Bioinformatics II
Winter Term 2016/17

universitätbonn
Rheinische
Friedrich-Wilhelms-
Universität Bonn

Chapter 3:
Multidimensional Data Visualization

Jun.-Prof. Dr.-Ing. Thomas Schultz

URL: http://cg.cs.uni-bonn.de
E-Mail: schultz@cs.uni-bonn.de
Office: Friedrich-Ebert-Allee 144, 53113 Bonn

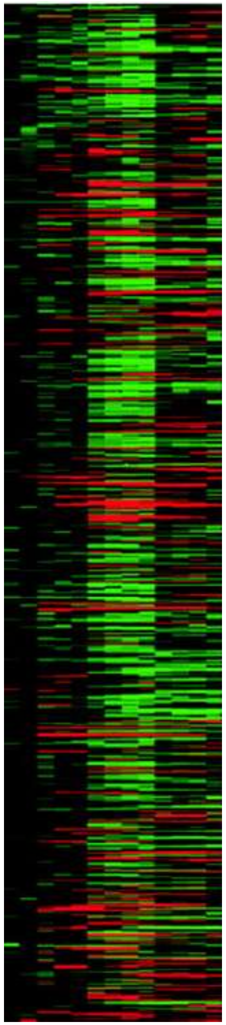
November 15, 2016

Motivation: Multidimensional Data Vis

- In science, we frequently deal with data items that have multiple attributes
  - e.g., expression levels of a large number of genes
- 2-3 attributes can be mapped to space
- ~3-10 attributes can be visualized using multidimensional visualization techniques
- Some techniques support even more (~100) attributes
- Topic of today‘s lecture, but first some basics...

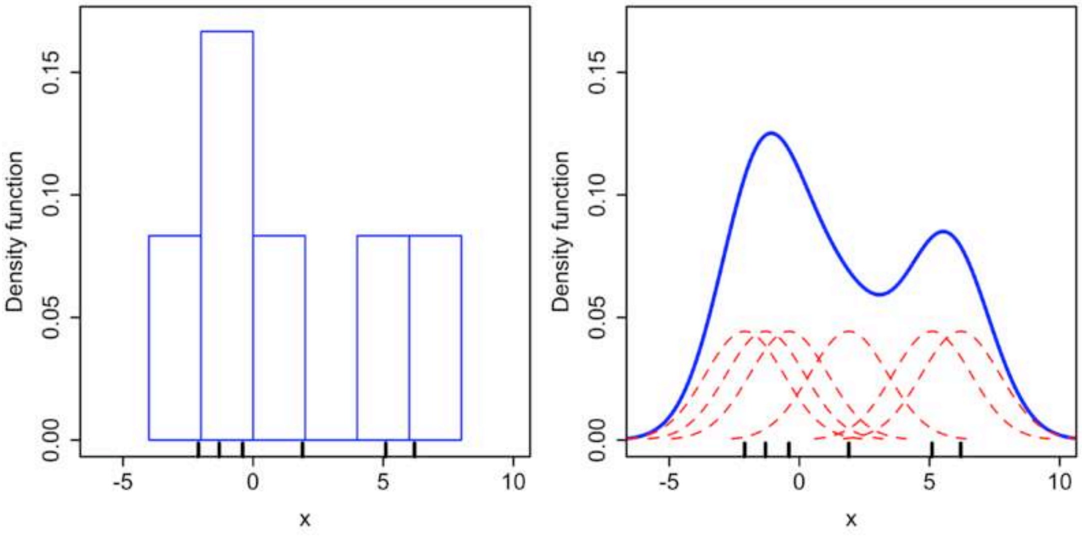
Heat Maps

- **Heat maps** are a direct visual representation of matrices
  - Colors encode numbers in each cell
  - Example [Eisen et al. 1998]:
    - Rows = Different genes
    - Columns = Different times after treatment
    - Values = log of relative expression level
- Refer to Chapter 2 when picking colors!
  - Example: Diverging color scheme green-black-red



Histograms and Kernel Density Estimation

- **Histogram:** Bar chart of number (or fraction) of values $\mathbf{x}_i$ in predefined ranges ("bins")
  - Result can depend strongly on bin width
  - If $\mathbf{x}_i$ are samples from a probability density function (PDF), histogram is a piecewise constant estimate of the PDF
- **Kernel Density Estimation (KDE):**
  - Informally, "smooth version of a histogram"



Sketch: Wikipedia

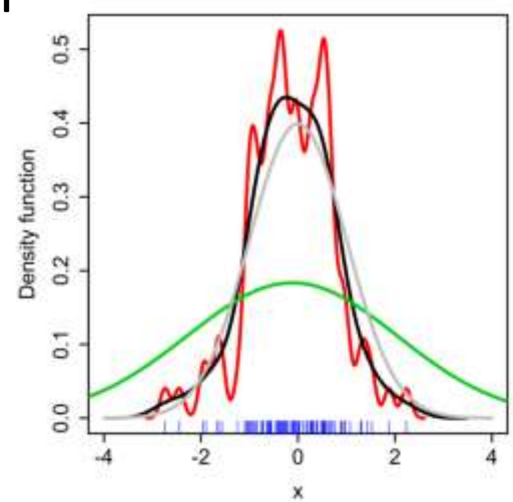
KDE: Formal Definition

- Given values $\mathbf{x}_i$, KDE is defined by

$$f(\mathbf{x}) = \frac{1}{n}\sum_i K(\mathbf{x} - \mathbf{x}_i)$$

  – Kernel K is normalized to integrate to unity, e.g., standard normal $K(x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{x^2}{2h^2}}$

- **Bandwidth** h controls smoothness of f:
  - Analogous to number of bins in histogram
  - h too small: estimate "rough" and noisy
  - h too large: oversmoothing
  - h should decrease with larger number n of samples
  - Silverman's rule of thumb: $h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}}$

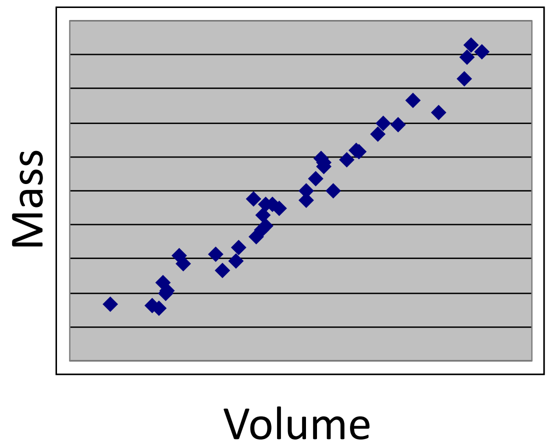


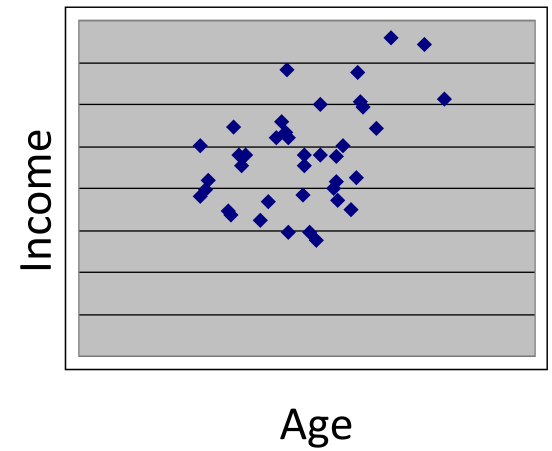
Sketch: Wikipedia

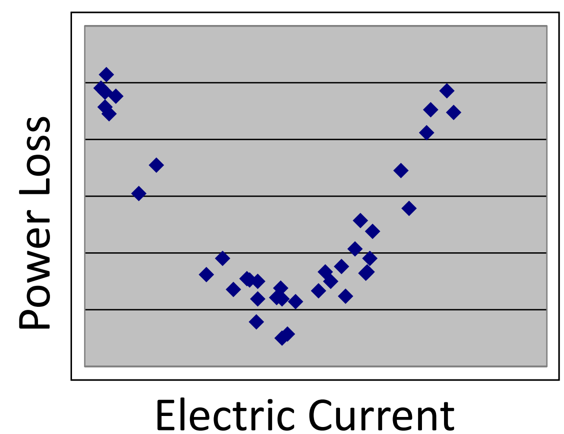
Section 3.1: Scatterplots

Scatterplots

- Scatter plots are useful for visually recognizing trends and correlations between pairs of variables



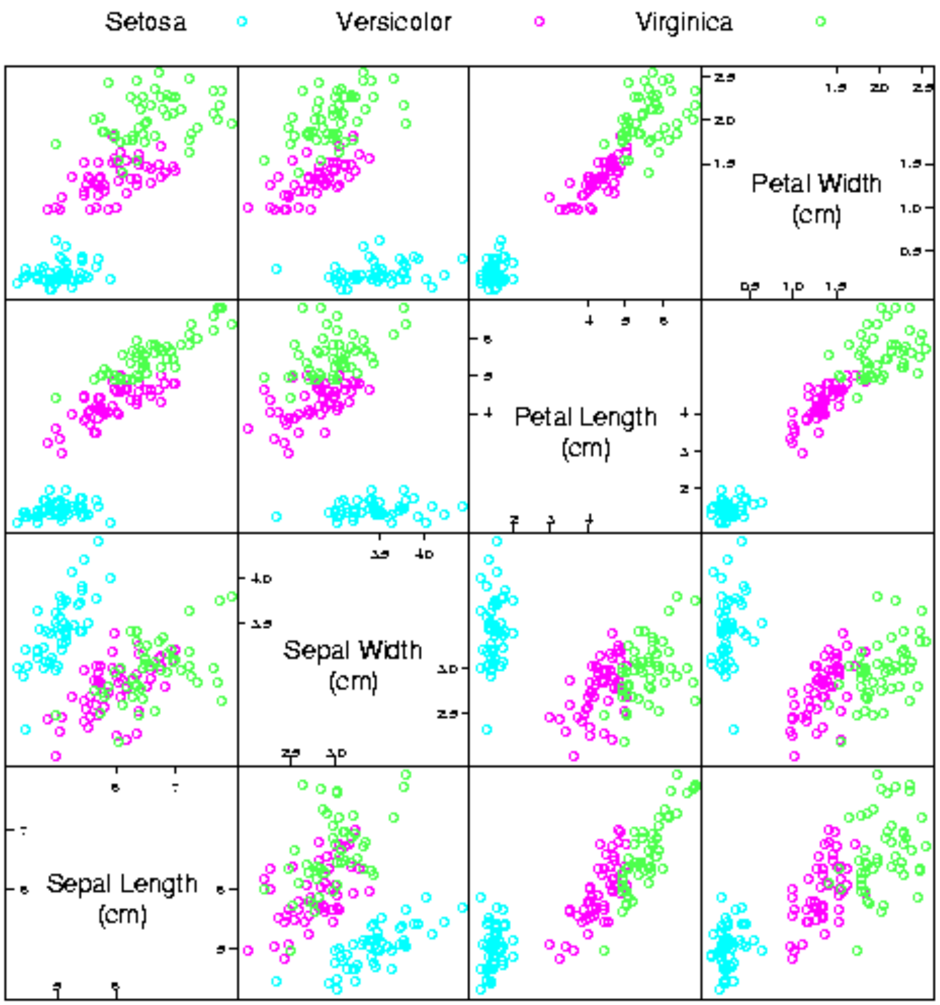




Scatterplot Matrices (SPLOM)

Produce scatterplots for all pairs of variables and place them into a matrix

Total of $(k^2-k)/2$ scatterplots



Order in SPLOMs

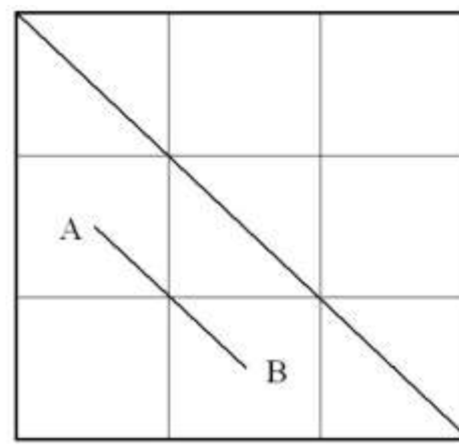
- Re-ordering the dimensions does not change the projections shown in a SPLOM, it only permutes individual plots
- [Peng et al. 2004] find layout so that similar scatterplots are located close to each other
  - Distinguish between *high-cardinality dimensions* (number of possible values > number of points) and *low-cardinality dimensions*
  - Sort low-cardinality by number of values
  - Rate ordering of high-cardinality dimensions based on their correlation

SPLOMs: Rating High-Cardinality Order

- Pearson Correlation Coefficient measures degree of *linear* dependence between **x** and **y**

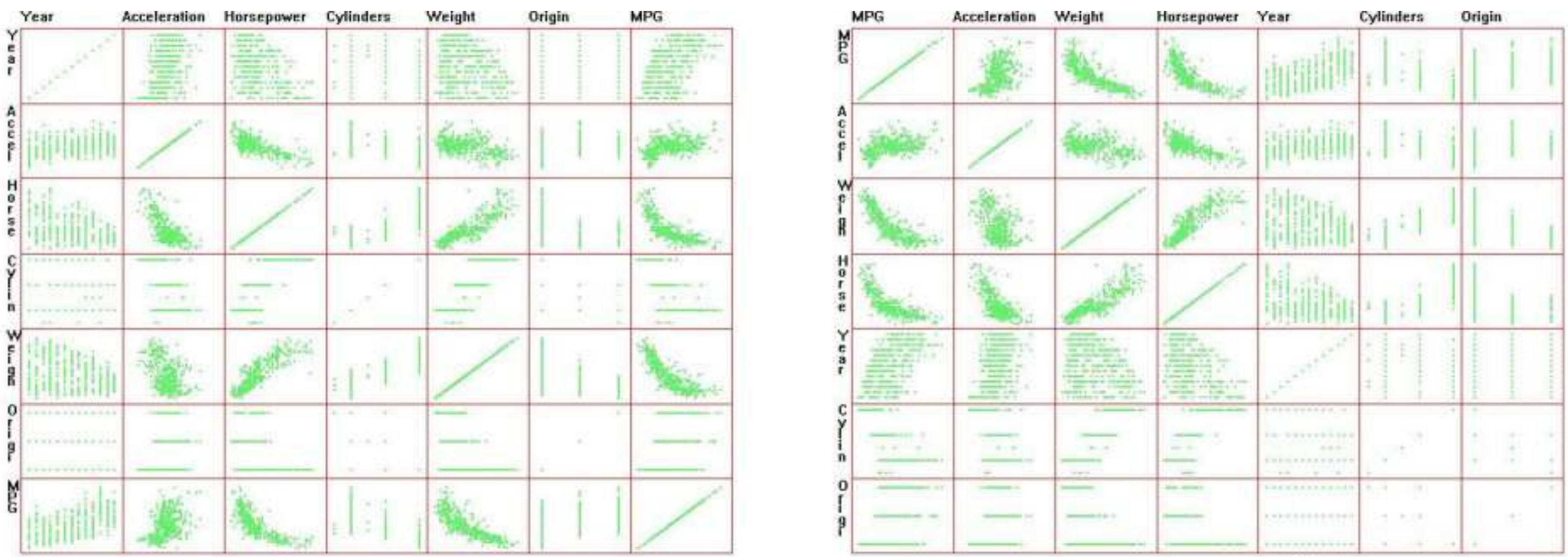
$$\rho_{xy} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$$

- Clutter measure of a SPLOM:
  - For each pair *(x,y)* of high-cardinality dimensions, find all other pairs *(x',y')* with $\left|\rho_{xy} - \rho_{x'y'}\right| < \epsilon$
  - Add the Euclidean distances of all those plots in the SPLOM (the smaller, the better)



SPLOMs with "Clutter Reduction"

- Exhaustive optimization requires $O(n^2 \cdot n!)$ computation
- Alternative heuristic: Random swapping

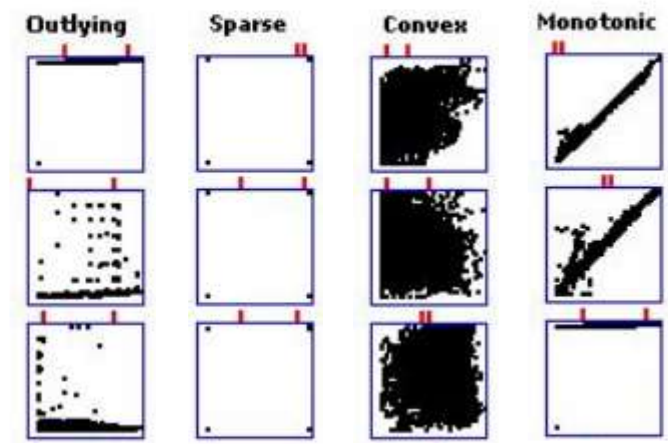


(a) (b)

Image taken from [Peng et al. 2004]

Ranking by Features

- To look for scatterplots that exhibit specific patterns, we can rank them by **features** such as
  - Fraction of outliers
  - Sparsity
  - Convexity
  - Monotonicity
  - etc.

