Visual Analytics: Arranging Non-Tabular Data

Ian Nabney

ian.nabney@bristol.ac.uk

Overview

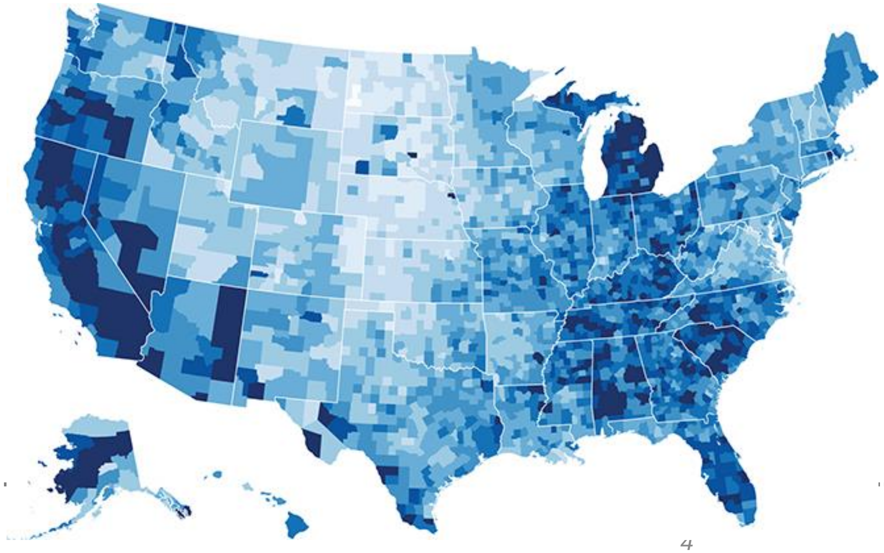
- Reading: Chapters 8 (sections 8.1 to 8.3) and 9 of Munzner
- Understand how to represent geographical data
- Understand how network and tree data can be represented
- Able to select appropriate methods for displaying non-tabular data for a particular task

Geographic data

- For datasets with spatial semantics, the usual choice for arrange is to use the given spatial information to guide the layout
- In this case, the choices of express, separate, order, and align do not apply because the position channel is not available for directly encoding attributes
- The two main spatial data types are
  - geometry, where shape information is directly conveyed by spatial elements that do not necessarily have associated attributes
  - spatial fields, where attributes are associated with each cell in the field
- The latter type is often associated with scientific visualisation – an important domain, but not one we will cover

Choropleth maps

- Cartographers have grappled with design choices for the visual representation of geographic spatial data for many hundreds of years
- A **choropleth map** shows a quantitative attribute encoded as colour over regions delimited as area marks, where the shape of each region is determined using given geometry
- Major design choices for choropleths are how to construct the colourmap, and what region boundaries to use

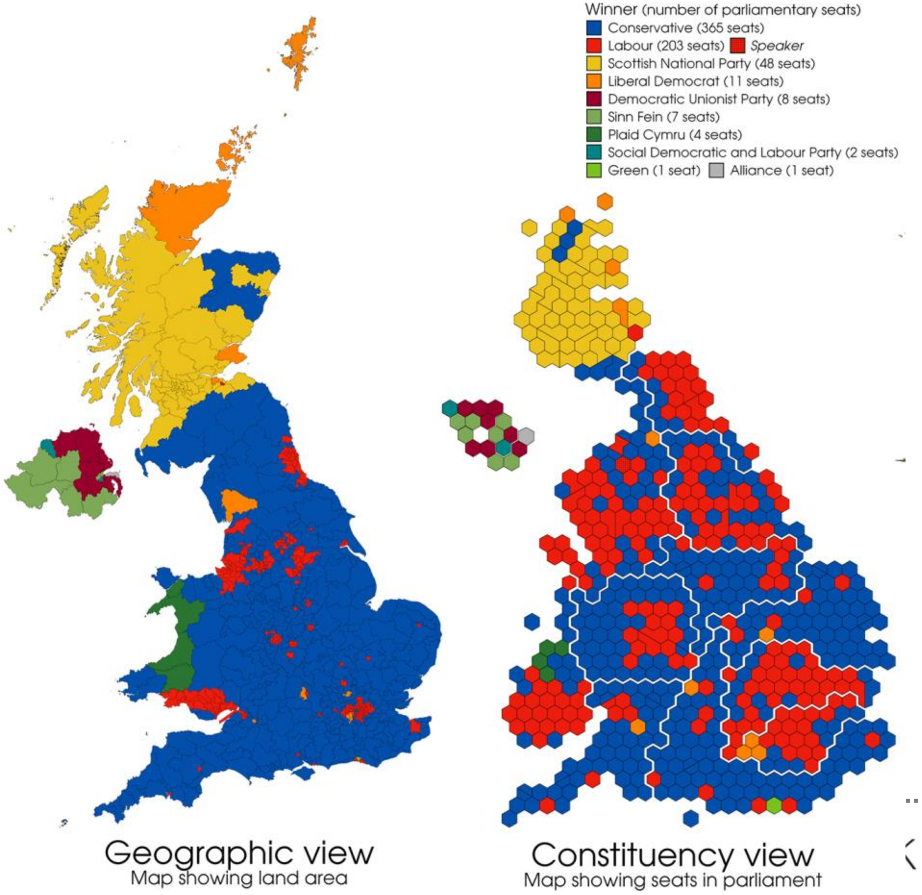


US unemployment rates from 2008 with a segmented sequential colormap. The white-to-blue colormap has a sequence of nine levels with monotonically decreasing luminance. The region granularity is counties within states

