VISUALIZING NETWORK DATA

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

SYNONYMS

Graph Layout, Network Topology, Graph Drawing, Information Visualization on Networks

DEFINITION

A network is a set of nodes with edges connecting the nodes. When the graph defined by that set of nodes and edges has other associated data, the result is termed "network data". Visualizing Network Data is the process of presenting a visual form of that structure so as allow insight and understanding.

HISTORICAL BACKGROUND

Although examples of informal drawing of networks and hand-drawn graph layouts can be found stretching back many decades, the discipline of visualizing network data in a systematic fashion dates back only to the availability of computers, with an early paper by Tutte in 1963 titled "How to Draw a Graph" being a prime example. 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. From around 1990 onwards, increased attention has been paid to *interactive*, or *dynamic* visualization of Network Data. These techniques have surfaced in a variety of different fields, including Statistical Graphics, Information Visualization, and many applied areas.

SCIENTIFIC FUNDAMENTALS

The basic goal behind a successful static visualization of network data is simply stated: Producing a layout of nodes and edges with a bounded 2-dimensional (or, less commonly, 3-dimensional) region such that the resulting display portrays the structure of the network as clearly as possible. To achieve this goal the following topics must be addressed:

- Techniques for representing the nodes and edges.
- Aesthetic criteria that define what is meant by a clear representation.
- Layout Techniques.
- Extensions to layout techniques to incorporate data on nodes and edges.
- Interactive techniques to augment static layouts.

Node and Edge Representation

The simplest form of display for a node is as a small glyph, commonly a circle or a square. It is simple and compact. Further, it is easy to add extra information to, such as color, pattern, label or other aesthetics. These can represent data on the nodes. The first three example layouts figure 1 below show nodes as circles. The last layout is different; it uses an extended representation to display a node, allowing the edges to be displayed as vertical lines. This representation is most common when the graph is *tree-like*, or more generally is a *hierarchical* network. When there is special known structure for a graph like this, it is possible to use representations that help elucidate such properties. For general network data, however, simple nodes are the most common and suitable choice.

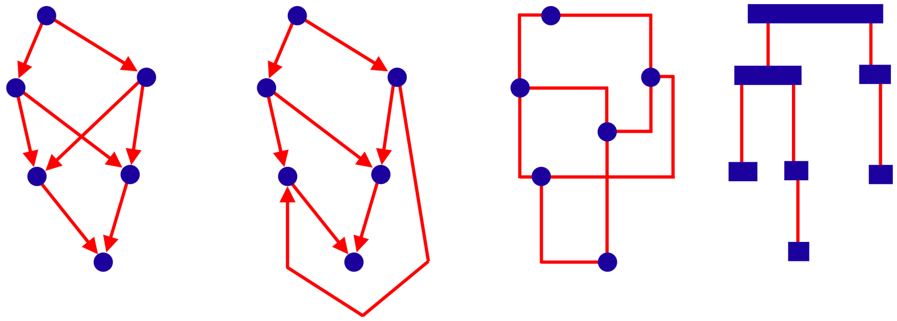

Figure 1. Representations: (a) Straight-Edge, (b) Polyline, (c) Orthogonal, (d) Hierarchical

Edge representation allows more freedom. If the edges are *directed*, so that they have a definite 'from' and 'to' node, then they are usually portrayed with arrows at the ends, unless the direction is obvious from the layout. Other common representational techniques are:

- **Straight Edge**. Figures 1(a) and 1(d) shows edges as straight lines directly linking nodes. This has the advantage of being simple and making connections clear, but tends to result in more edge crossings.
- **Polyline / Paths**. Using paths instead of straight edges allows us to reduce the number of edge crossings in figure 1(c), but the resulting display is more complex.
- **Orthogonal Paths.** A style of representation where paths consist of orthogonal lines. This form of representation harkens back to printed circuit board layouts, an early application of graph layout.

Quality Criteria

The question of what makes a good layout has been approached by many authorities. In practice trade-offs must be made, and specific applications stress some criteria more than others. Those criteria that are most often considered important are given below:

- Minimize edge crossings.
- Minimize the area needed to display.
- Maximize the symmetry of the layout, both globally and locally.
- Minimize number of bends in polyline layouts.
- Maximize the angle between edges, both at nodes and when they cross.
- Minimize total path lengths.

