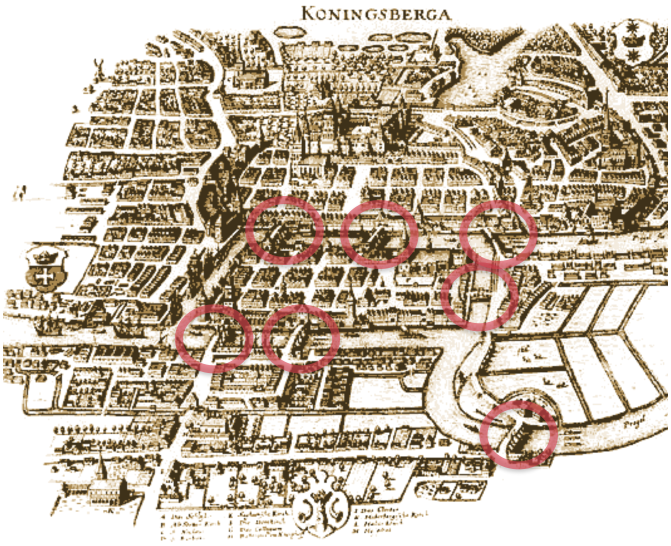
Graph Theory basics

Prof. Pascal Frossard
Signal Processing Laboratory (LTS4)

Outline

- *Networks and graphs*
- Definitions
- Network density
- Pathology
- Connectedness

Back to the XVIIIth century



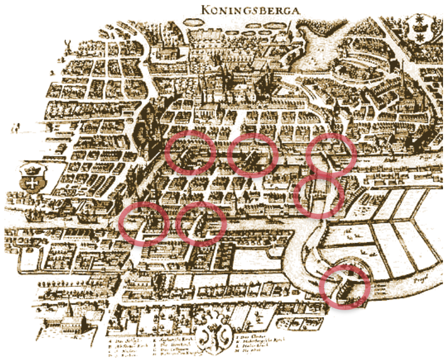
Leonhard Euler [1736]

Can one walk across the seven bridges and never cross the same bridge twice?

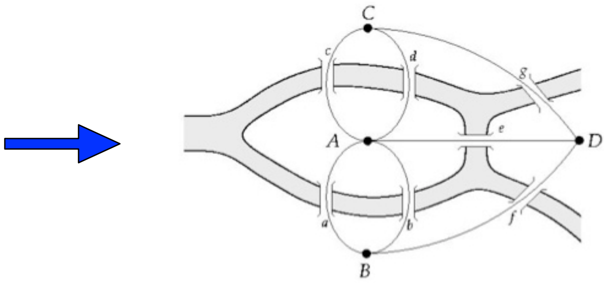
The seven bridges problem

From [1]

Euler's solution



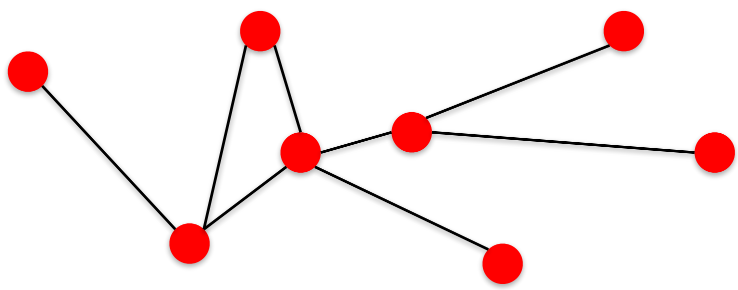
Key representation idea!



- Euler's theorem:
  - If a graph has more than two nodes of odd degree, there is no path
  - If a graph is connected and has no odd degree nodes, it has at least one path

Graph theory offers many analysis tools to use networks for all kinds of applications: from clustering to classification, visualization, recommendation, etc.

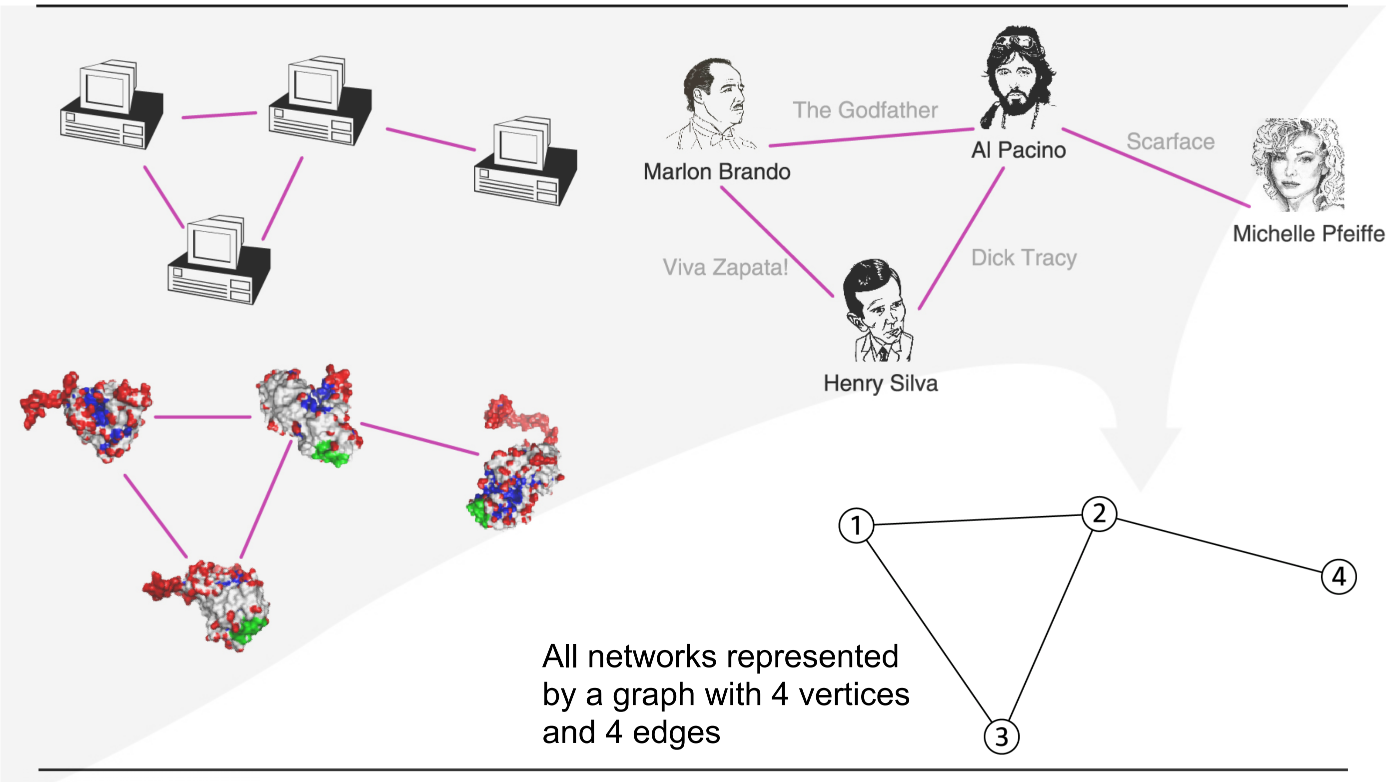
Networks and graphs



- Elements of a complex system
  - Nodes $\mathcal{N}$ or vertices $\mathcal{V}$ represent the components of the system
  - Edges $\mathcal{E}$ or links $\mathcal{L}$ capture the interactions between components
  - Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ or $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ describes the system

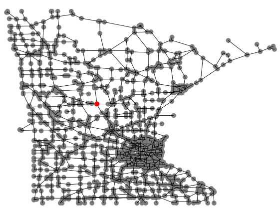
A *network* generally refers to the real system, a *graph* is its mathematical description.

