TABULAR DATA

Miriah Meyer
University of Utah
administrivia . . .
- Exam grades out after fall break

- Parallel coordinates assignment out today
last time . . .
Reducing Items and Attributes

Filter

- Items

- Attributes

Aggregate

- Items

- Attributes
filter
elements are eliminated
filter
elements are eliminated

dynamic queries
filter

elements are eliminated

dynamic queries
coupling between encoding and interaction so that user can immediately see the results of an action
ITEM FILTERING
aggregate

a group of elements is represented by a new derived element that stands in for the entire group
attribute aggregation

1) group attributes and compute a similarity score across the set
2) dimensionality reduction, to preserve meaningful structure
today . . .
dataset types

Tables
- Items
- Attributes

Networks & Trees
- Items (nodes)
- Links
- Attributes

Fields
- Grids
- Positions
- Attributes

Geometry
- Items
- Positions

Clusters, Sets, Lists
- Items
Arrange Tables

Express Values

Separate, Order, Align Regions

Separate → Order → Align

1 Key List → 2 Keys Matrix → 3 Keys Volume → Many Keys Recursive Subdivision

Axis Orientation

Rectilinear → Parallel → Radial

Layout Density

Dense → Space-Filling
arrange is the focus of all four design choices for tabular data
Spatial channels are the most effective for all attribute types.
recall: attribute semantics
when we arrange tabular data, attributes are chosen to be keys and values.

flat

multidimensional
encode using zero keys: scatterplots

Arrange Tables

- Express Values

- Separate, Order, Align Regions
  - Separate
  - Order
  - Align

- Axis Orientation
  - Rectilinear
  - Parallel
  - Radial

- Layout Density
  - Dense
  - Space-Filling
Why Is Her Paycheck Smaller?

Nearly every occupation has the gap — the seemingly unbridgeable chasm between the size of the paycheck brought home by a woman and the larger one earned by a man doing the same job. Economists cite a few reasons: discrimination as well as personal choices within occupations are two major factors, and part of the gap can be attributed to men having more years of experience and logging more hours.

Hannah Fairfield and Graham Roberts/The New York Times
encode one key attribute: bar, dot, & line charts
Four Ways to Slice Obama’s 2013 Budget Proposal

Four Ways to Slice Obama’s 2013 Budget Proposal

Explore every nook and cranny of President Obama’s federal budget proposal.

How $3.7 Trillion Is Spent

Mr. Obama’s budget proposal includes $3.7 trillion in spending in 2013, and forecasts a $901 billion deficit.

Circles are sized according to the proposed spending.

- $100 billion
- $1 billion

Color shows amount of cut or increase from 2012.

-25% -5% 0 +5% +25%

The proposal forecasts a $901 billion deficit.

Chart shows $3.7 trillion authorized to be spent in 2013. (Total spending is estimated to be $3.8 trillion, including funds authorized in other years). Negative budget authority, which results from fees or other collections, is shown only on the department totals tab, but is included in other totals.

By RYAN CARSON – Special To The Times
design considerations
don’t use line charts for categorical attributes!

ok: “Men are taller than women (on average)”

bad: “The more male a person is, the taller he/she is”

ok: “Twelve year olds are taller than ten year olds”

ok: “Height increases with age”
BANKING TO 45°

The aspect ratio of a graph is an important factor for judging rate of change.

perceptual principle:
most accurate angle judgement is at 45°
MULTISCALE BANKING TO 45°

- frequency domain analysis
- find interesting regions at multiple scales
  - FFT the data and smooth by convolution with Gaussian
  - select aspect ratios as spikes in power spectrum
- create trend curves with low-pass filter
- bank all to 45°
ARC LENGTH-BASED

Minimize the arc length of the data curve while keeping the area under the plot constant.

