Part IV

Iconic and pixel based displays

Outline

Introduction

Two opposite ways of addressing scalability issues:

1. Glyph/Iconic displays:
   - Scatter plot “dots” are replaced by rich icon or glyphs
   - Locality is enforced: all characteristics of an object are mapped in a small area of the display
2. Pixel based displays:
   - A numerical value is mapped to a single pixel
   - Displays are dense: nothing is “vasted”

Outline

Iconic displays

Replacing dots by icons or glyphs:

- Each object is represented by an icon
- Variations in shape/color/etc. of the icon encode variables
- Icon position:
  - Structure driven
  - Data driven
- Examples:
  - Chernoff's faces
  - Star glyph
  - Stick-figure icon
  - etc.

Iconic displays

- Pros:
  - complete independent display of each object
  - pre-attentive processing
  - can be very intuitive (good learning curve)
- Cons:
  - glyph design
  - variable mapping: important variation in feature expressive power
  - glyph positioning
  - glyph overlapping

Chernoff's faces

Proposed by Chernoff in 1973:

- Values mapped to characteristics of faces:
  - eyes, nose, mouth, ears, head, hair, eyebrows, etc.
  - position, size, curvature, etc.
  - combined with color
- Pros:
  - Easy to understand
  - Can convey strong messages
- Cons:
  - Numerous variants
  - Strongly depends on the chosen characteristics
  - Uses a lot of screen surface

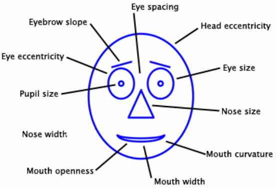
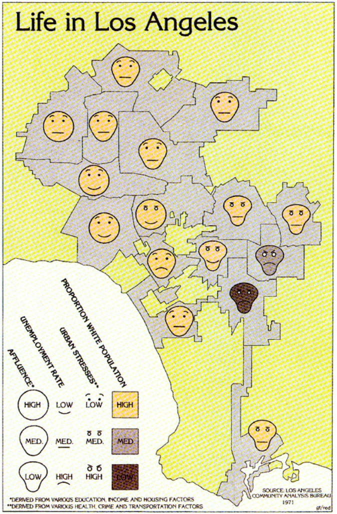
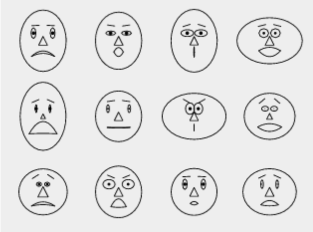
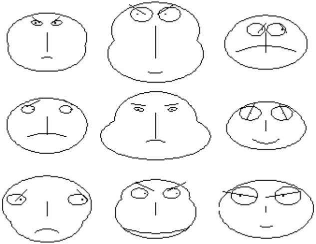


image from `http://kspark.kaist.ac.kr/Human%20Engineering.files/Chernoff/Chernoff%20Faces.htm`

Examples



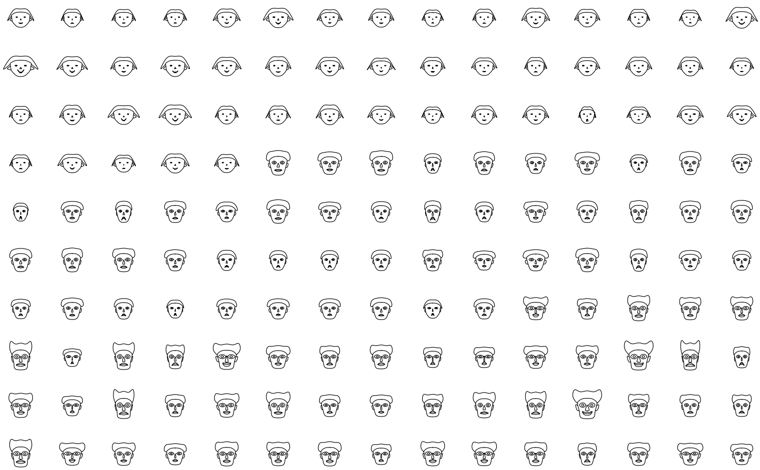




http://kspark.kaist.ac.kr/Human%20Engineering.files/Chernoff/Chernoff%20Faces.htm
http://mathworld.wolfram.com/ChernoffFace.html http://aoki2.si.gunma-u.ac.jp/R/face.html

Iris

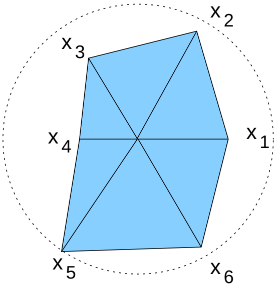
Anderson's/Fisher's Iris



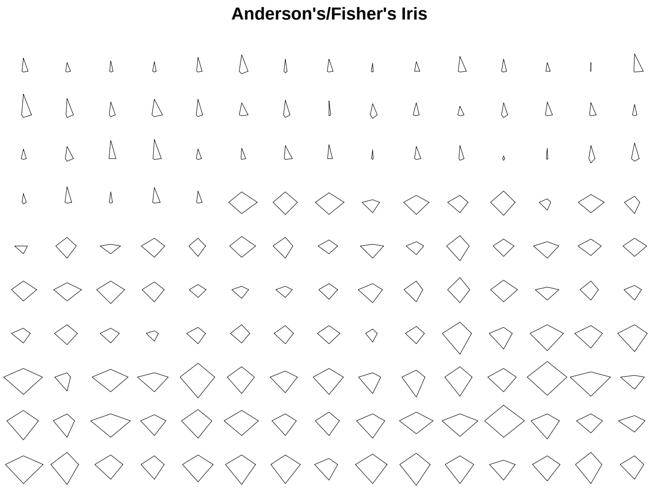
Star Glyph

Proposed by Siegel, Farrell, Goldwyn & Friedman in 1972:

- p variables are mapped to a polygon with $p - 1$ edges (p vertices)
- the distance between the center and vertex i is proportional to x_i
- radii are optional
- corresponds to individual polar parallel coordinates

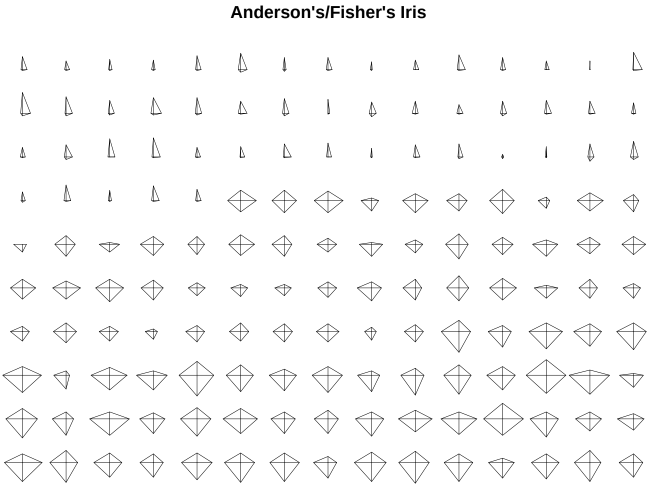


Iris



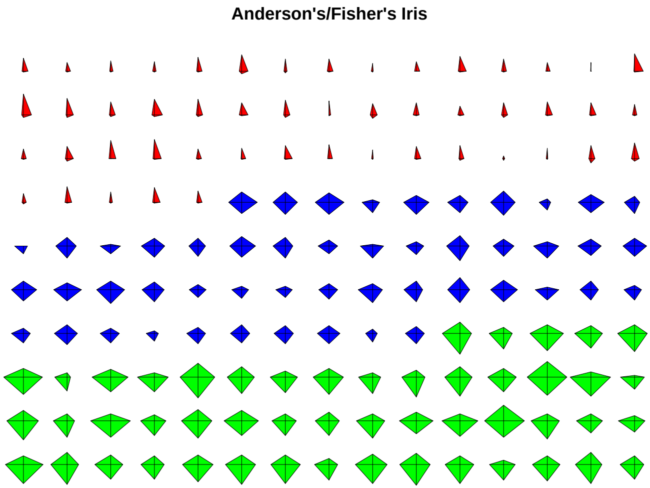
Obviously two classes

Iris



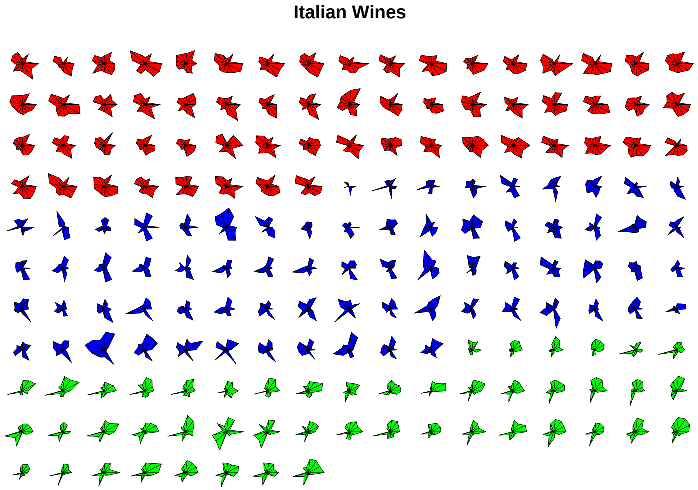
Obviously two classes

Iris



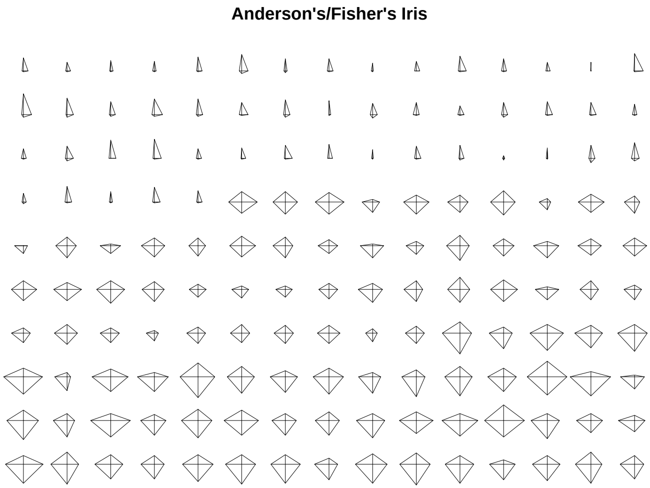
Separable?

