Chap 15: Analysis Case Studies
Paper: D3

Tamara Munzner
Department of Computer Science
University of British Columbia

UBC CPSC 547: Information Visualization
Wed Nov 5 2014

http://www.cs.ubc.ca/~tmm/courses/547-14/#chap15
Analysis Case Studies

Scagnostics

VisDB

InterRing

HCE

PivotGraph

Constellation
Graph-Theoretic Scagnostics

• scatterplot diagnostics
  – scagnostics SPLOM: each point is one original scatterplot

[Graph-Theoretic Scagnostics Wilkinson, Anand, and Grossman. Proc InfoVis 05.]
## Scagnostics analysis

<table>
<thead>
<tr>
<th>System</th>
<th>Scagnostics</th>
</tr>
</thead>
<tbody>
<tr>
<td>What: Data</td>
<td>Table.</td>
</tr>
<tr>
<td>What: Derived</td>
<td>Nine quantitative attributes per scatterplot (pairwise combination of original attributes).</td>
</tr>
<tr>
<td>Why: Tasks</td>
<td>Identify, compare, and summarize; distributions and correlation.</td>
</tr>
<tr>
<td>How: Encode</td>
<td>Scatterplot, scatterplot matrix.</td>
</tr>
<tr>
<td>How: Manipulate</td>
<td>Select.</td>
</tr>
<tr>
<td>How: Facet</td>
<td>Juxtaposed small-multiple views coordinated with linked highlighting, popup detail view.</td>
</tr>
<tr>
<td>Scale</td>
<td>Original attributes: dozens.</td>
</tr>
</tbody>
</table>
VisDB

• table: draw pixels sorted, colored by relevance
• group by attribute or partition by attribute into multiple views

VisDB Results

- partition into many small regions: dimensions grouped together

VisDB Results

• partition into small number of views
  – inspect each attribute
## VisDB Analysis

<table>
<thead>
<tr>
<th>System</th>
<th>VisDB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What: Data</strong></td>
<td>Table (database) with $k$ attributes; query returning table subset (database query).</td>
</tr>
<tr>
<td><strong>What: Derived</strong></td>
<td>$k + 1$ quantitative attributes per original item: query relevance for the $k$ original attributes plus overall relevance.</td>
</tr>
<tr>
<td><strong>Why: Tasks</strong></td>
<td>Characterize distribution within attribute, find groups of similar values within attribute, find outliers within attribute, find correlation between attributes, find similar items.</td>
</tr>
<tr>
<td><strong>How: Encode</strong></td>
<td>Dense, space-filling; area marks in spiral layout; colormap: categorical hues and ordered luminance.</td>
</tr>
<tr>
<td><strong>How: Facet</strong></td>
<td>Layout 1: partition by attribute into per-attribute views, small multiples. Layout 2: partition by items into per-item glyphs.</td>
</tr>
<tr>
<td><strong>How: Reduce</strong></td>
<td>Filtering</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Attributes: one dozen. Total items: several million. Visible items (using multiple views, in total): one million. Visible items (using glyphs): 100,000</td>
</tr>
</tbody>
</table>
Hierarchical Clustering Explorer

- heatmap, dendrogram
- multiple views

[Interactively Exploring Hierarchical Clustering Results. Seo and Shneiderman, IEEE Computer 35(7): 80-86 (2002)]
HCE

- rank by feature idiom
  - 1D list
  - 2D matrix

<table>
<thead>
<tr>
<th>System</th>
<th>Hierarchical Clustering Explorer (HCE)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What: Data</strong></td>
<td>Multidimensional table: two categorical key attributes (genes, conditions); one quantitative value attribute (gene activity level in condition).</td>
</tr>
<tr>
<td><strong>What: Derived</strong></td>
<td>Hierarchical clustering of table rows and columns (for cluster heatmap); quantitative derived attributes for each attribute and pairwise attribute combination; quantitative derived attribute for each ranking criterion and original attribute combination.</td>
</tr>
<tr>
<td><strong>Why: Tasks</strong></td>
<td>Find correlation between attributes; find clusters, gaps, outliers, trends within items.</td>
</tr>
<tr>
<td><strong>How: Encode</strong></td>
<td>Cluster heatmap, scatterplots, histograms, boxplots. Rank-by-feature overviews: continuous diverging colormaps on area marks in reorderable 2D matrix or 1D list alignment.</td>
</tr>
<tr>
<td><strong>How: Reduce</strong></td>
<td>Dynamic filtering; dynamic aggregation.</td>
</tr>
<tr>
<td><strong>How: Manipulate</strong></td>
<td>Navigate with pan/scroll.</td>
</tr>
<tr>
<td><strong>How: Facet</strong></td>
<td>Multiform with linked highlighting and shared spatial position; overview-detail with selection in overview populating detail view.</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Genes (key attribute): 20,000. Conditions (key attribute): 80. Gene activity in condition (quantitative value attribute): 20,000 × 80 = 1,600,000.</td>
</tr>
</tbody>
</table>
InterRing

