VISUALIZING HIERARCHICAL DATA

Graham Wills
SPSS Inc., http://willsfamily.org/gwills

SYNONYMS

Hierarchical Graph Layout, Visualizing Trees, Tree Drawing, Information Visualization on Hierarchies; Hierarchical Visualization; Multi-Level Visualization

DEFINITION

Hierarchical data is data that can be arranged in the form of a tree. Each item of data defines a node in the tree, and each node may have a collection of other nodes as *child nodes*. The relationship between the parent nodes and the child nodes forms a *tree network*. The formal definition of a tree is that the graph formed by the nodes and edges (defined between parent and child node) is both connected and contains no cycles. The following properties of a tree are of more practical use from the point of view of displaying visualizations:

- One node, called the *root* node, has no parent.
- All other nodes have exactly one parent.
- Nodes with no children are termed *leaf* nodes. Nodes with children are termed *interior* nodes.
- For all nodes in a tree, there is a single unique path up the tree going from parent to parent's parent and so on, which will terminate in the *root* node.
- The number of nodes on the path from a node to the root is termed its *depth*.

Although not strictly required, the vast majority of hierarchical data, and the main application area, consists of trees where the parent nodes define some form of aggregation on the child nodes, so that the data for the parent is equal to, or is expected to be close to, the aggregation of the data for that node's children. We often term the relationship between parent and child in terms of inclusion; it is common to state that nodes *contain* their children. This aspect is often brought out in visualizations.

A simple example of hierarchical data would consist of populations for the world, hierarchically broken down into sub-regions. At the top level, the root would consist of the world, with data being the population of the world. The next level could be the four main continents, each with their individual population, and each continent would have countries as children, each with their population counts. In each case, we would expect that the population of the parent node would roughly equal the population of the child nodes. If the data were collected at slightly different times, or from different sources, the populations for the continents might not exactly equal the sum of their children's populations, but we would expect it to be very close.

Note that in this example, every *leaf* node has the same *depth*. Although this is a common property in many applications, it is not a required property. In fact, if we count Antarctica

as a continent, it would have no contained countries, and the hierarchy would lose this property. If we further increased the level of the hierarchy, dividing countries into regions, we would have very different levels. In the United States, we might divide into states and then into zip codes, whereas for the Vatican City, we would not divide up at all.

HISTORICAL BACKGROUND

Leonhard Euler is regarded as the founder of graph theory, publishing initially in the 1730s, but displays of trees as predate even this, especially displays of family trees, some of which survive from significantly previously. However, the discipline of visualizing hierarchical data in a systematic fashion dates back only to the availability of computers. In 1992, the conference *International Symposium on Graph Drawing* commenced meeting and their conference proceedings, available from 1994, provide an excellent overall reference to the state of the art in this subject, although limited mainly to static graphs. From around 1990 onwards, increased attention has been paid to *interactive*, or *dynamic* visualization of hierarchical data, although information on such techniques is scattered across many disciplines. User interface controls for interacting with hierarchies became common with the advent of windowed operating systems; trees of folders and files are commonplace and techniques for filtering and pruning such views have seen significant research and user testing.

SCIENTIFIC FUNDAMENTALS

There are a number of reasons why hierarchical data might be visualized, and it is important to identify the goal of any visualization, as different visualization techniques serve different goals. Common reasons to view hierarchies are:

- *Understanding Structure*: The goal is to understand the structure of the hierarchy; in our world population example, we might want to know how countries are distributed within continents, or see how many major regions countries have.
- *Understanding Data*: The goal is to understand the distribution of data across the hierarchy. In the example, we might want to know if each continent has a similar distribution of population, or if the lowest levels have similar populations (do zip codes in the US, on average, have the same populations as the Vatican City?), or similar questions. Often, the goal is to understand the data within the context of the structure.
- *Summarizing large amounts of data*: The goal is to reduce information overload by providing a summary of low-level data into aggregated data. A hierarchy allows the user of a visualization to set the level of detail they want, by only showing data up to a certain level in the hierarchy.