Generic description



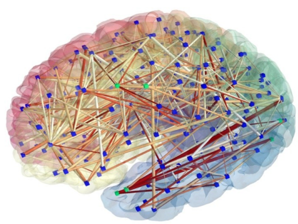
All networks represented by a graph with 4 vertices and 4 edges

Network examples

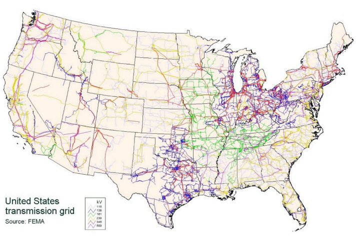


Minnesota Road Network

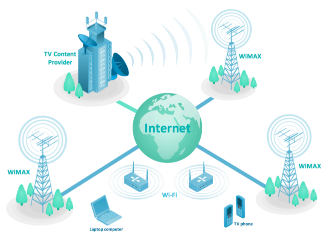
Facebook



Brain Connectivity



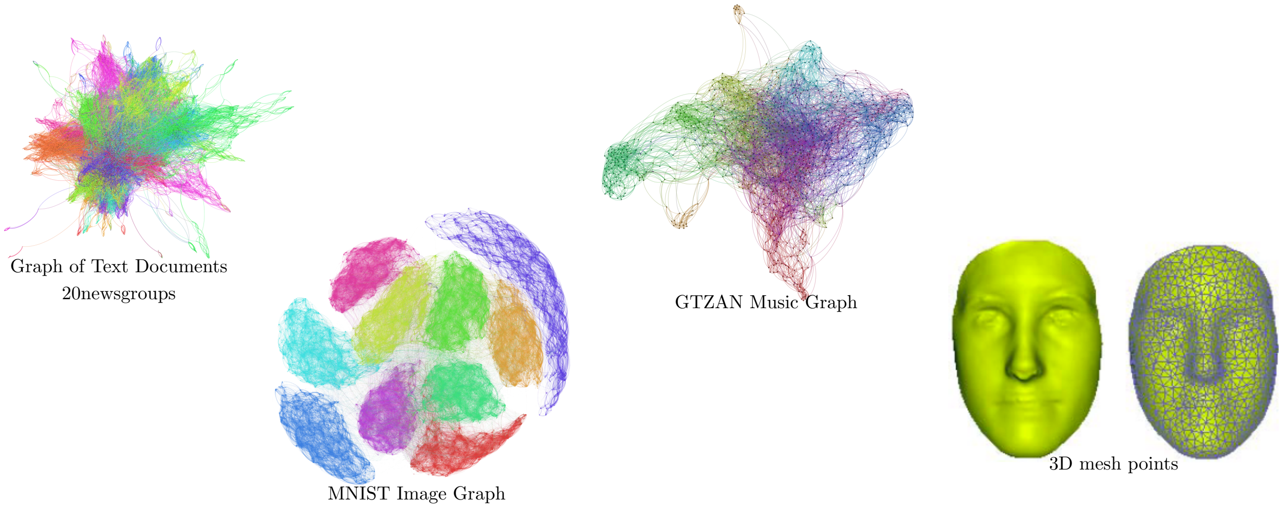
US Electrical Network



Telecommunication Network

- These different examples all form natural networks, that can be directly represented by graphs.

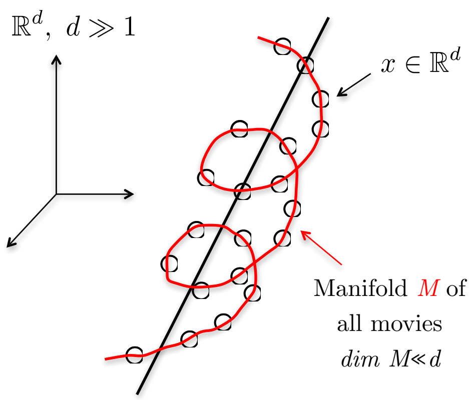
Constructed graphs



- Graphs can be constructed from data, using good common practice (for example by nearest neighbours) or domain expertise

Data structure and manifolds

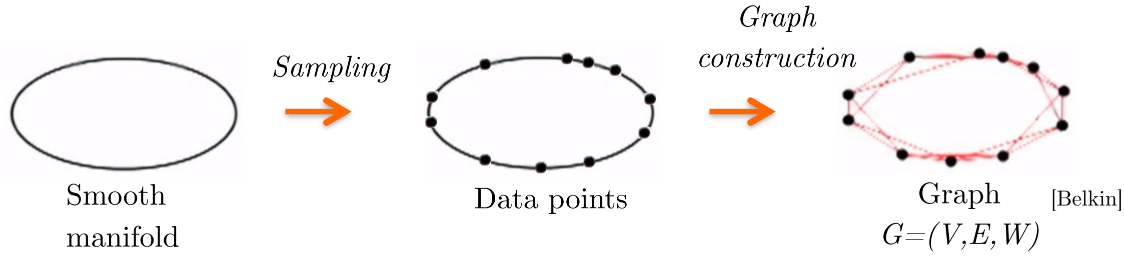
- Manifold assumption: High-dim data are sampled from a low-dim manifold.
  - Let *x* be a movie, each movie is defined by *d* features/attributes like genre, actors, release year, origin country, etc such that *x* in R^d. Then we can make the assumption that all movies form a manifold in R^d.



- Assumption validity: It is a good working hypothesis for:
  - Several types of data (images, text documents, music, etc)
  - Most data science tasks (classification, visualization, recommendation, etc)

From manifolds to graphs

- Graphs can be seen as manifold sampling: the manifold information is encoded by neighborhood graphs



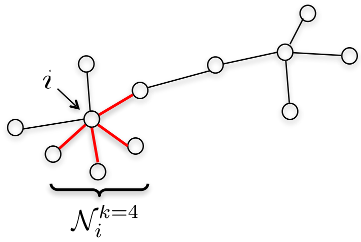
- Neighborhood Graphs: k-NN graphs are among the most popular

distance between samples

$$W_{ij} = \begin{cases} e^{-\frac{\mathrm{dist}(x_i,x_j)^2}{\sigma^2}} & \text{if } j \in \mathcal{N}_i^k \\ 0 \text{ otherwise} \end{cases}$$