There are also criteria that are suitable for specific graph types. For example, in a graph that is *directed* and *acyclic* (there is no path from nodes following edges that loops back on itself), one strong criterion is that the edges generally head in the same direction (all upward, or all downward, as in figure 1(a). Note that figure 1(b) does not exhibit this quality, an example of the trade-offs made when considering different layout algorithms.

Layout Techniques
Layout techniques for general networks employ a variety of techniques, the most common of which are:

Planar Embeddings. A planar graph is one that has a 2-dimensional representation that has no edge crossings. It is also possible to represent any planar graph using only straight-line edges. Although testing and subsequent layout can be done in linear time, the algorithms for doing so are complex. When each vertex has at most four edges, orthogonal representations are possible, but minimizing the number of bends is *NP-hard*, and approximate algorithms are employed in practice.

Planarization. If a graph is not planar, it can be made planar by adding 'fake' nodes at the crossing points. One such technique would be to find the maximal planar subgraph and laying it out, then adding the additional edges and inserting the additional nodes at crossing points. The resulting graph is laid out using straight edges and then the fake nodes removed, leaving polylines connecting some remaining nodes.

Directed Embeddings. A similar technique is to extract a directed graph from the overall graph (by orienting the edges if necessary) and use a hierarchical drawing technique to place the nodes. A simple example would be finding a minimum spanning tree within the graph and laying it out directly.

Force-directed. Using a mixture of aesthetic criteria, the "energy" of a layout can be defined where low energy corresponds to a good layout The resulting energy function can be minimized using techniques including randomization, simulated annealing, steepest descent, and simulation. Force-directed techniques are very general, and often can be implemented using iterative algorithms, which makes them amenable to a choice of stopping criteria that can deal with large networks more easily.

Layouts for Networks with Data
If, in addition to the network connections, we also have data on either or both of the nodes or the edges, we can use standard *Information Visualization* techniques to augment the network visualizations. The simplest augmentation is to map a variable for nodes or edges onto an aesthetic, such as color, size, shape, pattern, transparency or dashing. The basic act of labeling nodes is already an example of this use of aesthetics to convey information. In figure 2 we show an application to understanding correlation patterns between variables in a data set consisting of information on Major League Baseball players in 2004. Only correlations passing a statistical test of adequacy have been retained, resulting in a non-connected graph. The edges contain data on two measures of

association between then variables connected by the edge, which have been mapped to aesthetics as follows:

- *Color*: The color represents the statistical significance of the association, with green being weak and red being strong.
- *Size*: The width of the edges indicates how strong the association is in the sense of how much of the variation with one variable can be explained by the other.

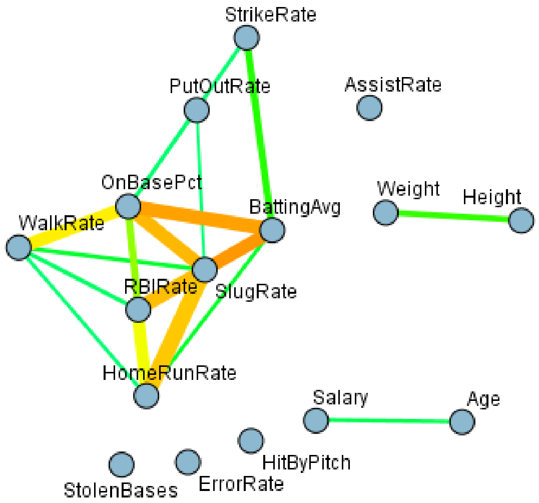

Figure 2. Correlations between Variables; baseball player data, 2004

The overall layout has a straight-edge representation, and has been arrived at via a force-directed algorithm.

When the data are similar to the above, where we have a measure (or, in this case, two measures) of an edge's strength, the algorithms used should be modified so as to take the strength into account. Although this is possible for all algorithms, it is most simple to implement using force-directed techniques. The goal is to ensure that nodes that are highly associated with each other are close together, and that leads to the criterion that path length should be inversely proportional to edge weight. Reviewing the quality criteria above, some of the criteria can be modified for weighted networks as follows:

- Minimize the weighted summed deviation between path lengths and inverse edge weights.
- Penalize edge crossings with crossings involving strong edges penalized more than weak edges.
- Maximize the angle between edges, both at nodes and when they cross, penalizing angles involving strong edges more than weak edges.

