Chapter 3 - Multidimensional Information Visualization II

Concepts for visualizing univariate to hypervariate data

Vorlesung „Informationsvisualisierung”
Prof. Dr. Florian Alt, WS 2013/14

Konzept und Folien (4th revised edition):
Thorsten Büring, Andreas Butz, Michael Burch
Outline

• Reference model and data terminology
• Visualizing data with < 4 variables
• Visualizing multivariable data
  – Geometric transformation
  – Glyphs
  – Pixel-based
  – Dimensional Stacking
  – Downscaling of dimensions
• Case studies: support for exploring multidimensional data
  – Rank-by-feature
  – Value & relation display
  – Dust & magnet
• Clutter reduction techniques
Trivariate Data
Trivariate Data

• Tempting: map each variable to each dimension of a 3D scatterplot
• Occlusion of points with different positions
• Problem with static representation?

Occlusion of points with different positions and in general difficulty to visually map the (x,y,z)-coordinates of the single points
Trivariate Data

• Idea 1:
  – Map each variable to a dimension (axis) of a 3D scatterplot
  – Enhancement by using guiding lines

Still problematic: Many lines lead to visual clutter and even more occlusions
Trivariate Data

• Idea 2:
  – Map each variable to a dimension (axis) of a 3D scatterplot
  – Rotation is used to explore the data but if the dataset is very dense we have to also look inside it

More complex interactive features needed to analyze the trivariate data
Scatterplot Matrix

• Matrix of all pairwise scatterplot views of the data
• Easy to understand by using familiar and powerful scatterplot representation
• Can serve as a good starting point for data exploration
• Increased demand for display space
• Increased cognitive load caused by redundant data

Cleveland 1993
Trivariate Data

• Idea 3:
  – 2D scatterplot with additional encoding
  – In this case color and shape
  – Shows relationship between three variables
  – For color / shape coding: assumes categorical variable or classing of quantitative variable
    • pot. loss of information
Multivariate Data
Multivariate Data

- Geometric Transformation
- Glyph-based Visualizations
- Pixel-based Visualizations
- Dimensional Stacking
- Downscaling of Dimensions
- Clutter Reduction
Geometric Transformations
Geometric Transformations

• Idea: present projections of the multidimensional data to find interesting correlations

• Most common techniques
  – Scatterplot matrix
  – Prosection matrix
  – Parallel coordinates plot
Scatterplot Matrix

• Scatterplot matrix can be scaled to > 3 variables
• Number of scatterplots increases rapidly
• n variables means n x n plots
• Diagonal maps the same variable twice
• Each pair is plotted twice, once on each side of the diagonal
• Allows convenient sequential browsing of one variable compared to all other variables
Scatterplot Matrix
Prosection Matrix

- Scatterplot matrix with interactive linking and brushing (Tweedie & Spence 1996)
- Projection of a section of parameter space
- User selects multivariable ranges, which are colored differently
Prosection Matrix
Parallel Coordinate Plot

- Create $M$ equidistant vertical axes, each corresponding to one of the $M$ variables.
- Each axis scaled to $[\text{min}, \text{max}]$ range of the variable.
- Each observation corresponds to a line drawn through point on each axis corresponding to value of the variable.
Parallel Coordinate Plot

- One vertical axis for each variable
- Every case is represented by a line
- Line intersects each of the vertical axis at the point corresponding to the attribute value of the case
- Popular visualization technique
- Complexity (number of axes) is directly proportional to the number of attributes (comp. scatterplot matrix)
- All attributes receive uniform treatment
- Potential problems of this visualization?

Inselberg 1997
Parallel Coordinate Plot

• The Technique
  – Technique lays out coordinate axes in parallel rather than orthogonal to each other (in contrast to scatterplots).
  – But there's a much simpler way of looking at it: as the representation of a data table.
  – Table describes car models released from 1970 to 1982, and contains their mileage (MPG), number of cylinders, horsepower, weight, and year they were introduced (among others).
Parallel Coordinate Plot

• Each of the table columns (left) is mapped onto a vertical axis in the plot (right).
• Each data value placed somewhere along the line, scaled to lie between the minimum at the bottom and the maximum at the top.
• Pure collection of points would not be very useful, so the points belonging to the same case (row) are connected with lines (polylines).
• By this the characteristic effect occurs where parallel coordinates are famous for.

