1. (Multivariate) Network Visualization

Information Visualization (186.141)
TU Vienna, Austria
May 3 & 4, 2016
1.1 Motivation

Examples for networks and graph related data

- Molecular and genetic maps, biochemical pathways
- Object-oriented systems and data structures, scene graphs (VRML)
- Real-time systems (state diagrams)
- Semantic networks and knowledge representation diagrams
- Project management (PERT diagrams) or documentation management
- Social networks, co-authorship networks, food webs
- …
1.2 Definitions

Graphs are abstract structures, that can be used for modeling relational information.

Graph $G = (V, E)$

- $V$: Set of nodes (objects)
- $E$: Set of edges connecting nodes (relation)

Data structures:

Graph Drawing: automatic drawing of graphs in 2D and 3D

Adjacency matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Adjacency list

1: 2
2: 1, 3
3: 2
1.2 Definitions

Terminology

- Graphs can have *cycles*
- Edges can be *directed* or *undirected*
- The *degree* of a node is the number of edges that are connected with this node
  - At directed graphs
    - *In-degree* is the number of the incoming edges
    - *Out-degree* is the number of the outgoing edges
- Edges can have *values* (edge weights) with different types
1.2 Definitions

- Types of graphs
  - Trees
    - Properties
      - Special case of a graph
      - No cycles
      - Special root node
    - Free trees
    - Binary trees
    - Root trees
    - Ordered trees
  - Planar graphs
1.2 Definitions

- Types of graphs (cont.)
  - Directed/Undirected graphs
  - Extended graph models
    - Hierarchical graphs
    - Clustered graphs
    - Hypergraphs
    - ...
1.3 Graph Drawing

- Own research community
  - very large field!
  - I can only give an overview
  - A good starting point for literature search and further information are the annual *Graph Drawing* conferences (*GD*) or the IEEE *InfoVis* conferences

- Research areas of GD
  - Graph layout and positioning of nodes
  - Scalability
1.3 Graph Drawing

Research areas of GD (cont.)

- Navigation in large graphs
- Dynamic graphs
- Heterogeneous node and edge types
- Massive node degrees
- Visualization of isomorphic subgraphs
- (Embedding of additional information)
- (Focus & Context)
- (Comparison of graphs)
- ...

1. Network Vis.
Independent from layout and interaction techniques, there are many different possibilities to draw nodes and edges

- **Nodes**
  - Shape, color, size, position, label, …

- **Edges**
  - Color, size, thickness, direction, label, …
  - Shape
    - straight, curved, planar, orthogonal, …
1.3 Graph Drawing

Drawing Conventions

- Polyline Drawing
- Straight-line Drawing
- Orthogonal Drawing
- Grid Drawing
- Planar Drawing
- Upward Drawing
- Circular Drawing
- …

[Inspired by S. Hong und P. Eades’ course]
1.3.1 Aesthetics

A graph layout should be easy to read and to understand, easy to remember, as well as have a certain aesthetics

-less readable

-more readable

[taken from S. Hong und P. Eades’ course]
1.3.1 Aesthetics

- All layout algorithms fulfill more or less a set of *aesthetics criteria*

- Furthermore, the layout itself affects the perception of graphs

- **Problem:** These aesthetics criteria are sometimes contradictory and their computation mostly NP hard!

- Thus, the most GD algorithms are heuristics

[taken from S. Hong und P. Eades' course]
1.3.1 Aesthetics

**Aesthetics Criteria**

- Edge crossings ↓
- Area ↓
- Symmetry ↑
- Edge length ↓
  - Maximal edge length, uniform edge length, total edge length
- Bends of edges ↓
  - Maximal bends, uniform bends, total bends
- Resolution ↑
1.3.1 Aesthetics

- Example: crossings and bends

[taken from S. Hong und P. Eades’ course]
Example: Conflict between two criteria

Minimize edge crossings
Maximize symmetry

[taken from S. Hong und P. Eades’ course]
1.3.2 Force-directed GD

- Force-directed methods model nodes and edges as physical objects

- Examples
  - Spring forces for the edges
  - Gravitation forces for the nodes

- Aim is to find a stable configuration, that gets by with as few energy as possible

- We have here also optimization problems, that are solved locally
1.3.2 Force-directed GD

Spring Embedder

- Firstly presented by P. Eades, 1984
- Approach realizes two criteria
  - Symmetry
  - Uniform edge lengths

[taken from S. Hong und P. Eades’ course]
1.3.2 Force-directed GD

- Problems of the classic Spring Embedder algorithm is the high runtime and its (possibly) breakdown with very large graphs

- Layout example:
There are many improvements of this approach, e.g.:

- Inclusion of local, minimal energies
  - Algorithm of Kamada and Kawai, 1989

- Iterative, force-directed node positioning
  - Fruchterman and Rheingold, 1991

- Simulated Annealing
  - Davidson and Harel, 1996
In another approach, called *Layered GD*, the layout method firstly looks for a suitable layering that assigns each node an integer number.

