Data Visualization (CIS 468)

Interaction & Networks

Dr. David Koop
Good: Data magnitude <=> Mark magnitude

If Bush tax cuts expire...

Top tax rate

40%  39.6%
30%  39.6%
20%  39.6%
10%  39.6%
0%   39.6%

Now  Jan. 1, 2013

[Flowing Data, 2012]
Tufte's Lie Factor

- Size of effect = \( \frac{(2nd \ value - 1st \ value)}{1st \ value} \)
- Lie factor = \( \frac{\text{size of effect in graphic}}{\text{size of effect in data}} \)
- In the graphic:

\[
\text{Lie Factor} = \frac{5.3 - 0.6}{0.6} = \frac{27.5 - 18}{18} = 14.8
\]
Avoid Chartjunk

Extraneous visual elements that distract from the message.
Maximize Data-Ink Ratio

[Bar chart showing data for males and females in two income brackets: 0-$24,999 and $25,000+]

[Chart showing data for males and females in two income brackets: 0-$24,999 and $25,000+]

[Text: Maximize Data-Ink Ratio]

[Diagram: Data-to-Ink Ratio and Unjustified 3D]
Eyes Beat Memory

• Reduce cognitive load (using up working memory)
  - Our ability to do visual search degrades quickly

• Animation versus side-by-side views
  - Jump cuts vs. animated transitions
  - Side-by-side views

• Change blindness
  - We may fail to notice drastic changes if we are focused on a specific task those changes are not relevant to
Selection

• Selection is often used to initiate other changes
• User needs to select something to drive the next change
• What can be a selection target?
  - Items, links, attributes, (views)
• How?
  - mouse click, mouse hover, touch
  - keyboard modifiers, right/left mouse click, force
• Selection modes:
  - Single, multiple
  - Contiguous? (all together in one region)
Highlighting

• Selection is the user action
• Feedback is important!
• How? Change selected item's visual encoding
  - Change color: want to achieve visual popout
  - Add outline mark: allows original color to be preserved
  - Change size (line width)
  - Add motion: marching ants
Highlighting

• Selection is the user action
• Feedback is important!
• How? Change selected item's visual encoding
  - Change color: want to achieve visual popout
  - Add outline mark: allows original color to be preserved
  - Change size (line width)
  - Add motion: marching ants
Assignment 3

- Link
- Geographic Vis & Colormaps in D3
- Citibike Station Data aggregated by New York City Community Districts
- Due November 1
Interaction Overview

- **Change over Time**

- **Select**

- **Navigate**

  - **Item Reduction**
    - Zoom
      - Geometric or Semantic
    - Pan/Translate
    - Constrained

  - **Attribute Reduction**
    - Slice
    - Cut
    - Project

[Munzner (ill. Maguire), 2014]
Navigation

• Fix the layout of all visual elements but provide methods for the viewpoint to change

• Camera analogy: only certain features visible in a frame
  - Zooming
  - Panning (aka scrolling)
  - Translating
  - Rotating (rare in 2D, important in 3D)
Navigation

- **Item Reduction**
  - **Zoom**
    - *Geometric* or *Semantic*
  - **Pan/Translate**
  - **Constrained**

- **Attribute Reduction**
  - **Slice**
  - **Cut**
  - **Project**

[Munzner (ill. Maguire), 2014]
Zooming
Zooming
Semantic Zooming
Zooming

- Geometric Zooming: just like a camera
- Semantic Zooming: visual appearance of objects can change at different scales
- LiveRAC Example: (focus + context)

[McLachlan et al., 2008]
Navigation Constraints

• **Unconstrained** navigation: walking around in the world or an immersive 3D environment
  - Fairly standard in computer games to go where you want
  - Constrained by walls, objects (collision detection)

• Constrained navigation:
  - 3D: camera must be right-side up
  - Limit pan/zoom to certain areas
  - Comes up often with **multiple views**: want to show an area in one view that corresponds to a selection in another view
van Wijk Smooth Zooming

[van Wijk, 2003, M. Bostock]
van Wijk Smooth Zooming

[van Wijk, 2003, M. Bostock]
Networks
Networks

• Why not graphs?
  - Bar graph
  - Graphing functions in mathematics

• Network: nodes and edges connecting the nodes

• Formally, G = (V,E) is a set of nodes V and a set of edges E where each edge connects two nodes.

• Nodes == items, edges connect items

• Both nodes and edges may have attributes
Arrange Networks and Trees

- **Node–Link Diagrams**
  - Connection Marks
  - ![Networks and Trees](image)

- **Adjacency Matrix**
  - Derived Table
  - ![Networks and Trees](image)

- **Enclosure**
  - Containment Marks
  - ![Networks and Trees](image)

[Muñzner (ill. Maguire), 2014]
Molecule Network
Molecule Network

- Nodes may have attributes (e.g. element)
Molecule Network

- Nodes may have attributes (e.g. element)
- Edges may have attributes (e.g. number of bonds)
Web Sites as Graphs (amazon.com)

[M. Salathe, 2006]
Social Networks
Networks as Data

Nodes

<table>
<thead>
<tr>
<th>ID</th>
<th>Atom</th>
<th>Electrons</th>
<th>Protons</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Edges

<table>
<thead>
<tr>
<th>ID1</th>
<th>ID2</th>
<th>Bonds</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Node-Link Diagrams

- Data: nodes and edges
- Task: understand connectivity, paths, structure (topology)
- Encoding: nodes as point marks, connections as line marks
- Scalability: hundreds

...but high **density** of links can be problematic!

