

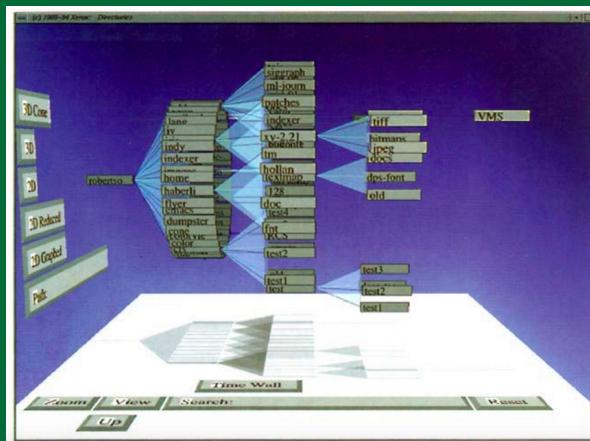
**DSBA 5122: Visual Analytics**

**Class 10: Networks and Trees**

Ryan Wesslen

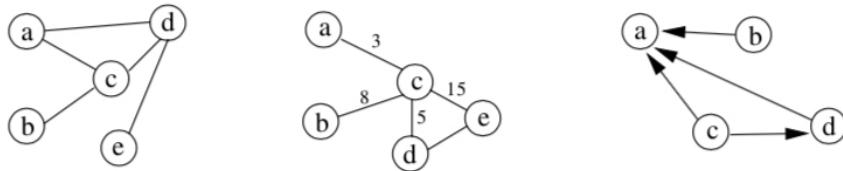
April 1, 2019

# Networks and Trees



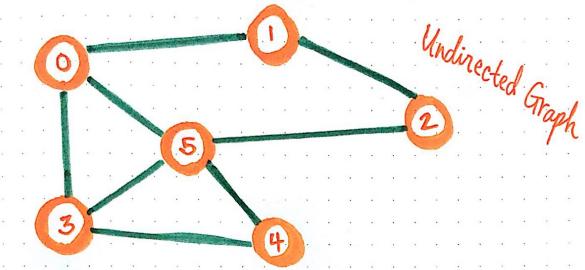
Read Chapter 5: Networks and Hierarchies of Mazza

# Networks (a.k.a. graphs)



**Fig. 5.1** Examples of graphs. On the left is a normal graph, in the center is a graph in which each edge is given a numerical value, and to the right is a directed graph.

- Graphs are visual representations in which the points, called **nodes** or **vertices**, represent instances of the data.
- Nodes are correlated by connections, called **edges**, which represent relationships between the instances.
- Possible features of a network: weights, direction, labels.



E	0	0	1
D	1	0	3
G	2	0	5
E	3	1	2
L	4	2	5
I	5	3	4
S	6	3	5
T	7	4	5

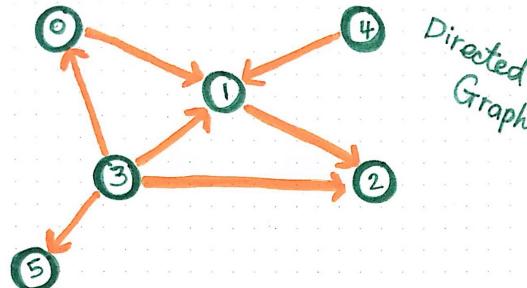
	0	1	2	3	4	5
0	0	1	0	1	0	1
1	0	0	1	0	0	0
2	1	0	0	0	0	0
3	2	0	1	0	0	0
4	3	1	0	0	0	0
5	4	0	0	0	1	0
6	5	1	0	0	0	0
7	6	0	1	1	1	0

A  
D  
M  
J  
A  
C  
E  
R  
N  
Y

0	1	3	5
1	0	2	
2	1	5	
3	0	4	5
4	3	5	
5	0	2	3
6			4

ADJACENCY LIST

notice that each edge appears twice ; 2's adjacency list contains a reference to 5, and 5's list correspondingly has a reference to 2 ! thus, there are  $2|E|$  elements.



