COMPLEX NETWORKS

WHO AM I

- Rémy Cazabet
- Associate Professor (Maître de conférences)
  - Université Lyon 1
  - LIRIS, DM2L Team (Data Mining & Machine Learning)
- Computer Scientist => Network Scientist
- Member of IXXI

RESOURCES

- Website of the course:
  - http://cazabetremy.fr/Teaching/CN/ComplexNetworks.html
  - Slides, Cheat sheets, notebooks, etc.

- Contact me: remy.cazabet@univ-lyon1.fr

CLASS OVERVIEW

Day	Time	Room	Teacher	Topic	Resources
Thursday Sep. 14	13h30-15h30	F	Rémy Cazabet	Lecture: Introduction, Describing Networks, Centralities	CheatSheet_intro - CheatSheet_matrices - CheatSheet_centralities - Slides
Thursday Sep.21	10h15-12h15	F	Rémy Cazabet	Experiments: Gephi, networkx intro	
Thursday Sep.28	10h15-12h15	F	Rémy Cazabet	Lecture: Random Graphs	
Thursday Oct. 5	10h15-12h15	F	Rémy Cazabet	Experiments: Random Graphs	
Thursday Oct. 12	10h15-12h15	F	**Pierre Borgnat**	Spreading processes ; Dynamic on networks	
Thursday Oct. 19	10h15-12h15	F	**Pierre Borgnat**	Graph Signal processing	
Thursday Oct. 26	10h15-12h15	F	**Pierre Borgnat**	Representation Learning for graphs ; embeddings	
Thursday Nov. 9	15h45-17h45	F	**Pierre Borgnat**	Graph Neural Networks	
Thursday Nov. 16	15h45-17h45	F	Rémy Cazabet	Lecture: Communities+ ML classic	
Thursday Nov. 23	15h45-17h45	F	Rémy Cazabet	Experiments: Communities	
Thursday Nov. 30	15h45-17h45	F	Rémy Cazabet	Lecture: Dynamic of Networks + dynamic communities	
Thursday Dec. 7	15h45-17h45	F	Rémy Cazabet	Experiments: Dynamic of networks	
Thursday Dec. 14	15h45-17h45	F	?		

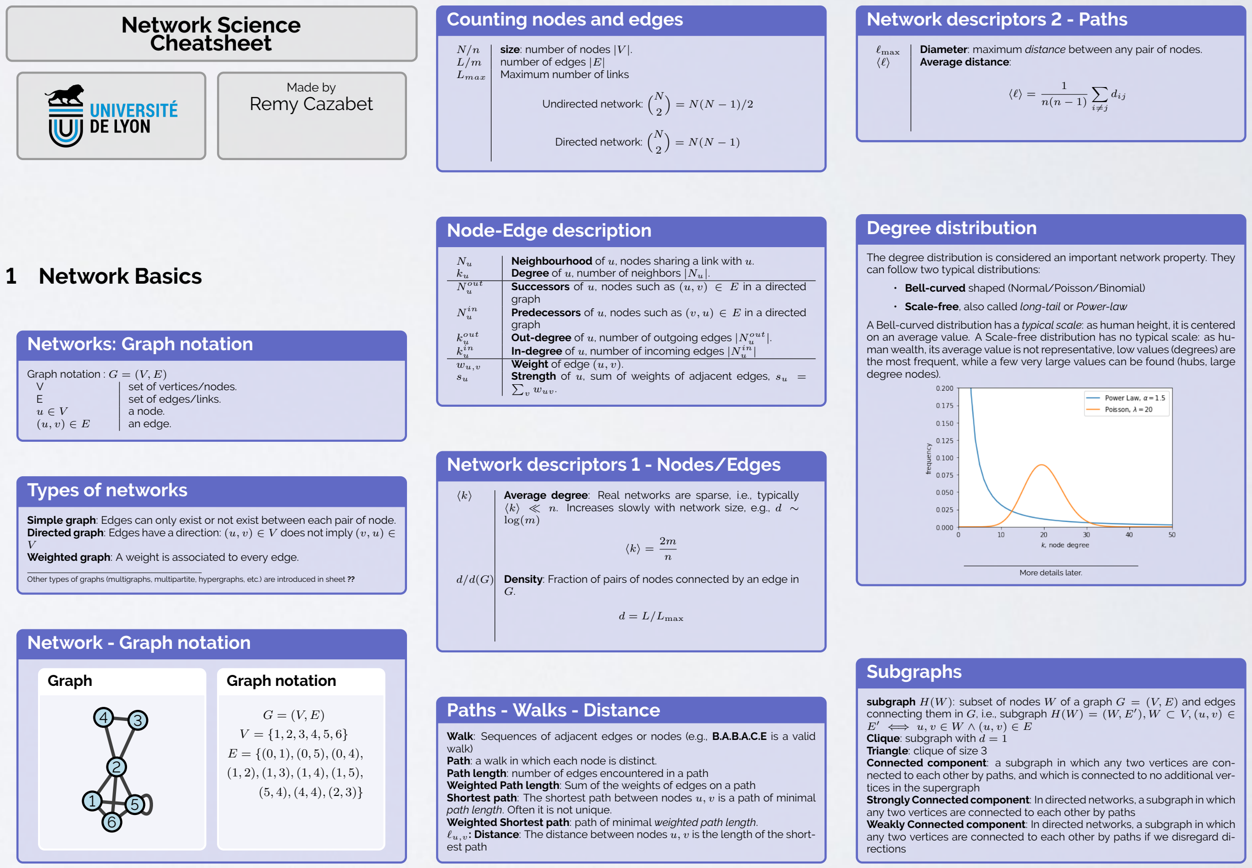
EVALUATION

- 50% Project (Mostly my part)
- 50% Final Exam (all parts)
- Project
  - In groups of 2 or 3.
  - Apply class content to analyze a network of your choice
  - More details later

LECTURES

- No need to write down definitions, etc.
  - Slides, Cheatsheet

- Questions welcomed



Network Science Cheatsheet

UNIVERSITÉ DE LYON

Made by Remy Cazabet

Counting nodes and edges

N/n	**size**: number of nodes $\|V\|$.
L/m	number of edges $\|E\|$
L_{max}	Maximum number of links

Undirected network: $\binom{N}{2} = N(N-1)/2$

Directed network: $\binom{N}{2} = N(N-1)$

Network descriptors 2 - Paths

ℓ_{max}	**Diameter**: maximum *distance* between any pair of nodes.
$\langle \ell \rangle$	**Average distance**:

$$\langle \ell \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} d_{ij}$$

Node-Edge description

N_u	**Neighbourhood** of u, nodes sharing a link with u.
k_u	**Degree** of u, number of neighbors $\|N_u\|$.
N_u^{out}	**Successors** of u, nodes such as $(u,v) \in E$ in a directed graph
N_u^{in}	**Predecessors** of u, nodes such as $(v,u) \in E$ in a directed graph
k_u^{out}	**Out-degree** of u, number of outgoing edges $\|N_u^{out}\|$.
k_u^{in}	**In-degree** of u, number of incoming edges $\|N_u^{in}\|$
$w_{u,v}$	**Weight** of edge (u,v).
s_u	**Strength** of u, sum of weights of adjacent edges, $s_u = \sum_v w_{uv}$.

Degree distribution

The degree distribution is considered an important network property. They can follow two typical distributions:

- **Bell-curved** shaped (Normal/Poisson/Binomial)
- **Scale-free**, also called *long-tail* or *Power-law*

A Bell-curved distribution has a *typical scale*: as human height, it is centered on an average value. A Scale-free distribution has no typical scale: as human wealth, its average value is not representative, low values (degrees) are the most frequent, while a few very large values can be found (hubs, large degree nodes).

More details later

1 Network Basics

Networks: Graph notation

Graph notation : $G = (V, E)$

V	set of vertices/nodes.
E	set of edges/links.
$u \in V$	a node.
$(u,v) \in E$	an edge.

Network descriptors 1 - Nodes/Edges

$\langle k \rangle$	**Average degree**: Real networks are sparse, i.e., typically $\langle k \rangle \ll n$. Increases slowly with network size, e.g., $d \sim \log(m)$

$$\langle k \rangle = \frac{2m}{n}$$

$d/d(G)$	**Density**: Fraction of pairs of nodes connected by an edge in G.

$$d = L/L_{max}$$

Types of networks

Simple graph: Edges can only exist or not exist between each pair of node.
Directed graph: Edges have a direction: $(u,v) \in V$ does not imply $(v,u) \in V$
Weighted graph: A weight is associated to every edge.

Other types of graphs (multigraphs, multipartite, hypergraphs, etc.) are introduced in sheet ??

Network - Graph notation

Graph

Graph notation

$G = (V, E)$
$V = \{1, 2, 3, 4, 5, 6\}$
$E = \{(0,1), (0,5), (0,4), (1,2), (1,3), (1,4), (1,5), (5,4), (4,4), (2,3)\}$

Subgraphs

subgraph $H(W)$: subset of nodes W of a graph $G = (V, E)$ and edges connecting them in G, i.e., subgraph $H(W) = (W, E'), W \subset V, (u,v) \in E' \iff u, v \in W \wedge (u,v) \in E$
Clique: subgraph with $d = 1$
Triangle: clique of size 3
Connected component: a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph
Strongly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths
Weakly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths if we disregard directions

Paths - Walks - Distance

Walk: Sequences of adjacent edges or nodes (e.g., B,A,B,A,C,E is a valid walk)
Path: a walk in which each node is distinct.
Path length: number of edges encountered in a path
Weighted Path length: Sum of the weights of edges on a path
Shortest path: The shortest path between nodes u, v is a path of minimal *path length*. Often it is not unique.
Weighted Shortest path: path of minimal *weighted path length*.
$\ell_{u,v}$: **Distance**: The distance between nodes u, v is the length of the shortest path

COMPLEX NETWORKS

(NETWORK SCIENCE)

WHAT?
WHY?
WHY NOW?
WHAT FOR?

SCIENCE

- Science: understanding how things work
  - The human body, the motion/characteristics of objects, societies, etc.
- Step 1: understand properties of things and rules applying to them
  - Fall of objects, classifications of species, etc.
  - Macro-scale properties: temperature, pression

SCIENCE

- 2)Great success of the 19/20 centuries: **Reductionism**
- To understand things, I need to understand what they are made of:
  - A human body: organs, vessels => cells => DNA, proteins & stuff => Nucleotides ….
  - Objects: Organic compounds => atoms => protons/electrons/neutrons => stuff
- => Now we know. And then what?

SCIENCE

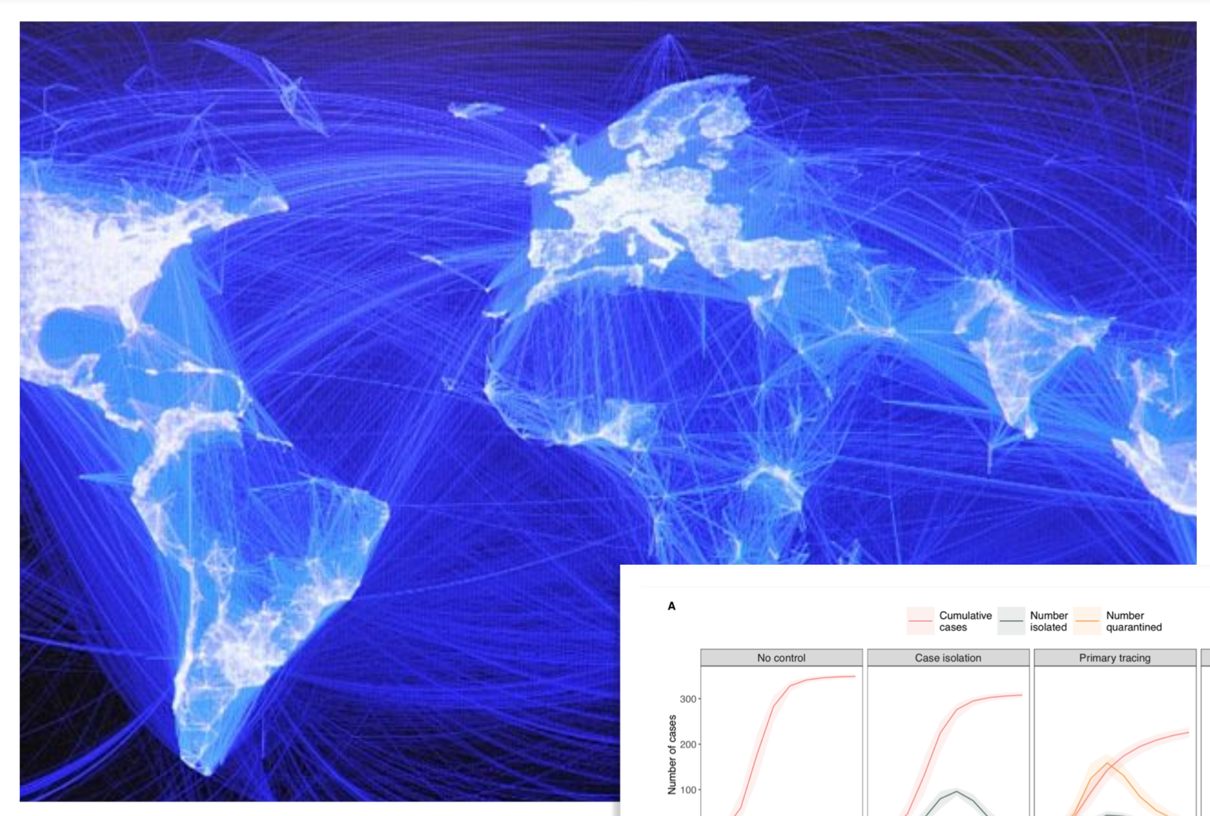
- 3) Two situations:
  - The system is **homogeneous** and/or has a **regular** structure
    - => You can explain it with equations (statistical physics…)
  - The system is **heterogeneous** and/or **has a complex structure**
    - => Understanding each component is not enough to understand the system
    - Understanding each neuron tells you little about how the brain works.
    - Understanding how each individual works/behaves tells you little about societies
    - etc.

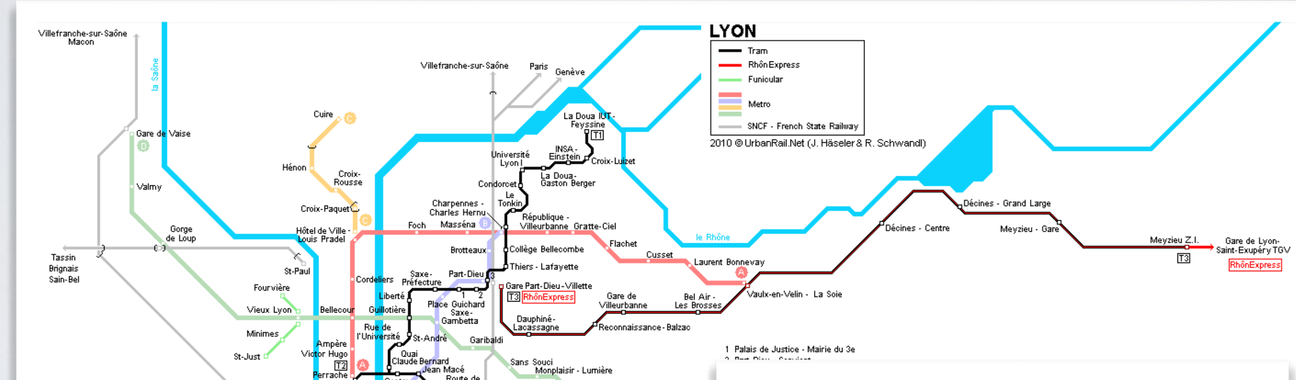
- => The structure/relations/interactions/organization matters.
  - Networks allow representing complex heterogeneous organization

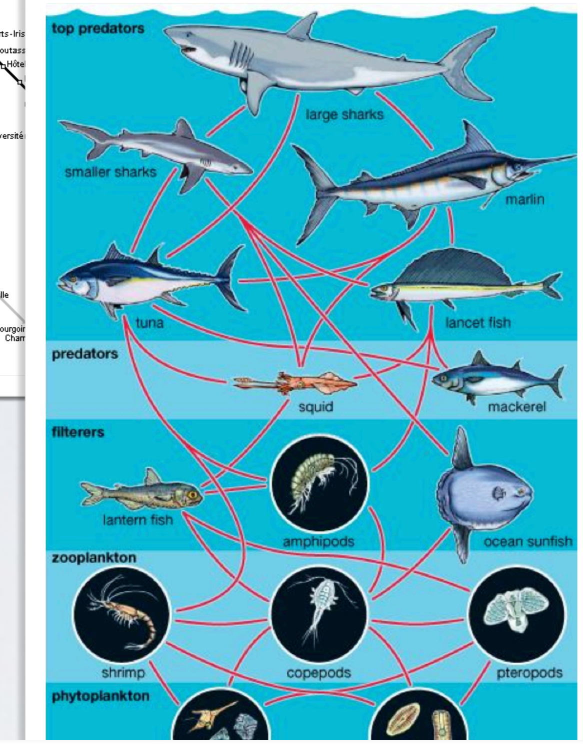
COMPLEX SYSTEMS

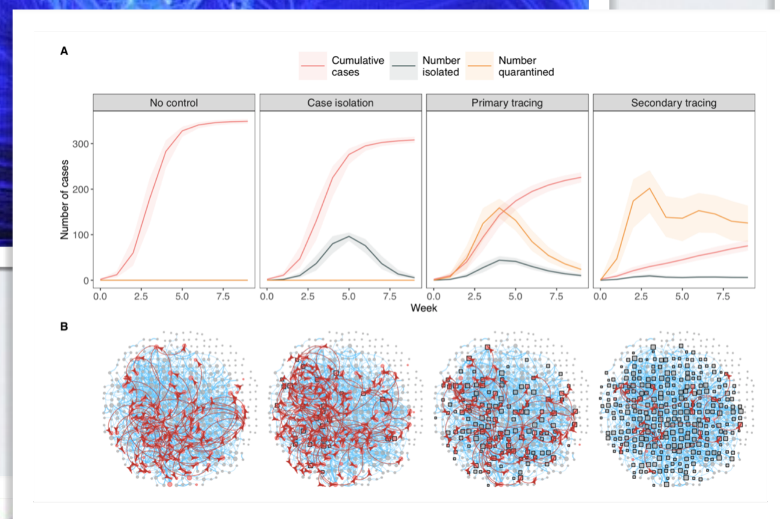
- **Complex systems**: Systems composed of multiple **parts** in **interactions**

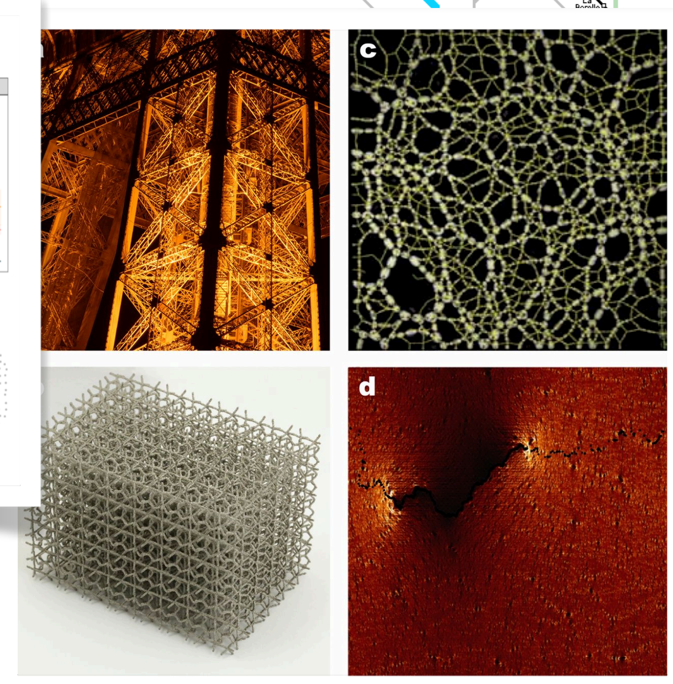
- Complex networks model the interactions between the parts
  - A common framework applicable to many systems
  - =>Many networks share similar characteristics
  - =>Similar processes shape the networks

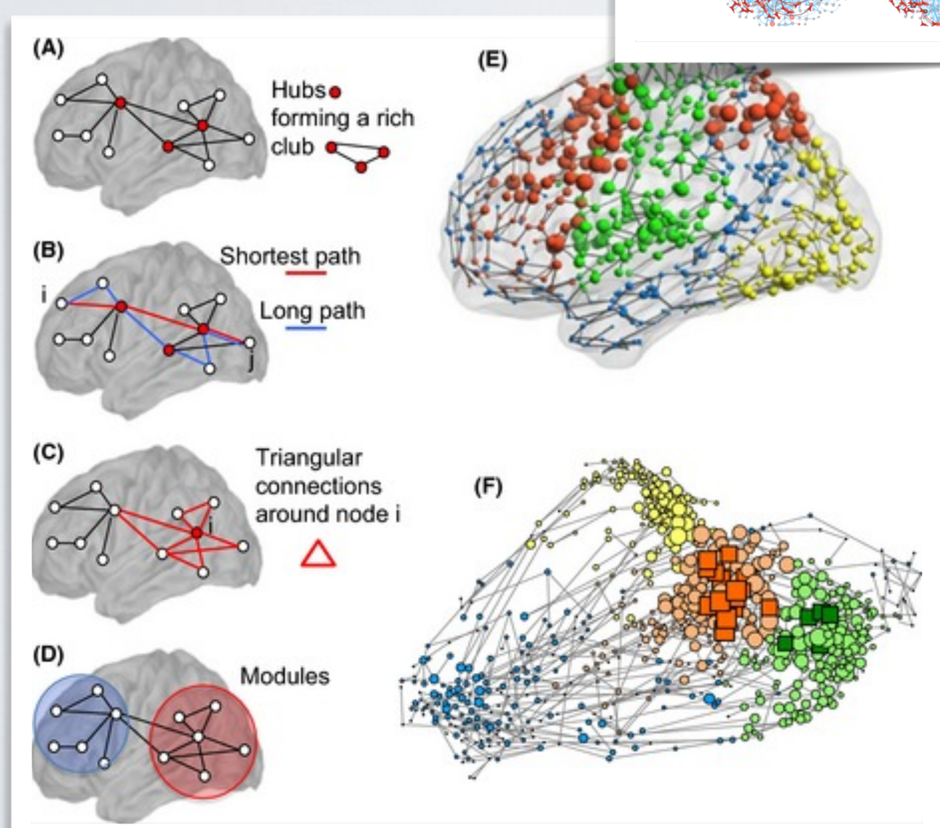


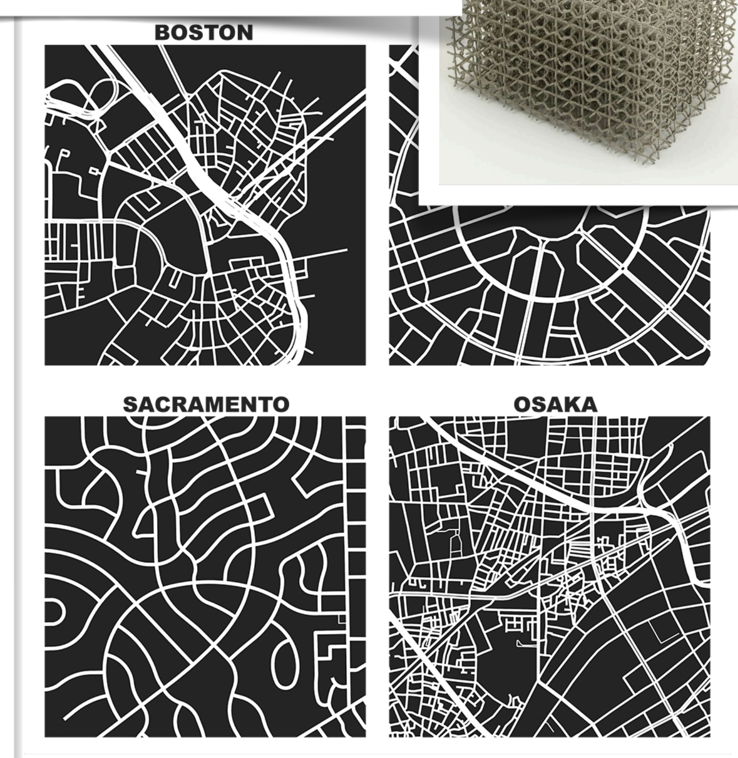

LYON
Tram
RhônExpress
Funicular
Metro
SNCF - French State Railway
2010 © UrbanRail.Net (J. Häseler & R. Schwandl)