Wines

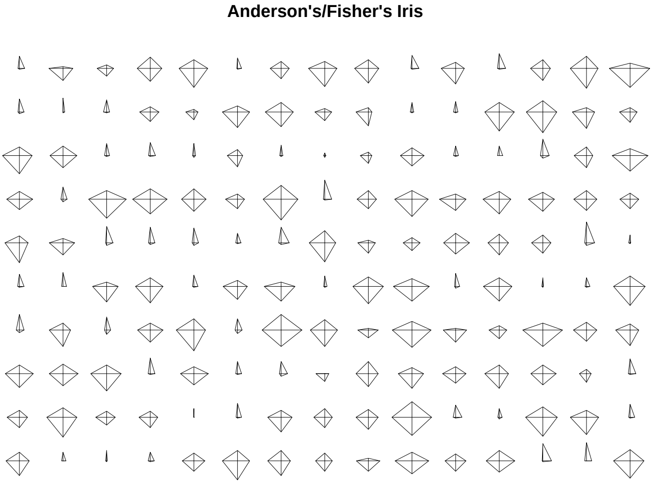


Separable?

Glyph Positioning



Class based order: two classes



Random order: no structure

Glyph Positioning

- Strong impact on perception
- Two classes of algorithms:
  - Structure driven:
    - Prior choice of a structure (e.g., a grid)
    - Layout in the structure can be data driven (ordering)
  - Data driven:
    - Positions of the glyphs is extracted from the data
    - User guided (scatter plot like)
    - Automated (e.g., via a projection)
- Overlapping is a major problem:
  - Additional constraints for data driven solution
  - Rendering solutions (transparency, auto-size, post processing)

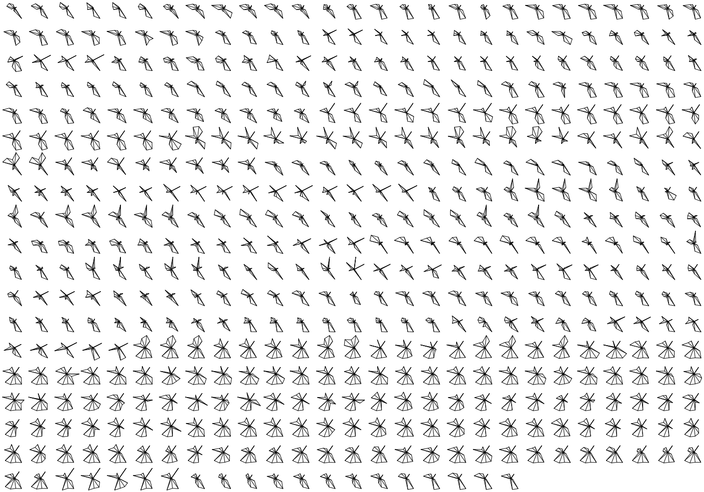
Structure driven

Based on a prior structure

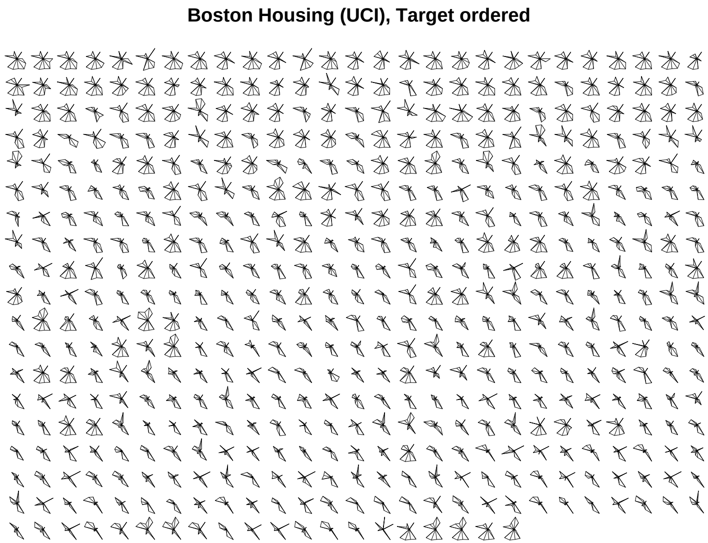
- No overlapping
- “Reading order”
- Glyph ordering:
  - A seriation problem (like variable ordering)
  - User guided (target variable, chosen input variable, etc.): also called query based order
  - Automated:
    - Minimize a dissimilarity between close glyphs
    - NP complete
    - Heuristics

Structure driven with user ordering

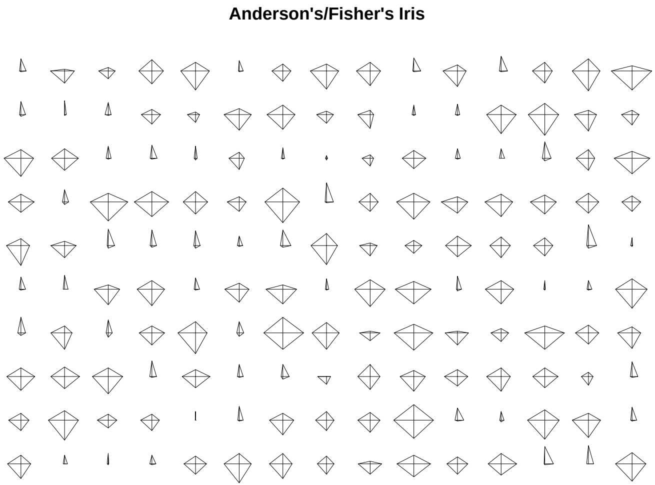
Boston Housing (UCI)



