Network Visualization

- Information visualization deals with presentation of data in visual format.
- Goal is to take advantage of human's natural abilities to see patters, anomalies, relationships, features in visual data.
- Visualization can help in the process of extracting insight from data during decision making.
- Its advantages are based on the ability to rapidly interpret large quantities of data.

Examples

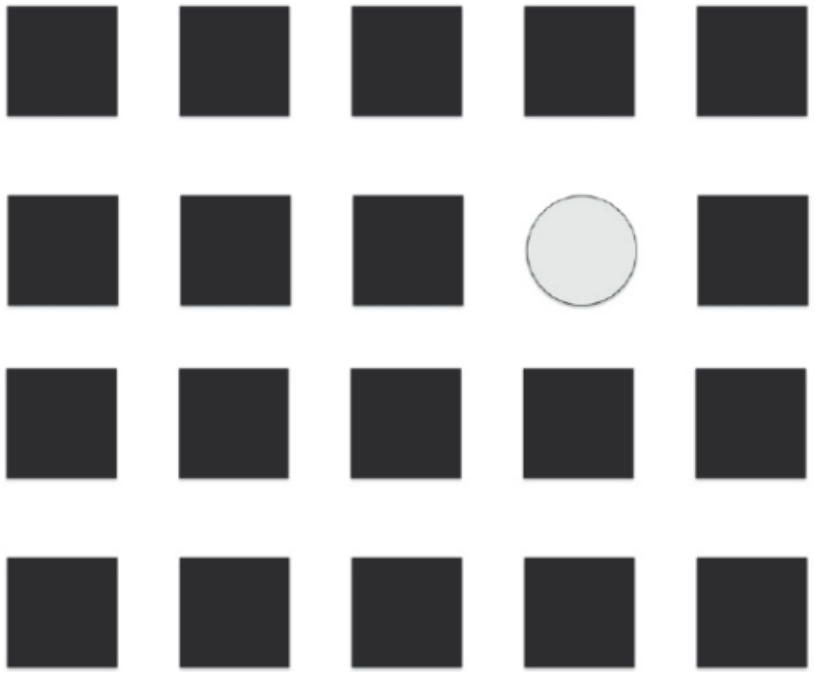


FIGURE 4.1

Without conscious analysis, it is easy to pick out the circle as an anomaly in the pattern.

Examples

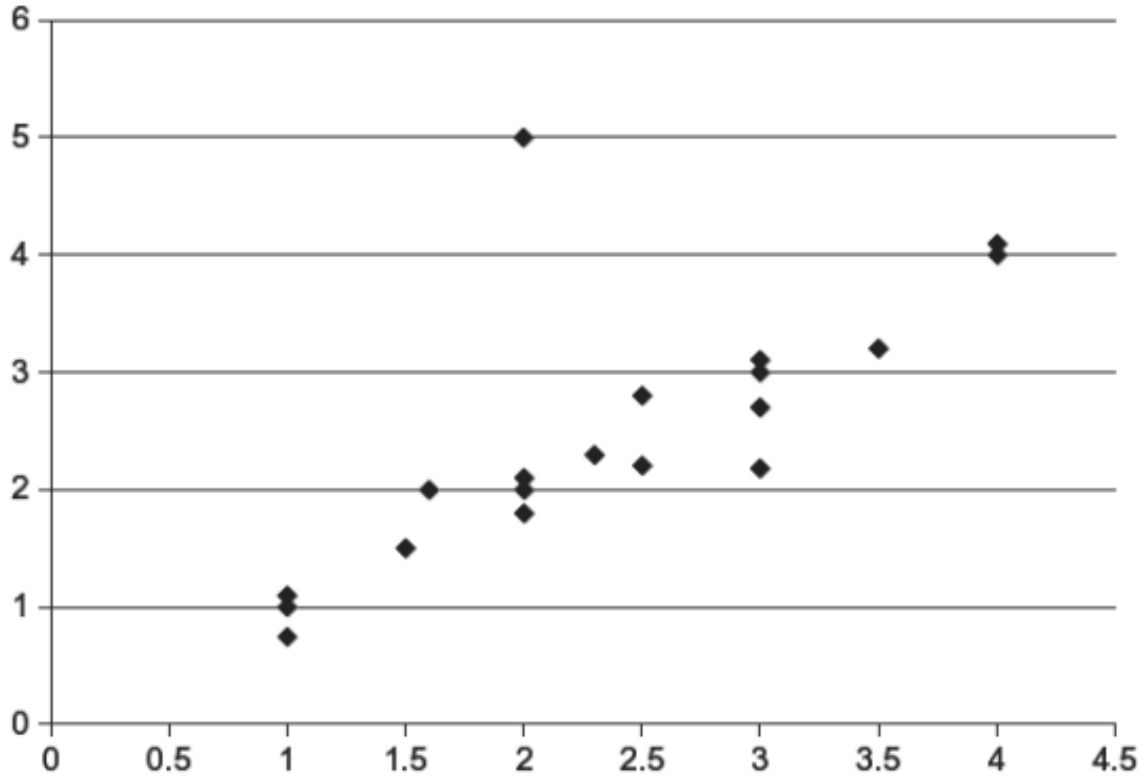


FIGURE 4.2

A single outlier point at value 2 on the x-axis is easy to see separated from the pattern of values.