Spatial representation: cartograms

- One drawback of choropleth maps is that regions tend to have different areas – this channel has the largest effect on perception which can be misleading
- 2019 UK general election

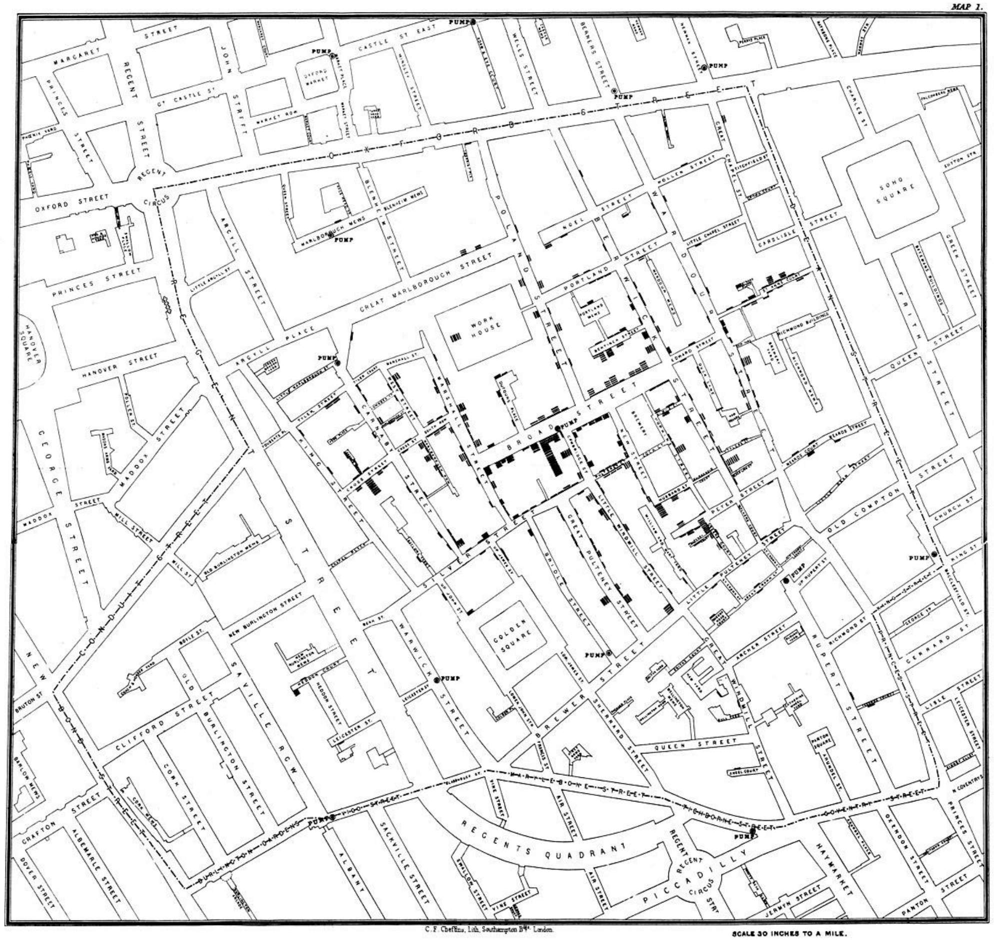
- Geographic view gives the impression of an overwhelming Conservative win
- Constituency view shows a much more even balance
- This is because Labour tends to do better in city constituencies which have a higher population density


Winner (number of parliamentary seats)
Conservative (365 seats)
Labour (203 seats)
Speaker
Scottish National Party (48 seats)
Liberal Democrat (11 seats)
Democratic Unionist Party (8 seats)
Sinn Fein (7 seats)
Plaid Cymru (4 seats)
Social Democratic and Labour Party (2 seats)
Green (1 seat)
Alliance (1 seat)
Geographic view
Map showing land area
Constituency view
Map showing seats in parliament

- https://worldmapper.org/uk-general-election-2019/
- https://resource.esriuk.com/blog/eleven-ways-to-map-a-general-election/