- scale and parameterization invariant
- preserves symmetries
- robust on a wide range of inputs
- fast
An Empirical Model of Slope Ratio Comparisons

(Corrected February 1, 2013)*

Justin Talbot, John Gerth, and Pat Hanrahan

1820 1840 1860 1880 1900 1920 1940 1960

1820 1840 1860 1880 1900 1920 1940 1960

Fig. 1. Both plots show the same data set—the change in the length of a day (in microseconds) over 140 years—with different aspect ratios. The segment redness corresponds to the error that viewers will make when comparing the slope of that segment to all other slopes in the plot—e.g., predicted by our empirical model. The aspect ratio on the right minimizes the total absolute predicted error.

Abstract—Comparing slopes is a fundamental task in visualizing data and it influences how easy these comparisons are to make. Through exploratory pilot studies that expand Cleveland et al.'s experimental design, we develop an empirical model of slope ratio estimation that fits more extreme slope ratio judgments and two common slope ratio estimation strategies. We then run two experiments to validate our model. In the first, we show that our model fits more generally than the one proposed by Cleveland et al. and we find that, in general, slope ratio errors are not minimized around 45°. In the second experiment, we explore a novel hypothesis raised by our model: that visible baselines can substantially mitigate errors made in slope judgments. We conclude with an application of our model to aspect ratio selection.

Index Terms—Banking to 45 degrees, slope perception, orientation resolution, aspect ratio selection.

1 INTRODUCTION

Banking to 45° is a classic design guideline in information visualization. We describe our methodological approach. Section 4 describes our pilot experiments.
results

- people use two different strategies to estimate slope
  - angle and height

- slope angle accuracy NOT minimized at 45
AUTOMATIC TICKS

- optimization of tick placement
- optimization of label formatting, font size, and orientation

(a) Heckbert
(b) R's pretty
(c) Wilkinson
(d) Extended
AUTOMATIC TICKS

- optimization of tick placement
- optimization of label formatting, font size, and orientation
encode multiple key attributes

Arrange Tables
→ Express Values

Separate, Order, Align Regions
→ Separate
→ Order
→ Align
→ 1 Key List
→ 2 Keys Matrix
→ 3 Keys Volume
→ Many Keys Recursive Subdivision

Axis Orientation
→ Rectilinear
→ Parallel
→ Radial

Layout Density
→ Dense
→ Space-Filling
encode using two keys: stacked bar chart
encode using two keys: heatmap

- uses heatmap representation
  - matrix layout using keys
  - encode values with color

- often augmented with clustering
encode using two keys: heatmap

- uses heatmap representation
- matrix layout using keys
- encode values with color

- often augmented with clustering
encode using two keys: heatmap

- uses heatmap representation
  - matrix layout using keys
  - encode values with color

- often augmented with clustering
encode using two keys:

- uses heatmap representation
  - matrix layout using keys
  - encode values with color

- often augmented with clustering

[Eisen98]
Interactively Exploring Hierarchical Clustering Results

The Hierarchical Clustering Explorer provides a dendrogram and color mosaic linked to two-dimensional scattergrams, a variety of visualization options, and dynamic query controls for use in genomic microarray data analysis.

Jinwook Seo
Ben Shneiderman
University of Maryland, College Park

Molecular biologists and geneticists seek to understand the function of genes, including the more than 6,000 genes in the yeast genome and the estimated 40,000 genes in the human genome. Recently developed for genome analysis, DNA microarrays—also known as gene arrays or gene chips—usually consist of glass or nylon substrates that measure 1 x 3 inches or smaller. These chips contain specific DNA gene samples spotted in an array by a robotic printing device. Researchers spread fluorescently labeled messenger RNA (mRNA) from an experimental condition onto the DNA gene samples in the array. This mRNA binds (hybridizes) strongly with some DNA makes it impossible to display a large microarray experiment—on one screen.

Researchers also struggle to understand the implications of a specific clustering result. Because the clusters occupy a high-dimensional space and involve so many experimental conditions, researchers find it difficult to view patterns on a 2D or even a 3D display. Further, data can contain hundreds of variously sized clusters, which makes spotting the meaningful clusters a challenge, especially when using a static display. Users need an efficient interactive visualization tool to facilitate pattern extraction from microarray data sets.