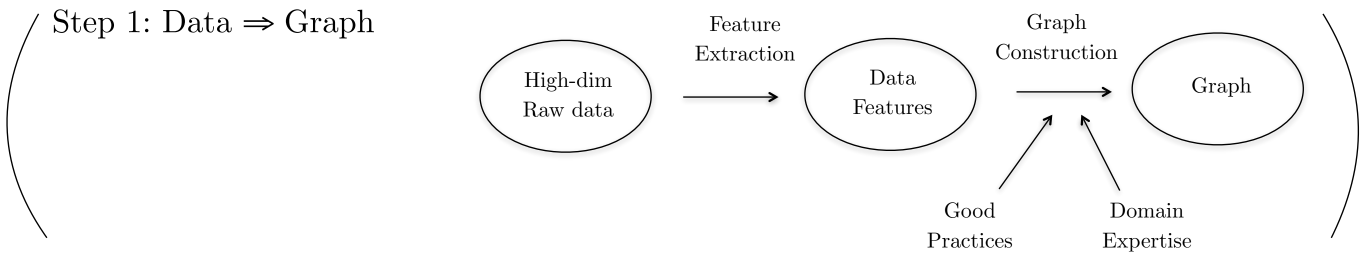
neighbourhood

scale parameter

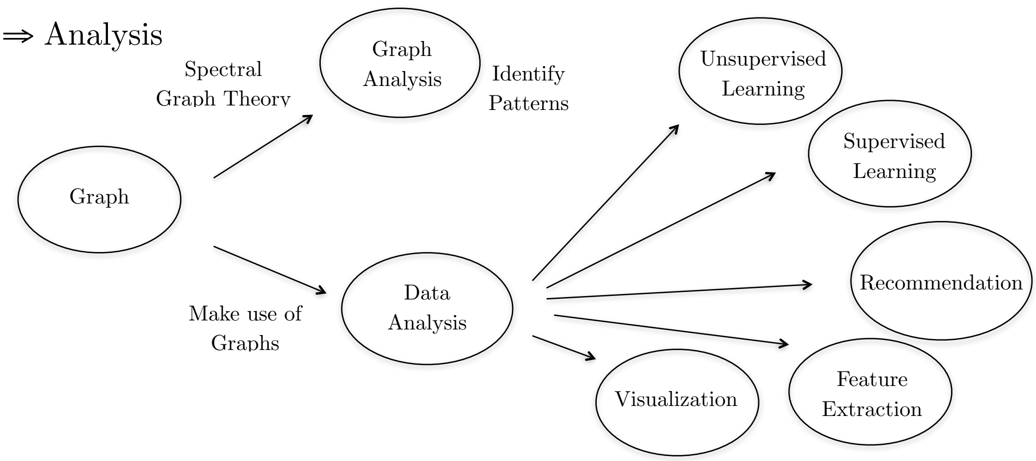


Structured data analysis pipeline

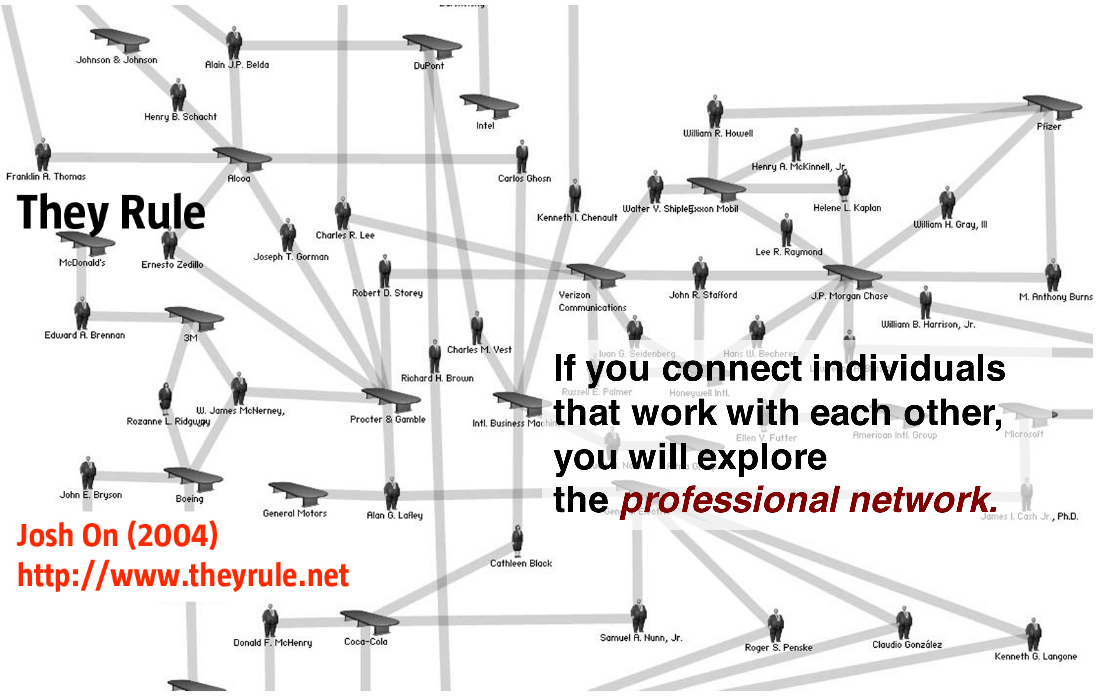
Step 1: Data ⇒ Graph



Step 2: Graph ⇒ Analysis



Importance of the representation



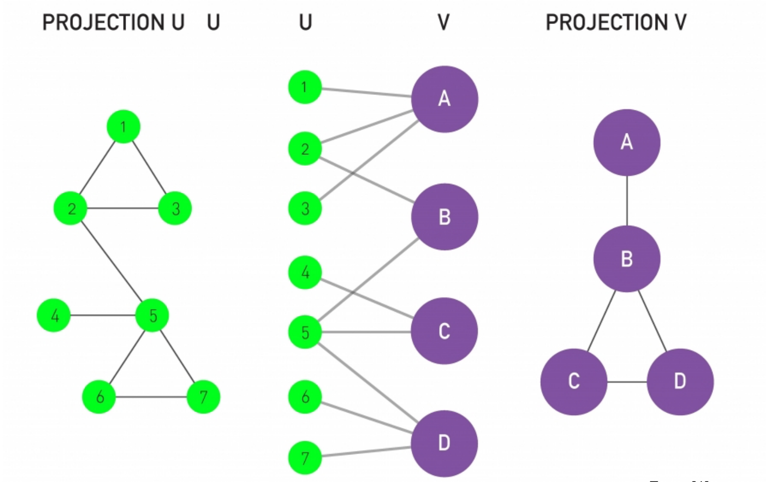
Common networks

NETWORK	NODES	LINKS	DIRECTED UNDIRECTED	Number of nodes	Number of links
Internet	Routers	Internet connections	Undirected	192,244	609,066
WWW	Webpages	Links	Directed	325,729	1,497,134
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594
Mobile Phone Calls	Subscribers	Calls	Directed	36,595	91,826
Email	Email addresses	Emails	Directed	57,194	103,731
Science Collaboration	Scientists	Co-authorship	Undirected	23,133	93,439
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908
Citation Network	Paper	Citations	Directed	449,673	4,689,479
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930

From [1]

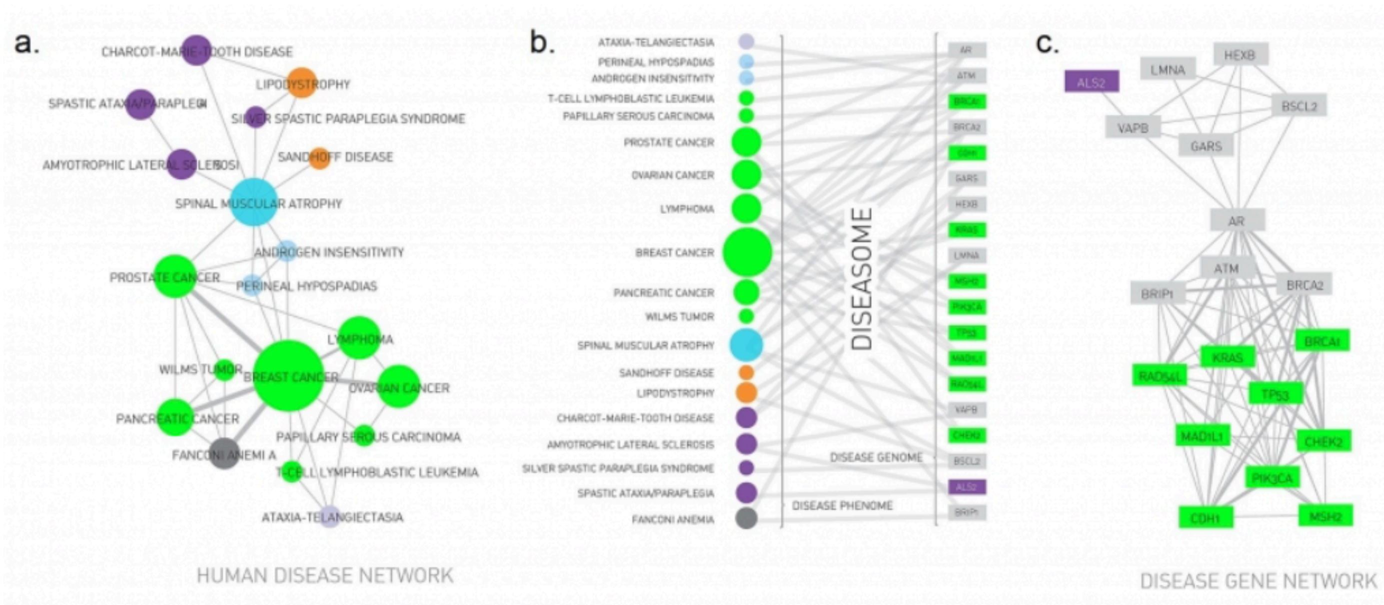
Special case: bipartite graphs

- A *bipartite graph* (or bigraph) is a graph whose nodes can be divided into two disjoint sets *U* and *V* such that every link connects a node in *U* to one in *V*; that is, *U* and *V* are independent sets.



From [1]

Bipartite network: disease



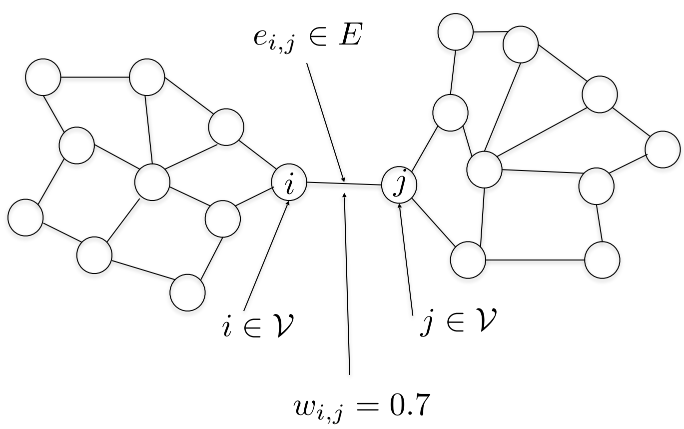
From [1]

Overview

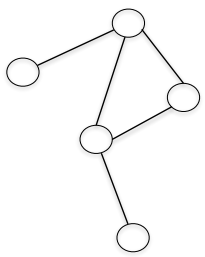
- Networks and graphs
- *Definitions*
- Network density
- Pathology
- Connectedness

Weighted graphs

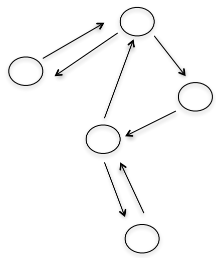
- A *weighted* graph is fully defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$
  - $\mathcal{V}$ is the set of vertices, $\mathcal{E}$ the set of edges, and W the weight or similarity matrix
  - the total number of vertices is $|\mathcal{V}| = N$ and the total number of edges is $|\mathcal{E}| = L$
  - The graph is *unweighted* if the edge weights are binary, i.e., $e_{i,j} \in \{0, 1\}, \forall i, \forall j$



