Interactive Clustering
Overview and Tools

Wayne Oldford

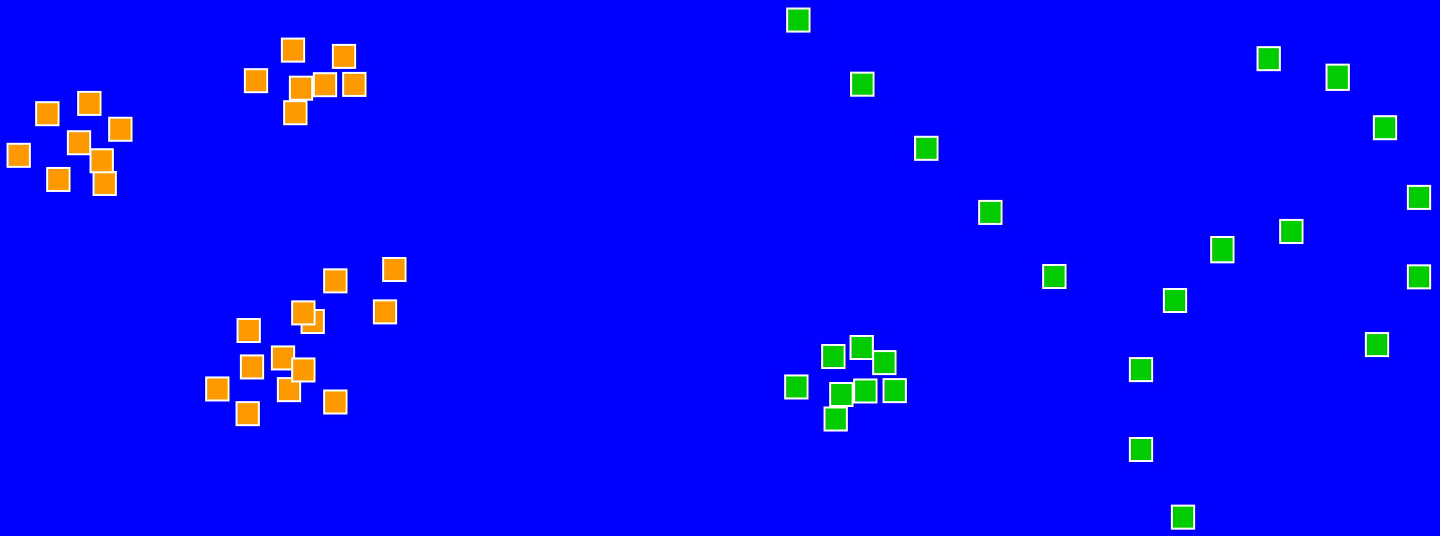
University of Waterloo

Overview

1. Finding groups in data
2. Interactive data analysis
3. Enlarging the problem
4. Putting it together
5. Software modelling (illustration)
6. Summary

1.Finding groups in data

- Objects to be grouped together
  - locations
  - pairwise (dis)similarities
- Applications:
  - Web documents as objects to be grouped
  - Building groups to use later as classification
  - Building groups to serve as templates
  - Building groups to understand/model



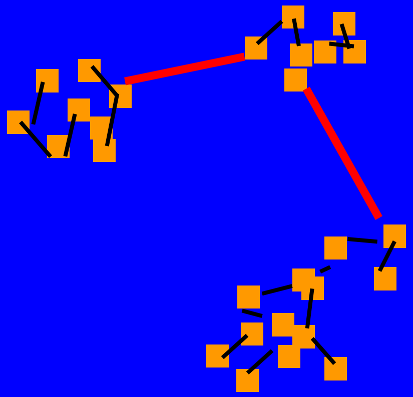
Group definition (like with like)