# InterRing Analysis

<table>
<thead>
<tr>
<th>System</th>
<th>InterRing</th>
</tr>
</thead>
<tbody>
<tr>
<td>What: Data</td>
<td>Tree.</td>
</tr>
<tr>
<td>Why: Tasks</td>
<td>Selection, rollup/drilldown, hierarchy editing.</td>
</tr>
<tr>
<td>How: Facet</td>
<td>Linked coloring and highlighting.</td>
</tr>
<tr>
<td>How: Reduce</td>
<td>Embed: distort; multiple foci.</td>
</tr>
<tr>
<td>Scale</td>
<td>Nodes: hundreds if labeled, thousands if dense. Levels in tree: dozens.</td>
</tr>
</tbody>
</table>
PivotGraph

• derived rollup network

[Visual Exploration of Multivariate Graphs, Martin Wattenberg, CHI 2006.]
## PivotGraph Analysis

<table>
<thead>
<tr>
<th>Idiom</th>
<th>PivotGraph</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What</strong>: Data</td>
<td>Network.</td>
</tr>
<tr>
<td><strong>What</strong>: Derived</td>
<td>Derived network of aggregate nodes and links by roll-up into two chosen attributes.</td>
</tr>
<tr>
<td><strong>Why</strong>: Tasks</td>
<td>Cross-attribute comparison of node groups.</td>
</tr>
<tr>
<td><strong>How</strong>: Encode</td>
<td>Nodes linked with connection marks, size.</td>
</tr>
<tr>
<td><strong>How</strong>: Manipulate</td>
<td>Change: animated transitions.</td>
</tr>
<tr>
<td><strong>How</strong>: Reduce</td>
<td>Aggregation, filtering.</td>
</tr>
<tr>
<td>Scale</td>
<td>Nodes/links in original network: unlimited. Roll-up attributes: 2. Levels per roll-up attribute: several, up to one dozen.</td>
</tr>
</tbody>
</table>
Analysis example: Constellation

• data
  – multi-level network
    • node: word
    • link: words used in same dictionary definition
  – subgraph for each definition
    – not just hierarchical clustering
  – paths through network
    • query for high-weight paths between 2 nodes
      – quant attrib: plausibility

Using space: Constellation

• visual encoding
  – link connection marks between words
  – link containment marks to indicate subgraphs
  – encode plausibility with horiz spatial position
  – encode source/sink for query with vert spatial position

• spatial layout
  – curvilinear grid: more room for longer low-plausibility paths

Using space: Constellation

• edge crossings
  – cannot easily minimize instances, since position constrained by spatial encoding
  – instead: minimize perceptual impact

• views: superimposed layers
  – dynamic foreground/background layers on mouseover, using color
  – four kinds of constellations
    • definition, path, link type, word
      – not just 1-hop neighbors

## Constellation Analysis

<table>
<thead>
<tr>
<th>System</th>
<th>Constellation</th>
</tr>
</thead>
<tbody>
<tr>
<td>What: Data</td>
<td>Three-level network of paths, subgraphs (definitions), and nodes (word senses).</td>
</tr>
<tr>
<td>Why: Tasks</td>
<td>Discover/verify: browse and locate types of paths, identify and compare.</td>
</tr>
<tr>
<td>How: Encode</td>
<td>Containment and connection link marks, horizontal spatial position for plausibility attribute, vertical spatial position for order within path, color links by type.</td>
</tr>
<tr>
<td>How: Reduce</td>
<td>Superimpose dynamic layers.</td>
</tr>
</tbody>
</table>
Analysis