Examples

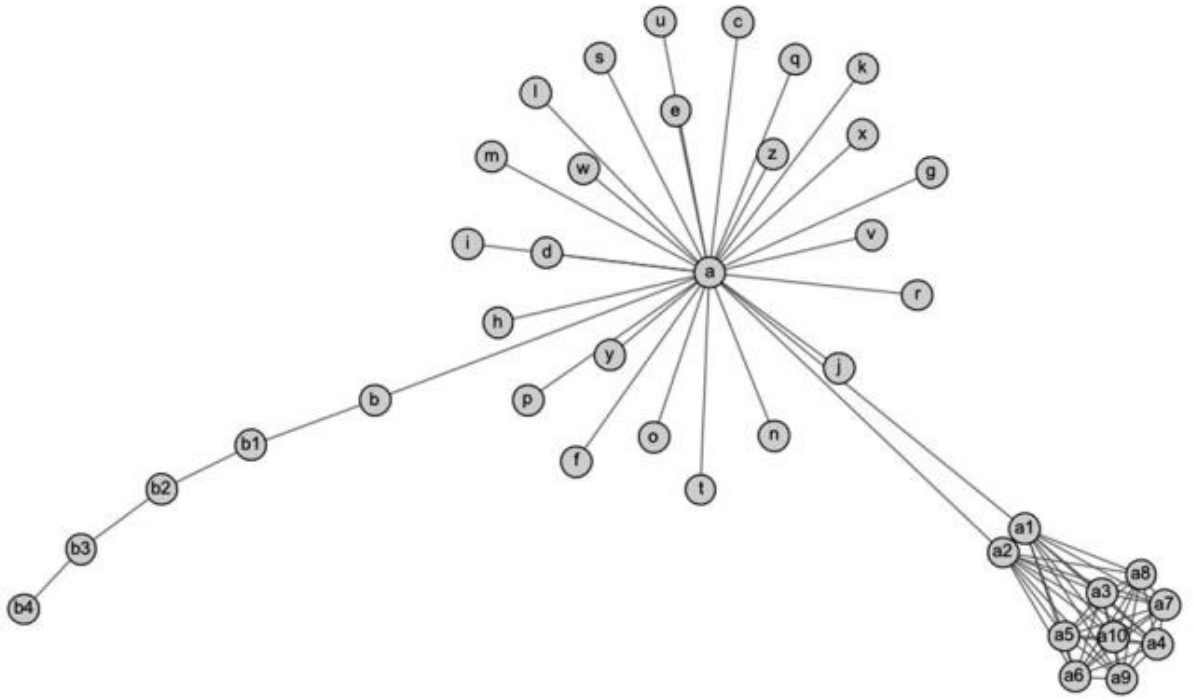


FIGURE 4.3

A sample network visualization.

- **Node Edge diagrams:**
- A node-edge diagram is an intuitive way to visualize social networks. With the node edge visualization, many network analysis tasks, such as component size calculation, centrality analysis, and pattern sketching, can be better presented.
- Social Graphs are basically a network of nodes and edges of entities and the connections between them.

Graph Visualization Techniques for Networks

➢ **Graph Layout**

✓ Most graphs are powered by a two-dimensional data system consisting of two core items: nodes and edges.

✓ How edges and nodes in a graph are laid is critical to understand the content.

✓ Researchers have presented some general guidelines that make network visualizations easier to work with:

1. Every node is visible.
2. For every node you can count its degree.
3. For every link you can follow it from source to destination.
4. Clusters and outliers are identifiable

- **Network Layout Algorithms:**
- Starts with the nodes randomly placed and iteratively move them around into better positions.
- As a result, running the algorithm multiple times will produce graphs that look different.

➢**Random layout**

✓When loading data into a visualization tool, the nodes are placed randomly.

✓It often does not provide much insight into the structure of the network.

✓It simply places the nodes at randomly computed positions inside a user-defined region.

✓It can efficiently draw the social network graph in linear time O(N).

✓ Nevertheless, the Random Layout algorithm can be useful when a random, initial placement is needed by another layout algorithm or in cases where an aesthetic, readable drawing is not important.

- ✓ This algorithm is not recommended for automata with many high-degree vertices and for those with many vertices, as there is more potential for edge-intersection and vertex overlap respectively.

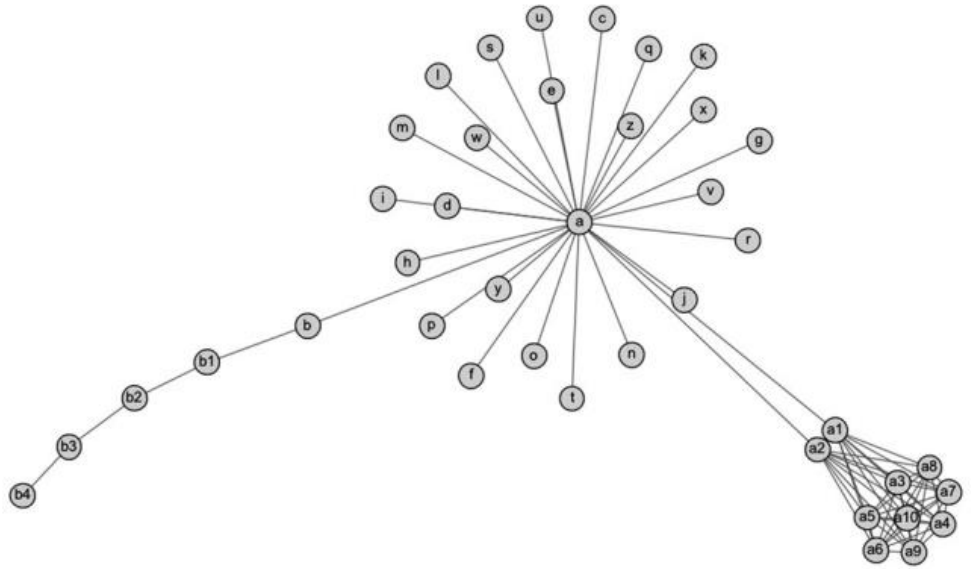


FIGURE 4.3

A sample network visualization.

