2008-09

SI 508 - Networks: Theory and Application, Fall 2008

Adamic, Lada


<http://hdl.handle.net/2027.42/64962>
http://hdl.handle.net/2027.42/64962
SI 508/708 CS 608
Network visualization & GUESS
Outline

- Visualization
  - General tips for effective visualizations
  - Visualizing networks
    - layout algorithms
    - options for large networks
    - longitudinal data
    - visualization software besides Pajek & GUESS

- Exploratory data analysis
  - GUESS – the graph exploration system
Tips for effective visualizations

"The success of a visualization is based on deep knowledge and care about the substance, and the quality, relevance and integrity of the content."
(Tufte, 1983)

- know thy network!

Five Principles in the Theory of Graphic Display
- Above all else show the data.
- Maximize the data-ink ratio, within reason.
- Erase non-data ink, within reason.
- Erase redundant data-ink.
- Revise and edit.

Source: http://www.edwardtufte.com/tufte/
Aesthetic criteria for network visualizations

- minimize edge crossings

- uniform edge lengths
  - (connected nodes close together but not too close)

- don’t allow nodes to overlap with edges that are not incident on them
Cool looking visualizations are not always most informative


http://news.bbc.co.uk/2/hi/science/nature/2288621.stm

slide adapted from Katy Borner
Viewing a subset of the network and highlighting node attributes through shape and color enhances understanding.

An Attraction Network in a Fourth Grade Class (Moreno, 'Who shall survive?', 1934).

Alden Klovdahl: The core (n~ 450) of a social network of over 5,000 urban residents in Canberra, Australia
Overlaying a network on geographical context

byte traffic into the ANS/NSFnet T1 backbone for the month of September, 1991. Cox & Patterson, NCSA.

Walrus images of Skitter internet mapping data
Walrus is available under GPL
http://www.caida.org/tools/visualization/walrus/gallery1/
Longitudinal comparison

Sources:
1971 - "Casting the Net", page 64;
1980 - http://mappa.mundi.net/maps/maps_001/
http://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
Circular layout
IPv4 internet graph
AS-level internet map
copyright UC Regents 2004

Source: http://www.caida.org/research/topology/as_core_network/
What counts in a network visualization

- **Use of color**
  - Internet nodes were colored by outdegree
  - Edges colored by degree of endpoints
- **Use of meaningful coordinates**
  - Polar coordinates
    - $r$ – nodes with higher degree closer in
      - throws leaf nodes toward the outer edge of the graph
    - or distance from the most central node
    - position along ring denotes geographical longitude
- **Use of different sizes**
  - nodes sized by degree
- **What else is left?**
  - node shape
  - edge thickness
Random Layout

- Choose x & y coordinates at random
  - advantage: very fast
  - disadvantage: impossible to interpret

layout in GUESS
Circular layout

- Layout nodes along a circle and draw in all edges between them

- Advantages
  - Circular coordinates can represent a property of the data (e.g. latitude or ‘age’)
  - Very fast

- Disadvantages
  - difficult to interpret for large networks
    - many overlapping edges
    - many long edges (connected nodes need not be close together)
  - clusters hard to identify

layout in GUESS
Circular layout in GUESS

- `circleLayout(edge_weight, center_node)`
- Place all nodes on a circle
- Place center node in the middle
- Place center node’s neighbors in a circle around at a radius depending on the weight of the edge

image: Andrea Wiggins
http://www.andreawiggins.com/work.html
Radial Layout

- Start with one node, draw all other nodes in circular layers according to how many hops it takes to reach them
Spring embedding algorithms

- Two parts
  - Force (or energy) model that quantifies the quality of drawing
  - Optimization algorithm that computes a network configuration that is locally optimal with respect to this algorithm
- Final layout depends on starting positions
  - Simulated annealing introduces randomness to help the algorithm find global minima
- At equilibrium, the force on each vertex is 0
“manual” spring layouts

Grant's Drawing of a Target Sociogram of a First Grade Class (from Northway, 1952).

McKenzie's Target Sociogram Board (from Northway, 1952).

Pegs and rubber bands used to determine an individual’s location in the sociogram.
computerized spring layouts

- Iterative procedure
- At each time step, allow springs to expand or contract toward a neutral position

select optimal edge length (node distance) $k$

repeat
  for each node $v$ do
    for each pair of nodes $(u, v)$
      compute repulsive force $f_r(u,v) = -c \cdot$
    for each edge $e = (u,v)$
      compute attractive force $f_a(u,v) = c \cdot$
    sum all force vectors $F(v) = \sum f_r(u,v) + \sum f_a(u,v)$
    move node $v$ according to $F(v)$
  until DONE
Spring layout algorithms: Fruchterman and Reingold

- Model roughly corresponds to electrostatic attraction between connected nodes
- Use adjacency matrix directly
- Iterative optimization
  - at each step, every node reacts to the pulls and pushes of the springs that tie it to all the other nodes
- Can be slow as the network grows

layout in GUESS
Spring layout algorithms: Kamada Kawai

- All nodes are connected by springs with a resting length proportional to the length of the shortest path between them.
- Need to calculate all pairs shortest paths first.
- Iterative optimization.
- Advantage: can be used on edge-weighted graphs.
- Can be slow as the network grows.