Structure driven with user ordering

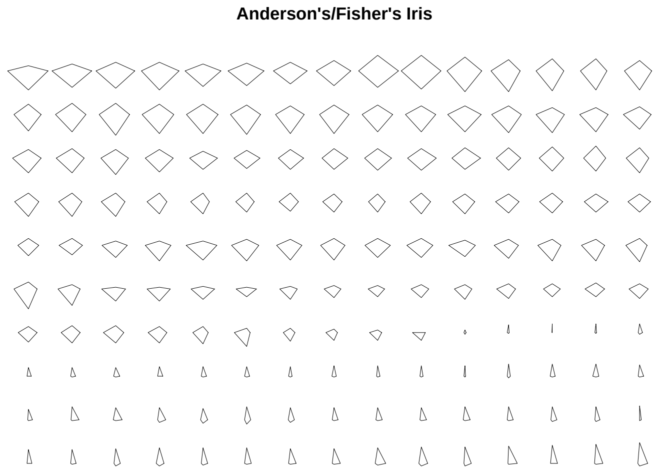


Structure driven with automated ordering



Random order: no structure

Structure driven with automated ordering



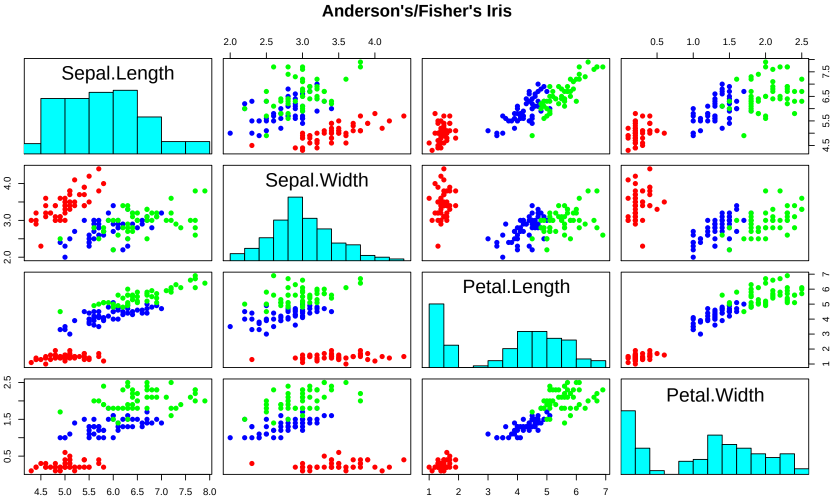
Hierarchical clustering based order: two classes

Data driven

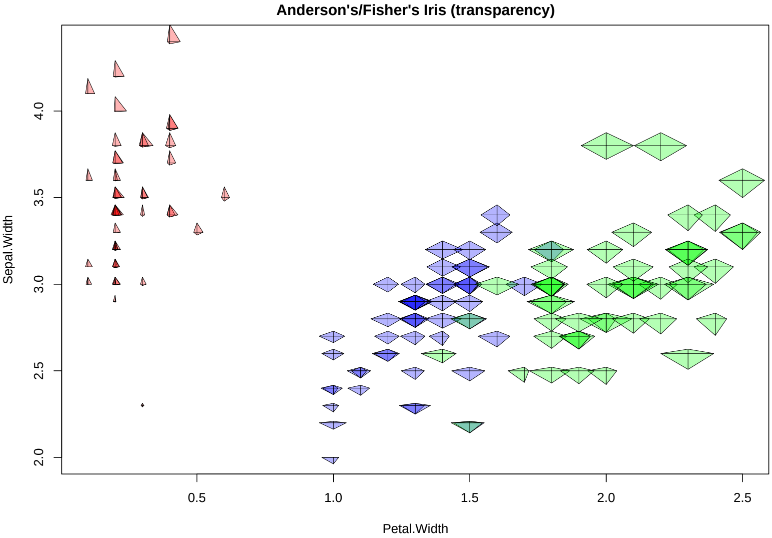
Based on the data

- Coordinates of glyphs are computed from the data
- User guided:
  - Choose two variables among p
  - Scatter plot of the glyphs
- Automated: projection based approaches
- Overlapping:
  - Rendering (as for scatter plots)
  - Built in prevention (e.g., SOM, Relational Perspective Map)
  - Post processing:
    - Jitter, GridFit, Force directed placement, etc.
    - Adapted size: small glyphs in dense area, larger ones in empty places
    - Zooming interface and clustering

