes dissimilarities)
- Allows for judging cluster quality (block structure)
- Visual analysis of cluster structure (placement of clusters)
- Visual analysis of micro-structure (placement of objects)
- Makes misspecification of number of clusters apparent (placement of clusters/objects)

Planed enhancements for large number of objects/clusters:

- Image downsampling: pixel skipping, pixel averaging, 2D discrete wavelet transformation
- Separate plot for each cluster (inter-cluster structures) and a plot with only average between-cluster similarities.

Dissimilarity plot and seriation methods are implemented in the R extension package **seriation** (Hahsler *et al.*, 2008) and are freely available via the Comprehensive R Archive Network at

`http://CRAN.R-project.org`.

References

P. Arabie and L. J. Hubert. An overview of combinatorial data analysis. In P. Arabie, L. J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 5–63. World Scientific, River Edge, NJ, 1996.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

Michael Brusco and Stephanie Stahl. *Branch-and-Bound Applications in Combinatorial Data Analysis*. Springer, 2005.

G. Caraux and S. Pinloche. Permutmatrix: A graphical environment to arrange gene expression profiles in optimal linear order. *Bioinformatics*, 21(7):1280–1281, 2005.

Chun-Houh Chen. Generalized association plots: Information visualization via iteratively generated correlation matrices. *Statistica Sinica*, 12(1):7–29, 2002.

N. Gale, W. C. Halperin, and C. M. Costanzo. Unclassed matrix shading and optimal ordering in hierarchical cluster analysis. *Journal of Classification*, 1:75–92, 1984.

G. Gutin and A. P. Punnen, editors. *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*. Kluwer, Dordrecht, 2002.

Michael Hahsler, Christian Buchta, and Kurt Hornik. *seriation: Infrastructure for seriation*, 2008. R package version 0.1-6.

J. A. Hartigan. Representation of similarity matrices by trees. *Journal of the American Statistical Association*, 62(320):1140–1158, 1967.

Lawrence Hubert, Phipps Arabie, and Jacqueline Meulman. *Combinatorial Data Analysis: Optimization by Dynamic Programming*. Society for Industrial Mathematics, 1987.

L. J. Hubert. Some applications of graph theory and related nonmetric techniques to problems of approximate seriation: The case of symmetric proximity measures. *British Journal of Mathematical Statistics and Psychology*, 27:133–153, 1974.

L. Kaufman and P. J. Rousseeuw. *Finding groups in data: An introduction to cluster analysis*. John Wiley and Sons, New York, 1990.

Friedrich Leisch. Visualizing cluster analysis and finite mixture models. In Chunhouh Chen, Wolfgang Härdle, and Antony Unwin, editors, *Handbook of Data Visualization*, Springer Handbooks of Computational Statistics. Springer Verlag, 2008.

Robert L. Ling. A computer generated aid for cluster analysis. *Communications of the ACM*, 16(6):355–361, 1973.

Greet Pison, Anja Struyf, and Peter J. Rousseeuw. Displaying a clustering with clusplot. *Computational Statistics & Data Analysis*, 30(4):381–392, June 1999.

W. S. Robinson. A method for chronologically ordering archaeological deposits. *American Antiquity*, 16:293–301, 1951.

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

E. H. Ruspini. Numerical methods for fuzzy clustering. *Information Science*, 2:319–350, 1970.

Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy*. Freeman and Company, San Francisco, 1973.

A. Strehl and J. Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2):208–230, 2003.