There are two basic branches of visualization techniques for hierarchies. The first is based on a node-edge graph-layout approach which focuses attention on the structure and relationships, and the second on space-filling approaches, which focus attention on the relative sizes of nodes in the hierarchy. Each is discussed below.

Node-Edge Layouts

These displays are essentially a specialization of general graph layout techniques, as described in the encyclopedia entry “Visualizing Network Data”. Each node in the hierarchy is displayed as a small glyph, commonly a circle or a square. Data on the node can be represented by changing aesthetic attributes of the node such as its size, color, pattern, etc. Figure 1 below gives an example of a traditional node-edge display for hierarchical data. Each node represents a country, and they are aggregated into a hierarchy where the net level up is a sociological grouping. The populations have been aggregated by a mean average, so the node representing the “new world” countries has been given the mean population of countries in the new world. It is clear that there is little difference between populations of countries when they are aggregated into these groupings.

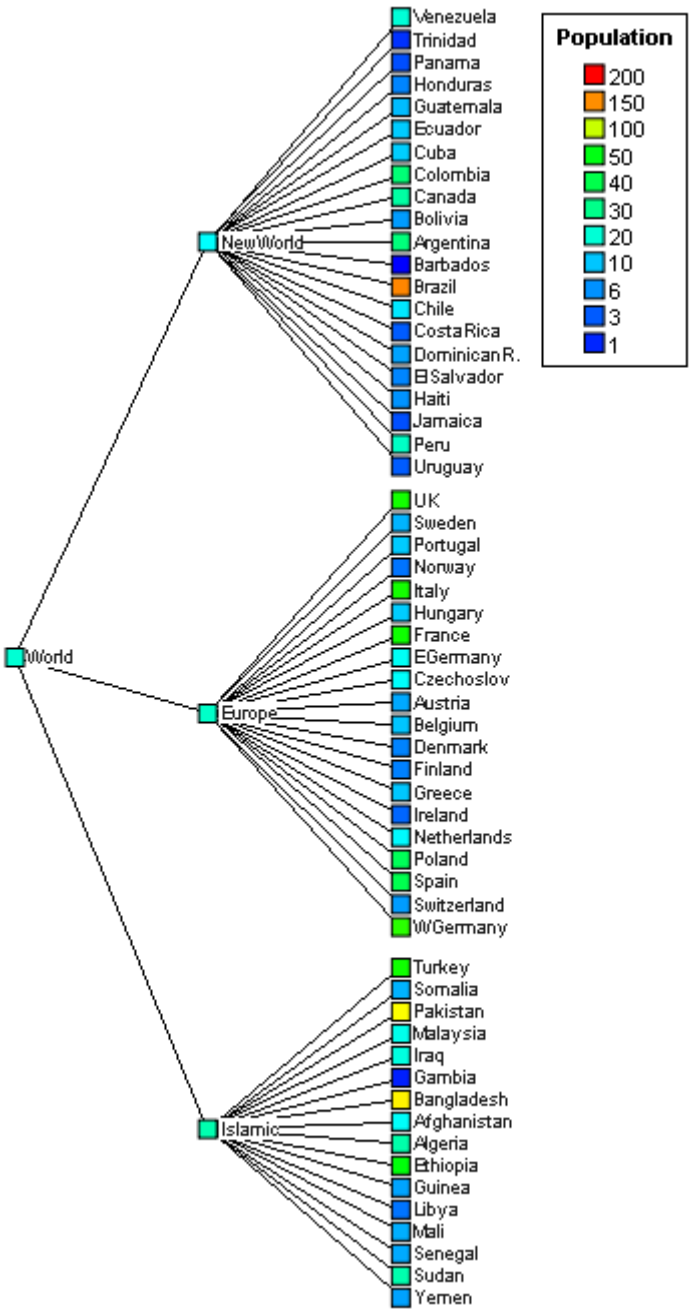

Figure 1. Standard Node-Edge layout for a hierarchical network.

This visualization has the advantage that it is good for showing the *structure* of the hierarchy – we can see clearly how each many countries are in each group, and the distribution of the variable of interest. However, the display is not compact – we waste a lot of space showing links which have little information. Modifications to this basic display include the following:

- *Ordering the nodes within each parent*. We could sort the child nodes for each parent by a variable. In figure 1, it would make sense to sort by the population to

create a better overview of the distribution of that variable across countries within a group.