EDGE LIST	0	0	1
0	1	1	2
1	2	3	0
2	3	3	1
3	4	3	2
4	5	3	5
5	6	4	1

EDGE LIST	0	0	1	2	3	4	5
0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0
4	3	1	1	1	0	0	0
5	4	0	1	0	0	0	0
6	0	0	0	0	0	0	0

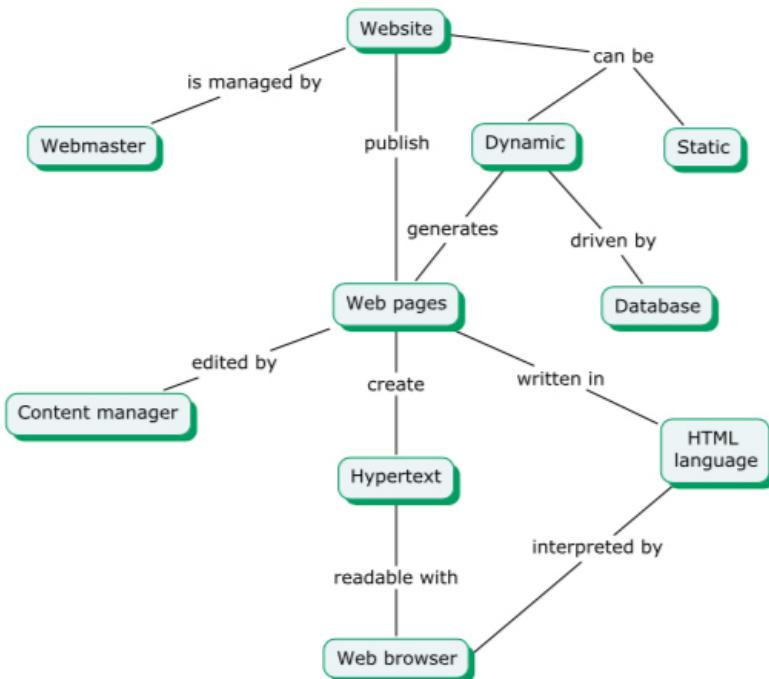
↑ notice that this matrix is not symmetrical!

0	1
1	2
2	
3	0
4	1
5	

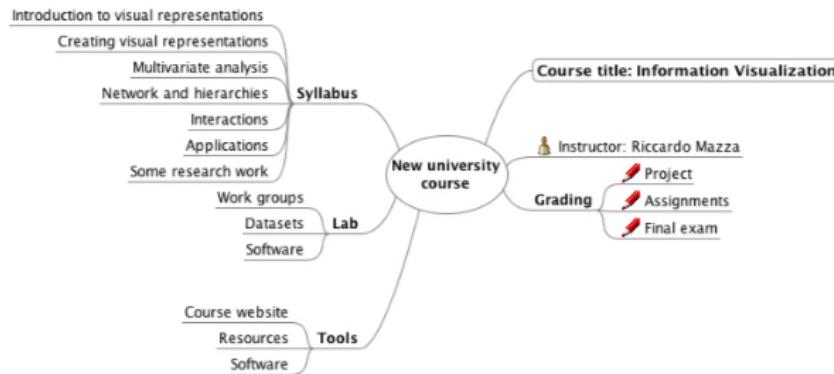
ADJACENCY LIST

↑ notice that this adjacency list contains a total of  $|E|$  elements, one for each edge, since the edges are directed.