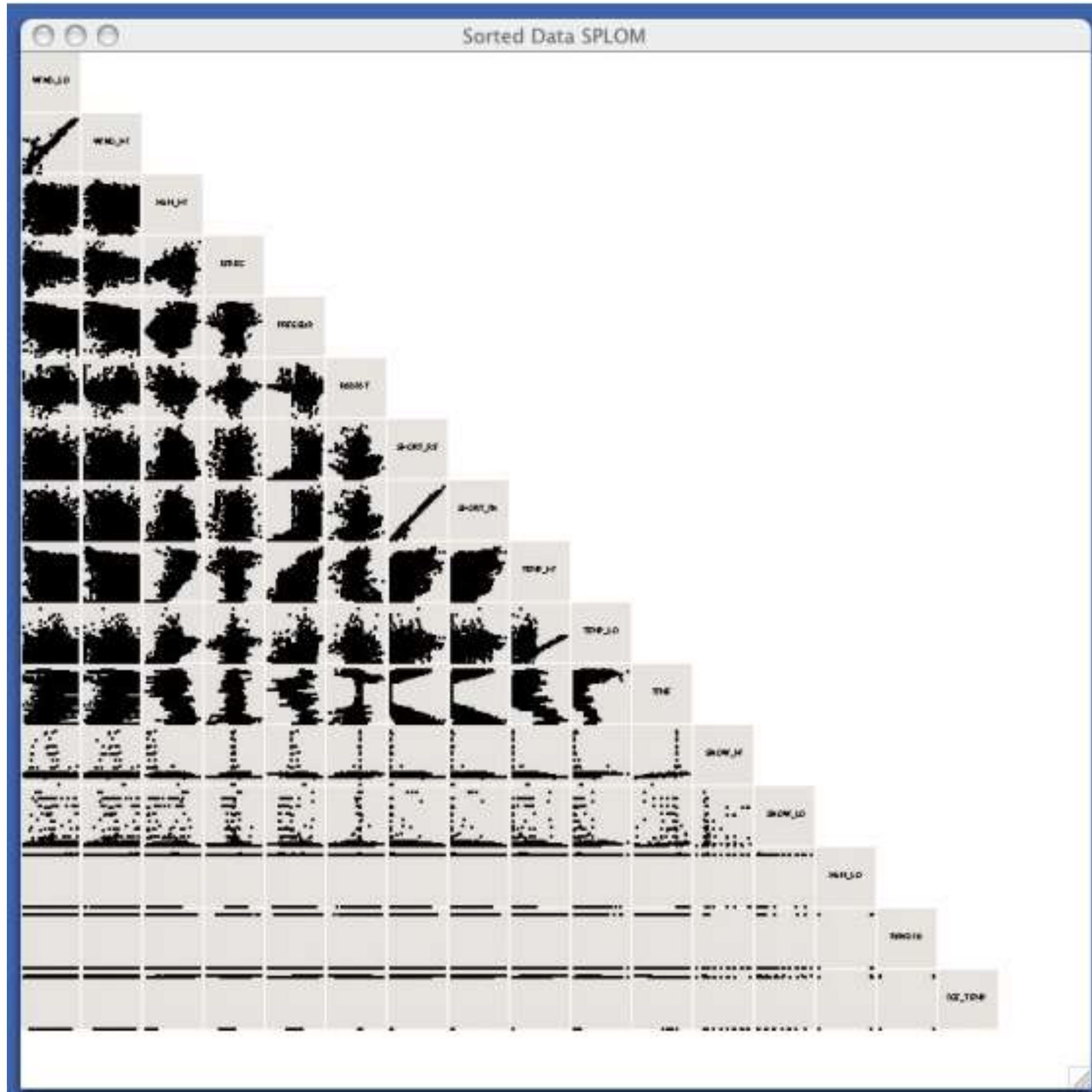


- See [Wilkinson et al. 2006] for a more complete overview and how to compute them

Images from [Wilkinson et al. 2006]

Sorting by Features

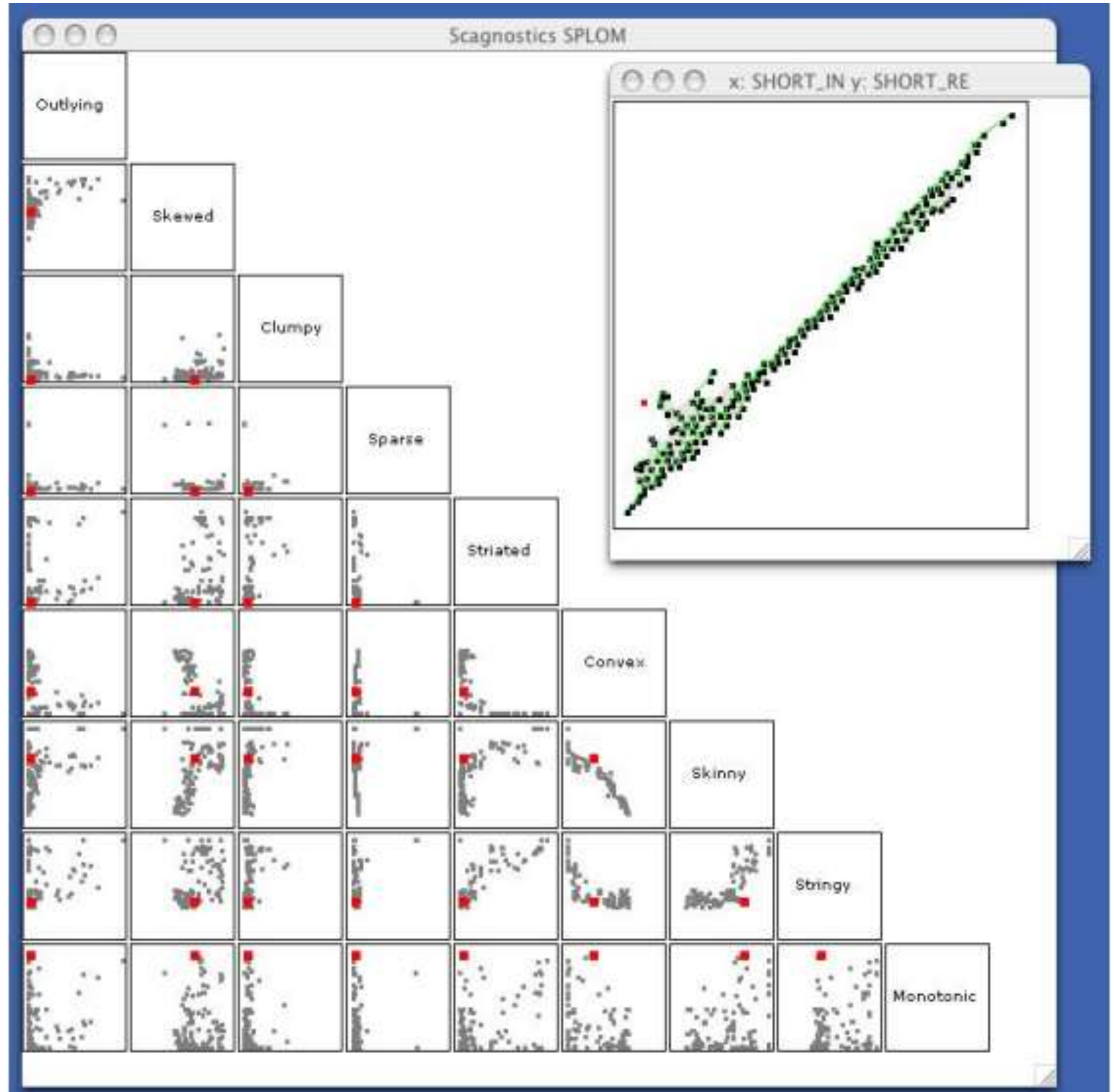
- Computing many features and sorting SPLOMs according to the **principal PCA mode** is an alternative way to group similar plots together



Images from [Wilkinson et al. 2006]

Scagnostics

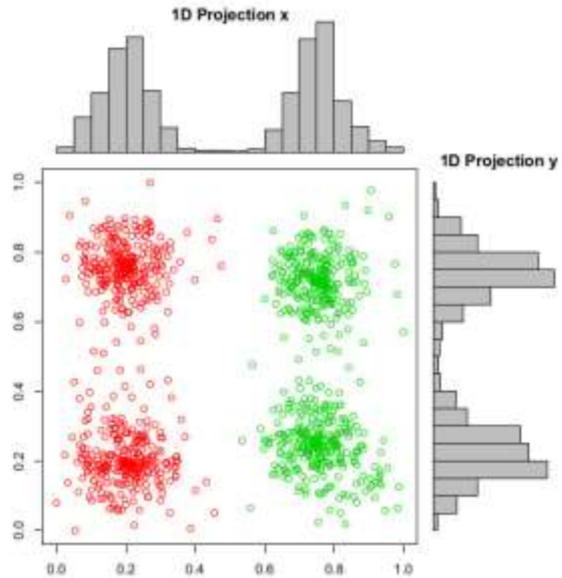
- **Scagnostics** creates a second-level „feature“ SPLOM to visualize scatterplots from the original „data“ SPLOM
- Links back to data SPLOM

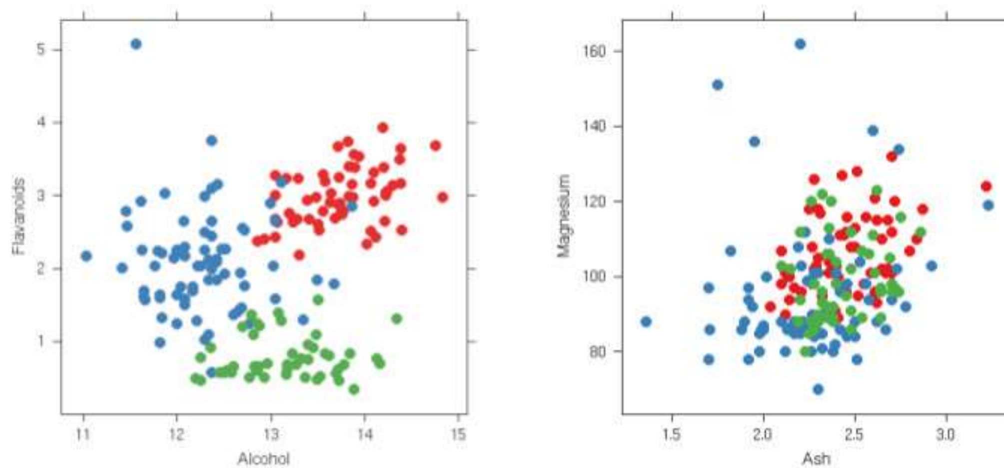


Images from [Wilkinson et al. 2006]

Selecting Good Views

- Select scatterplots in which classes are well-separated
  - Labels can be part of the data or obtained by clustering (Chapter 3.4)





(a) **DSC**=90 (b) **DSC**=49

Images from [Sips et al. 2009]

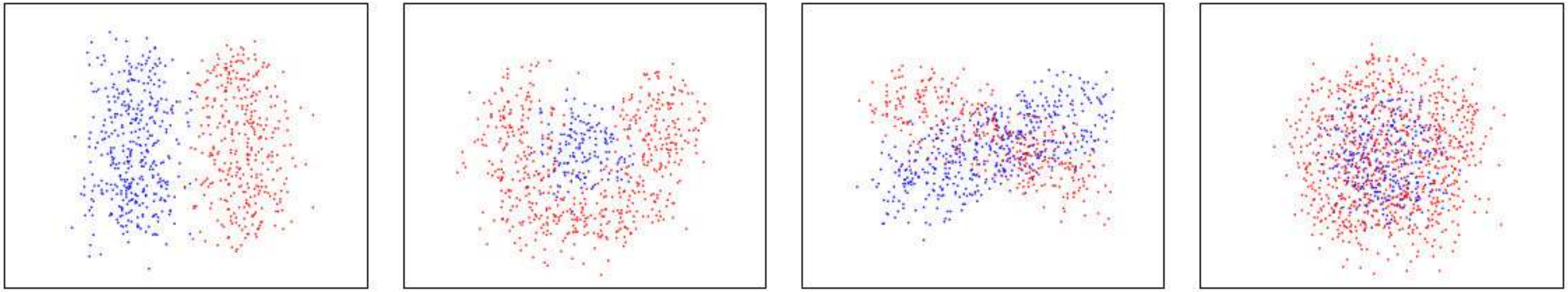
Distance and Distribution Consistency

- **Distance consistency (DSC)**
  - Measures how many percent of all points are – in the given projection – closer to their own cluster center than to all others
  - Fast and simple to compute
  - Assumes spherical clusters

- **Distribution consistency (DC)**
  - Based on penalizing local entropy (amount of uncertainty / constraint violation) in high-density regions
  - Does not assume particular cluster shapes

Distribution Consistency

- Definition of distribution consistency
  - Local entropy: $$H(x,y) = -\sum_{c \in C(X)} \frac{p_c}{\sum p_c} \log_2\left(\frac{p_c}{\sum p_c}\right)$$
    - Reminder: $H \in [0, \log_2 |C|]$
  - Distribution consistency: $$\mathbf{DC} = 100 - \frac{1}{Z} \sum_{x,y} p(x,y) H(x,y)$$
    - $Z = \sum_{x,y} p(x,y) \log_2 |C| /100$
  - *p* estimated using Kernel Density Estimation

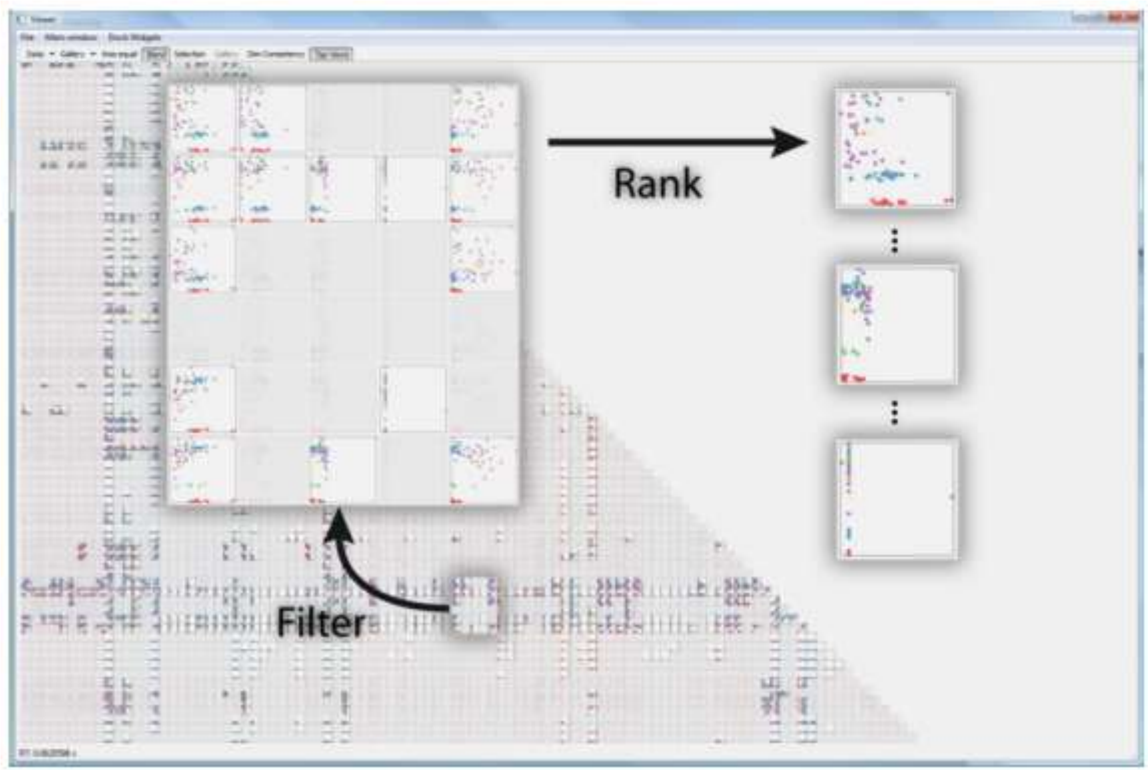


(a) 99 (b) 74 (c) 51 (d) 29

Images from [Sips et al. 2009]

