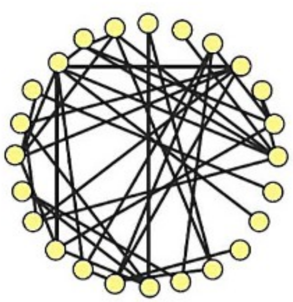
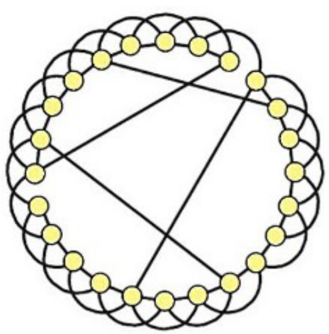
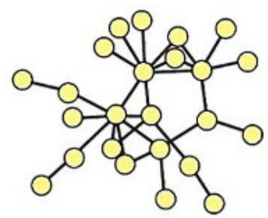
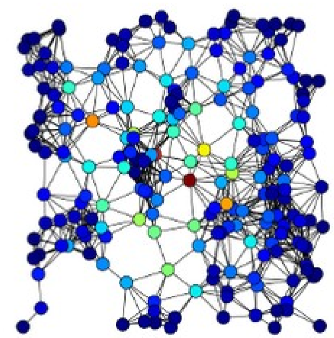
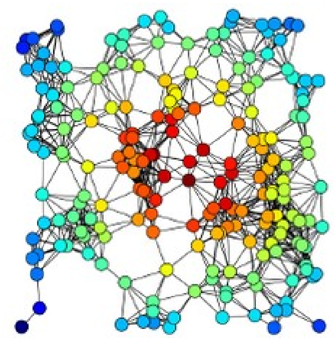
Introduction to the Analysis and Visualisation of Complex Networks

U.PORTO
FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Pedro Ribeiro
(DCC/FCUP & CRACS/INESC-TEC)

CRACS INESCTEC ASSOCIATE LABORATORY PORTUGAL

(this part includes some slides heavily based on material from Jure Leskovec and Lada Adamic @ Stanford University)

CLAD
Associação Portuguesa de Classificação e Análise de Dados

Network Properties:
how to measure a network?

Plan: Key Network Properties

- (1) Degree distribution **$P(k)$**
- (2) Path Length **h**
- (3) Clustering coefficient **C**
- (4) Connected components **s**

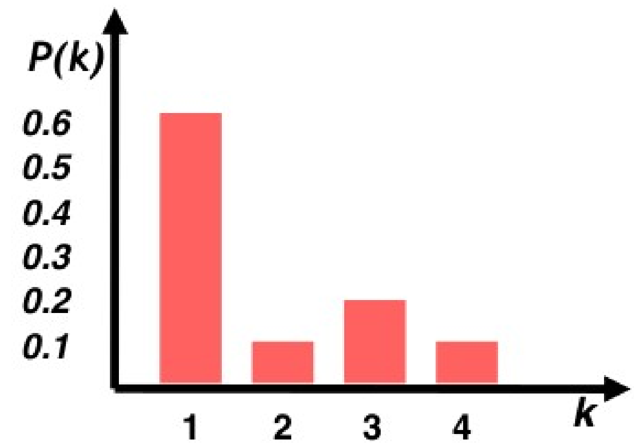
(1) Degree Distribution

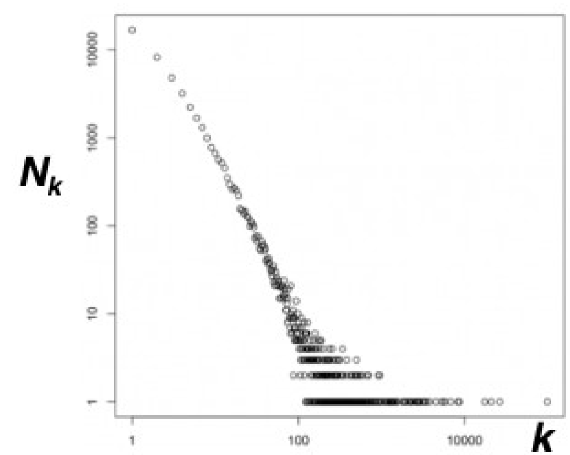
- Degree distribution ***P(k)***: probability that a randomly chosen node has degree k

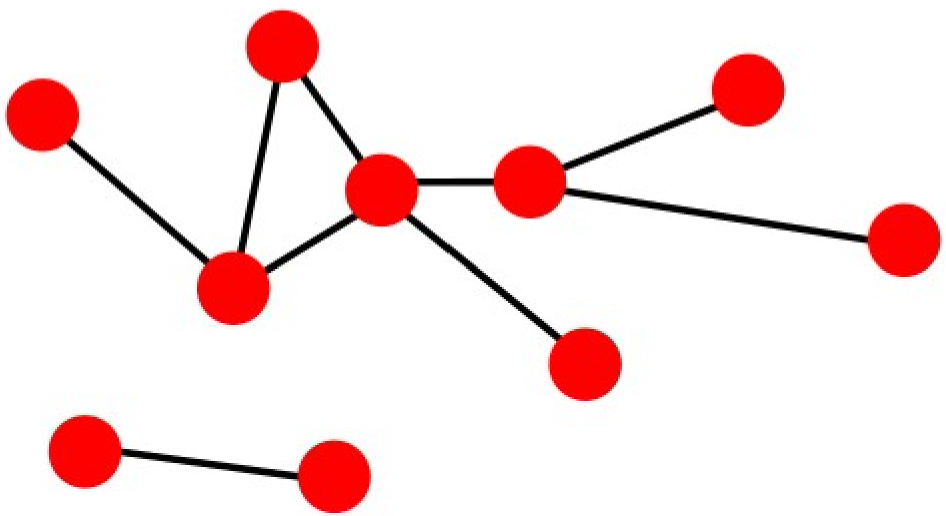
  N_k = # nodes with degree k

- Normalized histogram:

  $P(k) = N_k / N$ → plot







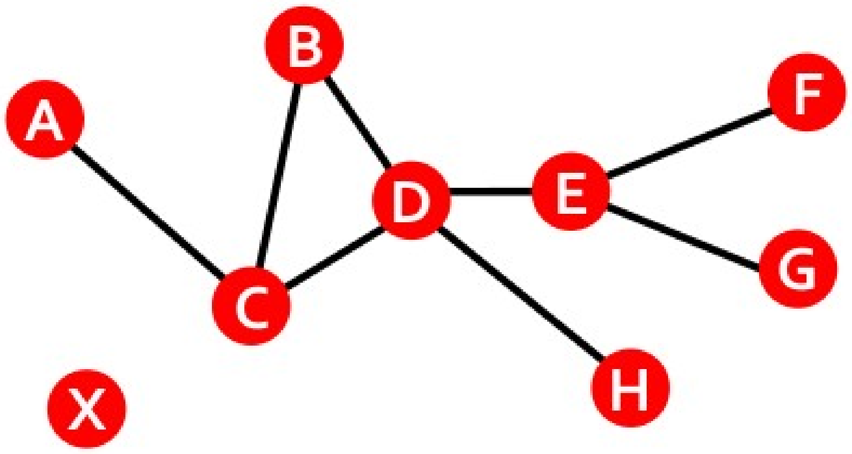
(2) Paths in a Graph

- A ***path*** is a sequence of nodes in which each node is linked to the next one

$$P_n = \{i_0, i_1, i_2, \dots, i_n\} \quad or$$

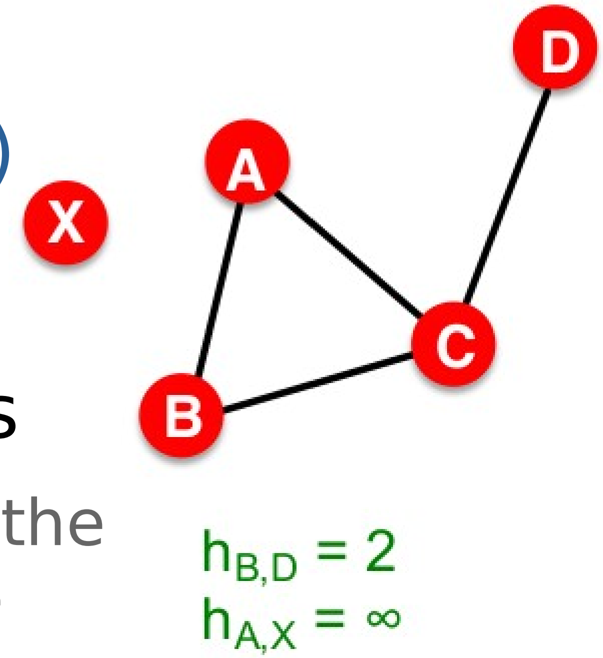
$$P_n = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n),\}$$

- A path can intersect itself and pass trough the same edge multiple times
  - E.g. ACBDCDEG
  - In a directed graph, a path can only follow the direction of the “arrow”

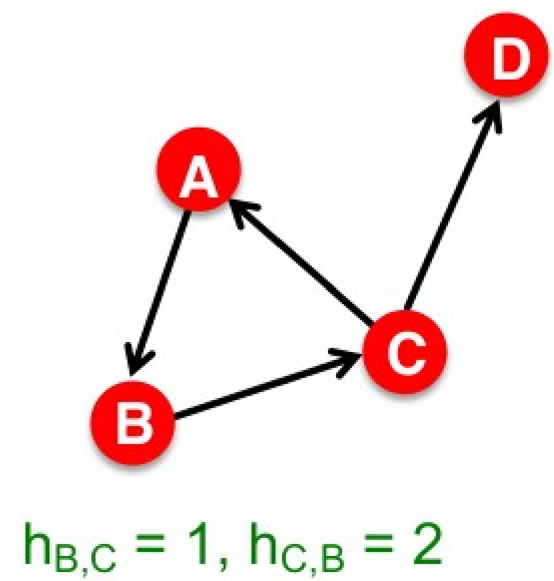


Distance in a Graph

- **Distance** (shortest path, geodesic) between a pair of nodes is defined as the number of edges along the shortest path connecting the nodes
  - If the two nodes are **not connected**, the distance is usually defined as **infinite**



- In **directed graphs** paths need to follow the direction of the arrows
  - Consequence: distance is **not symmetric**: $h_{B,C} \neq h_{C,B}$



Network Diameter

- **Diameter:** The maximum (shortest path) distance between any pair of nodes in a graph

- **Average path length** for a connected graph (component) or a strongly connected (component of a) directed graph

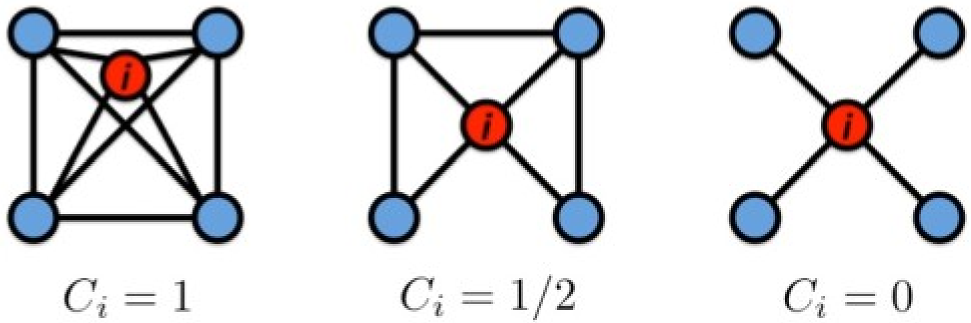
$$\bar{h} = \frac{1}{2E_{\max}} \sum_{i,j \neq i} h_{ij}$$

Where h_{ij} is the distance from node i to node j
E_{max} is max number of edges (total number of node pairs) = $n(n-1)/2$

  – Many times we compute the average only over the connected pairs of nodes (that is, we ignore “infinite”length paths)

(3) Clustering Coefficient

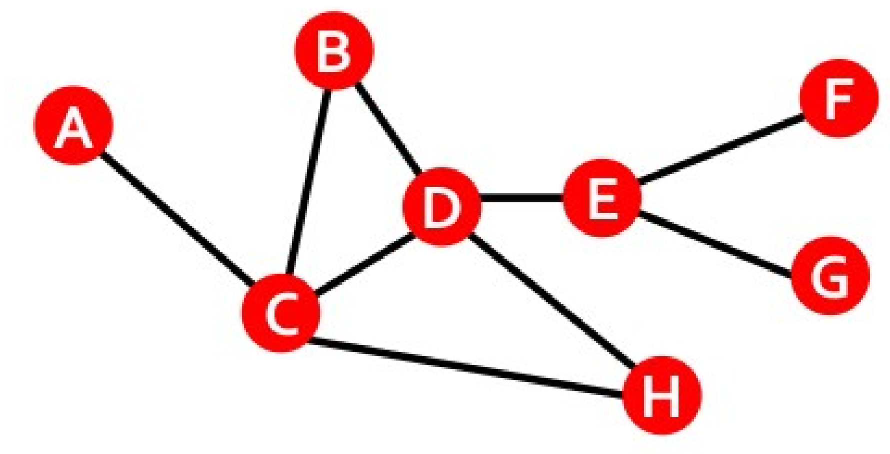
- **Clustering coefficient**:
  - What portion of ***i***'s neighbors are connected?
  - Node ***i*** with degree $\boldsymbol{k_i}$
  - $C_i \in [0,1]$
  - $C_i = \dfrac{2e_i}{k_i(k_i - 1)}$ where e_i is the number of edges between the neighbors of node *i*



- **Average clustering coefficient:** $C = \dfrac{1}{N} \sum_{i}^{n} C_i$

Clustering Coefficient

- **Clustering coefficient**:
  - What portion of ***i***'s neighbors are connected?
  - Node ***i*** with degree $\boldsymbol{k_i}$
  - $C_i = \frac{2e_i}{k_i(k_i - 1)}$ where e_i is the number of edges between the neighbors of node *i*



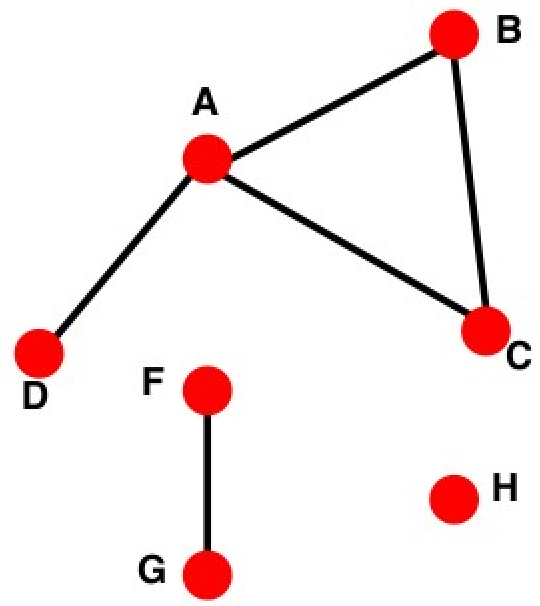
$k_B=2, \quad e_B=1, \quad C_B = 2/2 = 1$

$k_D=4, \quad e_D=2, \quad C_D = 4/12 = 1/3$

Avg. Clustering: C = 0.33

(4) Connectivity

- Size of the largest connected component
  - Largest set where any two vertices can be joined by a path
- **Largest component = Giant component**



How to find connected components:

- Start from random node and perform Breadth First Search (BFS)
- Label the nodes BFS visited
- If all nodes are visited, the network is connected
- Otherwise find an unvisited node and repeat BFS

Summary: Key Network Properties

- (1) Degree distribution **$P(k)$**
- (2) Path Length **h**
- (3) Clustering coefficient **C**
- (4) Connected components **s**

Measuring these properties in a Real World Graph

MSN Messenger

- **MSN Messenger**
  - 1 month activity
    - 245 million users logged in
    - 180 million users engaged in conversations
    - More than 30 billion conversations
    - More than 255 billion exchanged messages

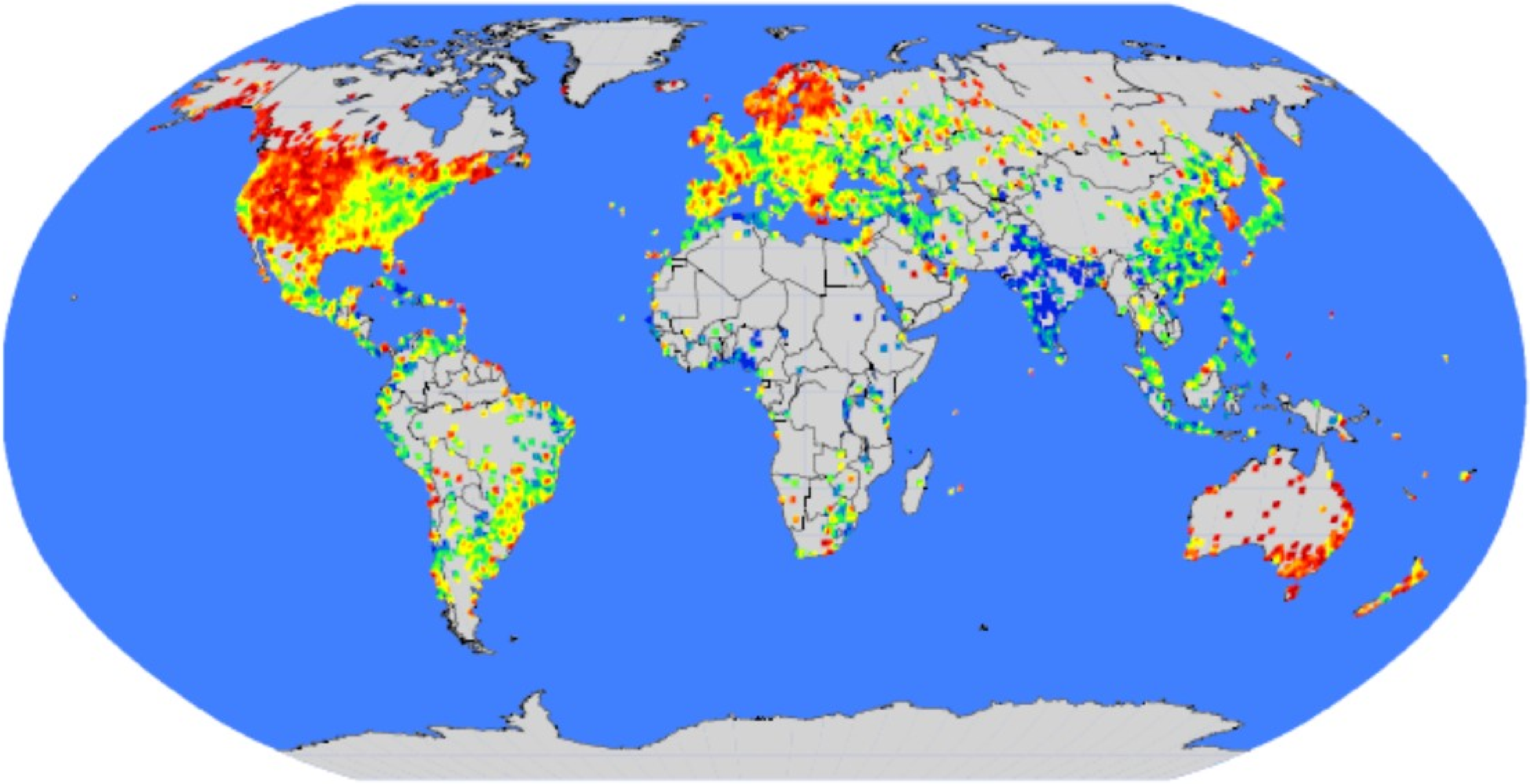
Planetary-Scale Views on a Large Instant-Messaging Network

WWW 2008

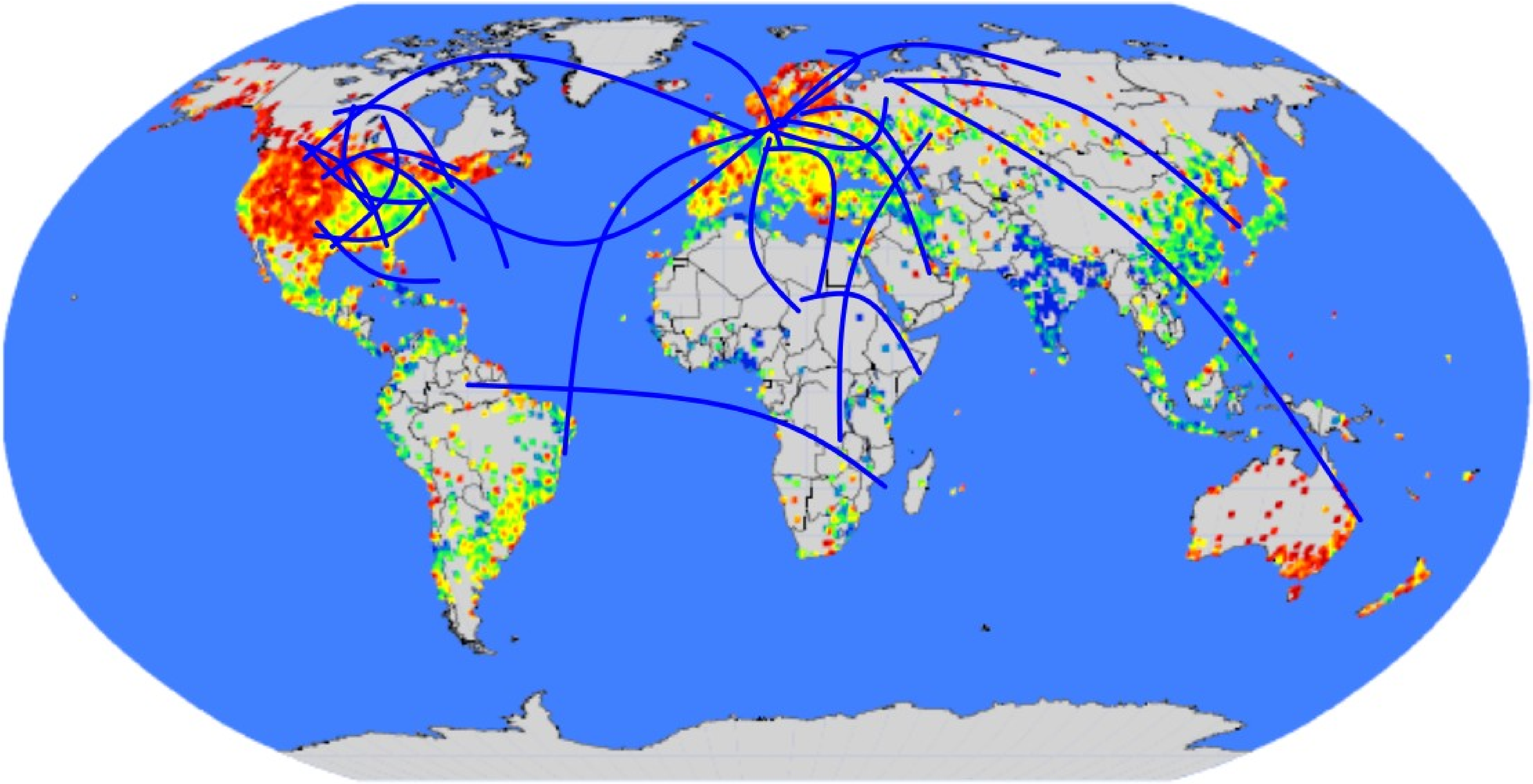
Jure Leskovec*
Carnegie Mellon University
jure@cs.cmu.edu