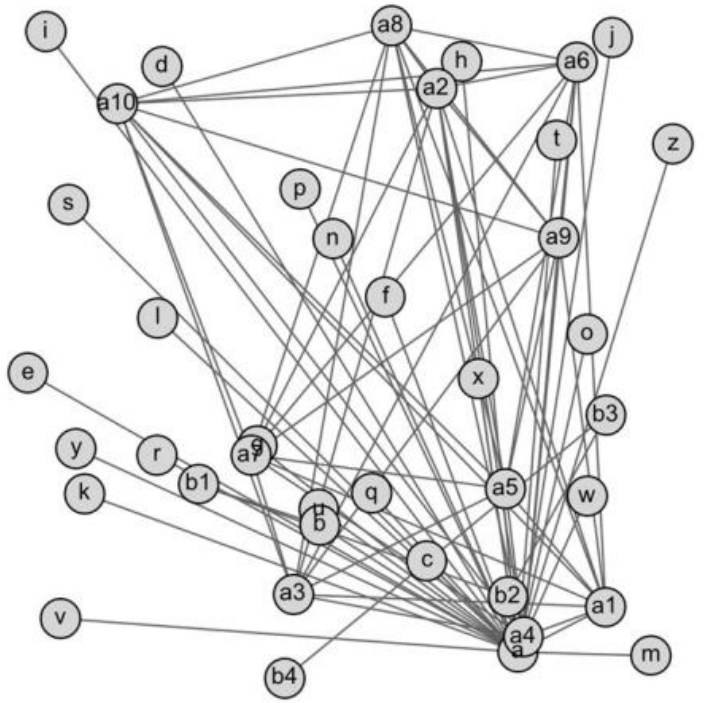


FIGURE 4.4

A random layout of the graph shown in Figure 4.3.

➢Circular layout

- ✓Circular layouts place all the nodes in a circle and then add edges between them.
- ✓ Some circular layouts place nodes closer to one another when they are more closely connected.

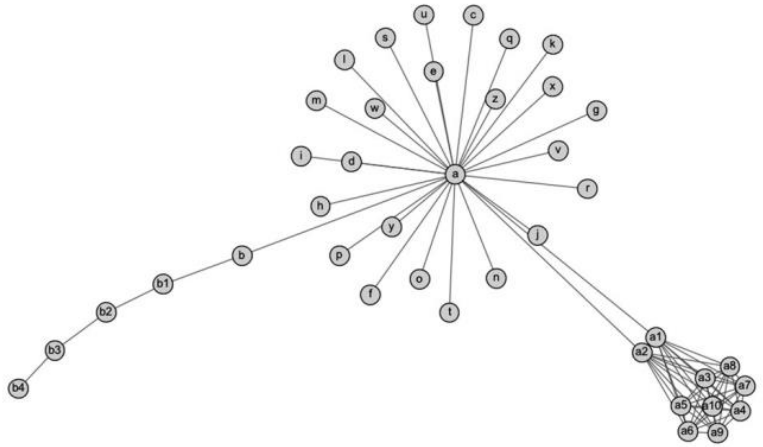


FIGURE 4.3

A sample network visualization.

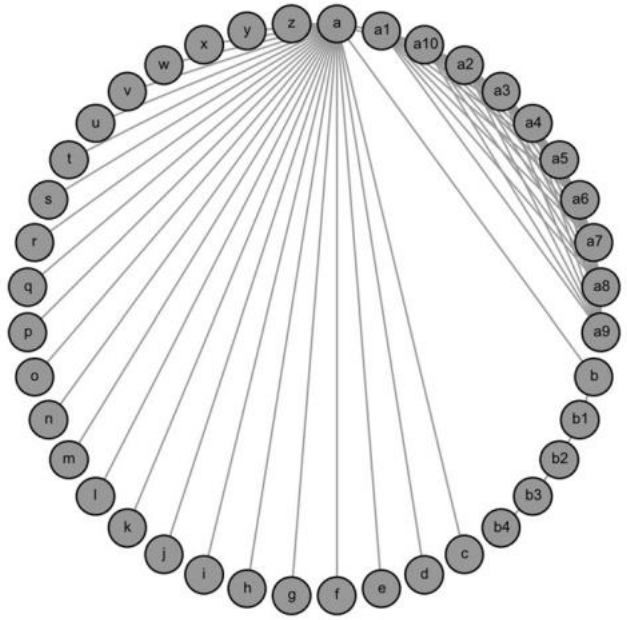
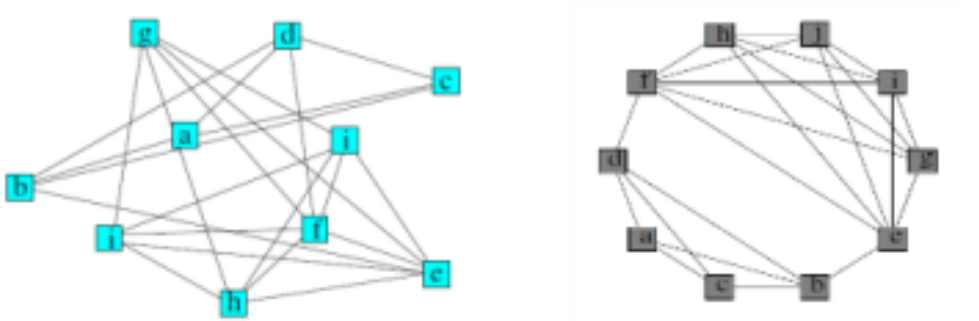


FIGURE 4.5

A circular graph layout for the same graph shown in Figure 4.3.