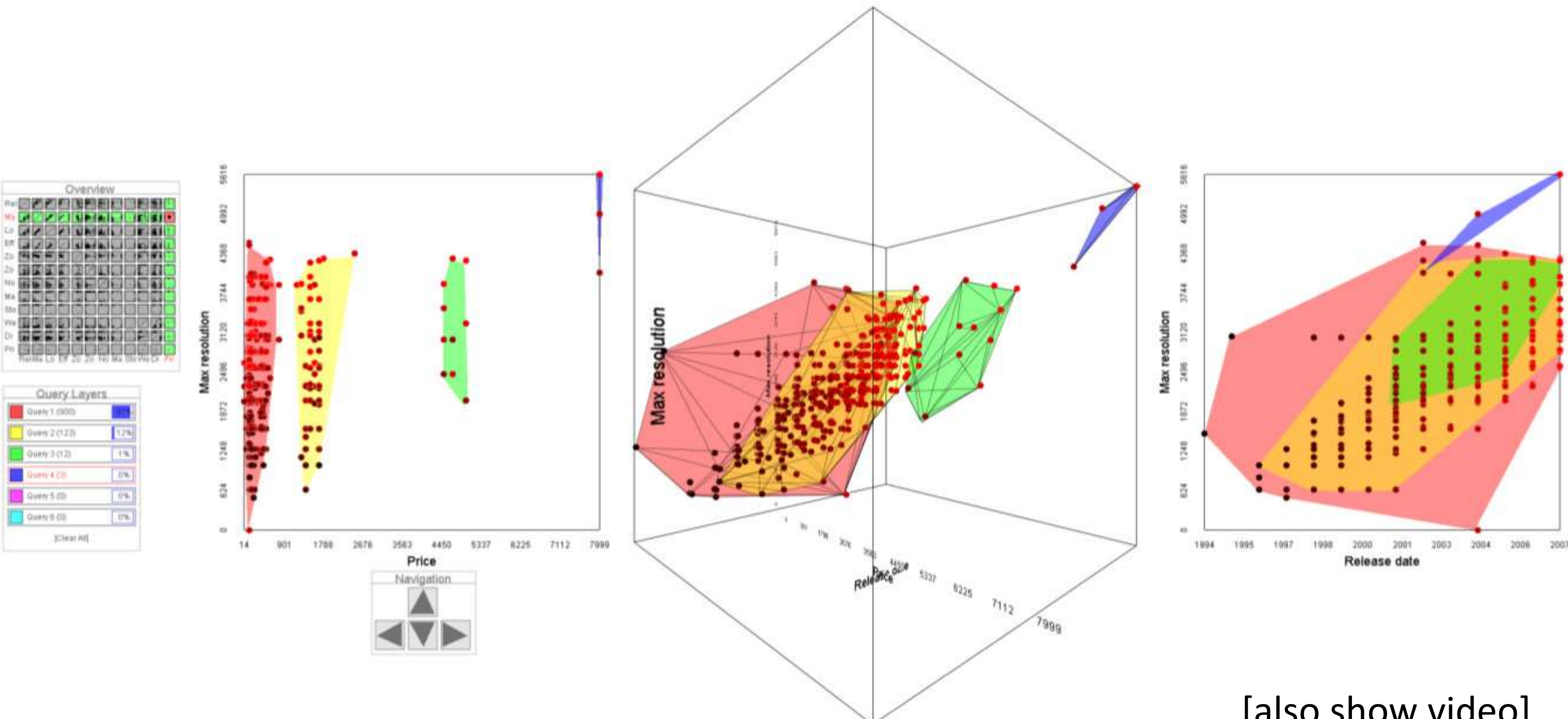
DC: WHO Example

- **WHO data**
  - 194 countries, 159 attributes, 6 HIV risk groups (>12,000 unique scatter plots)
  - Focusing on DC > 80 eliminates 97% of the plots
  - Highlighted rows: Single discriminative attributes
    - *Example:* Total health expenses



SPLOM Navigation

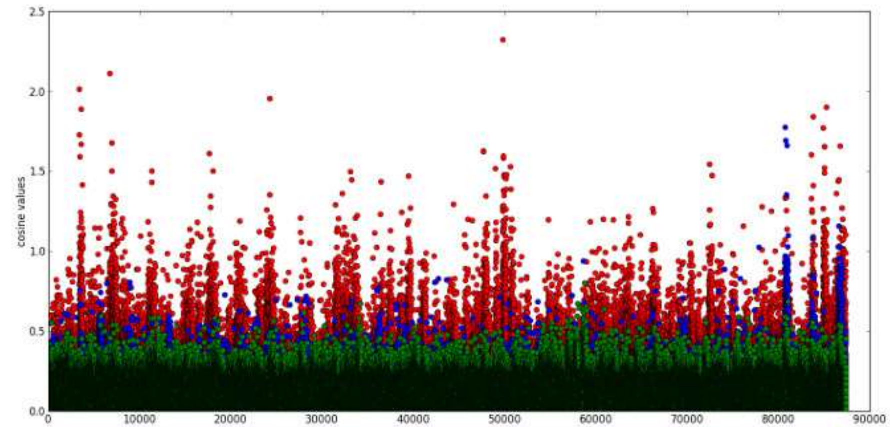
- Animated 3D transitions between neighboring views [Elmqvist et al. 2008]
  - Analogy: “Rolling the dice”

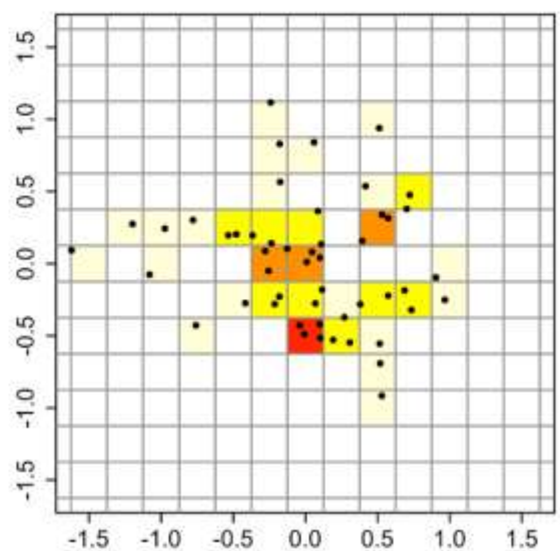


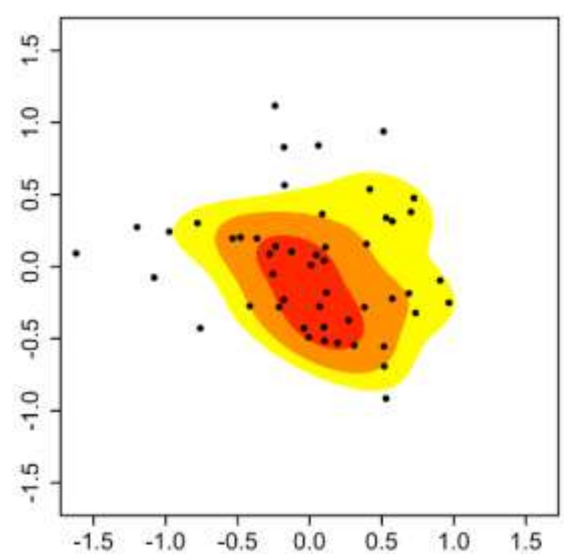
[also show video]

Scatterplots with Many Points

- Too many data points lead to **overdraw**
  - Color code 2D histogram (“heat map”)
  - Kernel Density Estimation
  - *Disadvantage:* Cannot see individual points



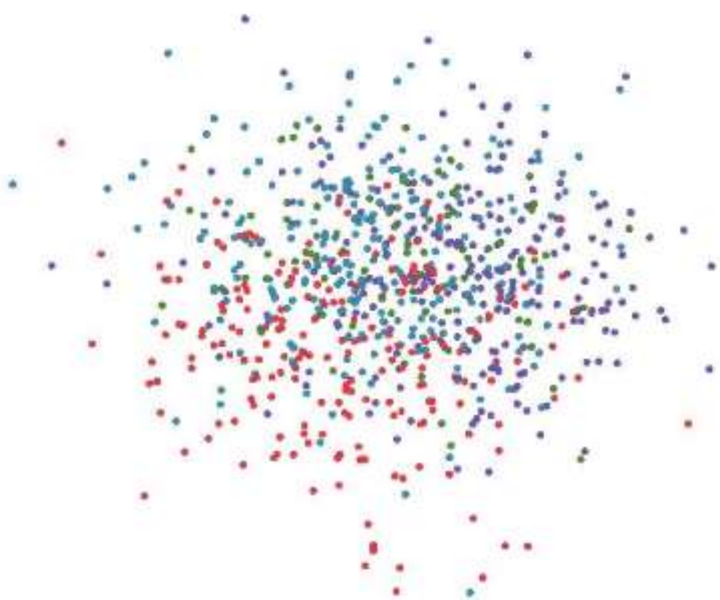


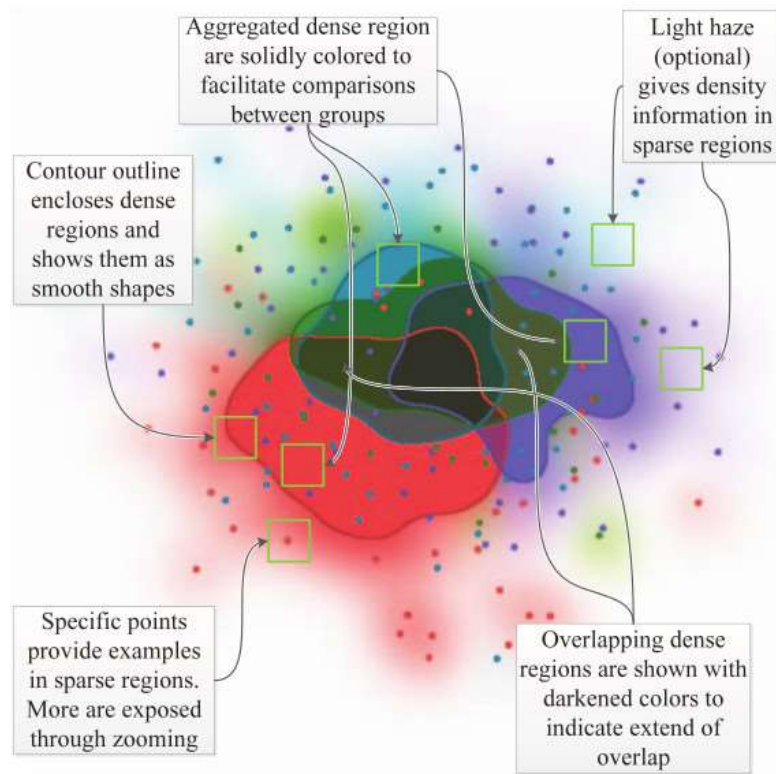


Images from drleft / Wikimedia

Splatterplots

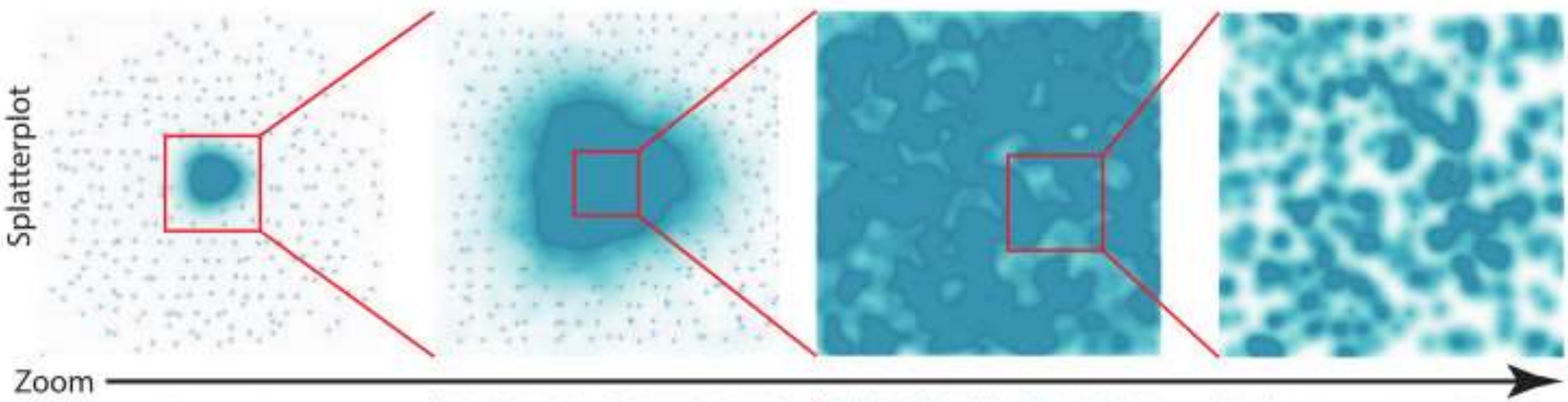
- Splatterplots [Mayorga / Gleicher 2013]
  - Visual abstraction in image space, detail added when zooming in





Splatterplots: Dense Regions

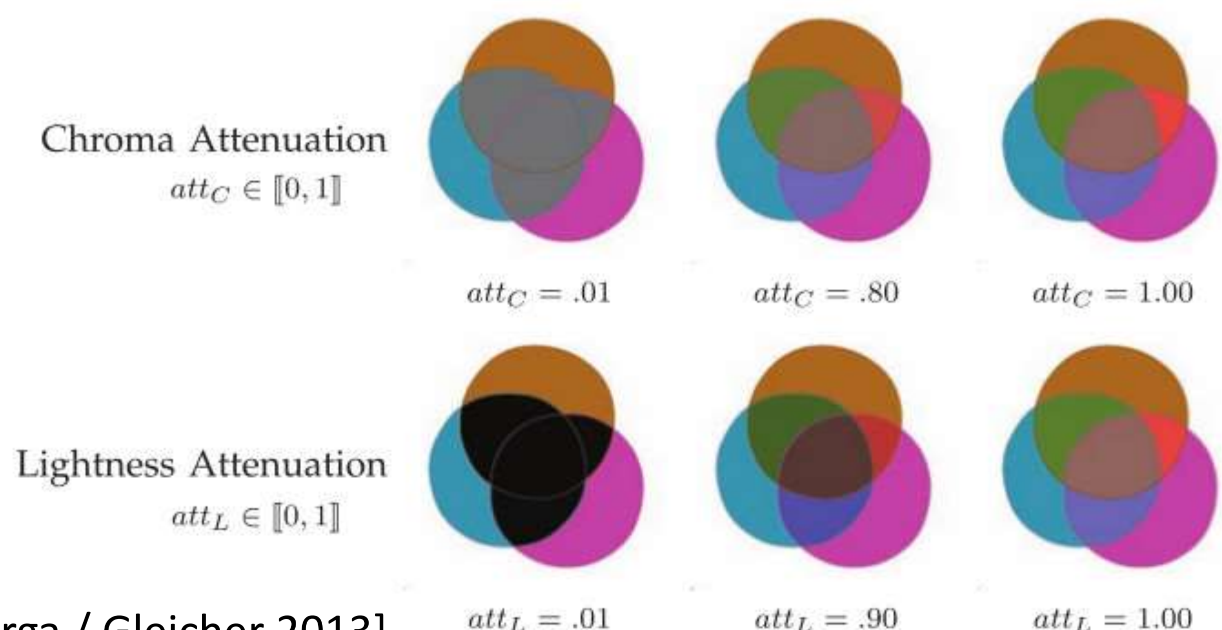
- Based on **Kernel Density Estimation**
  - Kernel width defined *in screen space*
  - High-density regions shown as smooth, filled, and bounded shapes
  - *Drawback:* Shape can depend on density threshold



Images from [Mayorga / Gleicher 2013]

Splatterplots: Color Blending

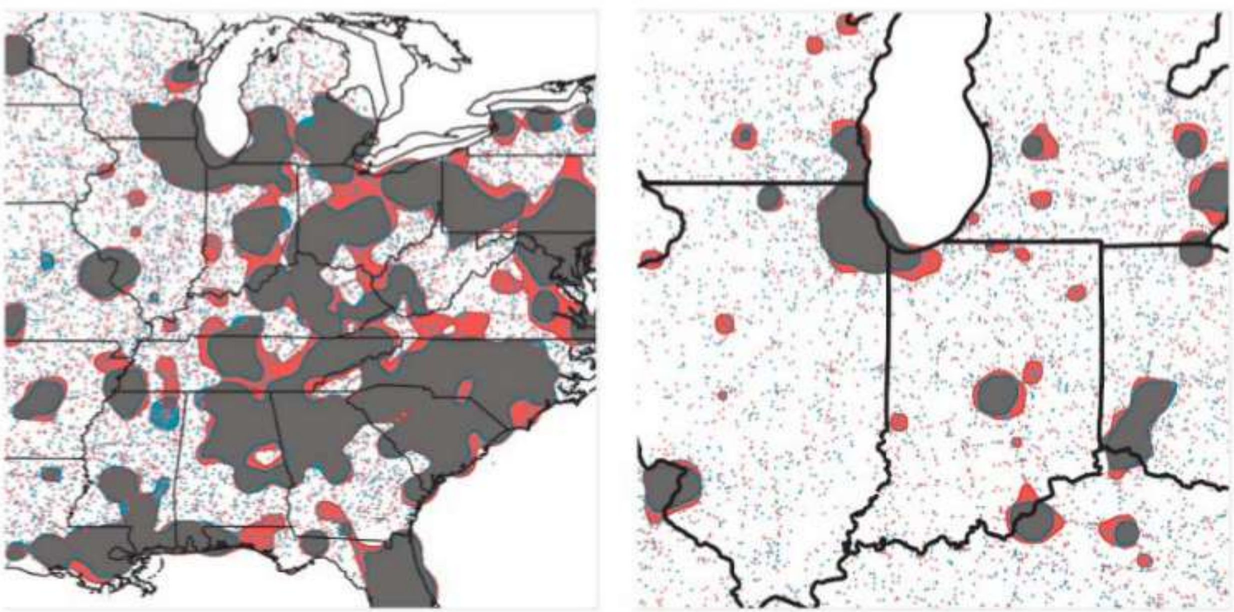
- Where dense regions overlap:
  - Use hue to encode classes
  - Perform blending in LAB color space
  - Reduce luminance and saturation to indicate overlap