Most methods compute on an extracted, acyclic subgraph that contains all nodes:

- A layer number is assigned to all nodes. Thus, the nodes are arranged top-down in rows, i.e., all nodes of an acyclic graph direct down.
- The placement (order) within the rows is used for the minimization of the number of edge crossings; mostly only until the next layer is reached.
- Even this problem is NP hard, i.e., one tries to find heuristics.
1.3.3 Layered GD

- Parallel layers
- Radial layers

[partly taken from S. Hong und P. Eades' course]
1.3.3 Layered GD

- Classic algorithm of Sugiyama, 1981

- Another heuristic defines a fixed order of the first and last layer and demands that each node is in the barycenter of its neighbors in the graph. This yields a linear system of equations

- Comparison of different heuristics, e.g.
1.3.4 Graph Drawing in 3D

- Challenge because of the growing size of real world networks: **Scalability**

- Solutions

  - Clustering
    - Collapse strongly connected nodes to super nodes
  
  - 3D (more space)
    - Classic 2D algorithms are extended to 3D
    - Problems
      - Navigation, massive overlaps, mental map, …
Example: 3D force-based layout
Example: 3D orthogonal layout

by D. Wood et al.
There are hundreds of applications (also in InfoVis) that use or extend classic GD techniques.

Tools available:
- JUNG, Gephi, Walrus, …
1.3.5 Applications

- Map layouts like MetroMaps
1.3.5 Applications

Drawing of syntax diagrams for context-free grammars

JSON grammar

http://arxiv.org/abs/1509.00818
During the past years, networks became more important that change with time, e.g.

- Biochemical networks have to be modified because of new discovered paths
- Social networks change through new contacts between people
- ...

Visualizations must preserve the „Mental Map“

- „Old structures“ should be recognized again
1.4.1 Morphing

There are several approaches to address this problem.

One of them is the so-called **Morphing**

**Idea**

- Visualize the transitions between two layouts using smooth animations

**Advantages**

- Looks very good (good aesthetics)

**Disadvantages**

- Nodes usually change their position
- Eventually, new added nodes or deleted nodes are not correctly identified
Morphing can be applied to each 2D/3D layout algorithm:

- If a node is changing its position in the new layout then compute an animation path between the old and the new position with the help of interpolation.

**Example system: GraphAEL**


- Here, mainly force-based methods are used.
1.4.1 Morphing

1. Network Vis.
1.4 Dynamic GD

http://graphael.cs.arizona.edu
If we know a sequence of graph in advance or if it is possible to precalculate it, then there is another method:

**Foresighted Graphlayout**

- [S. Diehl, C. Görg, and A. Kerren. „Preserving the Mental Map using Foresighted Layout“. In Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization, VisSym ’01, Springer Verlag, 2001.]

**Idea**

- Compute a supergraph based on the sequence of graphs
- Position the nodes at the beginning in such a way that they don’t change their positions later

http://cs.lnu.se/isovis/pubs/docs/VisSym01.pdf
1.4.2 Foresighted GD

Advantages

- Preserving the mental map
- Independent of the used graph layout algorithm

Disadvantages

- Sequence of graphs is often unknown
- Partly bad aesthetical results (gaps at the beginning, etc.)
If the graphs are represented as matrices, then we can stack the individual matrices to display a dynamic graph.

The *Matrix Cube* approach realizes the so-called space-time-cube metaphor:


**Idea**

- Matrix stack builds a cube (3D)
- Time is ordered along the slices (2D)
- Many interaction features support the analysis of dynamic graphs
1.4.3 Matrix Cubes

- Design space

![Design space diagram](http://www.aviz.fr/Research/Cubix)

- Example: trend analysis between vertex slices

Figure 3. Cubix View design space. Columns indicate operations applied to the cube. Rows indicate operations applied to time (red x red) and vertex slices (blue x red), respectively. (a) 3D view, (b) time-projection view, (c) vertex-projection view, (d) time small multiples, (e) vertex small multiples, (f) time-slice-rotation, and (g) vertex-slice-rotation.
1.4.3 Matrix Cubes