Eric Horvitz
Microsoft Research
horvitz@microsoft.com

Spatial Network: Geography

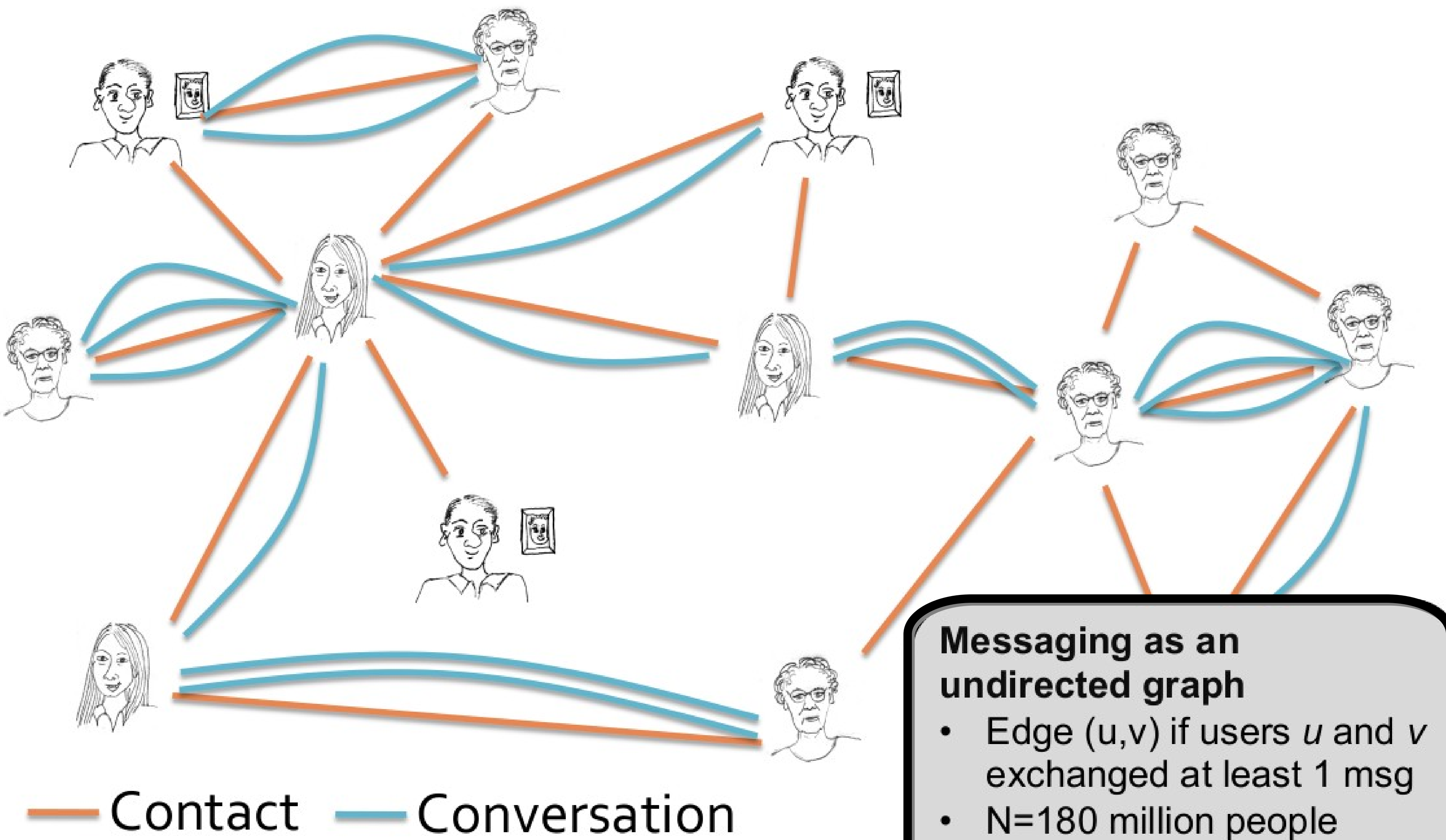


Communication → Connections



Network: 180M people, 1.3B edges

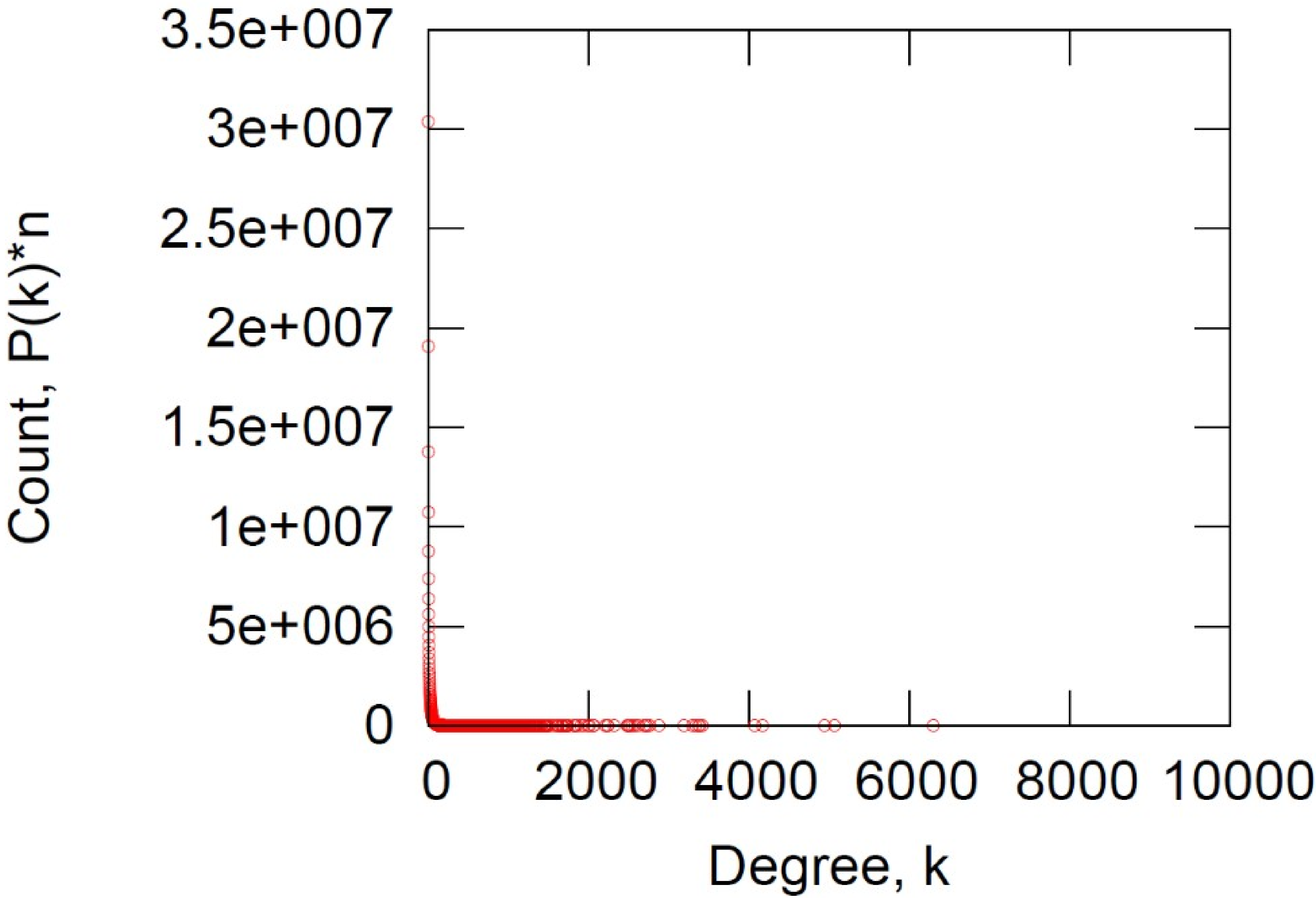
Messaging as multigraph



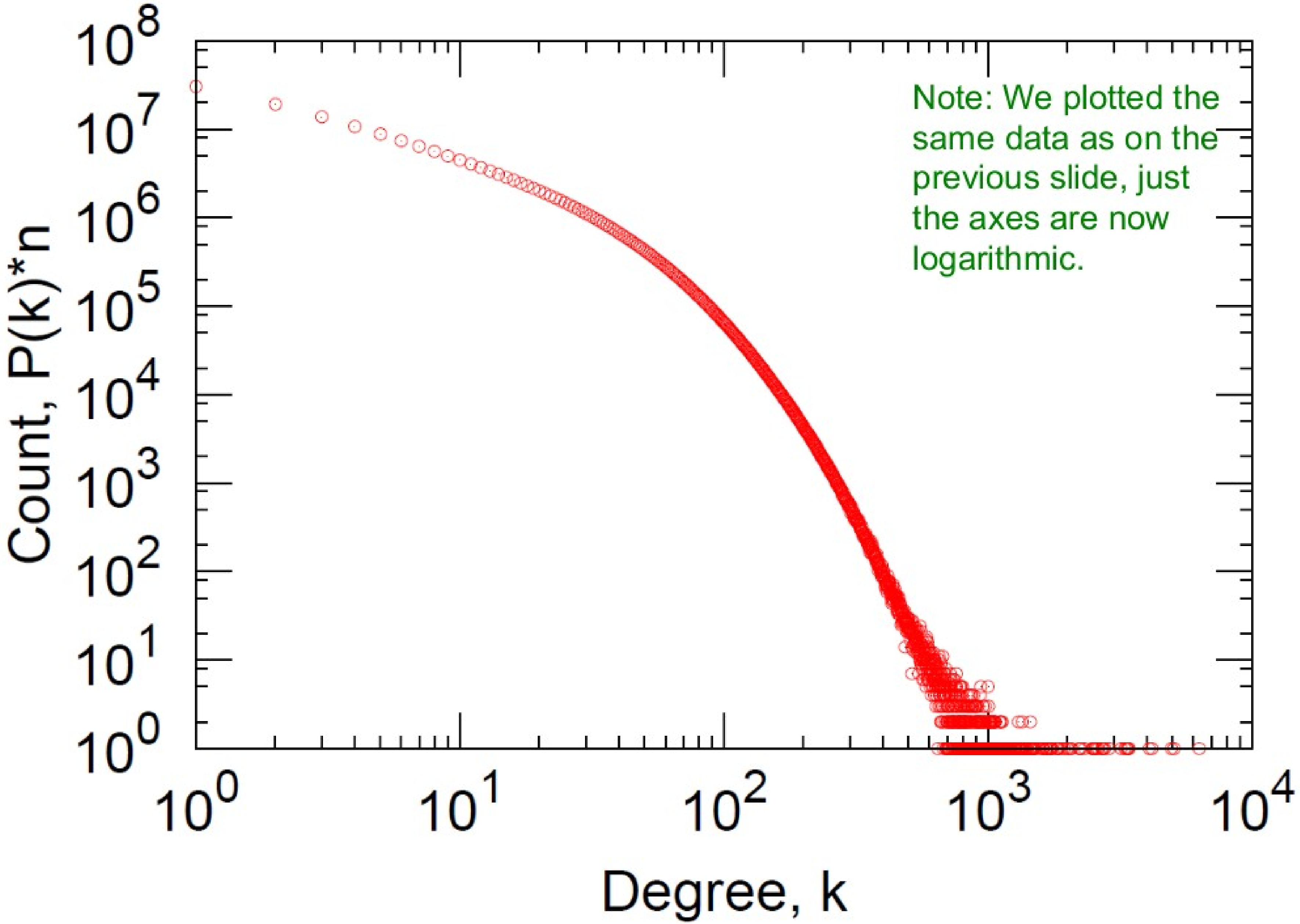
Messaging as an undirected graph

- Edge (u,v) if users *u* and *v* exchanged at least 1 msg
- N=180 million people
- E=1.3 billion edges

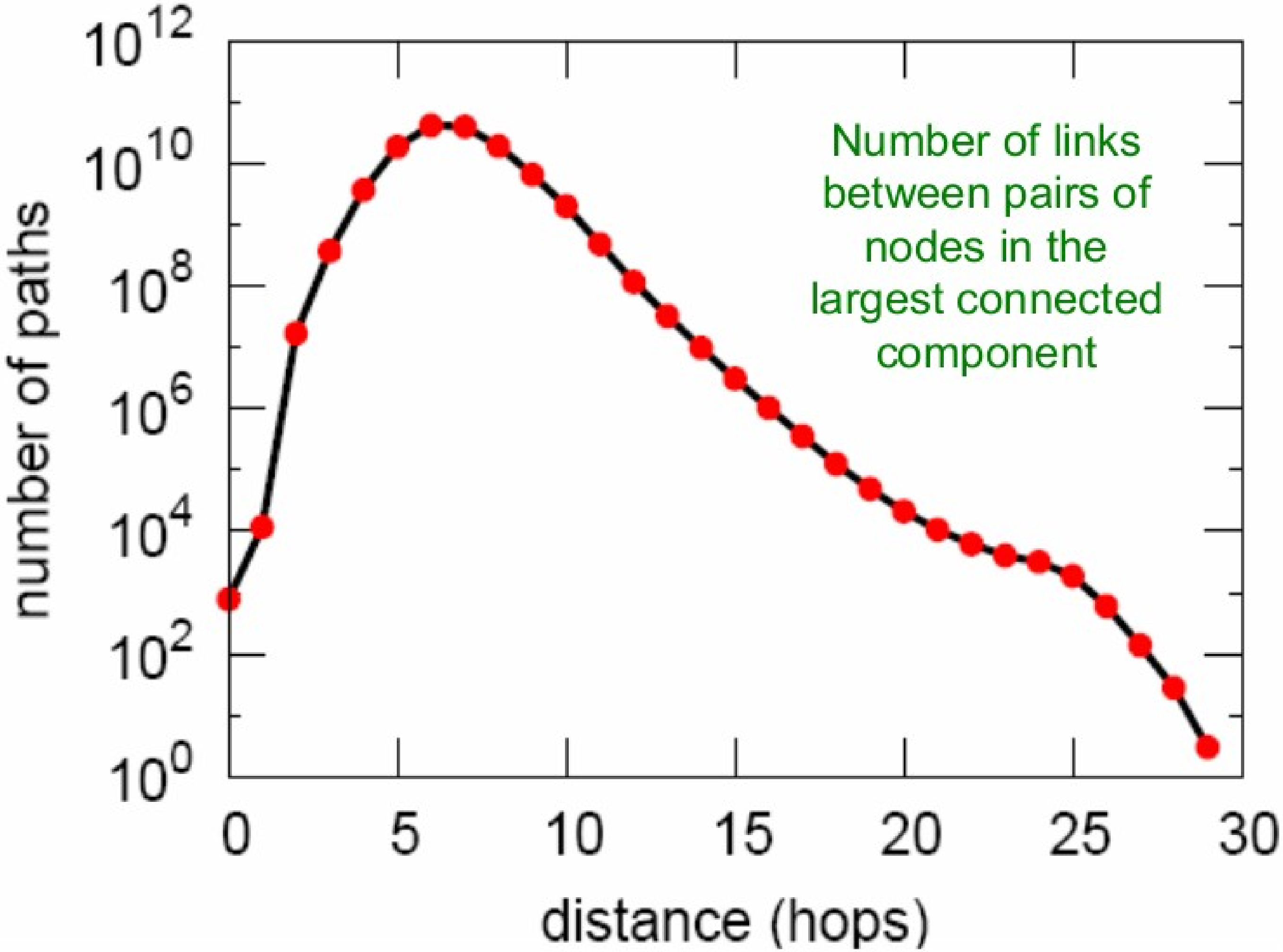
MSN: (1) Degree Distribution



MSN: Log-Log Degree Distribution



MSN: (2) Diameter

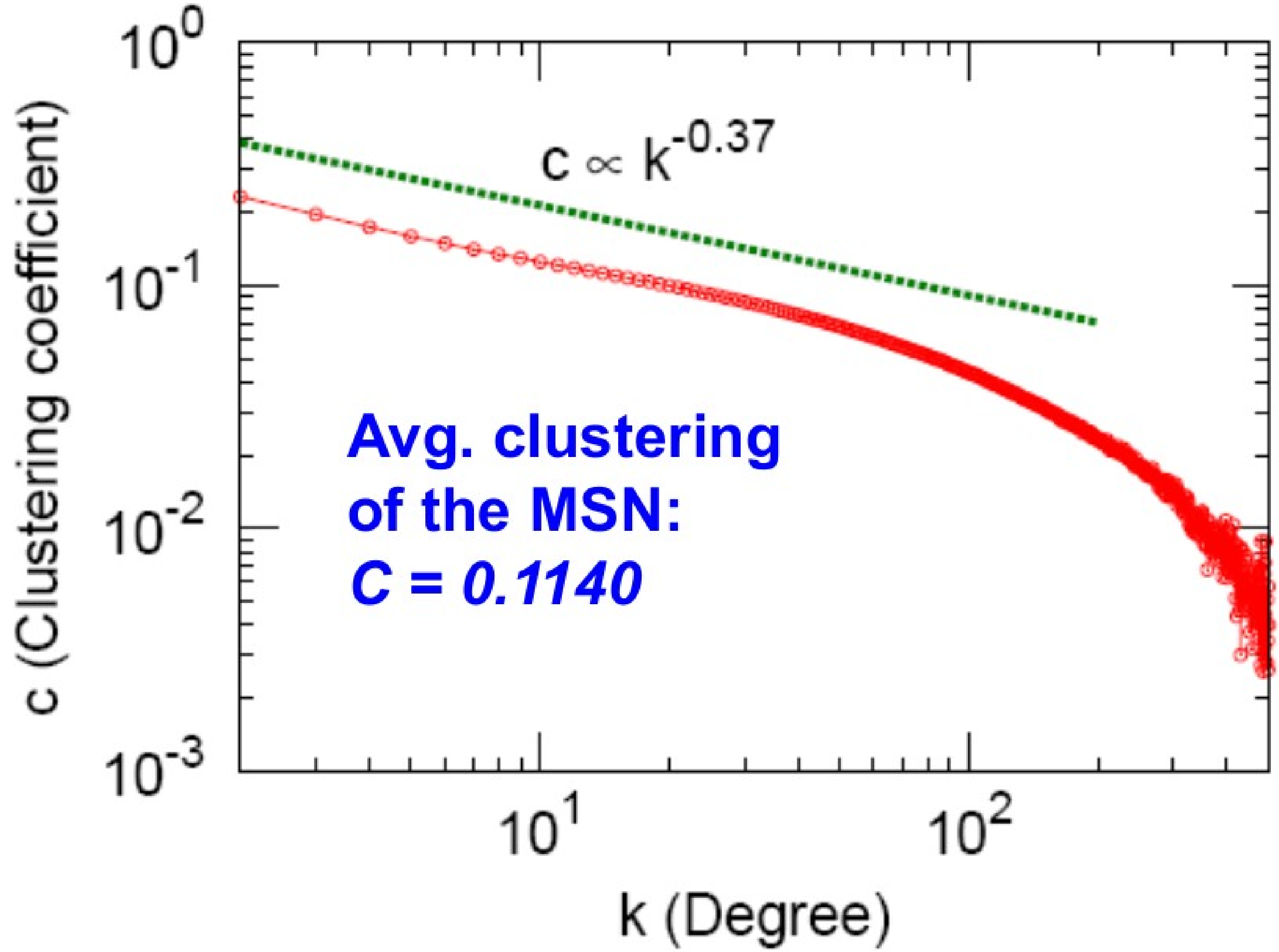


Avg. path length **6.6**
90% of the nodes can be reached in < 8 hops

nodes as we do BFS out of a random node

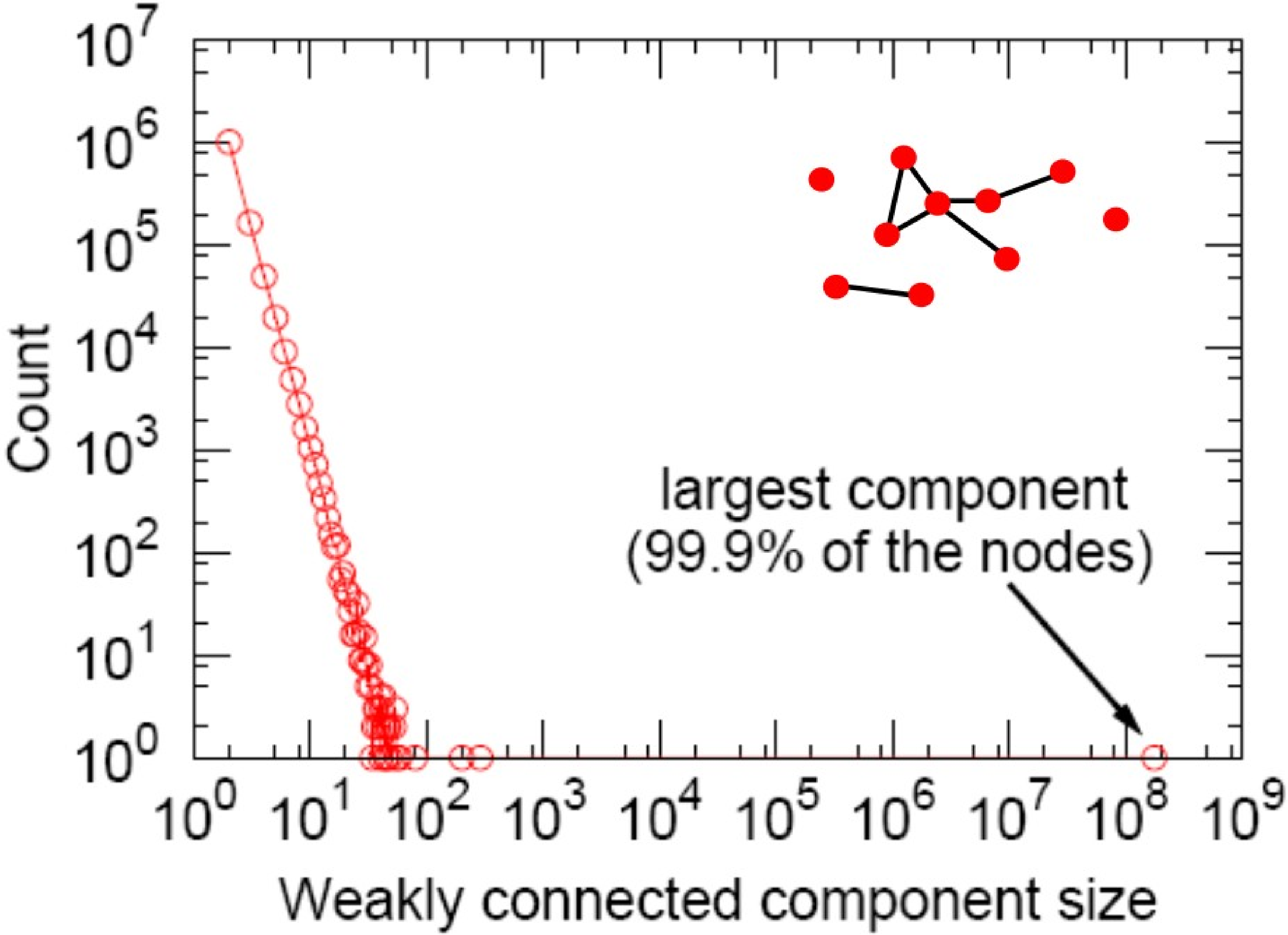
Steps	#Nodes
0	1
1	10
2	78
3	3,96
4	8,648
5	3,299,252
6	28,395,849
7	79,059,497
8	52,995,778
9	10,321,008
10	1,955,007
11	518,410
12	149,945
13	44,616
14	13,740
15	4,476
16	1,542
17	536
18	167
19	71
20	29
21	16
22	10
23	3
24	2
25	3

MSN: (3) Clustering Coefficient



C_k: average C_i of nodes *i* of degree *k*: $C_k = \frac{1}{N_k} \sum_{i:k_i=k} C_i$

MSN: (4) Connected Components



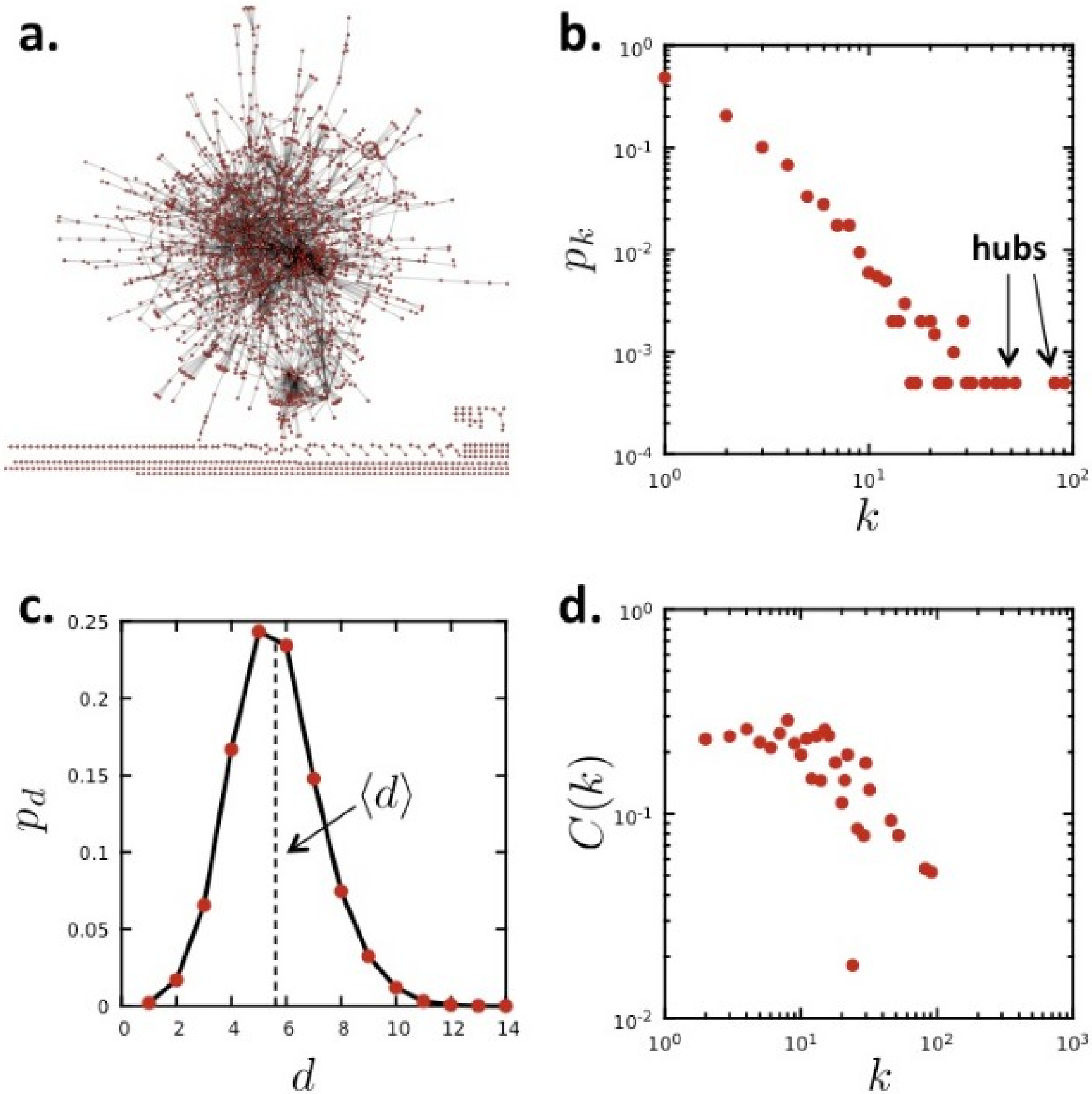
MSN: Key Network Properties

- (1) Degree distribution — ***Heavily skewed avg. degree = 14.4***
- (2) Path Length — ***6.6***
- (3) Clustering coefficient — ***0.11***
- (4) Connected components — ***giant component***

Are these values “expected”?
Are they “surprising”?

To answer this we need a null-model!