- Problem with the above encoding?
Arc Diagram

[D. Eppstein, 2013]
Network Layout

• Need to use spatial position when designing network visualizations
• Otherwise, nodes can **occlude** each other, links hard to distinguish
• How?
  - With bar charts, we could order using an attribute…
  - With networks, we want to be able to see connectivity and topology (not in the data usually)
• Possible metrics:
  - Edge crossings
  - Node overlaps
  - Total area
Force-Directed Layout

- Nodes push away from each other but edges are springs that pull them together
- Weakness: nondeterminism, algorithm may produce different results each time it runs

[M. Bostock, 2012]
sfdp

[Hu, 2005]
“Hairball”

[Hu, 2014]
Hierarchical Edge Bundling

Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the actual connections than when compared to the non-bundled versions (figures 2a and 11a). Bundled visualizations also show relations between sparsely connected systems more clearly (encircled regions); these are almost completely obscured in the non-bundled versions. The encircled regions highlight identical parts of the system for (a), (b), and figure 15.

Fig. 14. Using the bundling strength $\beta$ to provide a trade-off between low-level and high-level views of the adjacency relations. The value of $\beta$ increases from left-to-right; low values mainly provide low-level, node-to-node connectivity information, whereas high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

More specifically, most of the participants particularly valued the fact that relations between items at low levels of the hierarchy were automatically lifted to implicit relations between items at higher levels by means of bundles. This quickly gave them an impression of the high-level connectivity information while still being able to inspect the low-level relations that were responsible for the bundles by interactively manipulating the bundling strength. This is illustrated in figure 14, which shows visualizations using different values for the bundling strength $\beta$. Low values result in visualizations that mainly provide low-level, node-to-node connectivity information. High values result in visualizations that provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Another aspect that was commented on was how the bundles gave an impression of the hierarchical organization of the data as well, thereby strengthening the visualization of the hierarchy. More specifically, a thick bundle shows the presence of two elements at a fairly high level of the hierarchy, whereas the fanning out of a bundle shows the subdivision of an element into subelements.

Most participants preferred the radial layout over the balloon layout and the squarified treemap layout. Another finding was the fact that the rooted layout and the slice-and-dice treemap layout were considered less pleasing according to several participants. This is probably due to the large number of collinear nodes within these layouts, which causes bundles to overlap along the collinearity axes. This is illustrated in figure 17.

Although our main focus while developing hierarchical edge bundling was on the visualization itself, interaction is an important aspect in determining the usability of our technique. Based on our own insight and feedback gathered from participants, we contend that bundle-based interaction as described below could provide a convenient way of interacting with the visualizations.

Figure 16 shows how the bundling strength $\beta$ could be used in conjunction with interaction. [Holten, 2006]
Hierarchical Edge Bundling

Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the actual connections than when compared to the non-bundled versions (figures 2a and 11a). Bundled visualizations also show relations between sparsely connected systems more clearly (encircled regions); these are almost completely obscured in the non-bundled versions. The encircled regions highlight identical parts of the system for (a), (b), and figure 15.

Fig. 14. Using the bundling strength $\beta$ to provide a trade-off between low-level and high-level views of the adjacency relations. The value of $\beta$ increases from left-to-right; low values mainly provide low-level, node-to-node connectivity information, whereas high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

More specifically, most of the participants particularly valued the fact that relations between items at low levels of the hierarchy were automatically lifted to implicit relations between items at higher levels by means of bundles. This quickly gave them an impression of the high-level connectivity information while still being able to inspect the low-level relations that were responsible for the bundles by interactively manipulating the bundling strength.

This is illustrated in figure 14, which shows visualizations using different values for the bundling strength $\beta$. Low values result in visualizations that mainly provide low-level, node-to-node connectivity information. High values result in visualizations that provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Another aspect that was commented on was how the bundles gave an impression of the hierarchical organization of the data as well, thereby strengthening the visualization of the hierarchy. More specifically, a thick bundle shows the presence of two elements at a fairly high level of the hierarchy, whereas the fanning out of a bundle shows the subdivision of an element into subelements.

Most participants preferred the radial layout over the balloon layout and the squarified treemap layout. Another finding was the fact that the rooted layout and the slice-and-dice treemap layout were considered less pleasing according to several participants. This is probably due to the large number of collinear nodes within these layouts, which causes bundles to overlap along the collinearity axes. This is illustrated in figure 17.