- *Displaying information on the edges.* In the above figure, we could code the edges according to the similarity between the country and its group, using any of a number of statistical techniques to define such similarity. One such technique starts simply with data items and then generates a hierarchy by successively clustering items into groups based on similarity. This technique is called hierarchical clustering and is often visualized using a dendrogram as in figure 2. The dendrogram shows when groups are formed using the vertical dimension; the lower on the vertical dimension, the more similar the groups that were merged were. In figure 2, F and G were merged first and because they care the most similar pair, then A and B were formed into a group, then D and E. Then C was added to {A, B}, following which {D, E} was merged with {F, G}. The resulting groups {A, B, C}, {D, E, F, G}, {H} are only merged together at a much higher level of dissimilarity. The dendrogram therefore lets us see not only what clusters were created, but it also, by placing the merge information at a location in the vertical dimension proportional to the similarity of the groups being merged, allows us to see how good the resulting clusters are.

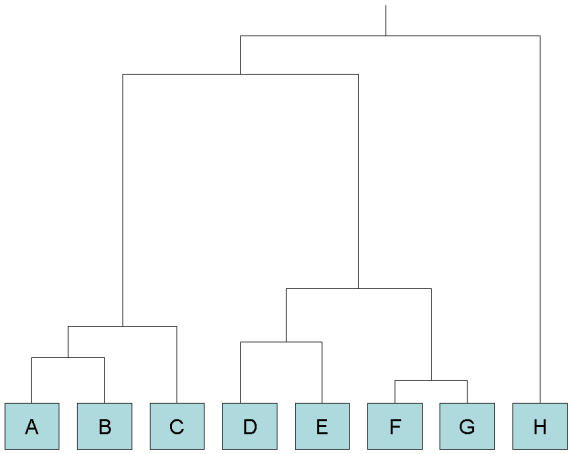

Figure 2. Dendrogram of a hierarchical clustering

Space-Filling Layouts

In contrast to node-edge layouts, space-filling layouts take explicit advantage of the hierarchical nature of data directly. A space-filling layout is usable only when the main variable of interest is *summable*. Because these techniques lay out areas in proportion to sizes, and parents visually include their children, a variable that can be summed is necessary. It is not possible to use space-filling layouts directly to show means, minimums or similar statistics. The base layout is limited to sums. It is, of course, possible to color or use some other non-size aesthetic for such techniques, but then the essential space-filling nature of the display is of little value.

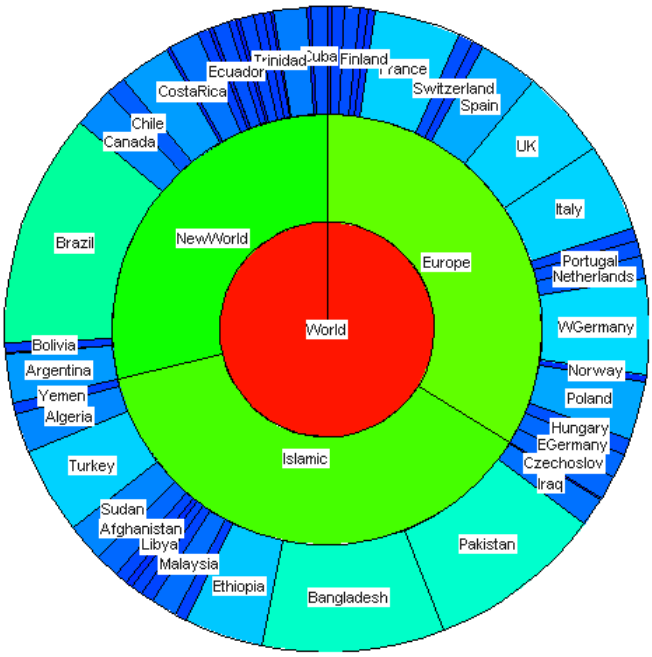

Figure 3. The Population data of figure 1, this time showing summed population, using a space-filling radial layout.