Directed graphs



undirected

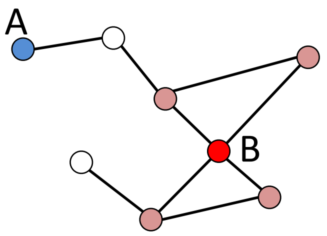


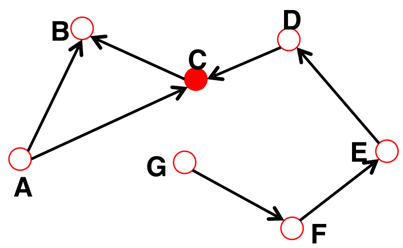
directed

- Links: undirected (symmetrical)
  - coauthorship links
  - Actor network
  - protein interactions

- Links: directed (arcs)
  - URLs on the www
  - phone calls
  - metabolic reactions
- *Directed graph = Digraph*

Node degree





- Undirected graph
  - degree k_i = number of connected edges

$$k_A = 1 \qquad k_B = 4 \qquad L = \frac{1}{2}\sum_{i=1}^{N} k_i$$

- Directed graph
  - in-degree k_i^{in} = sum of incident edges
  - out-degree k_i^{out} = sum of outgoing edges
  - (total) degree $k_i = k_i^{in} + k_i^{out}$
  - source = node with $k^{in} = 0$
  - sink = node with $k^{out} = 0$

$$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$$

For weighted graphs, the degree is the sum of the edge weights, and not the number of edges, i.e., $k_i = \sum_{j \in \mathcal{V}} w_{i,j}$

Super-fast statistics recap…

For samples of *N* values

Average (mean)

$$\langle x \rangle = \frac{x_1 + x_2 + ... + x_N}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

nth moment

$$\langle x^n \rangle = \frac{x_1^n + x_2^n + ... + x_N^n}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i^n$$

Standard deviation

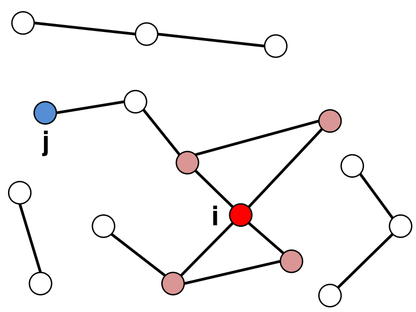
$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \langle x \rangle)^2}$$

Distribution of x

$$p_x = \frac{1}{N}\sum_i \delta_{x,x_i} \quad \text{with} \quad \sum_x p_x = 1 \left(\int p_x dx = 1 \right)$$

Average Degree

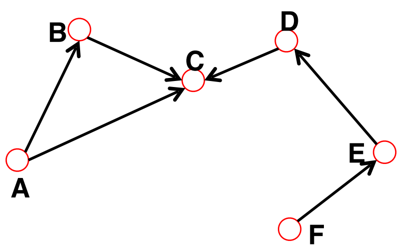
Undirected graphs



Average degree: $$\langle k \rangle \equiv \frac{1}{N}\sum_{i=1}^{N} k_i$$

$$\langle k \rangle \equiv \frac{2L}{N}$$

Directed graphs



Average degrees: $$\langle k^{in} \rangle \equiv \frac{1}{N}\sum_{i=1}^{N} k_i^{in}$$

$$\langle k^{out} \rangle \equiv \frac{1}{N}\sum_{i=1}^{N} k_i^{out}$$

$$\langle k^{in} \rangle \equiv \langle k^{out} \rangle = \frac{L}{N}$$

Average degree in reference networks

Network	Nodes	Links	Directed / Undirected	N	L	‹K›
Internet	Routers	Internet connections	Undirected	192,244	609,066	6.34
WWW	Webpages	Links	Directed	325,729	1,497,134	4.60
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594	2.67
Mobile-Phone Calls	Subscribers	Calls	Directed	36,595	91,826	2.51
Email	Email addresses	Emails	Directed	57,194	103,731	1.81
Science Collaboration	Scientists	Co-authorships	Undirected	23,133	93,437	8.08
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908	83.71
Citation Network	Papers	Citations	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802	5.58
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930	2.90

From [1]

Degree distribution

- p_k : probability that a node has degree k

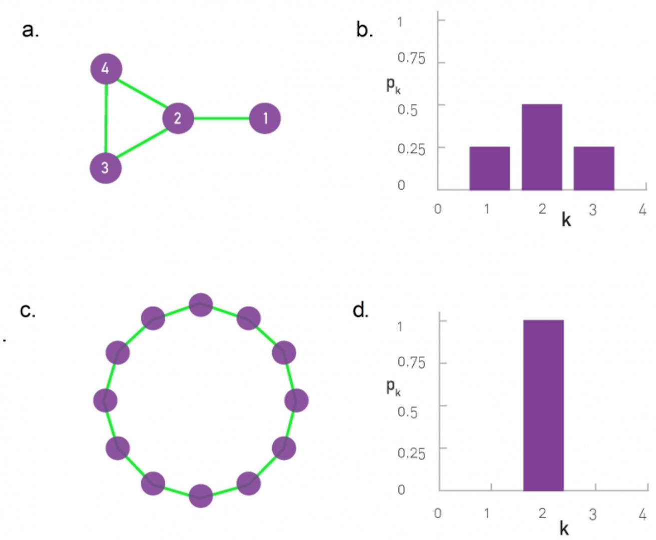
$$p_k = \frac{N_k}{N} \qquad \langle k \rangle = \sum_{k=0}^{\infty} k p_k$$

- with N_k the number of nodes of degree k

- Normalization conditions:

$$\sum_{k=0}^{\infty} p_k = 1$$

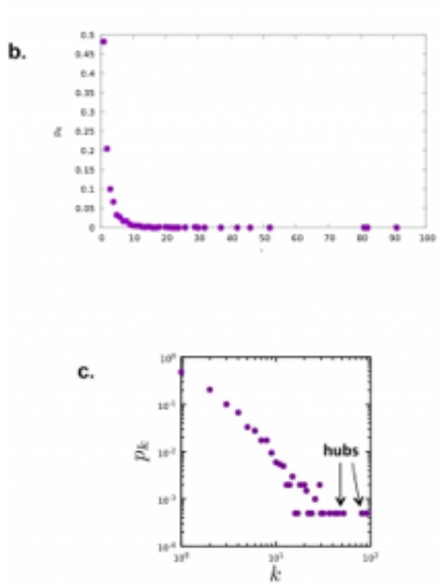
- Continuous' version : $p(k)$
  - with probability that a node has degree between k_1 and k_2 given by $\int_{k_1}^{k_2} p(k)dk$, and $\int_0^{\infty} p(k)dk = 1$



From [1]

Degree distribution in a real network

From [1]



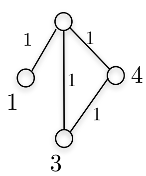
protein interaction network of yeast

Adjacency matrix

- The *adjacency* matrix A (aka *similarity*, or *weight* matrix W) captures all information about the network

- Binary matrix (unweighted graph): $w_{i,j} \in \{0,1\}$

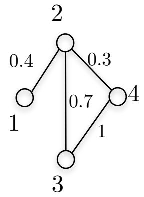
$$W_{i,j} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

$G=$ 

$$\Rightarrow W = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

- Weighted matrix: $w_{i,j} \in [0,1]$ (commonly normalised)

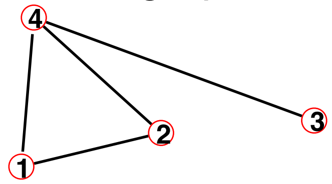
$$W_{i,j} = \begin{cases} w_{i,j} \in [0,1], & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

$G=$ 

$$\Rightarrow W = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0.4 & 0 & 0 \\ 0.4 & 0 & 0.7 & 0.3 \\ 0 & 0.7 & 0 & 1 \\ 0 & 0.3 & 1 & 0 \end{bmatrix} \end{array}$$

Adjacency matrix and degrees

Undirected graphs



$$W = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

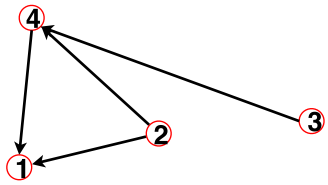
$$W_{ij} = W_{ji}$$
$$W_{ii} = 0$$

$$k_i = \sum_{j=1}^{N} W_{ij}$$

$$k_j = \sum_{i=1}^{N} W_{ij}$$

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i = \frac{1}{2}\sum_{i,j}^{N} W_{ij}$$