- homogeneous vs heterogeneous
- part of pattern

group definition is a problem

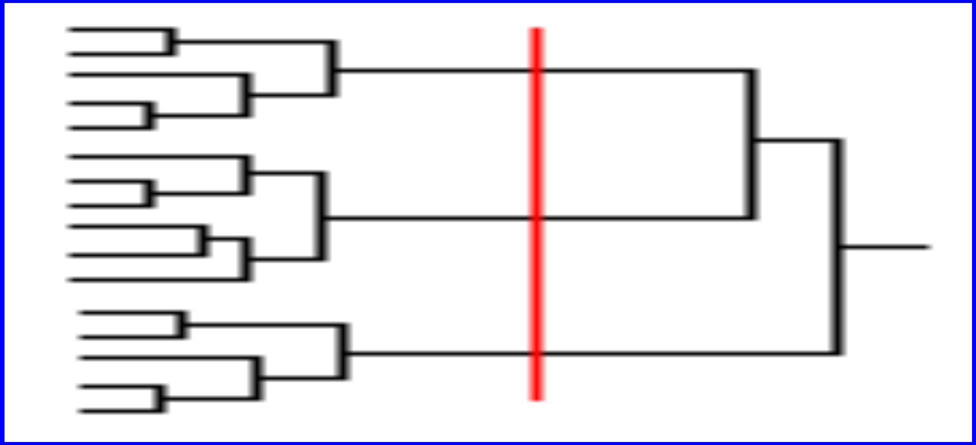
Clustering approaches

- Agglomerative (near points/clusters are joined)
  - Single linkage
  - Complete linkage
  - Average linkage



- Recursive splitting
  - e.g. minimal spanning tree

Cluster hierarchies

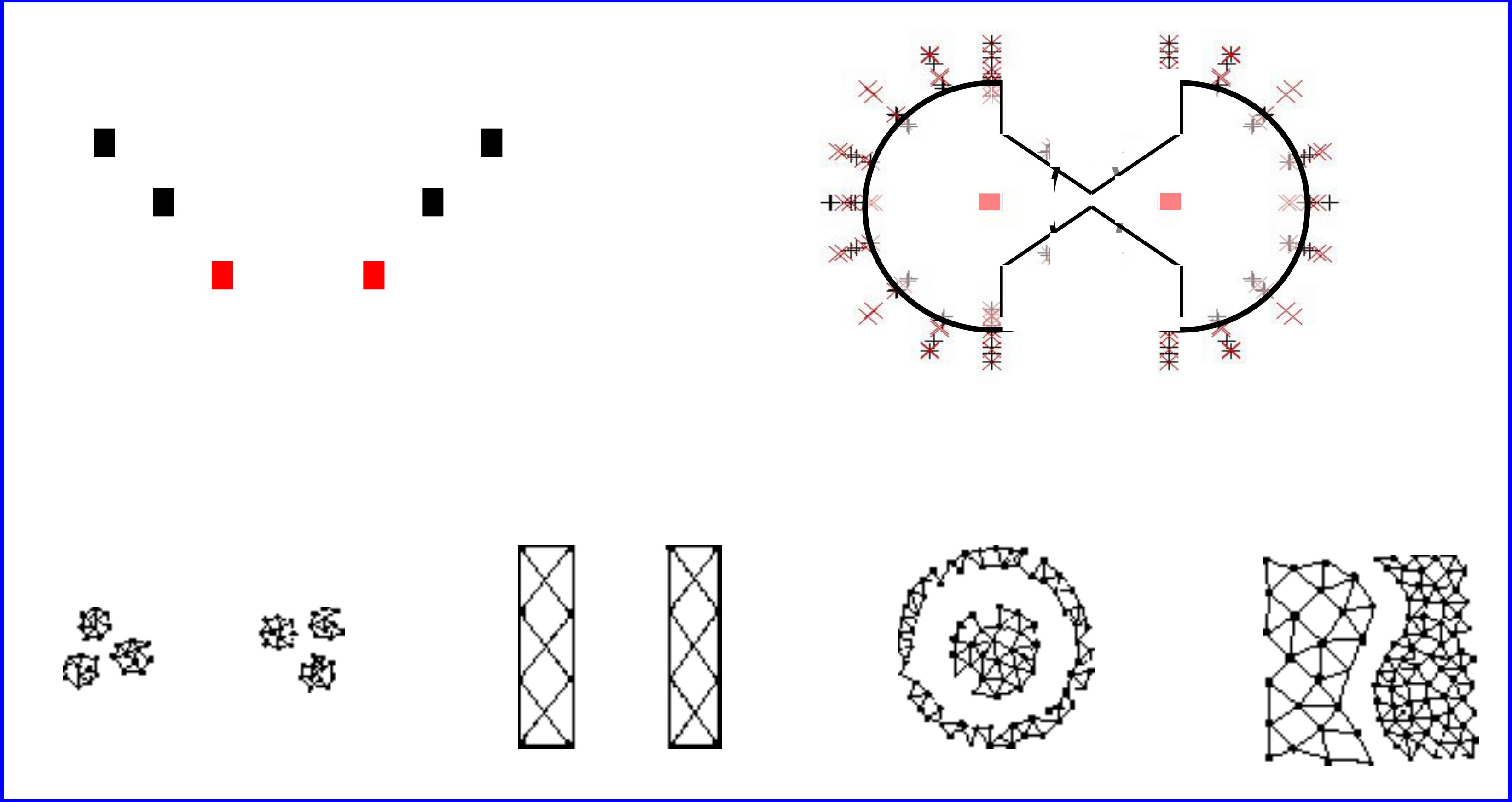


- Clusters are nested
- Often represented as a tree (dendrogram)
- Join/split history and ‘strength’ preserved