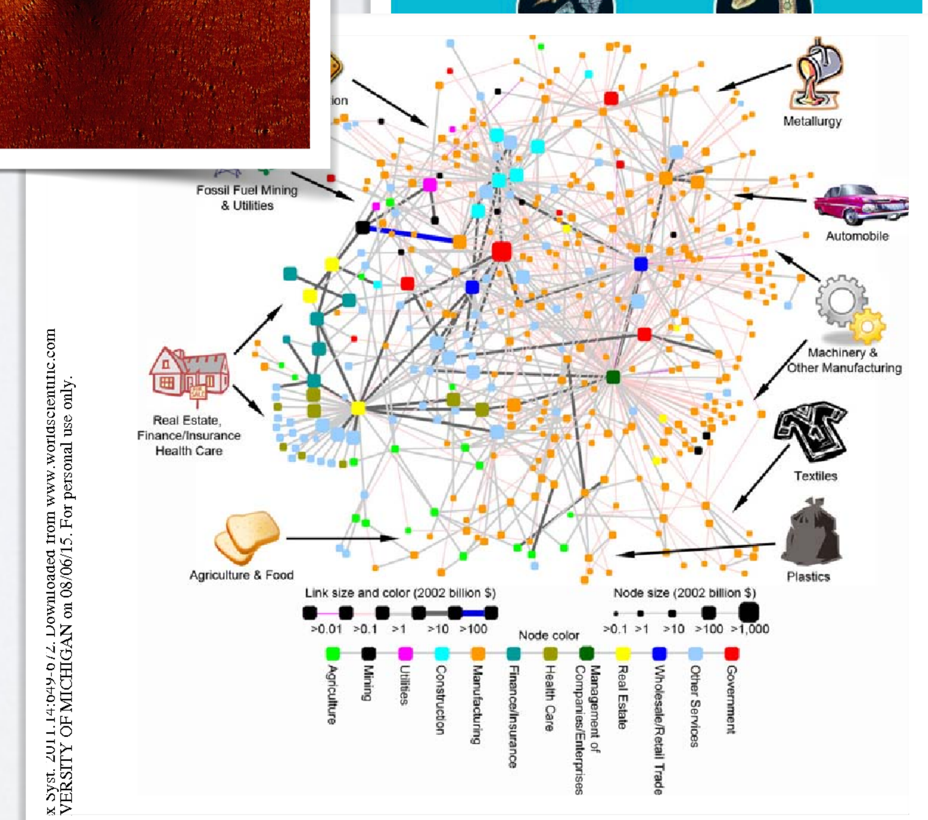

top predators
large sharks
smaller sharks
marlin
tuna
lancet fish
predators
squid
mackerel
filterers
lantern fish
amphipods
ocean sunfish
zooplankton
shrimp
copepods
pteropods
phytoplankton


A
Cumulative cases
Number isolated
Number quarantined
No control
Case isolation
Primary tracing
Secondary tracing
Number of cases
Week
B


c
d


(A)
Hubs forming a rich club
(B)
Shortest path
Long path
(C)
Triangular connections around node i
(D)
Modules
(E)
(F)


BOSTON
SACRAMENTO
OSAKA

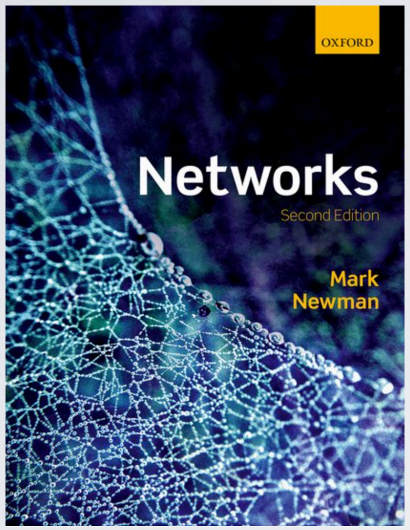

Metallurgy
Fossil Fuel Mining & Utilities
Automobile
Machinery & Other Manufacturing
Real Estate, Finance/Insurance Health Care
Textiles
Plastics
Agriculture & Food
Link size and color (2002 billion $)
Node size (2002 billion $)
Node color
Agriculture
Mining
Utilities
Construction
Manufacturing
Finance/Insurance
Health Care
Management of Companies/Enterprises
Real Estate
Wholesale/Retail Trade
Other Services
Government

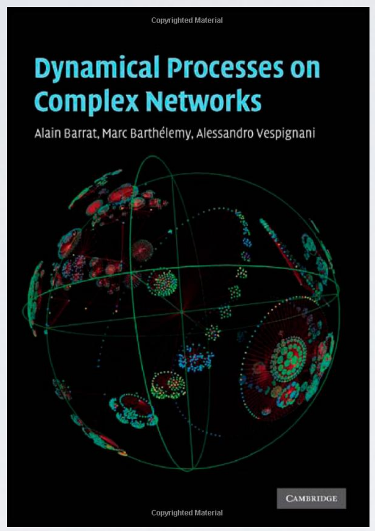
WHO ?

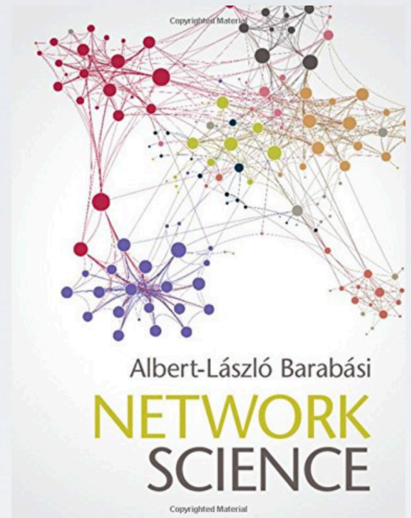
- Network scientists:
  - Physicists
  - Computer scientists
  - Mathematicians
  - Sociologists
  - => Work on similar problems, with converging vocabularies and references
- Applied network scientists
  - Geographers, biologists, social scientists, economists, etc.
  - =>Experts of i)their domain, and ii)complex networks analysis

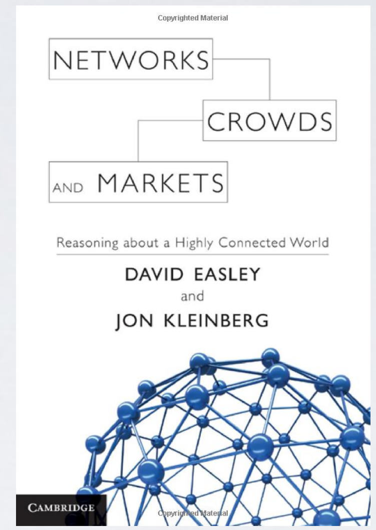
Materials

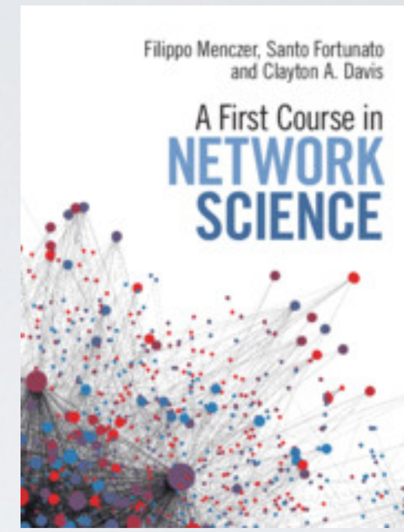
Lecture books











available free online ***available free online***

Reviews

SIAM Review
Vol. 45, No. 2, pp. 167–256
© 2003 Society for Industrial and Applied Mathematics

The Structure and Function of Complex Networks*

M. E. J. Newman†

REVIEWS OF MODERN PHYSICS, VOLUME 74, JANUARY 2002

Statistical mechanics of complex networks

Réka Albert* and Albert-László Barabási
Department of Physics, University of Notre Dame, Notre Dame, Indiana 46556

Characterization and Modeling of weighted networks

Marc Barthélemy[1], Alain Barrat[2], Romualdo Pastor-Satorras[3], and Alessandro Vespignani[2]

Physics Reports 486 (2010) 75–174
Contents lists available at ScienceDirect
Physics Reports
journal homepage: www.elsevier.com/locate/physrep
ELSEVIER

Community detection in graphs

Santo Fortunato *
Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I, Italy

Physics Reports 519 (2012) 97–125
Contents lists available at SciVerse ScienceDirect
Physics Reports
journal homepage: www.elsevier.com/locate/physrep
ELSEVIER

Temporal networks

Petter Holme [a,b,c,*], Jari Saramäki [d]
[a] IceLab, Department of Physics, Umeå University, 901 87 Umeå, Sweden
[b] Department of Energy Science, Sungkyunkwan University, Suwon 440–746, Republic of Korea
[c] Department of Sociology, Stockholm University, 106 91 Stockholm, Sweden
[d] Department of Biomedical Engineering and Computational Science, School of Science, Aalto University, 00076 Aalto, Espoo, Finland

Contents lists available at ScienceDirect
Physics Reports
journal homepage: www.elsevier.com/locate/physrep
ELSEVIER

Spatial networks

Marc Barthélemy *

Contents lists available at ScienceDirect
Physics Reports
journal homepage: www.elsevier.com/locate/physrep
ELSEVIER

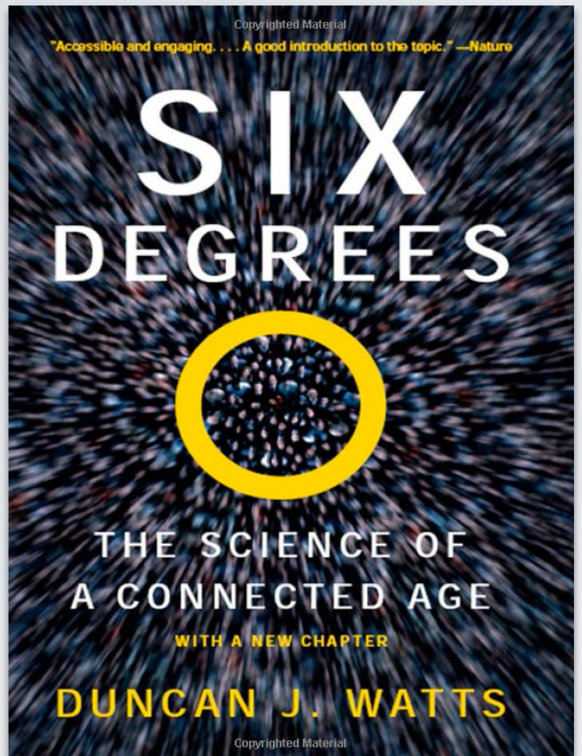
The structure and dynamics of multilayer networks

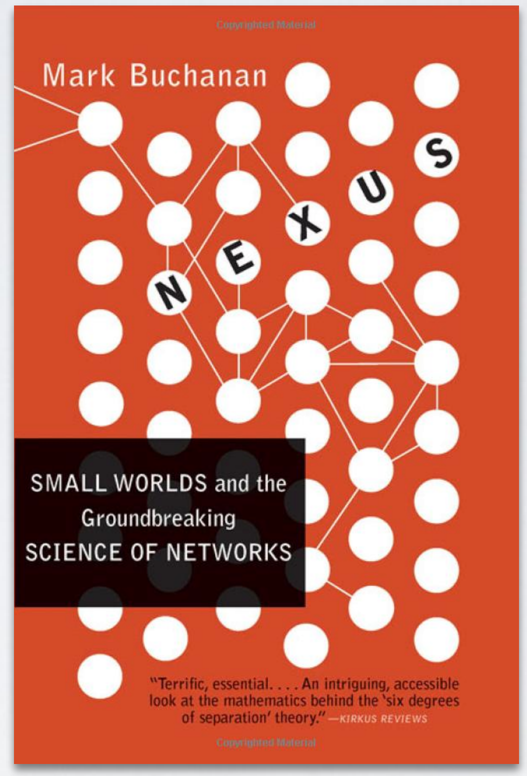
S. Boccaletti [a,b,*], G. Bianconi [c], R. Criado [d,e], C.I. del Genio [f,g,h], J. Gómez-Gardeñes [i], M. Romance [d,e], I. Sendiña-Nadal [j,e], Z. Wang [k,l], M. Zanin [m,n]

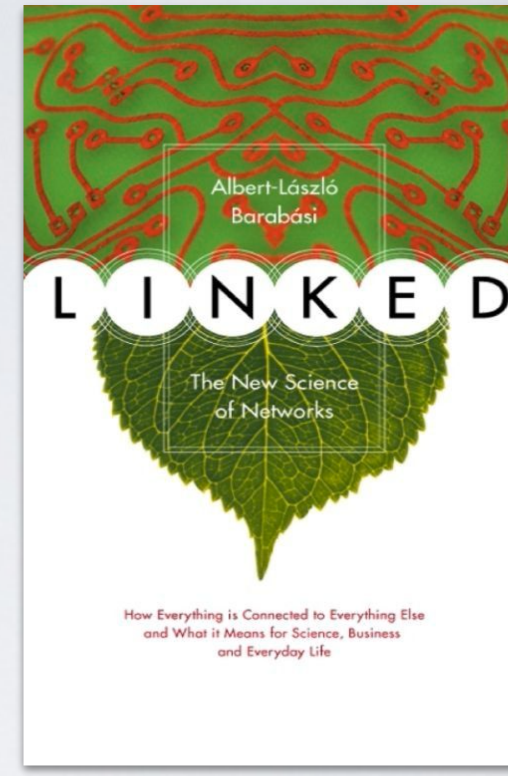
…and many more…all of them on arXiv.org!

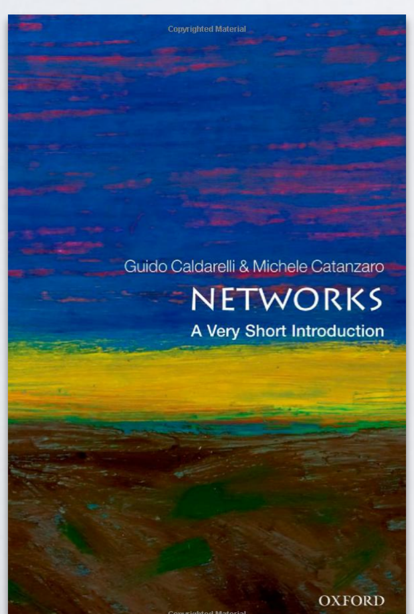
Materials

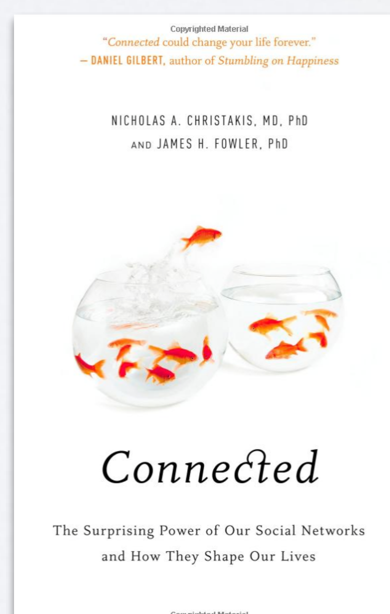
Pop-science books

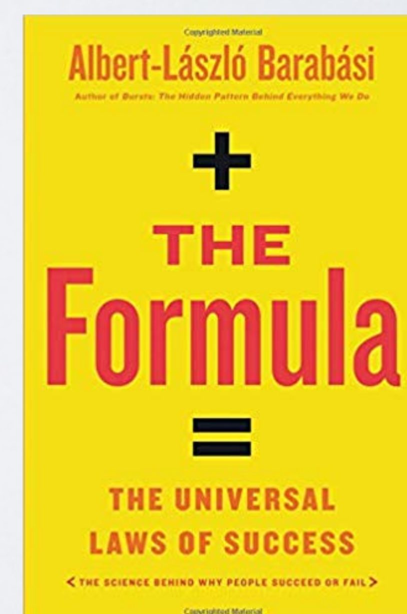








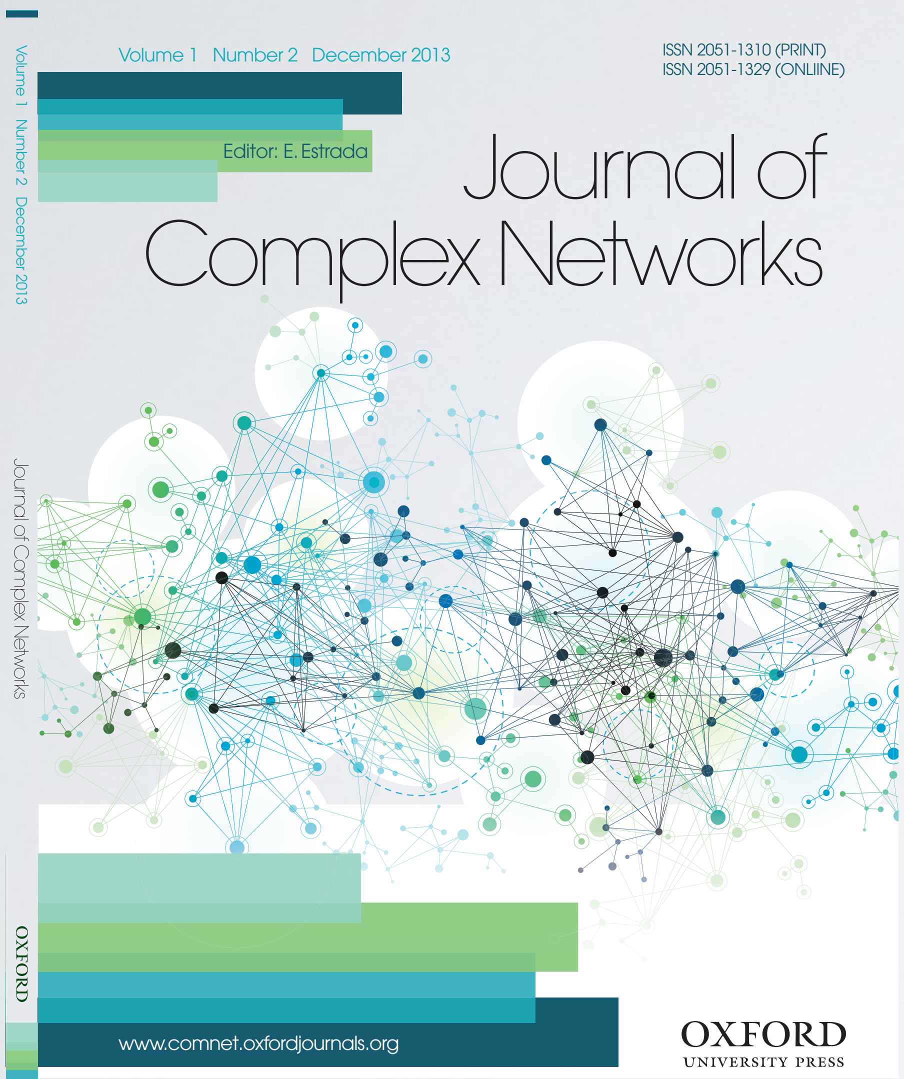


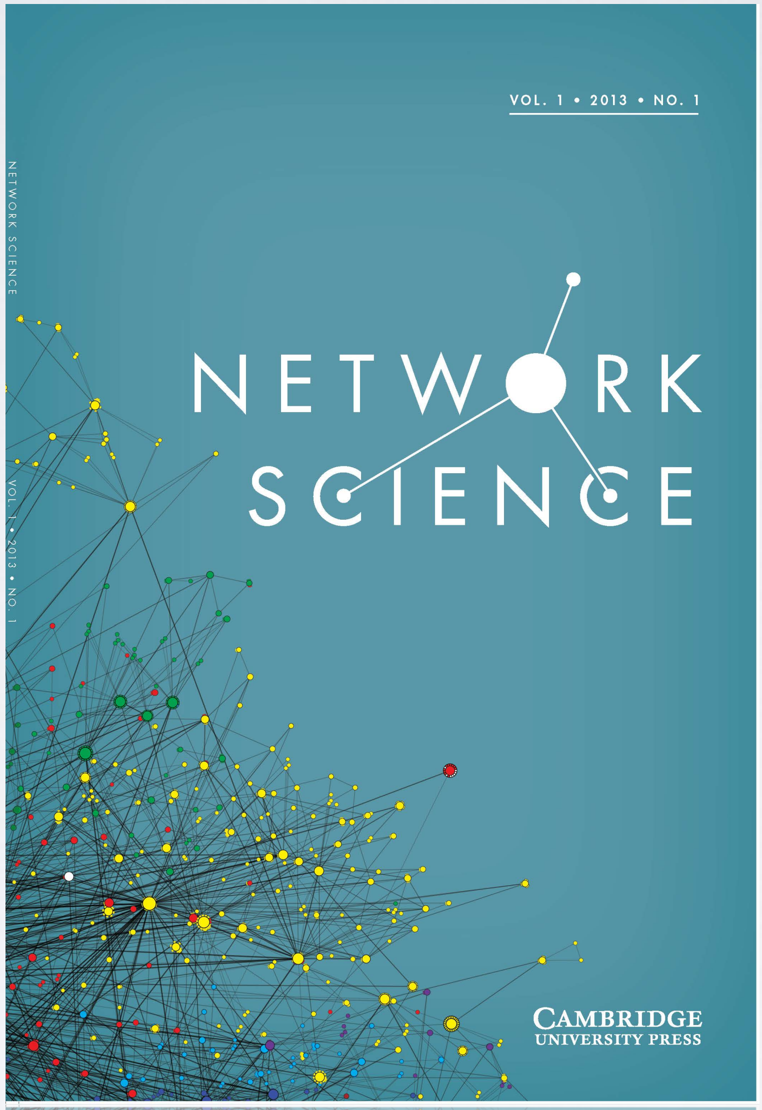


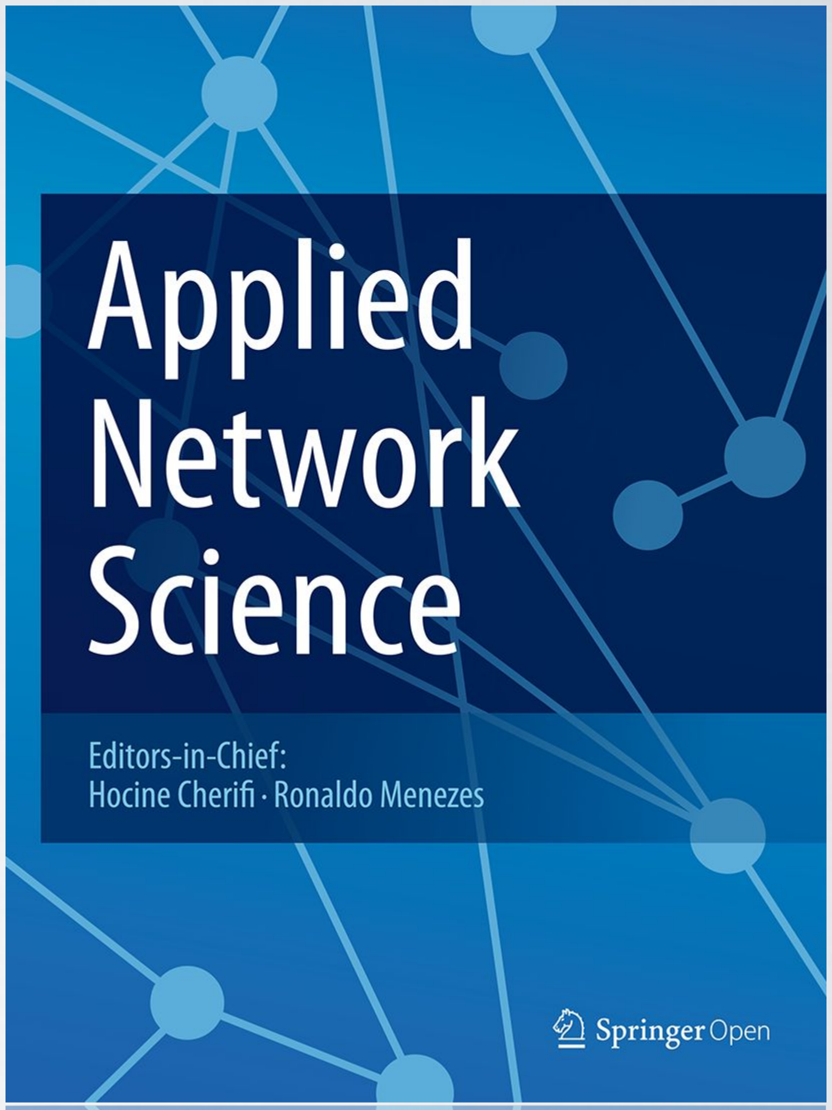
I have a copy I can lend

Materials

Specialized Journals







INTERNSHIPS

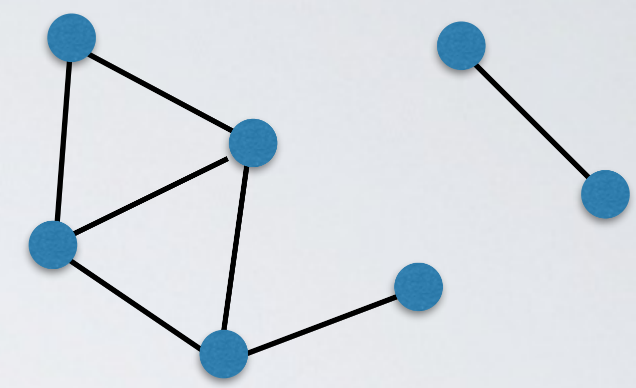
- If you are interested in doing an internship in this domain, feel free to contact me
- I'll officially propose some internships later