John Snow’s cholera map

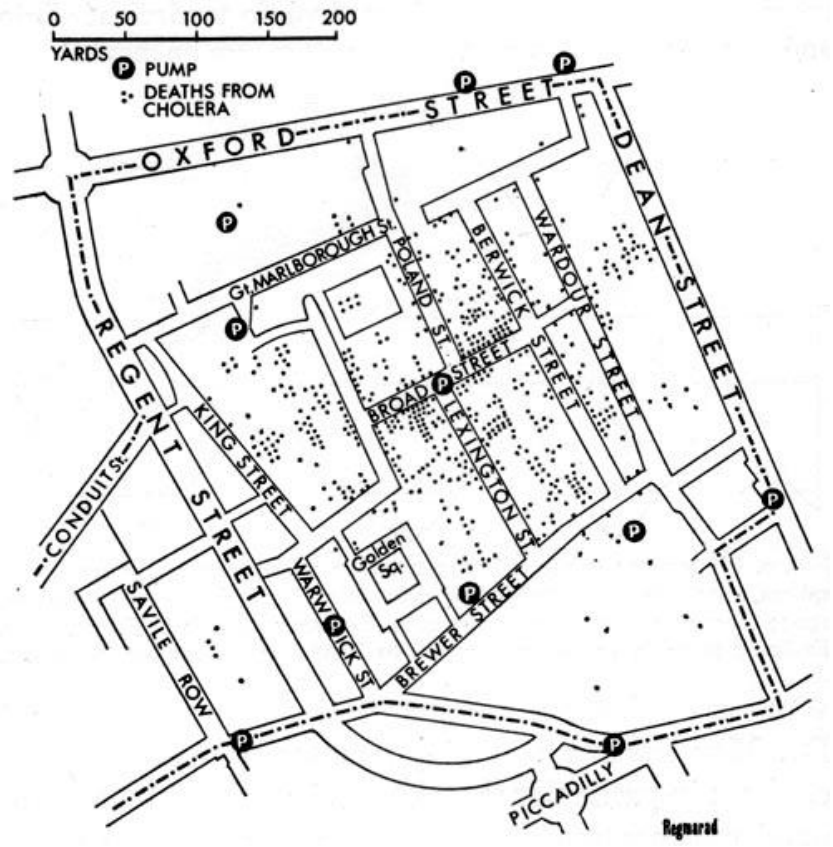
- For the 1854 cholera outbreak in London's Broad Street region, he presented two maps
- The first was shown on December 4, 1854 at a meeting of the London Epidemiological Society
- Several months later he published this map in his book, On the Mode of Communication of Cholera
- He used bars to represent deaths that occurred at the specified households



https://www.ph.ucla.edu/epi/snow/mapsbroadstreet.html
https://www.ph.ucla.edu/epi/snow/snowmap1a.html

Details of the maps

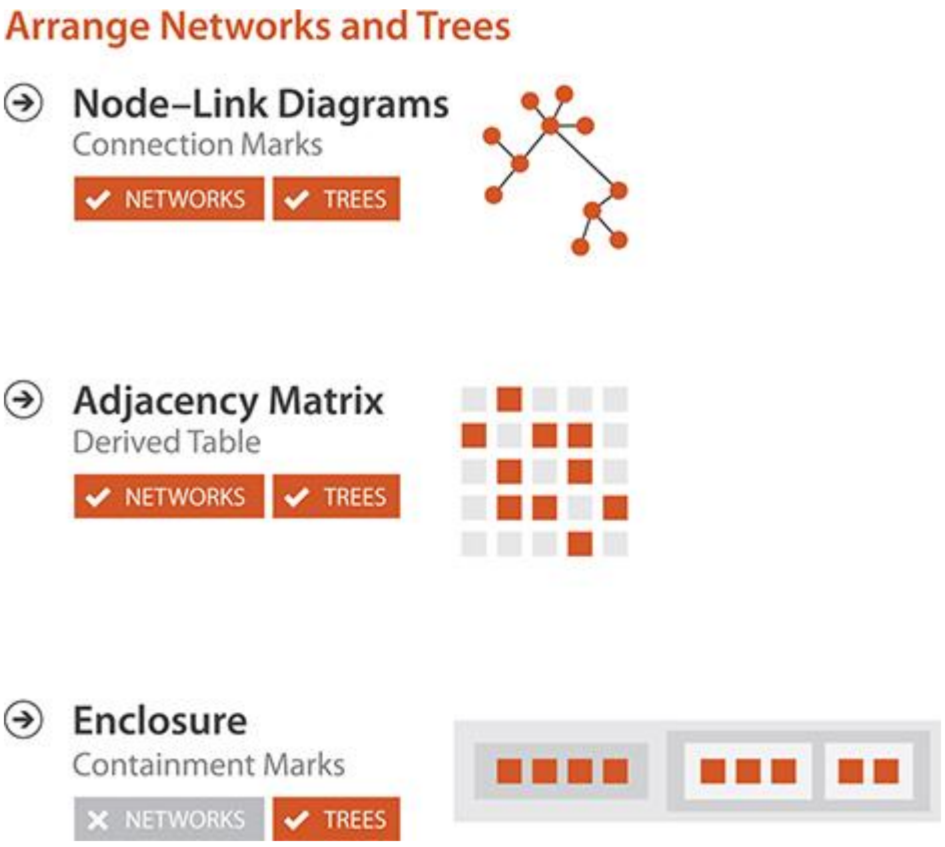
- Snow hypothesised that cholera was spread in water
- Note location of pumps (not quite correct in this redrawn dot-map version)
- The densest cluster of cases is around the Broad Street pump
- So John Snow ordered the handle of that pump to be removed, so it could not be used
- Cases declined dramatically (they were already going down)
- A great early example of visualisation for public health