Images from [Mayorga / Gleicher 2013]

Splatterplots: Subsampling

- Outside of dense regions, points are
  - subsampled to ensure a minimum distance between them
  - More points added when zooming in



Fatal car crashes in 2005 vs. 2010

Images from [Mayorga / Gleicher 2013]

Summary: Scatterplots

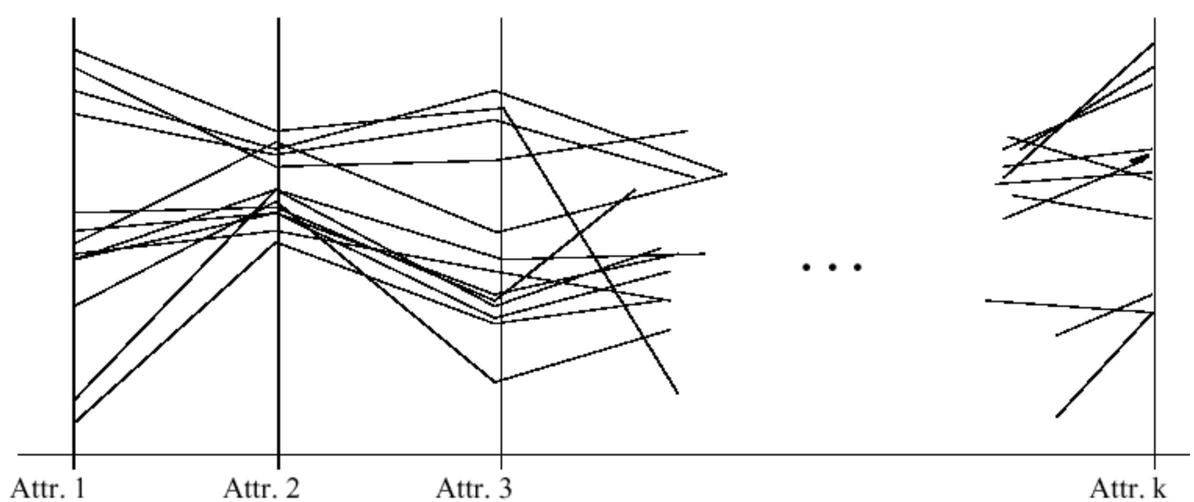
- **Scatterplots** are one of the most common techniques to visualize trends and correlations between pairs of variables
  - Main limitation: Overdraw
  - Partial solution: Splatterplots
- **Scatterplot matrices** (SPLOM) generalize this technique to multi-dimensional data
  - Special techniques for sorting, navigation, and view selection

Section 3.2: Parallel Coordinates

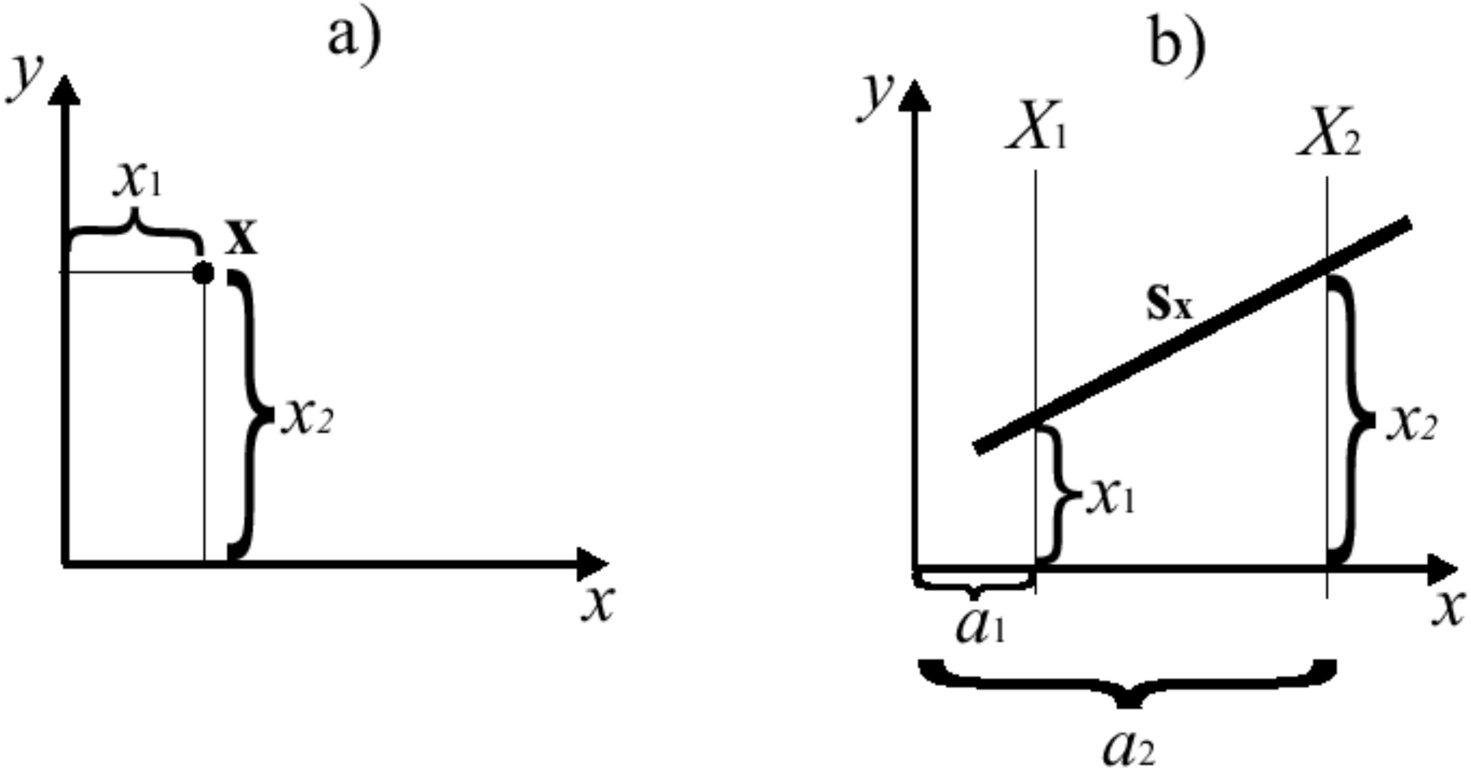
Parallel Coordinates

Parallel Coordinates can be used to visualize multi-dimensional data

- N equidistant axes which are parallel to one of the screen axes and correspond to the attributes
- The axes are scaled to the [min,max] - range of the corresponding attribute
- Every data item corresponds to a polygonal line which intersects each of the axes at the point which corresponds to the value for the attribute



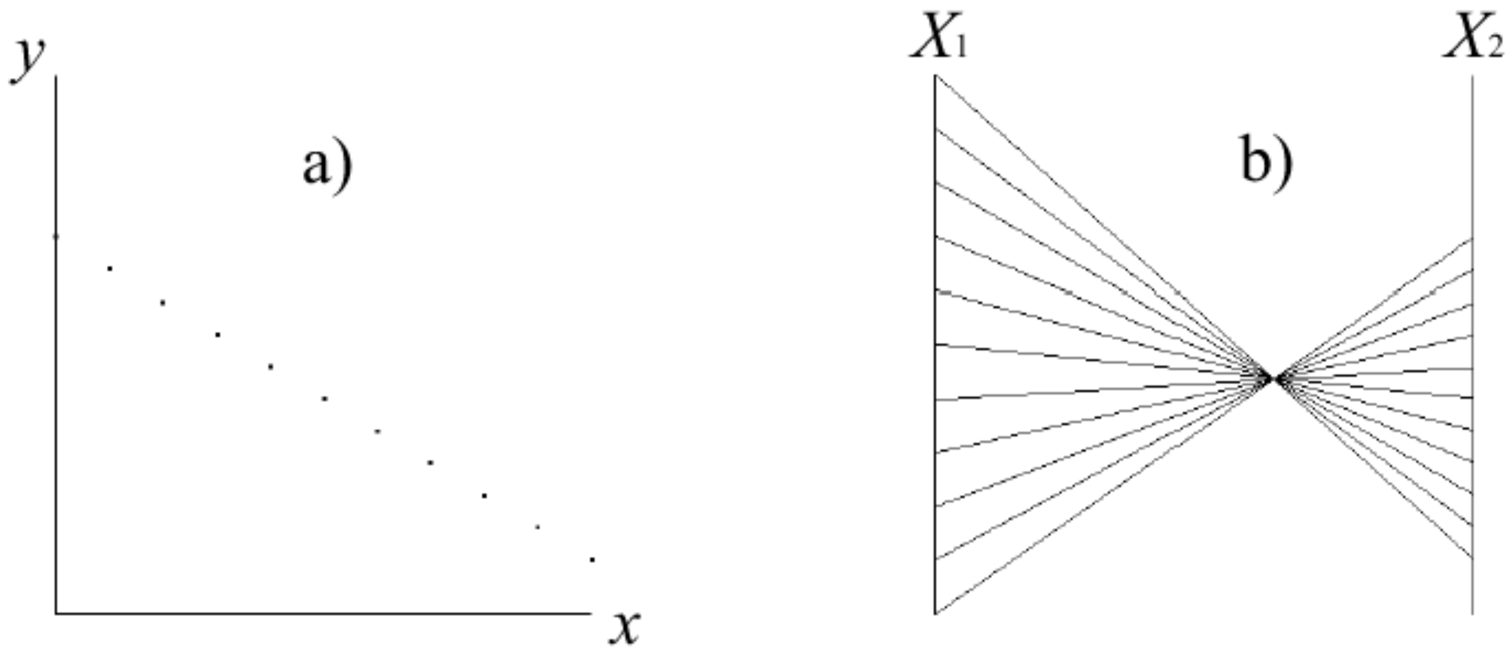
Point-line Duality for Parallel Coordinates



The point **x** in a) represented by the line $\mathbf{s}_\mathbf{x}$ in parallel coordinates in b).

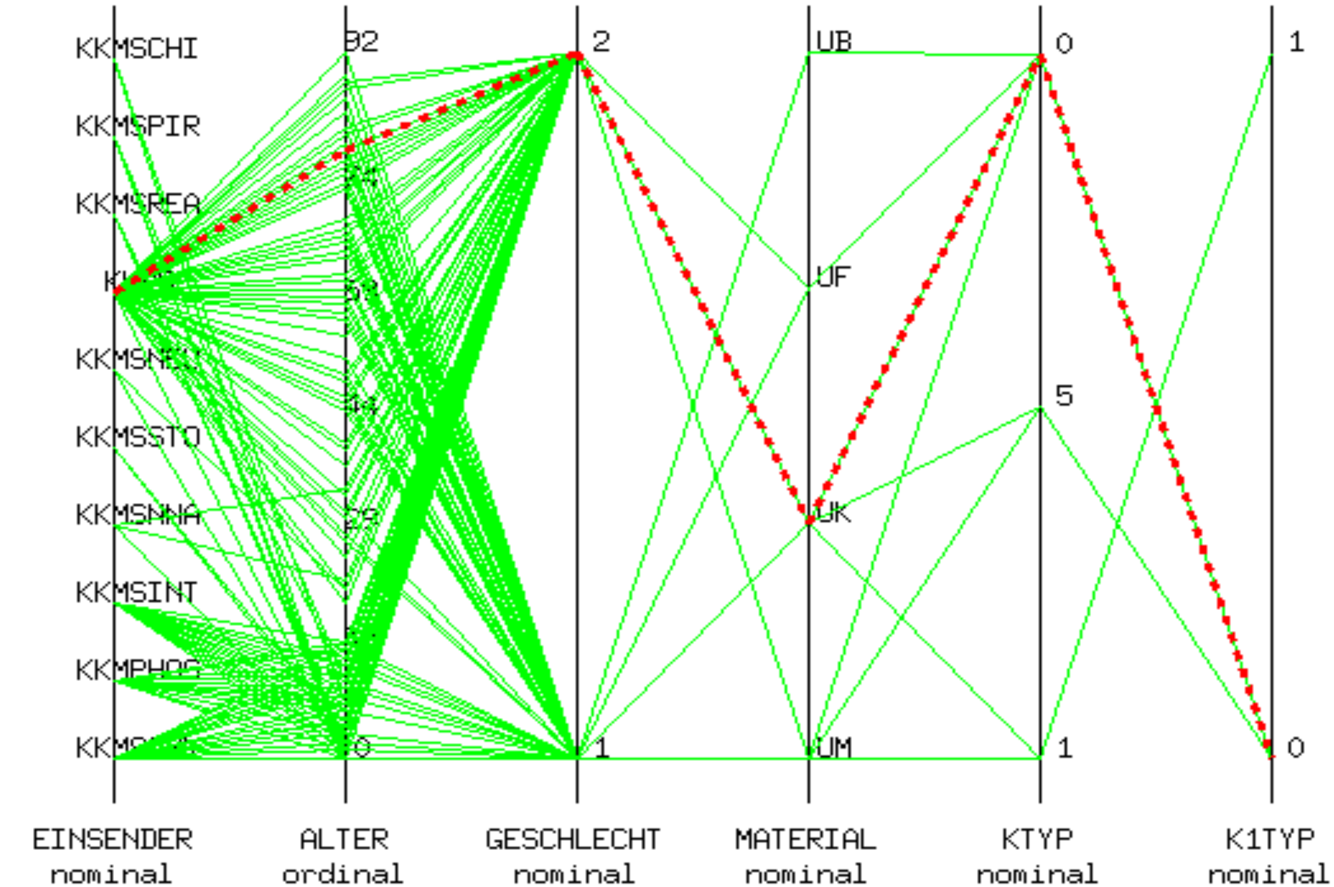
Point-line Duality for Parallel Coordinates

– A straight line in the Cartesian coordinate system (a) amounts to a single intersection point of the lines in Parallel Coordinates (b).
  - The intersection is not necessarily between X_1 and X_2
  - Parallel lines for a perfect positive correlation
  - Large number of intersection points indicates uncorrelated axes

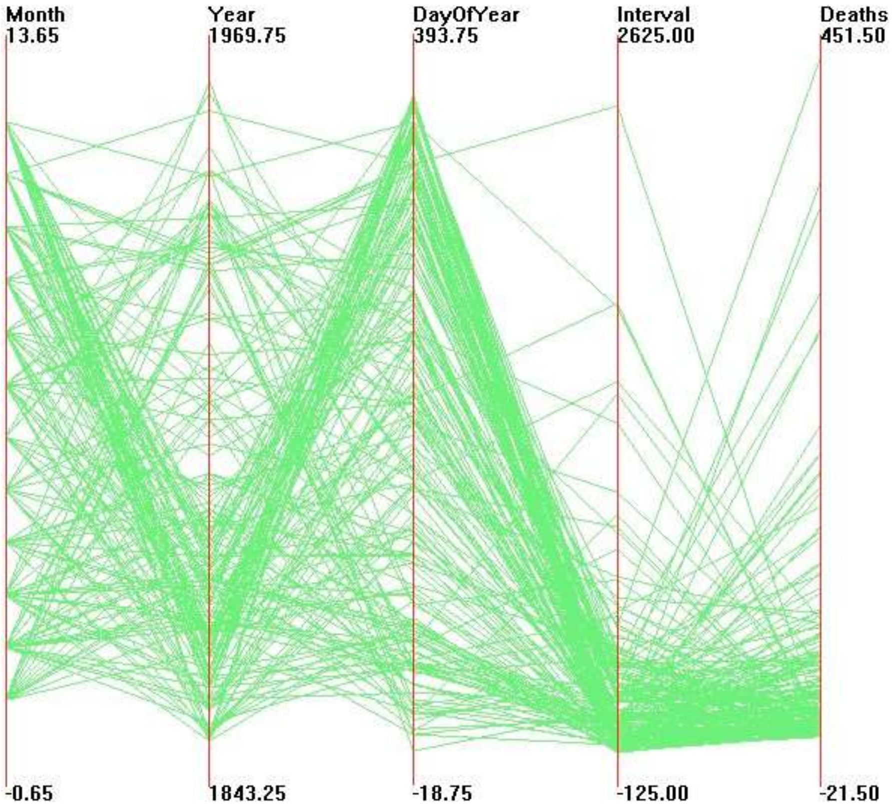


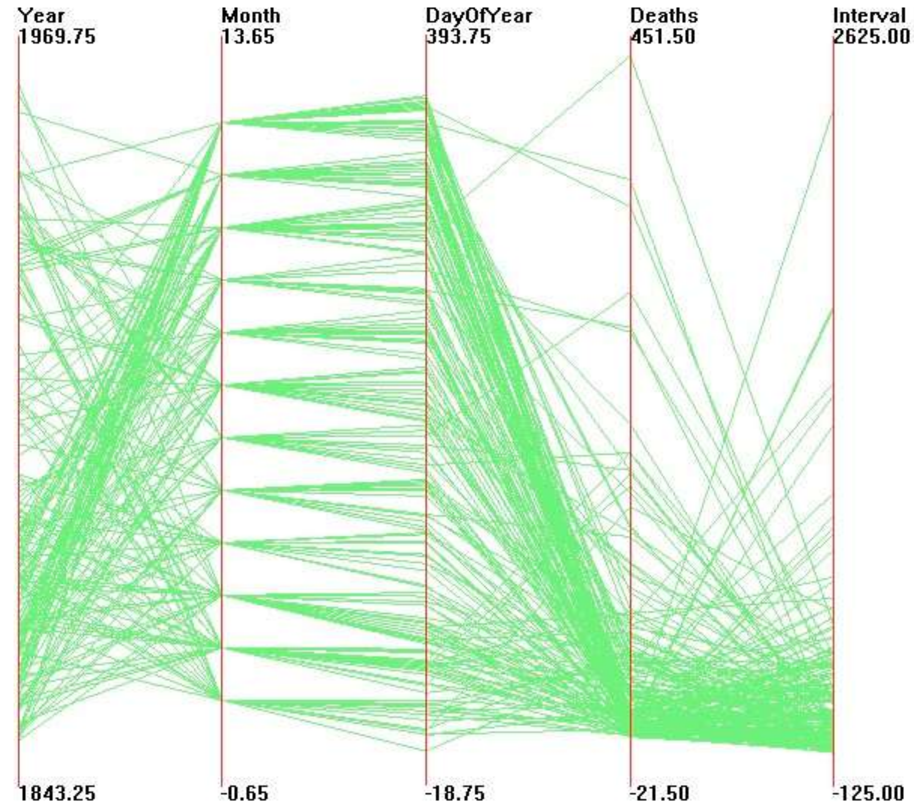
Example: Parallel Coordinates

- Visualization of a microbiological dataset:



Axis Order Matters





vs.