A circular drawing of a graph is a visualization of a graph with the following characteristics:

1) The graph is partitioned into clusters,
2) The nodes of each cluster are placed onto the circumference of an embedding circle,
3) Each edge is drawn as a straight line,
4) A node cannot be occluded by another node or by an edge.



- The circular layout is mainly used for small-medium sized graphs data analysis and in the applications where that application requires the visualization of information having more emphasis on the aesthetics of drawing.

- Grid layout

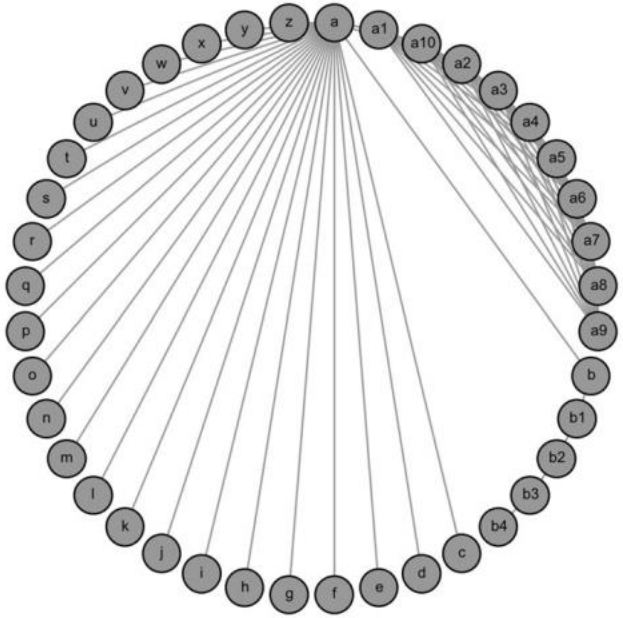


FIGURE 4.5

A circular graph layout for the same graph shown in Figure 4.3.

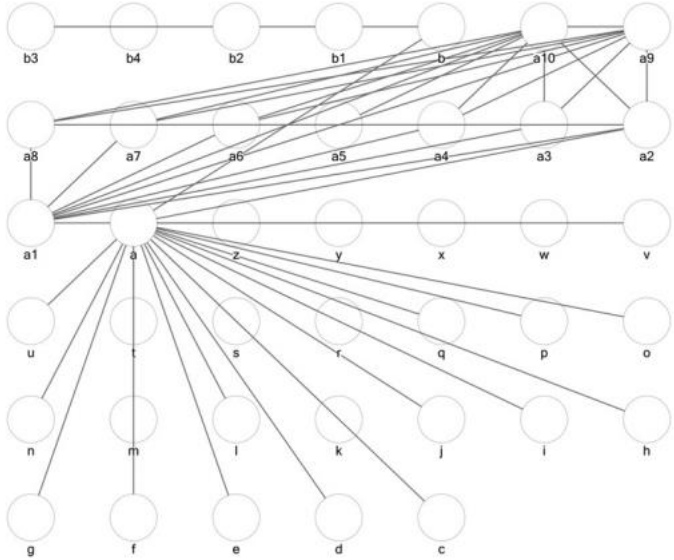


FIGURE 4.6

A grid layout of the modes in the sample graph.

➢ Force-directed layout

- The layout is dynamic and determined by the connections between the nodes.
- A force-directed layout is also known as a spring layout, which simulates the graph as a virtual physical system. In a force-directed layout, the edges act as spring and the nodes act as repelling objects.
- Those nodes that are more closely connected are laid out close to one another, and those that are distant are shown further apart.
- *A*rranges graphs in an organic and aesthetically pleasing way.
- Their purpose is to position the nodes of a graph in two-dimensional or three-dimensional space so that all the edges are of more or less equal length and there are as few crossing edges as possible, by assigning forces among the set of edges and the set of nodes, based on their relative positions, and then using these forces either to simulate the motion of the edges and nodes or to minimize their energy.
- The idea is to model the graph as a system of particles that repel or attract each other based on some criteria, such as the distance, the degree, or the group of the nodes.

- An initial random layout will be yielded first, and then the force-directed algorithms will run iteratively to adjust the positions of nodes until all graph nodes and attractive forces between the adjacent nodes run to convergence.
- Force-Directed Layout algorithms are graph drawing algorithms based only on information contained within the structure of the graph itself rather than relying on contextual information.
- The most straightforward Force-Directed algorithm uses repulsive forces between nodes and attractive forces between adjacent nodes.
- The main algorithm consists of a main loop that simulates the system for some iterations and then plots the graph.
- The simulation runs until the system reaches a state of equilibrium, where the forces are balanced and the graph is stable. The result is a layout that reflects the structure and the dynamics of the data.
- Repulsive forces push the nodes away from each other, preventing them from overlapping or clustering too much.
- Attractive forces pull the nodes together, based on the existence and the weight of the edges between them.
- The balance between these two forces determines the final shape and appearance of the graph.

- The running cost of a force-directed layout is much higher than that of a random layout, especially when the number of nodes is large It is therefore not suitable for graphs larger than hundreds of nodes.