The layout of figure 2 was achieved by a force-directed algorithm under these constraints. The measures of association were derived by *Scagnostic* algorithms of Wilkinson and Wills.

Interactive Augmentations

For small and medium-sized static networks, static layouts are adequate, but for larger data sets and for time-varying data, different techniques are required. These can roughly be divided into the following categories:

Distortion techniques. Suitable for large but not huge networks, distortion techniques allow users to place a focus point on a region of interest of the display, and the display redraws so as to magnify the area of interest and de-magnify the rest of the display. Since usable magnification levels can go no higher than a factor of about 5, this technique can improve the number of nodes that can be *visualized* by a maximum of about 25. Specific versions of these techniques include *fisheye* and *hyperbolic* transformations. In Figures 3 and 4 below we show a network showing associations between words in Melville's *Moby Dick*. The size of the nodes indicates word frequencies, and the links color indicates strength of association. The cluster at the top left is cluttered in figure 3, so we apply an interactive fisheye and drag the focus point to that cluster, allowing us to see that cluster in more detail, as shown in figure 4.

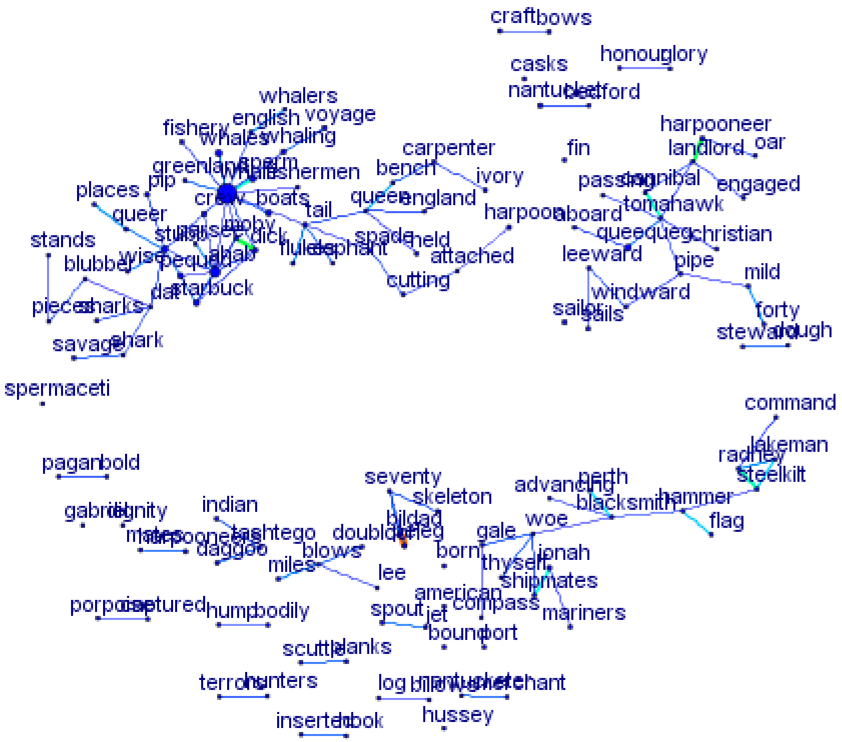

Figure 3. Force-directed layout of important words in *Moby Dick*. Node sizes indicate word frequencies. Edges link words are commonly found close to each other.