Directed graphs



$$W = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$W_{ij} \neq W_{ji}$$
$$W_{ii} = 0$$

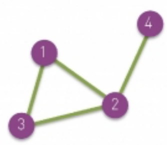
$$k_i^{out} = \sum_{j=1}^{N} W_{ij}$$

$$k_j^{in} = \sum_{i=1}^{N} W_{ij}$$

$$L = \sum_{i=1}^{N} k_i^{in} = \sum_{j=1}^{N} k_j^{out} = \sum_{i,j}^{N} W_{ij}$$

Graphology

a. Undirected



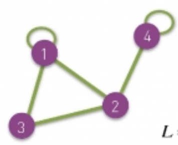
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$A_{ii} = 0$ $\quad A_{ij} = A_{ji}$

$L = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij}$ $\quad <k> = \frac{2L}{N}$

Internet, power grid, science collaboration networks

b. Self-loops



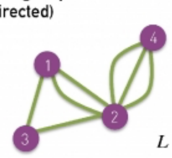
$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$\exists i, A_{ii} \neq 0$ $\quad A_{ij} = A_{ji}$

$L = \frac{1}{2}\sum_{i,j=1,i\neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii}$ $\quad ?$

Protein interaction network, www

c. Multigraph (undirected)



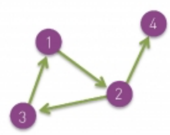
$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

$A_{ii} = 0$ $\quad A_{ij} = A_{ji}$

$L = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij}$ $\quad <k> = \frac{2L}{N}$

Social networks, collaboration networks

d. Directed



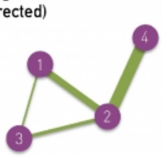
$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$A_{ij} \neq A_{ji}$

$L = \sum_{i,j=1}^{N} A_{ij}$ $\quad <k> = \frac{L}{N}$

WWW, mobile phone calls, citation network

e. Weighted (undirected)



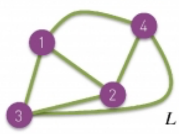
$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

$A_{ii} = 0$ $\quad A_{ij} = A_{ji}$

$<k> = \frac{2L}{N}$

Mobile phone calls, email network

f. Complete Graph (undirected)



$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$A_{ii} = 0$ $\quad A_{i\neq j} = 1$

$L = L_{max} = \frac{N(N-1)}{2}$ $\quad <k> = N-1$

From [1]

Actors in the cast of the same movie

Overview

- Networks and graphs
- Definitions
- *Network density*
- Pathology
- Connectedness

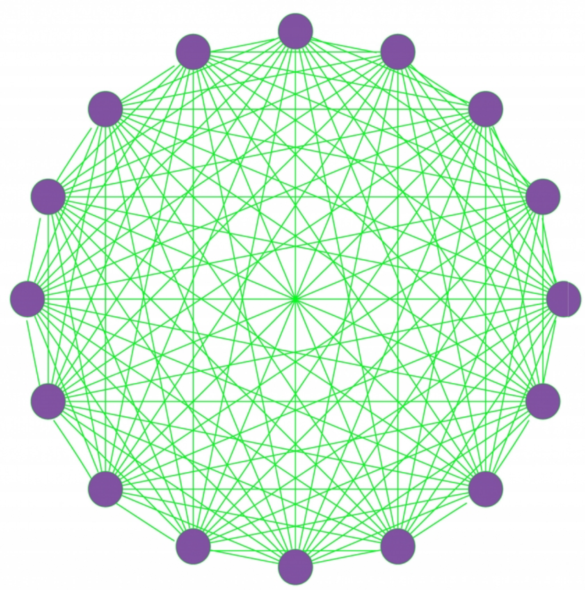
Network density

- Maximum number of links in a network:

$$L_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$

  - a graph with $L = L_{max}$ is a *complete graph,* and $L = \mathcal{O}(N^2)$
  - a complete graph has an average degree of

$$\langle k \rangle = N - 1$$



From [1]

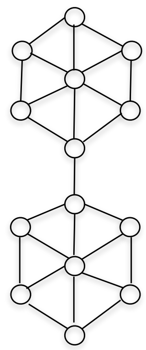
- Fortunately, most real networks are sparse:

$\langle k \rangle \ll N-1$ and $L \ll L_{max}$, typically $L = \mathcal{O}(N)$

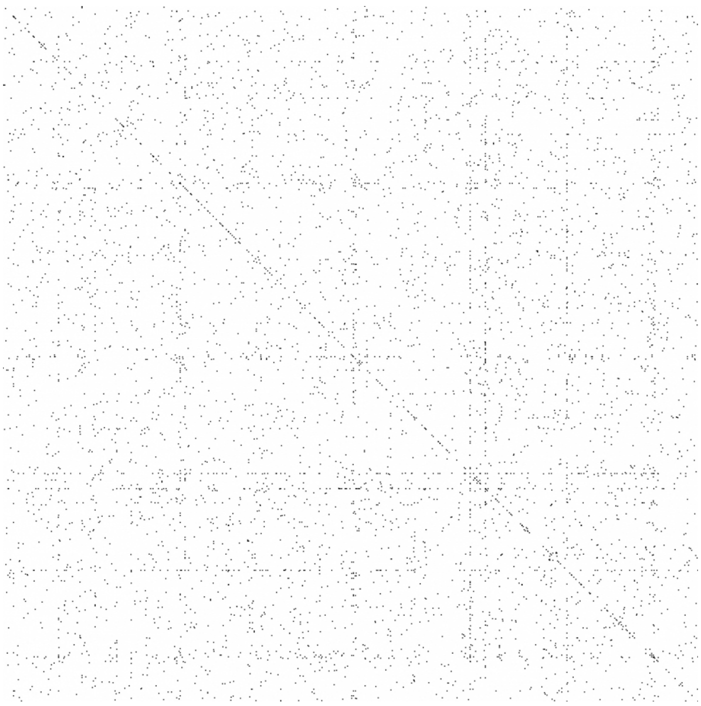
Real networks are sparse

- Sparse networks are highly desirable for memory and computational efficiency
- Good news: most natural/real-world networks (Facebook, Brain, Communication) are sparse.
- Sparsity may relate to non-trivial structure

	N	L	L_{max}	$\langle k \rangle$
WWW (ND Sample)	325729	$1.4\ 10^{6}$	10^{12}	4.51
Protein (*S. Cerevisiae*)	1870	4470	10^{7}	2.39
Coauthorship (Math)	70975	$2\ 10^{5}$	$3\ 10^{10}$	3.9
Movie Actors	212250	$6\ 10^{6}$	$1.8\ 10^{13}$	28.78
Internet (2016)	$4.73\ 10^{9}$	10^{11}	10^{18}	



Adjacency matrix of real network



From [1]

protein-protein interaction network of yeast

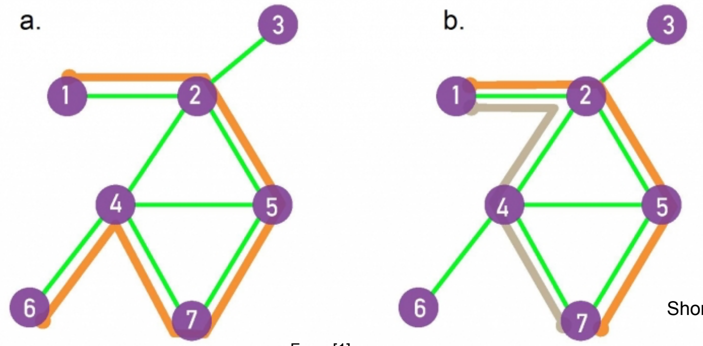
Overview

- Networks and graphs
- Definitions
- Network density
- *Pathology*
- Connectedness

Paths

- A path is a sequence of nodes in which each node is adjacent to the next one
  - In a directed network, the path can only follow the direction of an arrow.
- P_{i_0,i_n} of length n between nodes i_0 and i_n is an ordered collection of n links and $n+1$ nodes

$P_n = \{i_0, i_1, i_2, \ldots, i_n\}$ $\quad$ $P_n = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n)\}$



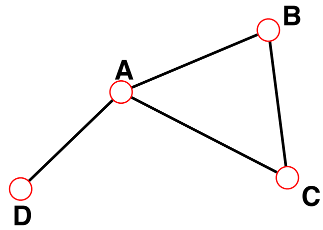
A path between 1 and 6

Shortest paths between 1 and 7
$d_{17} = 3$

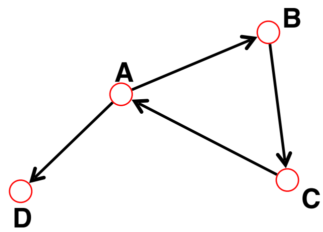
From [1]

Distance in a graph

- The *distance* d_{ij} (shortest path, geodesic path) between two nodes *i* and *j* is defined as the number of edges along the shortest path connecting them.
  - If the two nodes are disconnected, the distance is infinity.

