Data Visualization (CIS/DSC 468)

Networks & Trees

Dr. David Koop
Network
Network

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

• Nodes may have attributes (e.g. element)
• Edges may have attributes (e.g. number of bonds)
Arc Diagram

[D. Eppstein, 2013]
Arrange Networks and Trees

Arrange Networks and Trees

- **Node–Link Diagrams**
  - Connection Marks
  - [ ] NETWORKS  [ ] TREES

- **Adjacency Matrix**
  - Derived Table
  - [ ] NETWORKS  [ ] TREES

- **Enclosure**
  - Containment Marks
  - [ ] NETWORKS  [ ] TREES

[Munzner (ill. Maguire), 2014]
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 difference results each time it runs

[Force-Directed Layout, M. Bostock, 2012]
Assignment 2


Notes:
- Be careful about what `selectAll` is doing
- Check what data is included in a nested array
- `d3.line` is a function, and `x` and `y` are accessor functions
Exam 1

• Wednesday, March 1 in class (12-12:50pm)
• Use the restroom before the exam
• Format:
  - Multiple Choice
  - Short Answer
• Sample questions on web site:
Scalable Force-Directed Placement (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 other visualization techniques to provide a more interactive experience. For example, by dynamically adjusting the bundling strength, users can explore the hierarchy at different levels of detail, from low-level connectivity to high-level summaries.

[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 conjunction with...
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

Figure 7.5: Comparing matrix and node-link views of a five-node network.

- Matrix view.
- Node-link view. From [Henry et al. 07], Figure 3b and 3a.

Matrix views of networks can achieve very high information density, up to a limit of one thousand nodes and one million edges, just like cluster heatmaps and all other matrix views that uses small area marks.

7.1.3.3 Multiple Keys: Partition and Subdivide

When a dataset has only one key, then it is straightforward to use that key to separate into one region.

[Henry et al., 2007]
Cliques in Adjacency Matrices

[Gehlenborg and Wong]
Structures from Adjacency Matrices
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. Node-link diagrams
B. Tree maps
C. Nested circles
D. Concentric circles
E. Hierarchical circle diagrams
F. Circular treemaps
G. Rectangular treemaps
H. File browsers such as Microsoft Explorer.

This article identifies several metrics related to space-efficiency, including the use of a metric of the size of the node, the aspect ratio of the node, the label size, or other measures. However, there is more to space-efficiency than total area. Experience suggests that the representations within each of these pairs do not scale equally well with larger, deeper trees. This might be because they have a total area of 1, but also because they allow for partitioning of area, and incidentally have no need for margins between the nodes. This is indeed desirable properties, however they are not unique to treemaps.

Quantifying the Space-Efficiency

1. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

2. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

3. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

4. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

5. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

6. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

7. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

8. The nodes, measuring how much "useful" area they contain; and (3) which implicitly takes into account both the size and aspect ratio of the nodes, as treemaps often do.

[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

[Bostock, 2012]
Sunburst

- Icicle plot in a radial layout
- Reading labels?
- Intuitive navigation

[Heer et al., 2012]