GRAPHS & NETWORKS

GRAPHS & NETWORKS

Network often refers to real systems

- www,
- social network
- metabolic network.
- <u>Language: (Network, node, link)</u>



Vertex	Edge
person	friendship
neuron	synapse
Website	hyperlink
company	ownership
gene	regulation

Graph is the mathematical representation of a network

- <u>Language: (Graph, vertex, edge)</u>

In most cases we will use the two terms interchangeably.

GRAPH REPRESENTATION

NETWORK REPRESENTATIONS

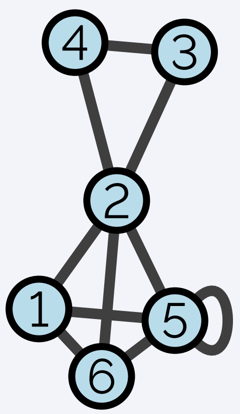
Networks: Graph notation

Graph notation : $G = (V, E)$

V	set of vertices/nodes.
E	set of edges/links.
$u \in V$	a node.
$(u, v) \in E$	an edge.

Network - Graph notation

Graph



Graph notation

$$G = (V, E)$$
$$V = \{1, 2, 3, 4, 5, 6\}$$
$$E = \{(1,2), (1,6), (1,5), (2,4), (2,3), (2,5), (2,6), (6,5), (5,5), (4,3)\}$$

NETWORK REPRESENTATIONS

- $G = (V, E)$
  - Often encoded as **edge list** or **adjacency list**

```
1 2
2 3
2 4
3 4
4 5
4 7
5 6
5 8
9 10
```

- Software: custom data structure and manipulation
  - add_nodes([i,j]), add_edge(i,j), …

```
1    2
2    1 3 4
3    2 4
4    2 3 5 7
5    4 6 8
6    5
7    4
8    5
9    10
10   9
```

- Libraries in many languages
  - Networkx (python)
  - igraph (python, C, R)
  - Graph-tools (python, C)

Types of Networks

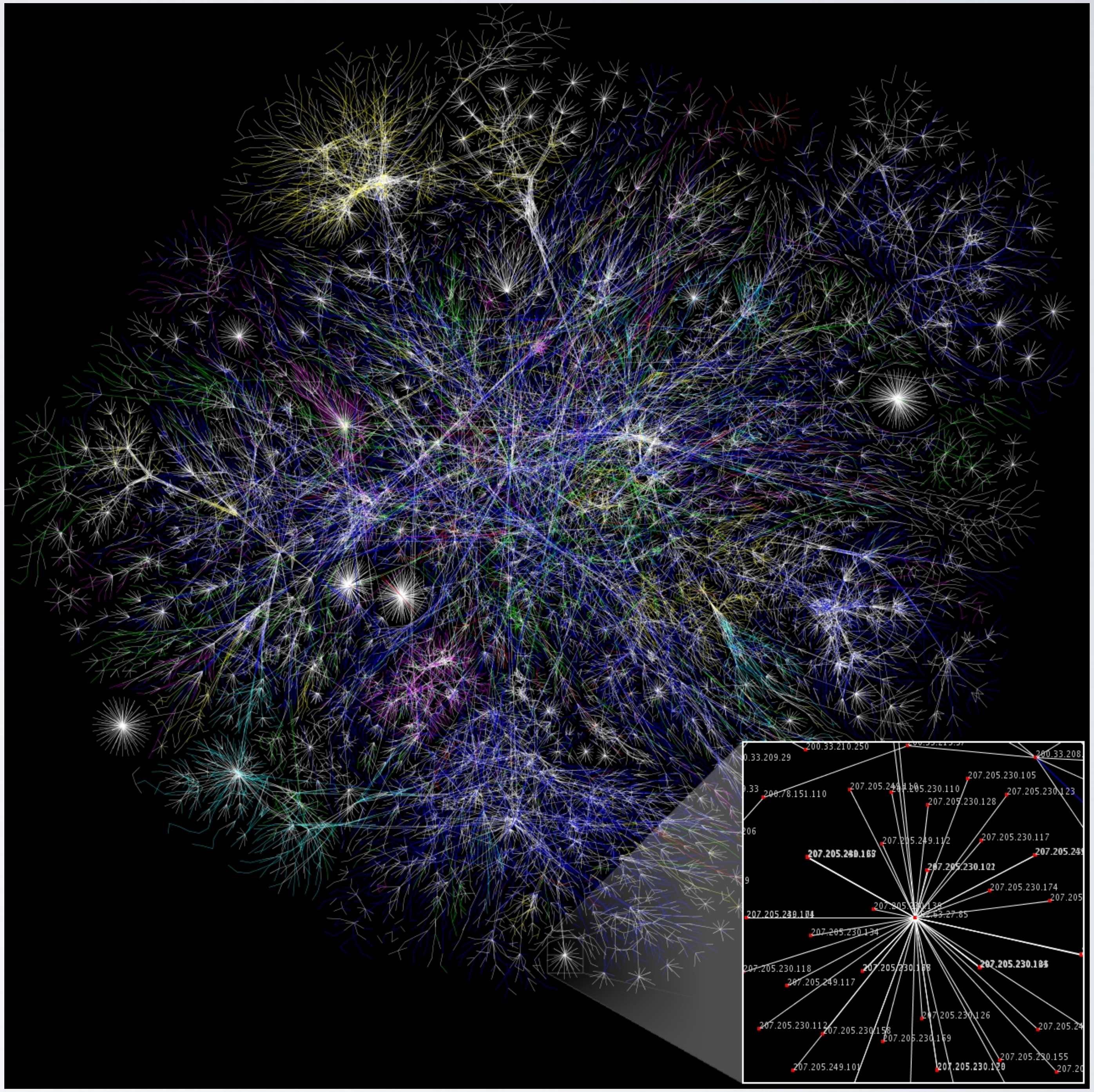
Undirected networks

$G=(V, E)$

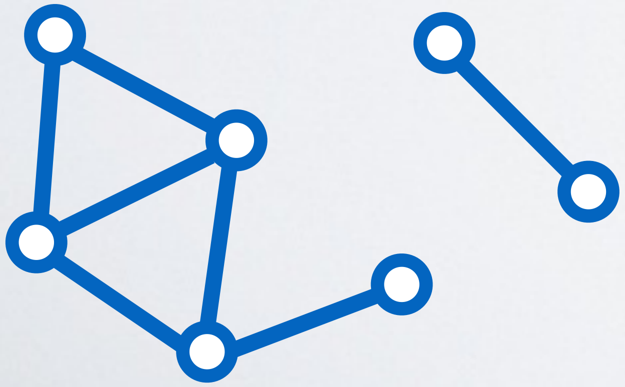
$(u,v) \in E \equiv (v,u) \in E$

- The directions of edges do not matter
- Interactions are possible between connected entities in both directions

Opte project



The Internet: Nodes - routers, Links - physical wires

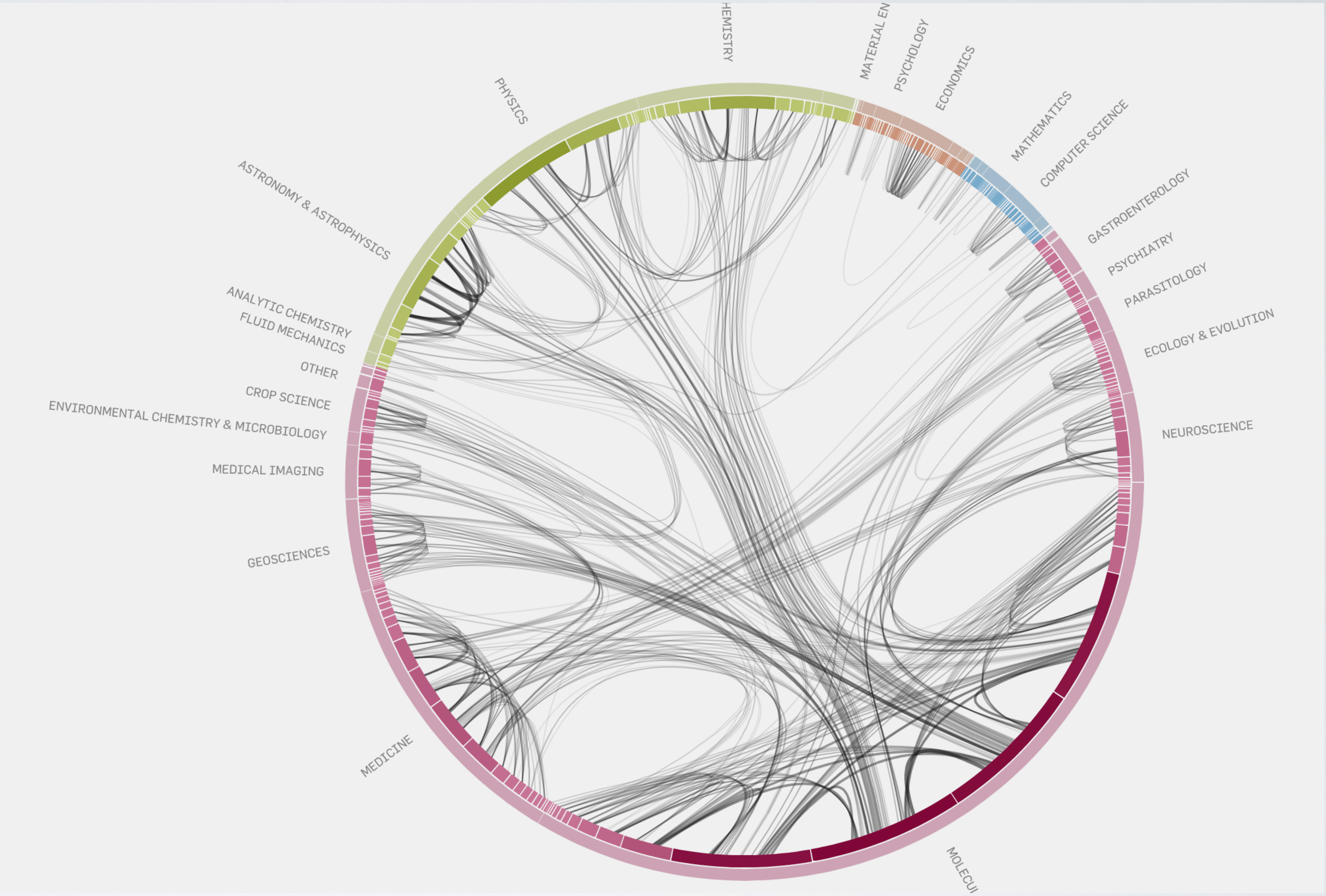


Directed networks

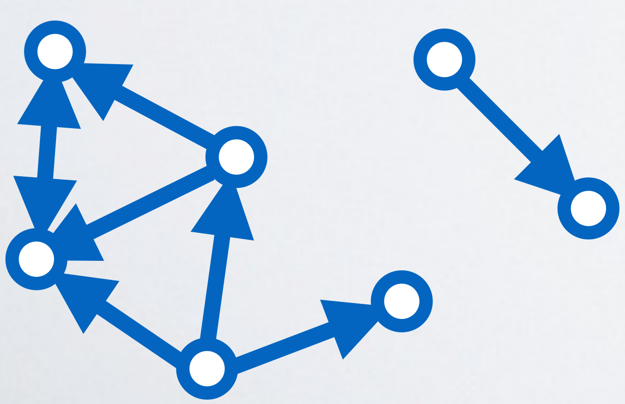
Moritz Stefaner, eigenfactor.com

$G=(V, E)$

$(u,v) \in E \not\equiv (v,u) \in E$



- The directions of edges matter
- Interactions are possible between connected entities only in specified directions



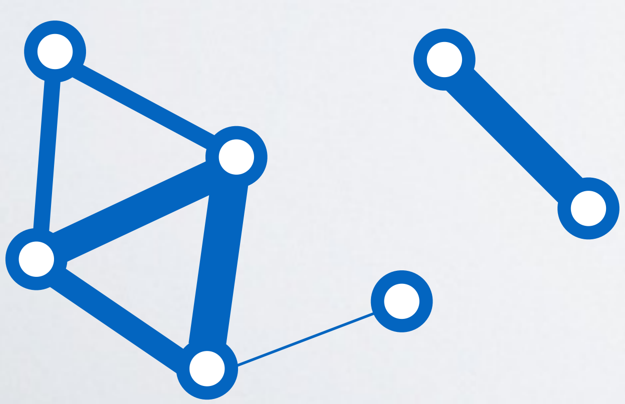
Citation network: Nodes - publications, Links - references

Weighted networks

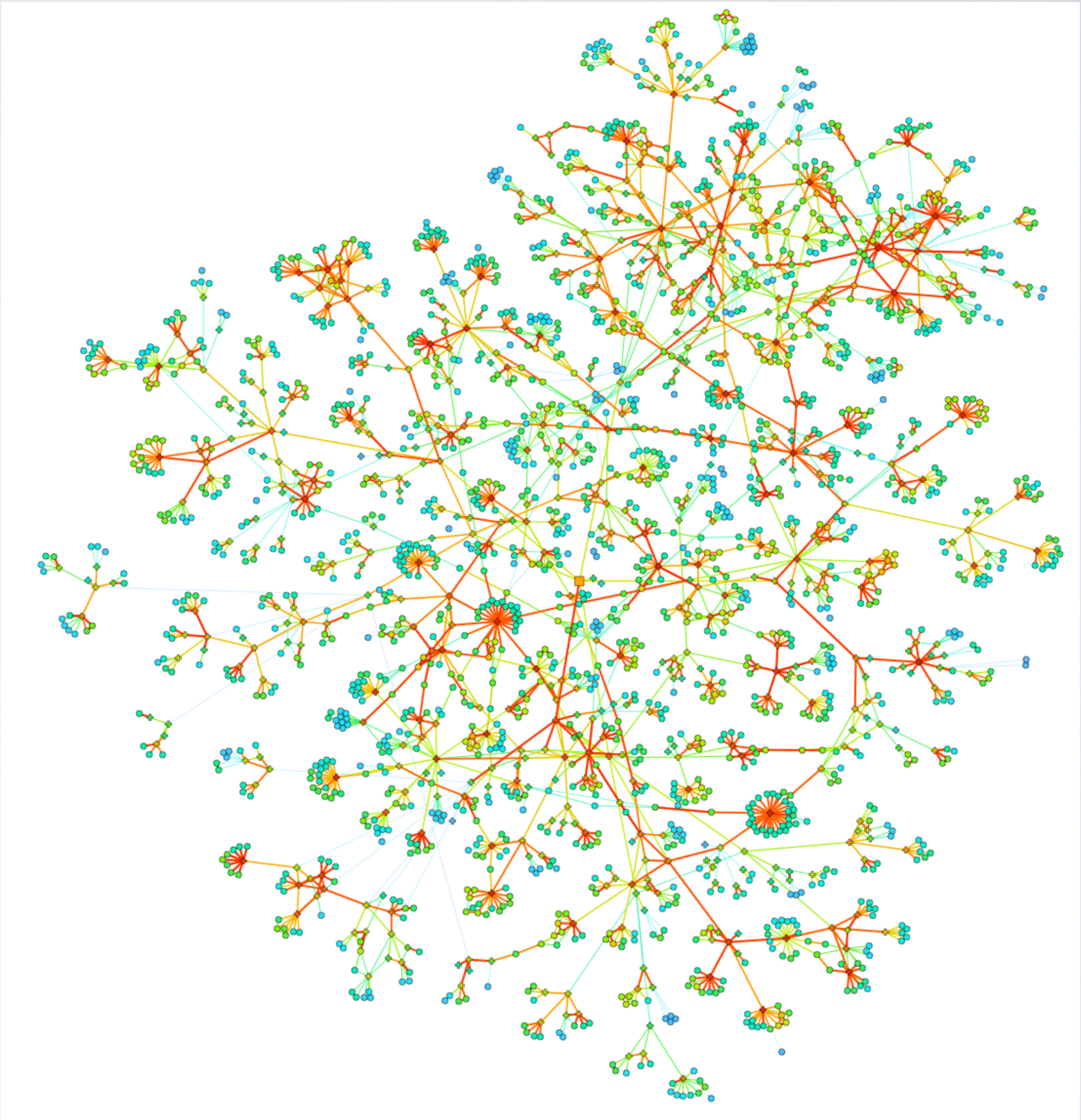
$G=(V, E, w)$

$w: (u,v) \in E \Rightarrow R$

- Strength of interactions are assigned by the weight of links

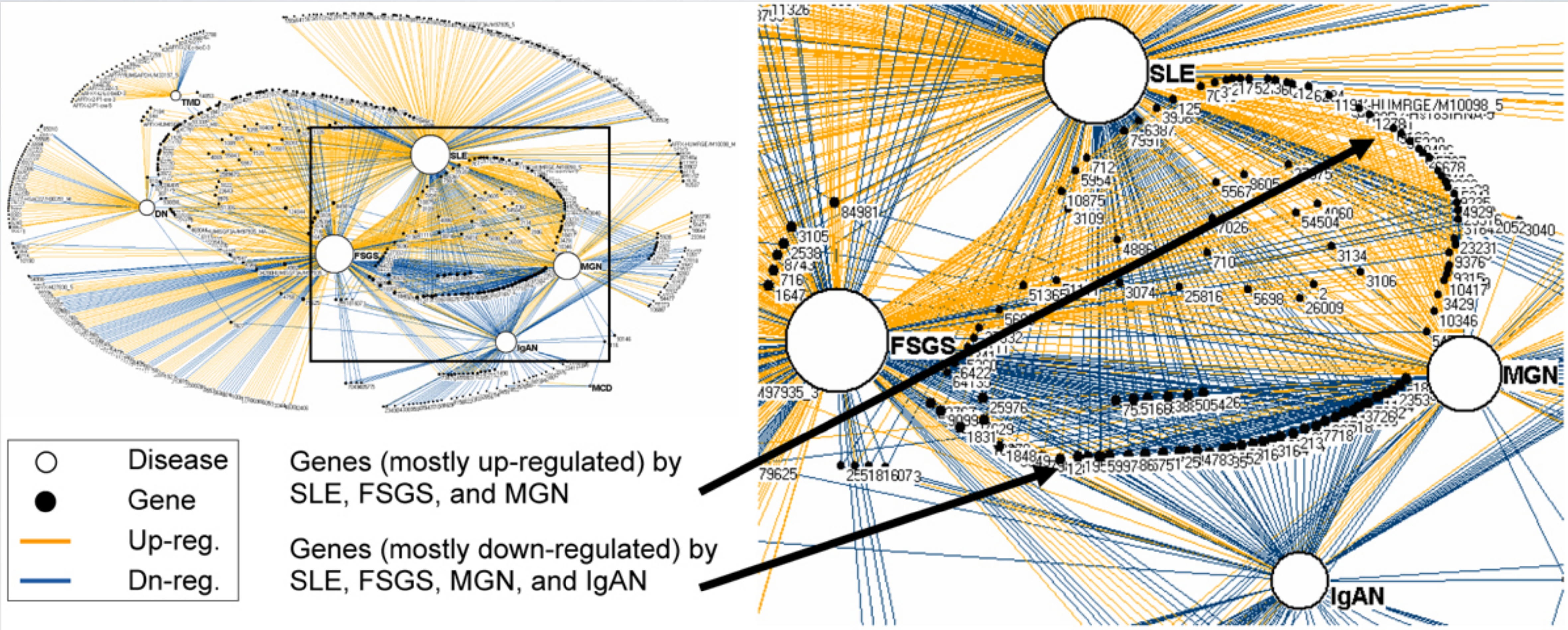


Onnela et.al. New Journal of Physics 9, 179 (2007).