layout in GUESS
Spring layout algorithms: GraphOpt

- Another physics approach with springs and electrostatic charges
- Iterative optimization
- Layering:
  - nodes assigned ‘layers’ based on relative positions
  - hide nodes in lower layers
  - lay out higher level nodes
- Advantage: can be used on somewhat larger graphs
- Can be slow as the network grows
There are many variations on spring layout algorithms...
Java applet demo of a spring layout

http://java.sun.com/applets/jdk/1.4/demo/applets/GraphLayout/example1.html
GEM (graph embedding) Layout

- Embedding algorithm with speed & layout optimizations
- Significantly faster than KK or FR
- In GUESS, you can lay out 1,000 – 10,000 node graphs, depending on the edge density
Multidimensional scaling concept

- Metric MDS gives an exact solution based on a Singular Value Decomposition of the input matrix.
- Input matrix can be the all pairs shortest path or another ‘distance matrix’
- Usually the data is plotted according to the eigenvectors corresponding to the two largest eigenvalues
Strategies for visualizing large graphs

- Reduce the number of nodes and edges
  - introduce thresholds
    - only authors who have written at least $x$ papers
    - only edges with weight $> y$
    - only nodes with degree $> z$ (e.g. removing leaf nodes)
  - show minimum spanning trees
    - can visualize all the nodes with a subset of the edges
  - use pathfinder network scaling (http://iv.slis.indiana.edu/sw/pfnet.html)
    - triangle inequality to eliminate redundant or counter-intuitive links
    - remaining edges are more representative of internode relationships than minimum spanning trees
- collapse nodes into clusters
  - show multiple nodes as a single node
  - display connections between clusters
  - e.g. displaying the internet graph on the autonomous system level rather than the individual router level
From the Pajek manual: approaches to deal with large networks

Example of coarsening network structure

- Newman & Girvan 2004
- co-authorship network of physicists writing papers on networks
- clustering algorithm identifies different subcommunities
- each node is a community – size represents number of authors
- each edge thickness represents the number of co-author pairs between communities

Zoomable interfaces

- GUESS lays out networks on an infinite plane that one can zoom in and out of (demo)
- hyperbolic browser (InXight demo): 
  - map a hyperbolic plane onto a circular layout
  - in a hyperbolic plane each child node gets as much space as its parent
  - focus of hyperbolic plane is displayed in the middle of a unit circle
  - rest fades off-perspective toward the edge of the disk
  - in the browser, change focus by clicking on node to bring it to the center
  - good for visualizing large hierarchies
- another demo with Lexis-Nexis: 
  http://www.lexisnexis.com/startree/interactiveview.asp
Displaying longitudinal data through animation

- Nodes should move little between different timepoints to make it easier to track them
- Most people can track 3-7 objects simultaneously (your network can have hundreds or more)

http://ruccs.rutgers.edu/finstlab/motMovies/mot.mov

http://graphexploration.cond.org/sample.mov
graphs over time

- consider keeping nodes in the same place, but having them appear/disappear....

example: information diffusion on a social network
Mark Lombadi’s (hand-drawn) networks
What else could be added to this visualization?

source: James Moody, *Race, School Integration, and Friendship Segregation in America*
AJS Volume 107 Number 3 (November 2001): 679–716
What else could be added to this visualization?

source: James Moody, *Race, School Integration, and Friendship Segregation in America*
AJS Volume 107 Number 3 (November 2001): 679–716
Visualizing attributes (gender)


GUESS
The Graph Exploration System

Eytan Adar

November 23, 2005
Design requirements

- Deal with different kinds of networks
  - But not by abstracting everything to a matrix
  - Nodes and edges have properties!
- Exploratory tool
  - Tolerate mistakes made in exploration
- Ability to easily do standard analysis
- Ability to add new analysis routines
- Scriptable
- Compile into application/applet
- Flexible front/back ends
- guess.bat (Windows)
- guess.sh (Mac)
Screenshot
“Gython”
- Python + graph data structures + operators + query language

Better (expandable/separable) architecture
- Back-end storage abstracted
- Front-end visualization abstracted
  - Prefuse
  - Touchgraph
  - Still have one main “zoomable” front end
    - The most complete
Query language built in

- Nodes and Edges have properties
  - The usual types (text, numbers, Booleans)
- Can use these to manipulate the display
  - (dept == ‘Human Resources’).color = blue
  - (freq > 10).width = 4
  - (cell_location == ‘wall’) & (expression_levels > 100)
  - (name like ‘Bob%’)
Getting data in

- GUESS lets you define your own properties

```
node_def> name, country VARCHAR
N1,"US"
N2,"France"
edge_def> node1,node2, delay INT default 5
N1,N2,20
```
GUESS knows about visual properties
- Nodes
  - location, color, size, shape, label, etc.
- Edges
  - width, color, etc.
(Non-visual) properties generated dynamically
- e.g. indegree, pagerank, betweeness
Everything accessed same way
- v3.color  v3.dept  v3.indegree
**Visual shortcuts**