Redrawn by Regmarad, 1960

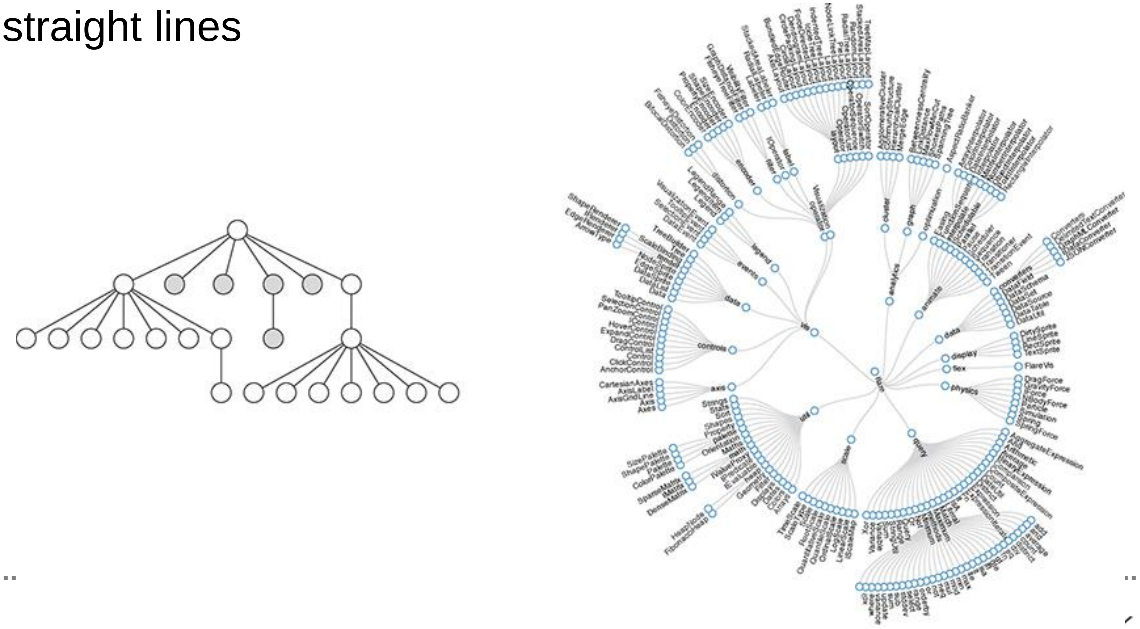
Arranging networks and trees

- The **node–link** diagram family of visual encoding idioms uses the connection channel, where marks represent links. Many different choices of layout algorithms
- **Matrix** views directly show adjacency relationships
- Tree structure can be shown with the **containment** channel, where enclosing link marks show hierarchical relationships through nesting.



Node-link diagrams

- Figure (a) shows a tree of 24 nodes laid out with a **triangular vertical node–link layout**, with the root on the top and the leaves on the bottom. In addition to the connection marks, it uses vertical spatial position channel to show the depth in the tree. The horizontal spatial position of a node does not directly encode any attributes
- Figure (b) shows a tree of a few hundred nodes laid out with a **spline radial layout**. This layout uses essentially the same algorithm for density without overlap, but the visual encoding is radial rather than rectilinear: the depth of the tree is encoded as distance away from the centre of the circle. Also, the links of the graph are drawn as smoothly curving splines rather than as straight lines

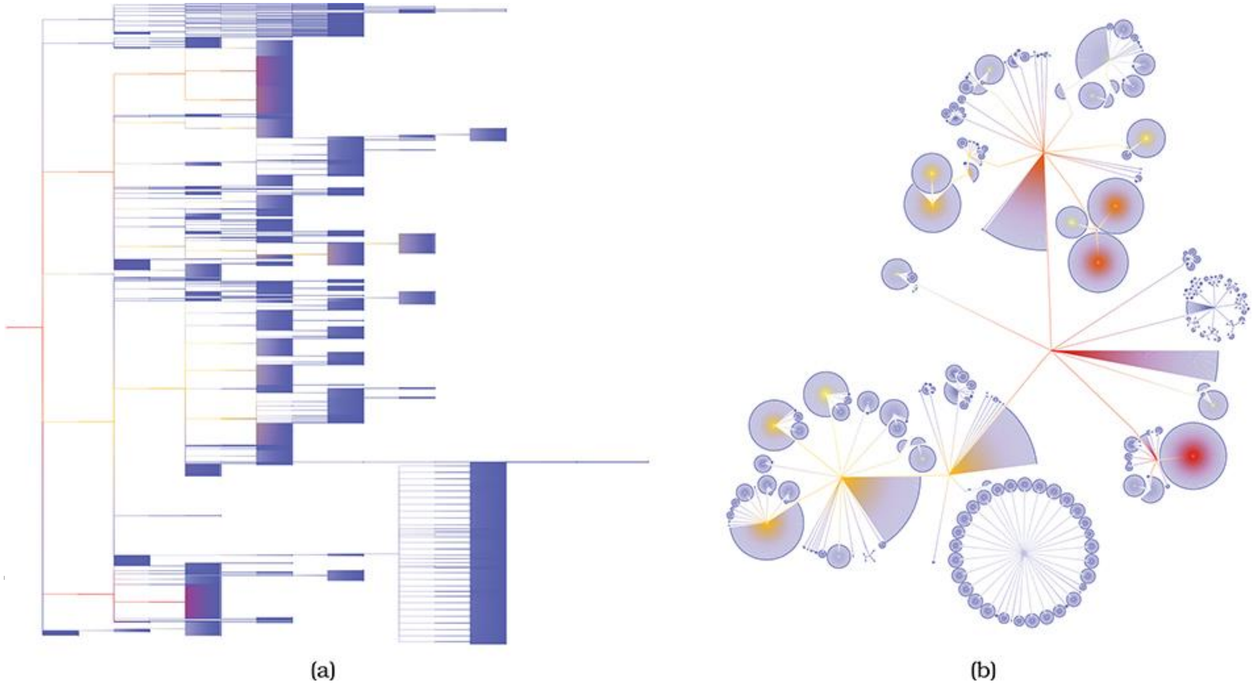


(a)

(b)