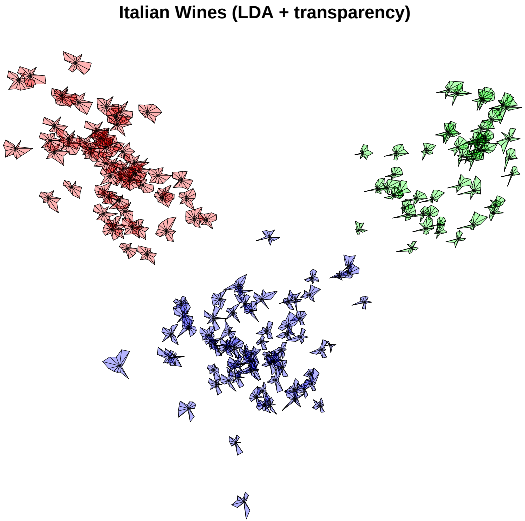
Data driven under user control



Data driven under user control

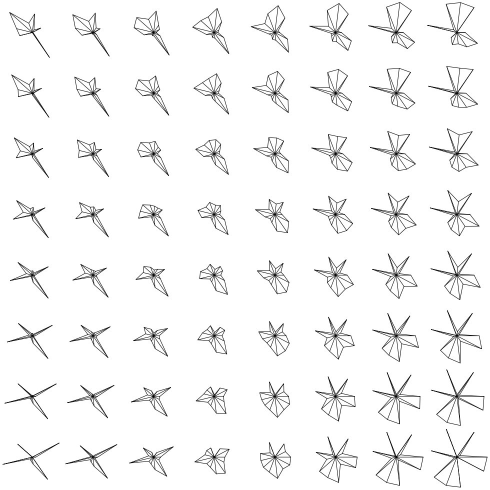


Automated data driven



Automated data driven

Self Organizing Map
(Boston Housing)



Post-processing

Reducing overlapping after positioning

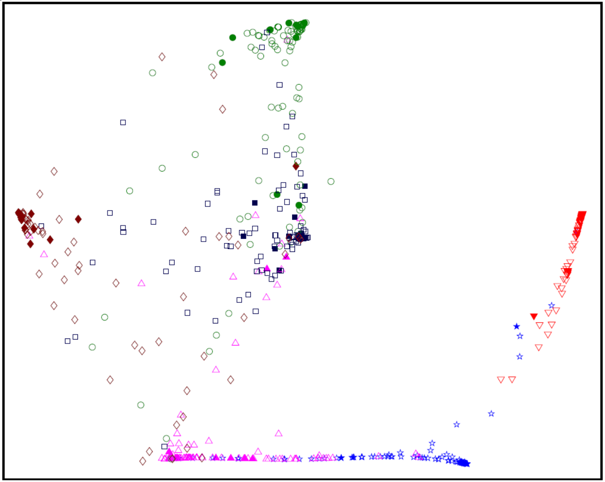
- Optimization problem:
  - original glyph positions: x_i
  - new positions: y_i
  - movement minimization: $y_i \simeq x_i$
  - overlapping minimization: $y_i \neq y_j$ when $i \neq j$
- Objective function

$$E(\mathbf{y}) = \sum_{i=1}^{n} \|y_i - x_i\|^2 + \eta \sum_{i \neq j} \exp\left(-\frac{\|y_i - y_j\|^2}{2\sigma^2}\right)$$

- η: trade off between faithfulness and overlapping (σ: overlapping radius)
- N-body like problem (frequent also in projection)

Separation

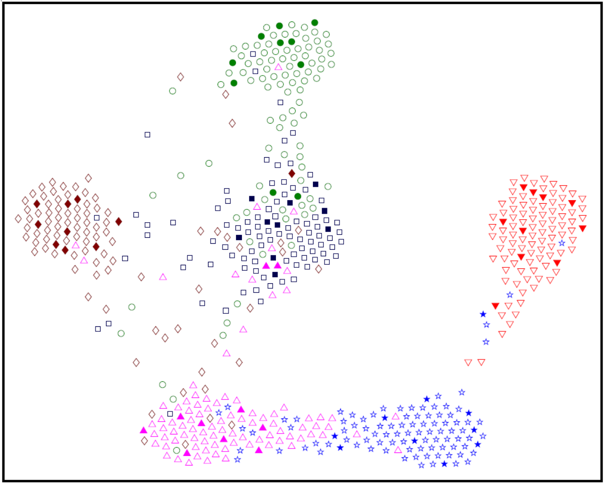
Original layout



from “Visual nonlinear discriminant analysis for classifier design” by Iwata et al., ESANN 2006

Separation

After optimization

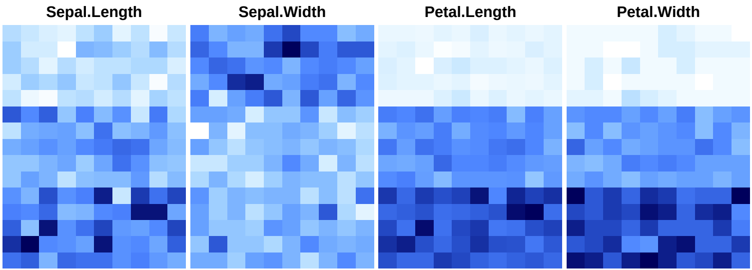


from “Visual nonlinear discriminant analysis for classifier design” by Iwata et al., ESANN 2006