Proposing Suitable Axis Orders

- [Ankerst et al. 1998]: Place similar dimensions next to each other
- Write dimension *k* as a vector $\mathbf{a}^k$ (length: number *n* of data points)
- Euclidean distance:

$$D_e\left(\mathbf{a}^k, \mathbf{a}^l\right) = \sqrt{\sum_{i=1}^{n} \left(a_i^k - a_i^l\right)^2}$$

- Translation-invariant distance ($\bar{a}^k$ = mean of $\mathbf{a}^k$):

$$D_t\left(\mathbf{a}^k, \mathbf{a}^l\right) = \sqrt{\sum_{i=1}^{n} \left((a_i^k - \bar{a}^k) - (a_i^l - \bar{a}^l)\right)^2}$$

- Scale-invariant distance $D_s\left(\mathbf{a}^k, \mathbf{a}^l\right) = D_e\left(\mathbf{t}\left(\mathbf{a}^k\right), \mathbf{t}\left(\mathbf{a}^l\right)\right)$

$$[\mathrm{t}(\mathbf{a})]_i = \frac{a_i - \min(\mathbf{a})}{\max(\mathbf{a}) - \min(\mathbf{a})}$$

Measuring Partial Similarity

- When data is from different points in time,
  - find the longest time during which two attributes are more similar than ϵ:

$$S\left(\mathbf{a}^k, \mathbf{a}^l\right) = \max_{i,j}\left\{(j-i)\,\middle|\,(1 \leq i < j \leq n) \wedge D_s\left(\mathbf{a}^k_{[i,j]}, \mathbf{a}^l_{[i,j]}\right) < \epsilon\right\}$$

  - $\mathbf{a}_{[i,j]}$ = vector $\mathbf{a}$, restricted to coefficients $[i, j]$

- If we would like to permit temporal shift:

$$S_{\mathrm{async}}\left(\mathbf{a}^k, \mathbf{a}^l\right) = \max_{i,x,y}\{i |\, D_s\left(\mathbf{a}^k_{[x,x+i]}, \mathbf{a}^l_{[y,y+i]}\right) < \epsilon\}$$

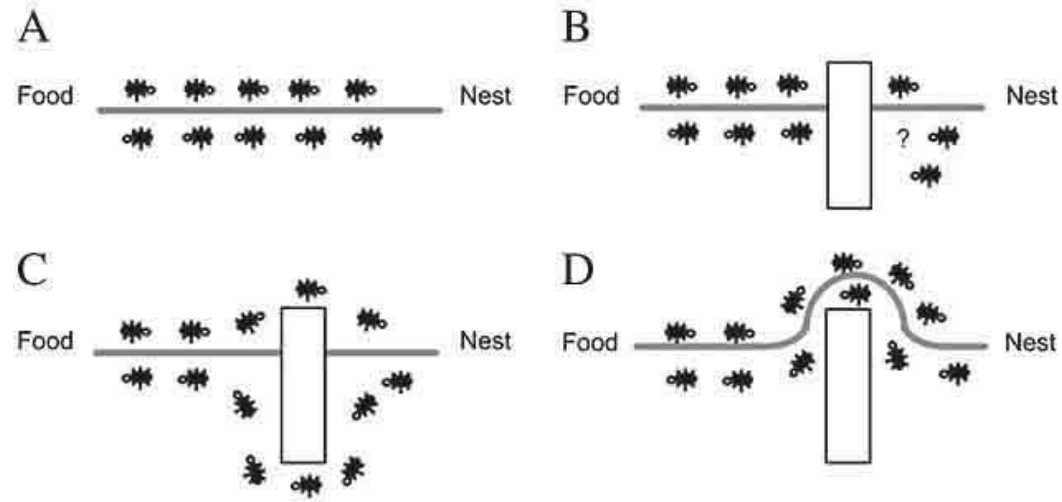
Computational Effort of Similarity Sorting

- Computing partial similarities can be expensive
  - Synchronized: $O(n^2)$ distances
  - Asynchronous: $O(n^3)$ distances
- Even given the distance matrix, finding the order which minimizes the sum of dissimilarities between neighbors is still equivalent to the traveling salesman problem
  - NP-complete, requires heuristic solution

Ant Colony Optimization

Image taken from Samer El-Khatib

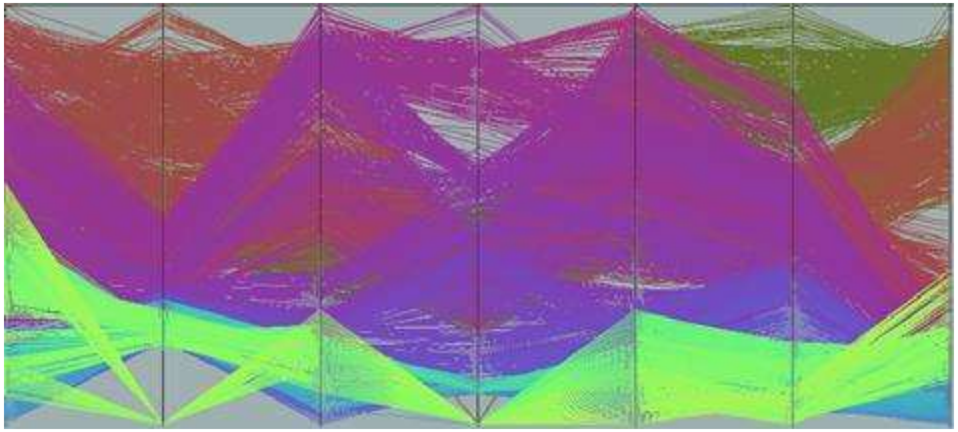
- Inspiration: Natural behavior of ants
  - Communication via pheromone trails



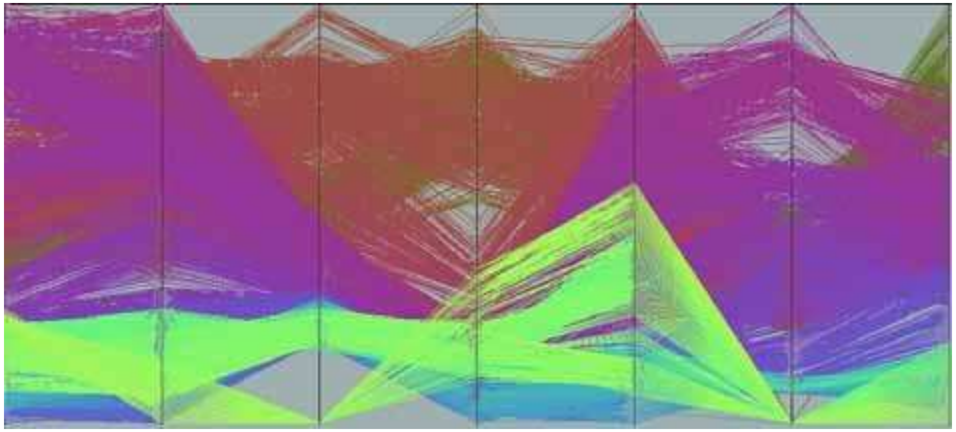
- Algorithm in [Ankerst et al. 1998]:
  - Seed *m* ants randomly, have each of them visit each dimension exactly once
  - Next dimension selected probabilistically, depending on distance and markers (initially zero)
  - At the end of each round, ant with shortest path leaves markers along its path
    - Magnitude inversely related to length of path

Similarity Sorting: Result

Parallel Coordinates using order given by the data



Parallel Coordinates using similarity sorting

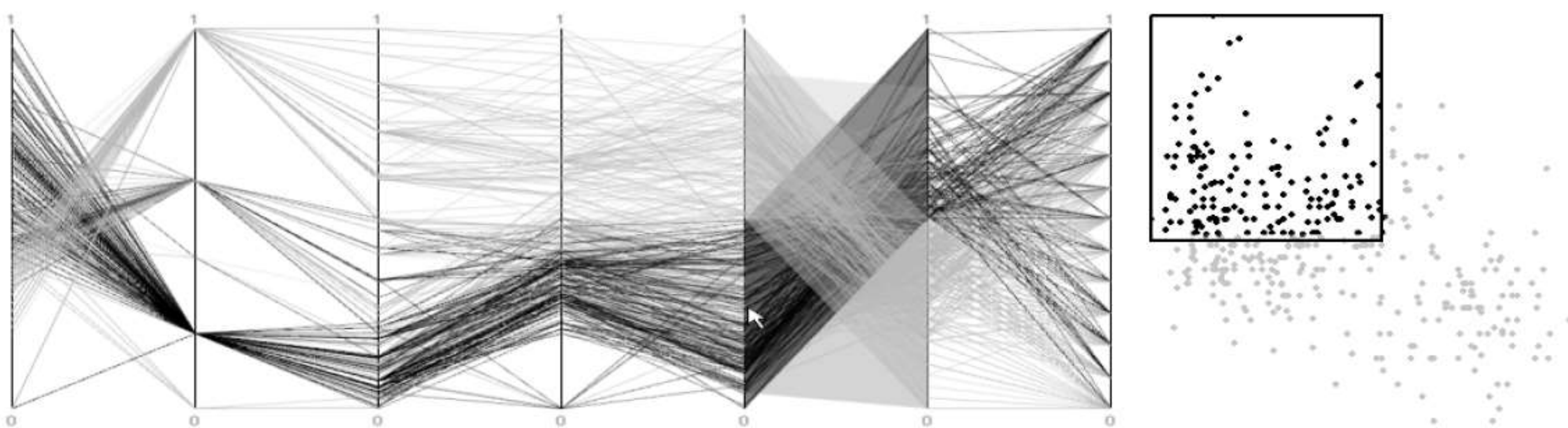


Brushing and Linking

Interaction technique:

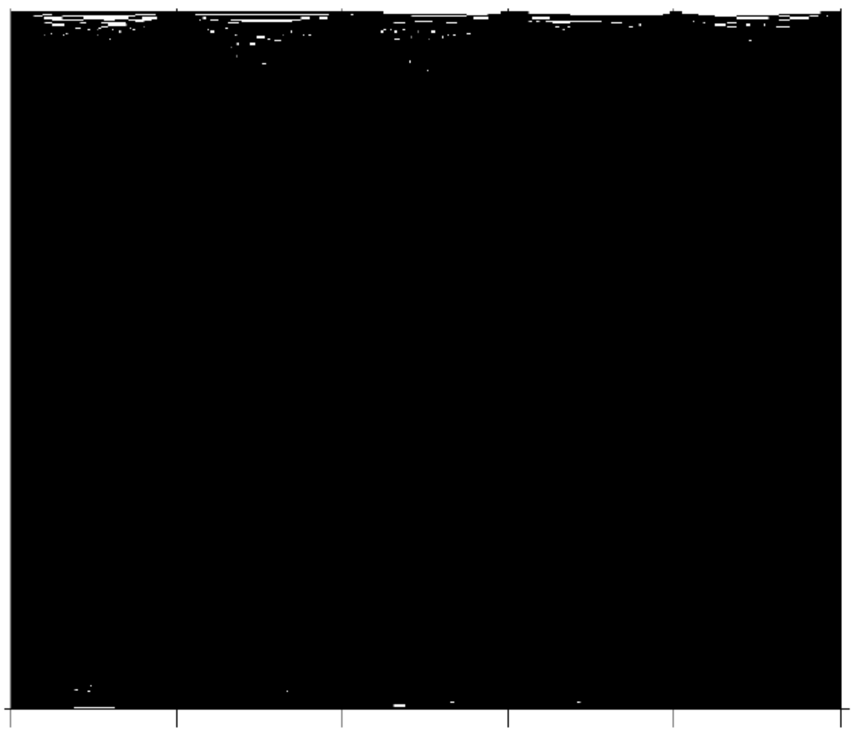
– "Brushing" = highlighting part of the data by selecting it with the mouse

– "Linking" = also highlighting the same data in another view

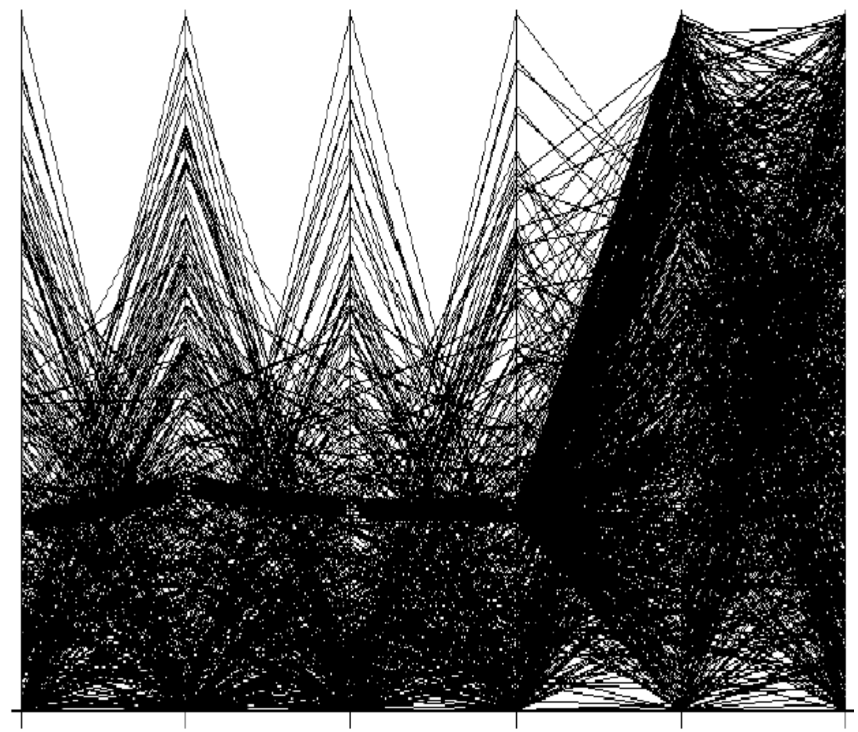


Subsampling the Data

Including all data can cause visual clutter:



15.000 data items with noise

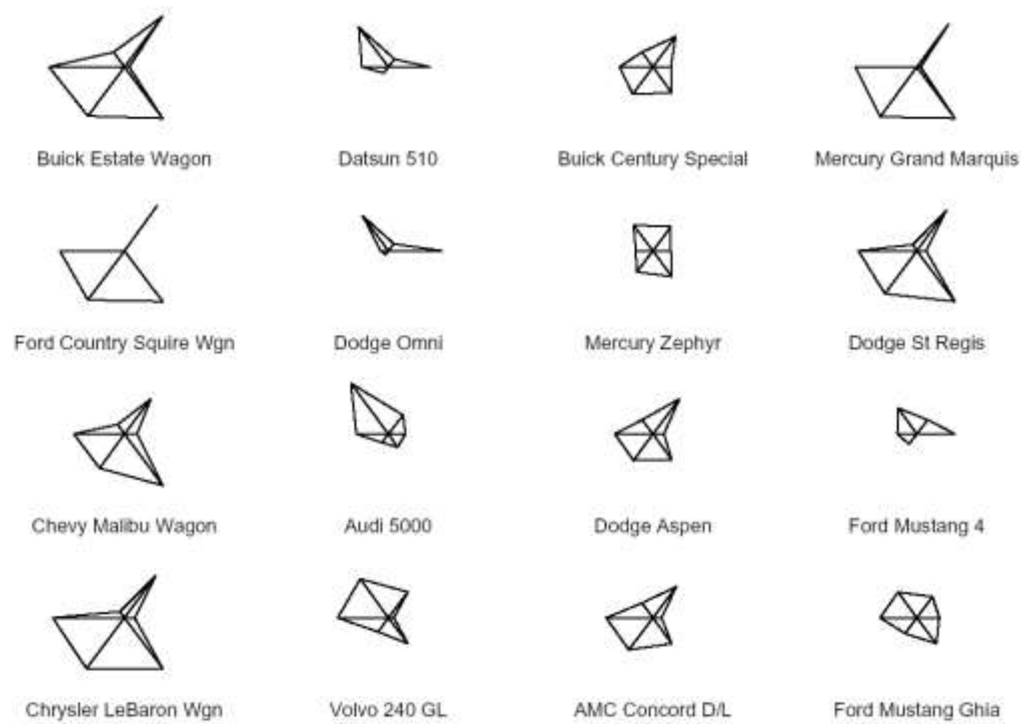


5% of the data (750 data items)