Another Example: PPI Network



a. Undirected network

N=2,018 proteins as nodes

E=2,930 binding interactions as links.

b. Degree distribution:

Skewed. Average degree <k>=2.90

c. Diameter:

Avg. path length = 5.8

d. Clustering:

Avg. clustering = 0.12

Connectivity: 185 components

the largest component 1,647 nodes (81% of nodes)

Intermezzo: Network Datasets

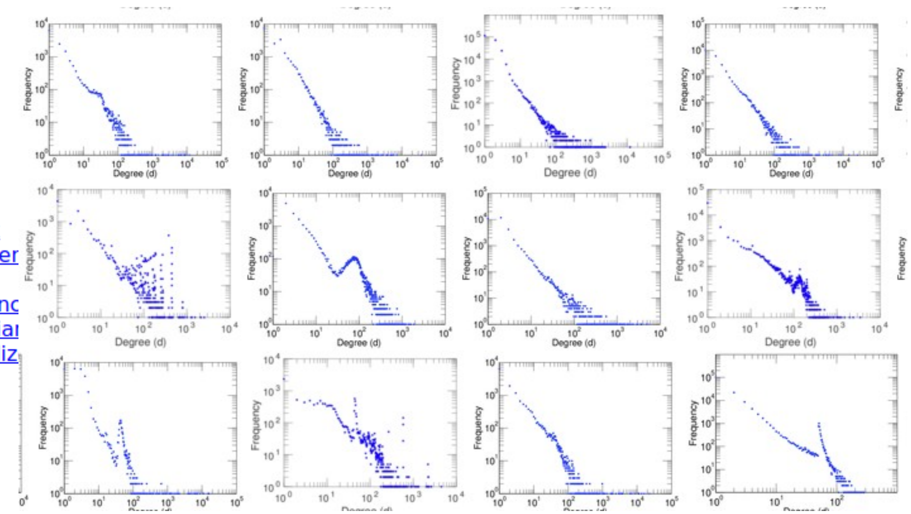
The KONECT Project

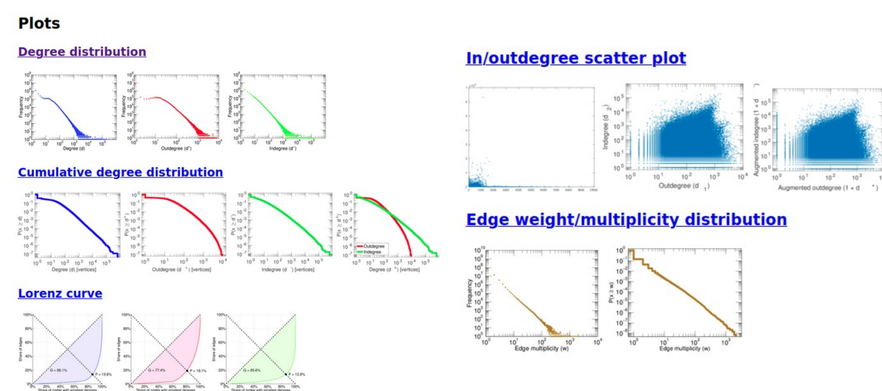
Networks • Statistics • Plots • Categories • Handbook

Jérôme Kunegis
University of Namur

n =	Size	$\in \mathbb{N}$
m =	Volume	$\in \mathbb{N}$
$\bar{m}$ =	Unique edge count	$\in \mathbb{N}$
l =	Loop count	$\in \mathbb{N}$
s =	Wedge count	$\in \mathbb{N}$
z =	Claw count	$\in \mathbb{N}$
x =	Cross count	$\in \mathbb{N}$
t =	Triangle count	$\in \mathbb{N}$
q =	Square count	$\in \mathbb{N}$
T_4 =	4-Tour count	$\in \mathbb{N}$
d_{max} =	Maximum degree	$\in \mathbb{N}$
d =	Average degree	$\in \mathbb{R}^+$
p =	Fill	$\in [0, 1]$
$\tilde{m}$ =	Average edge multiplicity	$\in \mathbb{R}^+$
N =	Size of LCC	$\in \mathbb{N}$
N_s =	Size of LSCC	$\in \mathbb{N}$
δ =	Diameter	$\in \mathbb{N}$
$\delta_{0.5}$ =	50-Percentile effective diameter	$\in \mathbb{R}^+$
$\delta_{0.9}$ =	90-Percentile effective diameter	$\in \mathbb{R}^+$
δ_M =	Median distance	$\in \mathbb{N}$
δ_m =	Mean distance	$\in \mathbb{R}^+$
G =	Gini coefficient	$\in [0, 1]$
P =	Balanced inequality ratio	$\in [0, 1]$
H_{er} =	Relative edge distribution entropy	$\in [0, 1]$

- Fruchterman–Reingold graph drawing
- Degree distribution
- Cumulative degree distribution
- Lorenz curve
- Spectral distribution of the adjacency matrix
- Spectral distribution of the normalized adjacer
- Spectral distribution of the Laplacian
- Spectral graph drawing based on the adjacenc
- Spectral graph drawing based on the Laplaciar
- Spectral graph drawing based on the normaliz
- Degree assortativity
- Zipf plot
- Hop distribution
- Double Laplacian graph drawing
- Delaunay graph drawing
- In/outdegree scatter plot
- Item rating evolution
- Edge weight/multiplicity distribution
- Clustering coefficient distribution
- Average neighbor degree distribution
- Temporal distribution
- Temporal hop distribution
- Diameter/density evolution
- Signed temporal distribution
- Rating class evolution
- SynGraphy
- Inter-event distribution
- Node-level inter-event distribution





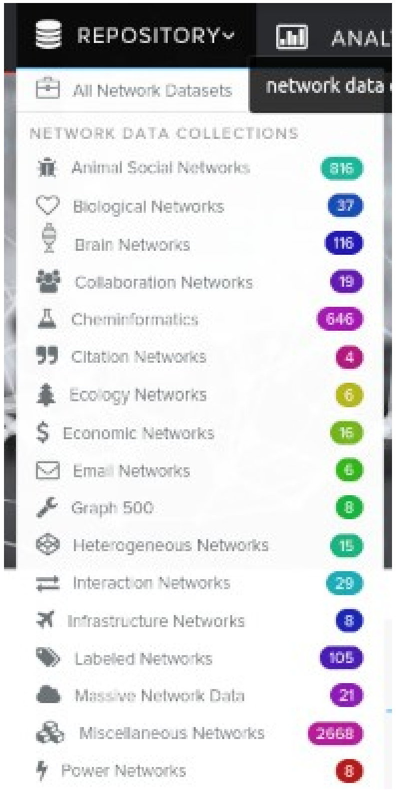
http://konect.cc/

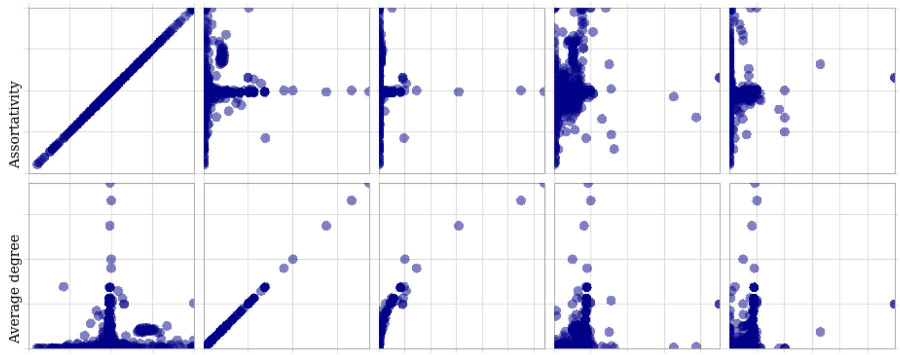
Intermezzo: Network Datasets

Network Repository. An **Interactive** *Scientific* Network Data Repository.

THE FIRST SCIENTIFIC NETWORK DATA REPOSITORY WITH INTERACTIVE VISUAL ANALYTICS.

NEW GraphVis: interactive visual graph mining and machine learning





Network Data Statistics	
Nodes	15.2K
Edges	246K
Density	0.00212112
Maximum degree	375
Minimum degree	1
Average degree	32
Assortativity	0.343624
Number of triangles	6.7M
Average number of triangles	442
Maximum number of triangles	14.5K
Average clustering coefficient	0.211165
Fraction of closed triangles	0.287461
Maximum k-core	79
Lower bound of Maximum Clique	43

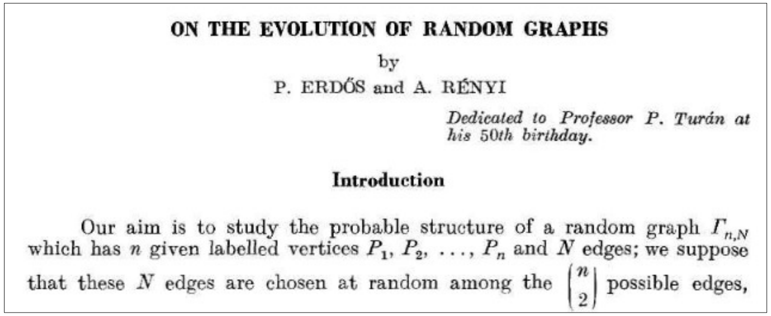
Graph Name	|V|	|E|	d_{max}	d_{avg}	r	|T|	T_{avg}	T_{max}	κ_{avg}	κ	K	ω_{heu}	Size	Download
bio-CE-CX	15K	246K	375	32	0.34	7M	442	14K	0.21	0.29	79	43	3 MB	Download
bio-CE-GN	2K	54K	242	48	0.07	686K	308	3K	0.18	0.14	49	16	512 KB	Download
bio-CE-GT	924	3K	151	7	-0.18	12K	12	684	0.61	0.13	10	8	30 KB	Download
bio-CE-HT	3K	3K	44	2	-0.30	87	-	4	0.01	0.01	4	4	19 KB	Download
bio-CE-LC	1K	2K	131	2	-0.17	699	-	31	0.08	0.04	7	7	10 KB	Download
bio-CE-PG	2K	48K	913	51	-0.15	2M	1K	30K	0.42	0.32	81	28	500 KB	Download

http://networkrepository.com/

Erdös-Renyi
Random Graph Model

Simplest Model of Graphs

- **Erdös-Renyi Random Graphs**
  [Erdös-Renyi, '60]



ON THE EVOLUTION OF RANDOM GRAPHS

by

P. ERDŐS and A. RÉNYI

Dedicated to Professor P. Turán at his 50th birthday.

Introduction

Our aim is to study the probable structure of a random graph $\Gamma_{n,N}$ which has n given labelled vertices $P_1, P_2, \ldots, P_n$ and N edges; we suppose that these N edges are chosen at random among the $\binom{n}{2}$ possible edges,

- $G_{n,p}$: undirected graph on n nodes and each (u,v) appears i.i.d. with probability p

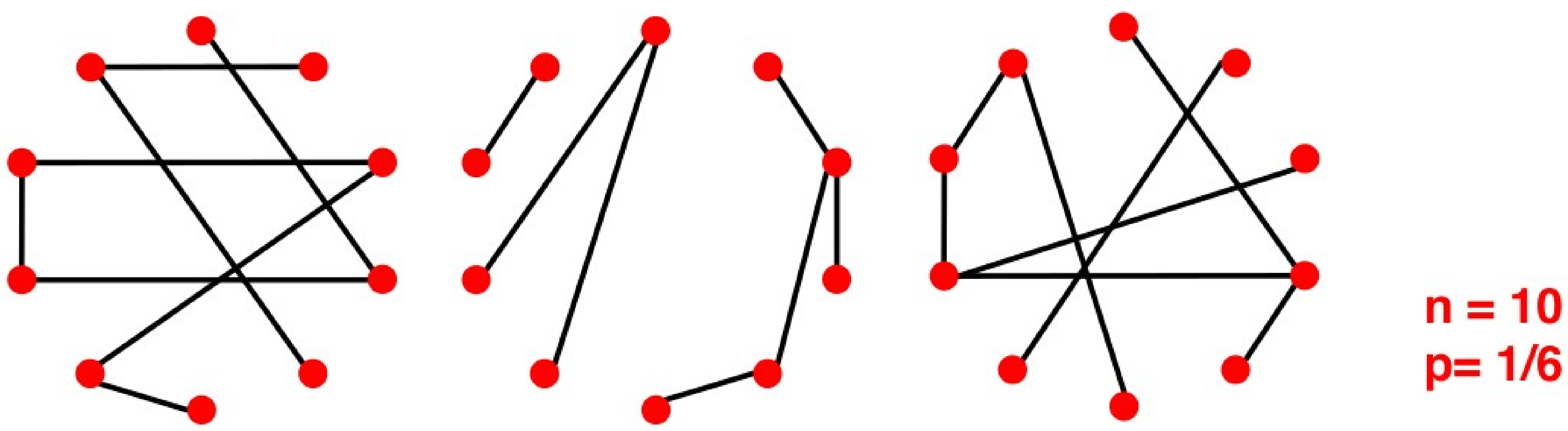
- $G_{n,m}$: undirected graph with n nodes and m uniformly at random picked edges

What kind of networks do such models produce?

Random Graph Model

- ***n*** and ***p*** do not uniquely determine the graph!
  - The graph is a result of a random process

- We can have many different realizations given the same ***n*** and ***p***



Properties of $G_{n,p}$

- Degree distribution $P(k)$
- Clustering coefficient C
- Path Length h
- Connected components s

What are the values of these properties for $G_{n,p}$?

$G_{n,p}$: degree distribution

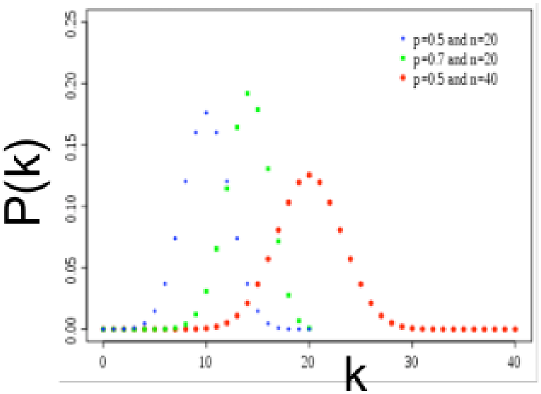
- Fact: Degree Distribution of $G_{n,p}$ is **binomial**
- Let ***P(k)*** denote the fraction of nodes with degree ***k***

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Select *k* nodes out of *n-1*

Probability of having *k* edges

Probability of missing the rest of the *n-1-k* edges



Mean, variance of a binomial distribution

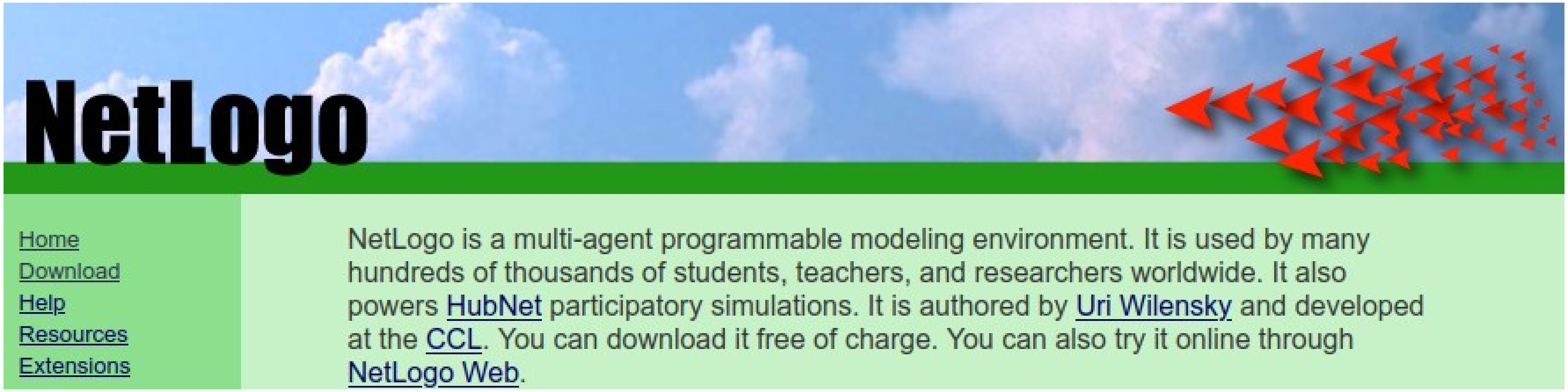
$$\bar{k} = p(n-1)$$

$$\sigma^2 = p(1-p)(n-1)$$

$$\frac{\sigma}{\bar{k}} = \left[\frac{1-p}{p}\frac{1}{(n-1)}\right]^{1/2} \approx \frac{1}{(n-1)^{1/2}}$$

By the law of large numbers, as the network size increases, the distribution becomes increasingly narrow—we are increasingly confident that the degree of a node is in the vicinity of *k*.

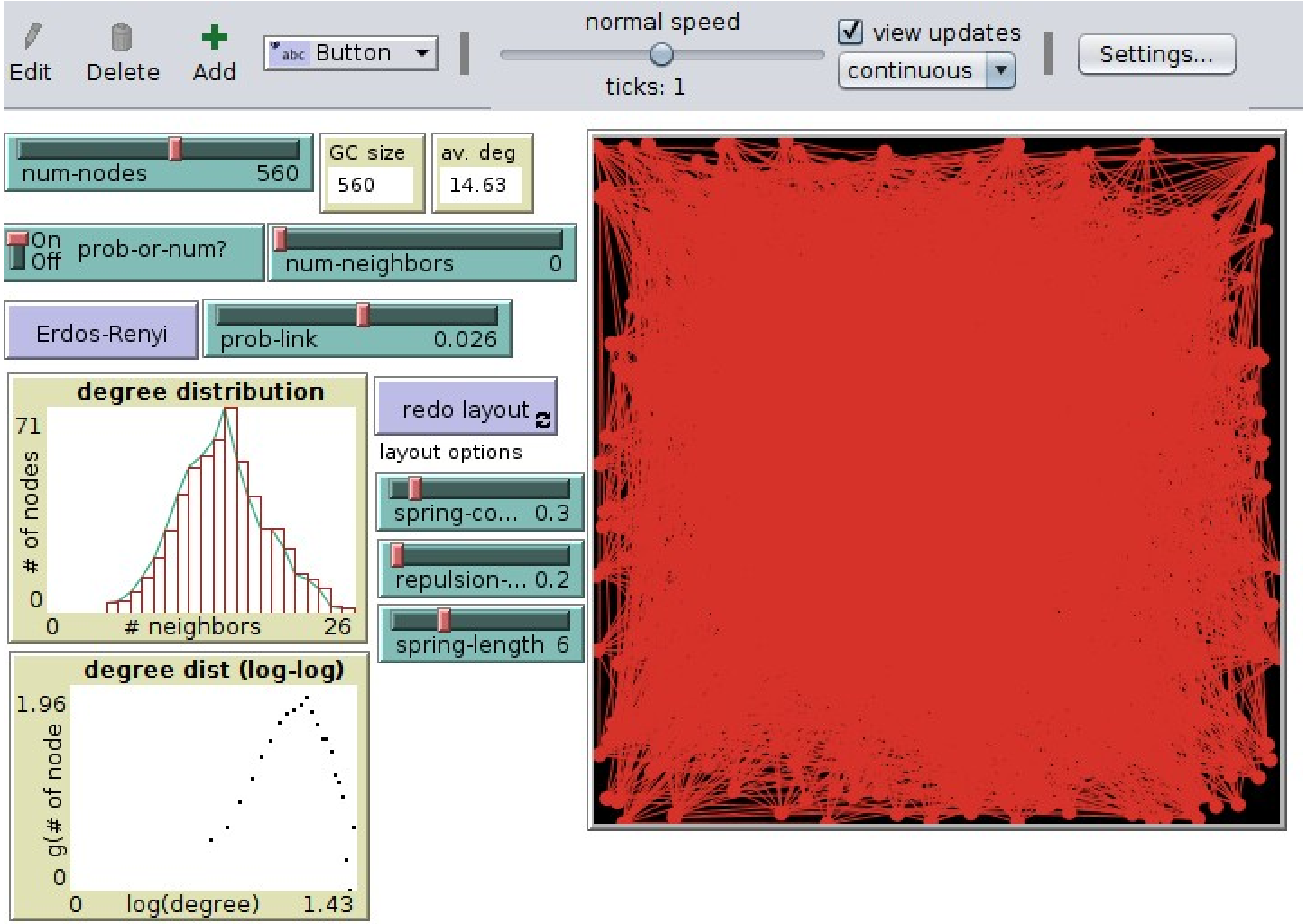
Intermezzo: NetLogo



Visualize some of the properties described in this course

https://ccl.northwestern.edu/netlogo/

NetLogo: $G_{n,p}$ *and* degree dist.



ErdosRenyiDegDist.nlogo

$G_{n,p}$: clustering coefficient

- Remember: $C_i = \frac{2e_i}{k_i(k_i-1)}$

where e_i is the number of edges between the neighbors of node i

- Edges in $G_{n,p}$ appear i.i.d. with prob. p

- So, expected $E[e_i]$ is $= p \frac{k_i(k_i-1)}{2}$

each pair is connected with prob. p

number of distinct pairs of neighbors of node i of degree k_i

- Therefore $E[C]$ = $\frac{p \cdot k_i(k_i-1)}{k_i(k_i-1)} = p = \frac{\bar{k}}{n-1} \approx \frac{\bar{k}}{n}$

Clustering coefficient of a random graph is small.
If we generate bigger and bigger graphs with fixed avg. degree k (that is we set $p = k \cdot 1/n$), then C decreases with the graph size n.

Properties of $G_{n,p}$

- Degree distribution $$P(k)=\binom{n-1}{k}p^k(1-p)^{n-1-k}$$
- Clustering coefficient $$C=p\approx\frac{\bar{k}}{n}$$
- Path Length ***next!***
- Connected components

What are the values of these properties for $G_{n,p}$?

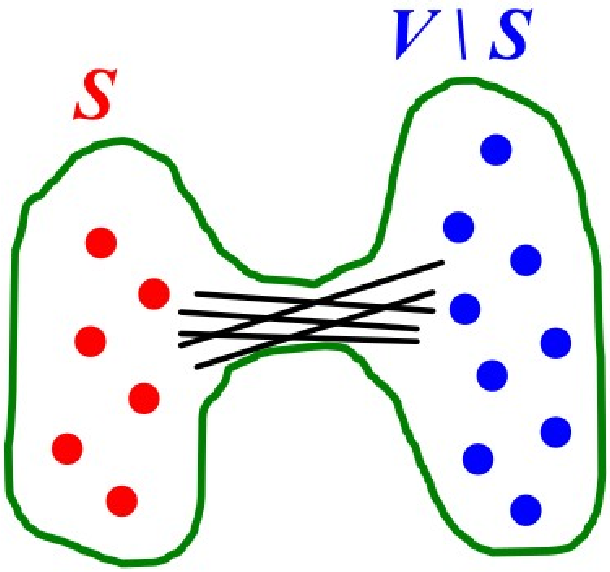
Definition: expansion

- Graph *G(V,E)* has **expansion α**: $if\ \forall S \subseteq V:$

  # of edges leaving $S \geq \alpha \cdot min\left(|S|,|V \setminus S|\right)$

- Or equivalently:

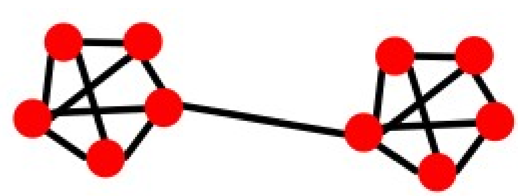
$$\alpha = \min_{S \subseteq V} \frac{\#\,edges\ leaving\ S}{\min(|S|,|V \setminus S|)}$$



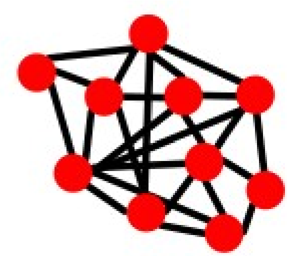
Expansion: measures robustness

- Expansion is measure of robustness:
  - to disconnect L nodes, we need to $cut \geq \alpha \cdot L\ edges$

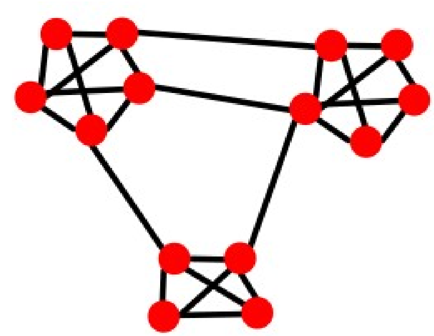
- Low expansion



- High Expansion

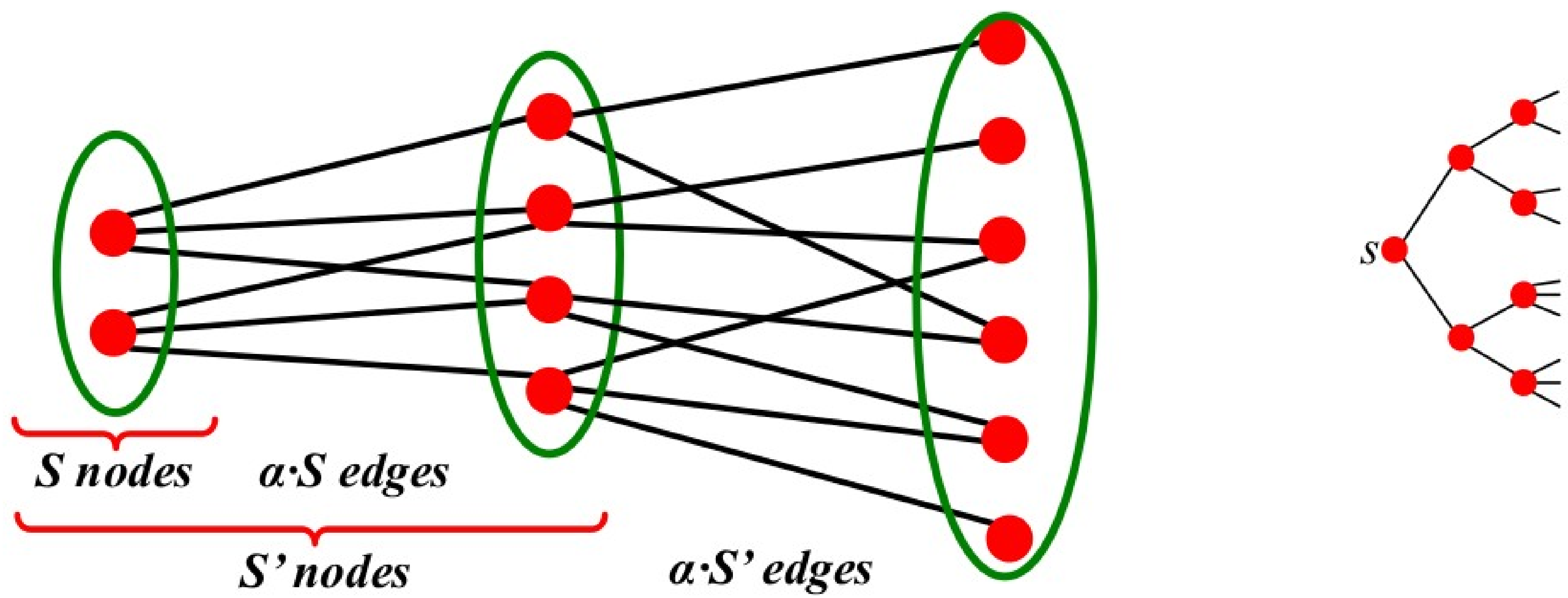


- Social Networks:
  - “communities”