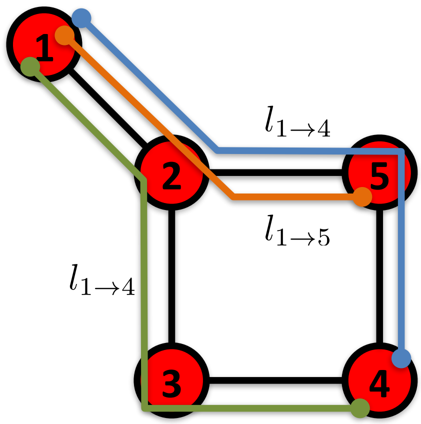


- In directed graphs each path needs to follow the direction of the arrows.
  - Thus in a digraph the distance from node *i* to *j* is generally different from the distance from node *j* to *i*.



- On weighted graphs, a weight distance can be defined in addition to the hop-distance

Shortest path



$l_{1\to4} = 3$

$l_{1\to5} = 2$

The path with the shortest length between two nodes (distance).

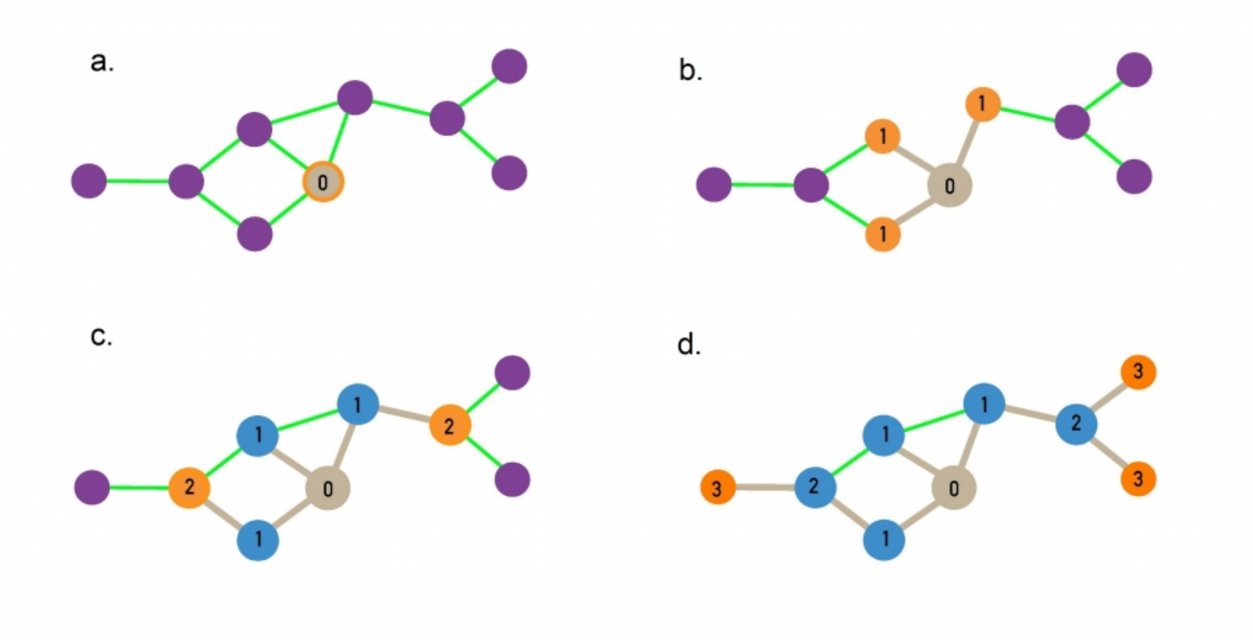
Finding shortest path: BFS

- Breadth-First Search (BFS): frequently used algorithm in network science for finding shortest path between *i* and *j*

Algorithm 1 BFS Algorithm: Shortest path between i and j

1: Start at node i, label it with "0".
2: Find the nodes directly linked to i. Label them distance "1" and put them in a queue
3: Take the first node, labeled n, out of the queue ($n = 1$ in the first step). Find the unlabeled nodes adjacent to it in the graph. Label them with $n+1$ and put them in the queue.
4: Repeat step 3 until you find the target node j or there are no more nodes in the queue.
5: The distance between i and j is the label of j. If j does not have a label, then $d_{i,j} = \infty$

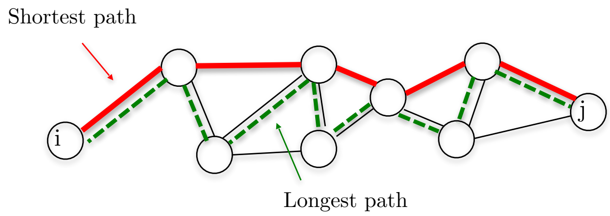
BFS algorithm illustration

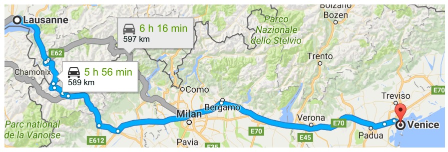


From [1]

Shortest path: a classic!

- Road navigation, e.g., Lausanne to Venice





- On weighted graphs, the shortest distance can be defined based on the sum of edge weights along the path, rather than the hop distance
  - Fast Algorithm: Dijkstra's algorithm (1956)

Number of paths between nodes

- We define with N_{ij} the number of paths between two nodes *i* and *j*

- Path of length $n=1$: If there is a link between *i* and *j*, then $A_{ij}=1$. Otherwise $A_{ij}=0$.
- Path of length $n=2$: If there is a path of length two between *i* and *j*, then $A_{ik}A_{kj}=1$. Otherwise $A_{ik}A_{kj}=0$.
  - Number of paths of length two:

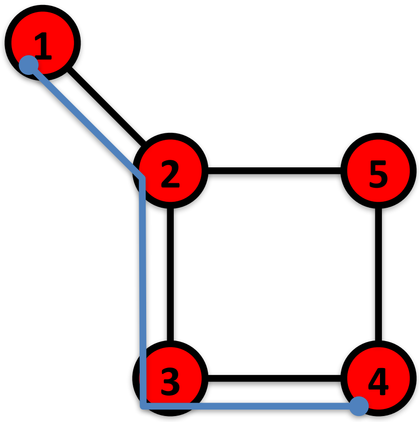
$$N_{ij}^{(2)} = \sum_{k=1}^{N} A_{ik}A_{kj} = [A^2]_{ij}$$

- Path of length *n*: in general, if there is a path of length *n* between *i* and *j*, then $A_{ik}\dots A_{lj}=1$. Otherwise $A_{ik}\dots A_{lj}=0$.
  - Number of paths of length *n*:

$$N_{ij}^{(n)} = [A^n]_{ij}$$

Diameter

- The network diameter d_{max} is the maximum distance between any pair of nodes in the graph (i.e., the longest shortest path).



$l_{1 \to 4} = 3$

Average distance

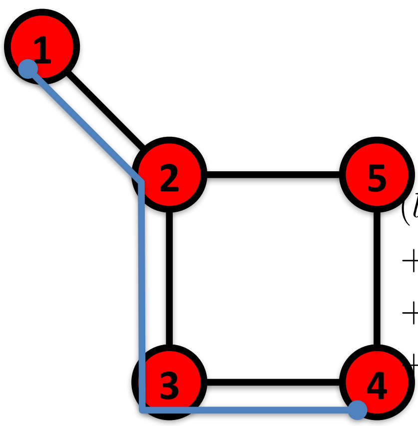
- The average path length (or average distance) for a connected graph is

$$\langle d \rangle \equiv \frac{1}{N(N-1)} \sum_{i,j \neq i} d_{ij} \text{ , or } \langle d \rangle \equiv \frac{2}{N(N-1)} \sum_{i,j>i} d_{ij} \text{ if } d_{ij} = d_{ji}$$

(undirected graphs)

- with d_{ij} the distance between node *i* and *j*

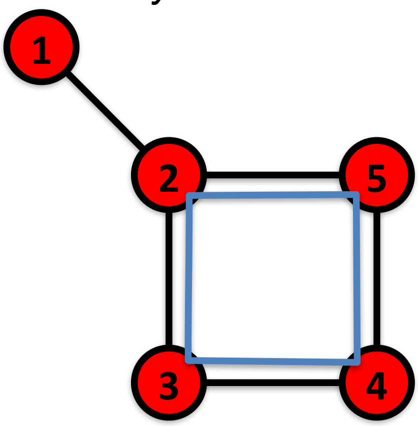
The average distance is the average of the shortest paths for all pairs of nodes.



$$(l_{1\to2} + l_{1\to3} + l_{1\to4} + l_{1\to5} + l_{2\to3} + l_{2\to4} + l_{2\to5} + l_{3\to4} + l_{3\to5} + l_{4\to5})/10 = 1.6$$

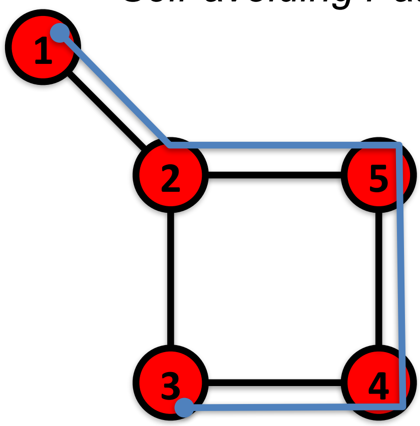
Other paths

Cycle



A path with the same start and end node.