Hierarchical clustering has been shown to be effective in microarray data analysis for identifying
- scalability through interaction
  - interactively controlled reduction
    - aggregation, filtering, and navigation
  - view coordination
    - overview+detail, small multiples, side-by-side
      multiform views with linked highlighting
critique: what do you think?
align using multiple keys

LineUp: Visual Analysis of Multi-Attribute Rankings

Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister and Marc Streit

Fig. 1. LineUp showing a ranking of the top Universities according to the QS World University Ranking 2012 dataset with custom attributes and weights, compared to the official ranking.

Abstract—Rankings are a popular and universal approach to structuring otherwise unorganized collections of items by computing a rank for each item based on the value of one or more of its attributes. This allows us, for example, to prioritize tasks or to evaluate the performance of products relative to each other. While the visualization of a ranking itself is straightforward, its interpretation is not.
challenge

- rankings based on single attribute are trivial to display

- when based on multiple attributes:
  - not clear how attributes contribute to ranking
  - not clear how changes to multiple attributes will affect ranking

- different contexts/people/situations will rank on multiple attributes differently
requirements

- encode rank
- encode cause of rank
- support multiple attributes
- support filtering
- enable flexible mapping of attribute values to scores
- adapt scalability to the task
- handle missing values
- interactive refinement and visual feedback
- rank-driven attribute optimization
- compare multiple rankings
LineUp
Visual Analysis of Multi-Attribute Rankings

Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister and Marc Streit
LineUp
Visual Analysis of Multi-Attribute Rankings

Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister and Marc Streit
LINEUP

- **problem**: support creation, refinement, and exploration of multi-attribute rankings

- **abstraction**
  - created a comprehensive list of requirements

- **design considerations**
  - use stacked bar charts to show scores
  - use links to show change in ranking
  - scented widgets to show distributions of scores
  - strong focus on interactivity for exploring and modifying rankings
critique: what do you think?
spatial axis orientation

Arrange Tables
- Express Values

Separate, Order, Align Regions
- Separate
- Order
- Align
  - 1 Key List
  - 2 Keys Matrix
  - 3 Keys Volume
  - Many Keys Recursive Subdivision

Axis Orientation
- Rectilinear
- Parallel
- Radial

Layout Density
- Dense
- Space-Filling
SPLOMs: scatterplot matrices

nine characteristics of Abalone (sea snails)

Wilkinson et al., 2005
parallel coordinates

- **scatterplot limitation**: visual representation with orthogonal axes
  - can show only two attributes with spatial position channel

- **alternative**: line up axes in parallel to show many attributes with position
  - item encoded with a line with n segments
  - n is the number of attributes shown
parallel coordinates
EXAMPLE

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>D2</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>D3</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
EXAMPLE
PARALLEL COORDINATES TASK

- show correlation
  - positive correlation: straight lines
  - negative correlation: all lines cross at a single pt

Figure 3. Parallel Coordinate Plot of Six-Dimensional Data Illustrating Correlations of $\rho = 1, .8, .2, 0, -.2, -.8,$ and $-1$. Wegman 1990
PARALLEL COORDINATES TASK

do you see any correlations?
PARALLEL COORDINATES TASK

- visible patterns only between neighboring axis pairs
- how to pick axis order?
  - usual solution: reorderable axes, interactive exploration
    - same weakness as many other techniques
    - downside: human-powered search
- not directly addressed in HPC paper

Fua 1999
What is this?

A multidimensional explorer of nutrient data from the USDA.

The parallel coordinates display the nutrient content of foods in the database across 14 dimensions, colored by food group.