- Screenshot

-----

GRAPH:
Nodes: 11
Edges: 208
Times: 6
Cube Density: 0.034903847

VISUAL PARAMETERS:
fps: 0.0

-----

Cell Color Encoding:
- Time (blue to orange)
- Edge Weight (light to dark)
- None (all same gray)

Cell Shape:
- Edge Weight 1 (small to l...)
- Edge Weight 2 (small to l...)
- None (equal size)
- Adapt Weight
- Logarithmic scale

-----

ORDERING:
- Topological Order
- Name Ordering

-----

CELL VISIBILITY:
- Time Range: 0
- Edge weight: 6

-----

Vertex Slices
Time Slices

-----

SHOW:
- Show Self Edges
- Show Non-Self Edges

-----

Animation Speed:
- Slow
- Normal
- Fast

-----

http://www.aviz.fr/Research/Cubix
Aim of Information Visualization

- InfoVis is a research area that focuses on the use of visualization techniques to help people understand and analyze abstract data.

Comparing to Graph Drawing, the focus is not on the pure layout of a graph.

More important are

- Interacting with the graph visualization
- Exploring the possibly huge graph topology
- Adding of additional information (attributes) into the drawing
- Alternative representations, …
1.5.1 The Influence of InfoVis

What is a network comparing to a graph?

- Network = graph + attributed information to nodes and edges (also called multivariate network)

Just to give you some impressions, we will look to some specific aspects

- Special graph drawing techniques that support InfoVis tasks
  - Interactive exploration and clustering
- Multivariate network visualization
- Collaborative network visualization
Hierarchical Edge Bundles [Holten, InfoVis06]

- Avoid Clutter in Networks
1.5.2 Techniques for Special InfoVis Tasks

- New solution with possibility to change the blending strength

[Demo]
1.5.2 Techniques for Special InfoVis Tasks

**NodeTrix** [Henry et al., InfoVis07]

- Combined techniques for a better structuring

[Video]

(a) PARC Community  
(b) Ed Chi’s influence

[Image: http://www.aviz.fr/Research/NodeTrix]
Multivariate Data Integration

- Often, we have multivariate data attached to network elements ⇒ a node/edge has many attributes
  - Primary data: directly measured, …
  - Secondary data: derived, computed, …

- Current state-of-the-art solutions
1.5.3 Multivariate Networks

- **Integrated Approaches**
  (mostly replacing nodes by diagrams)
  [Junker et al., BMC Bioinformatics, 2006]

Bar Charts
1.5.3 Multivariate Networks

- **Multiple Coordinated Views**
  [R. Shannon et al., UCD TechRep 2008]
### 1.5.3 Multivariate Networks

#### Semantic Substrates


#### Idea

- Layout is based on user-defined semantic substrates
  - Non-overlapping regions for nodes
  - Node positioning is dependent on the attributes
- Slider in order to control the visibility of the edges. Thus, it is possible to simplify the edge clutter
1.5.3 Multivariate Networks

- Each region corresponds to a level of jurisdiction in the legal system of the US.
- Nodes correspond to the different cases (1978-2005); Node size corresponds to the number of references on that case.
- Edges correspond to the single references.
Attribute-Driven Topology (ADT) and Hybrid Approaches


Idea of JauntyNets

- Hybrid approach, but the core contribution is an extension of force-based layout algorithms → ADT
- Use of interaction and data mining techniques together

The benefit of the approach is that the user can decide if the graph topology or the multivariate attributes should get more attention
1.5.3 Multivariate Networks

- JauntyNets
1.5.3 Multivariate Networks

- JauntyNets

http://doi.ieeecomputersociety.org/10.1109/IV.2013.3
Huge and growing amounts of data in social sciences, life sciences, or software visualization

Experts sitting at different locations want to explore and analyze large networks together

- Providing suitable environments for collaborative settings is still an open challenge
- Collaborative tools should be easy to use on the fly, spontaneously and without setup overhead

There are four different scenarios for collaborative analyses (cf. next slide)
1.5.4 Collaborative Analysis

1.5.4 Collaborative Analysis

- Collaboration & Awareness Challenges
  - Who worked on the data previously?
  - Where did the user look at?
  - What changes were performed on the data?
  - Was it an expert or a novice user?

- Pointing & Reference Challenges
  - Participants of an analysis session can follow activities of each other

- Idea
  - Use heatmaps to show which nodes where watched/changed

---


http://dx.doi.org/10.1007/978-3-319-19890-3_37
1.5.4 Collaborative Analysis

1. Network Vis.
1.5 InfoVis ↔ GD

- **OnGrax Tool**

[Video]

http://dx.doi.org/10.1007/978-3-319-27261-0_21