Outline

Dense pixel displays

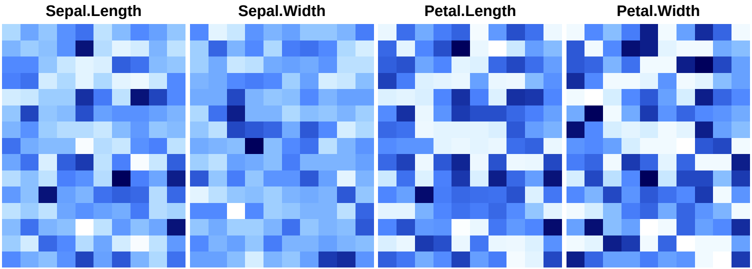
- One pixel per value
- n objects with p variables $\Rightarrow n \times p$ pixels
- Values are color coded
- One block of pixels per variable



Iris

Dense pixel displays

- Pros:
  - Very good scalability
  - No overlapping
  - Pre-attentive processing
- Cons:
  - Extremely order dependent



Iris (random order)

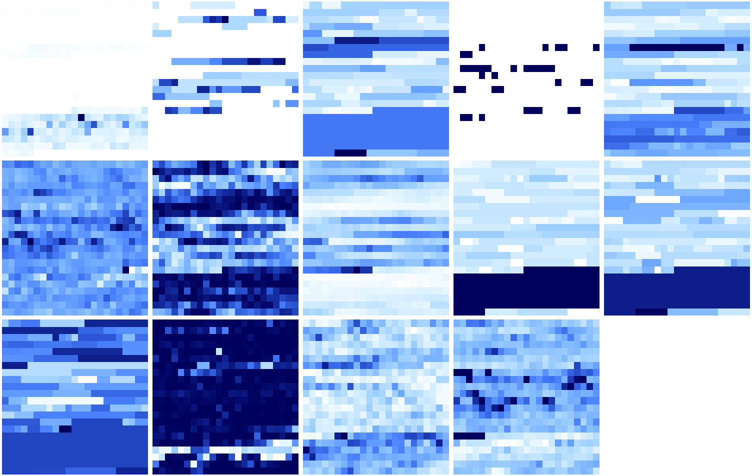
Ordering and layout

Two distinct problems

- Pixel layout for one variable:
  - Object order (user chosen or based on seriation)
  - Order mapping: grid “reading order” (same problem for glyphs)
- Variable layout:
  - “Related” variables should be close
  - Problem already encountered for parallel coordinates
  - NP complete

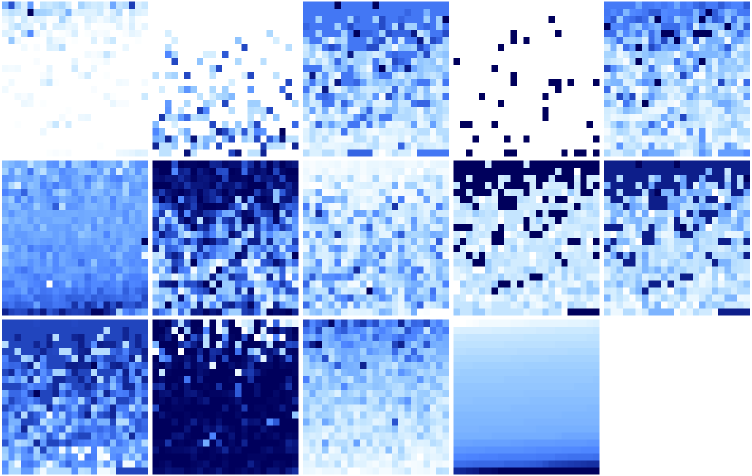
User directed pixel ordering

Boston Housing (no pixel order)



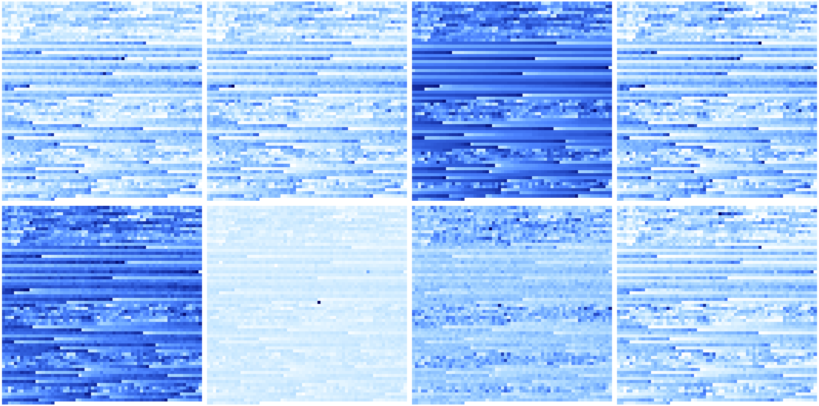
User directed pixel ordering

Boston Housing (target pixel order)