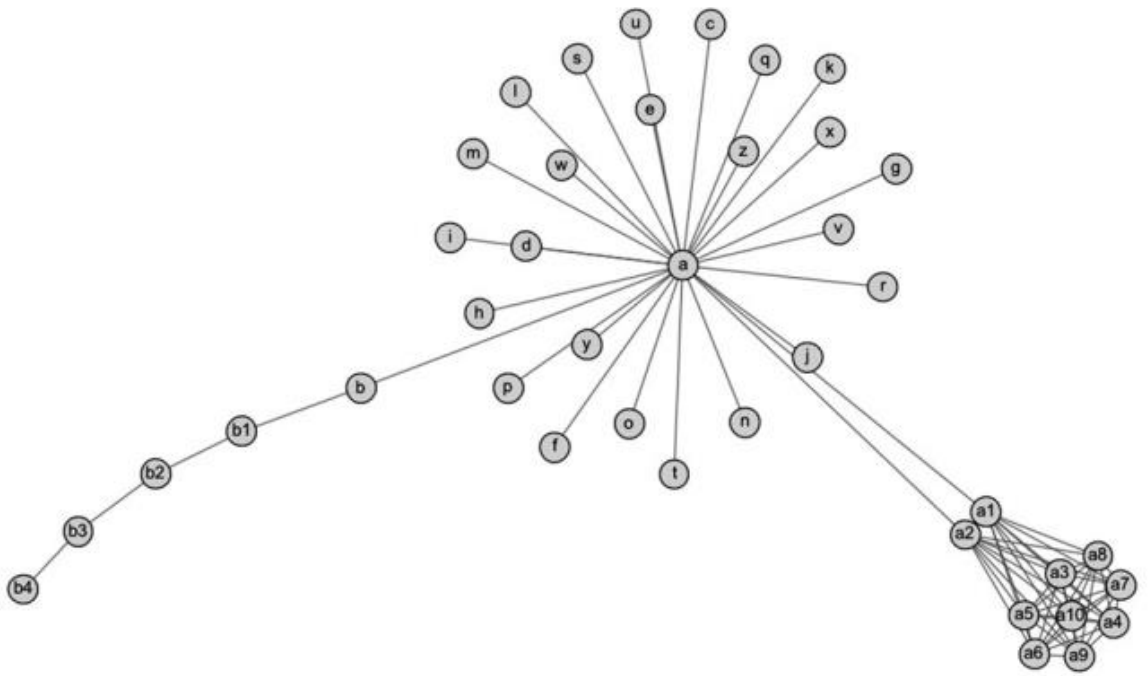


FIGURE 4.3

A sample network visualization.

➢ Yifan Hu layout

- Belongs to the category of force-directed algorithms,
- It optimizes the overall internode repulsions in the network.

➢ Harel-Koren fast multiscale layout

- Is designed to quickly lay out large, complex graphs.
- It is based on force-directed layout algorithms but uses optimizations in the underlying code to make the algorithm computationally efficient.
- For large graphs with thousands of nodes, generating a layout with many force-directed algorithms can take a very long time.
- With Harel-Koren, it often can be achieved in a few seconds, making it an ideal choice for large networks.

Visualizing network features

- Along with nodes and edges, the other network features, like edge weights, node properties, labels, and clusters, can also be visualized.
- **Labels:**
  - ✓Labels are the attributes to show in a network, both on nodes and on edges.
  - ✓Challenges can be minimized using the following:
    - Putting boxes around the text,
    - Showing a few labels of interest,
    - On interactive interfaces that only show labels on demand.
  - ✓Still, there are no solutions to eliminate this problem when producing fixed visualization images, so often labels are left off.

- **Size, shape, and color:**
- Categorical or quantitative attributes are easy to show by adjustments in size, shape, or color.
-

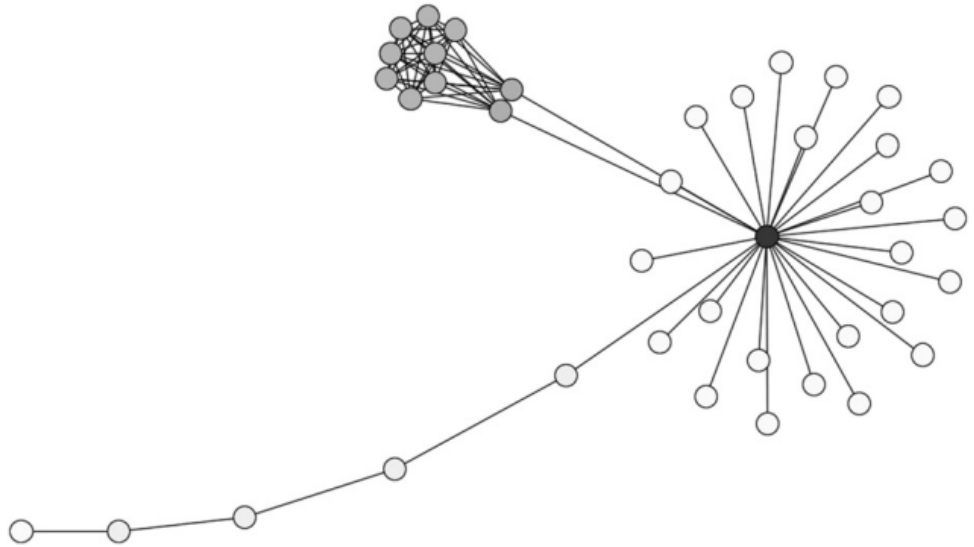
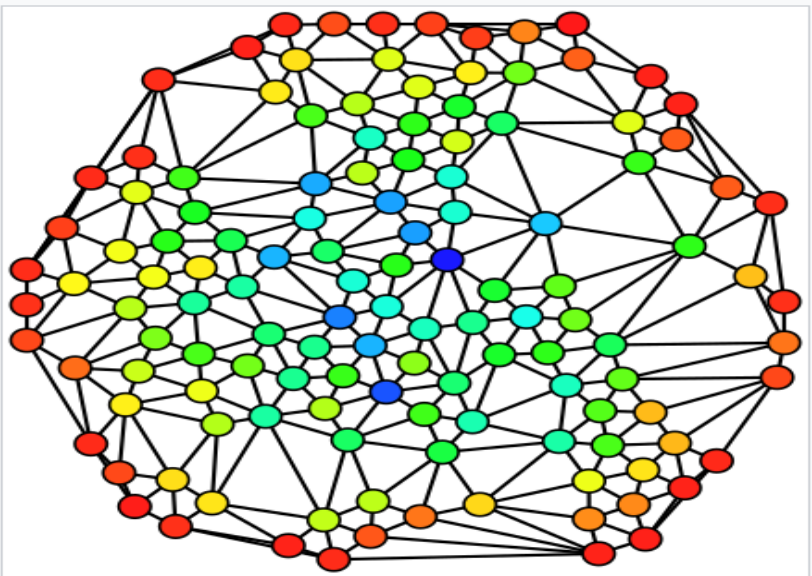