Social interaction network: Nodes - individuals
Links - social interactions

Bipartite network

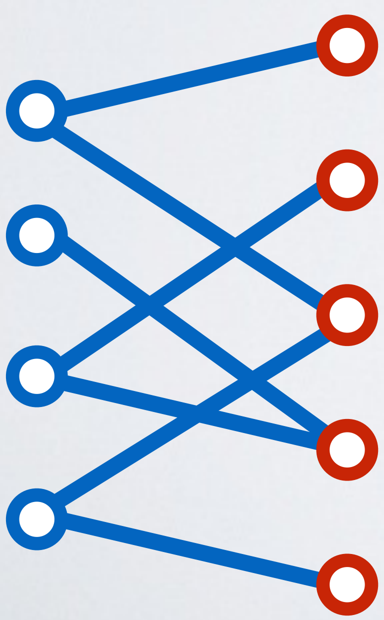


Bhavnani et.al. BMC Bioinformatics 2009, **10**(Suppl 9):S3

Gene-desease network:

Nodes - Desease (7)&Genes (747)

Links - gene-desease relationship



$$G=(U, V, E)$$

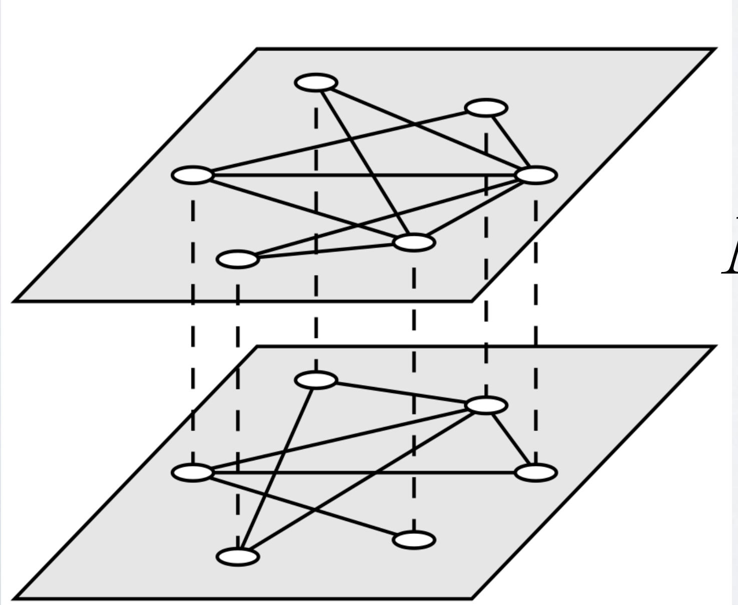
$$U \cap V = \varnothing$$

$$\forall (u,v) \in E,\ u \in U \ and\ v \in V$$

Multiplex and multilayer networks

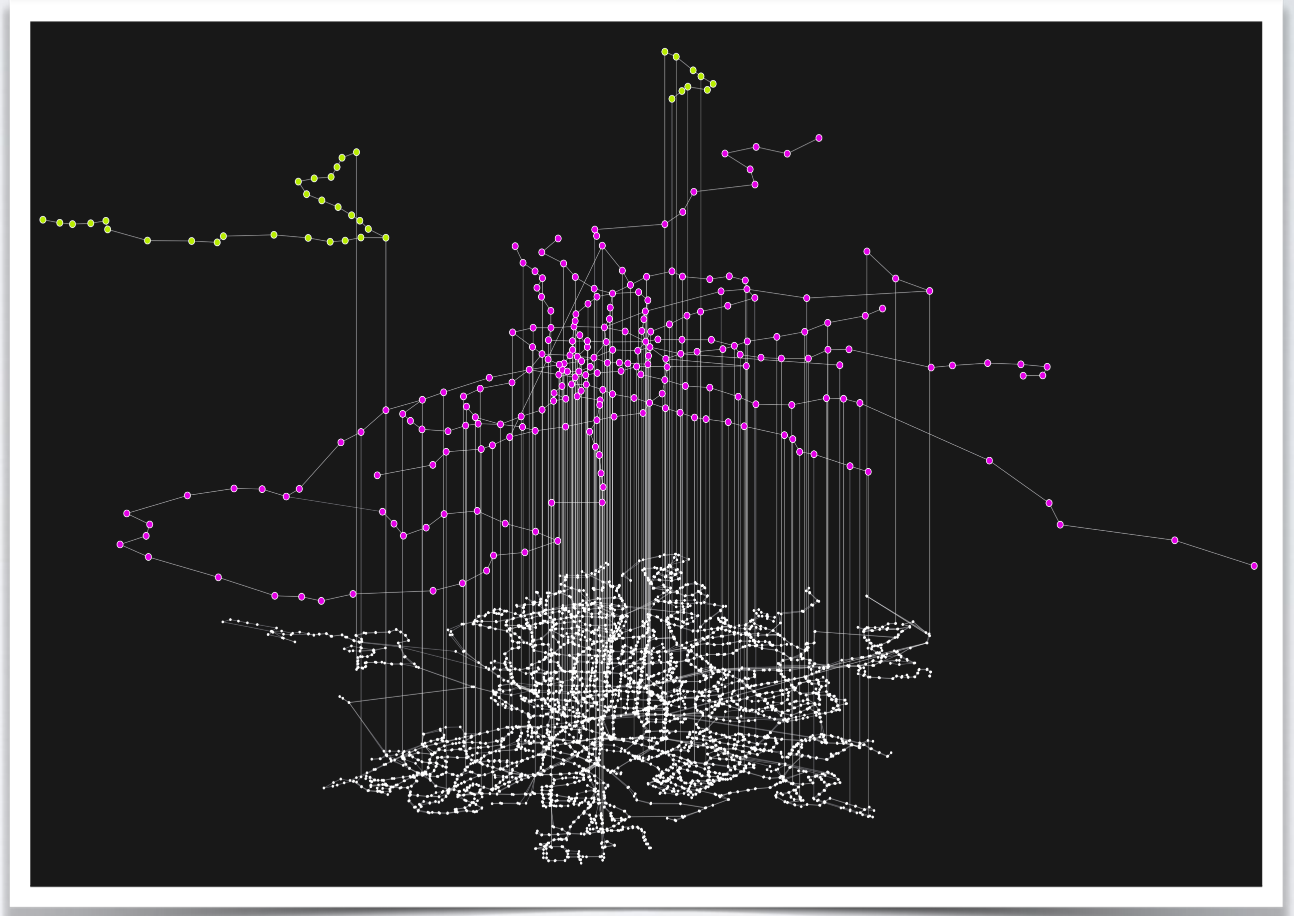
$G=(V, E_i),\ i=1\ldots M$

- Nodes can be present in multiple networks simultaneously
- These networks are connected (can influence each other) via the common nodes



Gomes et.al. Phys. Rev. Lett. 110, 028701 (2013)

$M=2$



[Mendez-Bermudez et al. 2017]

Temporal and evolving networks

$G=(V, E_t), (u,v,t,d) \in E_t$

t - time of interaction (u,v)
d - duration of interaction (u,v,t)

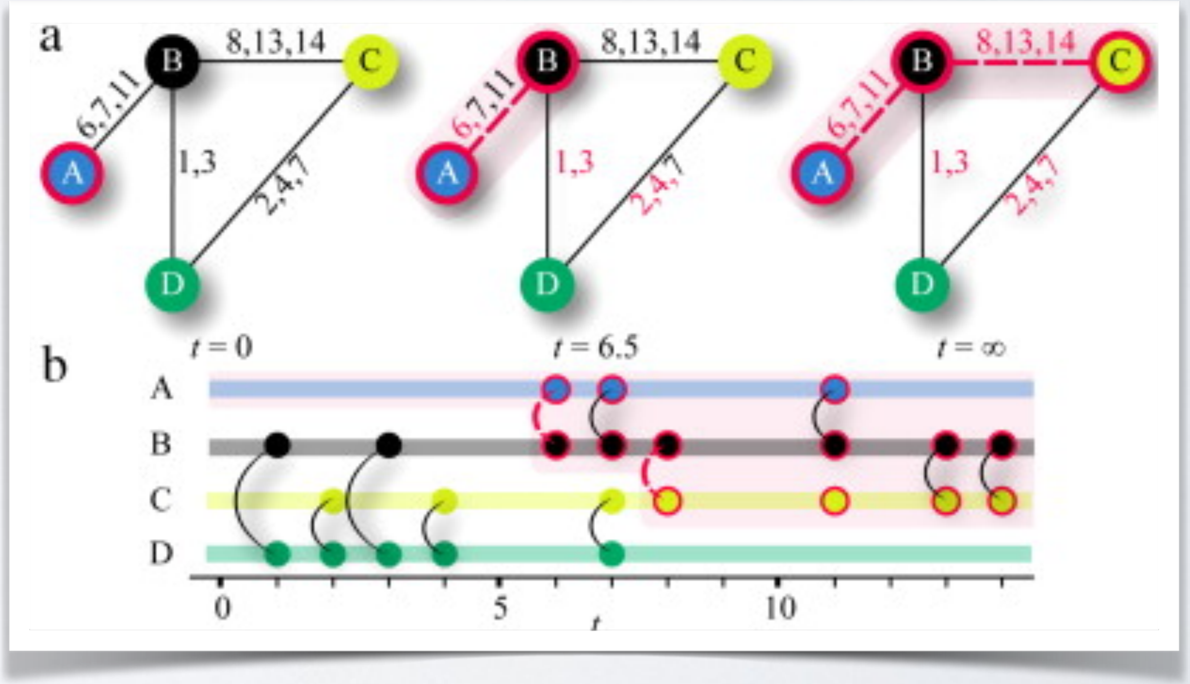
- Temporal links encode time varying interactions

$G=(V_{t'}, E_{t'})$

$v(t) \in V_{t'}$

$(u,v,t) \in E_{t'}$

- Dynamical nodes and links encode the evolution of the network



Mobile communication network
Nodes - individuals
Links - calls and SMS

GRAPH REPRESENTATION

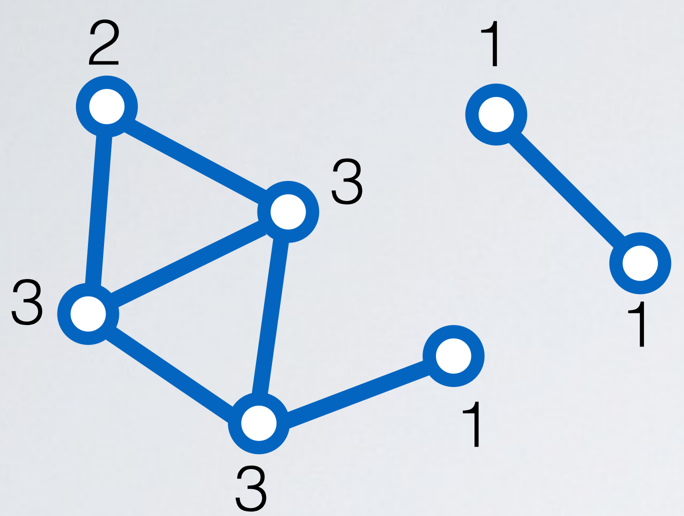
Node-Edge description

N_u	**Neighbourhood** of u, nodes sharing a link with u.
k_u	**Degree** of u, number of neighbors $\lvert N_u \rvert$.
N_u^{out}	**Successors** of u, nodes such as $(u,v) \in E$ in a directed graph
N_u^{in}	**Predecessors** of u, nodes such as $(v,u) \in E$ in a directed graph
k_u^{out}	**Out-degree** of u, number of outgoing edges $\lvert N_u^{out} \rvert$.
k_u^{in}	**In-degree** of u, number of incoming edges $\lvert N_u^{in} \rvert$
$w_{u,v}$	**Weight** of edge (u,v).
s_u	**Strength** of u, sum of weights of adjacent edges, $s_u = \sum_v w_{uv}$.

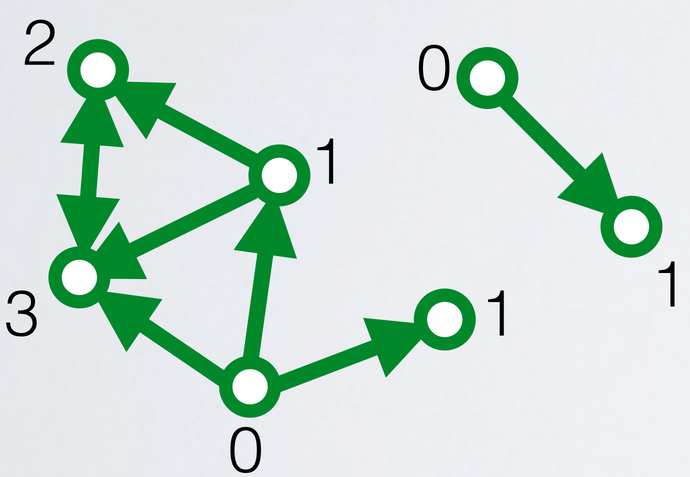
Node degree

Number of connections of a node

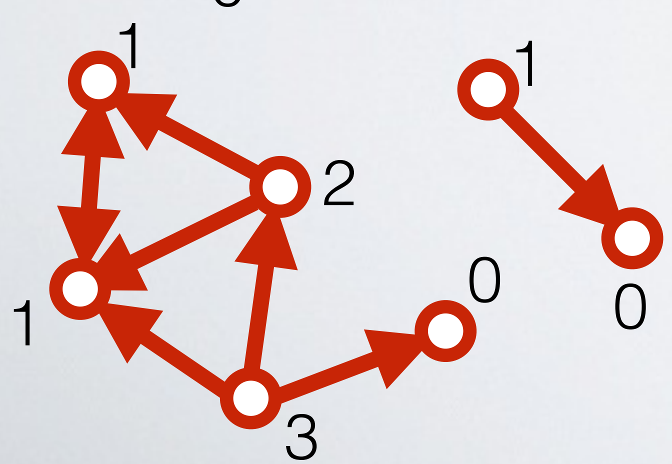
- Undirected network



- Directed network

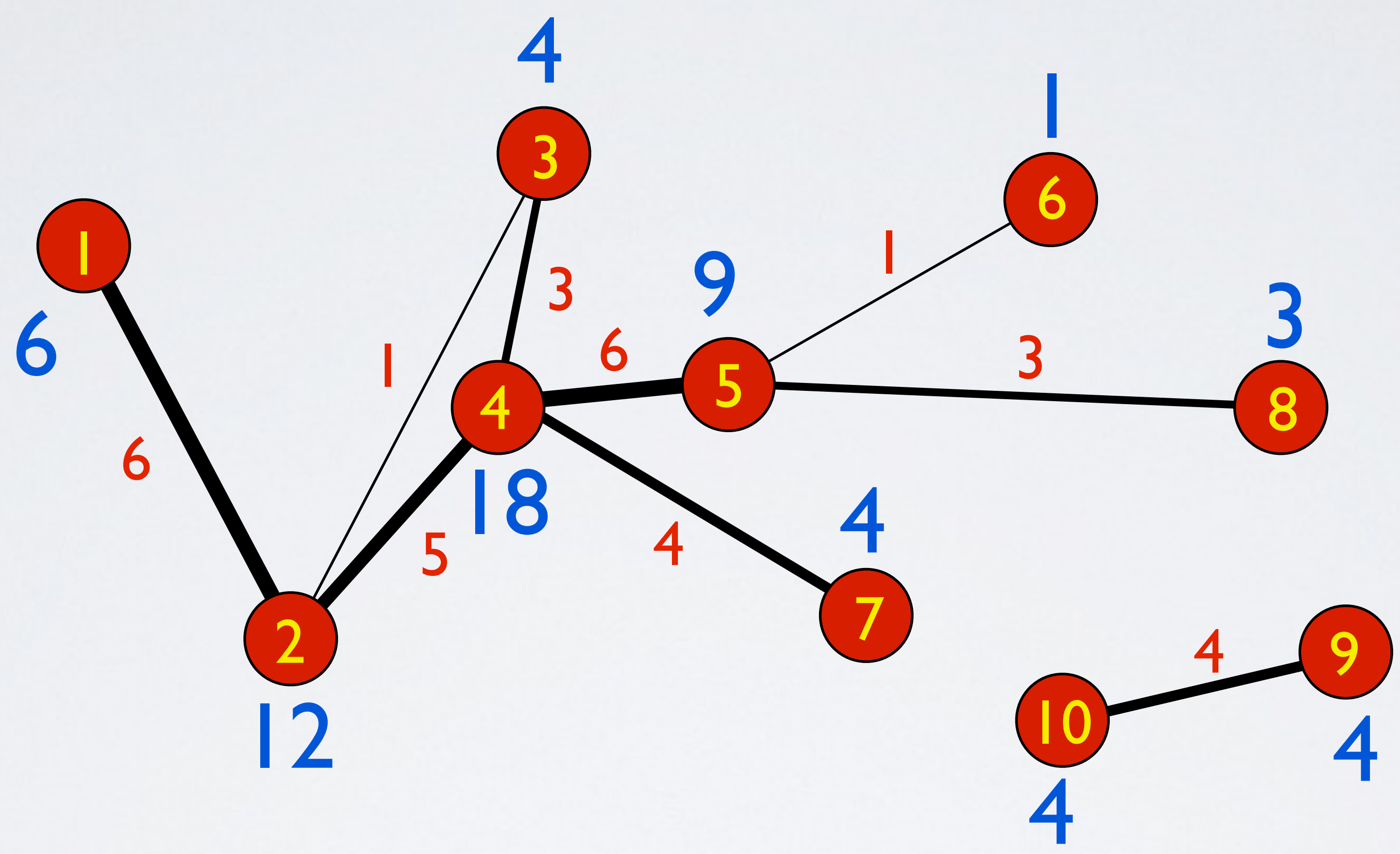


In degree



Out degree

Weighted degree: strength



DESCRIPTION OF GRAPHS

DESCRIPTION OF GRAPHS

- When confronted with a graph, how to describe it?
- How to compare graphs?
- What can we say about a graph?

SIZE

Counting nodes and edges

N/n	**size**: number of nodes $\|V\|$.
L/m	number of edges $\|E\|$
L_{max}	Maximum number of links

Undirected network: $\binom{N}{2} = N(N-1)/2$

Directed network: $\binom{N}{2} = N(N-1)$

DENSITY

Network descriptors - Nodes/Edges

$\langle k \rangle$ — **Average degree**: Real networks are sparse, i.e., typically $\langle k \rangle \ll n$. Increases slowly with network size, e.g., $\langle k \rangle \sim \log(m)$[a]

$$\langle k \rangle = \frac{2m}{n}$$

$d/d(G)$ — **Density**: Fraction of pairs of nodes connected by an edge in G.

$$d = L/L_{\max}$$

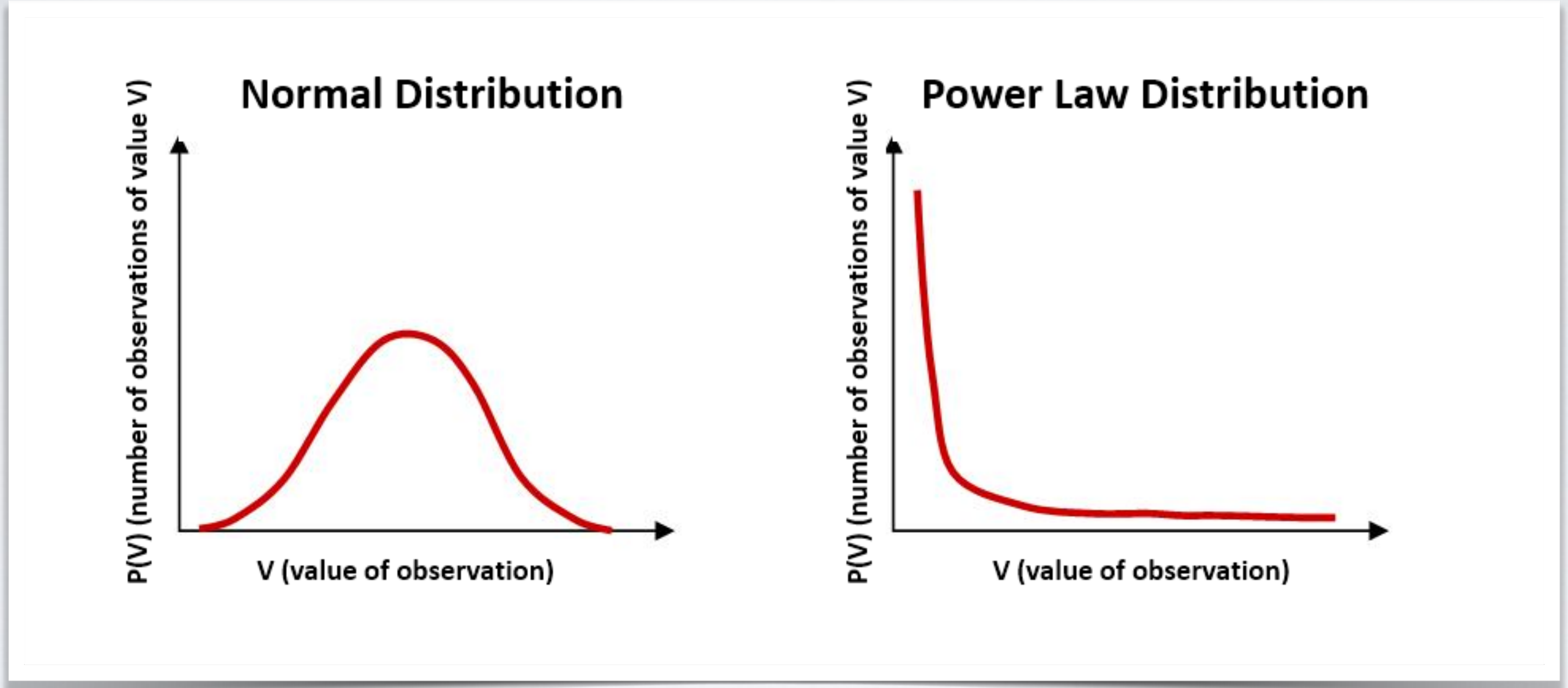
[a] Leskovec, Kleinberg, and Faloutsos 2005.