More node-link layouts

- Figure (a) shows a tree of 5161 nodes laid out as a **rectangular horizontal node–link diagram**, with the root on the left and the leaves stretching out to the right. The edges are coloured with a purple to orange continuous colormap according to the Strahler centrality metric
- Figure (b) shows the same tree laid out with the **BubbleTree algorithm**. BubbleTree is a radial rather than rectilinear approach where subtrees are laid out in full circles rather than partial circular arcs. Spatial position does encode information about tree depth, but as relative distances to the centre of the parent rather than as absolute distances in screen space



Node placement algorithms

- **Force-directed algorithm**: network elements are positioned according to a simulation of physical forces where nodes push away from each other while links act like springs that draw their endpoint nodes closer to each other
- Designed to minimize the number of distracting artifacts such as edge crossings and node overlaps, so the spatial location of the elements is a side effect of the computation rather than directly encoding attributes
- Spatial proximity is sometimes meaningful but sometimes arbitrary; this ambiguity can mislead the user.
- This situation is a general problem for idioms where spatial position is implicitly chosen rather than deliberately used to encode information

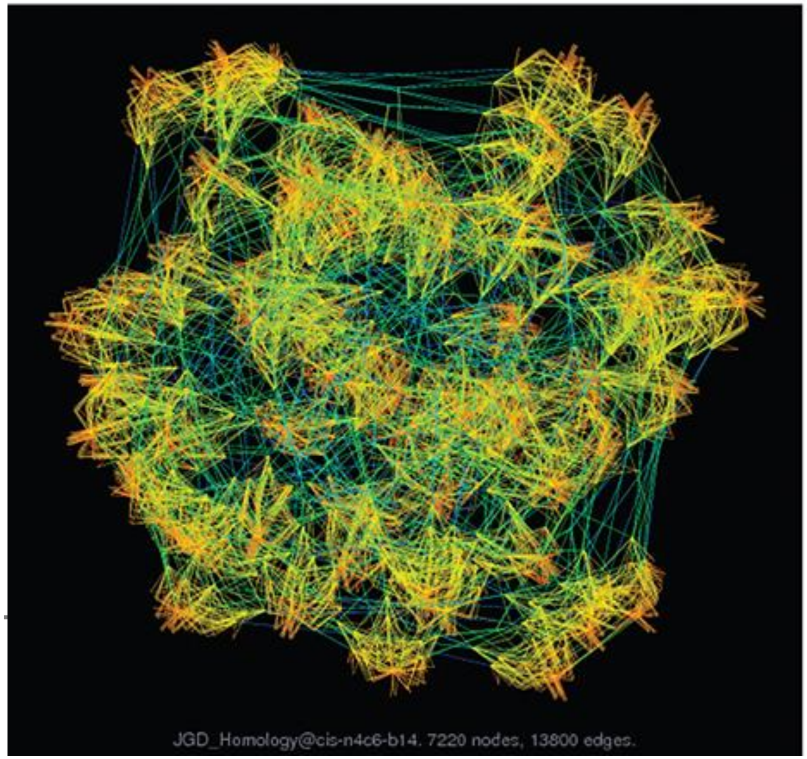
Drawbacks of force-directed algorithms

- **Scalability**, both in terms of the visual complexity of the layout and the time required to compute it. The layout quickly degenerates into a hairball of visual clutter with even a few hundred nodes, where the tasks of path following or understanding overall structural relationships become very difficult.
- Straightforward force-directed placement is unlikely to yield good results when the number of nodes is more than roughly four times the number of links.
- Many force-directed placement algorithms are notoriously **brittle**: they have many parameters that can be tweaked to improve the layout for a particular dataset, but different settings are required to do well for another.
- As with many kinds of computational optimization, they can get **stuck in a local minimum** energy configuration that is not the globally best answer.
- The simplest algorithms **do not converge**: the nodes never settle down to a final location if the user does not explicitly intervene to halt the layout process. More sophisticated algorithms automatically stop by determining that the layout has reached a good balance between the forces.

Multi-level network idioms

- In **multi-level network idioms**, the original network is augmented with a derived cluster hierarchy to form a compound network. The hierarchy is computed by coarsening the original network into successively simpler networks that attempt to capture the most essential aspects of the original's structure.
- By laying out the simplest version of the network first, and then improving the layout with the more and more complex versions, both the speed and quality of the layout can be improved. These approaches do better at avoiding the local minimum problem