Vaidehi Joshi Medium post

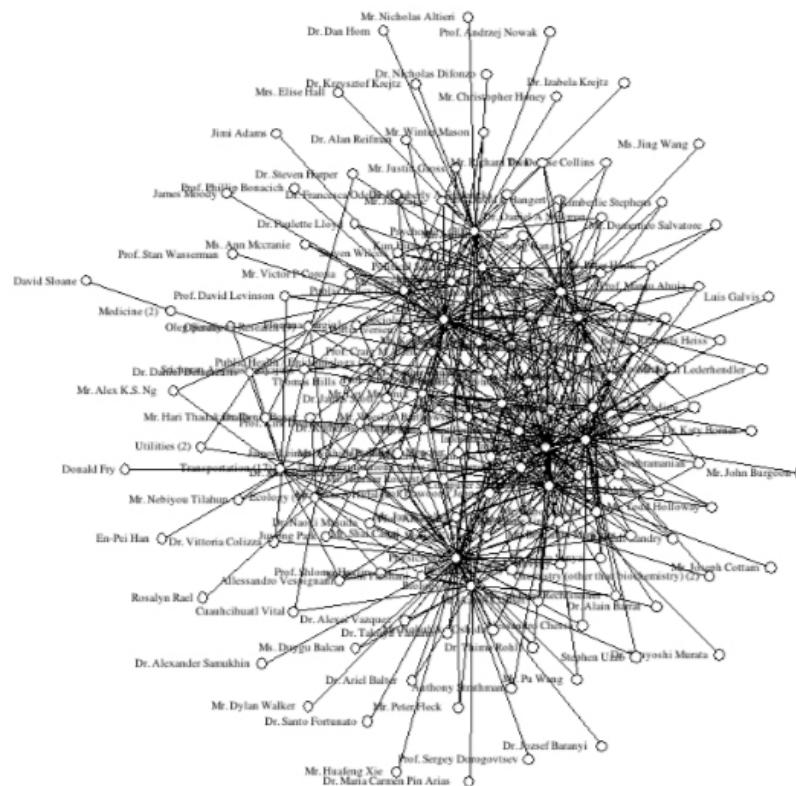


**Fig. 5.2** Example of a concept map that describes concepts regarding a website.

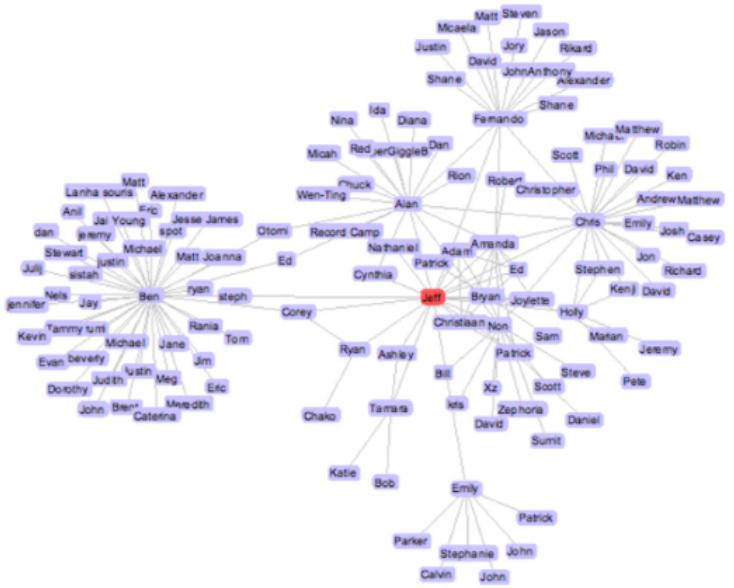


**Fig. 5.3** Example of mind map for the planning of a university course.

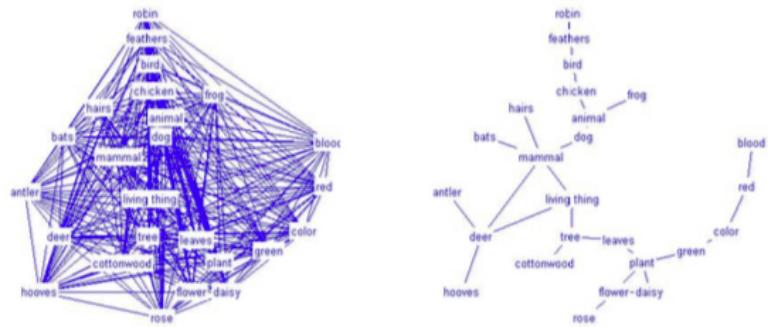
## BNOSAC: R NLP packages ecosystem



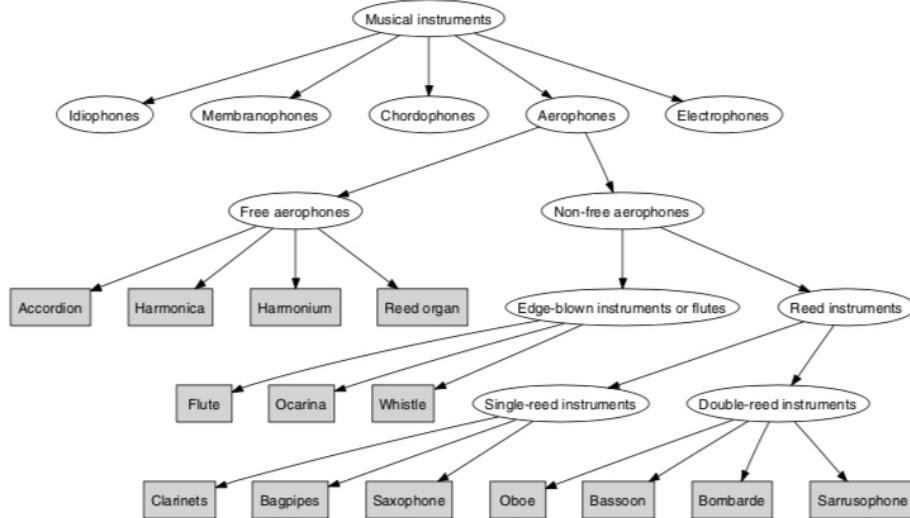
**Fig. 5.4** An example of a complex graph.



**Fig. 5.5** An example of a graph that uses a force-directed algorithm to