FIGURE 4.11

Color-coding nodes according to their degree, with higher degree shown by darker nodes.



An undirected graph colored based on the betweenness centrality of each vertex from least (red) to greatest (blue).

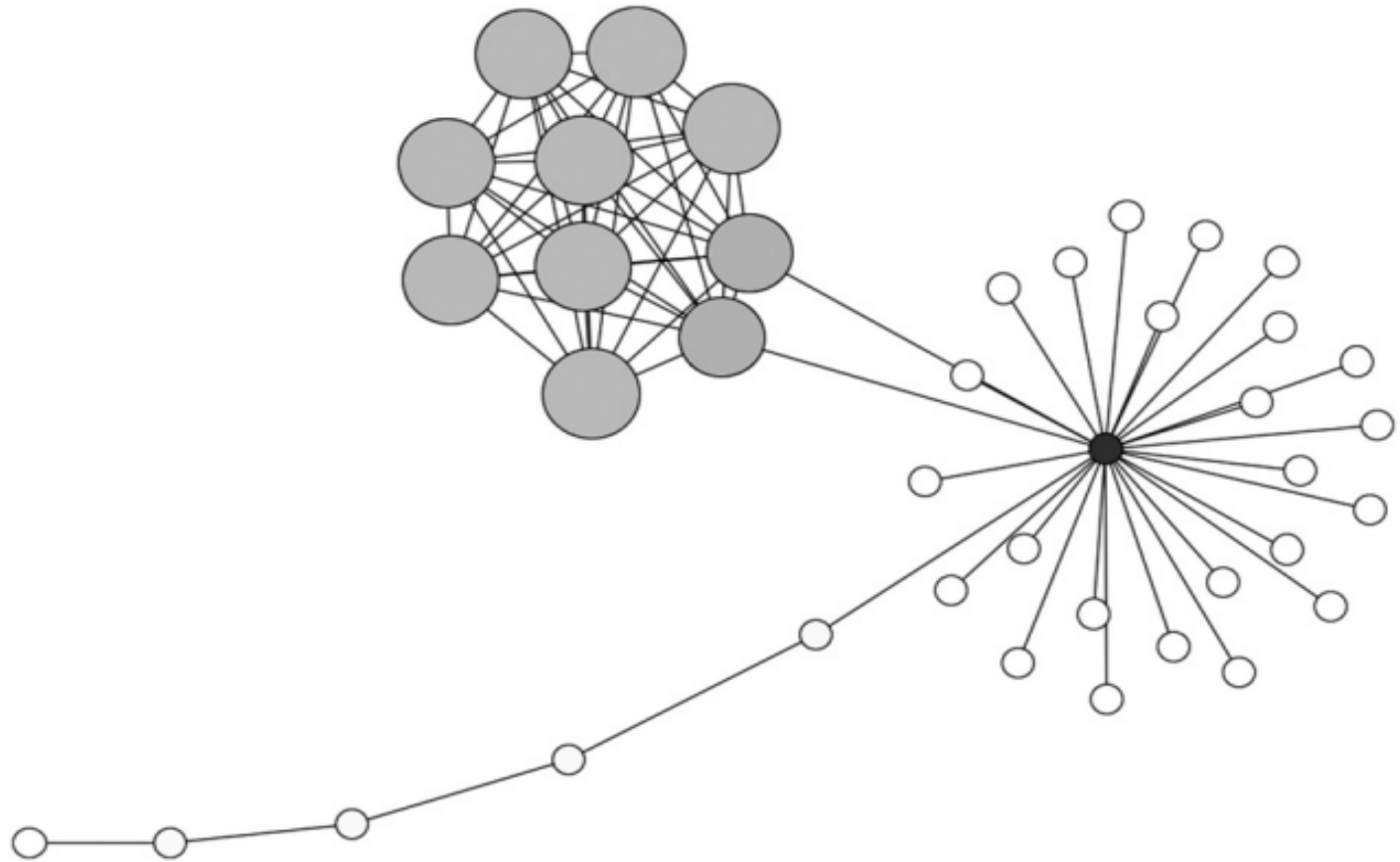


FIGURE 4.12

A graph indicating clustering coefficient with node size and degree with node color.