Food Groups

- 327 Baby Foods
- 619 Beef Products
- 278 Beverages

Sample of 25 entries

- Alcoholic Beverage, wine, table, red, Gamay
- Alcoholic beverage, distilled, whiskey, 86 proof
- Bananas, raw
- Carrot, raw
Hierarchical Parallel Coordinates for Exploration of Large Datasets

Ying-Huey Fua, Matthew O. Ward and Elke A. Rundensteiner
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609
{yingfua,matt,rundenst}@cs.wpi.edu

Abstract

Our ability to accumulate large, complex (multivariate) data sets has far exceeded our ability to effectively process them in search of patterns, anomalies, and other interesting features. Conventional multivariate visualization techniques generally do not scale well with respect to the size of the data set. The focus of this paper is on the interactive visualization of large multivariate data sets based on a number of novel extensions to the parallel coordinates display technique. We develop a multiresolutional view of the data via hierarchical clustering, and use a variation on parallel coordinates to convey aggregation information for the resulting clusters. Users can then navigate the resulting structure until the desired focus region and level of detail is reached, using our suite of navigational and filtering tools. We describe the design and implementation of our hierarchical parallel coordinates system which is based on extending the XmdvTool system. Lastly, we show examples of the tools and techniques applied to large (hundreds of thousands of records) multivariate data sets.

Keywords: Large-scale multivariate data visualization, hierarchical data exploration, parallel coordinates.

1 Introduction

- Dimensional embedding techniques, such as dimensional stacking [16] and worlds within worlds [6].
- Dimensional subsetting, such as scatterplots [5].
- Dimensional reduction techniques, such as multidimensional scaling [20, 15, 29], principal component analysis [12] and self-organizing maps [14].

Most of these techniques do not scale well with respect to the size of the data set. As a generalization, we postulate that any method that displays a single entity per data point invariably results in overlapped elements and a convoluted display that is not suited for the visualization of large data sets. The quantification of the term “large” varies and is subject to revision in sync with the state of computing power. For our present application, we define a large data set to contain $10^6$ to $10^9$ data elements or more.

Our research focus extends beyond just data display, incorporating the process of data exploration, with the goal of interactively uncovering patterns or anomalies not immediately obvious or comprehensible. Our goal is thus to support an active process of discovery as opposed to passive display. We believe that it is only through data exploration that meaningful ideas, relations, and subsequent inferences may be extracted from the data. The major hurdles we need to overcome are the problems of display density/clutter (too much information on the face) and interaction, and...
HIERARCHICAL PARALLEL COORDINATES

- goal: scale up parallel coordinates to large datasets
- challenge: overplotting/occlusion

Fua 1999
HPC: ENCODING DERIVED DATA

- visual representation: variable-width opacity bands
  - show whole cluster, not just single item
  - min / max: spatial position
  - cluster density: transparency
  - mean: opaque

Fua 1999
HPC: INTERACTING WITH DERIVED DATA

- interactively change level of detail to navigate cluster hierarchy
parallel sets

- builds on PC to better handle categorical data
  - discrete
  - small number of values
  - no implied ordering between attributes

- **task**: find relationship between attributes, not outliers

- interaction driven technique
visual encoding

- boxes scaled by frequency
- color coded by values for current active dimension
visual encoding

- boxes expand to show histogram
interaction: reorder
interaction: aggregate
interaction: filter

Bendix, Kosara, Hauser, 2005
interaction: highlight

Bendix, Kosara, Hauser, 2005
critique: what do you think?
radial layouts

Axis Orientation
- Rectilinear
- Parallel
- Radial

http://bl.ocks.org/mbostock/3888852
radial layouts use polar coordinates
radar plot & star graph

- “parallel” dimensions in polar coordinate space
- best if same units apply to each axis

von Mayr, 1887
pie charts: take care with accuracy
filling space

Arrange Tables

- Express Values

Separate, Order, Align Regions

- Separate
- Order
- Align

- 1 Key List
- 2 Keys Matrix
- 3 Keys Volume
- Many Keys Recursive Subdivision

Axis Orientation

- Rectilinear
- Parallel
- Radial

Layout Density

- Dense
- Space-Filling
dense pixel display: VisDB

- represent each data item, or each attribute in an item as a single pixel

- can fit as many items on the screen as there are pixels, on the order of millions

- relies heavily on color coding

- challenge: what’s the layout?
the data...