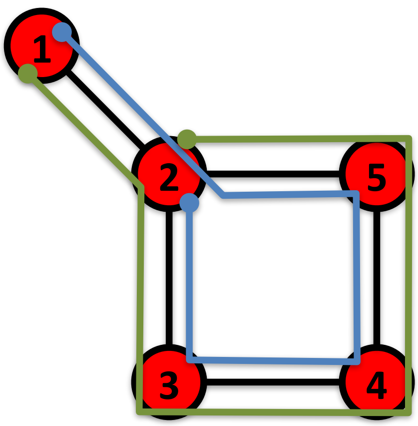
Self-avoiding Path



A path that does not intersect itself.

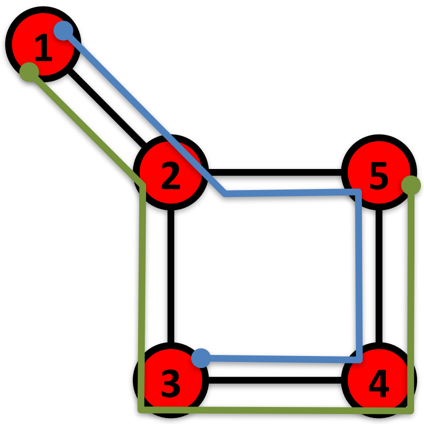
Yet more paths

Eulerian Path



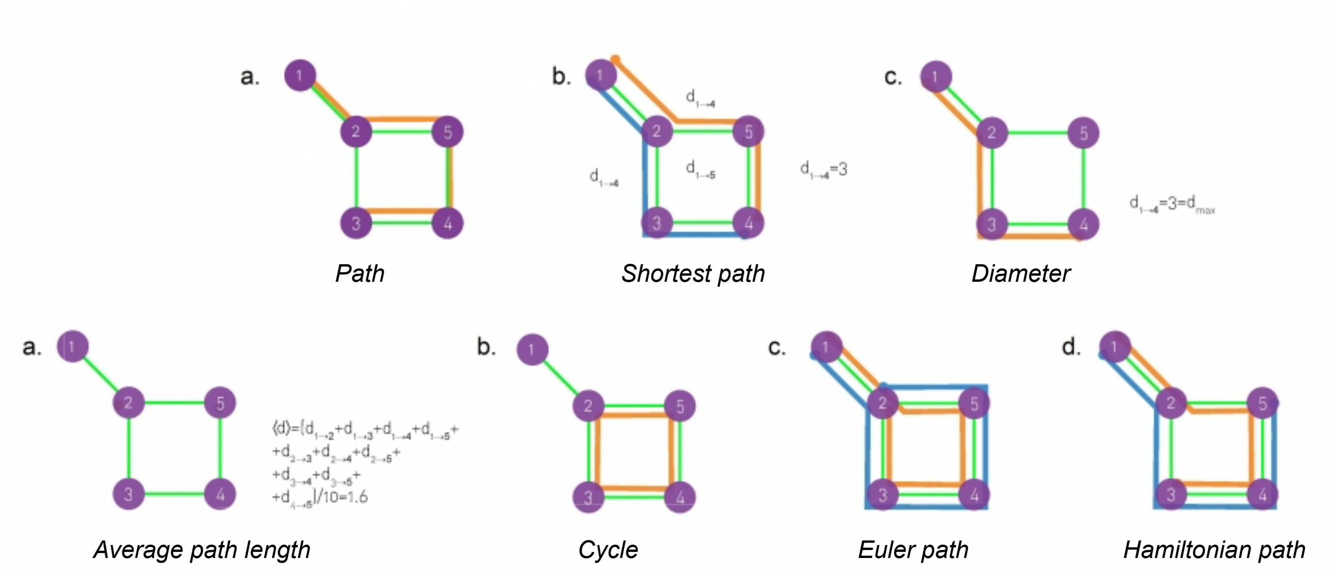
A path that traverses each link exactly once.

Hamiltonian Path



A path that visits each node exactly once.

Pathology: summary



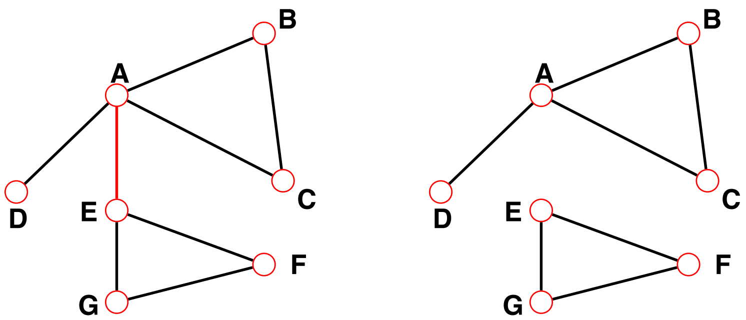
From [1]

Overview

- Networks and graphs
- Definitions
- Network density
- Pathology
- *Connectedness*

Connected graphs

- A connected graph is a graph where any two vertices can be joined by a path.

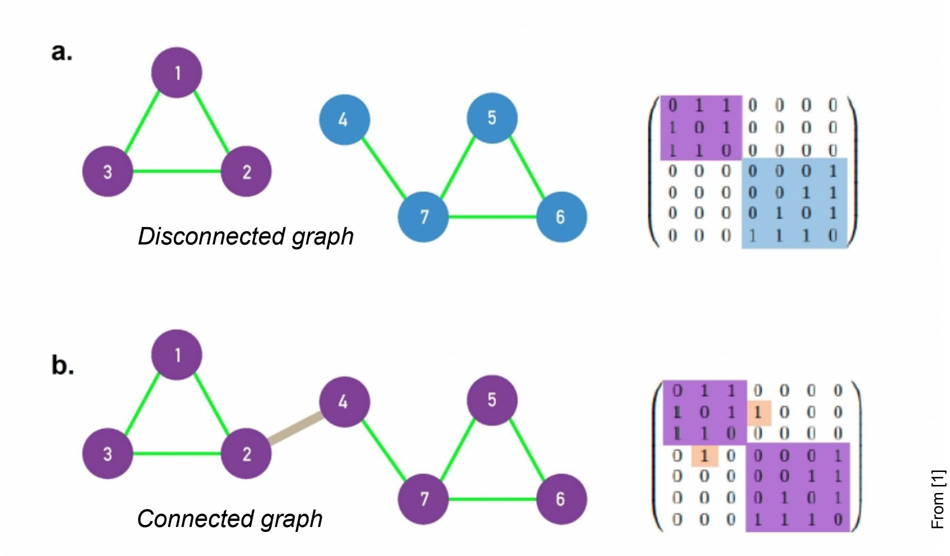


Largest Component: ***Giant Component***

The rest: ***Isolates***

- A disconnected graph is made up by two or more connected components.
  - A bridge is a link that, if erased, leads to a disconnected graph.

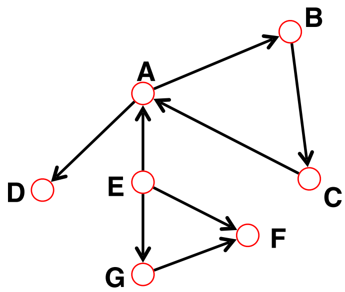
Connectivity of undirected graphs

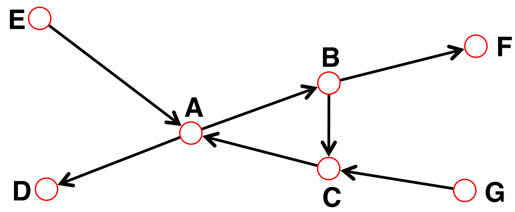


The adjacency matrix of a network with several components can be written in a block-diagonal form, so that nonzero elements are confined to squares, with all other elements being zero.