<table>
<thead>
<tr>
<th>MPG</th>
<th>Cylinders</th>
<th>Horsepower</th>
<th>Weight</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>8</td>
<td>170</td>
<td>3,563</td>
<td>70</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>160</td>
<td>3,609</td>
<td>70</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>150</td>
<td>3,761</td>
<td>70</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>225</td>
<td>3,086</td>
<td>70</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>95</td>
<td>2,372</td>
<td>70</td>
</tr>
<tr>
<td>22</td>
<td>6</td>
<td>95</td>
<td>2,833</td>
<td>70</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>97</td>
<td>2,774</td>
<td>70</td>
</tr>
<tr>
<td>21</td>
<td>6</td>
<td>85</td>
<td>2,587</td>
<td>70</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>88</td>
<td>2,130</td>
<td>70</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>46</td>
<td>1,835</td>
<td>70</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>90</td>
<td>2,430</td>
<td>70</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>95</td>
<td>2,375</td>
<td>70</td>
</tr>
</tbody>
</table>
Parallel Coordinate Plot

• Insights in this multivariate dataset
  – Cylinders axis stands out because it only has a few different values.
  – The number of cylinders can only be a whole number, and there aren't more than eight here, so all the lines have to pass through a small number of points.
  – Data like this, and also categorical data, are usually not well suited for parallel coordinates.
  – As long as there is only one or two, it's not a problem, but when the data is largely or completely categorical, parallel coordinates do not show any useful information anymore.
Parallel Coordinate Plot

- Insights in this multivariate dataset
  - In the space between MPG and cylinders, you can tell that eight-cylinder cars generally have lower mileage than six- and four-cylinder ones.
  - Just follow the lines and look at how they cross: lots of crossing lines are an indication of an inverse relationship, and that is clearly the case here: the more cylinders, the lower the mileage.
Parallel Coordinate Plot

• Insights in this multivariate dataset
  – Between horsepower and weight, the situation is similar: more horsepower means heavier cars in general, but there is some spread in the values of course.
  – There is also one exception of a high-horsepower eight-cylinder car that is very light. Can you spot that outlier?
Parallel Coordinate Plot

- Insights in this multivariate dataset
  - Finally, the lines between weight and year cross over a lot, indicating that cars got lighter over the years.
  - You can also easily tell that the year axis only records a small number of different values, similar to the cylinders, from the triangular shapes converging on a few points.
  - While this is a very simple example, it shows the typical structures you would find in most datasets.
Parallel Coordinate Plot

- Brushing
Parallel Coordinate Plot

• Brushing
  – In addition to some experience in reading parallel coordinates, the best way to get to know a dataset using the technique is clearly interaction.
  – The main one in parallel coordinates is called brushing, for reasons that should be obvious from looking at the image below.
  – For this to make sense, we need to look at all axes.
Parallel Coordinate Plot

• Brushing
  – Brushing the years 1980 to 1982 on the (right-most) year axis.
  – Results in a brushed part of the lines in heavy black, with the rest still in the background in gray for context.
  – Looking at the axes from right to left you can see that the car models in this selection were almost all in the lower half of the weight range, and all of them were in the lower half in terms of horsepower.
  – The distribution of cylinders is also interesting: there only seems to be a single eight-cylinder car in this selection, all others are six cylinders or below. Mileage is also mostly above the mean value for all cars.
Parallel Coordinate Plot

• Brushing
  – Brushing the years 1970 to 1972 yields a very different image: weights, power, etc. are all over the place, and mileage is mostly in the lower half.
  – While higher values are to be expected, it is interesting to see that there was quite a spectrum of cars at the beginning of the decade, not just heavy eight-cylinder ones.
  – The trend over the years was towards lighter, more efficient cars, though.
Parallel Coordinate Plot

• More interactive Features
  – There is more to be said about interaction, of course: you can usually reorder axes to compare different ones side by side, combine brushing on different axes, flip axes (the arrows at the top of the images show the direction of the axis), etc.
Parallel Coordinate Plot for sets

