Data Visualization (DSC 530/CIS 568)

Network Visualization

Dr. David Koop
Map Projections

Central Meridian (selected by mapmaker)

Great distortion at high latitudes
Examples of two rhumb lines (direction true between any two points)
Equator touches cylinder if cylinder is tangent
Reasonably true shapes and distances within 15 degrees of Equator

[USGS Map Projections]
Map Projection Classification

Myriahedral projections

Dymaxion map

Goode's homolosine

equal area

conformal

no interrupts

Lambert cyl. eq. area

Plate carree

Mercator

Angle-preserving

[J. van Wijk, 2008]
Geographically-aligned Myriahedral Projections

In standard maps it is left to the viewer to guess where and which distortion occurs. Methodologically interesting is that here a computer science approach is used, whereas map projection is traditionally the domain of mathematicians, cartographers, and mathematical cartographers. Myriahedral projections are generated using algorithms, partially originating from flow visualisation, and not by formulas. Implementation is not simple, but when the machinery is set up, a very large variety of maps can be generated just by changing parameters, such as $W_l$, $W_w$, $F$, $f_0$, $s$, and the size of the faces used. This leaves much room for serendipity, and indeed, some of the maps shown here were discovered by accident.

Maps are not only used for navigation or visualisation, but also for decorative, illustrative and even rhetoric. Figure 12. Myriahedral projections with geography aligned meshes, 5500 polygons.

[J. van Wijk, 2008]
Adding Data to Maps

- Discrete: a value is associated with a specific position
  - Size
  - Color Hue
  - Charts

- Continuous: each spatial position has a value (fields)
  - Heatmap
  - Isolines
Discrete Categorical Attribute: Shape
Discrete Categorical Attribute: Shape
Continuous Attribute: Wave Height

Predicted tsunami wave heights

Tokyo

Epicenter
2:46 PM local time
12:46 AM ET

1 hour later
3 hours
6 hours
9 hours
12 hours
15 hours
18 hours
21 hours

FEET

0 1 2 3 4 5 6 7 7.8+
Isolines: Derived Geometry
Choropleth (Two Hues)

[M. Ericson, New York Times]
Glyphs: xkcd's Map

2016 ELECTION MAP
EACH FIGURE REPRESENTS 250,000 VOTES
TRUMP  CLINTON  OTHER
VOTES ARE DISTRIBUTED BY STATE AS ACCURATELY AS POSSIBLE WHILE KEEPING NATIONAL TOTALS CORRECT.
LOCATION WITHIN EACH STATE IS APPROXIMATE.
Cartograms

US Presidential Election 2016
Results mapped at county level showing the candidate with the largest vote share in each area

**Overall result:**
- **Trump**
  - 62,979,636 votes (46.1%)
  - 306 electoral votes
  - (received 304 in the Electoral College)
- **Clinton**
  - 65,844,610 votes (48.2%)
  - 232 electoral votes
  - (receives 270 in the Electoral College)
- **Other candidates**
  - 7,504,213 votes (5.7%)

Vote share of candidate with most votes

<table>
<thead>
<tr>
<th>0</th>
<th>50</th>
<th>70</th>
<th>90%</th>
</tr>
</thead>
</table>

**Reference map**

**Gridded population cartogram:** areas resized according to the total number of people living there (Alaska and Hawaii not included)

Map by Benjamin Hennig
www.voids-of-the-world.net

D. Koop, DSC 530, Spring 2019
Aggregation: 2016 Election by Precinct

[Interactive Version, NYTimes] [R. Rohla and Washington Post, 2018]
Maps Aren't Always Best: Close House Races

### 12 Lean Democratic
- AZ-02 Open (McSally)
- CA-49 Open (Issa)
- CO-06 Coffman
- IA-01 Blum
- KS-03 Yoder
- MI-11 Open (Trott)
- MN-02 Lewis
- MN-03 Paulsen
- NV-03 Open (Rosen)
- NJ-11 Open (Frelinghuysen)
- PA-07 Vacant (formerly Dent)
- VA-10 Comstock