represent a social network. Image created with the prefuse tool and reproduced with the permission of Jeffrey Heer, University of California, Berkeley.

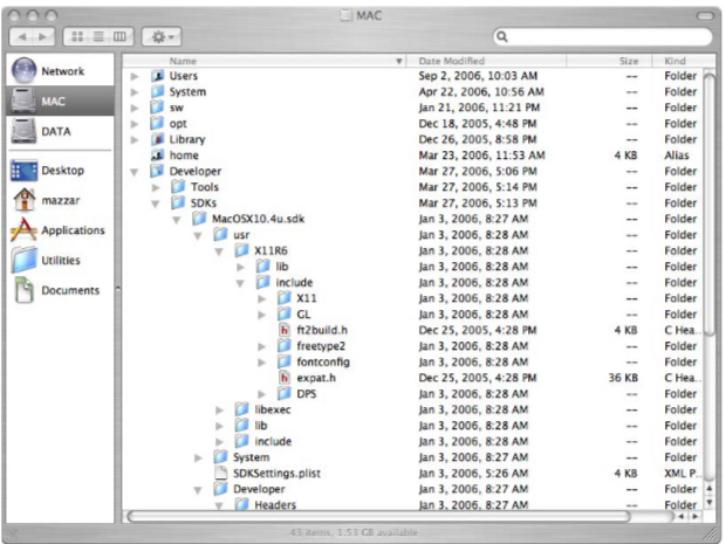


**Fig. 5.6** Complete graph (left) and the reduced link version (right), achieved using the pathfinder technique. Graphs were generated with the KNOT analysis tool. Images courtesy of Interlink.



**Fig. 5.13** A tree representing the classification of the wind instruments according to Curt Sachs and Erich von Hornbostel.