- Bendix et al. 2005: Parallel Sets
- Parallel coordinates for categorical data
- Substitute individual data points by a frequency-based representation

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>crew</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female (s)</td>
<td>141</td>
<td>93</td>
<td>90</td>
<td>3</td>
</tr>
<tr>
<td>Female (d)</td>
<td>4</td>
<td>13</td>
<td>106</td>
<td>20</td>
</tr>
<tr>
<td>Male (s)</td>
<td>62</td>
<td>25</td>
<td>98</td>
<td>670</td>
</tr>
<tr>
<td>Male (d)</td>
<td>118</td>
<td>154</td>
<td>422</td>
<td>192</td>
</tr>
</tbody>
</table>
3D Parallel Coordinates

- Parallel 2D planes instead of vertical axes

http://www-vis.lbl.gov/Events/SC05/Drosophilia/index.html
Parallel Coordinate Plot

• Limitations

  – When the number of data items gets very high, there is a lot of overplotting that can make it impossible to see anything.

  – The number of dimensions on screen also needs to be below about a dozen at the same time to really make sense, anything above that gets very difficult to read.

  – There is a lot of work in visualization on automatic axes reordering, clustering of similar axes, etc. that makes this easier for high-dimensional data.

  – But in general, the technique simply works best for datasets with a moderate number of dimensions and no more than a few thousand records.

  – Also, as mentioned above, the data needs to be numerical. The technique does not work well for categorical data or data with few values per axis.
Scatterplot vs. Parallel Coordinates

- Correlations also detectable in Parallel Coordinate
Parallel Coordinate Plot

• Try it out
  – XmdvTool http://davis.wpi.edu/%7Exmdv/index.html
Geometric Transformations: discussion

• Advantages
  – Users’ familiarity with scatterplots (scatterplot matrix)
  – 2D patterns can easily be identified

• Disadvantages
  – Rather limited scalability
    • limited number of cases (Parallel Coordinate Plot)
    • limited number of dimensions (scatterplot matrix)
  – Overplotting and overlap
  – Labeling (Parallel Coordinates)
Glyph-based Visualizations
Glyph-Based Visualizations

• An object or symbol for representing data values. Glyphs are generally a way of representing many data values and are sometimes called icons. A common glyph is the arrow, often chosen to represent vector fields. The arrow depicts both speed and direction at a point.  
  [Keller and Keller, 1993]

• A glyph is a graphical object designed to convey multiple data values.  
  [Ware, 2004]
Glyph-Based Visualizations

• A primitive example of a glyph is an arrow whose visual attributes length, width, angle, and color might be used to encode four different data attributes in a single graphical object.

• The most prominent example for glyphs are Chernoff Faces [Chernoff, 1973], where the different parts of a conceptualized human face (mouth, nose, head, eyes, eyebrows, etc.) encode different dimensions of an n-dimensional data set.

• Advantages of glyph representations are foremost their easy learnability, long remembering periods, and the possibility to represent value restrictions visually, especially when multiple attributes are involved.
Glyph-Based Visualizations

- Glyph-based techniques
  - Star glyph
  - Chernoff faces
  - Stick-figure
  - Shape coding
  - Color icons

- Glyph: small-sized visual symbol
  - Variables are encoded as properties of glyph
  - Each case is represented by a single glyph
Star glyphs

- Coekin 1996
- Radial axes with equal angles (spokes of a wheel)
- Each axis represents a variable
- Each spoke length encodes a variable’s value
- May also be overlaid for better comparison

Chernoff Faces

• Chernoff 1973
• Humans are sensitive to a wide range of facial characteristics (e.g., eye size, length of a nose, etc.)
• 18 characteristics to encode data by stylized faces
• Positive evaluation results (Spence & Parr 1991)
• Some facial features seem to be able to carry more information than others (Morris et al. 1999; De Soete 1986)
Chernoff Faces