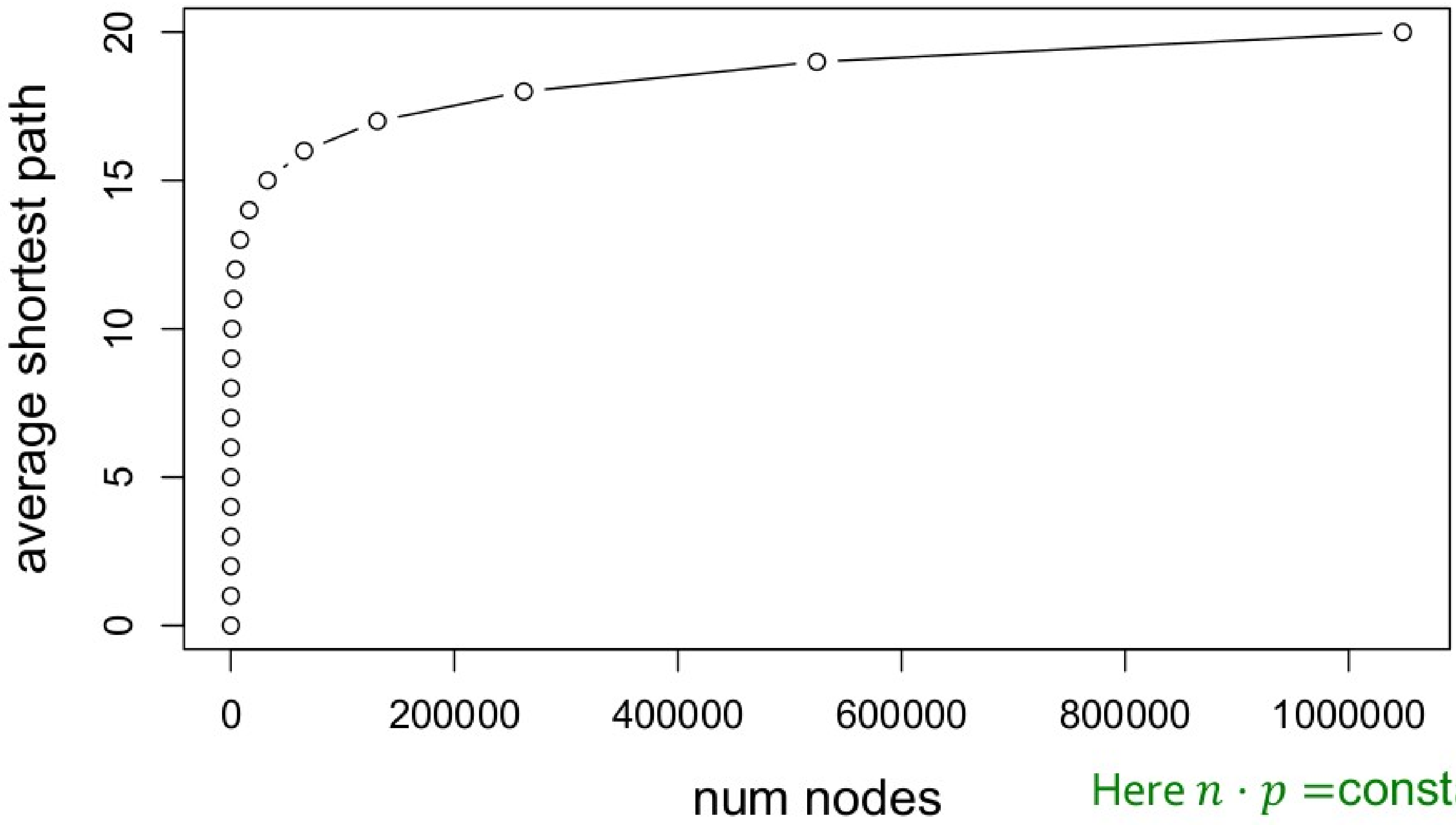
Expansion: $G_{n,p}$

- Fact: In a graph of ***n*** nodes with expansion α for all pairs of nodes there is a path of length ***O((log n)/α)***.
- **Random graph $G_{n,p}$:**
  For $log\ n > np > c$, diam$(G_{n,p}) = O(log\ n\ /\ log\ (np))$
  - random graphs have good expansion, so it takes a logarithmic number of steps for BFS to visit all nodes



$G_{n,p}$: average shortest path

Erdös-Renyi Random Graphs can grow very large but nodes will be just a few hops apart



Here $n \cdot p$ =constant
That is, avg deg k is const

Properties of $G_{n,p}$

- Degree distribution $$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$
- Clustering coefficient $$C = p \approx \frac{\bar{k}}{n}$$
- Path Length *O(log n)*
- Connected components ***next!***

What are the values of these properties for $G_{n,p}$?

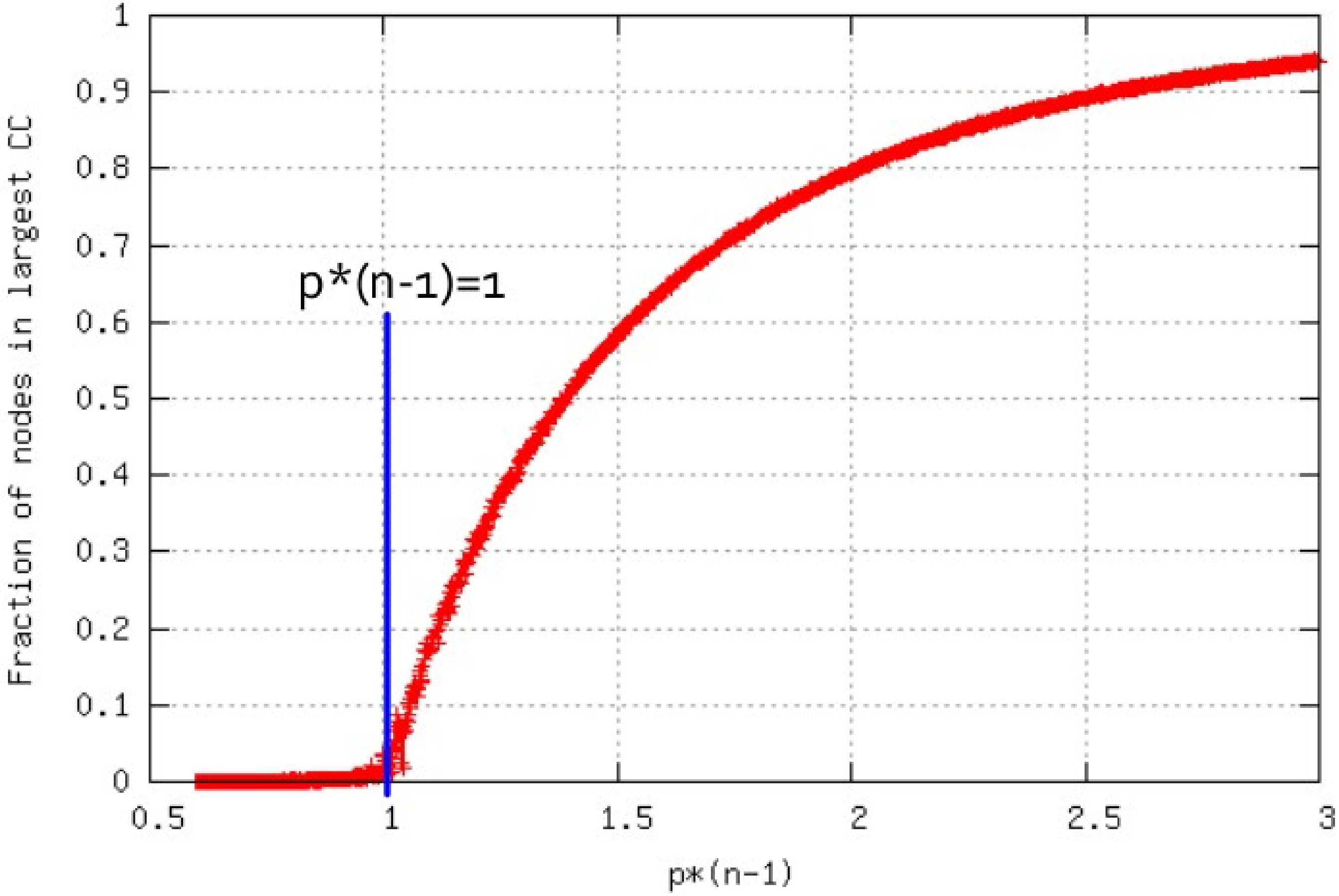
"Evolution" of a random graph

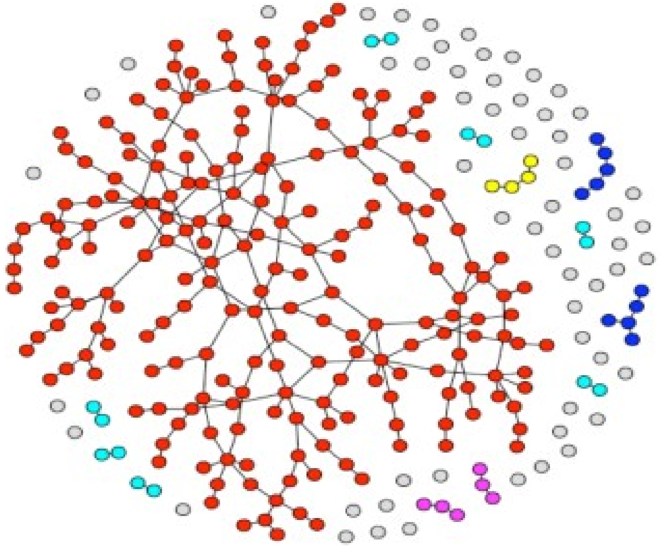
- Graph structure of $G_{n,p}$ as p changes

- Emergence of a **giant component**

  avg. degree ***k=2E/n or p=k/(n-1)***
  - *k=1-ε:* all components are of size *Ω(log n)*
  - *k=1+ε:* 1 component of size *Ω(n)*, others have size *Ω(log n)*
    - Each node has at least one edge in expectation

$G_{n,p}$ Simulation Experiment

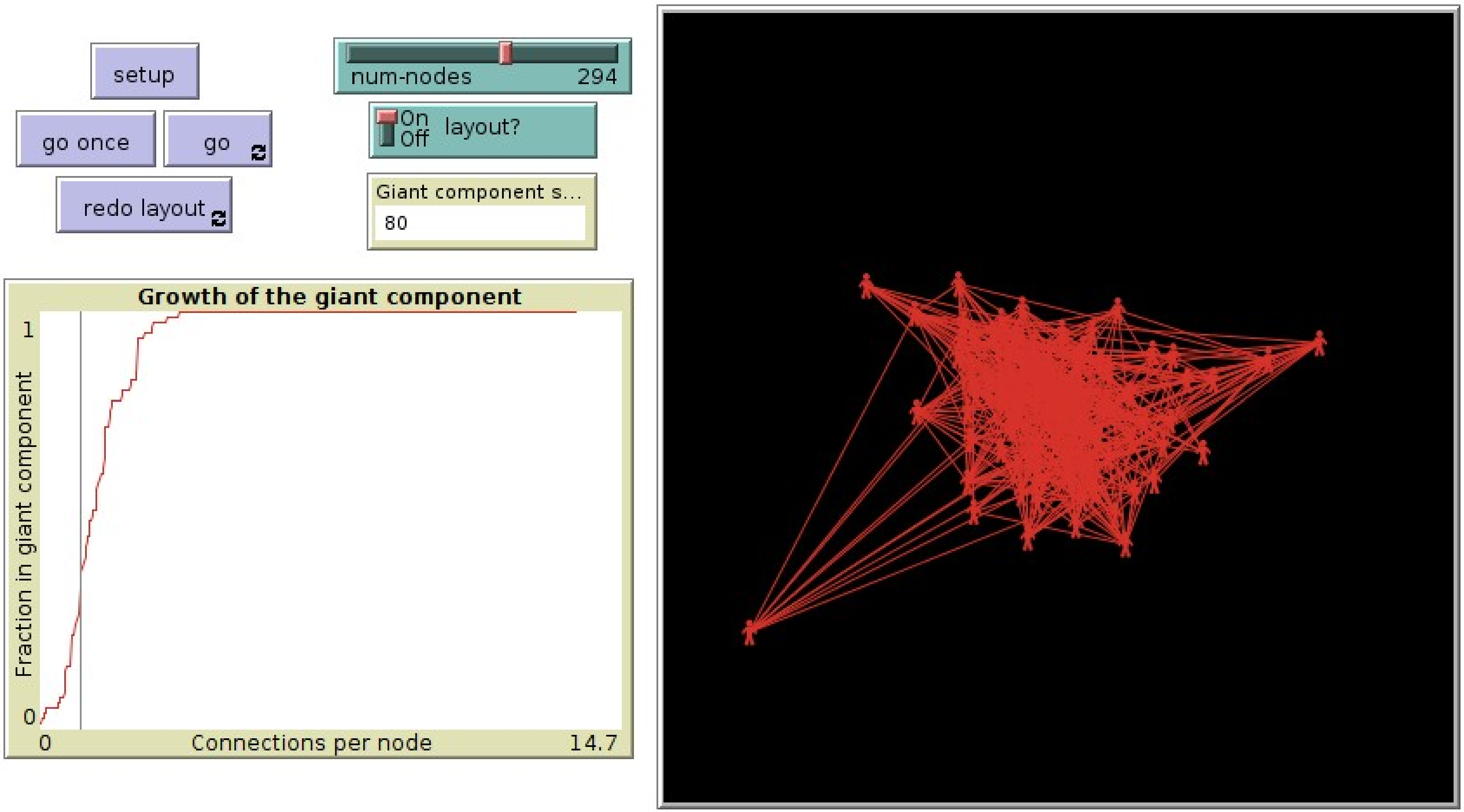




Fraction of nodes in the largest component

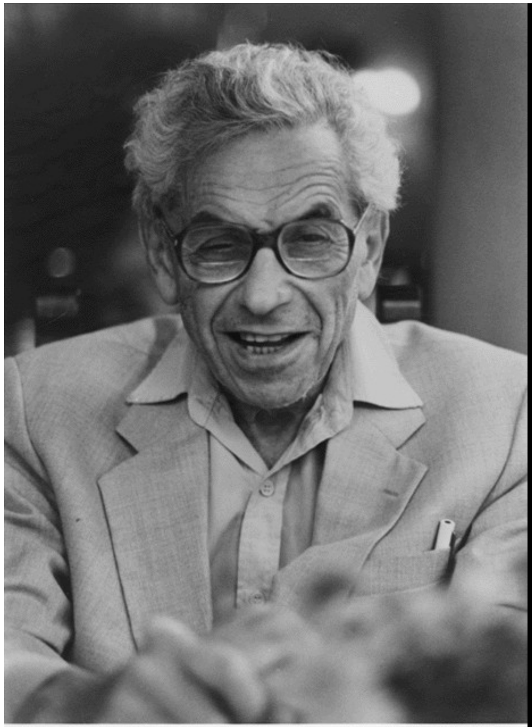
- $G_{n,p}$, $n=10^6$, $k=p(n-1) = 0.5 \ldots 3$

NetLogo: $G_{n,p}$ *and* giant component



GiantComponent.nlogo

$G_{n,p}$ - Erdös-Renyi Model



Paul Erdös, the most prolific mathematician who ever lived, has no home and no job, but he has wandered the world for over fifty years, inspiring other mathematicians. From the documentary N is a Number: A Portrait of Paul Erdös © 1993 by George Csicsery

"[When asked why are numbers beautiful?]

It's like asking why is Ludwig van Beethoven's Ninth Symphony beautiful. If you don't see why, someone can't tell you. I know numbers are beautiful. If they aren't beautiful, nothing is."

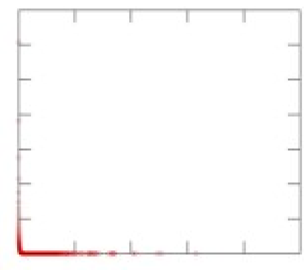
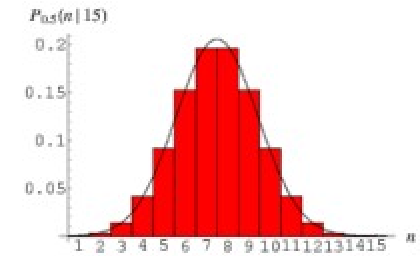
— Paul Erdos

- $G_{n,p}$ is a cool model!

  But let's compare it to real world networks

MSN vs $G_{n,p}$

n=180M

	MSN	$G_{n,p}$
• Degree distribution		
• Avg. Clustering coef.	*0.11*	$\bar{k}/n$ C ≈ $8 \cdot 10^{-8}$
• Path Length	*6.6*	*O(log n)* h ≈ 8.2
• Largest Conn. Comp.	99%	GCC exists when $\bar{k} > 1$ $\bar{k} \approx 14$

Real Networks *vs* $G_{n,p}$

- Are real networks like random graphs?
  - Average Path Length ✔
  - Giant Connected Component ✔
  - Degree Distribution ✘
  - Clustering Coefficient ✘
- **Problems** with the random networks model:
  - Degree distribution differs from that of real networks
  - Clustering Coefficient is much lower than on real networks
  - Giant component in most real network does NOT emerge through a phase transition
- Most important: **Are real networks random?**
  - The answer is simply: **NO!**

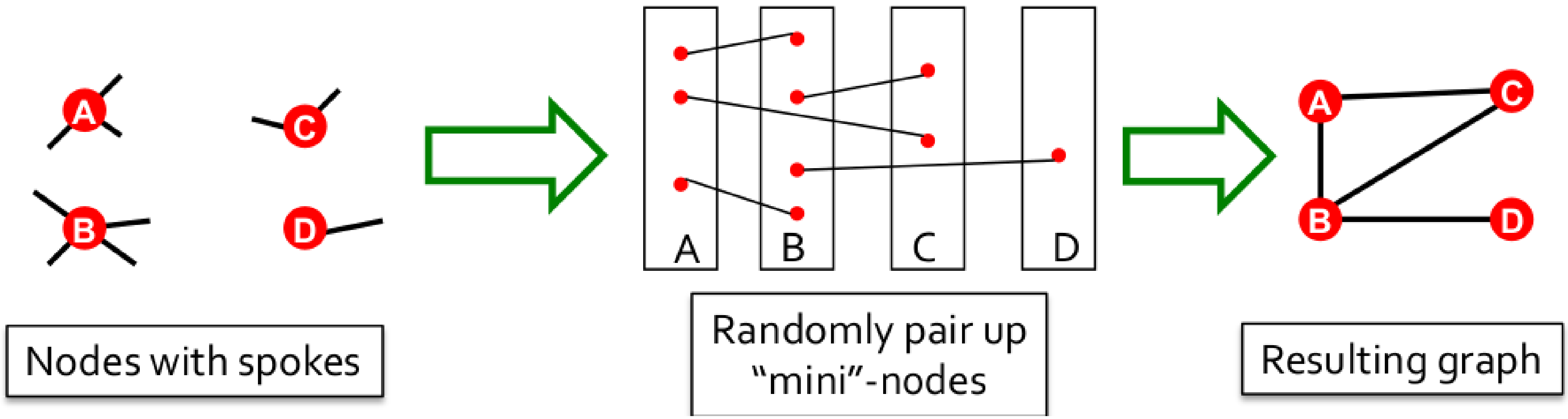
Real Networks *vs* $G_{n,p}$

- If $\mathbf{G}_{n,p}$ is wrong, why did we spend time on it?
  - It is the reference model
  - It will help us calculate many quantities, that can then be compared to the real data
  - It will help us understand to what degree is a particular property the result of some random process

So, while $G_{n,p}$ is "WRONG", it can turn out to be extremely USEFUL!

Intermezzo: Configuration Model

- Goal: Generate a random graph with a given degree sequence $k_1, k_2, \ldots k_N$
- **Configuration Model:**



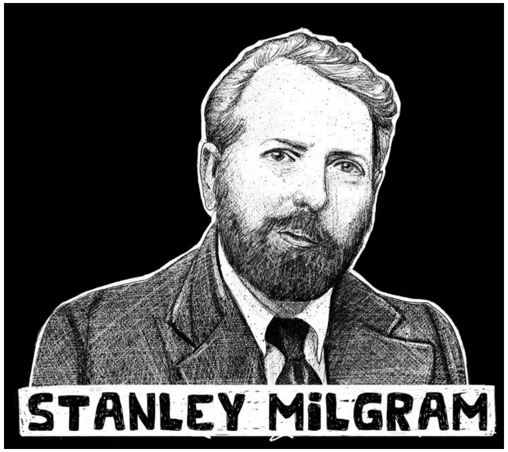
- Useful as a "null" model of networks:
  - We can compare the real network ***G*** and a "random" ***G'*** which has the same degree sequence as ***G***

The Small World Random Graph Model

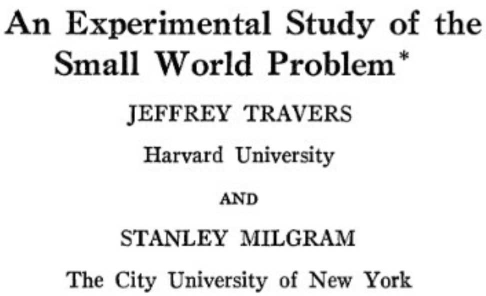
Can we have high clustering while also having short paths?

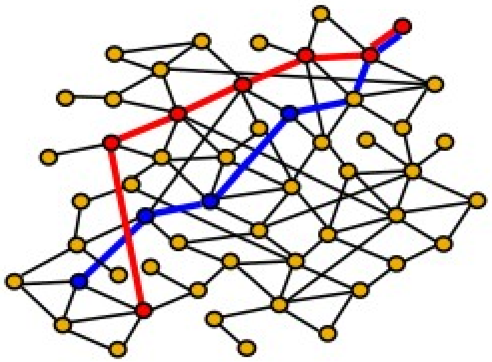
The Small World Experiment

- What is the **typical shortest path length** between any two persons?
  - Experiment on the global friendship network
    - Can’t measure, need to probe explicitly
- **Small-world experiment**
  [Milgram’67] [Travers and Milgram ’69]
  - Picked 296 people in Omaha, Nebraska and Wichita, Kansas
  - Ask them to get a letter to a stock-broker in Boston by passing it through friends
- How many steps did it take?




The Small-World Problem
By Stanley Milgram

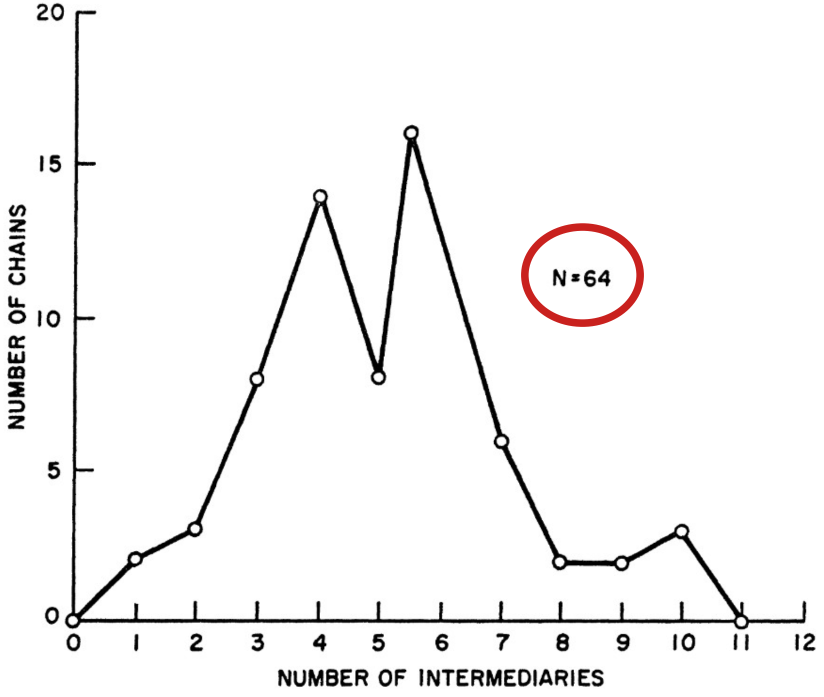

An Experimental Study of the Small World Problem*
JEFFREY TRAVERS
Harvard University
AND
STANLEY MILGRAM
The City University of New York



The Small World Experiment

- 64 chains completed:
  (i.e., 64 letters reached the target)
  - It took 6.2 steps on the average, thus **“6 degrees of separation”**

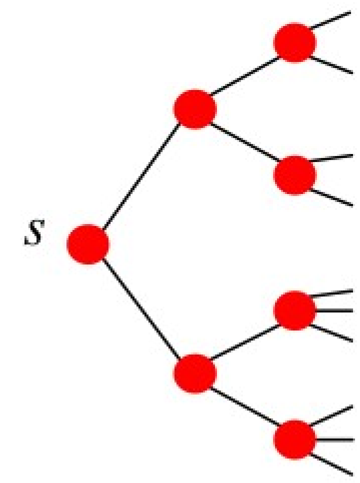
[Travers and Milgram ’69]



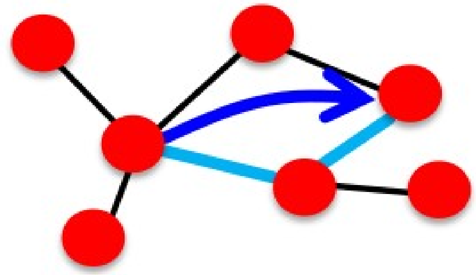
- Further observations:
  - People who owned stock had shorter paths to the stockbroker than random people: 5.4 vs. 6.7
  - People from the Boston area have even closer paths: 4.4

6 degrees: Should we be surprised?

- Assume each human is connected to 100 other people
  Then:
  - Step 1: reach 100 people
  - Step 2: reach 100*100 = 10,000 people
  - Step 3: reach 100*100*100 = 1M people
  - Step 4: reach 100*100*100*100 = 100M people
  - **In 5 steps we can reach 10 billion people!**

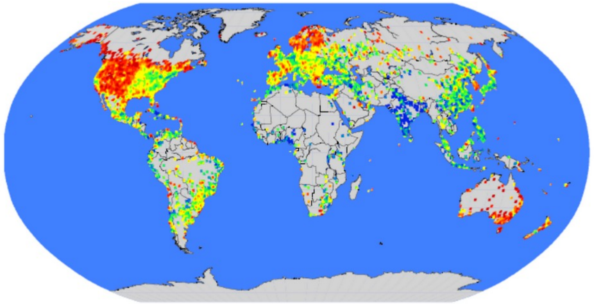


- What's wrong here? We ignore clustering!
  - Not all edges point to new people
    - 92% of FB friendships happen through a **friend-of-a-friend**



Clustering Implies Edge Locality

- MSN network has 7 orders of magnitude larger clustering than the corresponding $G_{n,p}$!