Connectivity of directed graphs

- A *strongly connected* directed graph has a path from each node to every other node and vice versa (e.g., AB path and BA path)
- A *weakly connected* directed graph is a graph that is connected if we disregard the edge directions.





- *Strongly connected components* can be identified, but not every node is part of a nontrivial strongly connected component.
  - *In-component*: nodes that can reach the scc
  - *Out-component*: nodes that can be reached from the scc

Components identification

Algorithm 1 Finding connected components

1: Start from a randomly chosen node i and perform a BFS. Label all nodes reached this way with $n = 1$.
2: If the total number of labeled nodes equals N, then the network is connected. If the number of labeled nodes is smaller than N, the network consists of several components. To identify them, proceed to step 3.
3: Increase the label $n \to n + 1$. Choose an unmarked node j, label it with n. Use BFS to find all nodes reachable from j, label them all with n. Return to step 2.

Clustering coefficient

- The clustering coefficient is an indication about the fraction of the node's neighbors that are connected:

$$C_i = \frac{2L_i}{k_i\,(k_i - 1)} \qquad \text{for} \qquad k_i \notin [0,1]$$

$C_i = 0$ otherwise

where L_i represents the number of links between the k_i neighbors of node *i*

- It represents the probability that two neighbors of a node, are linked to each other,

$$C_i \in [0,1]$$

- The average clustering coefficient is given by $\langle C \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i$

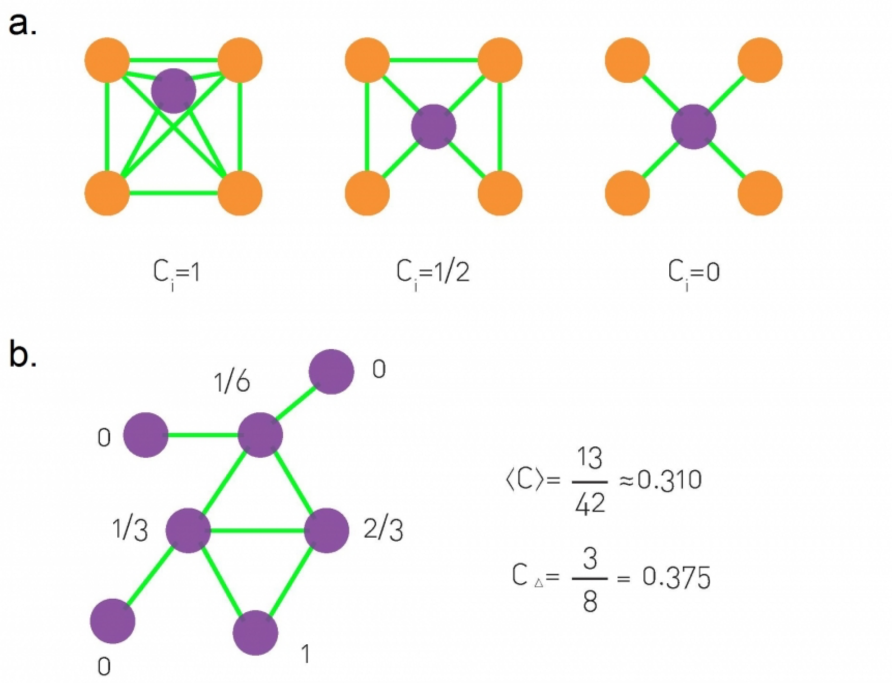
A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. PNAS, 2004.
J. P. Onnela, J. Saramäki, J. Kertész, and K. Kaski. Intensity and coherence of motifs in weighted complex networks. Physical Review E, 2005.
B. Zhang and S. Horvath. A general framework for weighted gene coexpression network analysis. Statistical Applications in Genetics and Molecular Biology, 2005.
P. Holme, S. M. Park, J. B. Kim, and C. R. Edling. Korean university life in a network perspective: Dynamics of a large affiliation network. Physica A, 2007.

Global clustering coefficient

$$C_\Delta = \frac{3 \times \text{NumberOfTriangles}}{\text{NumberOfConnectedTriplets}}$$

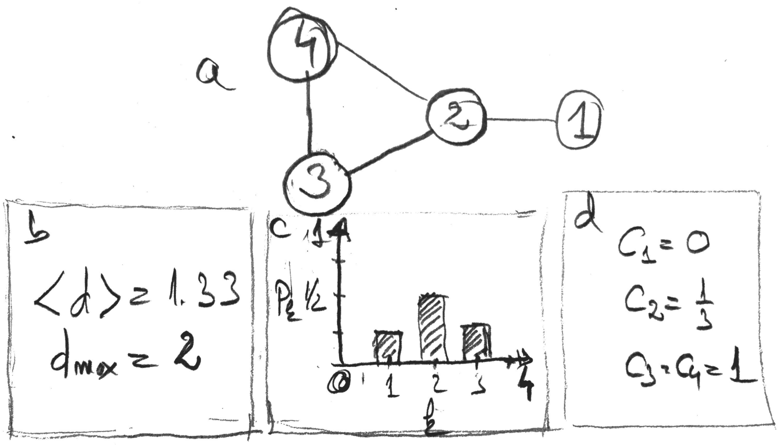
- It measures the total number of closed triangles in a network
  - each link between two neighbours of node *i* closes a triangle
- It is normalised by the number of triplets
  - a triplet is a series of 3 connected nodes (not necessarily forming a triangle)
  - a triangle contains 3 triplets
- It is not identical to the average clustering coefficient
  - they generally have comparable values

Clustering coefficient: examples



From [1]

Summary: three central quantities



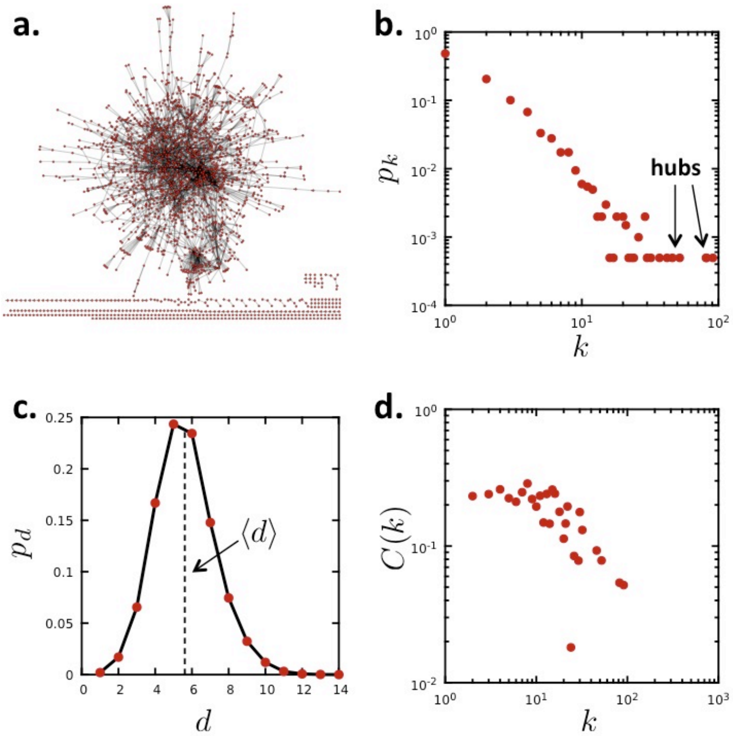
- Degree distribution: p_k
- Path length: $\langle d \rangle$
- Clustering coefficient: $C_i = \dfrac{2L_i}{k_i\,(k_i - 1)}$

A case study: protein-to-protein interaction network

Undirected network
N=2,018 proteins as nodes
L=2,930 binding interactions as links.
Average degree <k>=2.90.

Not connected: 185 components
the largest (giant component) 1,647 nodes

Analysing central quantities



$$p_k = \frac{N_k}{N}$$

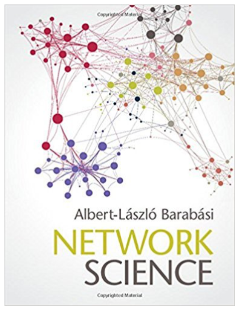
$d_{max} = 14$

$\langle d \rangle = 5.61$

$\langle C \rangle = 0.12$

References

[1] Network Science, by Albert-László Barabási



Some content of the slides is inspired from Prof. Xavier Bresson's class 'A Network Tour of Data Science (2016), and from Prof. Barabási's class on Network Science (www.BarabasiLab.com)