• expected in your paper/topic presentations
  – in addition to content summarization

• expected in your final projects
Paper: D3

• paper types
  – design studies
  – technique/algorithm
  – evaluation
  – model/taxonomy

– system
  • today’s emphasis

Toolkits

- imperative: how
  - low-level rendering: Processing, OpenGL
  - parametrized visual objects: prefuse
    - also flare: prefuse for Flash

- declarative: what
  - Protoviz, D3, ggplot2
  - separation of specification from execution

- considerations
  - expressiveness
    - can I build it?
  - efficiency
    - how long will it take?
  - accessibility
    - do I know how?
OpenGL

• graphics library
  – pros
    • power and flexibility, complete control for graphics
    • hardware acceleration
    • many language bindings: C, C++, Java (w/ JOGL)
  – cons
    • big learning curve if you don’t know already
    • no vis support, must roll your own everything
  – example app: TreeJuxtaposer

Processing

• layer on top of Java/OpenGL
• visualization esp. for artists/designers
• pros
  – great sandbox for rapid prototyping
  – huge user community, great documentation
• cons
  – poor widget library support
• example app: MizBee

prefuse

• infovis toolkit, in Java
• fine-grained building blocks for tailored visualizations
• pros
  – heavily used (previously)
  – very powerful abstractions
  – quickly implement most techniques covered so far
• cons
  – hasn’t been under active development for
  – nontrivial learning curve
• example app: DOIITrees Revisited

prefuse

- separation: abstract data, visual form, view
  - data: tables, networks
  - visual form: layout, color, size, ...
  - view: multiple renderers
InfoVis Reference Model

- conceptual model underneath design of prefuse and many other toolkits
- heavily influenced much of infovis (including nested model)
  - aka infovis pipeline, data state model

[Redrawn Fig 1.23. Card, Mackinlay, and Shneiderman. Readings in Information Visualization: Using Vision To Think, Chapter 1. Morgan Kaufmann, 1999.]
Declarative toolkits

• imperative tools/libraries
  – say exactly **how** to do it
  – familiar programming model
    • OpenGL, prefuse, ...

• declarative: other possibility
  – just say **what** to do
  – Protovis, D3
Protovis

• declarative infovis toolkit, in Javascript
  – also later Java version

• marks with inherited properties

• pros
  – runs in browser
  – matches mark/channel mental model
  – also much more: interaction, geospatial, trees,...

• cons
  – not all kinds of operations supported

• example app: NapkinVis (2009 course project)

[Fig 1, 3. Chao. NapkinVis. http://www.cs.ubc.ca/~tmm/courses/533-09/projects.html#will]
Protovis Validation

• wide set of old/new app examples
  – expressiveness, effectiveness, scalability
  – accessibility

• analysis with cognitive dimensions of notation
  – closeness of mapping, hidden dependencies
  – role-expressiveness visibility, consistency
  – viscosity, diffuseness, abstraction
  – hard mental operations

D3

• declarative infovis toolkit, in Javascript
• Protovis meets Document Object Model
• pros
  – seamless interoperability with Web
  – explicit transforms of scene with dependency info
  – massive user community, many thirdparty apps/libraries on top of it, lots of docs
• cons
  – even more different from traditional programming model
• example apps: many
D3

• objectives
  – compatibility
  – debugging
  – performance

• related work typology
  – document transformers
  – graphics libraries
  – infovis systems
    • general note: all related work sections are a mini-taxonomy!

D3 capabilities

• query-driven selection
  – selection: filtered set of elements queries from the current doc
    • also partitioning/grouping!
  – operators act on selections to modify content
    • instantaneous or via animated transitions with attribute/style interpolators
    • event handlers for interaction

• data binding to scenegraph elements
  – data joins bind input data to elements
  – enter, update, exit subselections
  – sticky: available for subsequent re-selection
  – sort, filter

D3 Features

• document transformation as atomic operation
  – scene changes vs representation of scenes themselves

• immediate property evaluation semantics
  – avoid confusing consequences of delayed evaluation

• validation
  – performance benchmarks
    • page loads, frame rate
  – accessibility
    • everybody has voted with their feet by now!