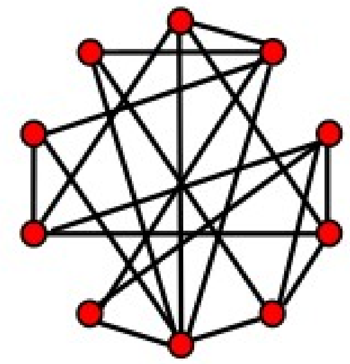
- Other Examples:
  - Actor Collaborations (IMDB): N = 225,226 nodes, avg. degree $\overline{k}$ = 61
  - Electrical power grid: N = 4,941 nodes, $\overline{k}$ = 2.67
  - Network of neurons: N = 282 nodes, $\overline{k}$ = 14

Network	h_{actual}	h_{random}	C_{actual}	C_{random}
Film actors	3.65	2.99	0.79	0.00027
Power Grid	18.70	12.40	0.080	0.005
C. elegans	2.65	2.25	0.28	0.05

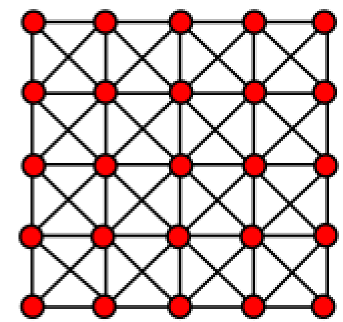
h … Average shortest path length
C … Average clustering coefficient
"actual" … real network
"random" … random graph with same avg. degree

The “Controversy”

- Consequence of expansion:
  - **Short paths: *O(log n)***
    - This is the smallest diameter we can get if we have a constant degree.
  - But clustering is low!
- However, **networks have “local” structure**:
  - **Triadic closure:**
    - Friend of a friend is my friend
  - High clustering but diameter is also high
- **How can we have both?**



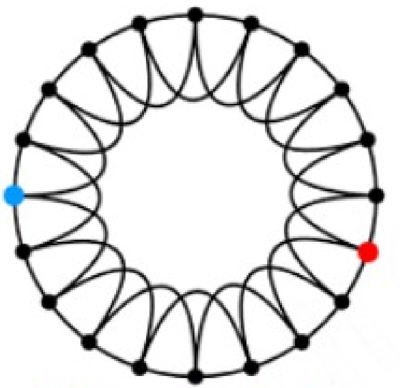
Low diameter
Low clustering coefficient

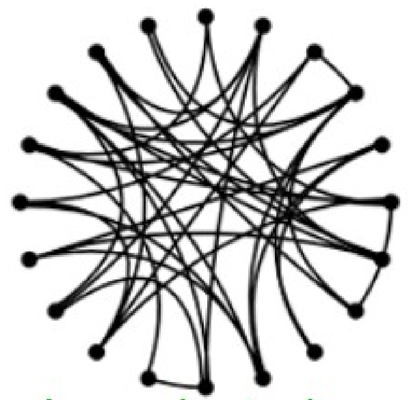


High clustering coefficient
High diameter

Small-World: How?

- Could a network with high clustering also be “small world” (*log n* diameter)?
  - How can we at the same time have **high clustering** and **small diameter**?





  - Clustering implies edge “locality”
  - Randomness enables “shortcuts”

Solution: The Small-World Model

Small-World Model

[Watts-Strogatz '98]

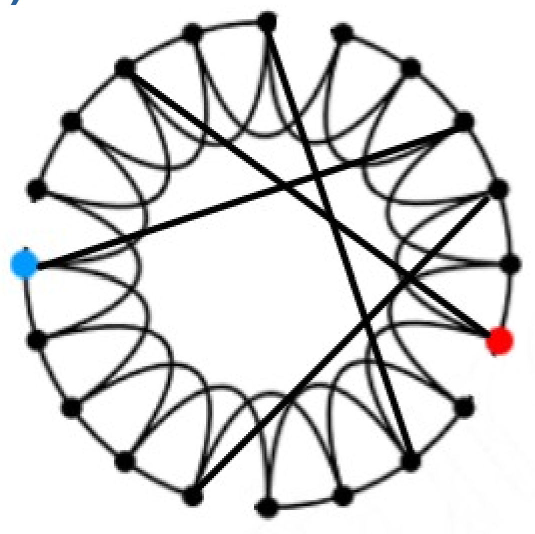
Collective dynamics of 'small-world' networks

Duncan J. Watts* **& Steven H. Strogatz**

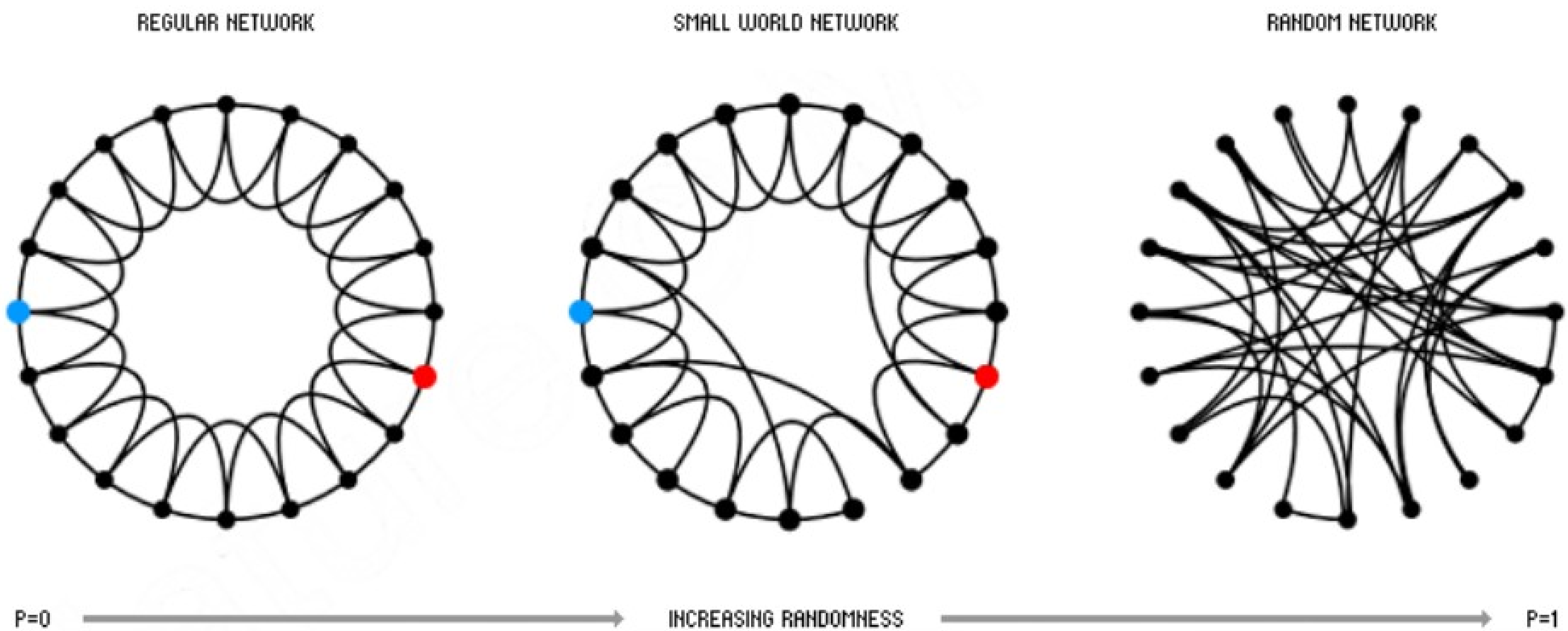
Department of Theoretical and Applied Mechanics, Kimball Hall, Cornell University, Ithaca, New York 14853, USA

Two components to the model:

- (1) Start with a **low-dimensional regular lattice**
  - (In our case we are using a ring as a lattice)
  - Has high clustering coefficient
- Now introduce **randomness** ("shortcuts")
- (2) **Rewire**:
  - Add/remove edges to create shortcuts to join remote parts of the lattice
  - For each edge with prob. p move the other end to a random node



The Small World Model



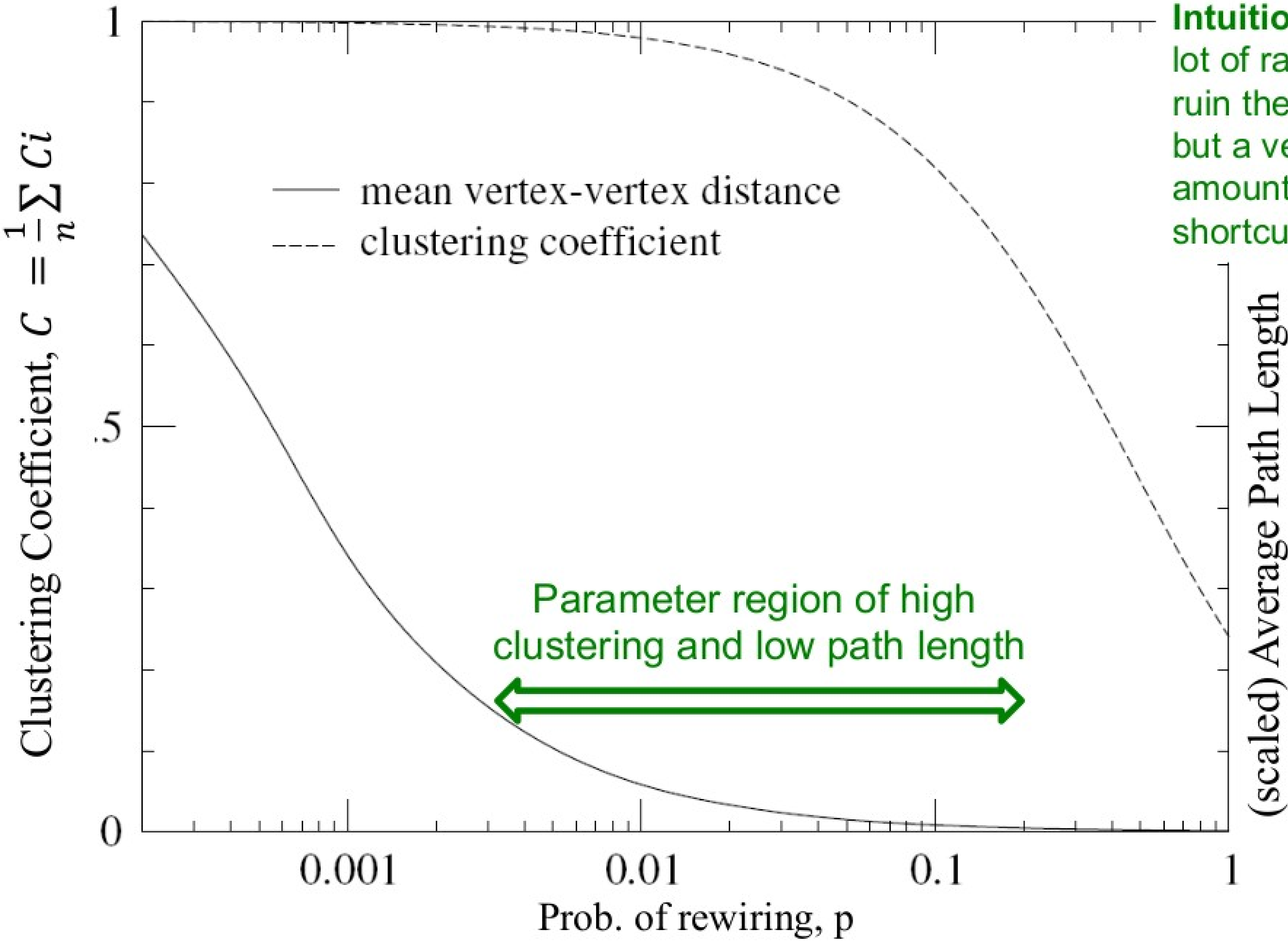
Regular network	Small world network	Random network
High clustering High diameter	High clustering Low diameter	Low clustering Low diameter

$$h = \frac{N}{2\bar{k}} \qquad C = \frac{1}{2}$$

$$h = \frac{\log N}{\log \alpha} \qquad C = \frac{\bar{k}}{N}$$

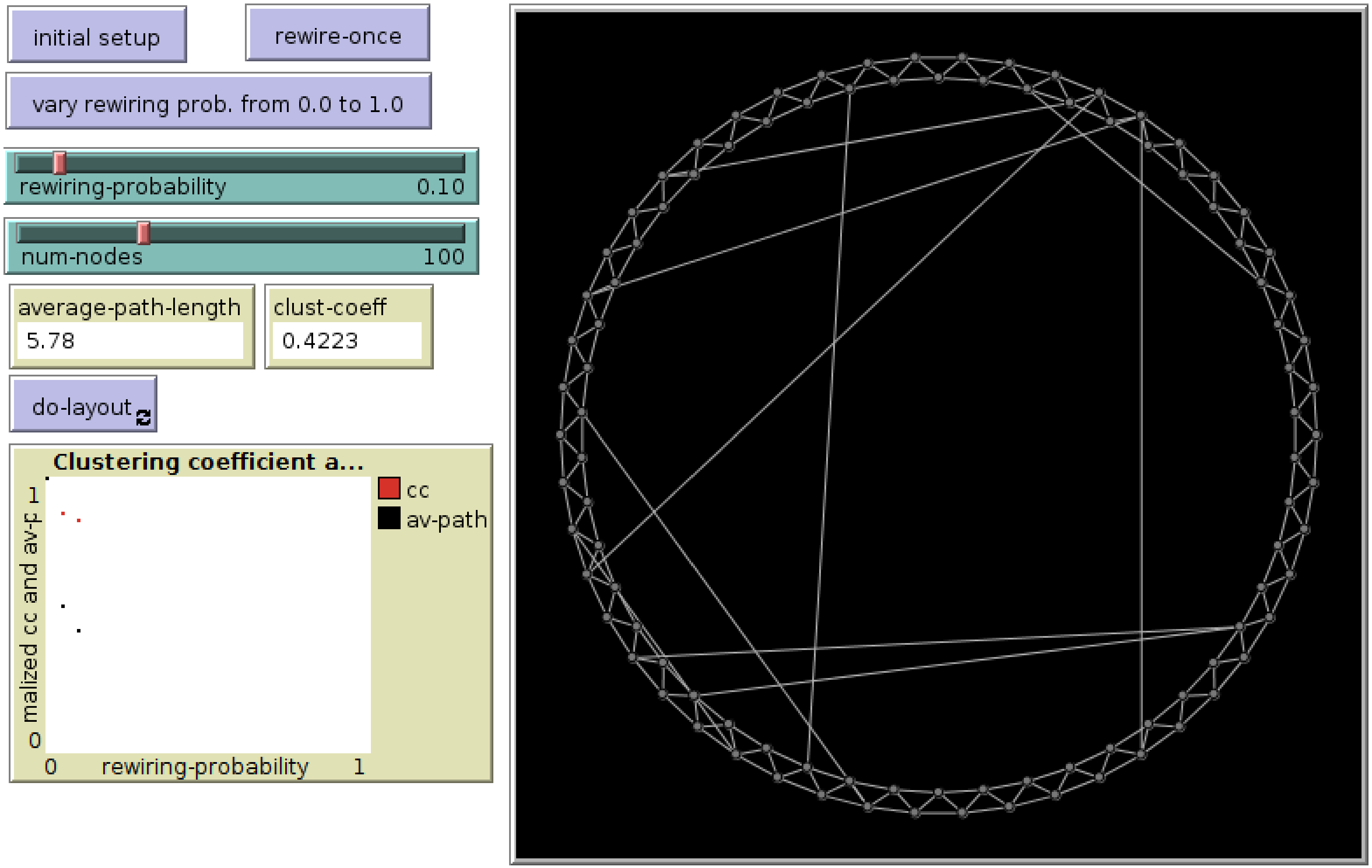
Rewiring allows us to “interpolate” between a regular lattice and a random graph

The Small World Model



Intuition: It takes a lot of randomness to ruin the clustering, but a very small amount to create shortcuts.

NetLogo: $G_{n,p}$ *and* Small-World



SmallWorldWS.nlogo

Small-World: Summary

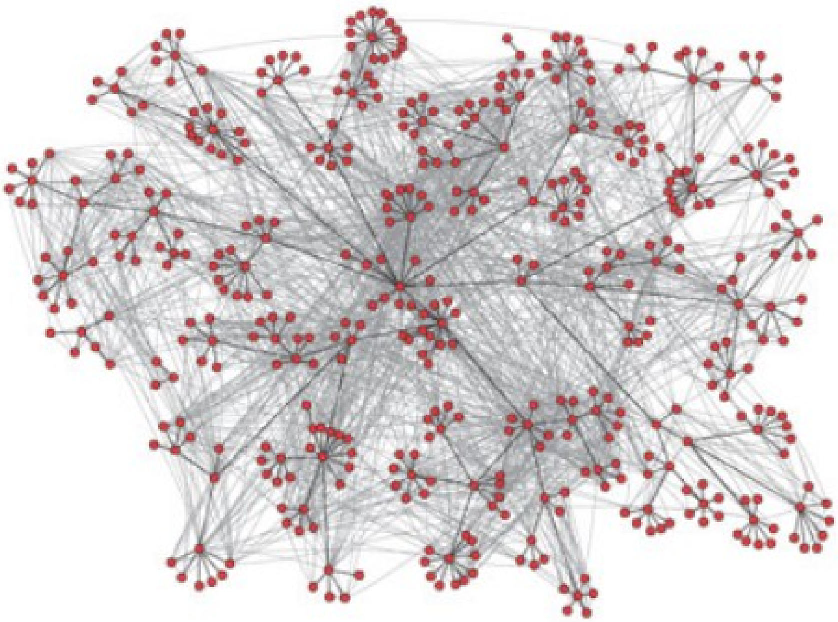
- Could a network with high clustering be at the same time a "small world"?
  - Yes! You don't need more than a few random links
- The Watts-Strogatz Model:
  - Provides insight on the interplay between clustering and being "small-world"
  - Captures the structure of many realistic networks
  - Accounts for the high clustering of real networks ✔
  - Does not lead to the correct degree distribution ✖

We usually call **small world** to networks which exhibit:
• Short avg. path length *(log n)*
• High clustering coefficient

Power Laws and Degree Distributions

Realistic Degree Distribution

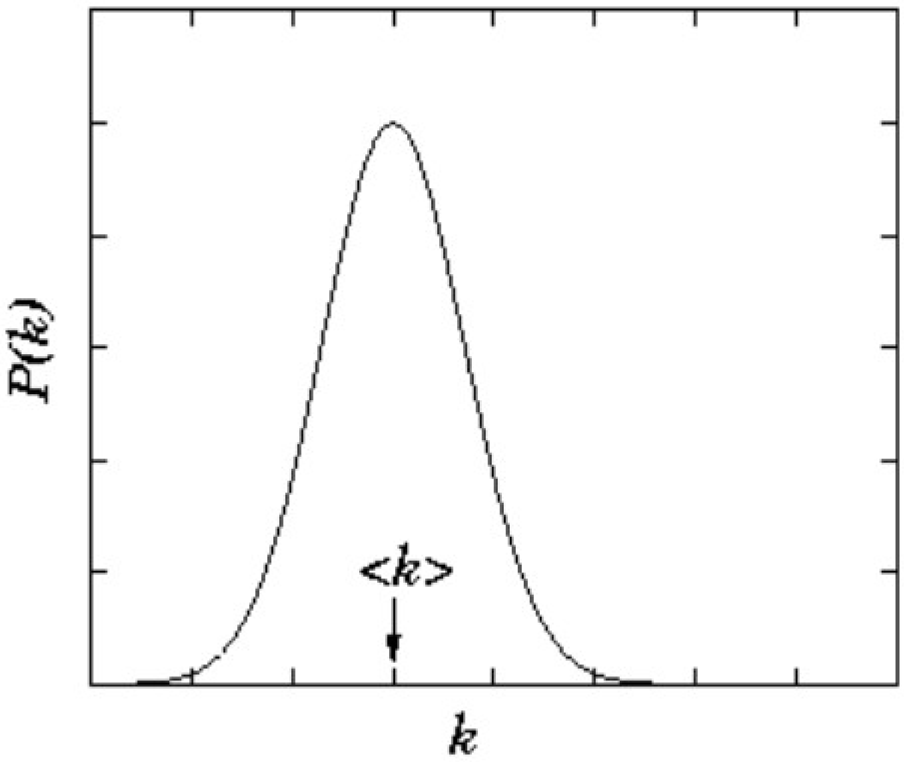
Which interesting graph properties do we observe that need explaining?



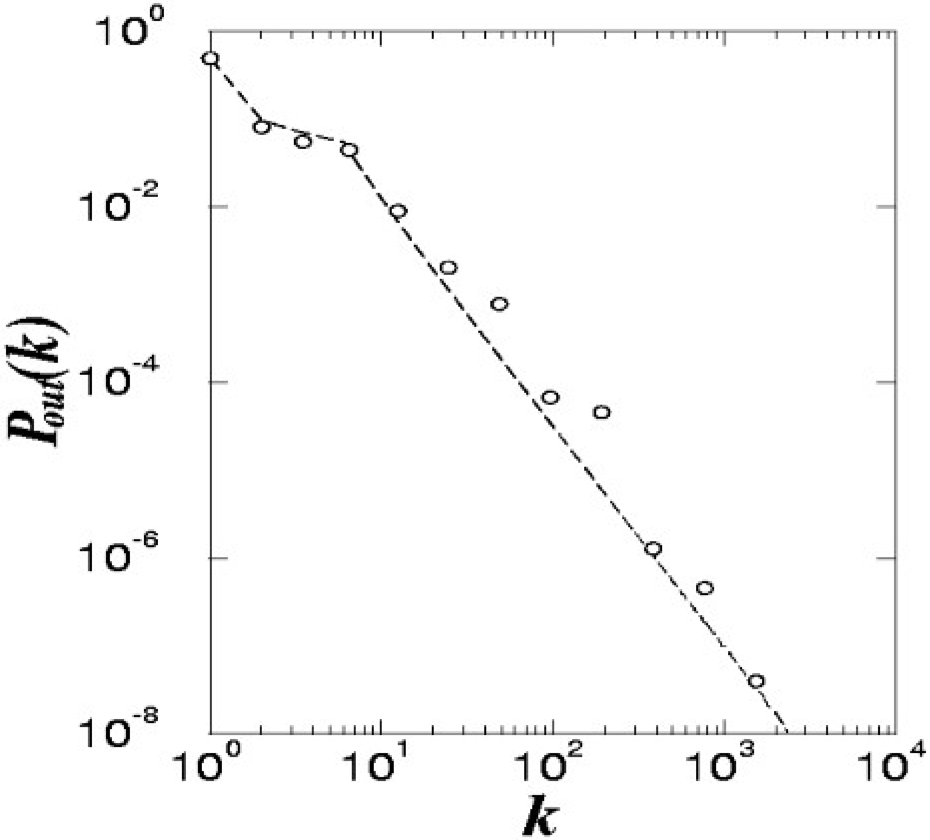
- Small-world model:
  - Avg. Path Length
  - Clustering coefficient
- What about **node degree distribution**?
  - What fraction of nodes has degree ***k*** (as a function of ***k***)?
  - Observation in **real networks**: very often a **power law**: $P(k) \propto k^{-\alpha}$
  - Small-World is similar to $G_{n,p}$: **pronounced peak at k** does not result in realistic distributions...