Figure 3 shows the example data with a radial space-filling layout. Each level in the hierarchy is represented by a band at a set radius, with the root note in the center, and children outside their parents. The angle subtended by a node is proportional to the percentage of the entire population that this node represents. Children are laid out directly outside their parents, so parents "divide up" the space for their children according to the child sizes.. This is a typical space-filling technique, many of which have been discovered in various disciplines. They share the following characteristics:

- The root node occupies 100% of the space, or dimension of interest (in this case, the radial dimension).
- Each node partitions its space according to the relative sizes of its children.

Figure 3 uses space more economically than figure 1, and also allows us to see sizes more clearly. It is generally to be preferred if the data support it. Compare this figure with figure 4, which uses a style of layout popularized under the name "TreeMap".

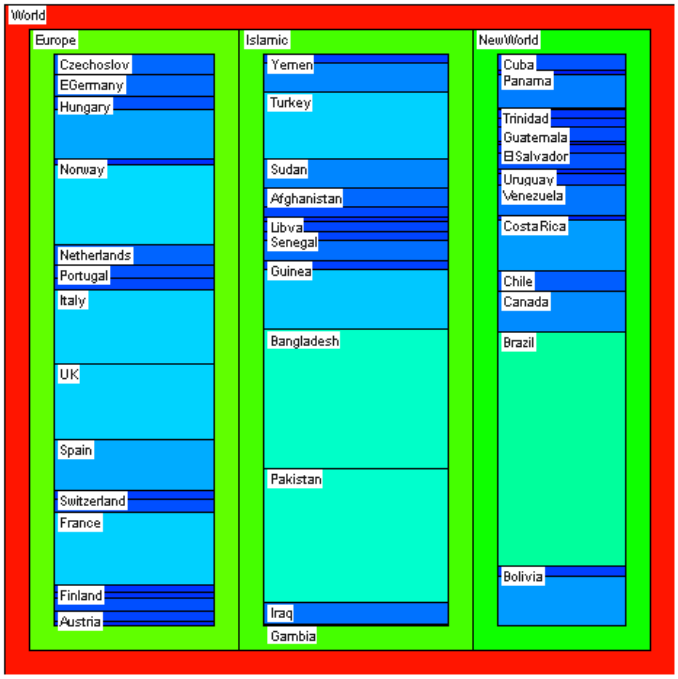

Figure 4. TreeMap version of figure 3.

The treemap, instead of allocating space for child nodes outside the parent node, lays them out directly on top of the parent, thus making a maximally compact representation. This causes the immediate problem that there is then no way to distinguish the tree structure, since the children completely occlude the parents. Typically, as in figure 4, a small border is left around the child nodes to allow the tree structure to be ascertained, although that does distort the overall relationship between visual size and the sizes of the hierarchical data points. The TreeMap is not a good technique for learning about *structure* because of this issue, but because it is very compact it can perform quite well in the domain to which it is applicable: displaying relative sizes of items in the hierarchy when the structure of the hierarchy is well-known or of little interest. There are numerous different ways to render this figure, and care should be taken to avoid situations where some rectangles are skinny and others are flat; it is harder to make size judgments based on areas with widely differing aspect ratios than it is on arcs with different angles as in figure 3, for example. In general, the TreeMap should be treated as a specialist tool for hierarchies to which it is applicable, not as a general purpose hierarchical visualization.

Interactive Visualization of Hierarchical Data.

The entry on Visualization of Network Data indicated some techniques for interacting with general networks. These can be applied to the special case of hierarchies, and of these the most applicable and important technique is that of brushing and linking.