Graph with two attributes

- Edges can also be treated with color or thickness to indicate their attributes.
- For example, different types of relationships could each be coded in a different color.
- Edge weights are also commonly visualized. These could indicate the strength of a relationship, the frequency of communication, or other factors.

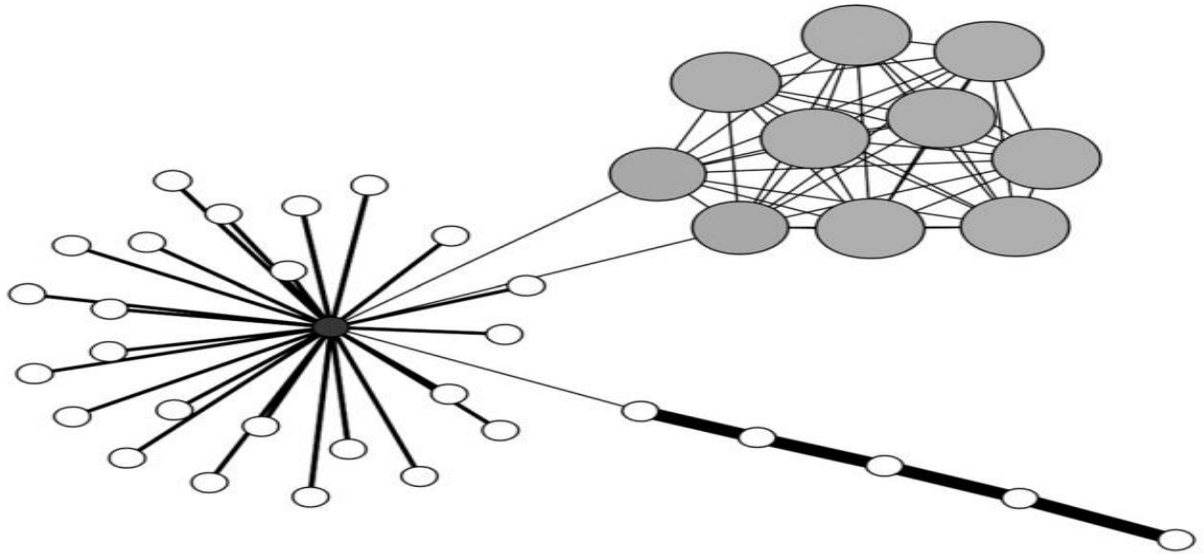


FIGURE 4.13

The sample network with edge width indicating the weight on each edge. Note that the central node has medium-strength relationships with most neighbors, but weak ones to the cluster in the upper right and the chain in the lower right. The chain of nodes in the lower right have high weights on the edges connecting them.

- **Larger graph properties**

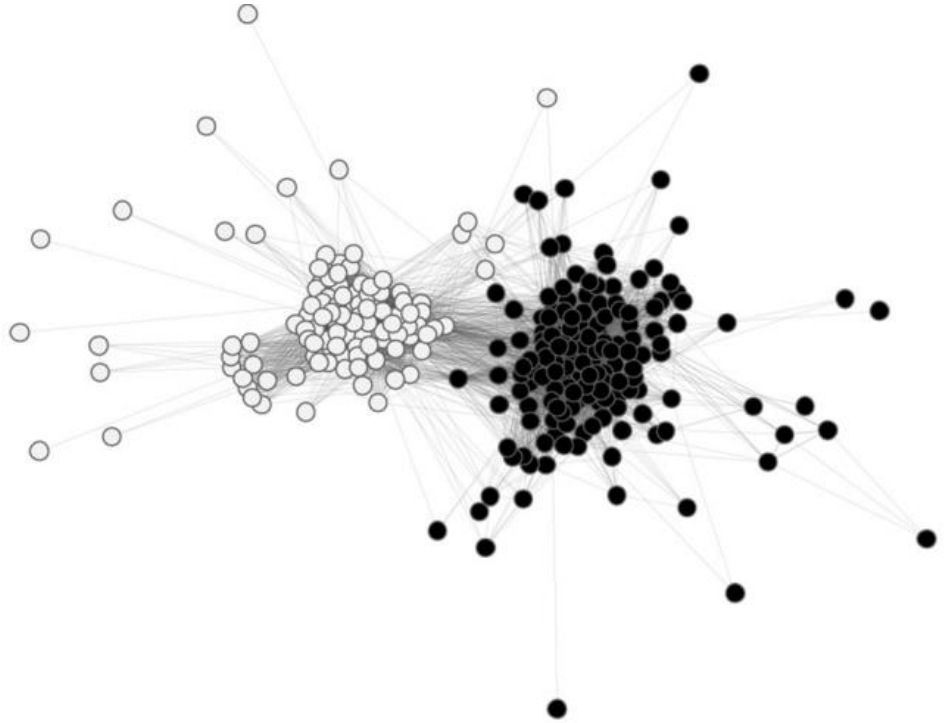


FIGURE 4.14

A network of YouTube videos where color indicates the community or cluster to which each node belongs.

Scale issues

- Visualization is very useful for analyzing smaller networks (few hundred to few thousand nodes)
- When networks become much larger, the quality of the visualization diminishes.