Realistic Degree Distribution

Expected based on G_{np}

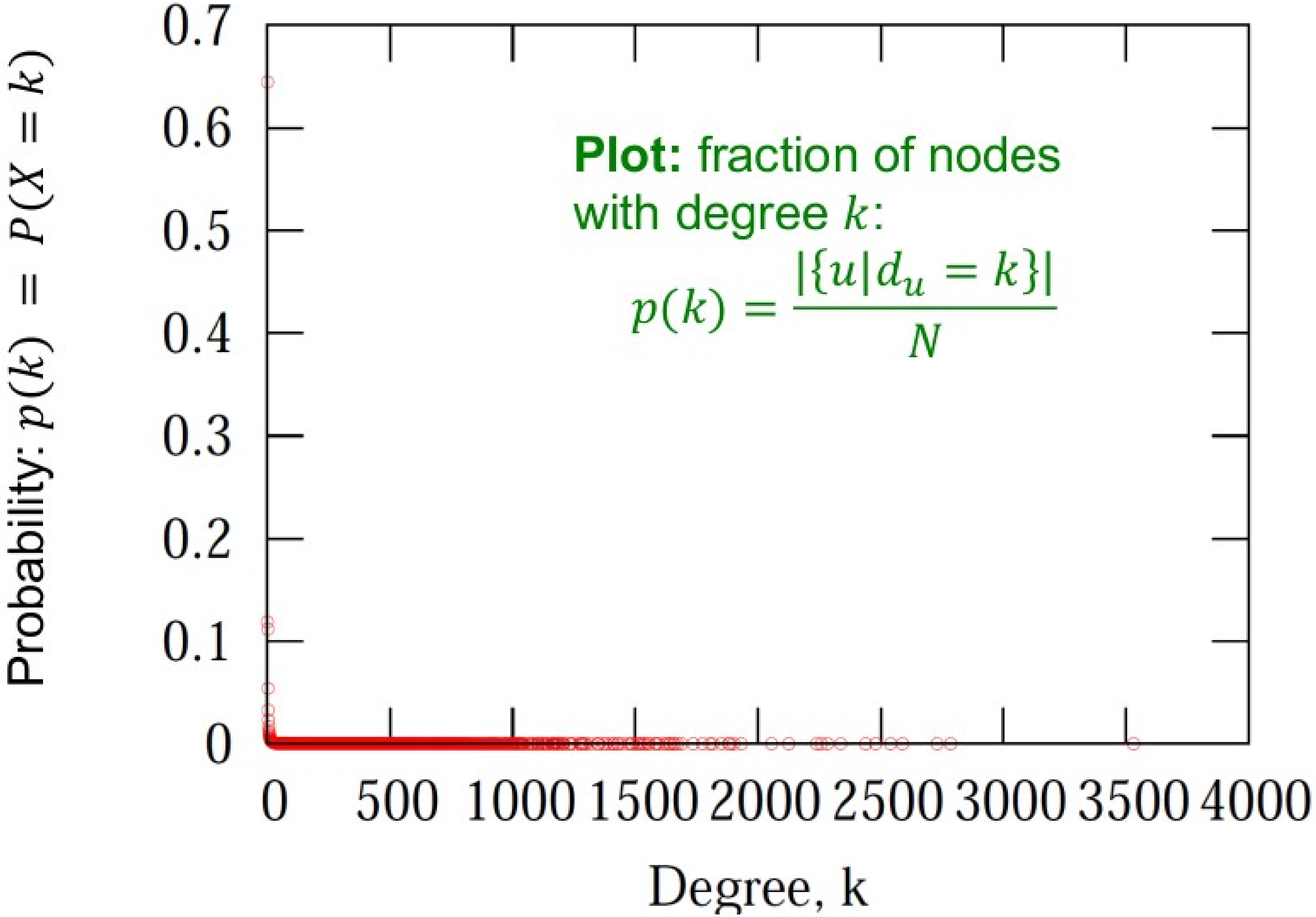


Found in data



$$P(k) \propto k^{-\alpha}$$

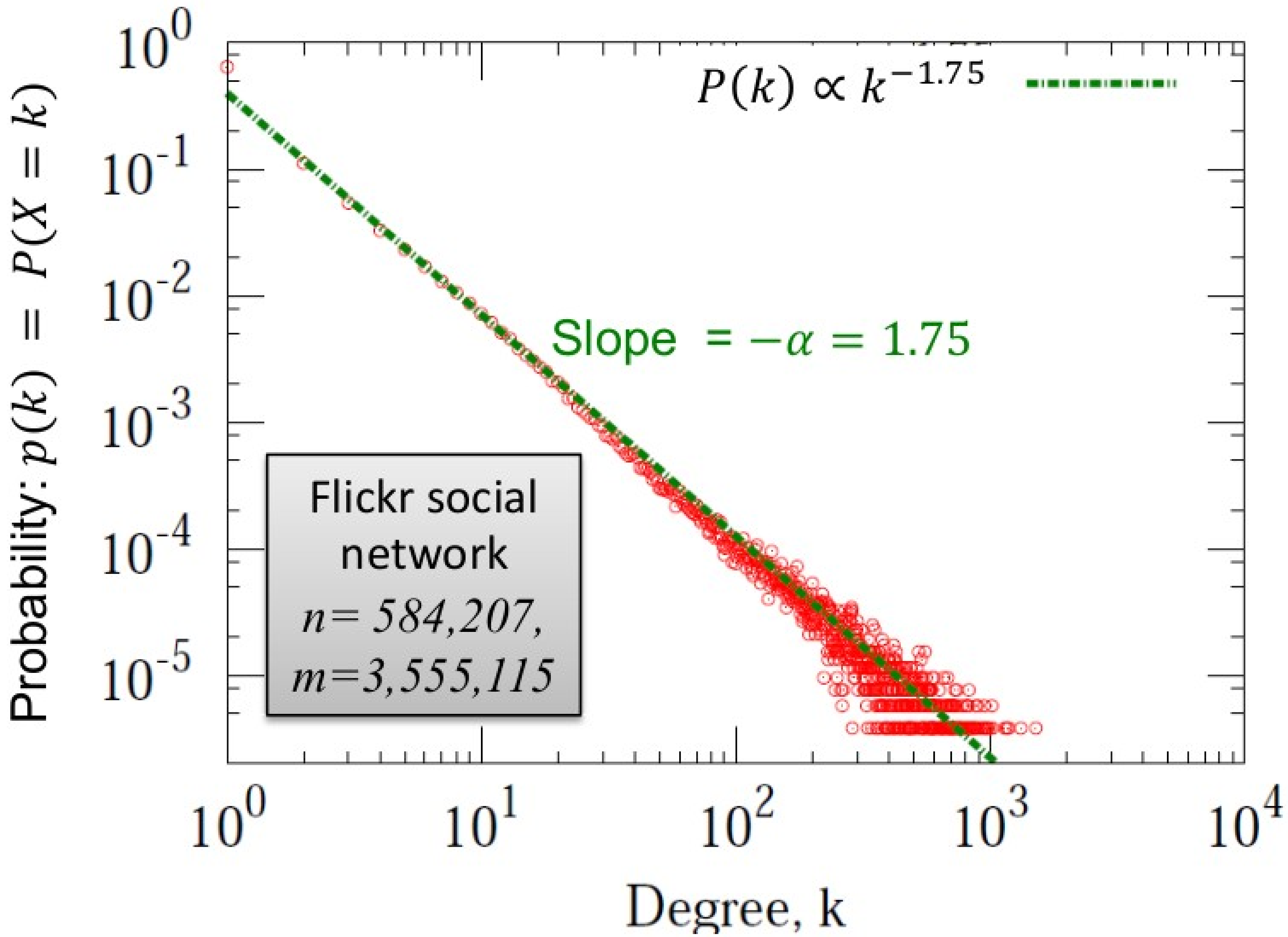
Example: Flickr



Flickr social network
$n= 584{,}207$,
$m=3{,}555{,}115$

[Leskovec et al. KDD '08]

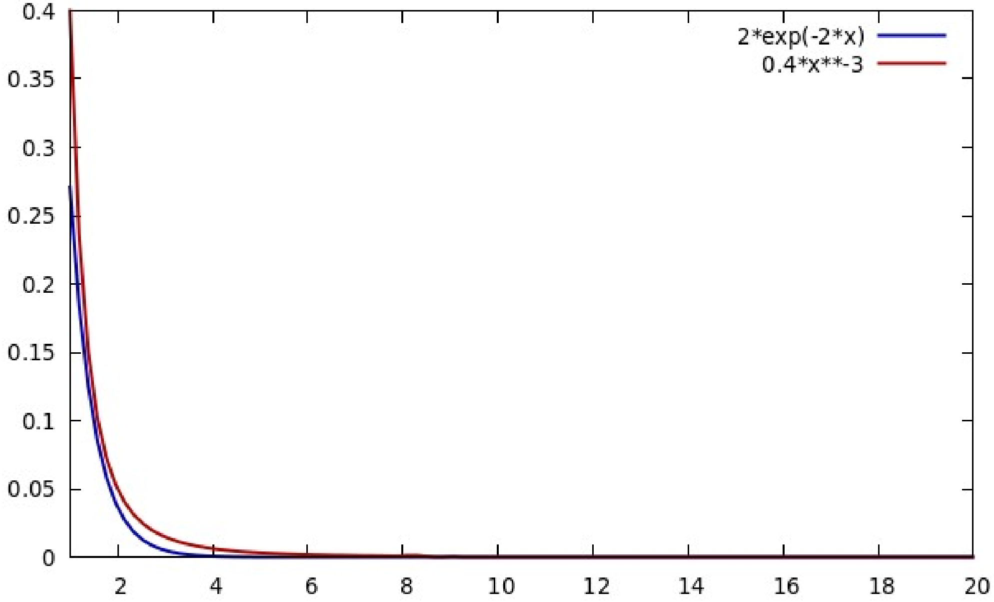
Example: Flickr



Same plot, but now on **log-log** scale

Intermezzo: exponential vs power-law

- How to distinguish:
  - **Exponential**: $P(k) \propto \lambda e^{-\lambda k}$
    vs
  - **Power-Law**: $P(k) \propto k^{-\alpha}$



gnuplot

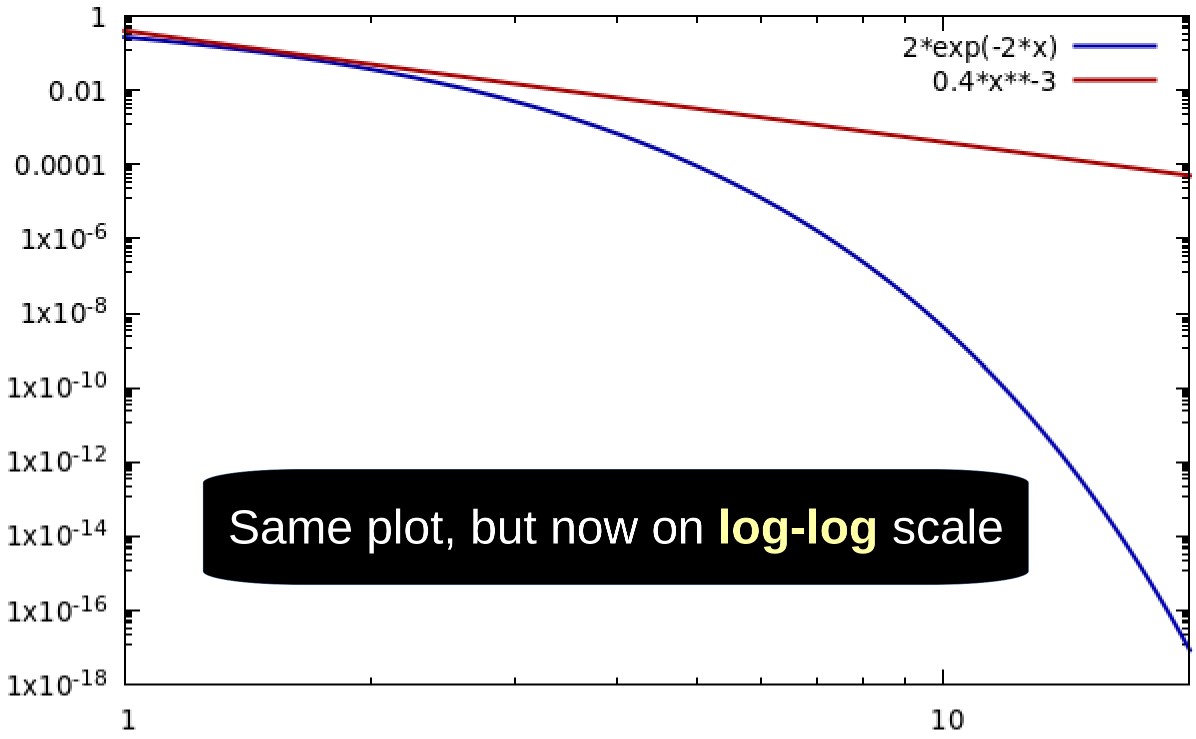
```
plot [1:20] 2*exp(-2*x) lt rgb "#0000aa" lw 2, 0.4*x**-3 lt rgb "#aa0000" lw 2
```

Intermezzo: exponential vs power-law

- **Exponential**: $P(k) \propto \lambda e^{-\lambda k}$

  vs

- **Power-Law**: $P(k) \propto k^{-\alpha}$



If $y = f(x) = x^{-\alpha}$, then
$\log(y) = -\alpha \log(x)$

On a log-log axis
a power law
looks like
a **straight line**
of **slope -α**

gnuplot

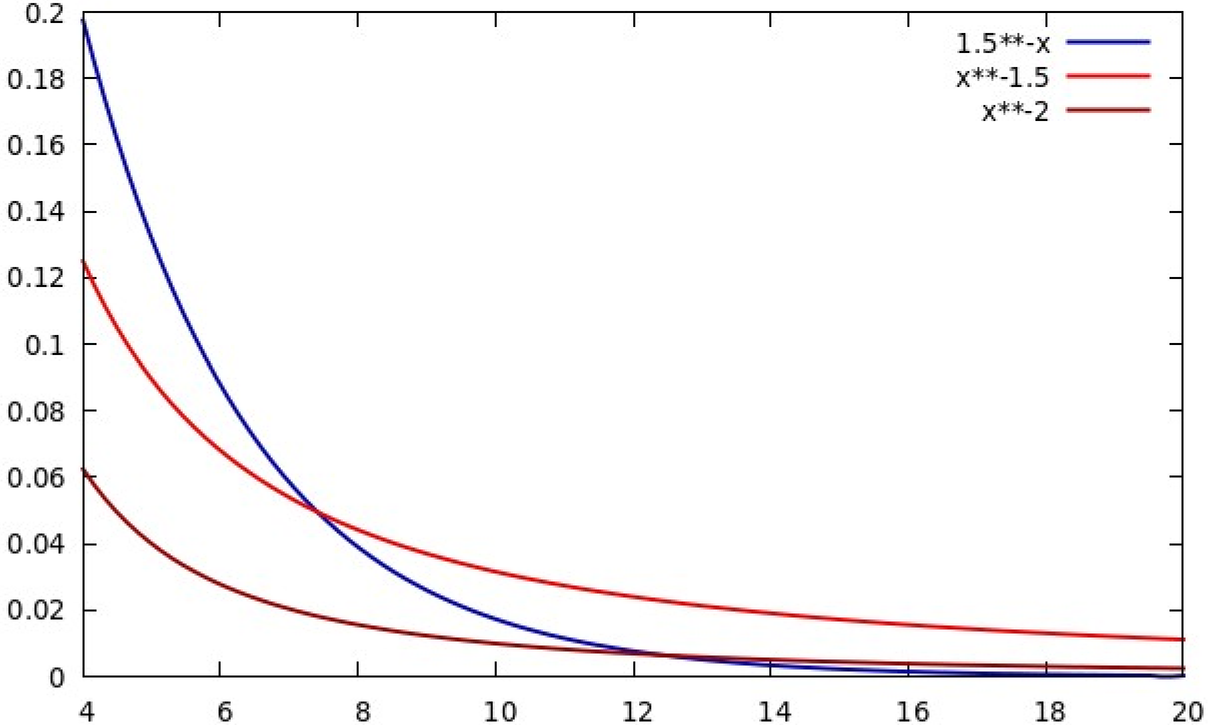
```
set logscale xy
```

Intermezzo: exponential vs power-law

- **Exponential**: $P(k) \propto \lambda e^{-\lambda k}$

  vs

- **Power-Law**: $P(k) \propto k^{-\alpha}$



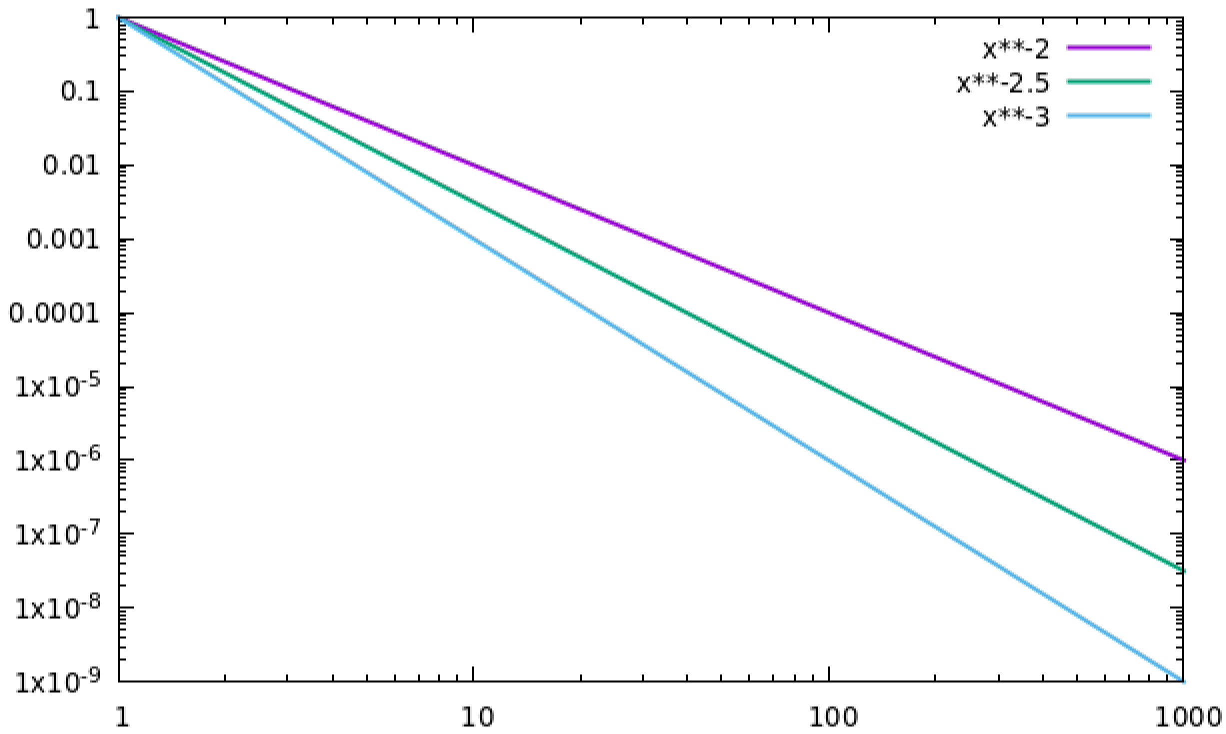
Above a certain ***x*** value, the power law is always higher than the exponential

gnuplot

```
plot [4:20] 1.5**-x, x**-1.5, x**-2
```

Intermezzo: power-law "slope"

- **Power-Law**: $P(k) \propto k^{-\alpha}$



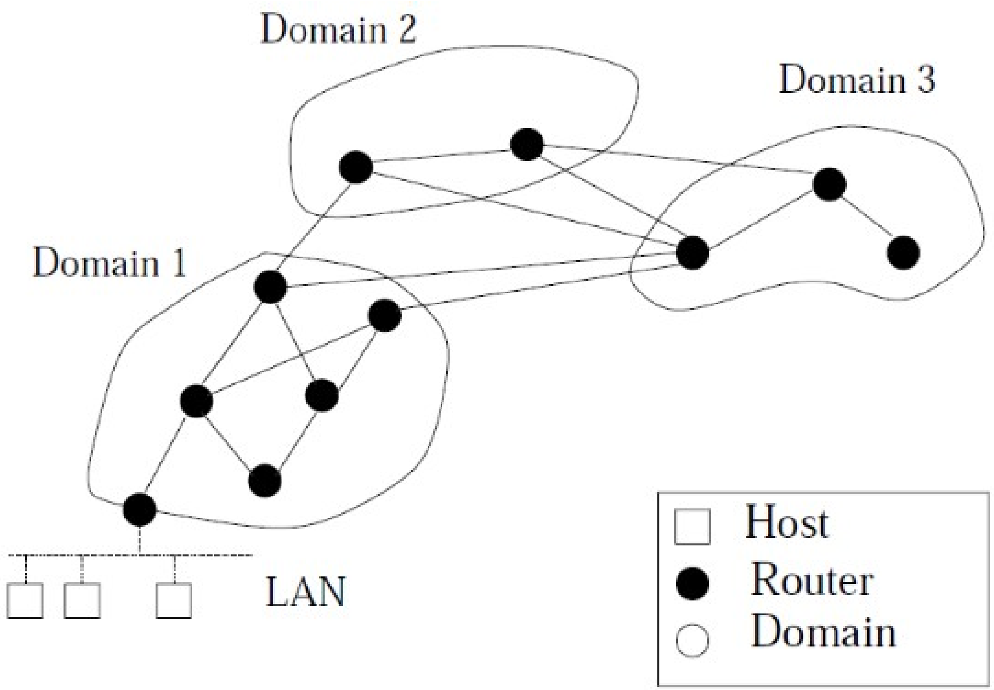
lower alpha (**α**) will mean less pronounced slope

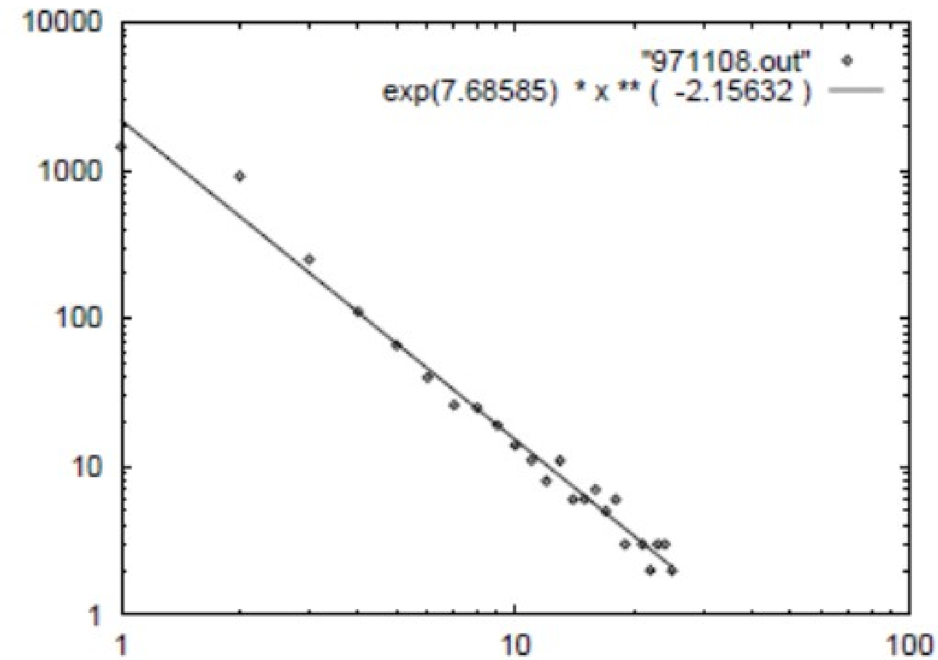
gnuplot

```
plot [1:1000] x**-2 lw 2, x**-2.5 lw 2, x**-3 lw 2
```

Example: Internet Autonomous Systems

- First observed in Internet Autonomous Systems
  [Faloutsos, Faloutsos and Faloutsos, 1999]





Internet domain topology

On Power-Law Relationships of the Internet Topology

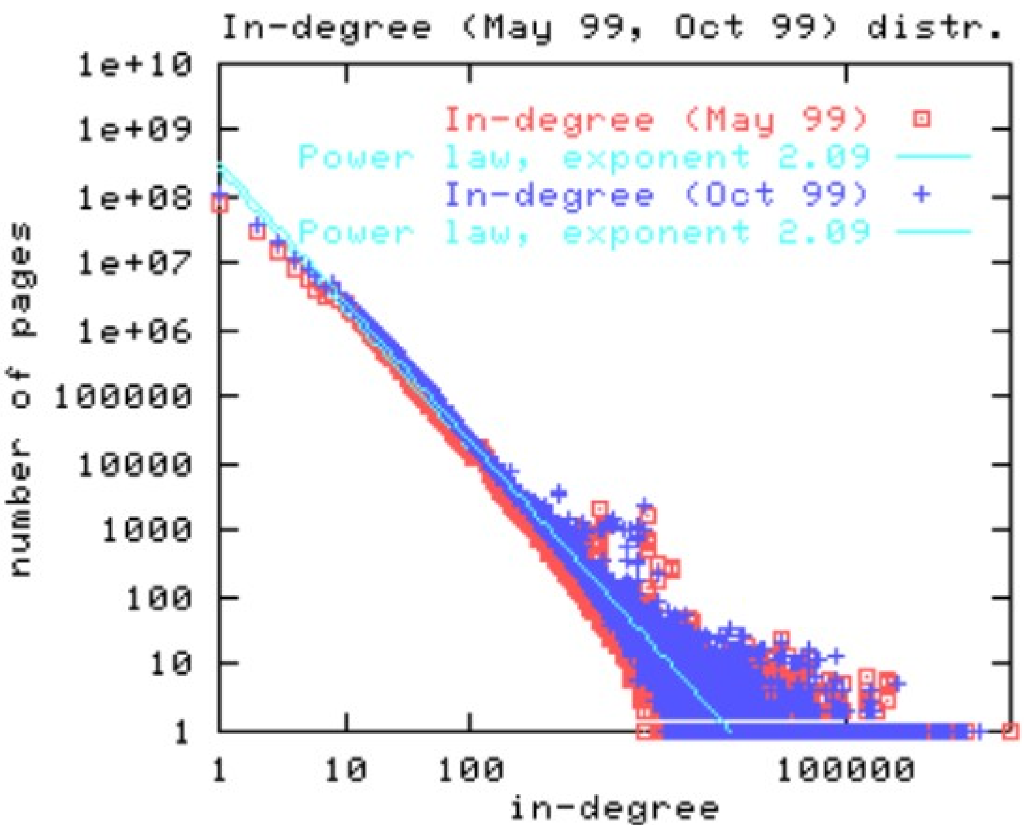
Michalis Faloutsos
U.C. Riverside
Dept. of Comp. Science
michalis@cs.ucr.edu

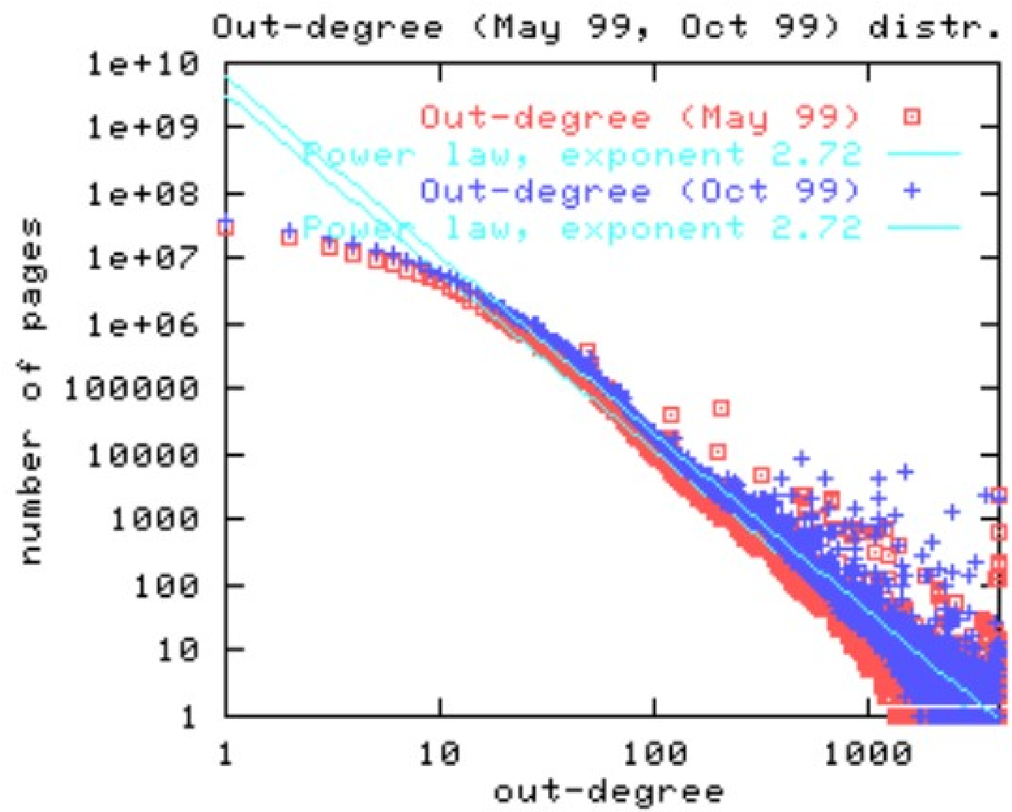
Petros Faloutsos
U. of Toronto
Dept. of Comp. Science
pfal@cs.toronto.edu