DENSITY

	#nodes	#edges	Density	avg. deg
Wikipedia	2M	30M	$1.5x10^{-5}$	30
Twitter 2015	288M	60B	$1.4x10^{-6}$	416
Facebook	1.4B	400B	$4x10^{-9}$	570
Brain c.	280	6393	0,16	46
Roads Calif.	2M	2.7M	$6x10^{-7}$	2,7
Airport	3k	31k	0,007	21

Beware: density hard to compare between graphs of different sizes

DEGREE DISTRIBUTION



PDF (Probability Distribution Function)

DEGREE DISTRIBUTION

- In a fully random graph (Erdos-Renyi), degree distribution is (close to) a normal distribution centered on the average degree

- In real graphs, in general, it is not the case:
  - A high majority of small degree nodes
  - A small minority of nodes with very high degree (Hubs)

- Often modeled by a **power law**
  - More details later in the course

SUBGRAPHS

Subgraphs

Subgraph $H(W)$ (induced subgraph): subset of nodes W of a graph $G = (V, E)$ and edges connecting them in G, i.e., subgraph $H(W) = (W, E'), W \subset V, (u, v) \in E' \iff u, v \in W \wedge (u, v) \in E$
Clique: subgraph with $d = 1$
Triangle: clique of size 3
Connected component: a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph
Strongly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths
Weakly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths if we disregard directions

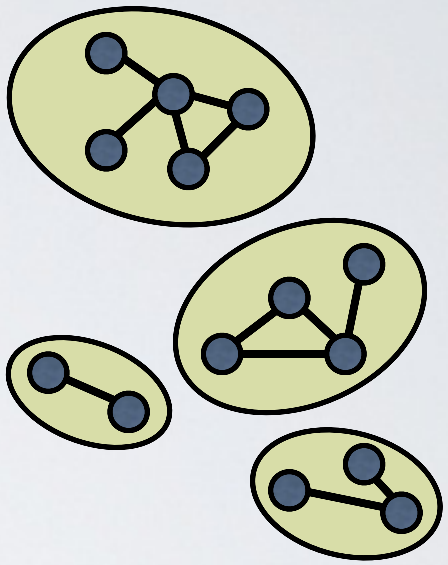
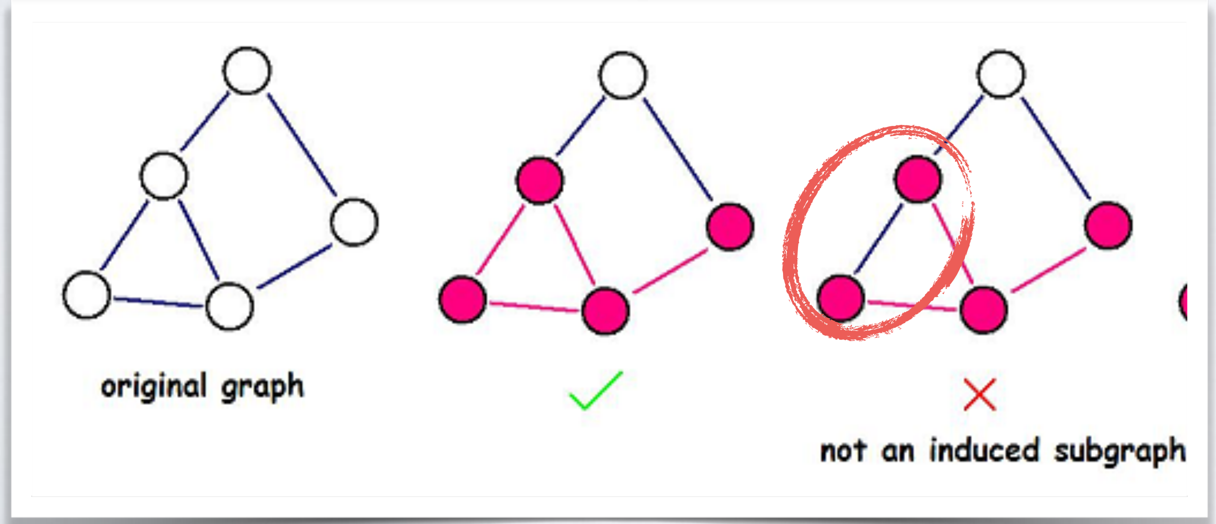
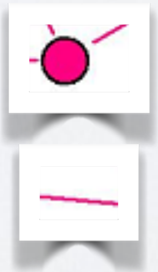


Figure after Newman, 2010



After "A. DZY Loves Physics"



Nodes/Edges in the subgraph

CLUSTERING COEFFICIENT

- **Clustering coefficient** or **triadic closure**
- Triangles are considered important in real networks
  - Think of social networks: *friends of friends are my friends*
  - # triangles is a big difference between real and random networks

CLUSTERING COEFFICIENT

Triangles counting

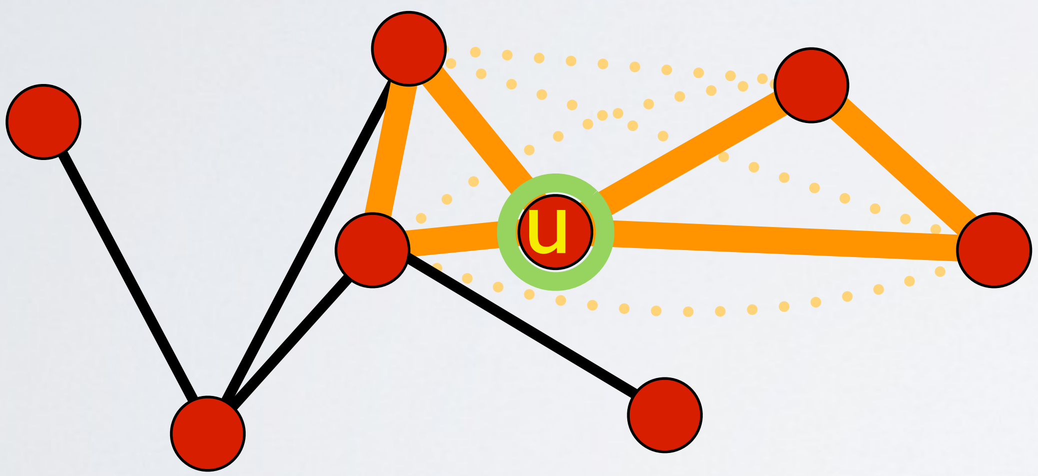
δ_u - **triads of u:** number of triangles containing node u
Δ - **number of triangles in the graph** total number of triangles in the graph, $\Delta = \frac{1}{3}\sum_{u \in V} \delta_u$.

> Each triangle in the graph is counted as a triad once by each of its nodes.

$\delta_u^{\max}$ - **triads potential of u:** maximum number of triangles that could exist around node u, given its degree: $\delta_u^{\max} = \tau(u) = \binom{k_i}{2}$
$\Delta^{\max}$ - **triangles potential of G:** maximum number of triangles that could exist in the graph, given its degree distribution: $\Delta^{\max} = \frac{1}{3}\sum_{u \in V} \delta^{\max}(u)$

CLUSTERING COEFFICIENT

C_u - **Node clustering coefficient:** density of the subgraph induced by the neighborhood of u, $C_u = d(H(N_u))$. Also interpreted as the fraction of all possible triangles in N_u that exist, $\frac{\delta_u}{\delta_u^{\max}}$



Edges: 2
Max edges: 4*3/2=6
C_u=2/6=1/3

Triangles=2
Possible triangles=$\binom{4}{2}$=6
C_u=2/6=1/3

CLUSTERING COEFFICIENT

$\langle C \rangle$ - **Average clustering coefficient:** Average clustering coefficient of all nodes in the graph, $\bar{C} = \frac{1}{N} \sum_{u \in V} C_u$.

Be careful when interpreting this value, since all nodes contributes equally, irrespectively of their degree, and that low degree nodes tend to be much more frequent than hubs, and their C value is very sensitive, i.e., for a node u of degree 2, $C_u \in 0, 1$, while nodes of higher degrees tend to have more contrasted scores.

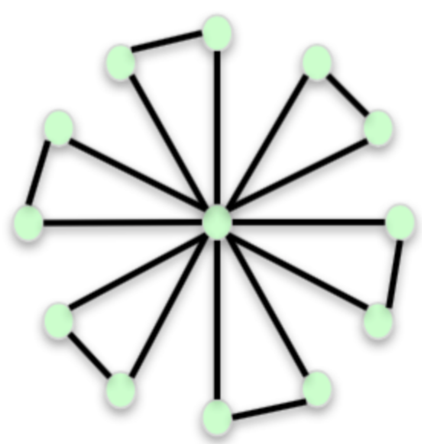
C^g - **Global clustering coefficient:** Fraction of all possible triangles in the graph that do exist, $C^g = \frac{3\Delta}{\Delta^{\max}}$

CLUSTERING COEFFICIENT

Global CC = Transitivity

Transitivity vs. Average Clustering Coefficient

Both measure the tendency for edges to form triangles.
Transitivity weights nodes with large degree higher.



- Most nodes have high LCC
- The high degree node has low LCC

Ave. clustering coeff. = 0.93
Transitivity = 0.23

- Most nodes have low LCC
- High degree node have high LCC

Ave. clustering coeff. = 0.25
Transitivity = 0.86

https://pynetwork.readthedocs.io/en/latest/connectivity.html

CLUSTERING COEFFICIENT

- Global CC:
  - In random networks, GCC = density
    - =>very small for large graphs

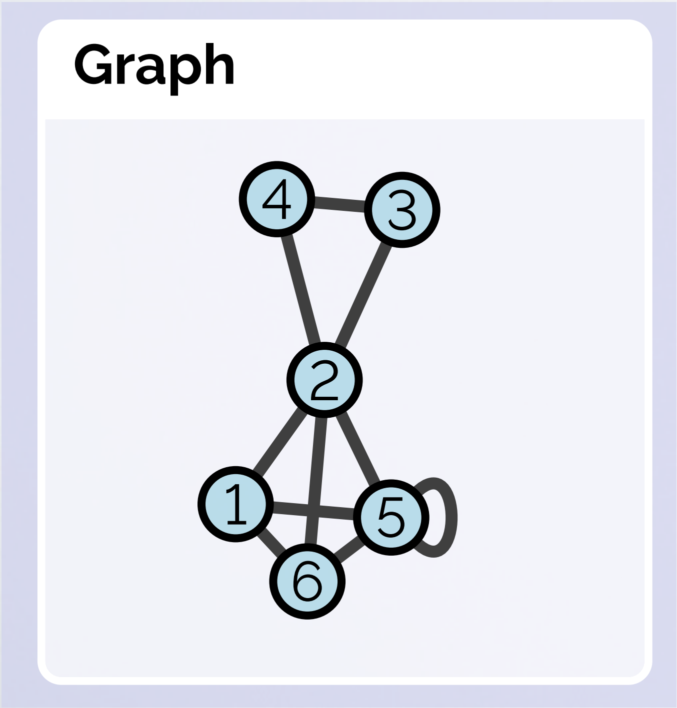
Network	Size	$\langle k \rangle$	C	C_{rand}	Reference
WWW, site level, undir.	153 127	35.21	0.1078	0.00023	Adamic, 1999
Internet, domain level	3015–6209	3.52–4.11	0.18–0.3	0.001	Yook *et al.*, 2001a, Pastor-Satorras *et al.*, 2001
Movie actors	225 226	61	0.79	0.00027	Watts and Strogatz, 1998
LANL co-authorship	52 909	9.7	0.43	1.8×10^{-4}	Newman, 2001a, 2001b, 2001c
MEDLINE co-authorship	1 520 251	18.1	0.066	1.1×10^{-5}	Newman, 2001a, 2001b, 2001c
SPIRES co-authorship	56 627	173	0.726	0.003	Newman, 2001a, 2001b, 2001c
NCSTRL co-authorship	11 994	3.59	0.496	3×10^{-4}	Newman, 2001a, 2001b, 2001c
Math. co-authorship	70 975	3.9	0.59	5.4×10^{-5}	Barabási *et al.*, 2001
Neurosci. co-authorship	209 293	11.5	0.76	5.5×10^{-5}	Barabási *et al.*, 2001
E. coli, substrate graph	282	7.35	0.32	0.026	Wagner and Fell, 2000
E. coli, reaction graph	315	28.3	0.59	0.09	Wagner and Fell, 2000
Ythan estuary food web	134	8.7	0.22	0.06	Montoya and Solé, 2000
Silwood Park food web	154	4.75	0.15	0.03	Montoya and Solé, 2000
Words, co-occurrence	460.902	70.13	0.437	0.0001	Ferrer i Cancho and Solé, 2001
Words, synonyms	22 311	13.48	0.7	0.0006	Yook *et al.*, 2001b
Power grid	4941	2.67	0.08	0.005	Watts and Strogatz, 1998
C. Elegans	282	14	0.28	0.05	Watts and Strogatz, 1998

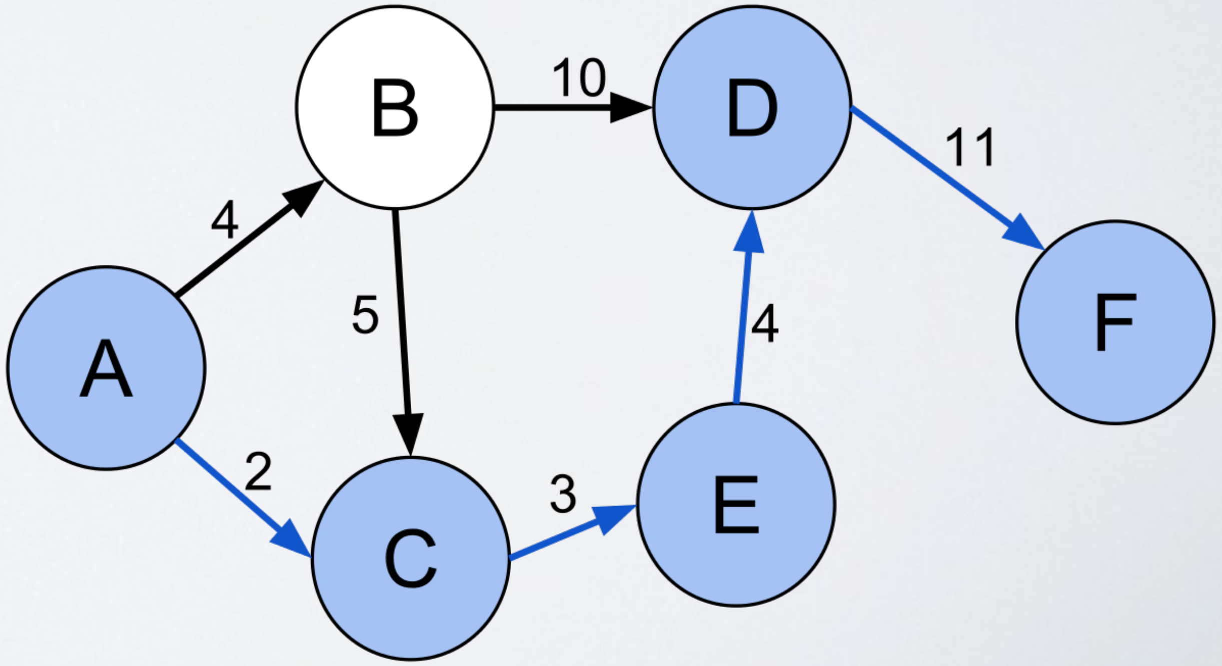
Albert, R. et.al. Rev. Mod. Phy. (2002)

PATH RELATED SCORES

Paths - Walks - Distance

Walk: Sequences of adjacent edges or nodes (e.g., **1.2.1.6.5** is a valid walk)
Path: a walk in which each node is distinct.
Path length: number of edges encountered in a path
Weighted Path length: Sum of the weights of edges on a path
Shortest path: The shortest path between nodes u, v is a path of minimal *path length*. Often it is not unique.
Weighted Shortest path: path of minimal *weighted path length*.
$\ell_{u,v}$**: Distance**: The distance between nodes u, v is the length of the shortest path





All shortest path algorithm

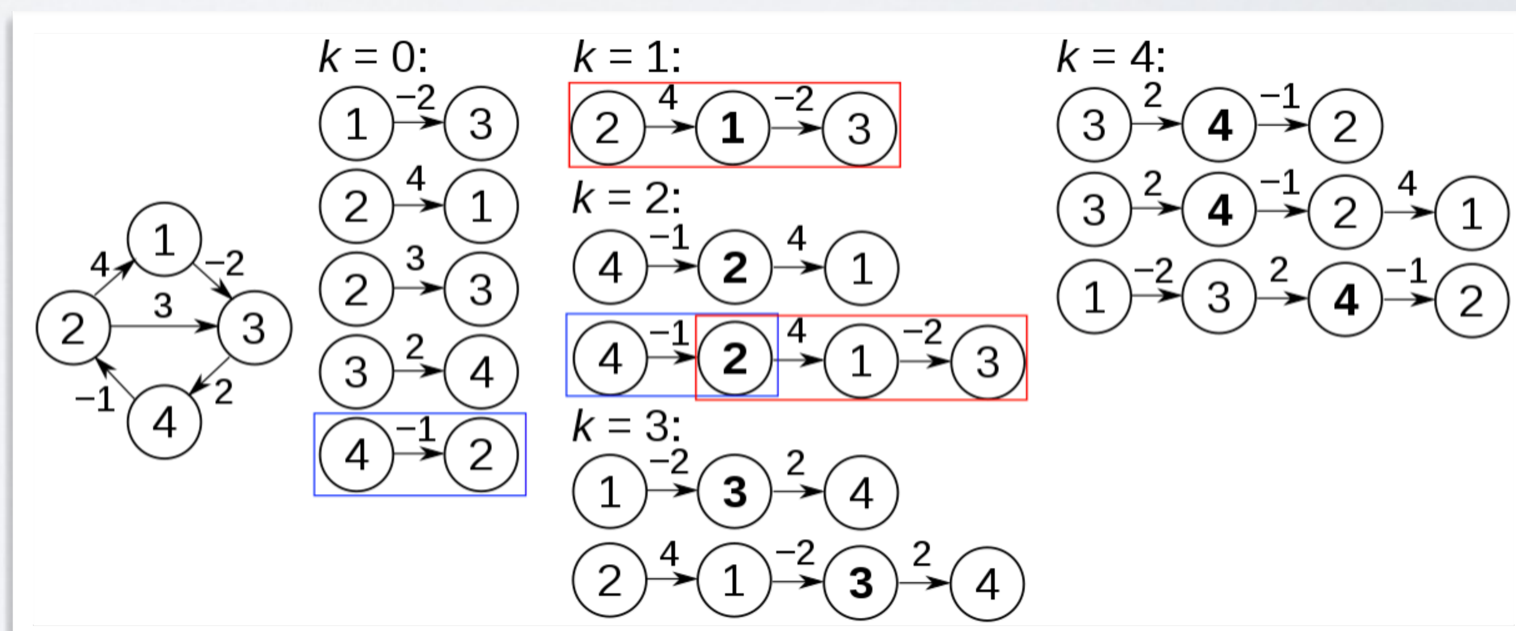
finding shortest paths in a **weighted graph** with **positive** or **negative edge weights** (but with no negative cycles)

```
proc FloydWarshall(G=(V,E,w))
// let dist be a |V| × |V| array of minimum distances initialized to ∞ (infinity)
for each edge (u,v)
   dist[u][v] ← w(u,v)  // the weight of the edge (u,v)
for each vertex v
   dist[v][v] ← 0
for k from 1 to |V|
   for i from 1 to |V|
      for j from 1 to |V|
         if dist[i][j] > dist[i][k] + dist[k][j]
             dist[i][j] ← dist[i][k] + dist[k][j]
         end if
```

Checking and updating all paths going through nodes k=1, 2, 3, … , N by assuming that:

$$shp(i,j,k)= min(shp(i,j,k-1)), shp(i,k,k-1)+shp(k,j,k-1))$$

Complexity: $O(n^3)$



PATH RELATED SCORES

Network descriptors 2 - Paths

ℓ_{max}	**Diameter**: maximum *distance* between any pair of nodes.
$\langle \ell \rangle$	**Average distance**:

$$\langle \ell \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} d_{ij}$$

AVERAGE PATH LENGTH

- The famous 6 degrees of separation (Milgram experiment)
  - (More on that next slide)
- Not too sensible to noise
- Tells you if the network is "stretched" or "hairball" like

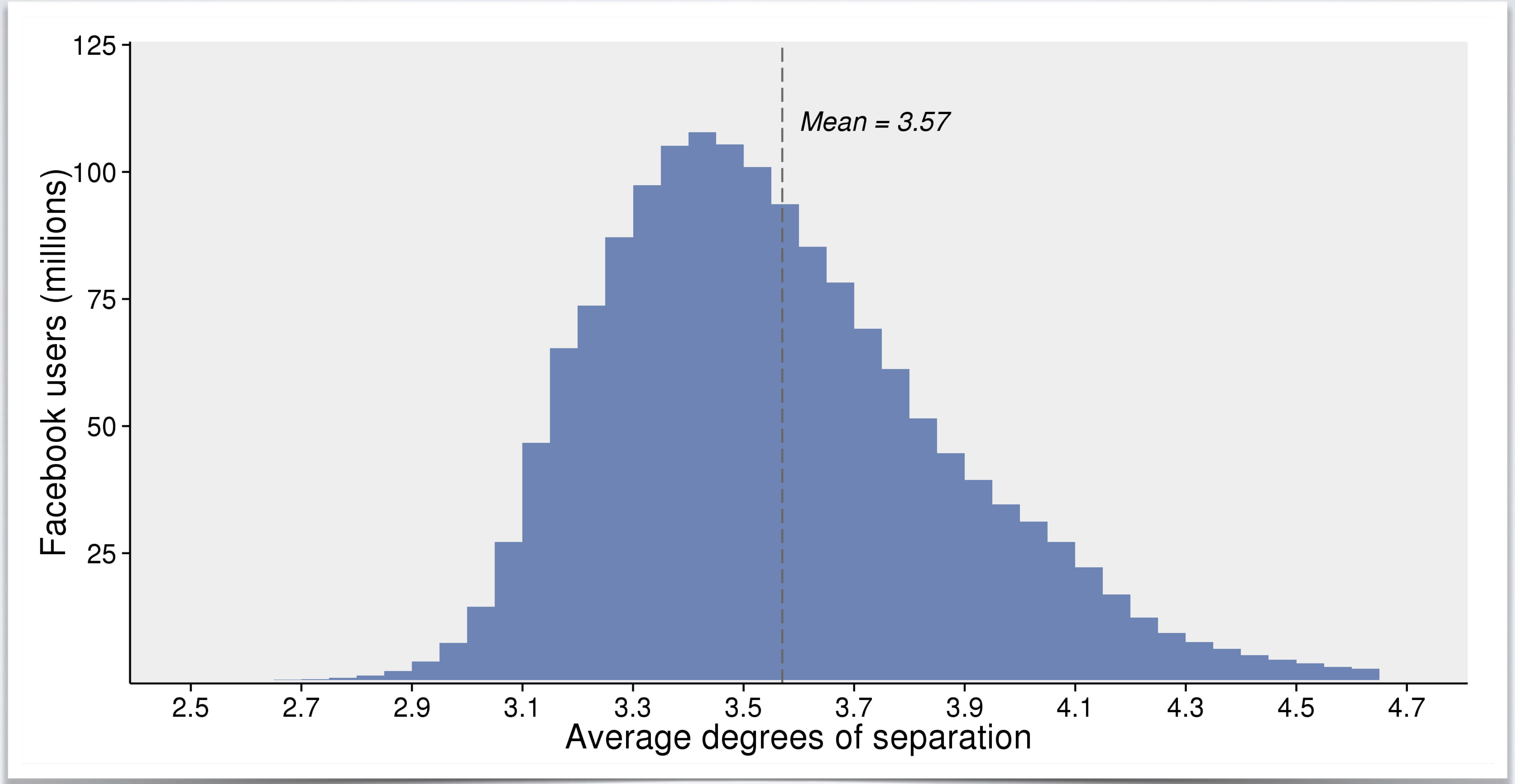
SIDE-STORY: MILGRAM EXPERIMENT

- Small world experiment (60's)
  - Give a (physical) mail to random people
  - Ask them to send to someone they don't know
    - They know his city, job
  - They send to their most relevant contact
- Results: In average, 6 hops to arrive

SIDE-STORY: MILGRAM EXPERIMENT

- Many criticism on the experiment itself:
  - Some mails did not arrive
  - Small sample
  - …
- Checked on “real” complete graphs (giant component):
  - MSN messenger
  - Facebook
  - The world wide web
  - …

SIDE-STORY: MILGRAM EXPERIMENT



Facebook

SMALL WORLD

Small World Network

A network is said to have the **small world** property when it has some structural properties. The notion is not quantitatively defined, but two properties are required:

- Average distance must be short, i.e., $\langle \ell \rangle \approx \log(N)$
- Clustering coefficient must be high, i.e., much larger than in a random network , e.g., $C^g \gg d$, with d the network density

More on this during the random network class

NETWORK DESCRIPTORS

- Many other network descriptors exist:
  - Modularity (later in community detection class)
  - Centralization (comparing the centrality scores between most central and less central, see later)
  - Rich-club coefficient: tendency of high-degrees to connected to high-degrees, cf random network class
  - Motif profiles (how often do specific subgraphs appear)
  - Network Resilience (see practicals)
  - etc.

