Hierarchies and Trees 1
(Mostly Node-link)

CS 4460 – Intro. to Information Visualization
October 30, 2017
John Stasko

Learning Objectives

- Define hierarchical data & related terms
  - root, node, link, leaf, depth, parent, child, sibling
- List example tasks for hierarchical data
- Be able to draw reasonable 2D tree
  - Understand basic approach/algorithm and method
- Describe particular drawing techniques and explain +/- of each
  - SpaceTree, Cone Tree, Hyperbolic tree
- Explain general limitations of node-link approach
- Understand treemap algorithm
  - Be able to draw slice-and-dice treemap given a hierarchy
Hierarchies

- Definition
  - Data repository in which cases are related to subcases
  - Can be thought of as imposing an ordering in which cases are parents or ancestors of other cases

Hierarchies in the World

- Pervasive
  - Family histories, ancestries
  - File/directory systems on computers
  - Organization charts
  - Animal kingdom: Phylum,..., genus,...
  - Object-oriented software classes
  - ...
Analysis Tasks

- Example tasks?
  - Describe/understand structure
  - Find items
  - What are the parent/children/siblings of $x$?
  - Where is this subtree?
  - Where are nodes with particular values located?
  - What kind of attributes does this subtree have?
  - ...

Activity

Draw a representation for the following hierarchy:
Node: Child1, Child2, ... (order means nothing)

A: J, H, U, F
J: E, P
H: D, R, L, W, B
F: S, M, N
E: T, K
P: V, C, O, I
S: Q, G
Trees

- Hierarchies often represented as trees
  - Directed, acyclic graph

- Two main representation schemes
  - Node-link
  - Space-filling

Node-Link Diagrams

- Root at top, leaves at bottom is very common
Sample Representation

From: Johnson & Shneiderman, '91

Examples

Good for?
Search
Bad for?
Understanding structure
Why Put Root at Top?

Root can be at center with levels growing outward too

Can any node be the root?

http://kimalbrecht.com/viztypo/?p=27
Even Georgia Does It!

Basic Algorithm

- Recursive algorithm
- Height on separate levels
- Width in unique columns
- Make room for subtrees upwards
Potential Problems

• For top-down, width of fan-out uses up horizontal real estate very quickly
  – At level n, there are $2^n$ nodes

• Tree might grow a lot along one particular branch
  – Hard to draw it well in view without knowing how it will branch

More Sophisticated

In what way?
• Regions compressed horizontally
Reingold-Tilford Algorithm

Compact layout
Uses symmetry
Depth on levels

Generalized from binary trees by Walker
Running time improved (linear) by Buchheim et al

Scale

- Real challenge – Get hundreds or thousands of nodes

- Approaches?
  - Interaction
  - Only show some items or attributes
  - Clustering & aggregation
  - Smart layout
SpaceTree

- Uses conventional 2D layout techniques with some clever additions

Grosjean, Plaisant, Bederson
InfoVis '02

Video
Characteristics

- Vertical or horizontal
- Subtrees are triangles
  - Size indicates depth
  - Shading indicates number of nodes inside
- Navigate by clicking on nodes
  - Strongly restrict zooming

Design Features

- Make labels readable
- Maximize number of levels opened
- Decompose tree animation
- Use landmarks
- Use overview and dynamic filtering
3D Approaches

- Add a third dimension into which layout can go
- Compromise of top-down and centered techniques mentioned earlier
- Children of a node are laid out in a cylinder “below” the parent
  - Siblings live in one of the 2D planes

Cone Trees

Developed at Xerox PARC

3D views of hierarchies such as file systems

Robertson, Mackinlay, Card
CHI ’91
Cone Trees

- **Pros**
  - More effective area to lay out tree
  - Use of smooth animation to help person track updates
  - Aesthetically pleasing

- **Cons**
  - As in all 3D, occlusion obscures some nodes
  - Non-trivial to implement and requires some graphics horsepower

Hyperbolic Browser

- **Focus+Context technique**
- **Approach:** Lay out the hierarchy on the hyperbolic plane and map this plane onto a display region
- **Use interaction & animation to move focus to different nodes**

**Video**
Demo from Prefuse system

Lamping and Rao
*JVLC* '96
Key Attributes

- Natural magnification (fisheye) in center
- Layout depends only on 2-3 generations from current node
- Smooth animation for change in focus
- Don’t draw objects when far enough from root (simplify rendering)

Problems

- What might be problems with this approach?
**Problems**

- **Orientation**
  - Watching the view can be disorienting
  - When a node is moved, its children don’t keep their relative orientation to it as in Euclidean plane, they rotate
  - Not as symmetric and regular as Euclidean techniques, two important attributes in aesthetics

**Node-link Shortcoming?**

- What if we want to portray more variables of each case?
  - Difficult to encode more variables of data cases (nodes)
    - Shape
    - Color
    - Size
  - ...but all quickly clash with basic node-link structure
Space-Filling Representation

Each item occupies an area

Children are “contained” under parent

One example: “Icicle plot”

Icicle Plot

http://blog.orgvue.com/ideal-span-control-see-intervene/
**Treemap**

- Space-filling representation developed by Shneiderman and Johnson, Vis '91
- Children are drawn inside their parent
- Alternate horizontal and vertical slicing at each successive level
- Use area to encode other variable of data items

**Example**

![Example Diagram]
Example

---

Example

---

Fall 2017

CS 4460

Fall 2017

CS 4460
Treemap

- Example

Directories

Treemap Example

Old one I built for our Sun workstations
Treemap Algorithm

Draw()
{
  Change orientation from parent (horiz/vert)
  Read all files and directories at this level
  Make rectangle for each, scaled to size
  Draw rectangles using appropriate size and color
  For each directory
    Make recursive call using its rectangle as focus
}
Learning Objectives

- Define hierarchical data & related terms
  - root, node, link, leaf, depth, parent, child, sibling
- List example tasks for hierarchical data
- Be able to draw reasonable 2D tree
  - Understand basic approach/algorithim and method
- Describe particular drawing techniques and explain +/- of each
  - SpaceTree, Cone Tree, Hyperbolic tree
- Explain general limitations of node-link approach
- Understand treemap algorithm
  - Be able to draw slice-and-dice treemap given a hierarchy

Upcoming

- Hierarchies 2 – More on Space-filling reps
  - Prep: Johnson & Shneiderman article

- Lab 7 – D3 Animation & transition 1