Christos Faloutsos *
Carnegie Mellon Univ.
Dept. of Comp. Science
christos@cs.cmu.edu

Example: World Wide Web

[Broder et al., 2000]





Graph structure in the Web

Andrei Broder [a], Ravi Kumar [b,*], Farzin Maghoul [a], Prabhakar Raghavan [b], Sridhar Rajagopalan [b], Raymie Stata [c], Andrew Tomkins [b], Janet Wiener [c]

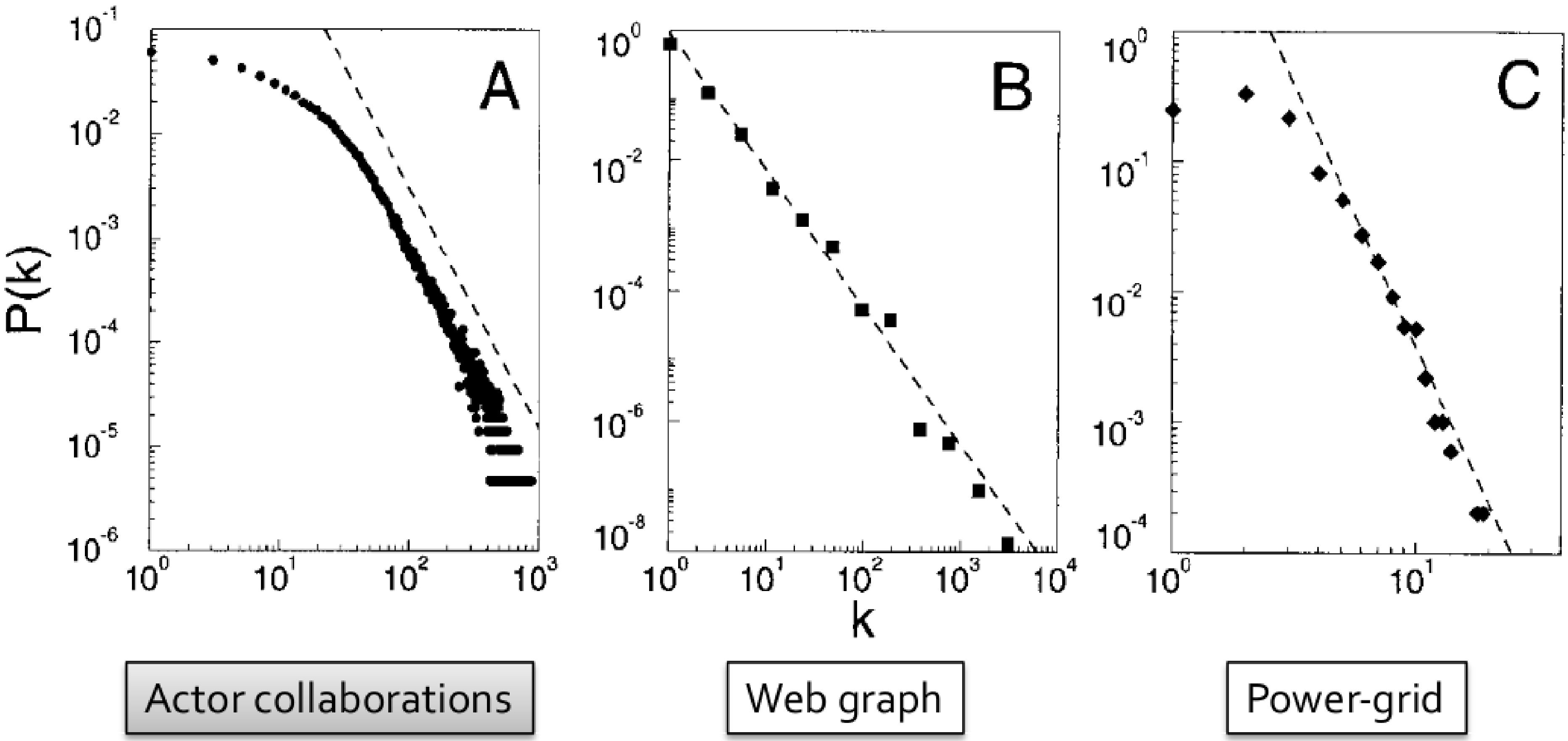
[a] *AltaVista Company, San Mateo, CA, USA*
[b] *IBM Almaden Research Center, San Jose, CA, USA*
[c] *Compaq Systems Research Center, Palo Alto, CA, USA*

Other Examples

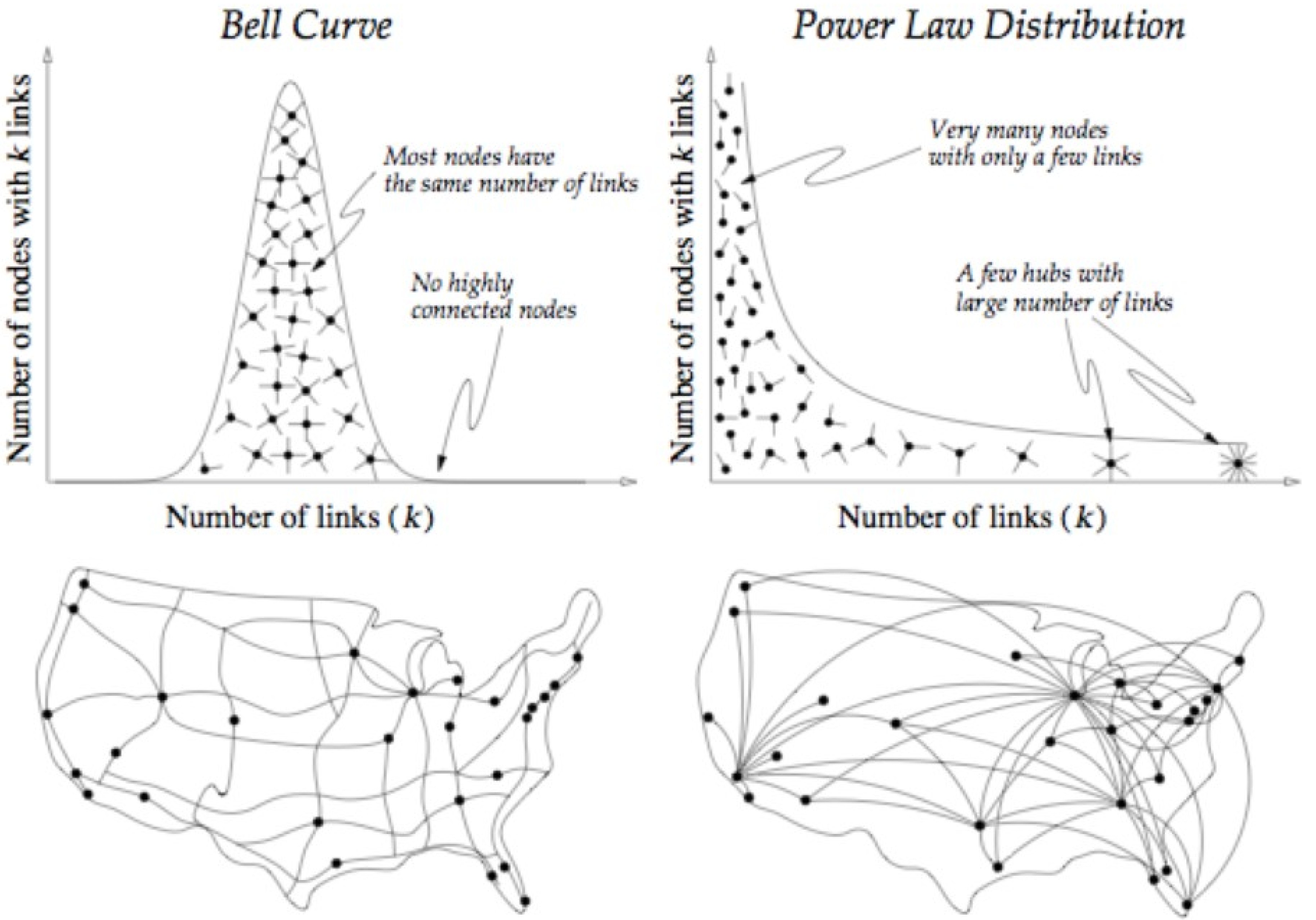
[Barabasi-Albert, 1999]

Emergence of Scaling in Random Networks

Albert-László Barabási* and Réka Albert



Interpreting Power-Laws

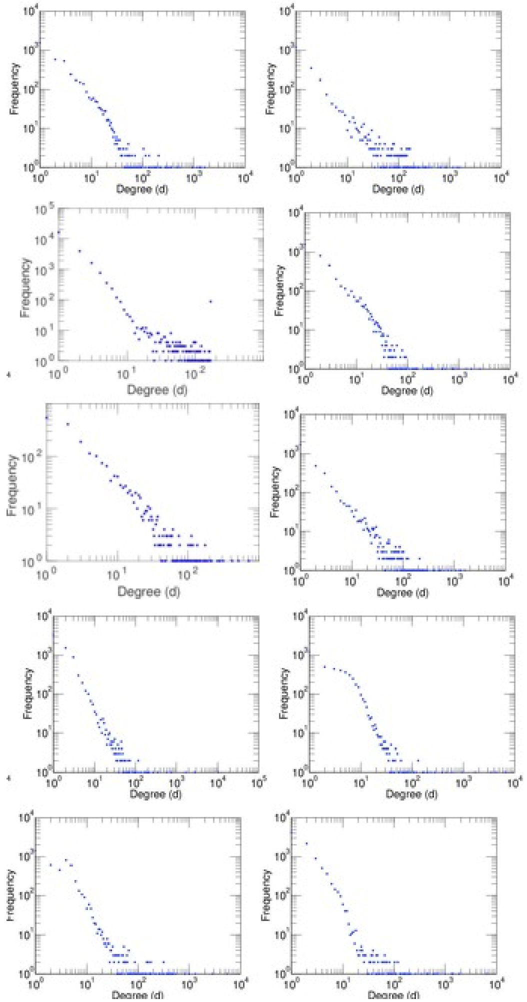


Power-Law Degree Exponent

- Power-law degree exponent is typically:

$$2 < \alpha < 3$$

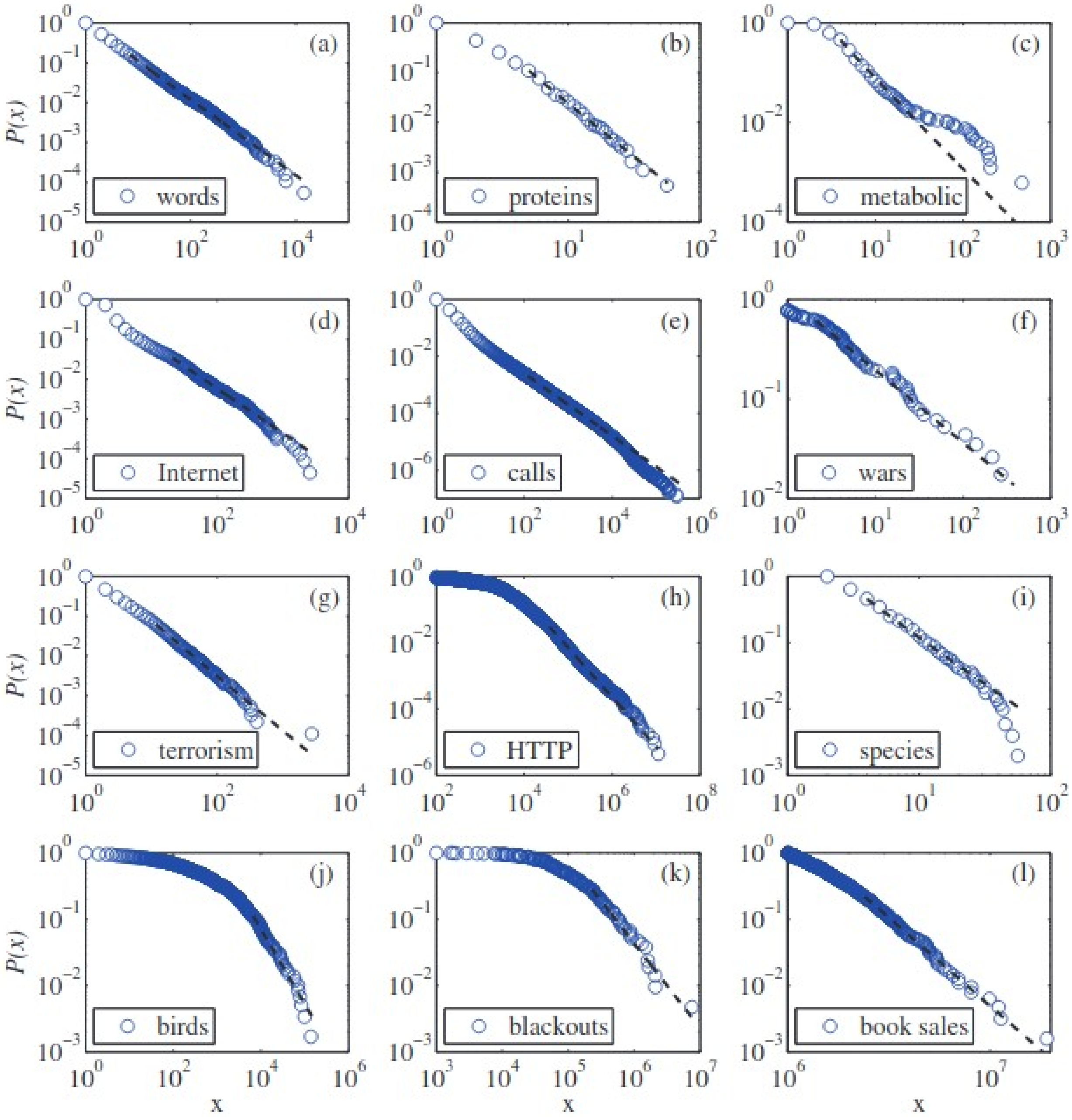
- Examples
  - Web graph:
    - $\boldsymbol{\alpha_{in} = 2.1}$, $\boldsymbol{\alpha_{out} = 2.4}$ [Broder et al. 00]
  - Autonomous systems:
    - $\boldsymbol{\alpha = 2.4}$ [Faloutsos 3 , 99]
  - Actor-collaborations:
    - $\boldsymbol{\alpha = 2.3}$ [Barabasi-Albert 00]
  - Citations to papers:
    - $\boldsymbol{\alpha \approx 3}$ [Redner 98]
  - Online social networks:
    - $\boldsymbol{\alpha \approx 2}$ [Leskovec et al. 07]



Many real world networks are power-law

	exponent α (in/out degree)
film actors	2.3
telephone call graph	2.1
email networks	1.5/2.0
sexual contacts	3.2
WWW	2.3/2.7
internet	2.5
peer-to-peer	2.1
metabolic network	2.2
protein interactions	2.4

Power Laws are Everywhere

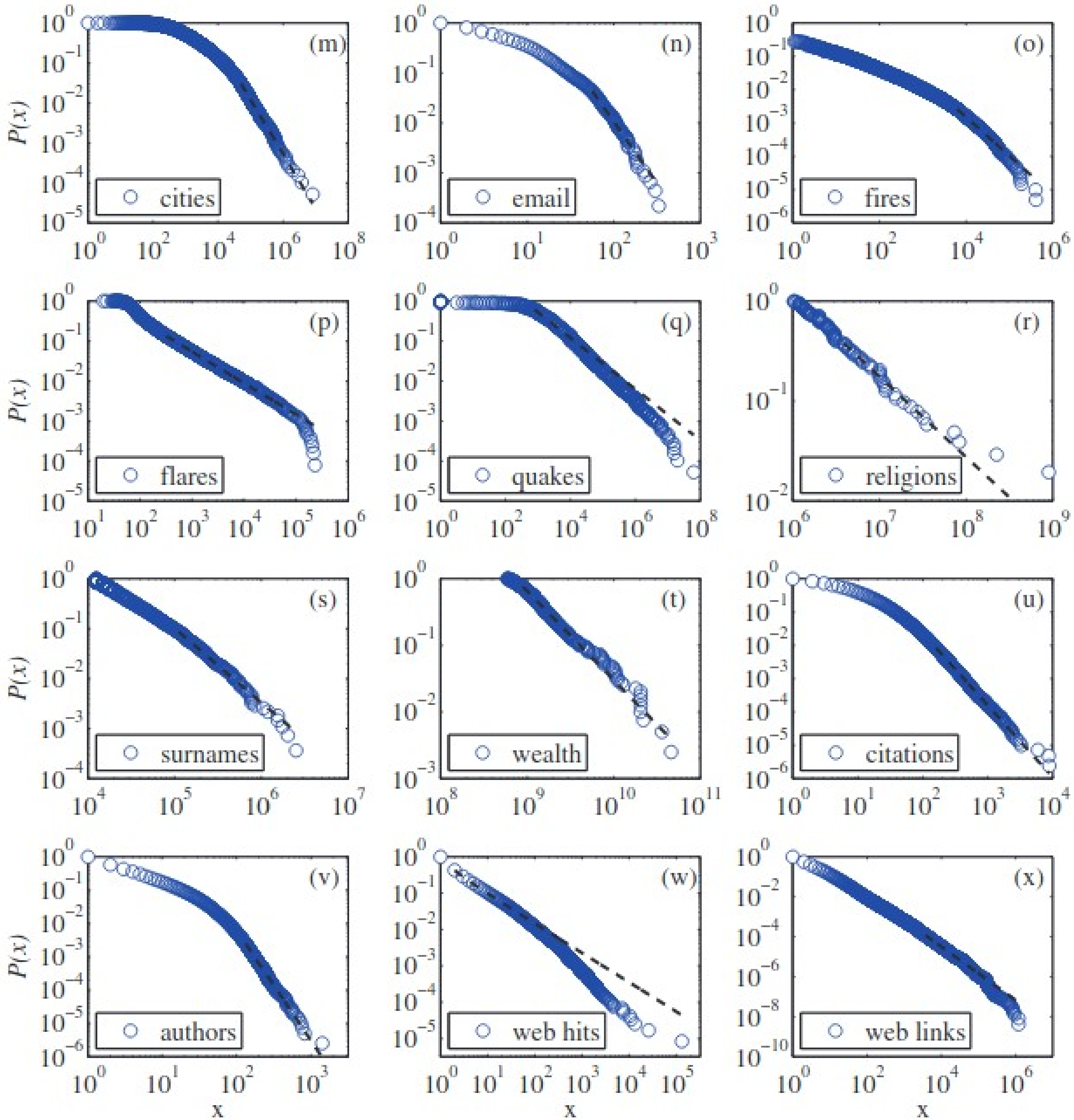


Power-Law Distributions in Empirical Data*

Aaron Clauset†
Cosma Rohilla Shalizi‡
M. E. J. Newman§

[Clauset, Shalizi, Newman, 2009]

Power Laws are Everywhere



Power-Law Distributions in Empirical Data*

Aaron Clauset†
Cosma Rohilla Shalizi‡
M. E. J. Newman§

[Clauset, Shalizi, Newman, 2009]

Some exponents for real world data

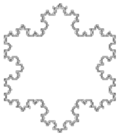
	x_{min}	exponent α
frequency of use of words	1	2.20
number of citations to papers	100	3.04
number of hits on web sites	1	2.40
copies of books sold in the US	2 000 000	3.51
telephone calls received	10	2.22
magnitude of earthquakes	3.8	3.04
diameter of moon craters	0.01	3.14
intensity of solar flares	200	1.83
intensity of wars	3	1.80
net worth of Americans	$600m	2.09
frequency of family names	10 000	1.94
population of US cities	40 000	2.30

Not everyone likes Power Laws 😊

CMU grad-students at the G20 meeting in Pittsburgh in Sept 2009

Scale Free Networks

- Networks with a **power-law** tail in their degree distribution are often called **“scale-free networks”**

- Where does the term scale-free com from?
  - **Scale invariance:** there is no characteristic scale
    - means laws do not change if scales of length, energy, or other variables, are multiplied by a common factor
  - **Scale free function:** $f(\lambda x) = C(\lambda)\, f(x) \propto f(x)$

    $C(\lambda)$ depends only on λ
    - Power-law: $f(x) = ax^{-\alpha}$

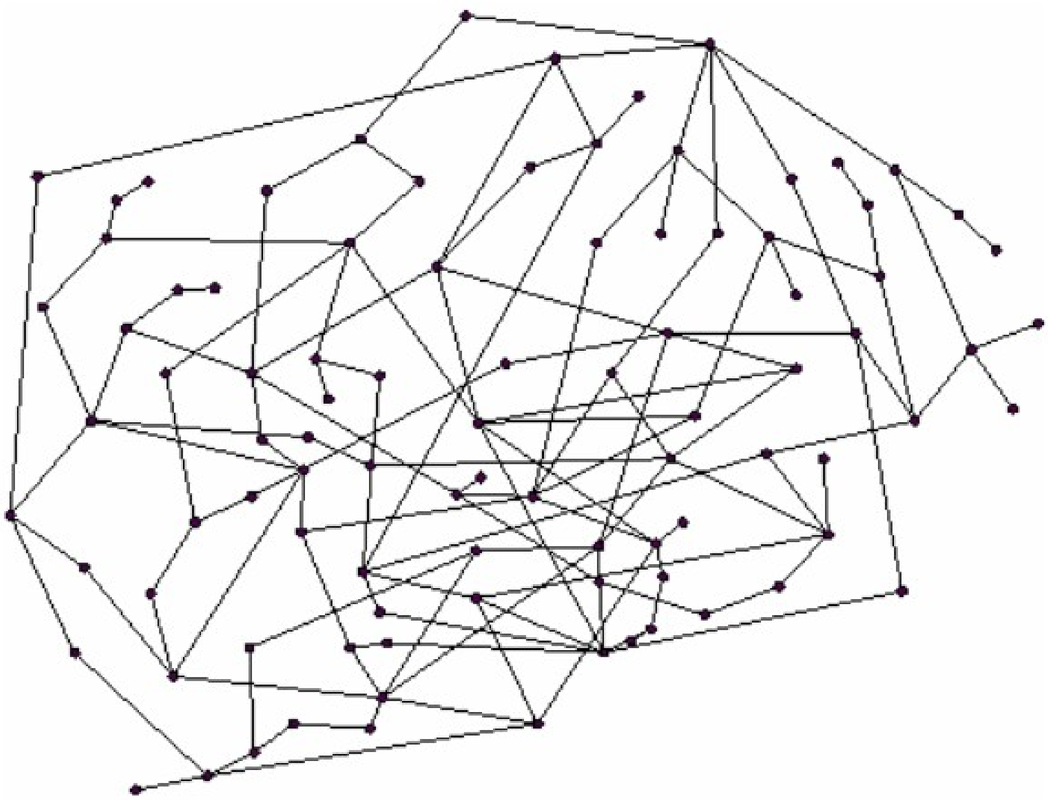
$$f(\lambda x) = a(\lambda x)^{-\alpha} = \lambda^{-\alpha}(ax^{-\alpha}) = \lambda^{-\alpha} f(x) \propto f(x)$$

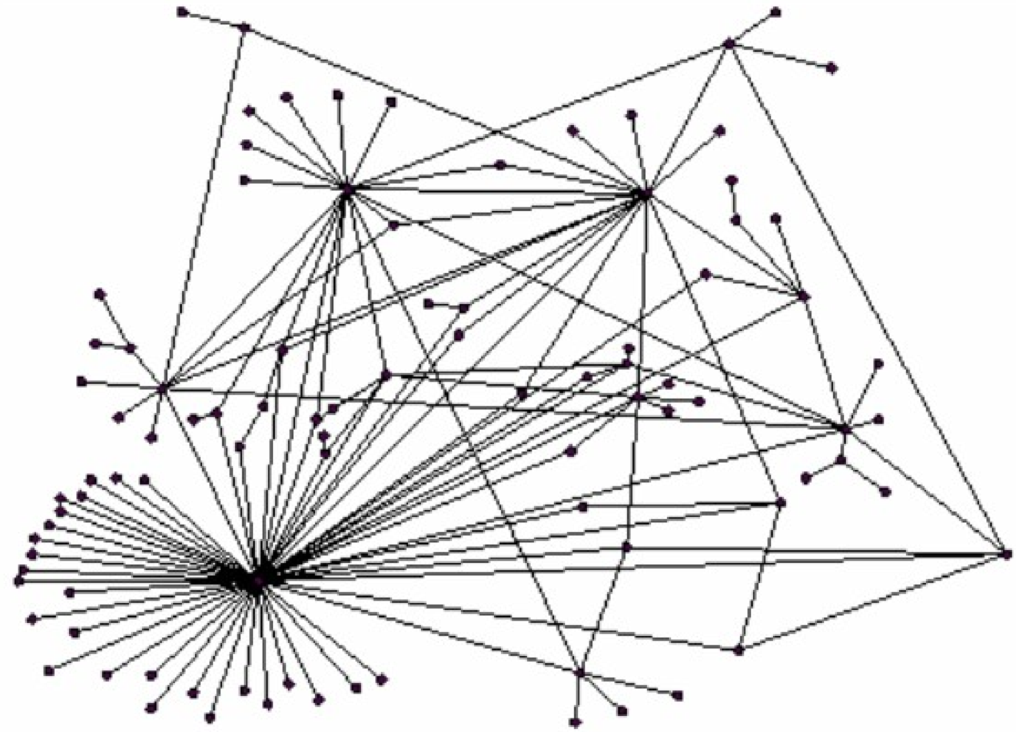
Log() or Exp() are not scale free

$f(\lambda x) = log(\lambda x) = log(\lambda) + log(x) = log(\lambda) + f(x)$

$f(\lambda x) = exp(\lambda x) = exp(x)^{\lambda} = f(x)^{\lambda}$

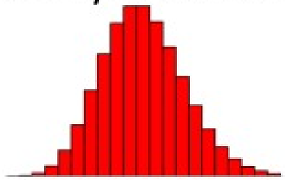
Random vs Scale Free





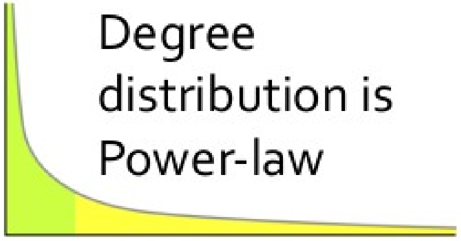
Random network

(Erdos-Renyi random graph)



Degree distribution is Binomial

Scale-free (power-law) network



Degree distribution is Power-law

Preferential Attachment Model

Rich Get Richer

- **New nodes are more likely to link to nodes that already have high degree**
- Herbert Simon's result:
  - Power-laws arise from *"Rich get richer"* (cumulative advantage)

ON A CLASS OF SKEW DISTRIBUTION FUNCTIONS

By HERBERT A. SIMON†

Carnegie Institute of Technology

- Examples:
  - **Citations** *[de Solla Price '65]*: New citations to a paper are proportional to the number it already has
    - Herding: If a lot of people cite a paper, then it must be good, and therefore I should cite it too

Networks of Scientific Papers

The pattern of bibliographic references indicates the nature of the scientific research front.