Star Glyphs

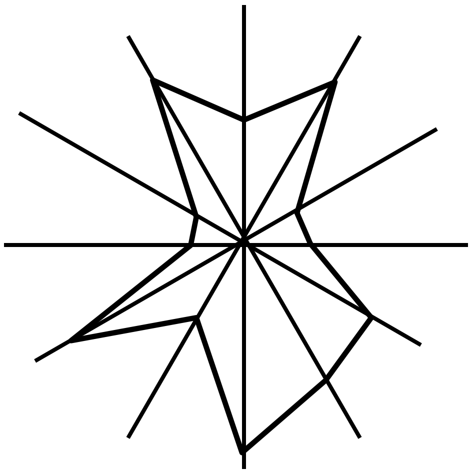
Proposed by [Fienberg 1979]

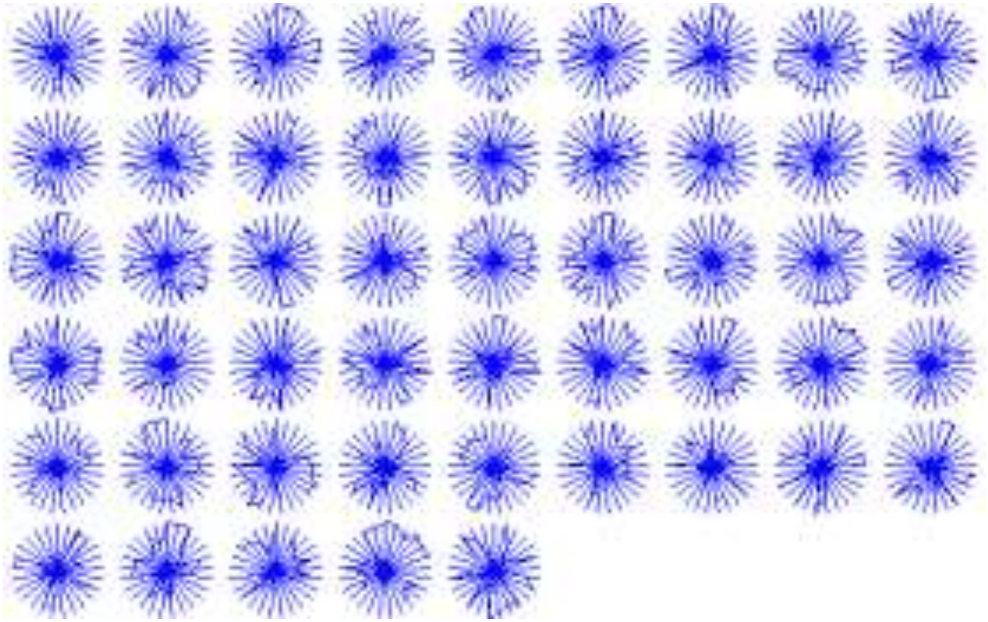
- Equally spaced radii with a common origin
- The length of each spike is proportional to the value of the respective attribute
- The ends are connected by a line



Sun Ray Plots

- Similar to star glyphs/plots
- Draw full axis, line indicates the value along each axis





Summary: Parallel Coordinates

- **Parallel coordinates** are a standard method for multi-dimensional data visualization
  - Provide overview over all dimensions
  - Correlation seen as parallel lines or single intersection
  - Easy mechanism for brushing / selection
  - Interaction and axis order are crucial!
    - Methods for judging similarity between dimensions
- *Variants:* Star glyph / sun ray plot

Section 3.3:
Other Techniques

Star Coordinates

- Idea in [Kandogan 2001]: Arrange coordinate axes as a star in 2D
  - Unlike in star glyphs, represent each data point as a point on the plane
  - Point given by vector sum of the scaled coordinate axes
  - Reveals cluster structure and allows for interactive exploration

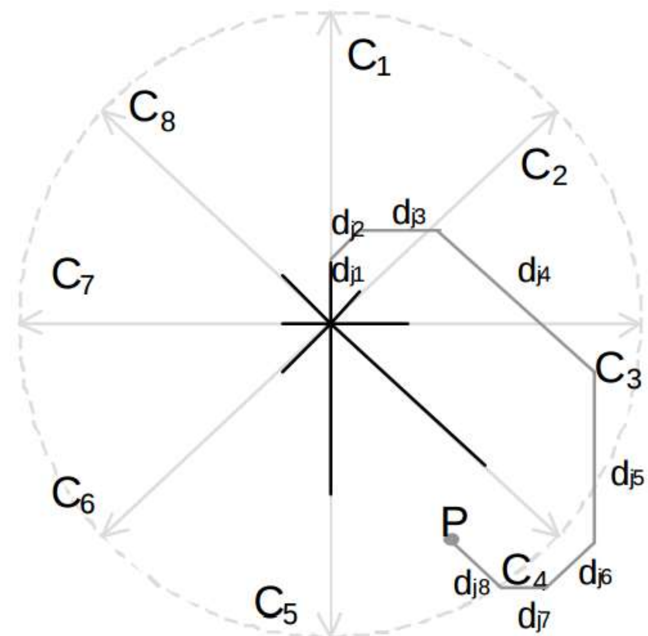
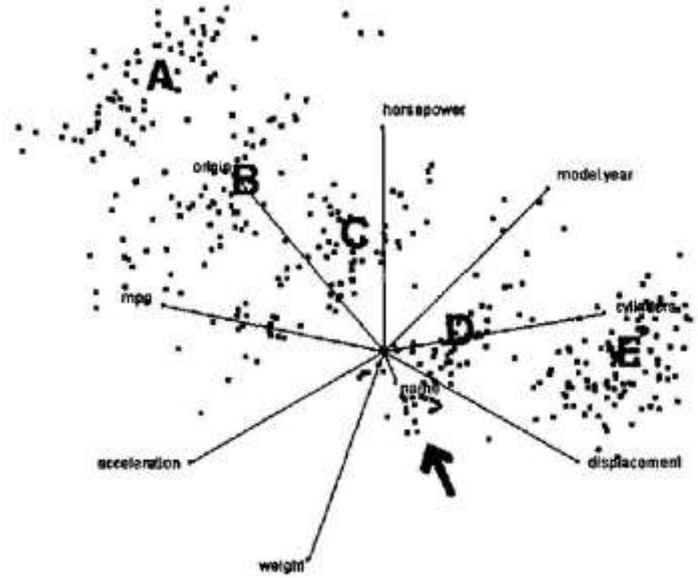
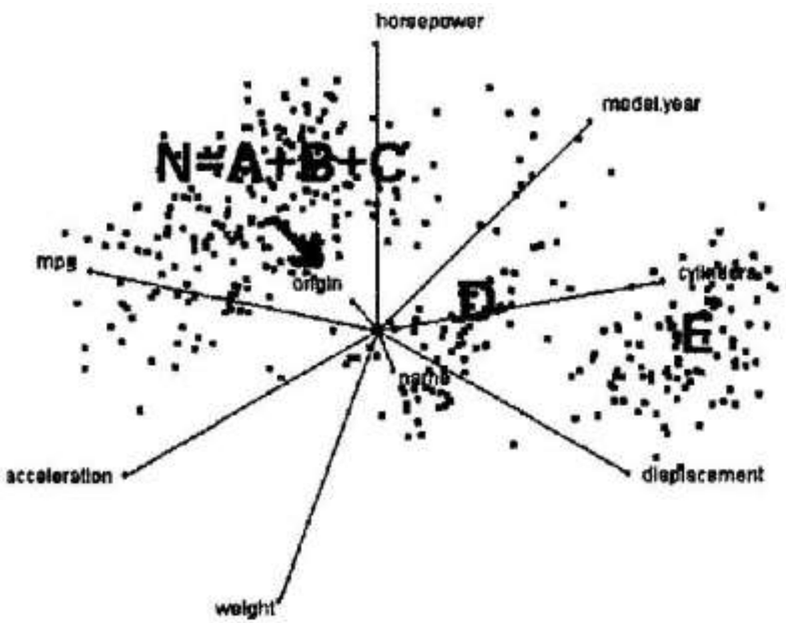


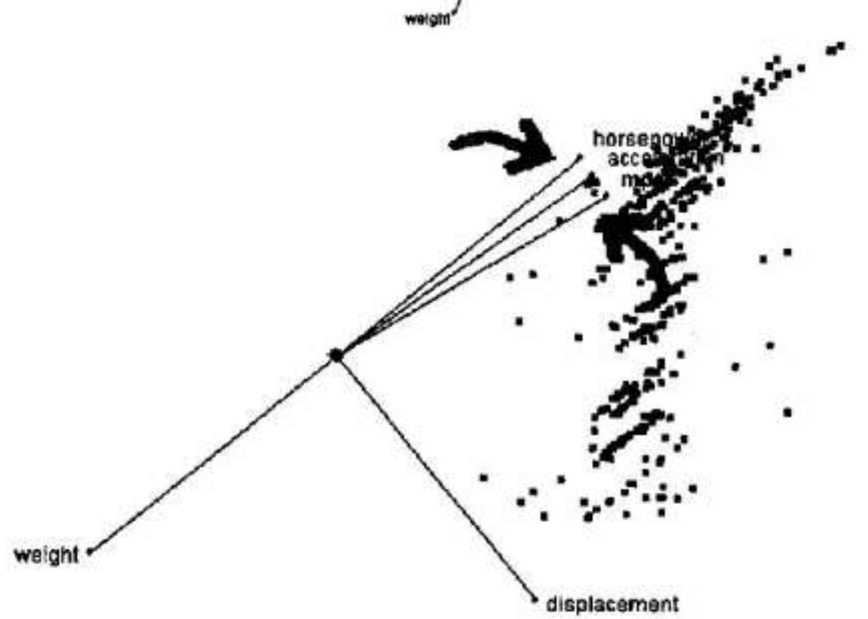
Image from [Kandogan 2001]

Star Coordinates: Interaction

- Projection to 2D results in ambiguities that are resolved by interaction
  - Axis scaling, rotation
  - Brush points or axis ranges







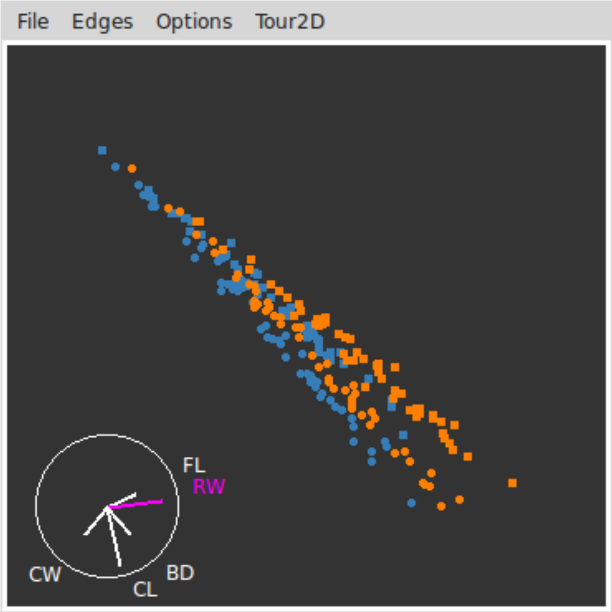
Images from [Kandogan 2001]

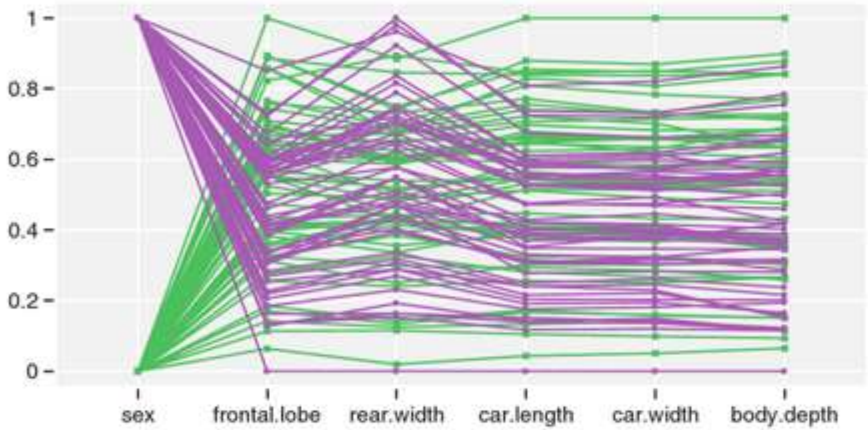
Grand Tours

- Idea in [Asimov 1985]: Once we have seen all possible 2D projections of a multi-dimensional space, we have seen the full space
- Create a sequence of projections (i.e., planes through the origin) that is…
  - *continuous* to allow for visual tracking
  - *dense* in the space of all projections
  - becoming dense *rapidly*
  - *uniform* (i.e., free from bias)
- Possible implementation: Interpolate between random projections

Grand Tour: Example Video

- From tutorial of open source software GGobi (ggobi.org)
- Dataset shows measurements of Australian crab, two species and two sexes each

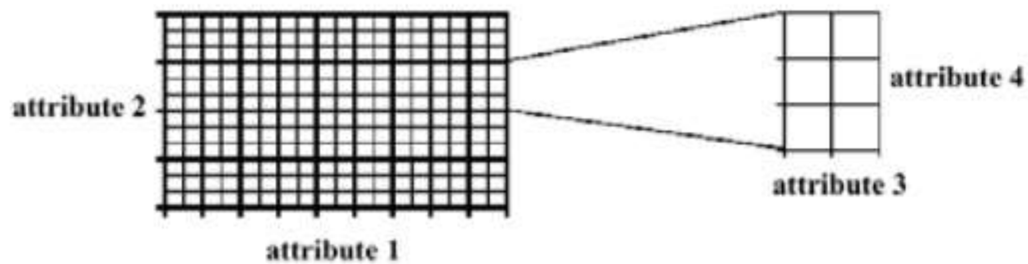




Images from Australian Museum / Cook and Swayne 2007

Dimensional Stacking

- Idea in [LeBlanc et al. 1990]: Recursively embed 2D histograms within each other
  - Bins of outer 2D histogram contain another 2D histogram, whose bins might contain yet another one…