- A **tree** (hierarchy) can be represented through a graph with a starting node called **root**.
- Each node has zero or more child nodes its ancestor is called the **parent** node. A node has at most one parent.



**Fig. 5.14** Representation of the file system by the graphical interface Finder in the Mac OS X system.

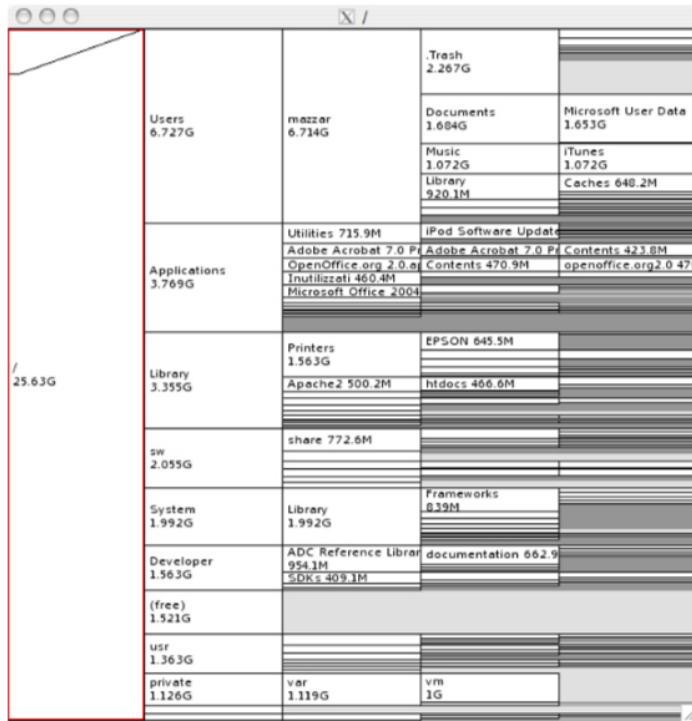
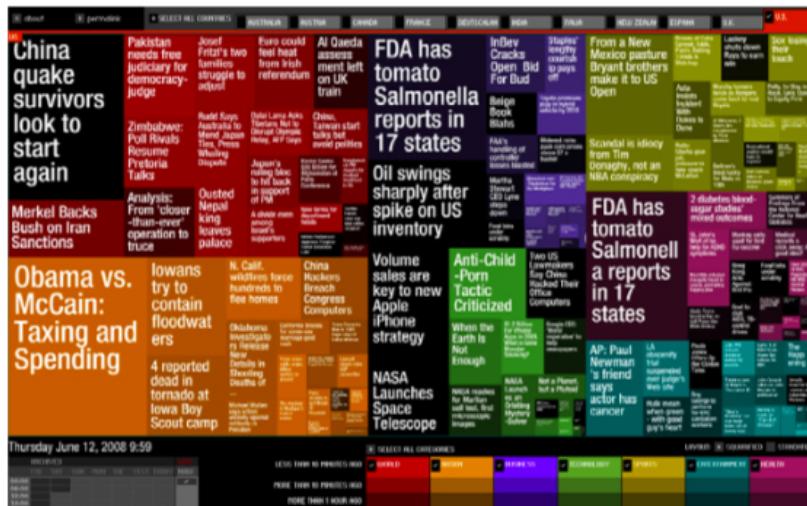
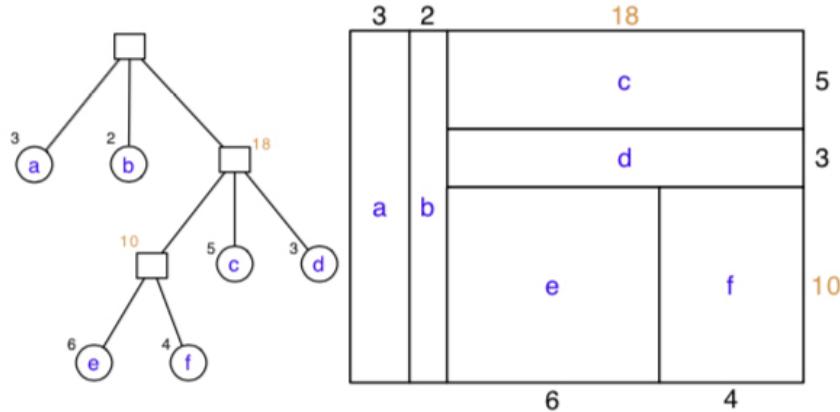