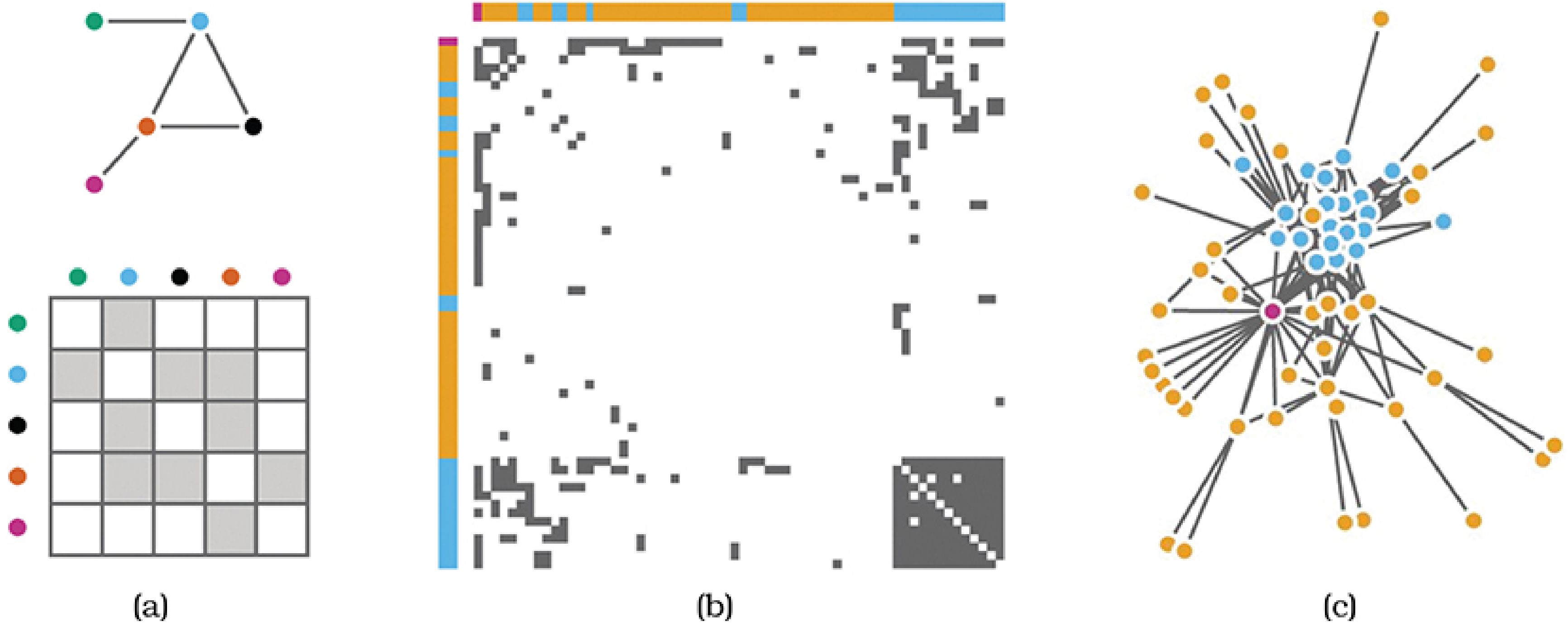
- Network of 7220 nodes and 13,800 edges using the multilevel scalable force-directed placement (sfdp) algorithm
- Cluster structure is visible:
  - dense clusters with short orange and yellow edges can be distinguished from the long blue and green edges between them
- There are still limits: see the notes



Adjacency matrix view

- The **adjacency matrix** of a network lays out the nodes in order along rows and columns and has $a_{ij} = 1$ if there is a connection from node *i* to node *j* and 0 otherwise. For undirected graphs, $a_{ij} = a_{ji}$, and the matrix is symmetric.
- A network can be visually encoded as an adjacency matrix view, where connections are indicated by colouring an area mark in the cell in the matrix that is the intersection between their row and column.
- Additional information about another **connection** attribute is often encoded by colouring matrix cells.
- The possibility of size-coding matrix cells is limited by the number of available pixels per cell; typically only a few levels would be distinguishable between the largest and the smallest cell size.
- Network matrix views can also show weighted networks, where each link has an associated quantitative value attribute, by encoding with an ordered channel such as luminance or size.

Examples



Matrix views can achieve very high information density, up to a limit of one thousand nodes and one million edges, due to the use of small area marks

Comparison of network views 1

- Node–link diagrams in general are well suited for tasks that involve understanding the network topology: the direct and indirect connections between nodes in terms of the number of hops between them through the set of links.
- Their weakness is that past a certain limit of network size and link density (ratio of links to nodes), they become impossible to read because of occlusion from edges crossing each other and crossing underneath nodes. Trees have a link density of one, with one edge for each node. The upper limit for node–link diagram effectiveness is a link density of around three or four.
- As the network size increases, the resulting visual clutter from edges and nodes occluding each other eventually causes the layout to degenerate into an unreadable hairball. A great deal of algorithmic work in graph drawing has been devoted to increasing the size of networks that can be laid out effectively, and multilevel idioms have led to significant advances in layout capabilities. Simpler algorithms can support hundreds of nodes while more state-of-the-art ones handle thousands well but degrade for tens of thousands.
- Interactive navigation and exploration idioms can address the problem partially but not fully. Filtering, aggregation, and navigation are design choices that can ameliorate the clutter problem, but they do impose cognitive load on the user who must then remember the structure of the parts that are not visible.