GRAPHS AS MATRICES

Matrices in short

Matrices are mathematical objects that can be thought as *tables* of numbers. The size of a matrix is expressed as $m \times n$, for a matrix with m rows and n columns. **The order (row/column) is important**.
M_{ij} is a notation representing the element on **row** m and **column** j.

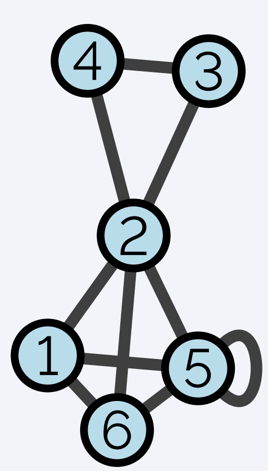
ADJACENCY MATRIX

A - Adjacency matrix

The most natural way to represent a graph as a matrix is called the Adjacency matrix A. It is defined as a square matrix, such as the number of rows (and the number of columns) is equal to the number of nodes N in the graph. Nodes of the graph are numbered from 1 to N, and there is an edge between nodes i and j if the corresponding position of the matrix A_{ij} is not 0.

- A value on the diagonal means that the corresponding node has a **self-loop**
- the graph is **undirected**, the matrix is **symmetric**: $A_{ij} = A_{ji}$ for any i, j.
- In an **unweighted** network, and edge is represented by the value 1.
- In a **weighted** network, the value A_{ij} represents the **weight** of the edge (i, j)

Graph



A - Adjacency Mat.

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

ADJACENCY MATRIX

Typical operations on A

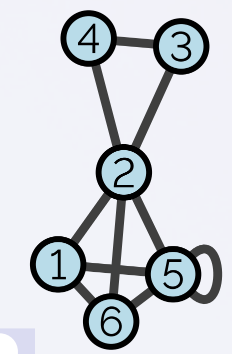
Some operations on Adjacency matrices have straightforward interpretations and are frequently used

---

Multiplying A by **itself** allows to know the number of walks of a given length that exist between any pair of nodes: A^2_{ij} corresponds to the number of walks of length 2 from node i to node j, A^3_{ij} to the number of walks of length 3, etc.

Multiplying A by a **column vector** W of length $1 \times N$ can be thought as setting the i th value of the vector to the ith node, and each node *sending* its value to its neighbors (for undirected graphs). The result is a column vector with N elements, the ith element corresponding to the sum of the values of its neighbors in W. This is convenient when working with **random walks** or **diffusion** phenomenon.

Graph



A - Adjacency Mat.

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

A^2

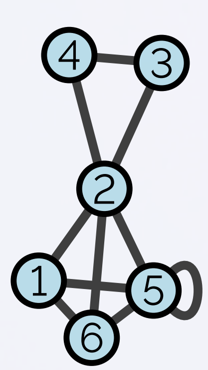
$$\begin{pmatrix} 3 & 2 & 1 & 1 & 3 & 2 \\ 2 & 5 & 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 \\ 3 & 3 & 1 & 1 & 4 & 3 \\ 2 & 2 & 1 & 1 & 3 & 3 \end{pmatrix}$$

LAPLACIAN

Graph Laplacian

The **Graph Laplacian**, or **Laplacian Matrix** of a graph is a variant of the Adjacency matrix, often used in *Graph theory* and *Spectral Graph Theory*.
It is defined as $D - A$, with D the *Degree matrix* of the graph, defined as a $N \times N$ matrix with $D_{ii} = k_i$ and zeros everywhere else.

Graph



A - Adjacency Mat.

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

D - Degree Matrix

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

L - Laplacian

$$\begin{pmatrix} 3 & -1 & 0 & 0 & -1 & -1 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 2 & 0 & 0 \\ -1 & -1 & 0 & 0 & 4 & -1 \\ -1 & -1 & 0 & 0 & -1 & 3 \end{pmatrix}$$

SPECTRAL GRAPH THEORY

Spectral properties of L

Eigenvalues of the Laplacian have many applications, such as *spectral clsutering*, *graph matching*, *embedding*, etc. Assuming G undirected with eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$, here are some interesting properties:

- The smallest eigenvalue λ_i equals 0
- The number of 0 eigenvalues gives the number of connected components

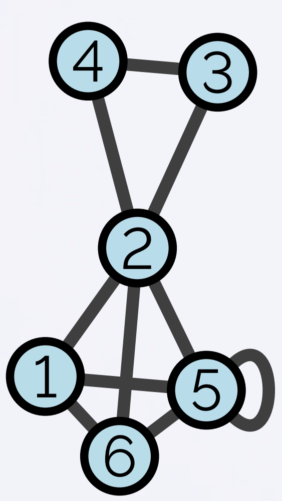
More with Pierre Borgnat

RANDOM WALK MATRIX

Random Walk matrix

Another useful matrix of a graph is the **Random Walk Transition Matrix** R. It is the column normalized version of the adjacency matrix. R_{ij} can be understood as the probability for a random walker located on node i to move to j.

Graph



A - Adjacency Mat.

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

Random W. mat.

$$\begin{pmatrix} 0 & \frac{1}{5} & 0 & 0 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ 0 & \frac{1}{5} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{5} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{5} & 0 & 0 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{5} & 0 & 0 & \frac{1}{4} & 0 \end{pmatrix}$$

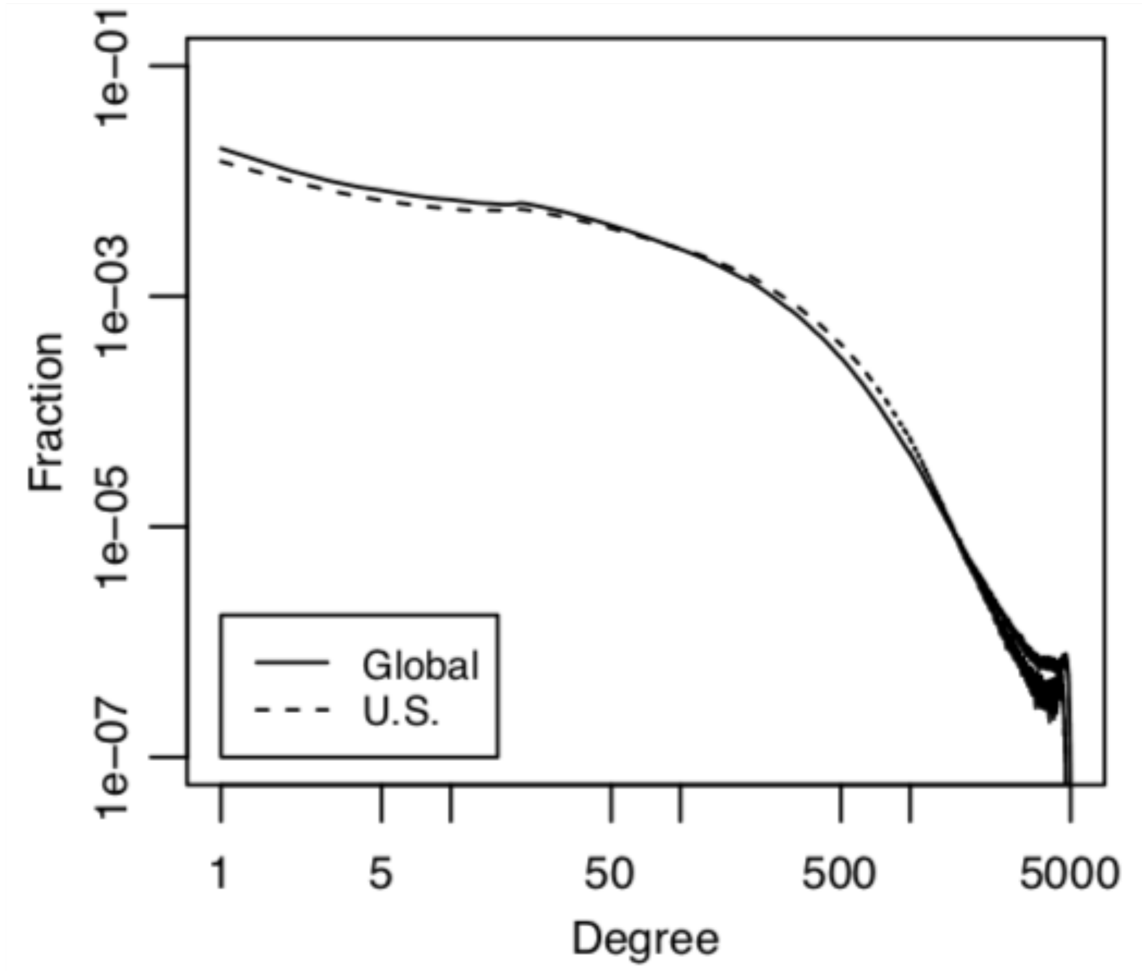
EXEMPLE OF GRAPH ANALYSIS

- Source: [The Anatomy of the Facebook Social Graph, Ugander et al. 2011]
- The Facebook friendship network in 2011

EXEMPLE OF GRAPH ANALYSIS

- 721M users (nodes) (active in the last 28 days)
- 68B edges
- Average degree: 190 (average # friends)
- Median degree: 99
- Connected component: 99.91%

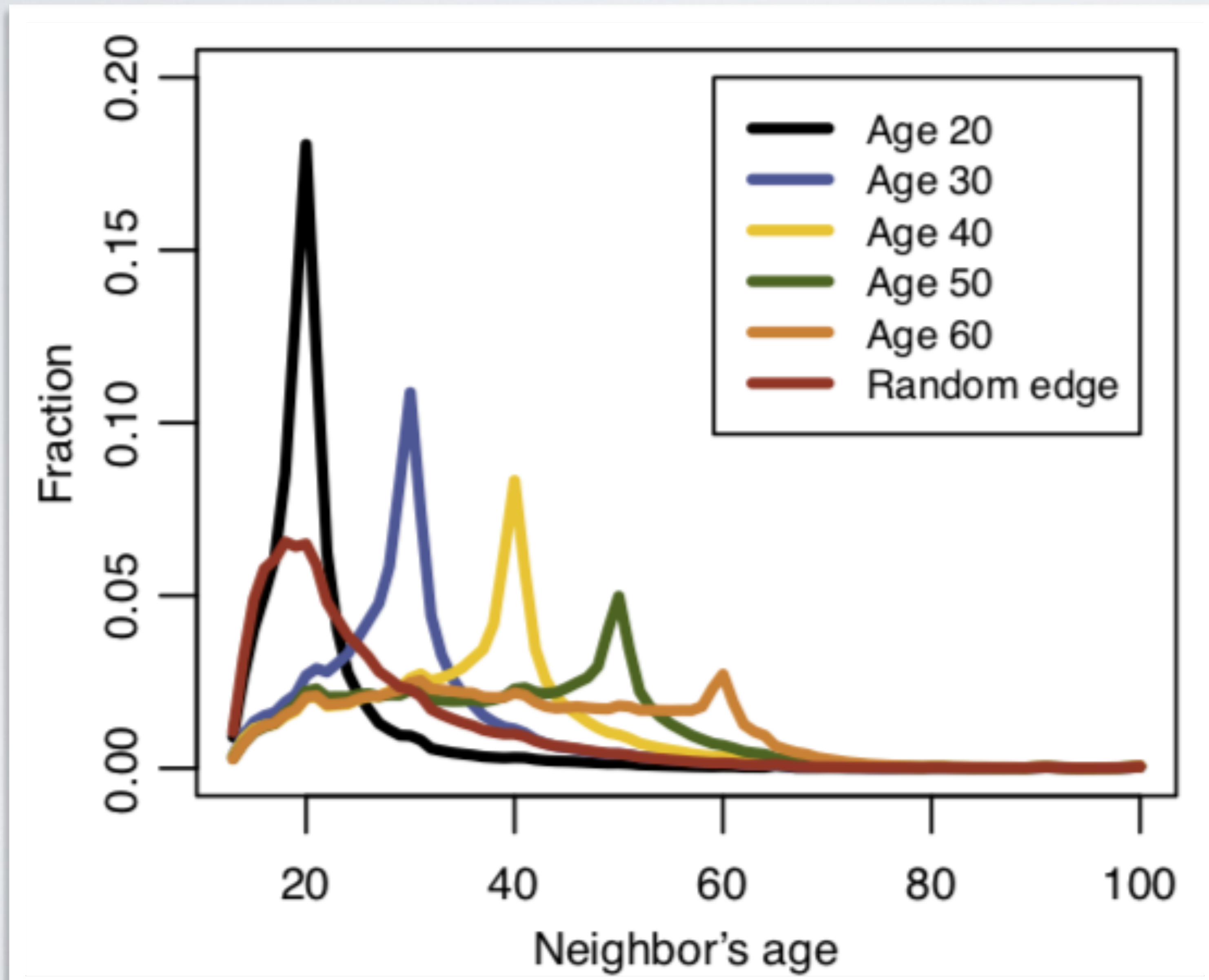
EXEMPLE OF GRAPH ANALYSIS



(a)

Degree distribution

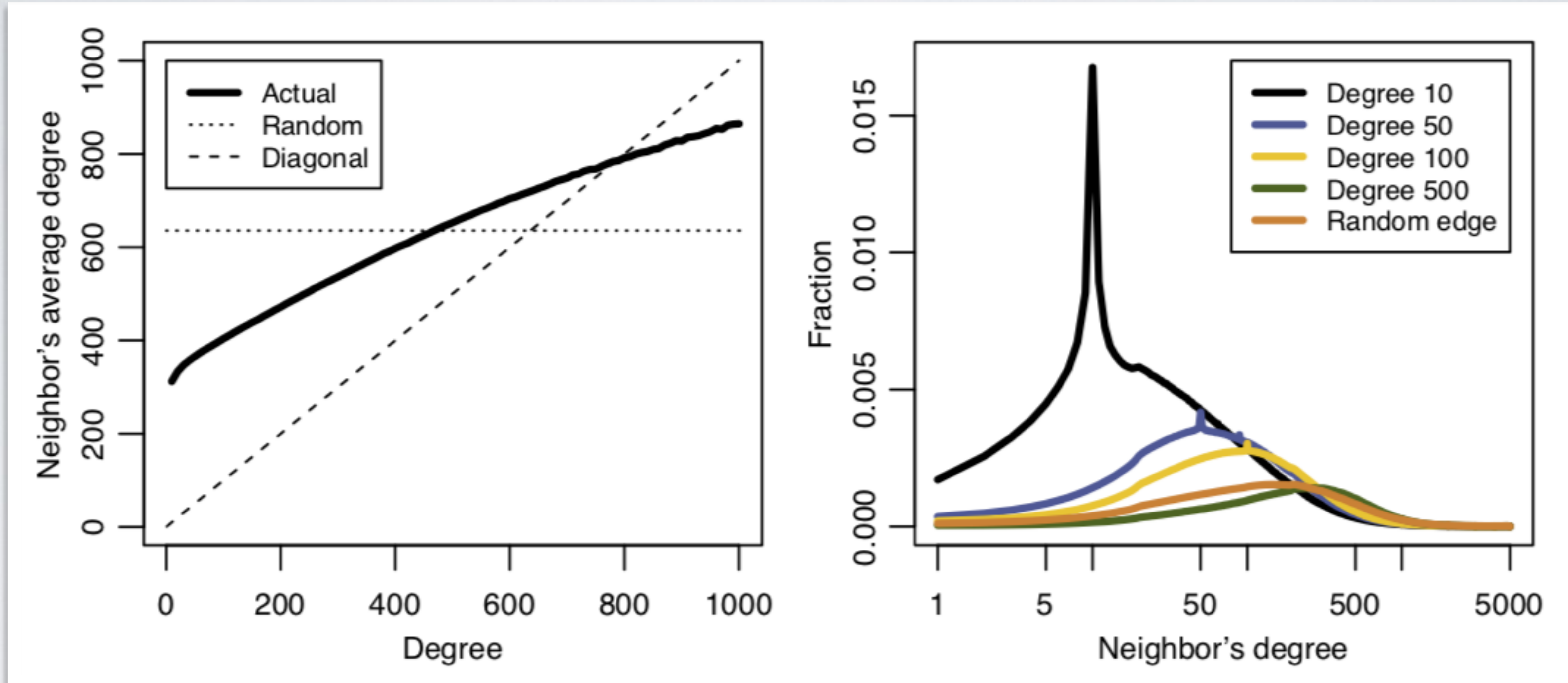
EXEMPLE OF GRAPH ANALYSIS



Age homophily

(More next class)

EXEMPLE OF GRAPH ANALYSIS



My friends have more Friends than me!

Many of my friends have the Same # of friends than me!

CENTRALITIES

Characterizing/Discovering important nodes

CENTRALITY

- We can measure nodes importance using so-called **centrality**.
- Poor terminology: nothing to do with being central in general
- Usage:
  - Some centralities have straightforward interpretation
  - Centralities can be used as *node features* for machine learning on graph
    - (Classification, link prediction, …)

NODE DEGREE

- **Degree**: how many neighbors

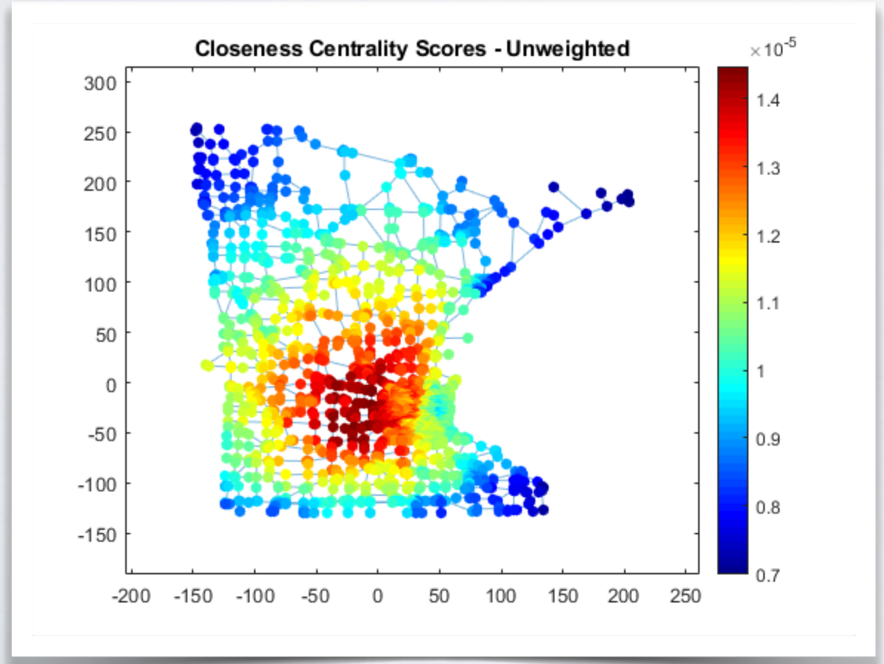
- Often enough to find important nodes
  - Main characters of a series talk with the more people
  - Largest airports have the most connections
  - …

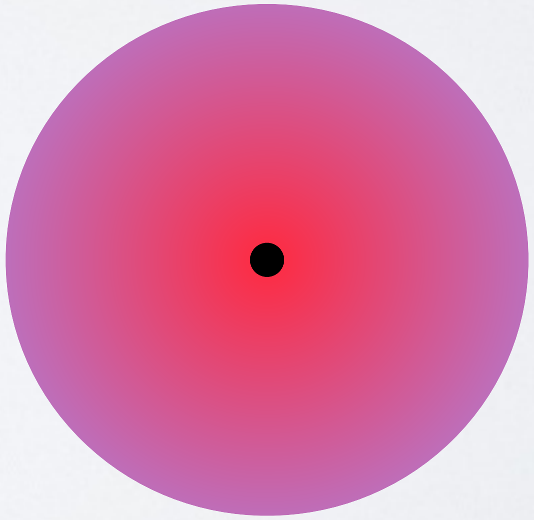
- But not always
  - Facebook users with the most friends are spam
  - Webpages/wikipedia pages with most links are simple lists of references
  - …

FARNESS, CLOSENESS HARMONIC CENTRALITY

FARNESS, CLOSENESS

- How close the node is to all other nodes
- Parallel with the center of a figure:
  - Center of a circle is the point of shorter average distance to any points in the circle





FARNESS, CLOSENESS

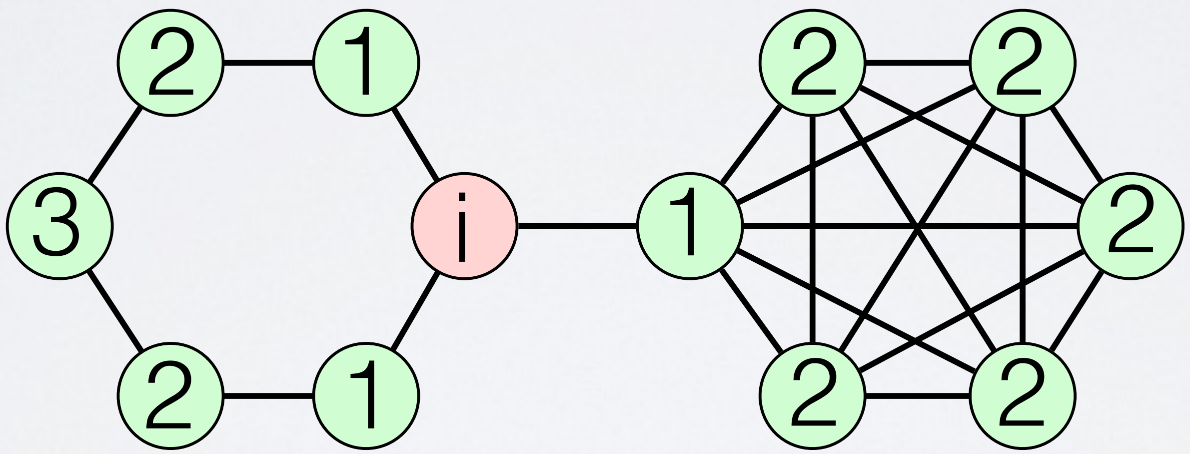
Farness: Average distance to all other nodes in the graph

$$\text{Farness}(u) = \frac{1}{N-1} \sum_{v \in V \setminus u} \ell_{u,v}$$

CLOSENESS CENTRALITY

Closeness: Inverse of the farness, i.e., how close the node is to all other nodes in term of shortest paths.