- Example
Stick-Figure Icons

• Pickett & Grinstein 1998
• Each case is represented by a stick figure
• Two attributes are mapped to XY position of the glyph
• Remaining dimensions are mapped to the angle and / or length of the 4 limbs
• When icons are densely packed a texture appears
• Texture pattern reveals characteristics of the data space
• Different members of stick-figure family for conveying different types of data structures
Stick-Figure Icons

• Stick-figure example
• Census data showing age (y), income (x), education, salary, language, marital status etc.
• Gender is encoded by two stick-figure families

Grinstein et al. 1989
Shape Coding

- Beddow 1990
- Each case is drawn as a glyph containing a rectangular grid
- Each grid cell represents one attribute
- Attribute value is encoded with gray scales
- Glyphs are positioned in a line, columns or encoded dimensions
- Highly compressed visualization without clutter and overlap (compare to stick figures)
- Identification of promising patterns
Shape Coding

- Attribute values encoded by white, grey, black
- 13 Variables gained from magnetosphere and solar wind data
- Includes one time variable (hour/day), which has been mapped to x/y

![Figure 1: Day by Hour: Thirteen Parameters of Magnetosphere and Solar Wind Data](image-url)
Color icons

- Levkowitz 1991, Keim & Kriegel 1994
- Shape coding with a focus on colors
- Arrangement is query-dependent (e.g., spiral: most relevant glyph is centered)
- What about compressing the visualization even more by using 1-pixel representations?
- Problem: users need at least 2x2 pixel per data value + pixels for borders to distinguish between the elements of the visualization
- This is different to pixel-based techniques, which will be discussed in the next lecture
8-dimensional result of a database query, 1,000 cases, Keim&Kriegel 1994
Glyph-Based Visualizations

• Advantages
  – Provide holistic overview of the information space
  – Exploit the human powerful ability of perceiving (texture) patterns and human face characteristics (Chernoff)
  – Direct metaphor of Chernoff-face-like icons (e.g. houses) may prove to be intuitive for novice users

• Disadvantages
  – Glyphs must be learned
  – Only suitable for small to medium data sets
  – Stick figures give a rather broad overview and may be difficult to interpret
  – Mappings may introduce biases in interpretation (e.g. the head shape of a Chernoff-face may be easier to perceive and compare than length of nose)
Pixel-based Visualizations
Pixel-Based Techniques

- Idea: each data item is represented by one colored pixel
- Value ranges are mapped to a fixed color sequence of full color (hue) scale but monotonically decreasing brightness
- Data values belonging to one attribute are displayed in a separate view – only one pixel per data value without need for a border
- But: users need to relate to different portions of the screen to perceive correlations
Pixel-Based Techniques

- Optimization Goal (OG) 1: arrangement of pixels in the sub-windows should preserve the 1D ordering into 2D plane as best as possible
  - Simple left-right or top-down arrangement do usually not provide useful results on a pixel-level
  - Space-filling curves (Paneo-Hilbert and Morton) provide maximum of locality preservation, but are difficult to follow and thus to relate between the sub-windows

- Recursive pattern
- Circle segments

Morton

Paneo-Hilbert
Recursive Pattern

- Keim et al. 1995
- Naturally ordered data set
  - Prices of IBM stock, Dow Jones index, Gold, exchange rate US-Dollar
  - September 1987-February 1995
  - 9 daily measurements for each stock
- Recursive pattern visualization
  - Lower-level patterns used as building blocks for higher level patterns
  - LP1 one day, LP2 one week, LP3 one month, LP4 one year

Keim 2000
Recursive Pattern

- 8 horizontal bars correspond to 8 years
- Subdivisions between the bars represent 12 months within each year
- Example analysis results
  - Gold price was very low in the sixth year
  - IBM price fell quickly after the first 1½ month
  - US-Dollar exchange rate was highest in the third year
Keim et al. 1995
Query Dependent Arrangement

• Ordering of data objects based on relevance to a given query
• Most relevant data object is placed in the center of the screen
• OG 2: for the pixel arrangement in each sub-window the distance to the center should correspond to the ordering of the data objects
• Simple spiral arrangement fulfills OG 2, but local clustering properties (OG 1) are weak, i.e. low probability that two pixels close on the screen are also close in the 1D ordered sequence of the query result set
• Generalized spiral technique: enhance the clustering qualities of the spiral technique by using screen-filling curves locally