Other approaches

- k-means
  - assign points to k groups
  - re-assign to improve objective function
- model-based
  - likelihood/Bayesian; model search/averaging
- density estimation
  - groups = high-density regions
- classification to cluster
- visually motivated methods

Visual Empirical Regions of Influence (VERI)



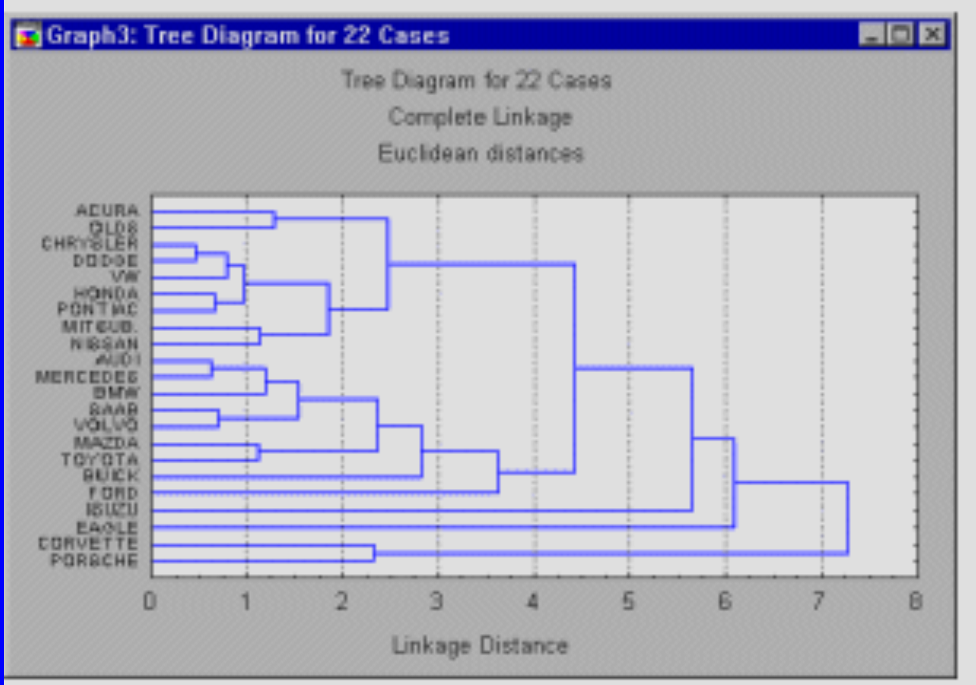
Notes

- many choices
  - between and within methods
- built-in biases for shapes
- computationally costly
  - $O(n^2)$ ...

Conceptual model: algorithmic, run to completion

typical software

- resources dedicated to numerical computation
  - teletype interaction
  - runs to completion
  - graphical “output”



Compare to interactive data analysis

Interactive data analysis

- exploratory, tentative
- graphical
- non-algorithmic
  - varied granularity
- integrated
- deep interaction

3. Enlarging the problem

Mutually exclusive and exhaustive groups

$$g_1, \quad g_2, \quad \ldots, \quad g_k$$

form a partition

$$P = \{g_1, \quad g_2, \quad \ldots, \quad g_k\}$$

of the set of data objects.

Goal: Explore the space of possible partitions.