$$\text{Closeness}(u) = \frac{N-1}{\sum_{v \in V \setminus u} \ell_{u,v}}$$



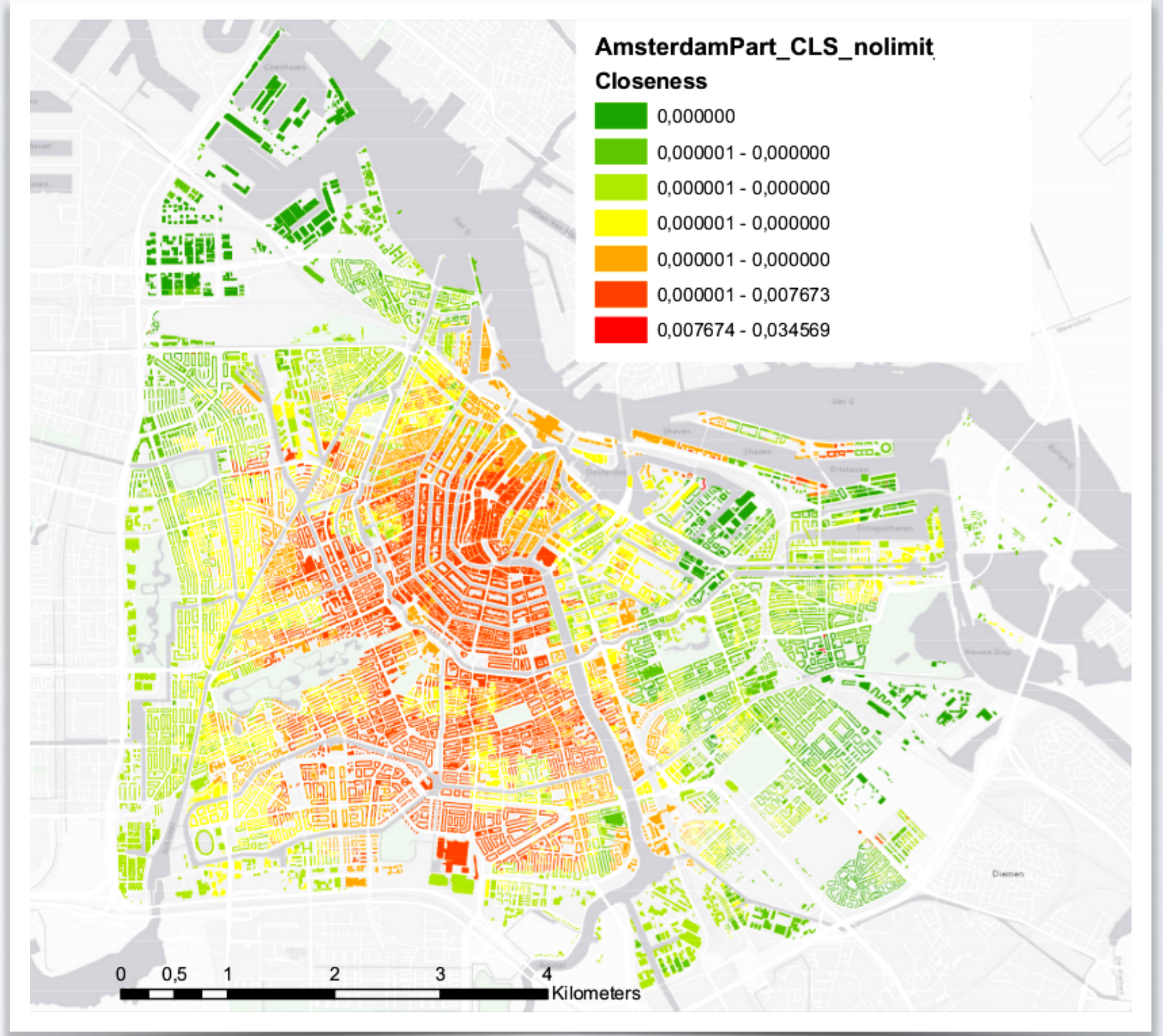
$$C_{cl}(i) = \frac{12-1}{(3 \times 1 + 7 \times 2 + 1 \times 3)} = \frac{11}{20} = 0.55$$

CLOSENESS CENTRALITY

Closeness: Inverse of the farness, i.e., how close the node is to all other nodes in term of shortest paths.

$$\text{Closeness}(u) = \frac{N-1}{\sum_{v \in V \setminus u} \ell_{u,v}}$$

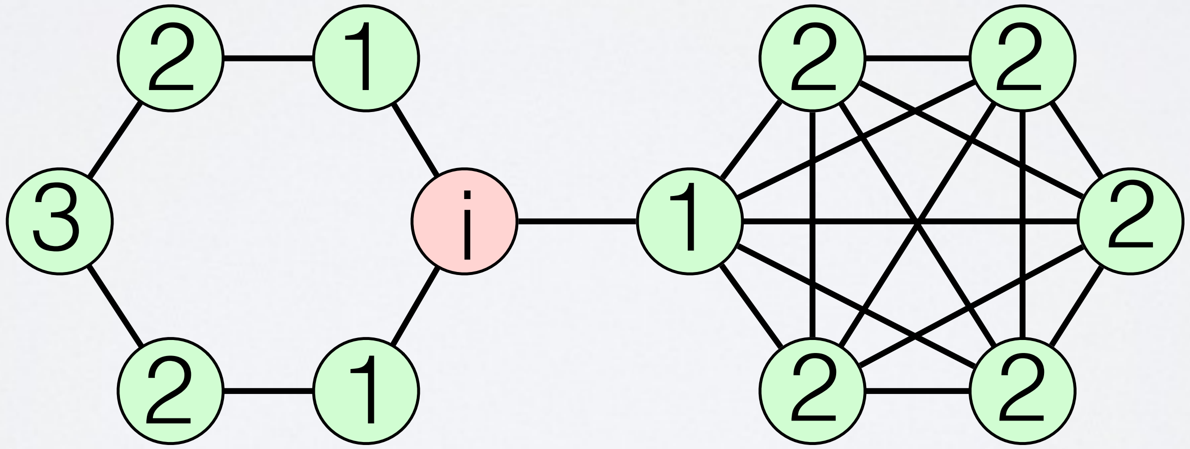
1=all nodes are at distance one



Harmonic Centrality

Harmonic centrality: A variant of the closeness defined as the average of the inverse of distance to all other nodes (Harmonic mean). Well defined on disconnected network with $\frac{1}{\infty} = 0$. Its interpretation is the same as the closeness.

$$\text{Harmonic}(u) = \frac{1}{N-1} \sum_{v \in V \setminus u} \frac{1}{\ell_{u,v}}$$



$$C_h(i) = \frac{1}{12-1}\left(3 \times \frac{1}{1} + 7 \times \frac{1}{2} + 1 \times \frac{1}{3}\right) = \frac{41}{66} = 0.6212$$

BETWEENNESS CENTRALITY

- Measure how much the node plays the role of a bridge
- Betweenness of *u: fraction* of all the shortest paths between all the pairs of nodes going through u.

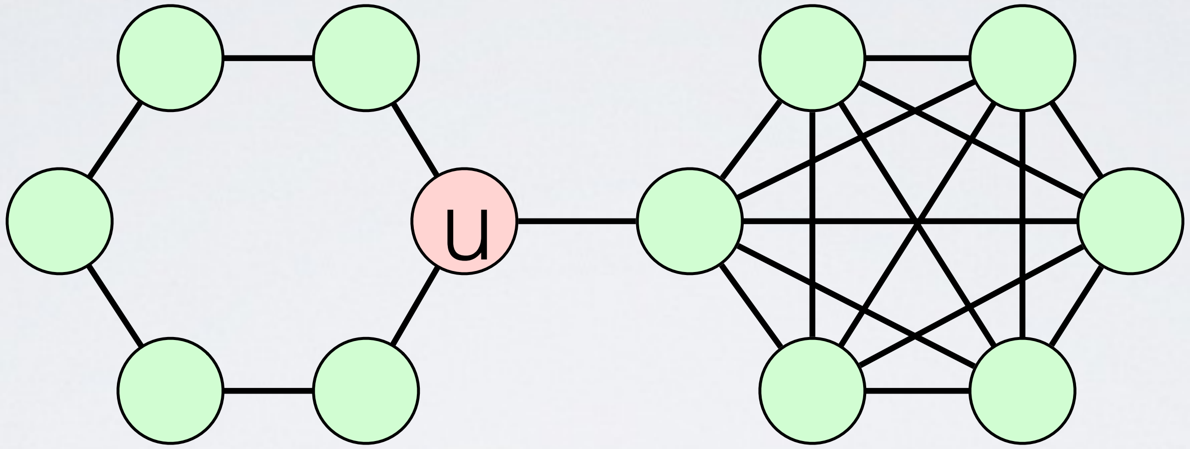
$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

with σ_{st} the number of shortest paths between nodes s and t and $\sigma_{st}(v)$ the number of those paths passing through v.
The betweenness tends to grow with the network size. A normalized version can be obtained by dividing by the number of pairs of nodes, i.e., for a directed graph: $C_B^{\text{norm}}(v) = \frac{C_B(v)}{(N-1)(N-2)}$.

Betweenness Centrality

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

directed graph: $C_B^{\text{norm}}(v) = \frac{C_B(v)}{(N-1)(N-2)}$.



$$C_B(u) = 2\frac{5*6+1+\frac{1}{2}+\frac{1}{2}}{11*10} = \frac{64}{110}$$

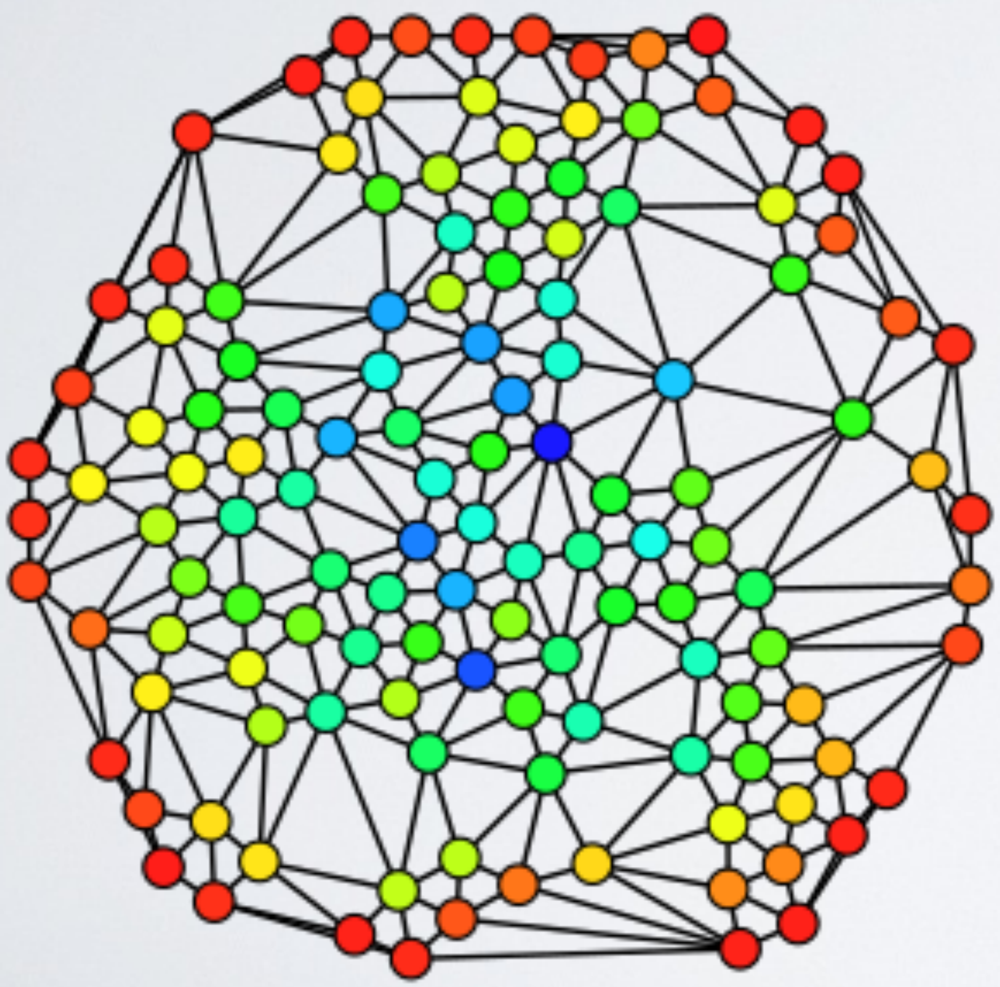
Exact computation:

Floyd-Warshall: *$O(n^3)$ time complexity*
$O(n^2)$ space complexity

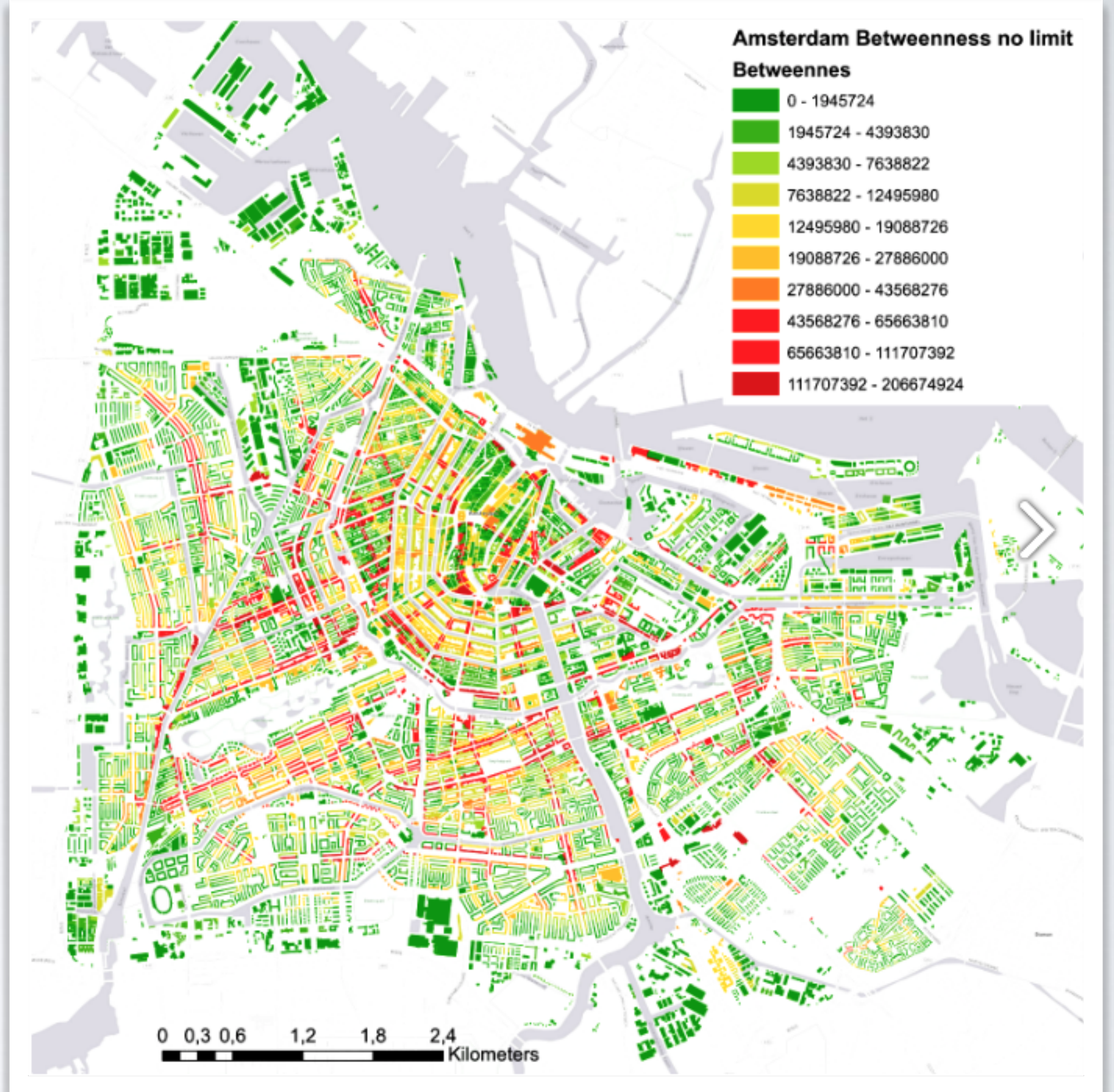
Approximate computation

Dijskstra: *$O(n(m+n \log n))$ time complexity*

BETWEENNESS CENTRALITY



(blue higher)



(red higher)

EDGE - BETWEENNESS

Same definition as for nodes

Can you guess the edge of highest betweenness in the European rail network ?

RECURSIVE DEFINITIONS

RECURSIVE DEFINITIONS

- Recursive importance:
  - **Important nodes** are those connected ***to important nodes***
- Several centralities based on this idea:
  - Eigenvector centrality
  - PageRank
  - …

RECURSIVE DEFINITION

- We would like scores such as :
  - Each node has a score (centrality),
  - If every node "sends" its score to its neighbors, the sum of all scores received by each node will be equal to its original score

$$C_u^{t+1} = \frac{1}{\lambda} \sum_{v \in N_u^{in}} C_v^t \tag{1}$$

- With λ a normalisation constant

RECURSIVE DEFINITION

- This problem can be solved by what is called the *power method:*
  - 1) We initialize all scores to random values
  - 2)Each score is updated according to the desired rule, until reaching a stable point (after normalization)
- Why does it converge?
  - Perron-Frobenius theorem (see next slide)
  - =>True for undirected graphs with a single connected component

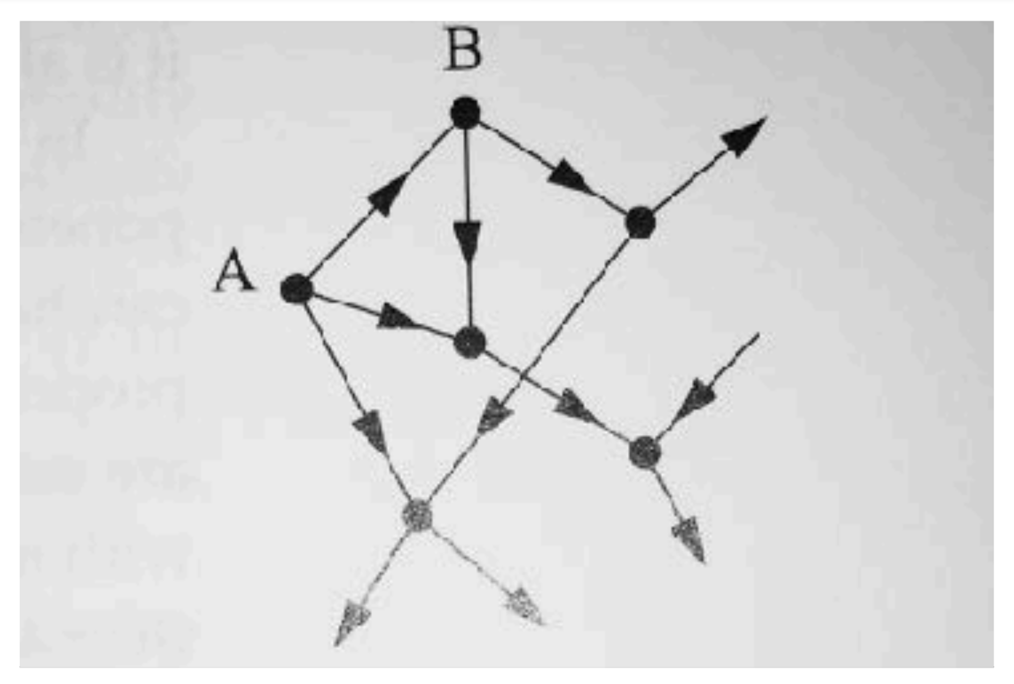
EIGENVECTOR CENTRALITY

- What we just described is called the Eigenvector centrality
- A couple eigenvector (x) and eigenvalue (λ) is defined by the following relation: $Ax = \lambda x$
  - x is a column vector of size *n*, which can be interpreted as the scores of nodes
- What Perron-Frobenius algorithm says is that the power method will always converge to the *leading eigenvector*, i.e., the eigenvector associated with the highest eigenvalue

Eigenvector Centrality

Some problems in case of directed network:

- Adjacency matrix is asymmetric
- 2 sets of eigenvectors (Left & Right)
- 2 leading eigenvectors
  - Use right eigenvectors : consider nodes that are pointing towards you



But problem with source nodes (0 in-degree)

-Vertex A is connected but has only outgoing link = Its centrality will be 0

-Vertex B has outgoing and an incoming link, but incoming link comes from A = Its centrality will be 0

-etc.

Solution: Only in strongly connected component

Note: Acyclic networks (citation network) do not have strongly connected component

PageRank Centrality

- Eigenvector centrality generalised for directed networks

PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Brin, S. and Page, L. (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

PageRank Centrality

- Eigenvector centrality generalised for directed networks

PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Brin, S. and Page, L. (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/

PageRank Centrality

(Side notes)

-"We chose our system name, Google, because it
is a common spelling of googol, or 10^{100} and fits well with our goal of building very large-scale search "

-"[…] at the same time, search engines have migrated from the academic domain to the commercial. **Up until now most search engine development has gone on at companies with little publication of technical details. This causes search engine technology to remain largely a black art and to be advertising oriented (see Appendix A). With Google, we have a strong goal to push more development and understanding into the academic realm."**

-"[...], we expect that advertising funded search engines will be inherently biased towards the advertisers and away from the needs of the consumers."

PageRank Centrality

(Side notes)

Sergey Brin received his B.S. degree in mathematics and computer science from the University of Maryland at College Park in 1993. Currently, he is a Ph.D. candidate in computer science at Stanford University where he received his M.S. in 1995. He is a recipient of a National Science Foundation Graduate Fellowship. His research interests include search engines, information extraction from unstructured sources, and data mining of large text collections and scientific data.

Lawrence Page was born in East Lansing, Michigan, and received a B.S.E. in Computer Engineering at the University of Michigan Ann Arbor in 1995. He is currently a Ph.D. candidate in Computer Science at Stanford University. Some of his research interests include the link structure of the web, human computer interaction, search engines, scalability of information access interfaces, and personal data mining.

PAGERANK

- 2 main improvements over eigenvector centrality:
  - In directed networks, problem of source nodes
    - => Add a constant centrality gain for every node
  - Nodes with very high centralities give very high centralities to all their neighbors (even if that is their only in-coming link)
    - => What each node "is worth" is divided equally among its neighbors (normalization by the degree)

$$C_u^{t+1} = \frac{1}{\lambda} \sum_{v \in N_u^{in}} C_v^t$$

=>

$$C_u^{t+1} = \alpha \sum_{v \in N_u^{in}} \frac{C_v^t}{k_v^{out}} + \beta$$

With by convention β=1 and α a parameter (usually 0.85) controlling the relative importance of β

PAGERANK

Matrix interpretation

Principal eigenvector of the "Google Matrix":
First, define matrix S as:
- -Normalization by columns of A
- -Columns with only 0 receives 1/n

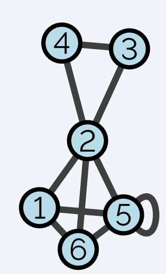
-Finally, $G_{ij} = \alpha S_{ij} + (1-\alpha)/n$

(a)
$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

(c)
$$S = \begin{pmatrix} 0 & 1/2 & 1/3 & 0 & 1/5 \\ 1 & 0 & 1/3 & 1/3 & 1/5 \\ 0 & 1/2 & 0 & 1/3 & 1/5 \\ 0 & 0 & 1/3 & 0 & 1/5 \\ 0 & 0 & 0 & 1/3 & 1/5 \end{pmatrix}$$

(e)
$$G = \begin{pmatrix} 0.03 & 0.455 & 0.313 & 0.03 & 0.2 \\ 0.88 & 0.03 & 0.313 & 0.313 & 0.2 \\ 0.03 & 0.455 & 0.03 & 0.313 & 0.2 \\ 0.03 & 0.03 & 0.313 & 0.03 & 0.2 \\ 0.03 & 0.03 & 0.03 & 0.313 & 0.2 \end{pmatrix}$$

Graph



A - Adjacency Mat.