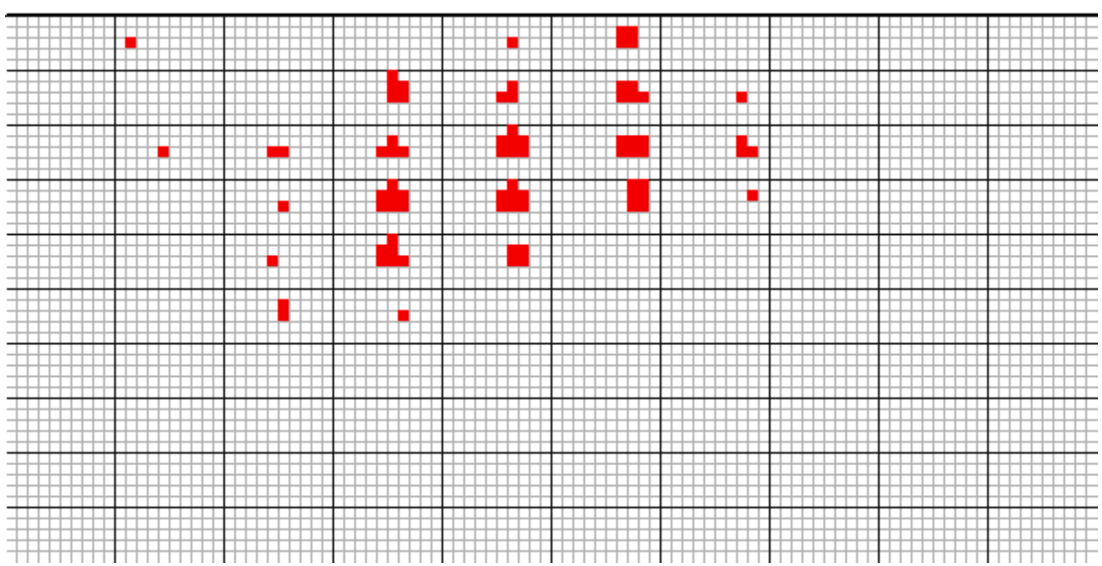
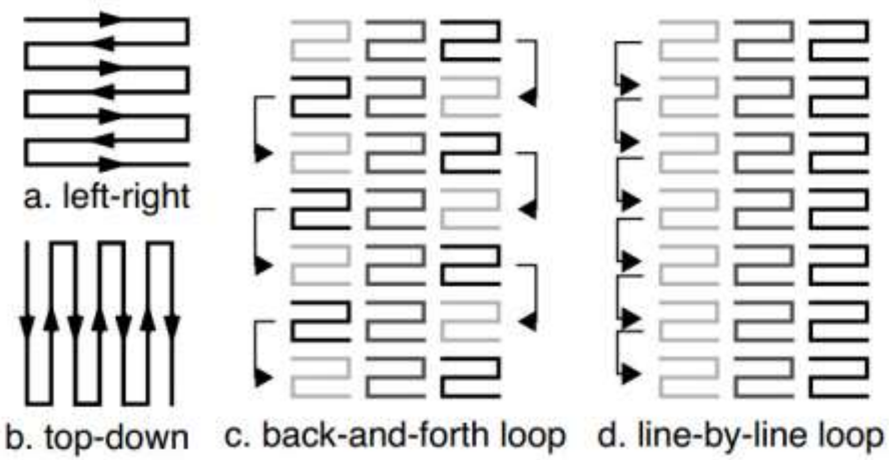
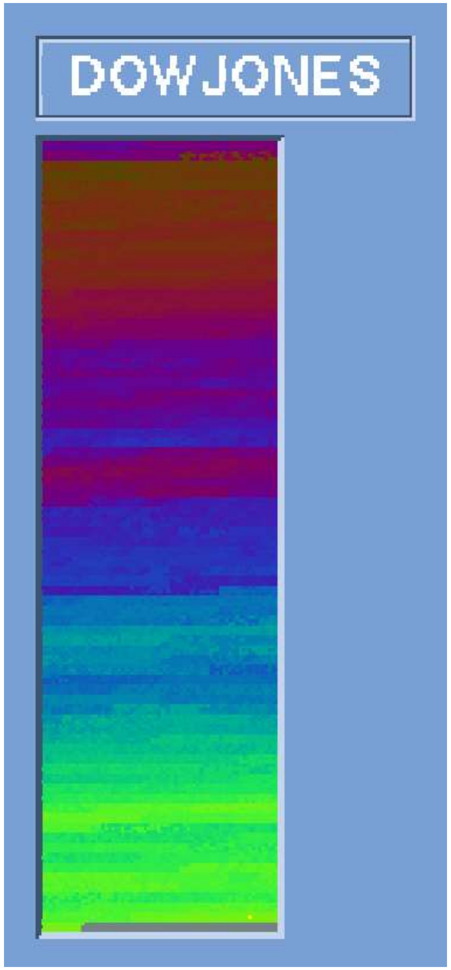


Image from XmdvTool homepage

Pixel-Oriented Techniques

- Idea in [Keim et al. 1995]: Map each data value to the color of a single pixel
  - Shows as much detail as possible
  - Lay out pixels in a recursive fashion to highlight patterns

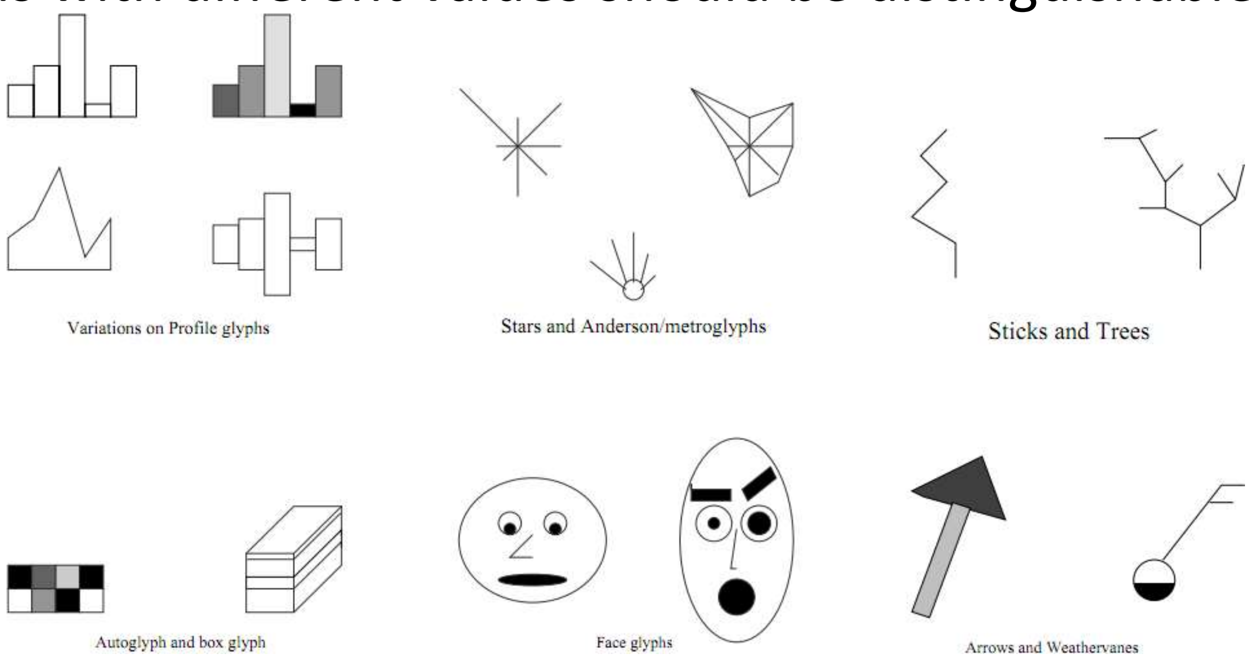




Level 1: (3x3)
Level 2: (1x24)
Level 3: (80x1)

Glyphs and Icons

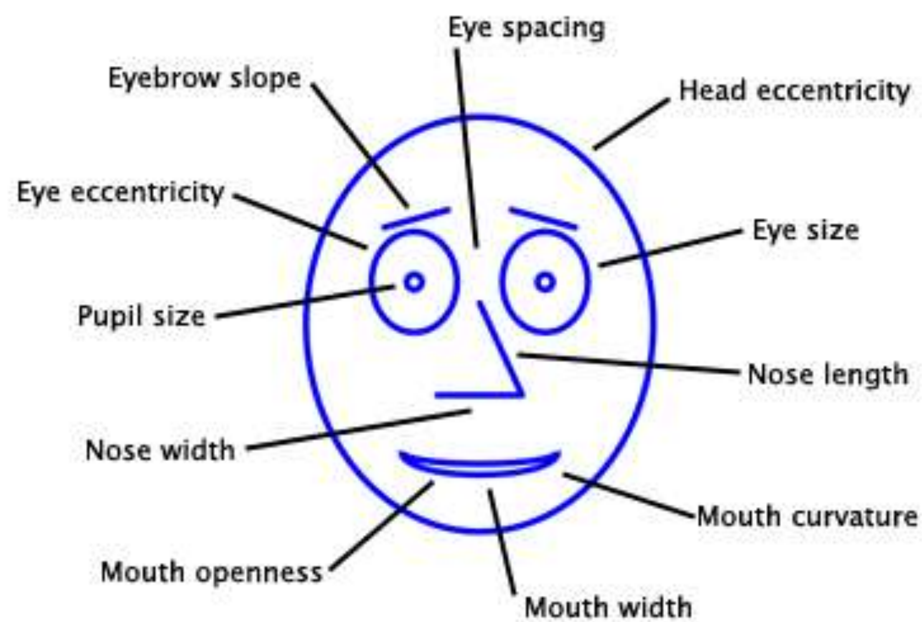
- Primitives that can be positioned exactly and represent variables by
  - geometric characteristics like length, angle or shape
  - other attributes like color and transparency
- Design rules
  - Features should be easy to distinguish and combine
  - Icons with different values should be distinguishable



Chernoff Faces

Proposed by [Chernoff 1973]

- Relies on human ability to distinguish small features in faces
- Similar to smileys: ☺ happy, ☹ sad, 😐 neutral, ;) wink, :] grin, :D laugh, etc.
- Each facial feature represents one variable



Face Morphing

- Variant proposed by [Alexa 98]

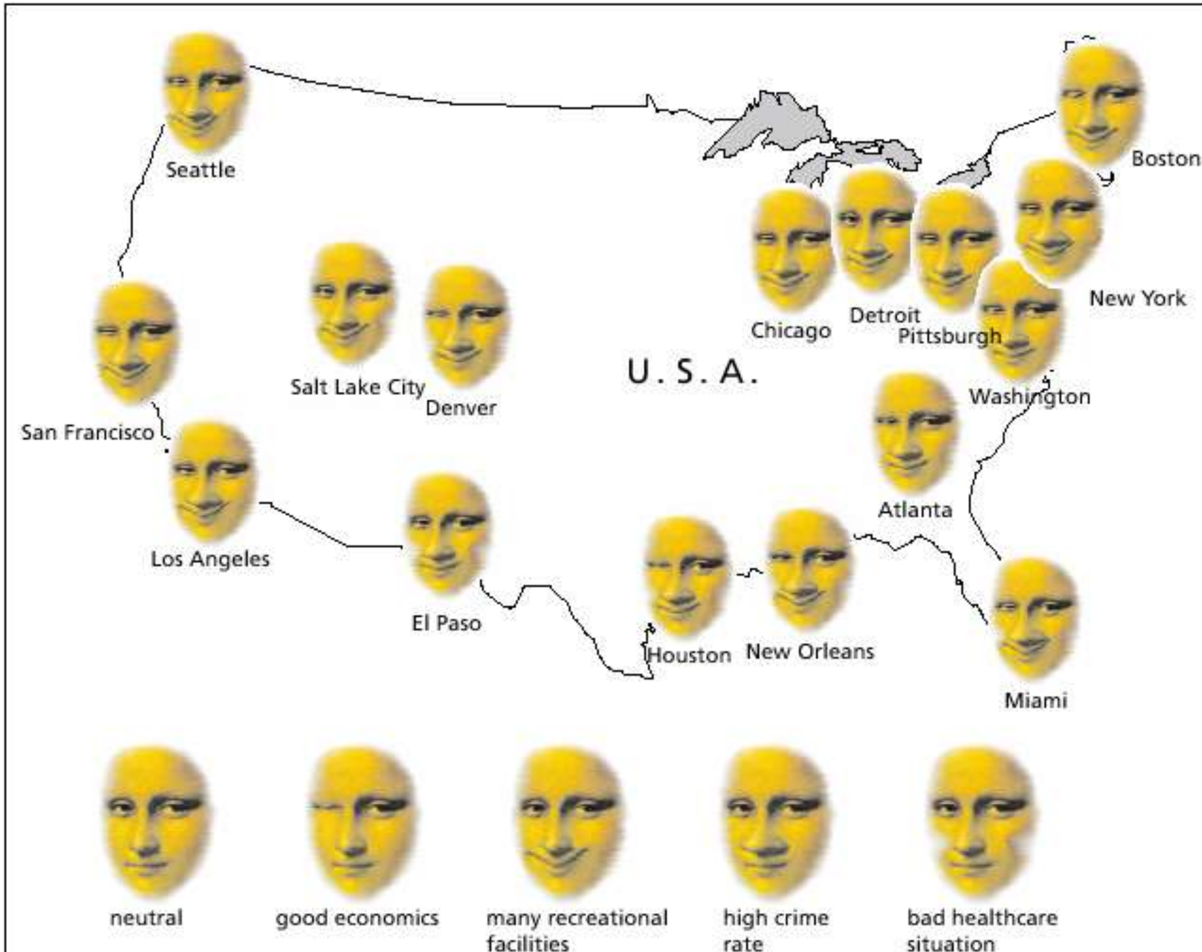


Figure 1: Distorted Mona Lisa faces as visual representations of city ratings in the U.S.

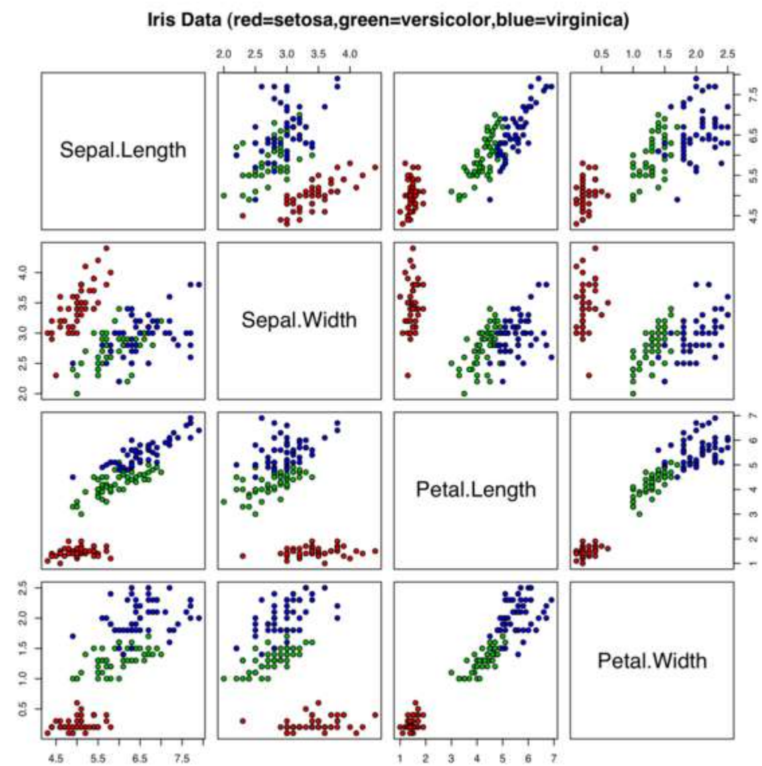
Summary

- Other alternatives for encoding multi-dimensional data are
  - Star coordinates (with interaction!)
  - Grand Tours
  - Dimensional Stacking
  - Glyphs

Section 3.4: Clustering

Clustering: Introduction

- **Goal of clustering:** Find meaningful groups in the data
  - Essentially without knowing anything about it ("unsupervised")
  - Based on similarity alone: Group similar data together, keep dissimilar data apart
  - Used, e.g., in view selection
- You should already know about two widely used methods:
  1. Hierarchical Agglomerative Clustering
  2. *K*-means



setosa

versicolor

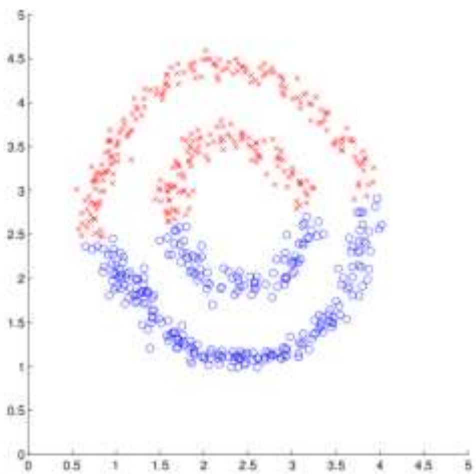
virginica

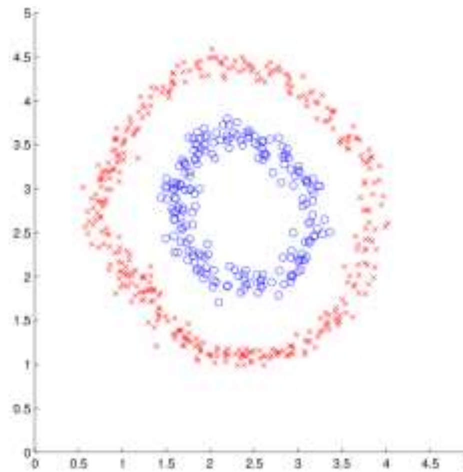
Iris Data (Fisher/Anderson, 1936)

Spectral Clustering: Motivation

- *k*-means assumes spherical clusters
- Agglomerative clustering assumes well-separated clusters

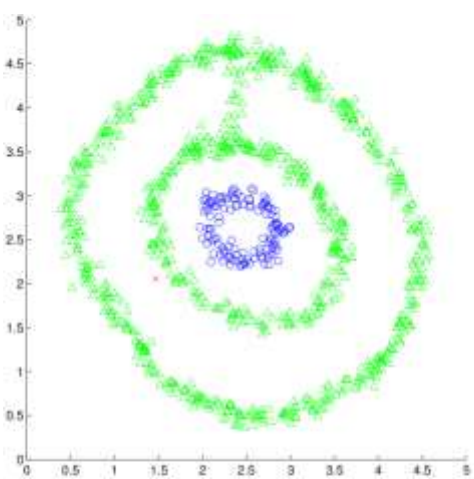
k-means

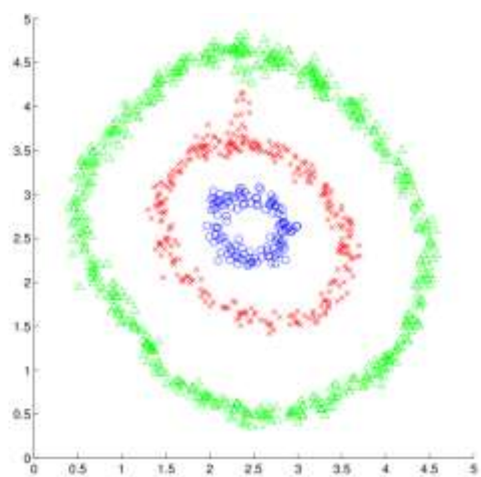




spectral

hierarchical agglomerative





spectral

[Ng et al. 2001]

Similarity Graphs

Undirected weighted **similarity graph** (V,E): Each data point is a vertex V, edge weights measure the similarity between points.