- large database where each item has multiple attributes (on the order of 10)

- **goal**: visualize the relevance of set of items which satisfy a query

- plot out data items in a spiral pattern, ordered by relevance

Keim, Kreigel, 1994
Figure 1: Arrangement of Windows for Displaying five-dimensional Data

Figure 2: 2D-Arrangement of one Dimension

Figure 3: Grouping Arrangement for five-dimensional Data

Figure 4: Spiral Shaped Arrangement of one Dimension

Keim, Kreigel, 1994
Figure 1: Relevance Factor

Figure 2: Arrangement of Windows for Displaying Five-Dimensional Data

Figure 3: 2D-Arrangement of One Dimension

Figure 4: Grouping Arrangement for Five-Dimensional Data

Keim, Kreigel, 1994
critique: what do you think?

Keim, Kreigel, 1994
L13: Trees & Graphs

REQUIRED READING
Chapter 9
Arrange Networks and Trees

9.1 The Big Picture

This chapter covers design choices for arranging network data in space, summarized in Figure 9.1. The node–link diagram family of visual encoding idioms uses the connection channel, where marks represent links rather than nodes. The second major family of network encoding idioms are matrix views that directly show adjacency relationships. Tree structure can be shown with the containment channel, where enclosing link marks show hierarchical relationships through nesting.

9.2 Connection: Link Marks

The most common visual encoding idiom for tree and network data is with node–link diagrams, where nodes are drawn as point marks and the links connecting them are drawn as line marks. This idiom uses connection marks to indicate the relationships between
Many people in the information visualization and graph-drawing communities consider tree visualization (see the sidebar) a solved problem. Although Kim Marriott and Peter Stuckey have shown that finding an optimal tree layout can be an NP-complete problem,1 reasonably good tree layouts can nevertheless be computed efficiently in terms of runtime and screen space utilization. In the course of the search for heuristics to generate ever-tidier tree layouts, the comparatively simple problem of transforming parent-child relationships into graphical representations has been solved many times and is still the subject of information visualization research. Researchers have explored and published almost every way of arranging a tree's nodes in 2D and 3D; encoding them in different shapes or forms; and folding, unfolding, or otherwise interactively manipulating them. The plethora of tree visualization techniques poses challenges to researchers and developers. Researchers, especially those new to the field, have no way of knowing every tree visualization that has been published, even over just the last two decades. So, they often reinvent existing techniques. Without pointing fingers—my colleagues and I have done our fair share of unwittingly reinventing visualizations—I've noticed that the published tree visualizations include a number of such reinventions. This is hardly surprising because it's almost impossible for peer reviewers as well to have a complete overview of prior research. The same holds true for developers who implement tree visualizations for their customers, but with potentially direr consequences. Developing something that already exists could lead to ugly intellectual-property issues. And even though it seems like a good starting point to assume that something similar to your own idea has already been done, finding that similar technique can be extremely difficult.

However, opportunities also exist. The long history and remarkable coverage of the design space offer the opportunity to step back, take a look at the bigger picture, and learn from it. For example, we can identify recurring design patterns. Moreover, we can trace back the evolution of our modern visualization techniques to the visual archetypes that might have inspired them. To address the challenges and exploit the opportunities, we must make a laborious but important first step: we must collect existing tree visualization techniques and form a reference for them that's as complete as possible. This is where the treevis.net project comes into the picture.

Hunting and Gathering Tree Visualizations

In early 2010, I set out to ramble through the available tree visualization literature and websites. Most tree visualizations could readily be excerpted from conference proceedings and journals. From these, I slowly built a “convex hull” by seeking those papers cited by the ones I found and those that cited the found ones. But this covered only the scholarly publications. Much harder to hunt down were the visualizations that appear on Flickr. Tree visualization (sometimes called hierarchy visualization) is a branch of information visualization dedicated to the graphical representation of connected, acyclic graphs—trees. Tree structures are common in many aspects of everyday life, such as ancestry (family trees) or file system organization (directory trees). Most tree visualizations are developed for rooted trees, which contain a selected top element, the root node; intermediate elements, the internal nodes; and bottom elements, the leaves. Drawing on the family tree metaphor, nodes standing in direct relation are called the parent node (the node closer to the root) and child node (the node further from the root).