OSU CS419/519 Document Analysis

Social Network Analysis

Link Prediction and Network Visualization

Marian-Andrei Rizoiu, Lexing Xie
Computer Science, ANU

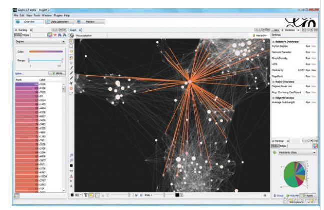
Lecture slides credit: Lada Adamic, Univ. Michigan, Jure Leskovec, Stanford University

Plan for today

- An introduction to link prediction in networks

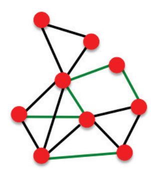
  David Liben-Nowell and Jon Kleinberg.
  The Link-Prediction Problem for Social Networks.
  In Journal of the American Society for Information Science and Technology, 58(7):1019–1031, May 2007
  http://cs.carleton.edu/faculty/dlibenno/papers/link-prediction/link.pdf
  earlier version published in CIKM 2003

- Network Visualization with Gephi



Link Prediction in Networks

- **The link prediction task:**
  - Given $G[t_0,t_0']$ a graph on edges up to time t_0' **output a ranked list L** of links (not in $G[t_0,t_0']$) that are predicted to appear in $G[t_1,t_1']$



- **Evaluation:**
  - $n = |E_{new}|$: # new edges that appear during the test period $[t_1,t_1']$
  - Take top n elements of L and count correct edges

Link Prediction via Proximity

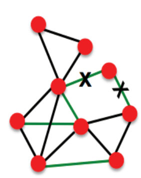
- **Predict links in a evolving collaboration network**

	training period			Core		
	authors	papers	collaborations[1]	authors	$\lvert E_{old}\rvert$	$\lvert E_{new}\rvert$
astro-ph	5343	5816	41852	1561	6178	5751
cond-mat	5469	6700	19881	1253	1899	1150
gr-qc	2122	3287	5724	486	519	400
hep-ph	5414	10254	47806	1790	6654	3294
hep-th	5241	9498	15842	1438	2311	1576

- **Core:** Since network data is very sparse
  - Consider only nodes with in-degree and out-degree of at least 3

Link Prediction via Proximity

- **Methodology:**
  - For each pair of nodes *(x,y)* compute score *c(x,y)*
    - For example: # of common neighbors *c(x,y)* of *x* and *y*
  - Sort pairs *(x,y)* by the decreasing score *c(x,y)*
    - **Note:** Only consider/predict edges where both endpoints are in the core (*deg.* > *3*)
  - **Predict top *n* pairs as new links**
  - **See which of these links actually appear in $G[t_1, t'_1]$**



Link Prediction via Proximity

- **Different scoring functions *c(x,y)***
  - **Graph distance:** (negated) Shortest path length
  - **Common neighbors:** $|\Gamma(x) \cap \Gamma(y)|$
  - **Jaccard's coefficient:** $|\Gamma(x) \cap \Gamma(y)|/|\Gamma(x) \cup \Gamma(y)|$
  - **Adamic/Adar:** $\sum_{z \in \Gamma(x) \cap \Gamma(y)} 1/\log|\Gamma(z)|$
  - **Preferential attachment:** $|\Gamma(x)| \cdot |\Gamma(y)|$
  - **PageRank:** $r_x(y) + r_y(x)$
    - $r_x(y)$ … stationary distribution weight of *y* under the random walk:
      - with prob. 0.15, jump to *x*
      - with prob. 0.85, go to random neighbor of current node

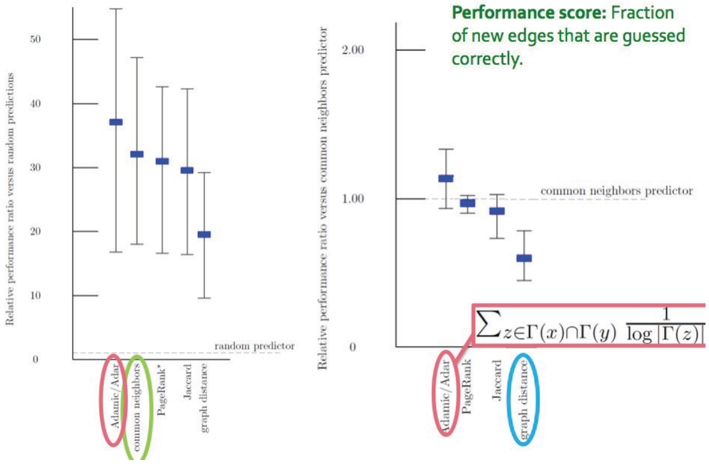
$\Gamma(x)$ … neighbors of node x

- **Then, for a particular choice of *c(·)***
  - For every pair of nodes *(x,y)* compute *c(x,y)*
  - Sort pairs *(x,y)* by the decreasing score *c(x,y)*
  - **Predict top *n* pairs as new links**

[LibenNowell-Kleinberg ' 03]

Results

Performance score: Fraction of new edges that are guessed correctly.



Link Prediction

- One useful task
  - –Can be tackled with 3 hrs of SNA primer
  - –Simple scoring preforms reasonably well
  - –Lots of possible scoring functions
- How can this be improved?