Fig. 5.15 Xdiskusage uses a representation with rectangles of the file system.



**Fig. 5.22** Newsmap uses a treemap algorithm to represent news from Google News U.S. on June 12, 2008. Image reproduced with the permission of Marcos Weskamp.



**Fig. 5.20** On the left is an example of a tree representation, where a numerical value is associated with each leaf. The internal nodes report the sum of the values of the nodes below. On the right is the treemap representation of the tree.

# Katya Ognyanova's "Network visualization with R."

The screenshot shows the homepage of Katya Ognyanova's website. The header features a large, abstract network visualization with red and grey nodes and connecting lines. Below the header, the name "Katya Ognyanova" is displayed. A navigation bar includes links for Home, Blog, Bio, Research, Publications, Teaching, Tutorials, Media, and a search icon. The main content area is titled "Static and dynamic network visualization with R". It includes a timestamp ("June 14th, 2017"), a list of tags, and a detailed description of the tutorial's content. A note about a June 2018 update is present, along with citation information and a loading notice.

**Static and dynamic network visualization with R**

June 14th, 2017 | Tags: animation, centrality, D3, dataset, graph, igraph, JavaScript, map, ndtv, network, network analysis, network science, network tutorial, network visualization, networkD3, node, plot, R, reciprocity, RStudio, sna, Statnet, threejs, transitivity, visNetwork, visualization

This is a comprehensive tutorial on network visualization with R. It covers data input and formats, visualization basics, parameters and layouts for one-mode and bipartite graphs; dealing with multiplex links; interactive and animated visualization for longitudinal networks; and visualizing networks on geographic maps. To follow the tutorial, download the code and data below and use R and RStudio. You can also check out the most recent versions of all my tutorials here.

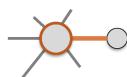
[June 2018 update] The tutorial is continuously updated and expanded. If you want to see earlier versions, they are still available here: 2015, 2016, and 2017. You can also get the new tutorial PDF and code here or on GitHub.

If you find the tutorial useful, please cite it in your work – this helps me make the case that open publishing of digital materials like this is a meaningful academic contribution: Ognyanova, K. (2018) *Network visualization with R*. Retrieved from [www.kateto.net/network-visualization](http://www.kateto.net/network-visualization).

Please be patient: the frame below contains a lot of images and javascript animations and may take a bit to load.

## Network visualization goals

Key actors and links



Relationship strength



Structural properties



Communities



Diffusion patterns



Network evolution



Networks as maps



Networks as persuasion

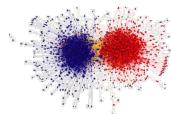


Networks as art



## Some network visualization types

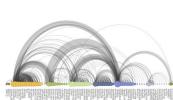
Network Maps



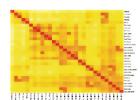
Statistical charts



Arc diagrams



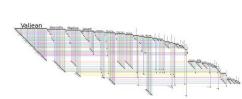
Heat maps



Hive plots



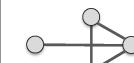
Biofabric



## Layout aesthetics

### Minimize edge crossing

No

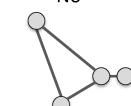


Yes



### Uniform edge length

No



Yes

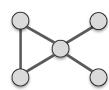


### Prevent overlap

No

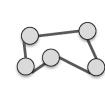


Yes



### Symmetry

No



Yes