- **ε-graph:** Connect to all data points within distance ε
- ***k*NN:** Connect each data point to its *k* nearest neighbors
- **Mutual *k*NN:** Connect pairs of points for which both are among each other's *k* nearest neighbors

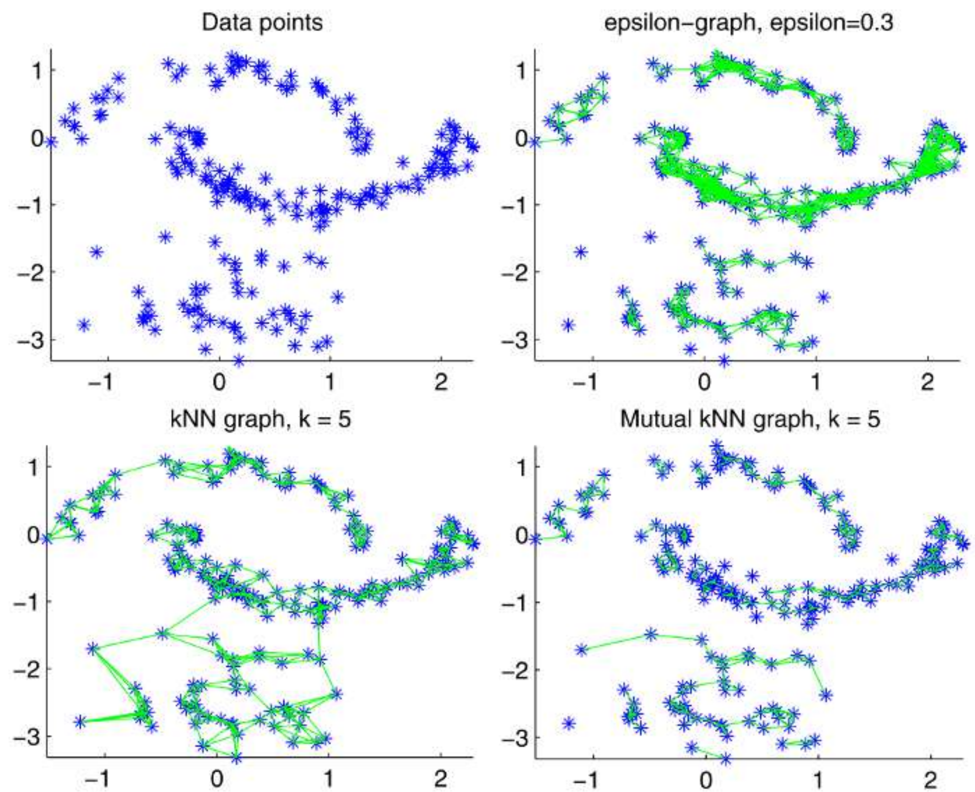
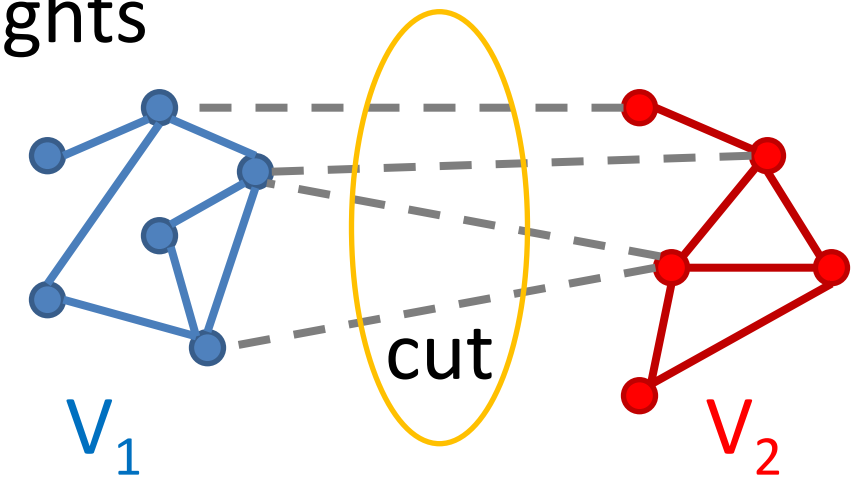


Image from [von Luxburg 2007]

Graph Cuts

When partitioning a graph (V,E) into two vertex classes V_1 and V_2, the corresponding "cut" consists of the edges between V_1 and V_2

- cut(V_1,V_2) denotes the corresponding sum of edge weights



- Simply trying to minimize the cut usually splits off individual vertices, or very small groups

RatioCut and NCut

- **Ratio cut** avoids splitting off small clusters by normalizing by the number of nodes:

$$\text{RatioCut} = \frac{\text{cut}(V_1, V_2)}{|V_1|} + \frac{\text{cut}(V_1, V_2)}{|V_2|}$$

- **Normalized cut** looks for partition such that
  - Similarities between classes are minimized
  - Similarities within classes are maximized

$$\text{NCut} = \frac{\text{cut}(V_1, V_2)}{\text{vol}(V_1)} + \frac{\text{cut}(V_1, V_2)}{\text{vol}(V_2)}$$

  - Uses vol(V_i) (sum of edge weights adjacent to V_i) rather than number of nodes

Finding exact optima of both criteria is NP-hard.

Graph Laplacian

Define:

- n=|V| (number of data points)
- Affinity matrix **W** (symmetric, nxn)
  - w_{ij} contains weight of edge between i and j
- Degree matrix **D** (diagonal, nxn)
  - sum of adjacent edge weights: $d_i = \sum_j w_{ij}$
- Graph Laplacian:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \quad \text{and} \quad \tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

(unnormalized) (normalized)

Indicator Vector

- Define **indicator vector** $f \in R^n$ of set $A \subset V$ such that $f_i = \sqrt{|\bar{A}|/|A|}$ if $v_i \in A$ and $f_i = -\sqrt{|A|/|\bar{A}|}$ if $v_i \in \bar{A}$
- f is orthogonal to the constant one vector **1**

$$\sum_{i=1}^{n} f_i = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}}$$

$$= |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0$$

- The Euclidean norm of f is $\|f\| = \sqrt{n}$

$$\sum_{i=1}^{n} f_i^2 = |A| \frac{|\bar{A}|}{|A|} + |\bar{A}| \frac{|A|}{|\bar{A}|} = n$$

Quadratic Form of Graph Laplacian

Lemma: The quadratic form associated with the graph Laplacian is

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Proof:

$$f^T L f = f^T D f - f^T W f = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right)$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

RatioCut in Terms of Graph Laplacian

- Given indicator vector f,

$$f^{\mathrm{T}}Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(f_i - f_j\right)^2$$

$$= \frac{1}{2}\sum_{i\in A, j\in \bar{A}} w_{ij}\,(\sqrt{|\bar{A}|/|A|} + \sqrt{|A|/|\bar{A}|})^2$$

$$+ \frac{1}{2}\sum_{i\in \bar{A}, j\in A} w_{ij}(-\sqrt{|\bar{A}|/|A|} - \sqrt{|A|/|\bar{A}|})^2$$

$$= \mathrm{cut}(A,\bar{A})\left(\frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2\right)$$

$$= \mathrm{cut}(A,\bar{A})\left(\frac{|A| + |\bar{A}|}{|A|} + \frac{|A| + |\bar{A}|}{|\bar{A}|}\right)$$

$$= |V| \cdot \mathrm{RatioCut}(A,\bar{A})$$

RatioCut Re-Written

- Our results allow us to **re-write RatioCut** as a minimization of $f^{\mathrm{T}}Lf$
  - minimization over $A \subset V$ with f defined as above
- This equivalent reformulation with discrete f is still NP-hard, but it suggests a **relaxation**
  - permit continuous f
    - maintain constraints $f \perp \mathbf{1}$, $\|f\| = \sqrt{n}$
  - We can use the **Rayleigh-Ritz theorem:** Since L is symmetric, its eigenvectors are the critical points of the Rayleigh quotient:

$$R(x) = \frac{x^{\mathrm{T}}Lx}{x^{\mathrm{T}}x}$$

Properties of the Graph Laplacian

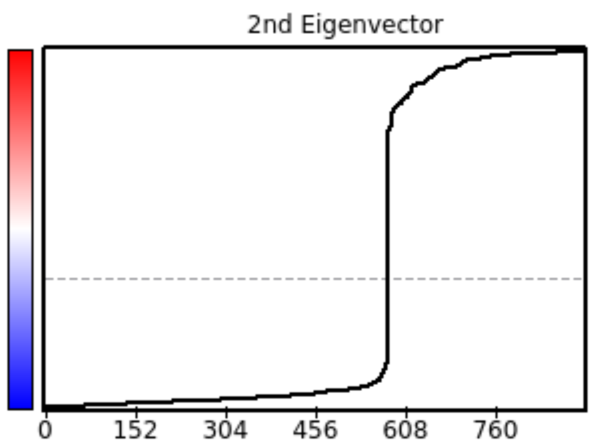
- L is symmetric and positive semi-definite
  - Symmetry: Direct consequence of definition
  - Semi-definiteness: Consequence of $f^T L f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$ and non-negativity of w_{ij}
- The constant one vector **1** is an eigenvector of L with corresponding eigenvalue 0
  - Direct consequence of definition L=D-W

Approximate RatioCut

- As a consequence of the observations above, the desired minimum of $f^{\mathrm{T}}Lf$ is attained when f is the eigenvector of L corresponding to its **second smallest eigenvalue**
  - "Fiedler vector", fuzzy cluster indicator
- A hard clustering is achieved by **discretizing the coefficients** of f, e.g., by
  - Thresholding at zero or at (max+min)/2
  - Applying a clustering method such as K means
  - Sorting the coefficients, computing RatioCut for each possible threshold, taking the optimum
- Note: **No approximation guarantees**, but works well in many real-world cases

Example: Approximate RatioCut

- *Example:* ε-graph over points sampled on fissure between lung lobes, edges weighted by similarity in surface normal



Fiedler vector (sorted)

Color coded on 3D data

Approximate Normalized Cut

- The **Normalized Cut** can be approximated in a very similar way, with the normalized Laplacian $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$ replacing its unnormalized counterpart $\mathbf{L} = \mathbf{D} - \mathbf{W}$
  – Can be re-written as generalized eigenproblem
$$\mathbf{L}f = \lambda\mathbf{D}f$$
  – See [von Luxburg 2007] for details
- **Implementation** (for both RatioCut and NCut):
  – In practice, construct sparse graph and exploit sparsity in (generalized) eigenvector computation! Methods for this are outside of our scope.

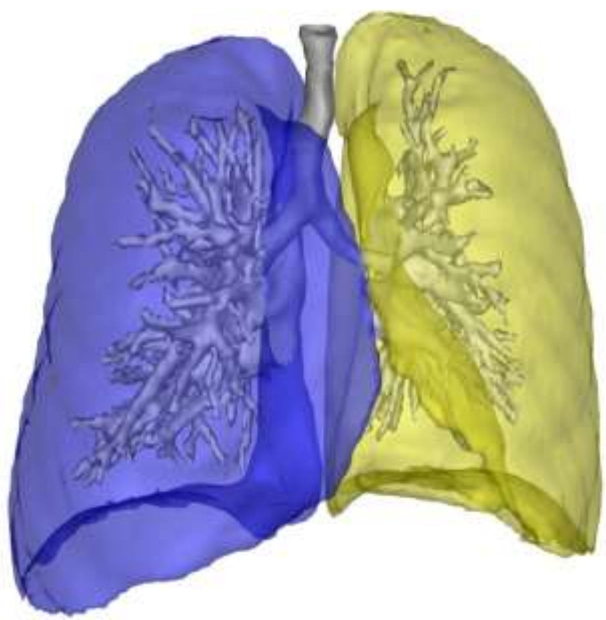
Spectral Clustering: Multi-Cluster Case

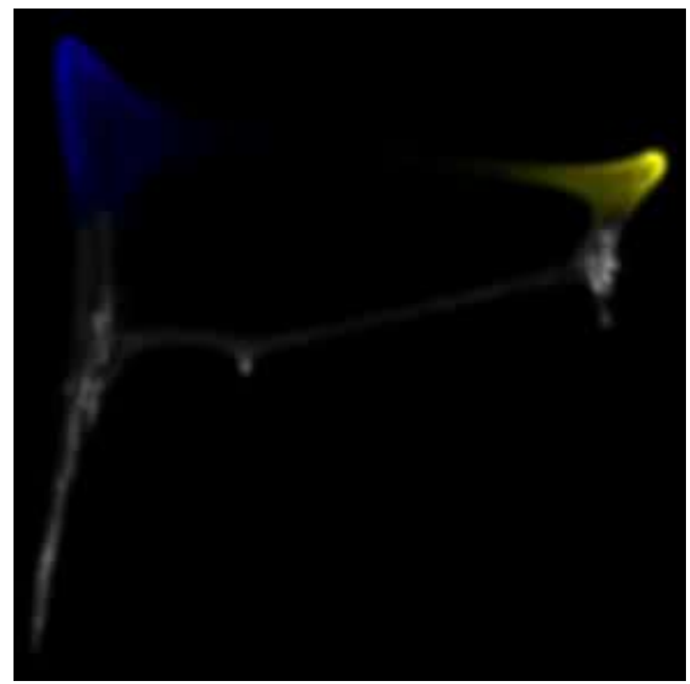
Strategies for *K*>2 clusters:

1. Recursively bisect (sub-)clusters [Shi/Malik]
2. Compute *K* smallest eigenvectors and use them to define *K*-dimensional coordinates for each point; use a traditional clustering method such as *K* means on that representation [Ng, Meila/Shi]
   - *Justification:* Approximation of multi-cluster RatioCut, see [von Luxburg 2007]

Visualizing the Spectral Embedding

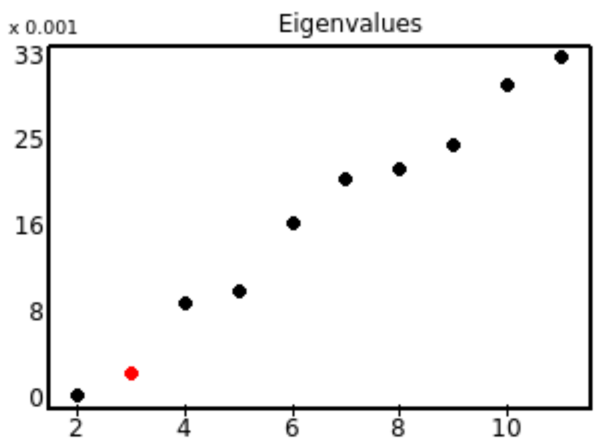
The plot on the right shows a „grand tour“ of the *K*-dimensional (here, *K*=5) feature space spanned by five eigenvectors



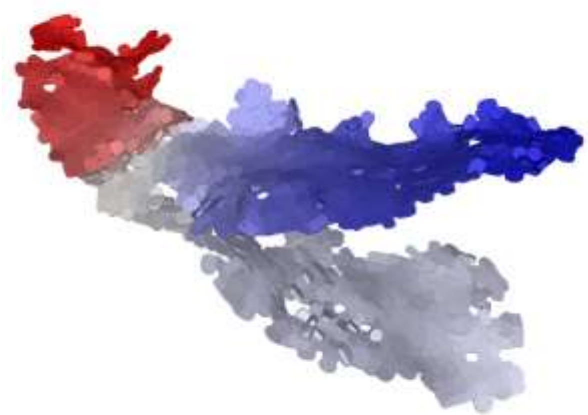


Spectral Clustering: Number of Clusters

- Spectral Gap:
  - In case of *K* connected components, null space of Laplacian is *K*-dimensional
  - A stable clustering requires that *K* near-zero eigenvalues should be followed by a "spectral gap" towards a markedly larger *K+1*st eigenvalue



Smallest Eigenvalues



3rd Eigenvector

Summary: Clustering

- Clustering attempts to find intrinsic groups in the data
- Hierarchical agglomerative clustering and k-means are widely used standard methods
- Spectral clustering can overcome their limitations by adapting to complex cluster shapes and being more robust
  - Linear algebra to solve relaxations of powerful, but NP hard clustering criteria
  - Spectral gap as an indicator of cluster number

References: Clustering

- Christopher M. Bishop: *Pattern Recognition and Machine Learning.* Springer, 2006
- Ulrike von Luxburg: *A Tutorial on Spectral Clustering.* Stat Comput 17:395-416, 2007
- Rui Xu: *Clustering.* Wiley, 2009