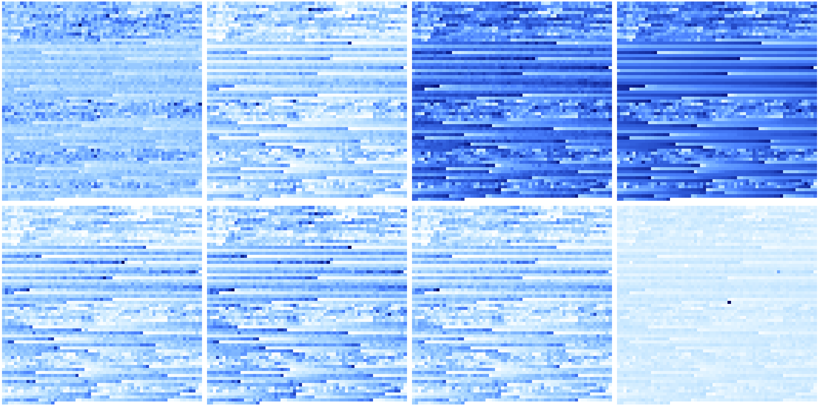
Automatic ordering

Abalone (random variable order)



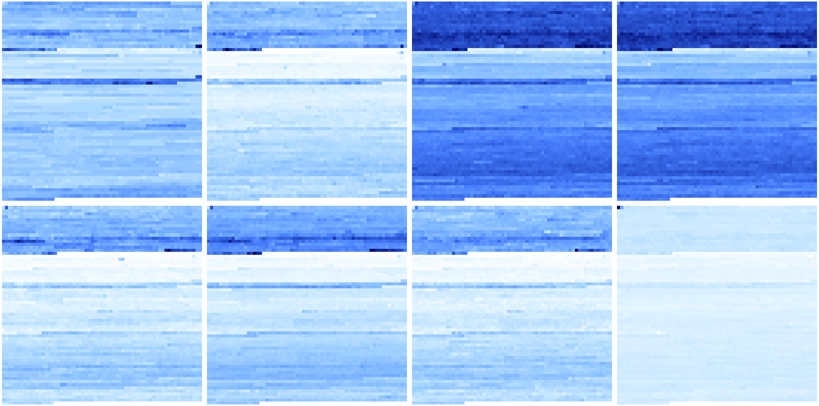
Automatic ordering

Abalone (ordered variables)



Automatic ordering

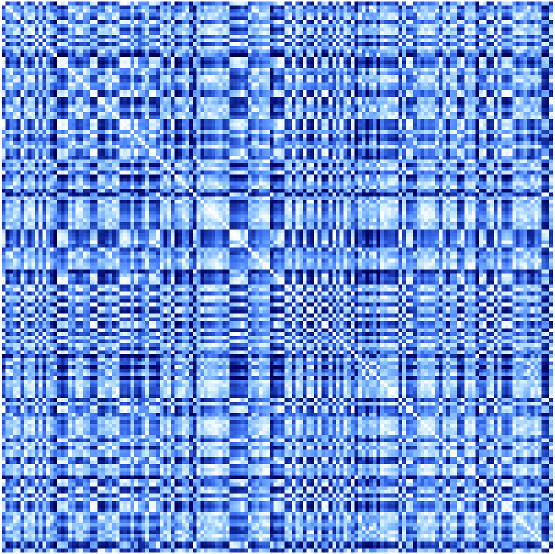
Abalone (ordered on variables and pixels)



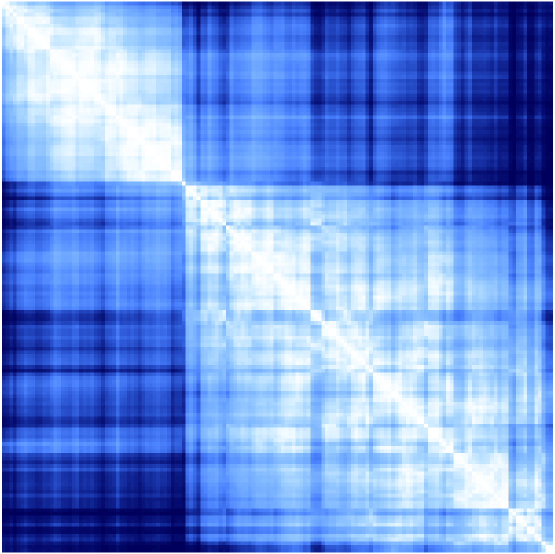
Dissimilarity matrix

- Pairwise (dis)similarities between objects
- One pixel per dissimilarity: n^2 pixels ($\frac{n(n-1)}{2}$)
- Square matrix display
- Values are color coded
- Ordering is mandatory:
  - random order gives random results
  - many different objective functions:
    - correlation between matrix rows
    - “small” diagonal
    - etc.
  - YAFSP (Yet Another Family of Seriation Problems)
  - YAFNPCP (Yet Another Family NP Complete Problems)