Comparison of network views 2

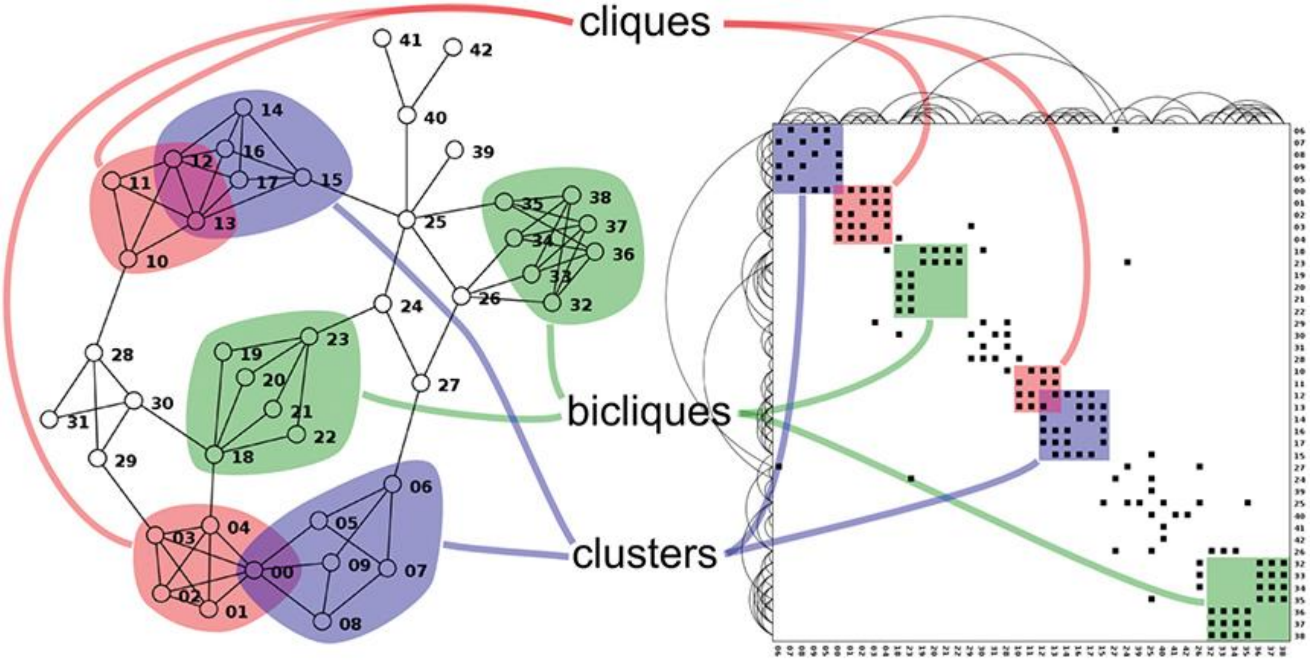
- A major strength of matrix views is perceptual scalability for both large and dense networks. Matrix views completely eliminate the occlusion of node–link views, and thus are effective even at very high information densities.
- Matrix views can be laid out within a **predictable** amount of screen space, whereas node–link views may require a variable amount of space depending on dataset characteristics.
- Matrix views are **stable**; adding a new item will only cause a small visual change. In contrast, adding a new item in a force-directed view might cause a major change. This stability allows multilevel matrix views to easily support geometric or semantic zooming.
- Matrix views can also be used in conjunction with reordering, where the linear ordering of the elements along the axes is changed on demand.
- Matrix views also shine for quickly estimating the number of nodes in a graph and directly supporting search through fast node lookup. Finding an item label in an ordered list is easy, whereas finding a node given its label in node–link layout is time consuming because it could be placed anywhere through the two-dimensional area. Node–link layouts can be augmented with interactive support for search by highlighting the matching nodes as the labels are typed.

Comparison of network views 3

09/02/2022

- One major weakness of matrix views is unfamiliarity: most users are able to easily interpret node–link views of small networks without the need for training, but they typically need training to interpret matrix views.
- The completely interconnected lines showing a clique in the node–link graph is instead a square block of filled-in cells along the diagonal in the matrix view.
- Similarly, the biclique structure of node subsets where edges connect each node in one subset with one in another is salient, but different, in both views.
- The degree of a node (the number of edges that connect to it), can be found by counting the number of filled-in cells in a row or column.

N.B. Ordering of nodes may affect visibility of these features

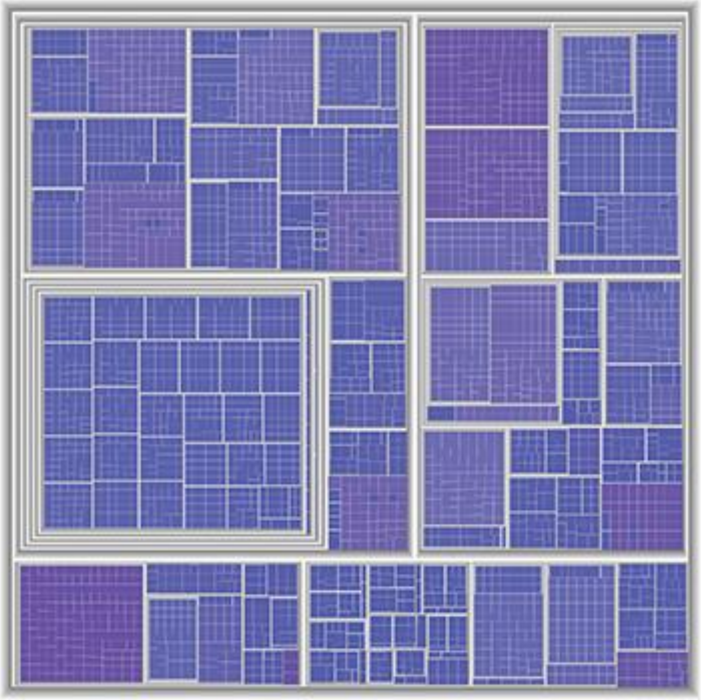


Comparison of network views 4

- Most crucial weakness of matrix views is their lack of support for investigating topological structure because they show links in a more indirect way than the direct connections of node–link diagrams. This weakness is a direct trade-off for their strength in avoiding clutter.
- One reason that node–link views are so popular, despite the many other strengths of matrix views listed above, might be that most complex domain tasks involving network exploration end up requiring topological structure inspection as a subtask.
- An empirical investigation [Ghoniem et al. 05] compared node–link and matrix views for many low-level abstract network tasks. It found that node–link views are best for small networks and matrix views are best for large networks.
- Several tasks became more difficult for node–link views as size increased, whereas the difficulty was independent of size for matrix views: approximate estimation of the number of nodes and of edges, finding the most connected node, finding a node given its label, finding a direct link between two nodes, and finding a common neighbour between two nodes. The task of finding a multiple-link path between two nodes was always more difficult in matrix views, even with large network sizes.