Network visualization

- How to make sense of a large network?
  1000s of nodes
  10,000s of nodes
  1,000,000 of nodes?
- What are efficient methods for displaying information about a network, manipulating it, and zooming in to reveal insight?

Tools for analyzing social network (non-exhaustive)

Gephi (visualization and basic network metrics)

NetworkX – programming in Python

- extensive functionality
- scales to large networks by taking advantage of existing C, Fortran libraries for large matrix computations
- open source
- http://networkx.lanl.gov/

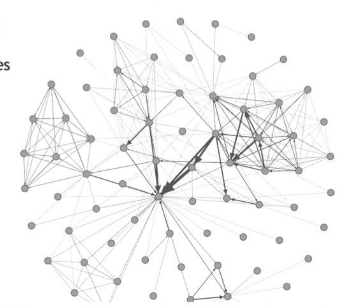
Elements of Graph Visualization with Gephi

- **Data** – graph structure, node labels, edge properties …
- **Graph Layout**
- **Styling** – color/size/font for nodes and edges

Fruchterman-Reingold layout

It simulates the graph as a system of mass particles. The nodes are the mass particles and the edges are springs between the particles. The algorithms try to minimize the energy of this physical system. It has become a standard but remains very slow.

Author:	Thomas Fruchterman & Edward Reingold[1]
Date:	1991
Kind:	Force-directed
Complexity:	$O(N^2)$
Graph size:	1 to 1 000 nodes
Use edge weight:	No



[1] Fruchterman, T. M. J., & Reingold, E. M. (1991). Graph Drawing by Force-Directed Placement. Software: Practice and Experience, 21(11).

ForceAtlas layout

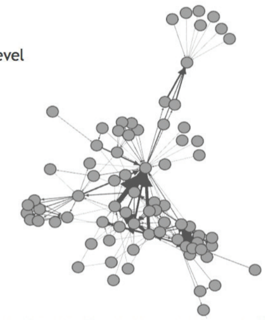
Home-brew layout of Gephi, it is made to spatialize Small-World / Scale-free networks. It is focused on quality (meaning "being useful to explore real data") to allow a rigorous interpretation of the graph (e.g. in SNA) with the fewest biases possible, and a good readability even if it is slow.

Author:	Mathieu Jacomy
Date:	2007
Kind:	Force-directed
Complexity:	$O(N^2)$
Graph size:	1 to 10 000 nodes
Use edge weight:	Yes

Yifan Hu Multilevel layout

It is a very fast algorithm with a good quality on large graphs. It combines a force-directed model with a graph coarsening technique (multilevel algorithm) to reduce the complexity. The repulsive forces on one node from a cluster of distant nodes are approximated by a Barnes-Hut calculation, which treats them as one super-node. It stops automatically.

Author:	Yifan Hu[1]
Date:	2005
Kind:	Force-directed + multilevel
Complexity:	O(N*log(N))
Graph size:	100 to 100 000 nodes
Use edge weight:	No



[1] Y. F. Hu, Efficient and high quality force-directed graph drawing. The Mathematica Journal, 10 (37-71), 2005.

So how to choose a layout?

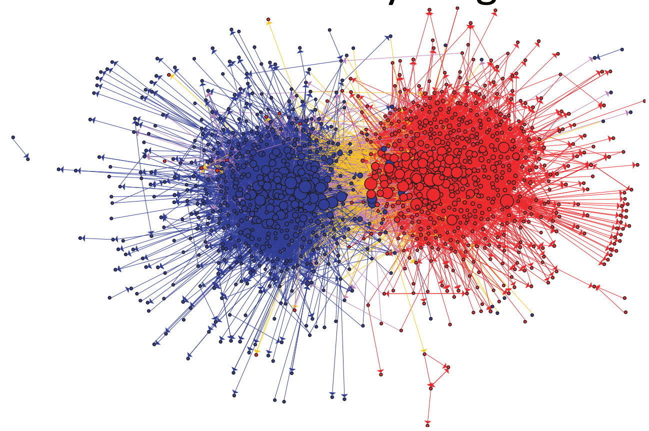
In general, select one according to the feature of the topology you want to highlight:

emphasis DIVISIONS *OpenOrd*	emphasis COMPLEMENTARITIES *ForceAtlas, Yifan Hu, Frushterman-Reingold*
emphasis RANKING *Circular, Radial Axis*	emphasis GEOGRAPHIC REPARTITION *GeoLayout*

Graphic Adjustements

- Label Adjust
- Noverlap
- Expansion
- Contraction

"Divided They Blog"



Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 U.S. election: divided they blog. In Proceedings of the 3rd international workshop on Link discovery (LinkKDD '05)

- Gephi comes with great tutorials
  - http://gephi.org/users/
  - esp http://gephi.org/tutorials/gephi-tutorial-layouts.pdf
- Lots of good, online walk-throughs
  - http://www.martingrandjean.ch/introduction-to-network-visualization-gephi/

Other Options for Network Visualization

- GraphViz – underlying engine for many
- Pajek
- NodeXL http://nodexl.codeplex.com/
- Network package + visualization
  - NetworkX/Python (example in your tutorial)
  - iGraph / R
- Web applications
  - D3.js
  - Sigma.js
- … …