Keim 2000
Spiral vs. Generalized Spiral

Keim1996
Circle Segment

- Rethink the shape of sub-window
- Rectangular shape of sub-windows makes efficient use of the screen
- For data sets with many dimensions, the pixels of one data object are rather far apart
- Makes it difficult to find patterns
- OP 3: minimize the average distance between the pixels (data values) belonging to one case

- Circle segment
  - Each dimension corresponds to a segment of a circle
  - Values of one dimension are drawn in a back and forth manner from the center of the circle to the outside
Circle Segment (CS) vs. Line Graphs

- 10 years of stock data for 7 stocks
- Line graph granularity is limited by the width of the screen
- CS: oldest data items in the middle of the circle, most recent ones are at the outside
- Easier to perceive patterns – no overlap of data

Keim et al. 1996b
Pixel-Based Techniques

• Advantages:
  – Large data sets can be visualized
  – Improved pattern detection due to non-overlap strategy

• Disadvantages
  – Labeling
  – Intuitiveness
  – Mapping color to quantitative data
  – ?

• Open questions
  – Drill-down to detail information?
  – ?
Downscaling of Dimensions
Downscaling of Dimensions

• Projecting n dimensions down to a lower dimensionality while retaining as much of the original information as possible

• Principal components analysis, Factor analysis, Multidimensional scaling
  – Statistical approaches to reduce the number of dimensions by finding the data’s main characteristics / patterns

• Self-organizing maps (SOM) aka Kohonen map
  – Reduce the dimensions of data by using self-organizing neural networks
  – Produces usually a 2D map which mirrors the similarity of cases (similar cases are grouped together)
  – Good tutorial: http://davis.wpi.edu/~matt/courses/soms/

• Problems: pruning of information; hard to interpret since display coordinates have no semantic meaning; SOM and MDS are iterative approaches (computationally hard, no unique result)
Exploring Multivariate Data
Rank-By-Feature

- Seo & Shneiderman 2004
- Part of the Hierarchical Clustering Explorer (HCE) (http://www.cs.umd.edu/hcil/multi-cluster/)
- Tabs: histogram and scatterplot ordering
- Implements systematic approach for data exploration
  - (1) study 1D, study 2D, then find features
  - (2) ranking guides insight, statistics confirm
- Tool provides low-dimensional projections as a histogram (1D) or scatterplot (2D)
- Users can select a feature detection criterion (e.g. test for normal distribution (1D), correlation coefficient (2D)) to rank projections
- The ranking facility is particularly helpful when the number of possible projections is too large to investigate: concentrate on the interesting ones
Rank-By-Feature

• Users start with 1D projections (histogram ordering)
• Four coordinated views
  – A: selection of ranking criterion
  – B: overview of scores for all dimensions (color coding: the brighter the color, the higher the score)
  – C: numerical / statistical detail for each dimension (e.g. score, mean, standard deviation)
  – D: display of histogram + boxplot (minimum, first quartile, median, third quartile, maximum)
Rank-By-Feature

• Some basic statistical terms
  – Mean: Sum of all values divided by the number of values
  – Median: Middle value of a distribution of values when ranked in order of magnitude
  – Mode: Single most common value
  – Variance: average squared deviation between the mean and the values
  – Standard deviation: square root of the variance (translates the variance into the original units of measurement)

• Statistical tests supported by HCE for 1D ranking
  – Normality of the distribution: distribution of items forms a symmetric, bell-shaped curve
  – Uniformity of the distribution: all of the values of a random variable occur with equal probability (results in a flat histogram)
  – Number of potential outliers
  – Number of unique values
Rank-By-Feature

- Move on to 2D projections (scatterplot ordering)
- Identify pairwise relationships between dimensions
- B: prism provides overview of scores for dimension pairs; score is color coded
- D: scatterplot browser; multiple browsers are possible;
- Ranking criteria
  - Correlation coefficient: direction and strength of linear relationship
  - Least square error for simple linear / curvilinear regression: how well does the regression model fit
  - Number of items in a user-defined region of interest & uniformity of scatterplot