### 31 Tossups
- CA-10 Denham
- CA-25 Knight
- CA-39 Open (Royce)
- CA-45 Walters
- CA-48 Rohrabacher
- FL-26 Curbelo
- FL-27 Open (Ros-Lehtinen)
- IL-06 Roskam
- IL-12 Bost
- IA-03 Young
- KS-02 Open (Jenkins)
- KY-06 Barr
- ME-02 Poliquin
- MI-08 Bishop
- MN-01 Open (Walz)
- MN-08 Open (Nolan)
- NJ-03 MacArthur
- NJ-07 Lance
- NM-02 Open (Pearce)
- NY-19 Faso
- NY-22 Tenney
- NC-09 Open (Pittenger)
- NC-13 Budd
- OH-01 Chabot
- PA-01 Fitzpatrick
- TX-07 Culberson
- TX-32 Sessions
- UT-04 Love
- VA-02 Taylor
- VA-07 Brat
- WA-08 Open (Reichert)

### 25 Lean Republican
- AR-02 Hill
- CA-50 Hunter
- FL-15 Open (Ross)
- FL-16 Buchanan
- GA-06 Handel
- GA-07 Woodall
- IL-13 Davis
- IL-14 Hultgren
- MO-02 Wagner
- MT-AL Gianforte
- NE-02 Bacon
- NY-24 Katko
- NY-27 Collins
- NC-02 Holding
- OH-12 Balderson
- PA-10 Perry
- PA-16 Kelly
- SC-01 Open (Sanford)
- TX-23 Hurd
- TX-31 Carter
- VA-05 Open (Garrett)
- WA-03 Herrera Beutler
- WA-05 McMorris Rodgers
- WV-03 Vacant (formerly Jenkins)
- WI-01 Open (Ryan)

[New York Times, 2018]
Assignment 3

- Due this Thursday (3/28)
- Geographic Visualization and Colormaps: D3 Map Example
- Uses data about affordable housing in Massachusetts
Maps: What trends do you see?

Number of Votes Cast

[Desaturated by D. Koop, M. Ericson, New York Times]
Don't Just Create Population Maps!

PET PEEVE #208:
GEOGRAPHIC PROFILE MAPS WHICH ARE BASICALLY JUST POPULATION MAPS
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 $\equiv$ items, edges connect items

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

Node–Link Diagrams
Connection Marks

Adjacency Matrix
Derived Table

Enclosure
Containment Marks

[Munzner (ill. Maguire), 2014]
Molecule Graph
Molecule Graph

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

- 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

[P. Butler, 2010]
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”

[JGD_Homology@cis-n4c6-b4. 26028 nodes, 100290 edges.]

[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 combination with dynamic interaction techniques.
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 other visual features.
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

[McGuffin and Robert, 2010]
Node-Link Diagram

- Trees are graphs
- …but we have more structure
- Horizontal or vertical
- Idea 1: partition space for each node via recursion
- Idea 2: “Tidy” Drawing
  - Wetherell & Shannon: Don’t waste space (overlapping parent nodes is ok)
  - Reingold and Tilford: Keep symmetry, subtrees look similar

[WS Alg., Reingold and Tilford, 1981]
Reingold-Tilford Algorithm

- Recurse on left and right subtrees
- Shift subtree over as long as it doesn’t overlap
- Place parent centered above the subtrees
- Originally, only binary trees, extended by Walker

[Reingold and Tilford, 1981]
Icicle Plot

- Line marks
- Vertical position shows depth
- Horizontal position shows links and sibling order
- Scalability: 1 pixel leaves, but harder to label

[Bostock, 2011]
Radial Node-Link

- Use polar coordinates instead of rectilinear
- Same layout algorithms work (e.g. Reingold-Tilford)
- Benefit: space usage, labels
Sunburst

• Icicle plot in a radial layout
• Reading labels?
• Intuitive navigation
Indented Outline

- Like a filesystem tree
- Use horizontal position to show depth, vertical positions show sibling/order