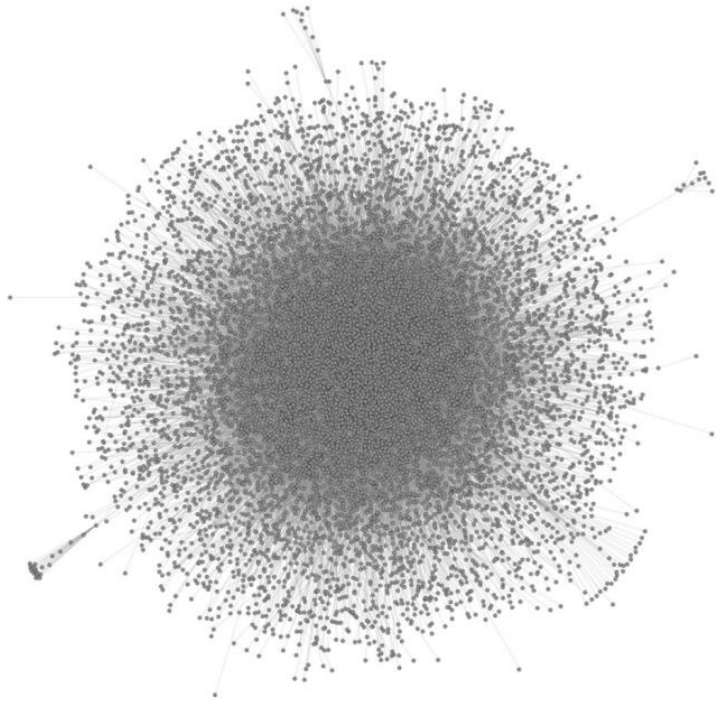


FIGURE 4.15
A network with 11,000 nodes and 40,000 edges.

- **Density**
- Density is also be a problem for visualization, even if the number of nodes is small.

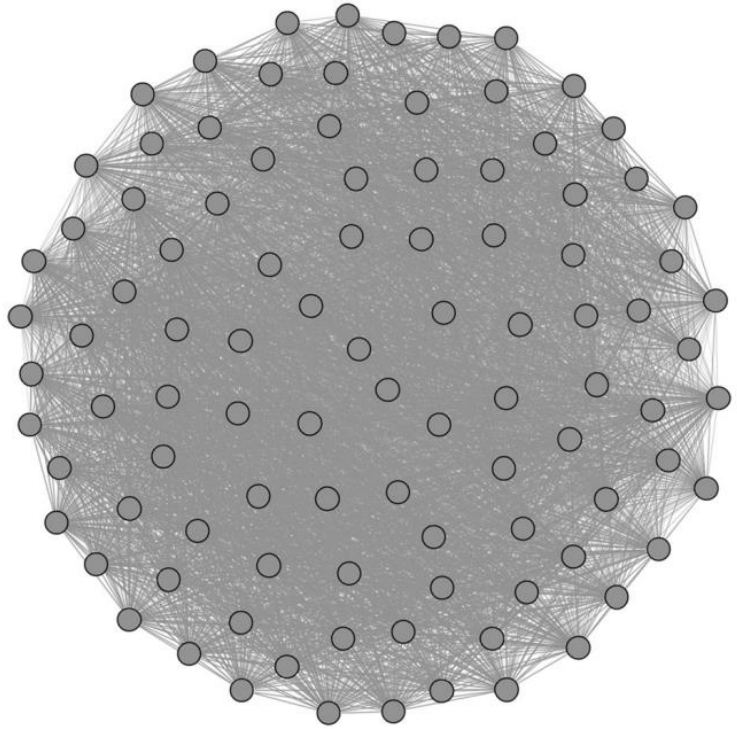


FIGURE 4.16

A network of senators (nodes) with edges connecting senators who have voted the same way at least 40% of the time. The network is very dense, so it is not possible to see any interesting patterns.

- **Filtering for visual patterns**
- It is often difficult to see any patterns in very dense networks.
- One way to compensate for this is to filter the networks when possible.

FIGURE 4.17

The same network of senators as [illegible], now filtered to include only edges between senators who have voted the same way on at least two-thirds of bills.

- **Graph simplification**
- Graph simplification techniques include grouping clusters of nodes into a single node and representing the edges between clusters as a single edge
- Rrepresenting structural patterns as representative shapes, or showing only part of the graph at a time.

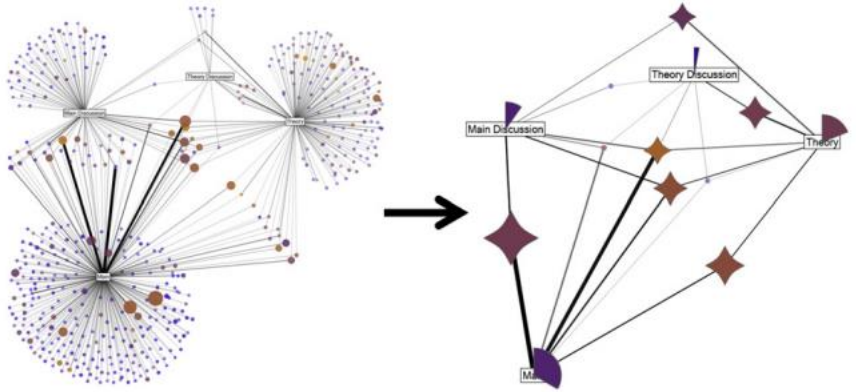


FIGURE 4.19

The graph on the left is summarized using simple glyphs into the graph on the right. This uses a technique called *motif simplification* (Dunne and Shneiderman, 2012).