Structuring the partition space

$P_A=\{g_1, \; g_2, \; \ldots, \; g_a\}$ and $P_B=\{h_1, \; h_2, \; \ldots, \; h_b\}$

- When $a > b$, P_A call a *finer partition* than P_B.
  - P_A is called a *refinement* of P_B (or P_B a *reduction* of P_A)
- P_A is *nested* in P_B only if $a > b$ and *every* g_i is a subset of a single h_j - write $P_A \, \} \, P_B$ or $P_B \, \{ \, P_A$
- When $a = b$, P_A is called a *reassignment* of P_B

Reduction

$P_1=\{g_1, ..., g_6\}$ -> $P_2=\{h_1, \dots, h_4\}$ -> $P_3=\{m_1, m_2, m_3\}$

- $h_i = g_i$ $i = 1, 2$; $h_3 =$ join (g_3, g_4) ; $h_4 =$ join (g_5, g_6)
  - nesting: P_1 } P_2
- disperse elements of h_4 over h_i $i = 1, 2, 3$ to give m_i for $i = 1, 2, 3$.
  - split $(h_4) = \{h_1^*, h_2^*, h_3^*\}$; $m_i =$ join (h_i^*, h_i)
  - P_2 } P_3 is false

Reduction decisions/options

- join operations: which groups?
  - e.g. inner, outer, centres, …
  - distance measures to use …
- dispersal operations:
  - selecting group(s)
    - Max volume, eigen-value, MST…
  - determining partitional method
    - random, VERI, MST, …
  - choosing join …

Refinement

$P_2=\{h_1, \ldots, h_4\}$ ---> $P_1=\{g_1, \ldots, g_6\}$

- $g_i = h_i$ $i = 1, 2$; split (h_3) -> g_3, g_4

  split (h_4) -> g_5, g_6

nesting: P_2 { P_1

Refinement decisions/options

- which groups to split?
  - e.g. inner, outer, directions, …
  - distance measures to use …
- how to split?
  - MST, outlying points, reassignment, ...

Reassignment

$$P_1=\{g_1, ..., g_k\} \text{ -> } P_2=\{h_1, \ldots, h_k\}$$

- objective function $d(P)$ to be minimized. P <- P_1
- for each object o in g_i, assign it to one of g_j (j != i) forming a new partition P_{ij} and find largest

$$\Delta_{ij}(o) = d(P) - d(P_{ij})$$

- repeat for all i, j. If $\max \Delta_{ij} > 0$ move o from g_i, to g_j
- Repeat until $\Delta_{\max} <= 0$

Reassignment decisions/options

- Objective function
  - distances, centres, …
  - within vs between/within, ...
  - variates/directions
- Iteration strategy
  - single-pass, k-means, complete looping (greedy), start, …

4. Putting it together

Series of moves in partition space:

1. Refine (P) -- > P_{new}

2. Reduce (P) -- > P_{new}

3. Reassign (P) -- > P_{new}

Additional ops on partitions

- Unary:
  - Subset (P)
  - Operate any of R (subset (P))
  - Manual (P) … change P according to manual intervention (e.g. colouring)

n-ary operators

- resolve (P_1, ..., P_m) --> P_{new}
- dissimilarity (P_i, P_j) --> $d_{i,j}$
- display (P_1, ..., P_m)
  - dendrogram if P_1 { …{ P_m
  - mds plot of all clusters in P_1, …, P_m
  - mds plot of all partitions P_1, …, P_m

5. Software modelling

- Principal control panel:
  - current partition and list of saved partitions
  - refine, reduce, re-assign, re-start buttons
  - cluster plot button (mds plot)
  - random select button
  - subset focus and join toggle
  - operation on partitions button
  - manual button (form partition from point colours)

Secondary panels

- Refine:
  - performs refine, offers access to arguments
- Reduce
  - performs reduce, offers access to arguments
- Reassign
  - performs reassign, offers access to arguments
- Each will operate on only those points highlighted or on all if none selected.

Secondary panels (continued)

- Operate on partitions
  - saved partitions list
  - resolve selected partition
  - plot selected partitions using selected dissimilarity
  - dendrogram of selected partitions (if nested)
  - cluster-plot for clusters of selected parttitions (esp. for non-nested)

Software modelling (details)

- Objects:
  - Point-symbols, case-objects (existing in Quail)
  - Cluster-points
  - Clusters
  - Partitions
- Methods
  - Reduce, refine, reassign, ...

Software illustration

- Two prototype displays (buggy)
  - Single-window
  - Separate windows
- Integration with existing Quail graphics
- Manual, dendrogram, cluster plots, …
- VERI clustering

6. Summary

- Cluster analysis is naturally exploratory and needs integration with modern interactive data analysis.
- Enlarging the problem to partitions:
  - simplifies and gives structure
  - encourages exploratory approach
  - integrates naturally
  - introduces new possibilities (analysis and research)

Acknowledgements

- Erin McLeish, several undergraduates and graduate students in statistical computing course at Waterloo
- Quail: Quantitative Analysis in Lisp

  http:/www.stats.uwaterloo.ca/Quail