- Lots of syntactic sugar to do certain things
  - Color each department differently
    - `colorize(dept)`
  - Color each edge by frequency from red to blue
    - `colorize(freq,red,blue)`

- Can group and sort by properties
  - `depts = groupBy(dept)`
  - `freqs = sortBy(freq)`
  - `whatever = groupAndSortBy(…)`
Built in functions

- Layouts
- Clustering algorithms
- Shortest path/Flow algorithms
- Centrality measures
- Graph statistics
- Plots and charts

- Can even connect to R for more
Connect interpreter to display

- Unique feature of GUESS
- Mouse motion over text results in highlighting of graph/visualization structures
- $[[v4,v5],[v6,v7,v8]]$
States and Time

- As if graphs weren’t complicated enough…
  - Time is a critical dimension
  - Graphs and properties change
  - We want to visualize them

- And users in an exploratory mode want undo

- Kill two birds with one stone…
States and Time

- Basics through simple commands
  - ss('state name')
  - ls('state name')
- Queries work between states
  - v44['q105'].dept
  - freq[2005] > freq[2003]
- Morphing
  - morph('state name', time)
  - output as movie
  - Camera tracking (in Zoomgraph and soon in GUESS)
- Also… “range” fields
  - “1,5-100,102-105”
  - Node rcontains (5,10)
  - Node rexact (102-105)
Extending GUESS

- Write your own routines/programs
- Change mouseover/click behavior
  - E.g. pop up a web page
- Control remotely or through Java
- Add “dockable” widgets
- Replace front end
- Compile into applet
Simple Example: Skitter

\[ \text{radius} = 1 - \log \left( \frac{\text{outdegree}(\text{AS}) + 1}{\text{maximum.outdegree} + 1} \right) \]

\[ \theta = \left( \frac{\text{longitude of the AS headquarters in whois records}}{} \right) \]

Source: http://www.caida.org/research/topology/as_core_network/
def skitter(_field):
    _maxangle = 2 * Math.PI
    _ordering = sortBy(_field)
    _increment = _maxangle / len(_ordering)
    _curangle = 0
    g.nodes[0].outdegree
    _maxdeg = outdegree.max + 1.0
    for _n in _ordering:
        _radius = 1 - Math.log((_n.outdegree + 1.0) / _maxdeg)
        _radius = _radius * 500.0
        _x = 500.0 + _radius * Math.cos(_curangle)
        _y = 500.0 + _radius * Math.sin(_curangle)
        _n.setX(_x)
        _n.setY(_y)
        _curangle += _increment
Skitter
Modify the interface

import ...

class dockexample1(com.hp.hpl.guess.ui.DockableAdapter):

    def __init__(self):
        testButton = JButton("center")
        action = lambda event: center()
        testButton.actionPerformed = action
        self.add(testButton)

    def getTitle(self):
        return("dockexample1")
def sc(self,evt):
    val = self.testSlider.getValue()
    g.nodes.visible = 1
    (freq < val).visible = 0
    (freq >= val).visible = 1
    self.hideDisconnectedNodes()
    self.label.setText("Frequency threshold ("+str(val)+")")
import ...

class dockexample2(com.hp.hpl.guess.ui.DockableAdapter):
    testSlider = JSlider()
    label = JLabel("Frequency threshold (0)   ")

    def __init__(self):
        self.testSlider.setMinimum(freq.min)
        ...
        self.testSlider.setValue(freq.min)  # default value

        self.testSlider.mouseReleased = self.sc
        self.add(self.label)
        self.add(self.testSlider)
        ui.dock(self)
def hideDisconnectedNodes(self):
    toHide = []
    for nod in g.nodes:  # for all nodes
        vis = 0  # default to invisible
        for ed in nod.getOutEdges():
            if (ed.visible == 1):
                vis = 1
                break
        if (vis == 0):  # should we hide the node?
            toHide += [nod]
    # hide all the nodes we put in our list
    toHide.visible = 0
Compiling and distributing...

- Users build applets/applications
  - Network simulation
  - Political blogs
  - Neuroscience and sewer/water lines

- Discussion group:
  - Guess-discuss on google groups
Front end flexibility

- Can replace the visualization
  - Eytan likes Piccolo
  - But...
    - Prefuse
    - Touchgraph
    - JUNG
    - and soon Wilma (3D)
  - Are also available
Scaling...

**Not bad...**
- Graphics will slow you down
- Algorithms are pretty fast

**You can...**
- Load up a big dataset
- Do a faster layout (gemLayout())
- Go to lunch
- Play with graph
~6000 nodes
~12000 nodes
Viral marketing

Social groups

Stanford personal homepages, ca. 1999

MIT personal homepages, ca. 1999

Email communications

Information Flow

Giant Microbes
(http://www.giantmicrobes.com)

CNN story on Walmart
(http://money.cnn.com/2003/05/06/news/companies/walmart_mags/index.htm)

Summary... (end Eytan’s slides)

- Exploratory data analysis

- Free (GPL)
  - http://www.graphexploration.org
lab

- discover the citation patterns between political blogs using Guess