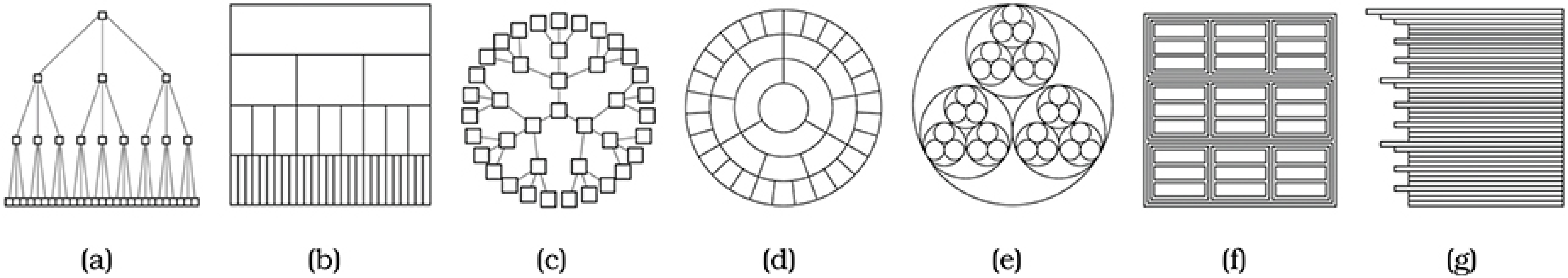
Containment marks: Treemaps

- Containment marks are very effective at showing complete information about hierarchical structure, in contrast to connection marks that only show pairwise relationships between two items at once
  - All of the children of a tree node are enclosed within the area allocated that node, creating a nested layout. The size of the nodes is mapped to some attribute of the node.
  - Here, node size encodes file size.
  - Containment marks are not as effective as the pairwise connection marks for tasks focused on topological structure, such as tracing paths through the tree, but they shine for tasks that pertain to understanding attribute values at the leaves of the tree.
  - They are often used when hierarchies are shallow rather than deep.



Same network as shown on slide 10

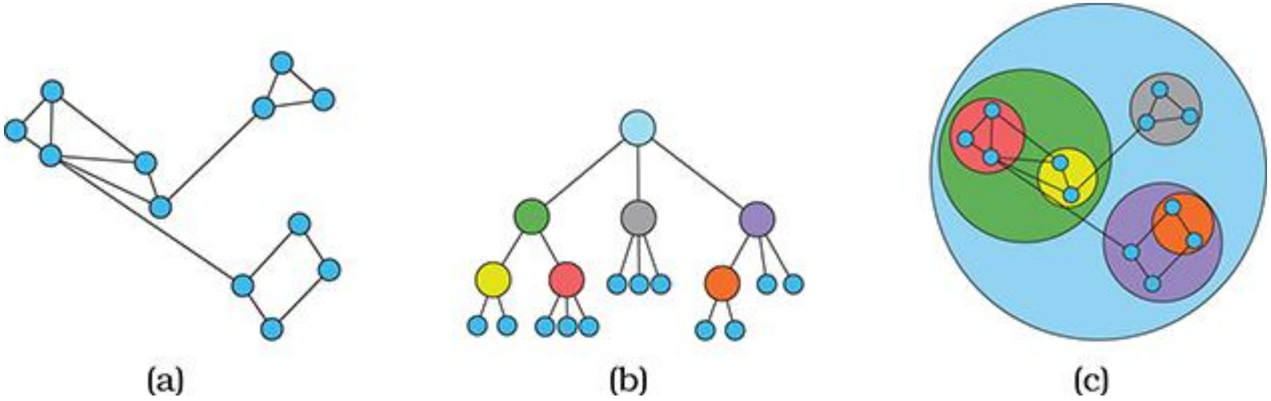
Other tree visual encoding idioms



- Containment: (e) and (f)
- Connection: (a) and (c)
- In others, spatial position channel is explicitly used to show the tree depth of a node.
    - The rectilinear icicle tree of (b) and the radial concentric circle tree of (d) show tree depth with one spatial dimension and parent–child relationships with the other.
    - The indented outline tree of (g) shows parent–child relationships with relative vertical position, in addition to tree depth with horizontal position.

Compound networks

- A **compound network** is the combination of a network and tree; that is, in addition to a base network with links that are pairwise relations between the network nodes, there is also a cluster hierarchy that groups the nodes hierarchically.
- In the GrouseFlocks system, users can investigate multiple possible hierarchies and they are shown explicitly. (a) shows a network and (b) shows a cluster hierarchy built on top of it. (c) shows a combined view using containment marks for the associated hierarchy and connection marks for the original network links.



Case study (optional)

- Note that displaying networks is a complex and task-dependent activity. Interaction plays an important role in supporting the user.
- Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Gord Davison, Ravin Balakrishnan. Expand-Ahead: A Space-Filling Strategy for Browsing Trees. Proceedings of IEEE Symposium on Information Visualization (InfoVis) 2004, pages 119-126. Paper available on Blackboard
- Video https://www.youtube.com/watch?v=-FkRzDegzAo

Summary

- Understand how to represent geographical data
- Understand how network and tree data can be represented
- Able to select appropriate methods for displaying non-tabular data for a particular task