Although our main focus while developing hierarchical edge bundling was on the visualization itself, interaction is an important aspect in determining the usability of our technique. Based on our own insight and feedback gathered from participants, we contend that bundle-based interaction as described below could provide a convenient way of interacting with the visualizations.

Figure 16 shows how the bundling strength $\beta$ could be used in con-
Hierarchical Edge Bundling

• Flexible and generic method
• Reduces visual clutter when dealing with large numbers of adjacency edges
• Provides an intuitive and continuous way to control the strength of bundling.
  - Low bundling strength mainly provides low-level, node-to-node connectivity information
  - High bundling strength provides high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes

[Holten, 2006]
Bundling Strength

\[ \beta = 0 \]

\[ \beta = 0.25 \]

\[ \beta = 0.5 \]

\[ \beta = 0.75 \]

\[ \beta = 1 \]

[Holten, 2006]
Adjacency Matrix

- Change network to tabular data and use a matrix representation
- Derived data: nodes are keys, edges are boolean values
- Task: lookup connections, find well-connected clusters
- Scalability: millions of edges
- Can encode **edge weight**, too
Cliques in Adjacency Matrices

a

b

[Gehlenborg and Wong]
Structures from Adjacency Matrices

cliques

bicliques

clusters
Node-Link or Adjacency Matrix?

- Empirical study: For most tasks, node-link is better for small graphs and adjacency better for large graphs
- Multi-link paths are hard with adjacency matrices
- Immediate connectivity or neighbors are ok, estimating size (nodes & edges also ok)
- People tend to be more familiar with node-link diagrams
- Link density is a problem with node-link but not with adjacency matrices
Trees

- Trees are directed acyclic graphs
  - each edge has a direction: the origin is the parent, the destination is the child
  - cannot get back to a node after leaving it
- A tree has a root (every other node hangs off it)
- Can consider enclosure in trees using parent-child relationships
Tree Visualizations

A variety of graphical representations are available for depicting tree structures (Figure 1), from "classical" node-link diagrams [23, 7], to file browsers such as Microsoft Explorer [26].

However, the effective use of space is an important attribute that is otherwise difficult to evaluate for these representations. Space-efficiency might be intuitively described in terms of how well a single figure fits into the screen, how much "useful" area is contained by a representation, how much space it allocates to nodes, or how much space it allocates to labels. There is no "correct" metric or combination of metrics that will apply to all the forms in Figure 1. Consequently, it is not clear initially if treemaps, or any other representation, will still be optimal in terms of space-efficiency.

This article identifies several metrics related to space-efficiency, quantifying the distribution of area across nodes in a tree representation, and that can be applied to the large variety of tree representations and that enables a fair comparison of them. Space-efficiency might be described in terms of area, aspect ratio, label size, or other measures. However, there is more to space-efficiency than total area. For example, concentric circles and nested circles (Figures 1E and 1F) both have a total area of $\pi \times \frac{1}{4} \approx 0.785$ (the area of a circle of diameter 1 square, both having a total area of 1, but also because they allow for the smallest nodes (i.e. the leaf nodes) in the representation, in addition to being equally efficient according to the metric of total area.

Treemaps are often described as optimally space-efficient, not just because they have a total area of 1, but also because they allow for the allocation of more or less area, depending on some attribute such as file size. Several other representations allocate space in a weighted manner, suggesting that a more space-efficient (in terms of label size) representation might be possible.

Furthermore, although a weighted partitioning is useful for showing the relative sizes, an alternative approach would be to give equal weight to the relative sizes of nodes in Figures 2A and 2C, an unfortunate side effect is that labels on small nodes are very difficult to read. If users are interested in augmenting the display to show "size" attributes, the labels in Figures 2B and 2D are clearly preferable in terms of legibility, labels, suggesting that a more space-efficient (in terms of label size) representation might be possible.

Figures 2 shows that icicle diagrams also allow for a weighted partitioning of area, and incidentally have no need for margins between the borders of nodes as treemaps often do. Nevertheless, it is not clear how to go about evaluating space-efficiency in a way that can be applied to all the forms in Figure 1.

Table 1 presents some of the commonly accepted standard set of metrics for evaluating the space-efficiency of a tree representation. The table includes metrics such as mean area exponent as well as the area they allocate to nodes or labels. Our analysis inspires a set of design guidelines as we consider the effects of alternative metrics on the space-efficiency of most of the basic tree representation styles in the information, and performs the first rigorous analysis and comparison of the space-efficiency of these representations.

The first step in evaluating space-efficiency is to define it in terms of the area that is contained by the representation. One basic metric of space-efficiency is the total area of a representation. Other metrics include the mean area exponent and the area that is allocated to nodes or labels. The table also includes two metrics that are used to evaluate the space-efficiency of the representations, and that are not necessarily based on the area of the representation. These metrics include the aspect ratio and the label size. The table shows that there are other metrics that can be applied to the representations, and that are not necessarily based on the area of the representation. These metrics include the mean area exponent and the area that is allocated to nodes or labels.

[McGuffin and Robert, 2010]