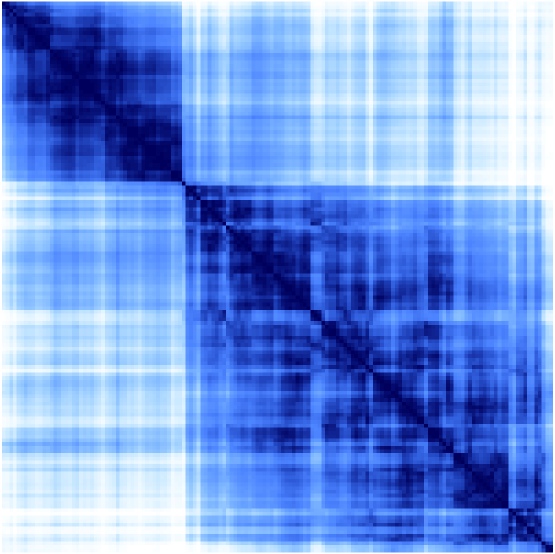
Iris dataset



Iris dataset



Iris dataset

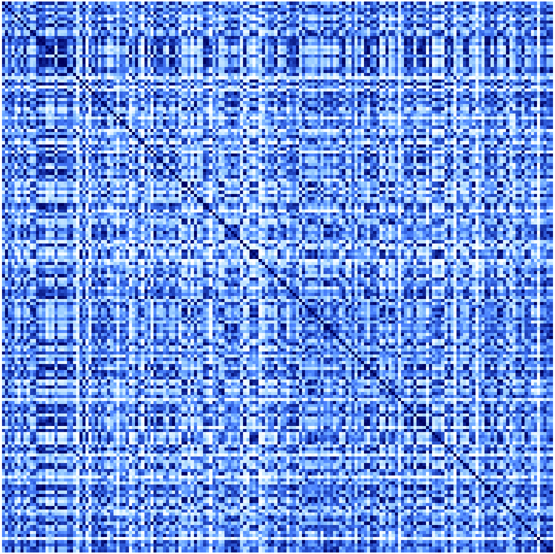


Matrix visualization

- Pros:
  - Dimension Independent (but beware of distances problems in high dimension)
  - Applies to non vector data
  - Clustering analysis
  - Outlier detection
- Cons:
  - Ordering algorithm are costly (NP complete!)
  - Scalability impaired by ordering costs and by screen occupation
  - Results depends on many parameters:
    - Dissimilarity
    - Ordering criterion
    - Ordering method
    - Color map
  - Usability has not been studied

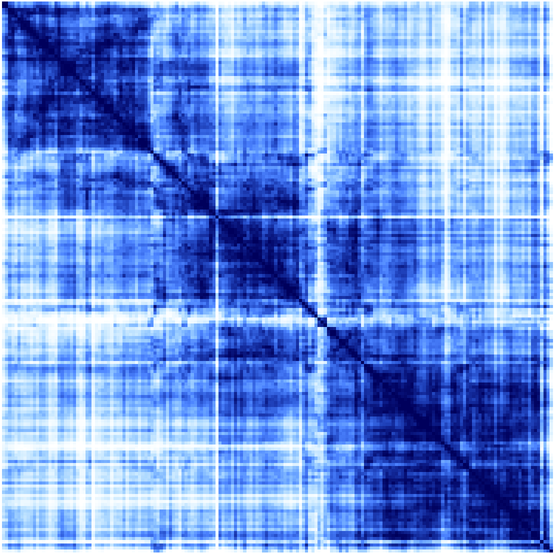
Wine dataset

Random order



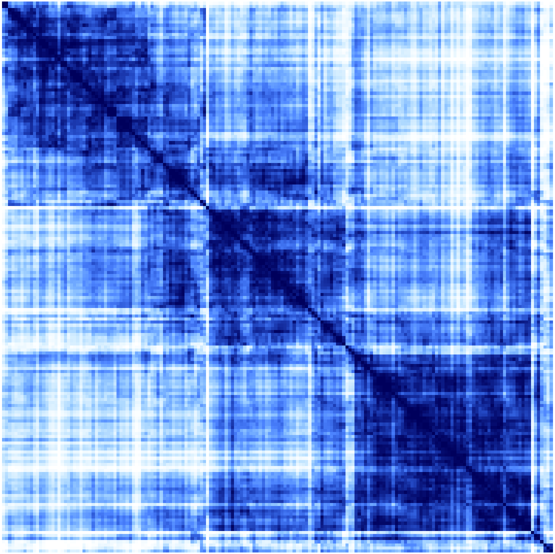
Wine dataset

Correlation similarity, complete linkage, optimal leaf order

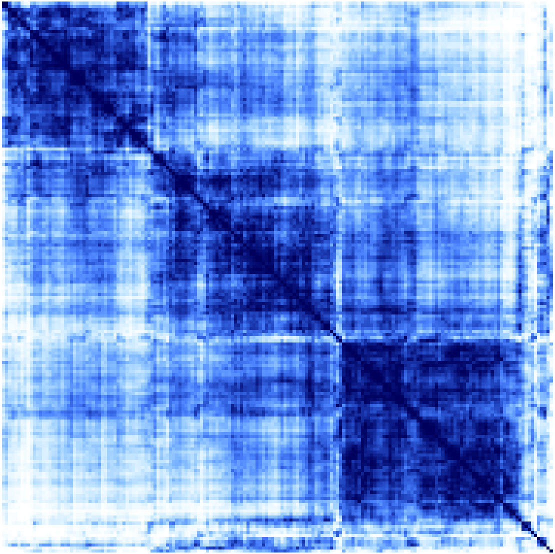


Wine dataset

Correlation similarity, average linkage, optimal leaf order



Wine dataset



Euclidean dissimilarity, average linkage, optimal leaf order

Wine dataset

Euclidean dissimilarity, average linkage, optimal leaf order, linear color map