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

Random W. mat.

$$\begin{pmatrix} 0 & \frac{1}{5} & 0 & 0 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ 0 & \frac{1}{5} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{5} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{5} & 0 & 0 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{5} & 0 & 0 & \frac{1}{4} & 0 \end{pmatrix}$$

PageRank - as Random Walk

Main idea: The PageRank computation can be interpreted as a Random Walk process with restart

Teleportation probability: the parameter α gives the probability that in the next step of the RW will follow a Markov process or with probability $1\text{-}\alpha$ it will jump to a random node

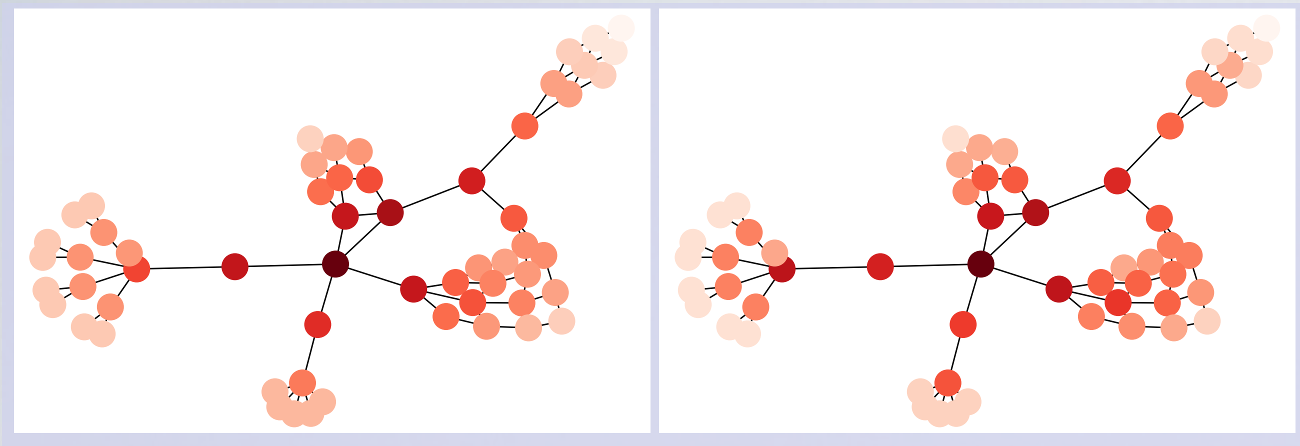
Pagerank score of a node thus corresponds to the probability of this random walker to be on this node after an infinite number of hops.

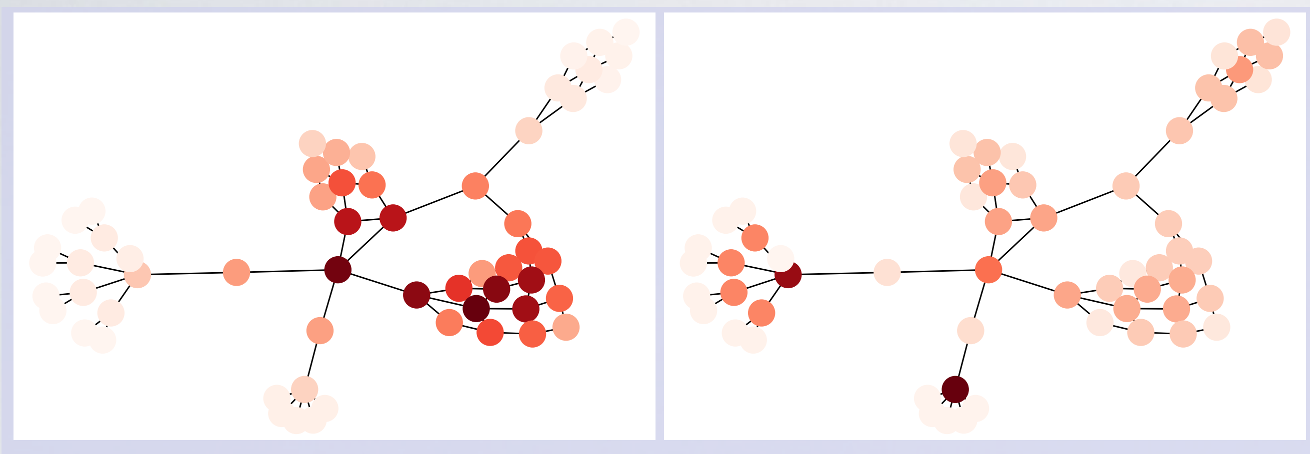
PAGERANK

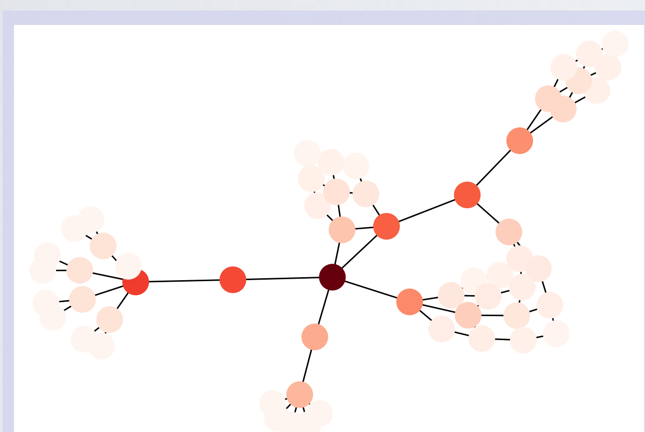
- Then how do Google rank when we do a research?
- Compute pagerank (using the power method for scalability)
- Create a subgraph of documents related to our topic
- Of course now it is certainly much more complex, but we don't really know:
  "Most search engine development has gone on at companies with little publication of technical details. This causes search engine technology to remain largely a black art" [Page, Brin, 1997]

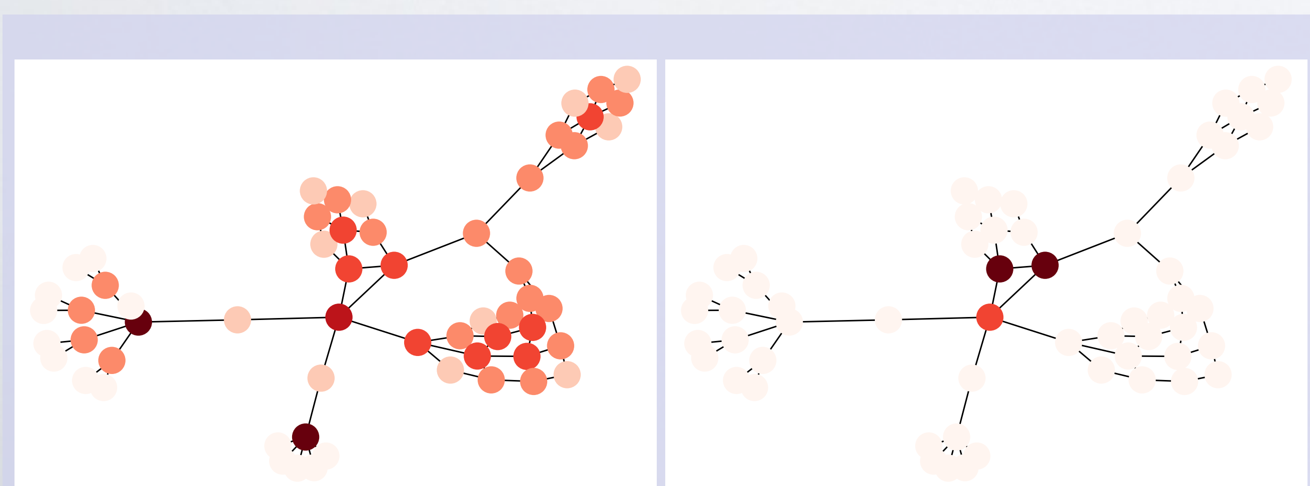
OTHERS

- Many other centralities can be found in the literature
  - Katz Centrality (Generalization of eigenvector centrality adding penalization with distance)
  - Hub and Authority Scores (HITS)
  - Random Walk Centrality (probability to pass on a node, linked to PageRank and betweenness)
  - Communicability Centrality => Betweeness but for a given length of walks
  - etc.





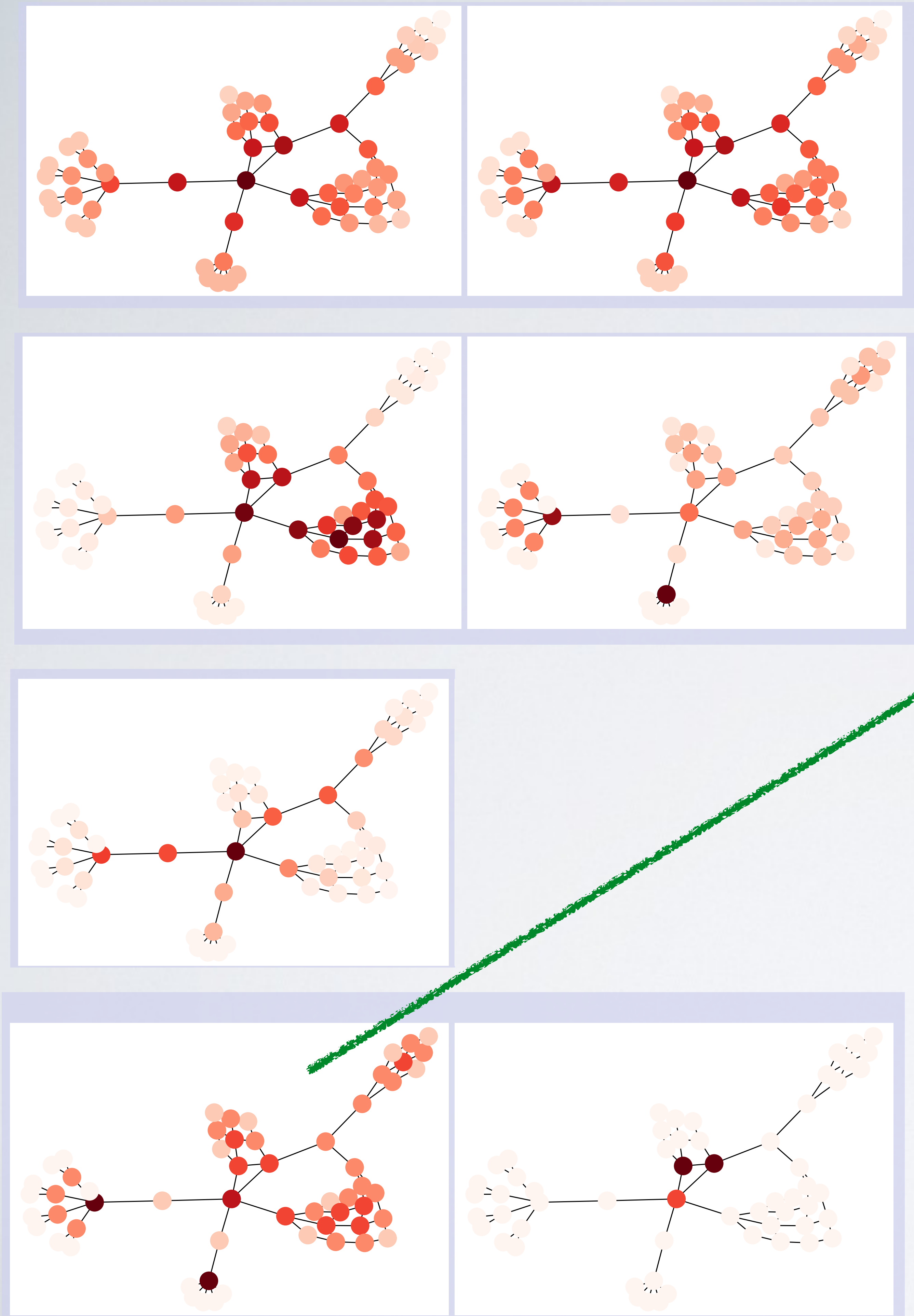




Which is which ?

Degree
Clustering coefficient
Closeness
Harmonic Centrality
Betweenness

Eigenvector
PageRank

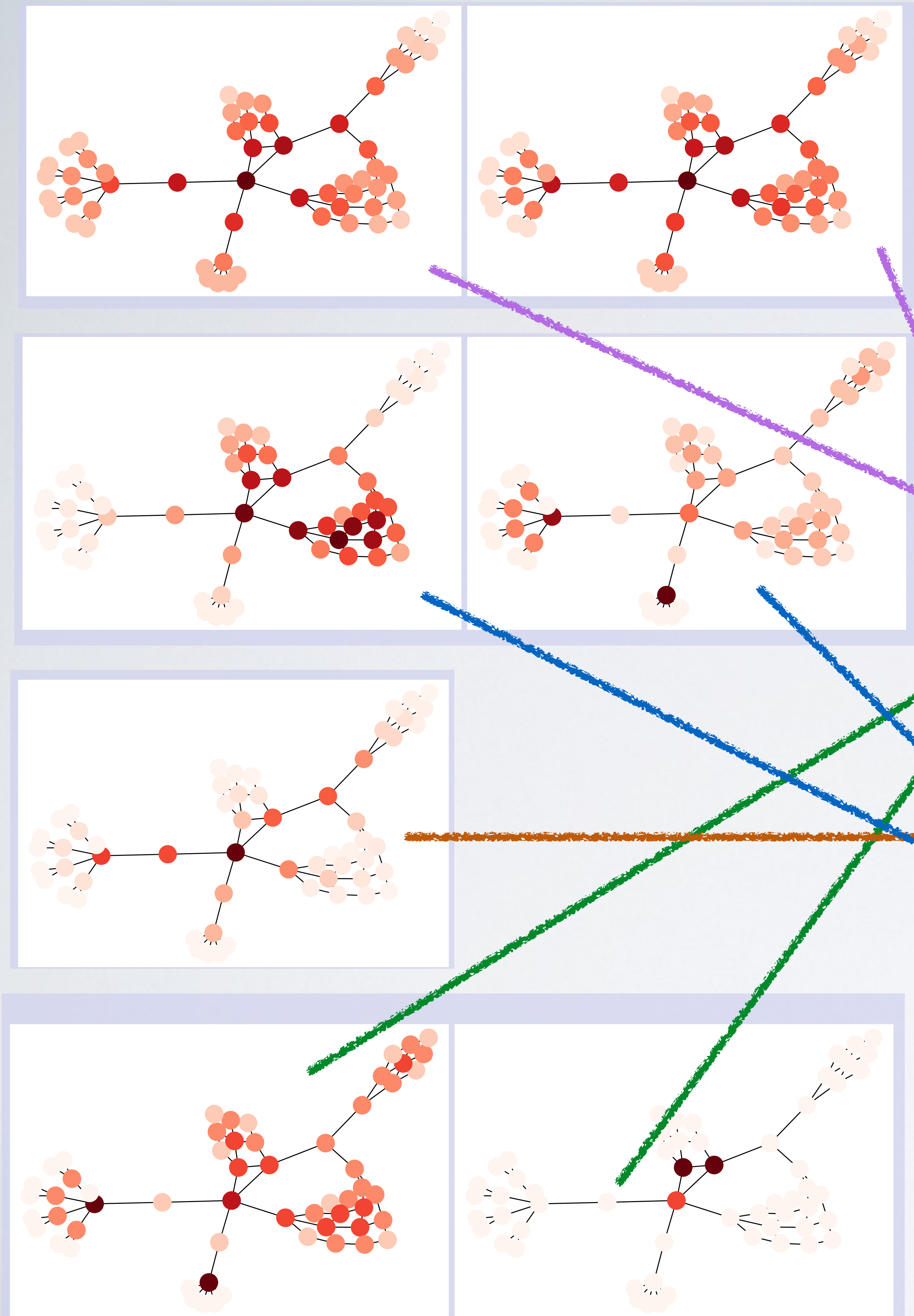


Which is which ?

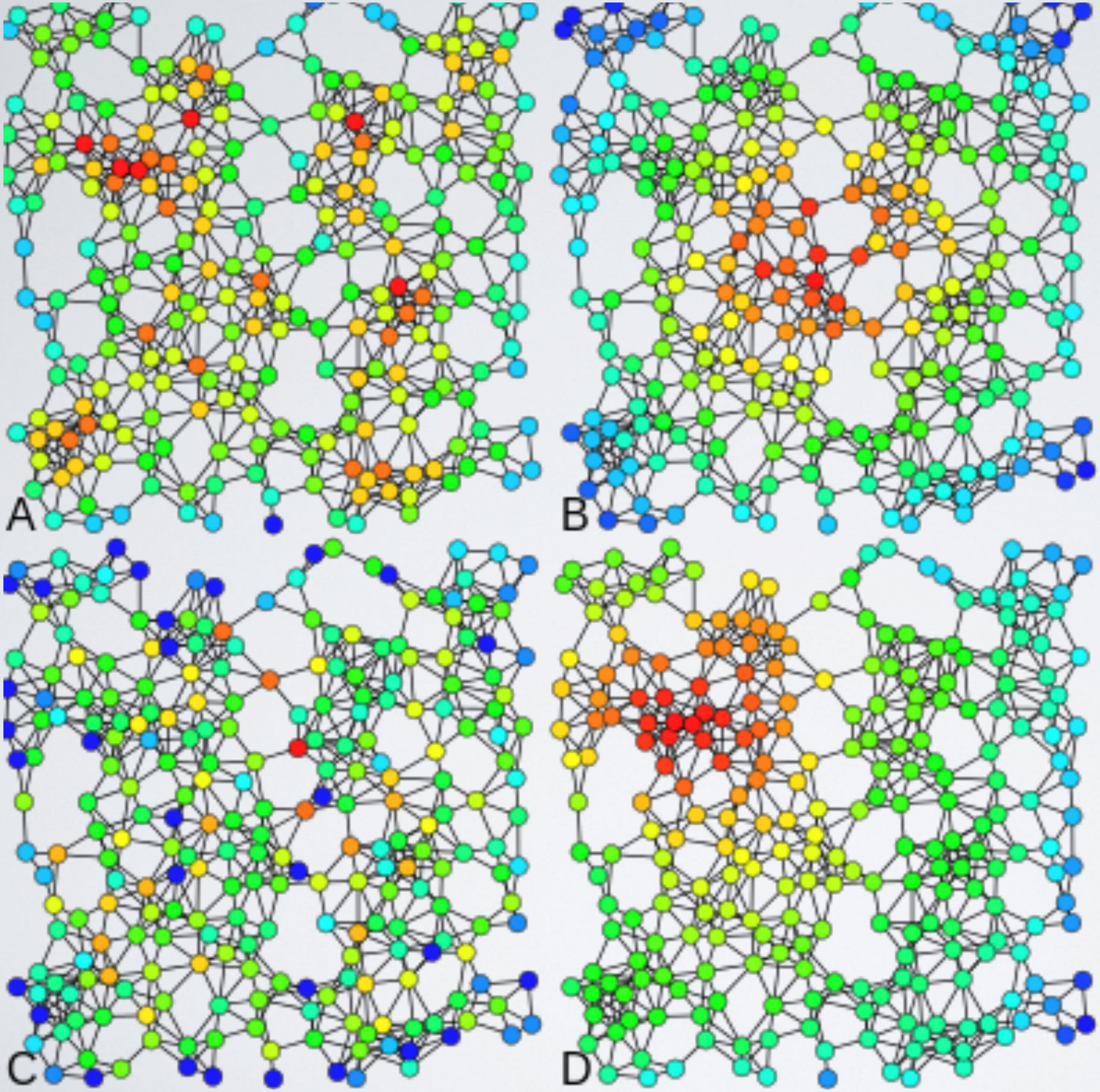
Degree
Clustering coefficient
Closeness
Harmonic Centrality
Betweenness

Eigenvector
PageRank

Which is which ?

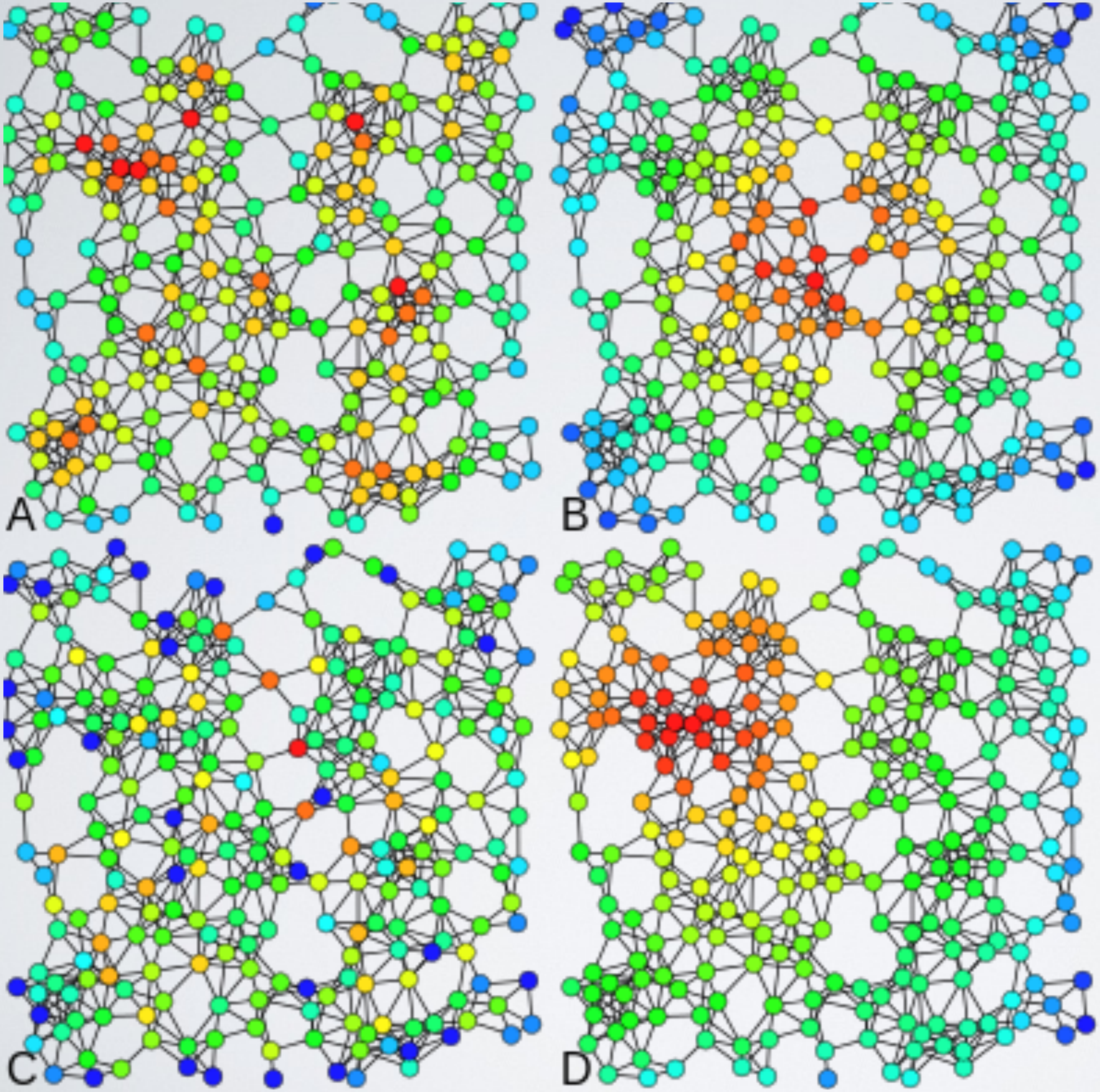


Degree
Clustering coefficient
Closeness
Harmonic Centrality
Betweenness
Eigenvector
PageRank



Try again :)

Degree
Betweenness
Closeness
Eigenvector



Try again :)

A: Degree
B:Closeness
C: Betweenness
D: Eigenvector