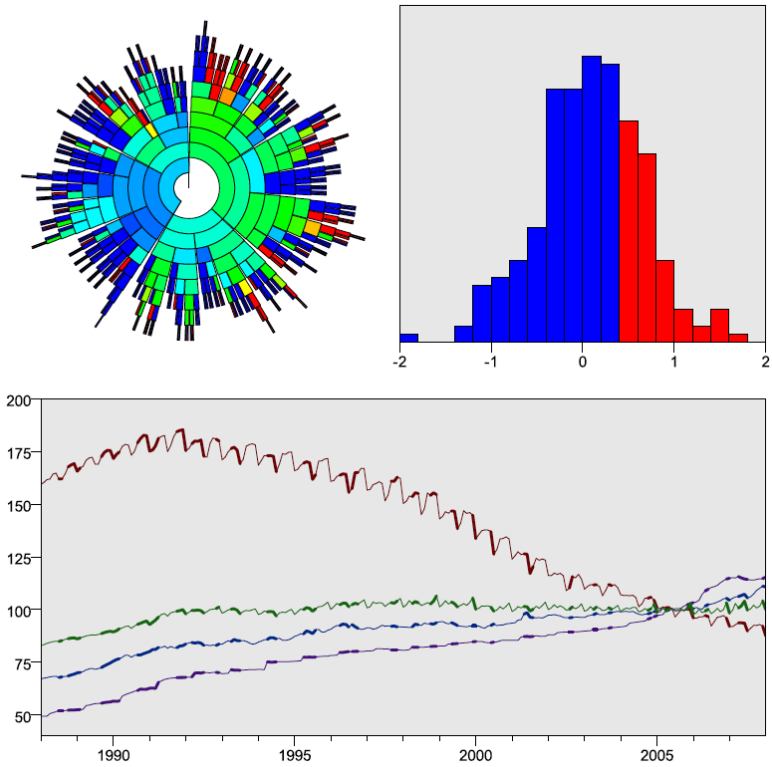

Figure 5. Linked Views of UK CPI data.

Figure 5 demonstrates a number of techniques that have been detailed earlier, and adds interaction to the visualization. The base data are a set of consumer price indices (CPI) from the United Kingdom, collected monthly. A hierarchical clustering has been performed on the months, based on all the CPIs in the data set except the overall CPI. At the bottom is multiple time series chart showing four indices (food, clothing, housing, furniture). At the top right is a histogram showing residuals from a fitted time series mode of the overall CPI. Each view is linked to each other view, so that selecting a region of one chart highlights the corresponding months in all other views.

In this figure, the higher residuals have been selected, denoting months where the actual CPI was higher than the model predicted. This is shown in red in the histogram, and in the time series view the segments corresponding to those months are shown with a thicker stroke. It appears that these residuals occur mainly on the downward part of regular cycles on the topmost time series (representing the clothing CPI).

In the hierarchical view, a generalization ahs been made to the linking. Because the months are aggregated at higher levels of the hierarchy, these interior nodes in the hierarchical tree map may be partially selected. In this figure the selection percentage is shown with a rainbow hue map, with blue indicating completely unselected, red completely selected, and green half-selected. Intermediate hues indicate intermediate selection percentages. At the outer levels some fully selected clusters can be seen, but more interestingly, looking at the inner bands, there are strong differences in color, indicating some relationship between the distributions of the residuals for the overall CPI,

and the distribution of clusters of the other CPIs. This indicates some form of inter-dependence between them should be investigated to improve the model.

KEY APPLICATIONS

Hierarchies are a very common data structure. Application areas include geographical data, which are invariably organized in hierarchies, data on organizations, such as businesses, military organizations, and political entities. Financial data is also often suitable. As well as *a priori* hierarchies, it is very common to generate hierarchies so as to aggregate data and allow higher level information to be viewed. Database systems like OLAP and other roll-up techniques can create hierarchies and allow users to move rapidly between levels to investigate data.

DATA SETS

UK figures for CPI were retrieved from http://www.statistics.gov.uk/cpi/, and similar statistics should be generally available from most countries National Statistics office.

The world population data are a subset taken from the CIA's world fact book, available at https://www.cia.gov/library/publications/the-world-factbook/

CROSS REFERENCES

Visualizing Network Data. OLAP

RECOMMENDED READING

Di Battista, G., Eades, P., Tamassia, R. and Tollis, I.. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999

Friendly, M. *The Gallery of Data Visualization*. http://www.math.yorku.ca/SCS/Gallery/. (This site contains a wealth of examples, including network and tree displays, together with some excellent examples of how not to visualize data)