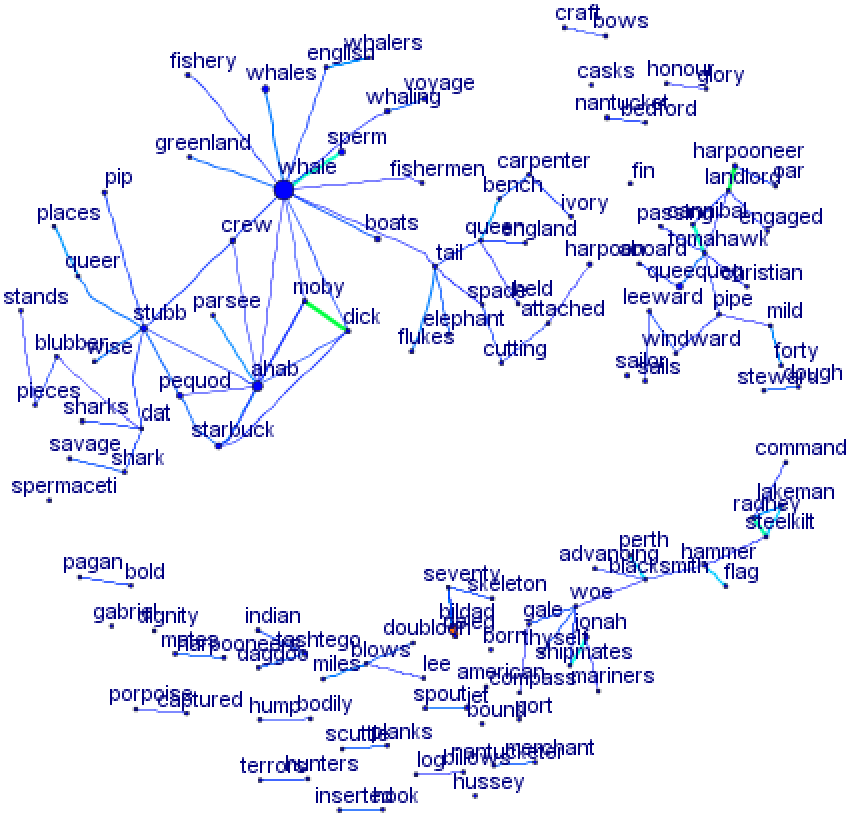

Figure 4. Force-directed layout of important words in *Moby Dick*. A fisheye transformation has been placed over the dense cluster around the word "whale", magnifying that cluster.

Brushing/Linking. Brushing and Linking techniques are generally applicable techniques for using one visualization in conjunction with another. In the basic implementation a binary pseudo-variable is added to the data set indicating the user's degree of interest. This variable is mapped to an aesthetic in each linked chart, and the user is allowed to drag a brush, or click on areas of interest in one chart so as to set corresponding values of the "degree of interest" variable and highlight corresponding parts of all linkd charts.

This technique is of particular value in the visualization of network data, as it allows any graph layout to be augmented with additional visualizations of data on nodes and links. A simple system for reducing visual complexity is to link histograms, bar charts or other summarized charts to a graph layout display, with the visibility of nodes and links based on the degree of interest. This allows users to select, for example, high values of one variable and then refine the selection by clicking on a bar of a categorical chart. The network display will then show only those items corresponding to the visually selected subset of rows.

A trivial application of this is simply to provide sliders for each variable associated with nodes and links, and by dragging the sliders define a subset of the data which is then displayed as a network display.

Animation. When data on networks varies over time, animating the results can provide insight into the nature of the variation. Visualization is particularly suitable for such data as modeling dynamically evolving networks is a hard problem, with no general models currently available. Hence visualization becomes an important early step in understanding the problem. Technically, animation is similar to brushing, and can be simulated in a brushing environment by making selections on a view of the time dimension. The main differences are in internal techniques to optimize for animation, and in the user interfaces provided for each.

KEY APPLICATIONS

Network Data Visualization is widely applicable. Communication networks such as telephony, internet and cellular are key areas, with particular emphasis on network security issues such as intrusion detection and fraud monitoring.

FUTURE DIRECTIONS

The field of network data visualization has been evolving steadily since its inception. The classic problem remains open; providing high-quality layouts for networks. It is known that most problems in this area are NP-hard, and so ongoing research will focus on approximate algorithms. Application areas are increasingly providing large networks with sometimes millions of nodes, which require new algorithms for such data. Further, evolutionary networks, in which the topology of the network itself changes over time, have become more important, and algorithms for evolving a layout smoothly from an old state to accommodate a new state are needed.

DATA SETS

The baseball data can be found online at the baseball archive: http://www.baseball1.com. This contains a wealth of tables, and the example data used above was extracted from the tables found there. The text of *Moby Dick* can be found at many sites, including Project Gutenberg: http://www.gutenberg.org.

CROSS REFERENCES

Visualizing Hierarchical Networks

RECOMMENDED READING

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

Herman I., Melancon, G. and Marshal, M.S.; *Graph Visualization and Navigation in Information Visualization: A Survey*; IEEE Transactions on Visualization and Computer

Graphics, Vol. 6 #1, 2000.

International Symposium on Graph Drawing; http://graphdrawing.org/