Derek J. de Solla Price

  - **Sociology: Matthew effect** (http://en.wikipedia.org/wiki/Matthew_effect)
    - "For whoever has will be given more, and they will have an abundance. Whoever does not have, even what they have will be taken from them."
    - Eminent scientists often get more credit than a comparatively unknown researcher, even if their work is similar

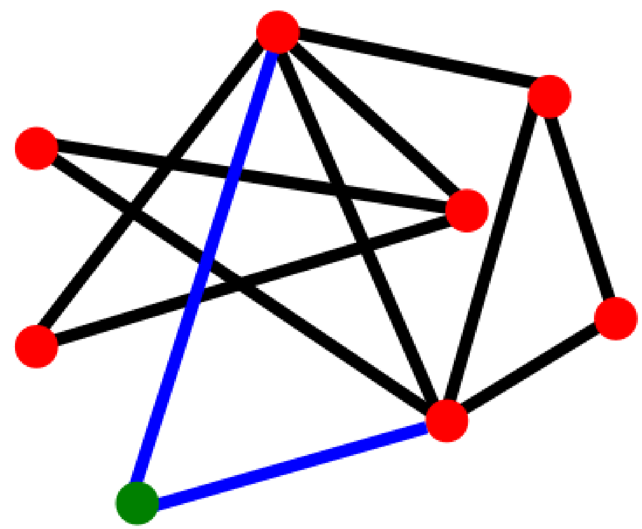
Model: Preferential Attachment

- **Preferential attachment:**
  [Barabasi-Albert '99] **(Barabasi-Albert model)**

  Emergence of Scaling in Random Networks
  Albert-László Barabási* and Réka Albert

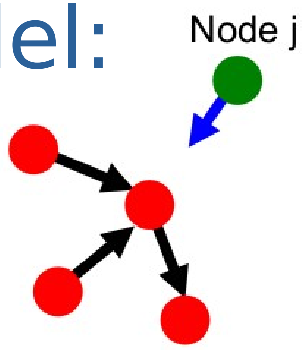
  - Nodes arrive in order **1,2,...,n**
  - At step $\boldsymbol{j}$, let $\boldsymbol{d_i}$ be the degree of a previous node $\boldsymbol{i}$
  - A new node $\boldsymbol{j}$ arrives and creates $\boldsymbol{m}$ out-links
  - Probability of $\boldsymbol{j}$ linking to a previous node $\boldsymbol{i}$ is proportional to degree $\boldsymbol{d_i}$ *of node* $\boldsymbol{i}$

$$P(j \rightarrow i) = \frac{d_i}{\sum_k d_k}$$



Results for Simple Model

- We analyze the following **simple** model:
  - Nodes arrive in order *1,2,3, … , n*
  - When *node j* is created it makes a **single out-link** to an earlier node ***i*** chosen:



    - **1)** With prob. $\boldsymbol{p}$, $\boldsymbol{j}$ links to $\boldsymbol{i}$ chosen **uniformly at random** (from among all earlier nodes)
    - **2)** With prob. $\boldsymbol{1-p}$, node *j* chooses *i* uniformly at random & links **to a random node *v* that *i* points to**
      - **This is same as saying:** With prob. $\boldsymbol{1-p}$, node $\boldsymbol{j}$ links to node $\boldsymbol{v}$ with prob. proportional to $\boldsymbol{d_v}$ (the in-degree of $\boldsymbol{v}$)
    - Our graph is **directed**: every node has out-degree 1

Results for Simple Model

- **Claim:** The described model generates networks where the fraction of nodes with **in-degree *k*** scales as:

$$P(d_i = k) \propto k^{-(1+\frac{1}{q})}$$

where q=1-p

So we get power-law degree distribution with exponent:

$$\alpha = 1 + \frac{1}{1-p}$$

The model gives a **power-law**

Preferential Attachment: The Good

- Preferential attachment gives **power-law** in-degrees!

- Intuitively reasonable process

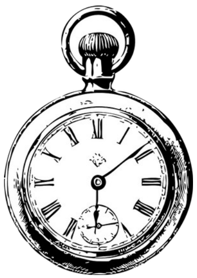
- Can **tune** model parameter p to get the observed exponent
  - On the web, ***P[node has in-degree k] ~ $k^{-2.1}$***
  - *$2.1 = 1+1/(1-p) \rightarrow$* ***$p \sim 0.1$***

$p = 0 \rightarrow P(d_i = k) \sim k^{-2}$

$p = 0.5 \rightarrow P(d_i = k) \sim k^{-3}$

Preferential Attachment: The Bad

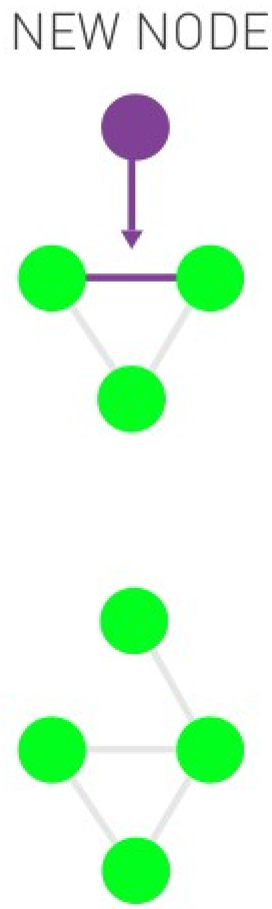
- Preferential attachment is **not so good at predicting network structure**
  - **Age-degree correlation**
    - Node degree is proportional to its age
    - Possible Solution: Node fitness (virtual degree)
  - **Links among high degree nodes:**
    - On the web nodes sometimes avoid linking to each other



- **Further questions:**
  - What is a reasonable model for **how people sample network nodes and link to them**?

Origins of Preferential Attachment

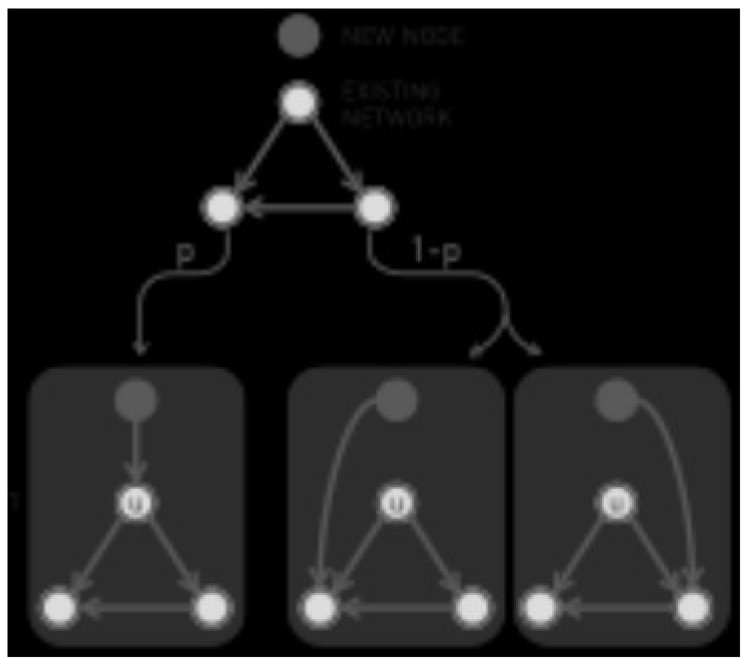
- **Link Selection Model:** perhaps the simplest example of a local or random mechanism capable of generating preferential attachment
  - **Growth:** At each time step we add a new node to the network
  - **Link selection:** We select a link at random and connect the new node to one of the nodes at the two ends of the selected link



- This simple mechanism generates **preferential attachment**
  - Why? Because nodes are picked with probability proportional to their number of edges

Origins of Preferential Attachment

- **Copying Model:**
  - (a) **Random Connection:** with prob. **p** the new node links to random node *v*
  - (b) **Copying:** With prob. $1 - p$ randomly choose an outgoing link of node *v* and connect the new node to the selected link's target
    - The new node *"copies"* one of the links of an earlier node



Origins of Preferential Attachment

- Analysis of the **copying model**:
  - **(a)** the probability of selecting a node is *1/N*
  - **(b)** is equivalent to selecting a node linked to a randomly selected link. The probability of selecting a degree-*k* node through the copying process of step (b) is *k/2E* for undirected networks
  - Again, the likelihood that the new node will connect to a degree-*k* node follows preferential attachment

- Examples:
  - **Social networks:** Copy your friend's friends.
  - **Citation Networks:** Copy references from papers we read
  - **Protein interaction networks:** gene duplication

Many models lead to power-laws

- **Copying mechanism** (directed network)
  - Select a node and an edge of this node
  - Attach to the endpoint of this edge
- **Walking on a network** (directed network)
  - The new node connects to a node, then to every first, second, ... neighbor of this node
- **Attaching to edges**
  - Select an edge and attach to both endpoints of this edge
- **Node duplication**
  - Duplicate a node with all its edges
  - Randomly prune edges of new node

Distances in Preferential Attachment

$$\bar{h} = \begin{cases} const & \alpha = 2 \\ \dfrac{\log\log n}{\log(\alpha-1)} & 2 < \alpha < 3 \\ \dfrac{\log\ n}{\log\log n} & \alpha = 3 \\ \log n & \alpha > 3 \end{cases}$$

Ultra small world

Small world

Avg. path length

Degree exponent

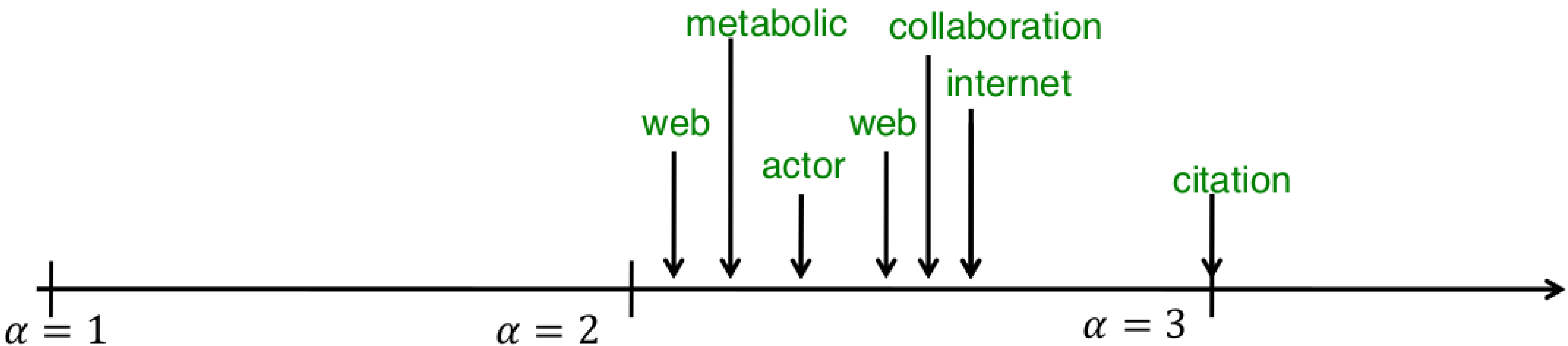
Size of the biggest hub is of order *O(N)*. Most nodes can be connected within two steps, thus the average path length will be independent of the network size *n*.

The avg. path length increases slower than logarithmically with *n*. In G_{np} all nodes have comparable degree, thus most paths will have comparable length. In a scale-free network vast majority of the paths go through the few high degree hubs, reducing the distances between nodes.

Some models produce $\alpha = 3$. This was first derived by Bollobas et al. for the network diameter in the context of a dynamical model, but it holds for the average path length as well.

The second moment of the distribution is finite, thus in many ways the network behaves as a random network. Hence the average path length follows the result that we derived for the random network model earlier.

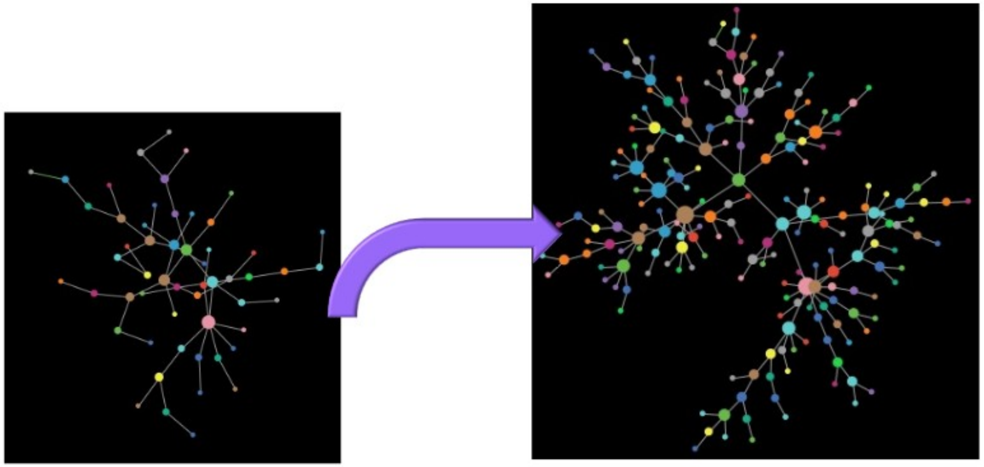
Scale-Free Networks: Overview



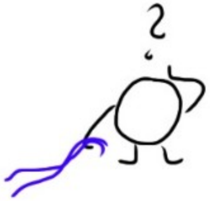
Second moment $\langle k^2 \rangle$ diverges		$\langle k^2 \rangle$ finite
Average $\langle k \rangle$ diverges	Average $\langle k \rangle$ finite	
Ultra small world behavior		Small world
Regime full of anomalies…	The scale-free behavior is relevant	Behaves like a random network

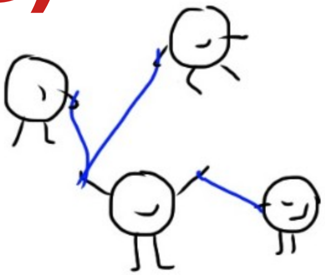
Scale-Free Networks: Ingredients

- Nodes appear over time **(growth)**

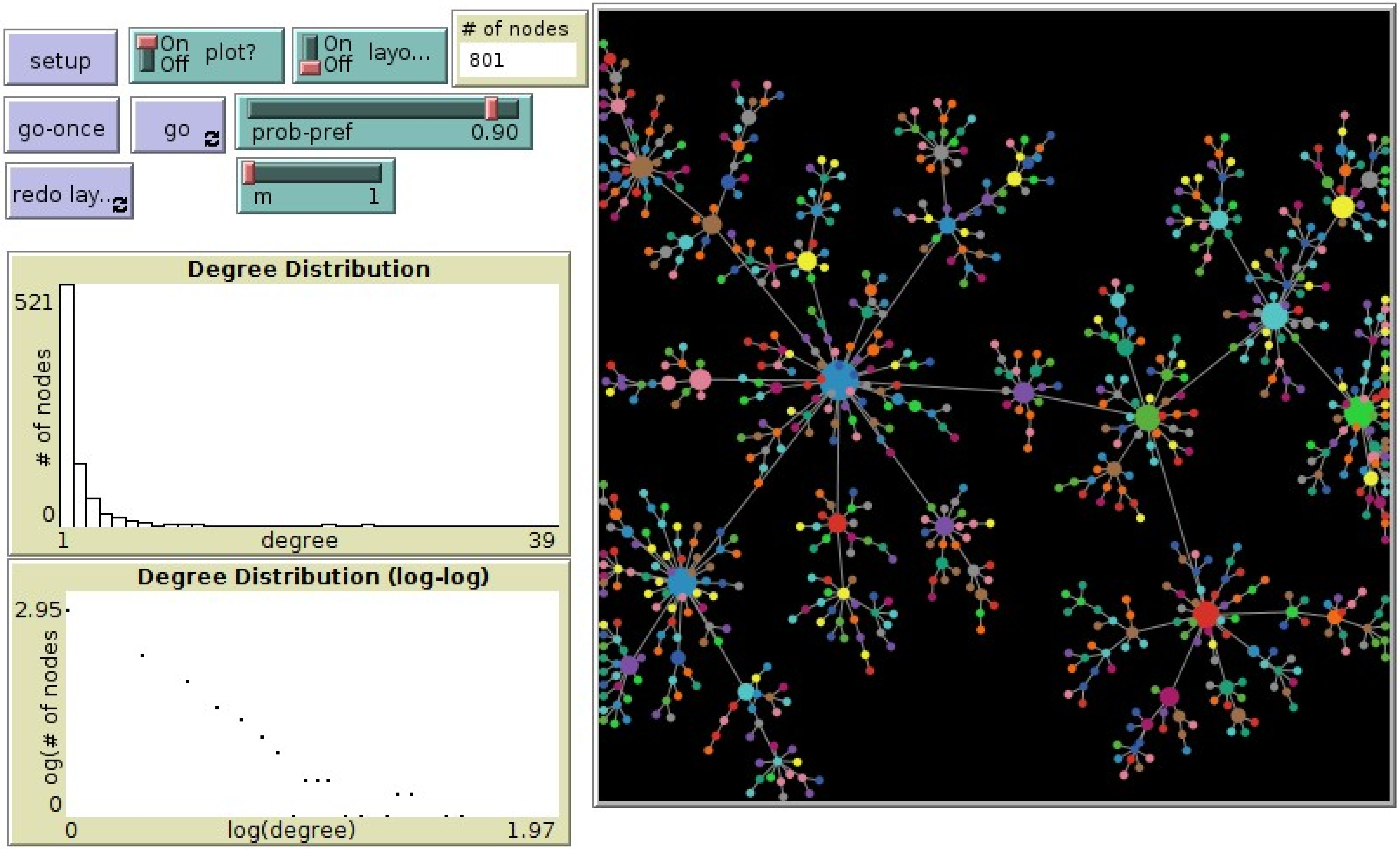


- Nodes prefer to attach to nodes with many connections **(preferential attachment, cumulative advantage)**





NetLogo: Preferential Attachment

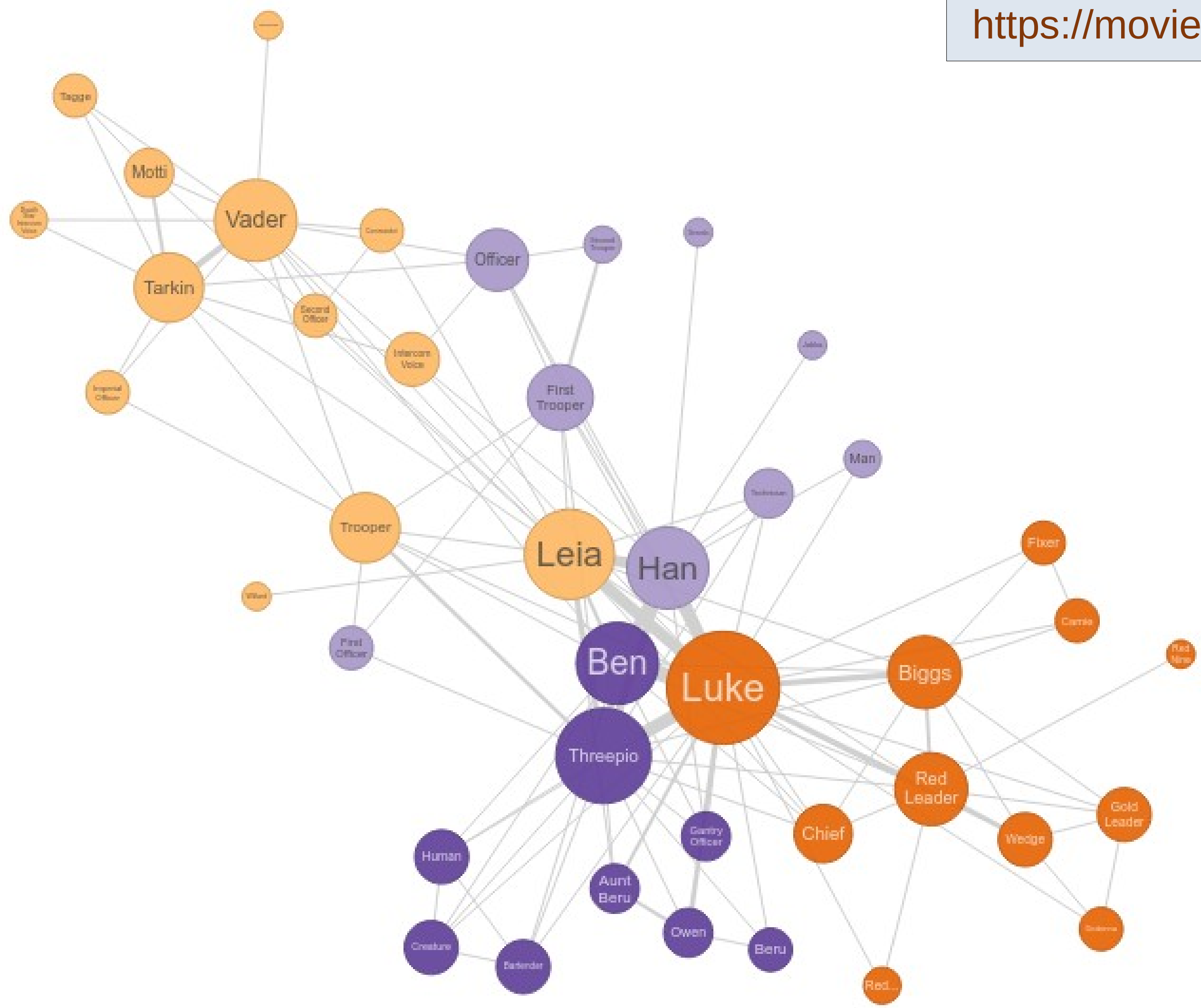


RAndPrefAttachment.nlogo

Node Centrality

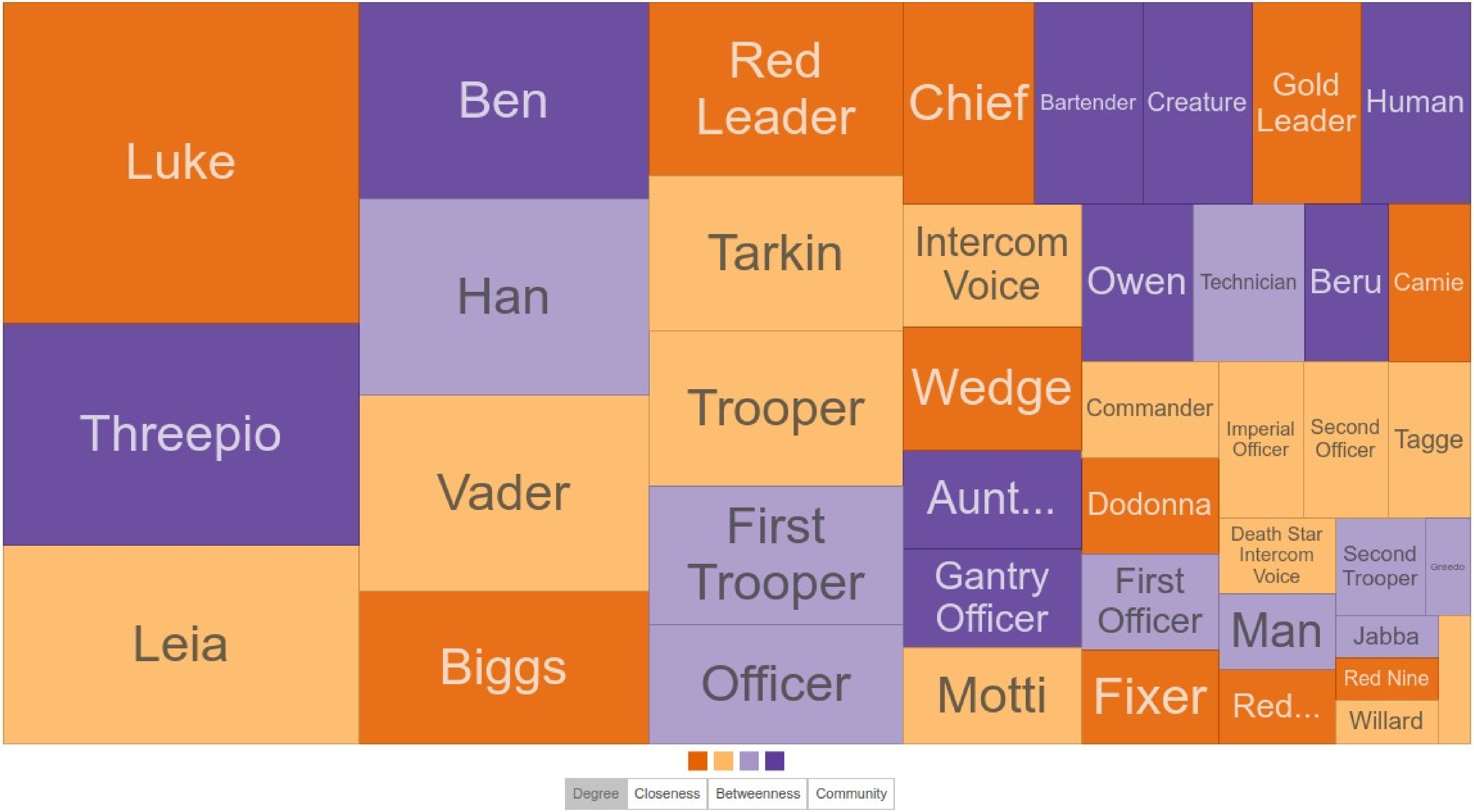
Star Wars IV Network

https://moviegalaxies.com/



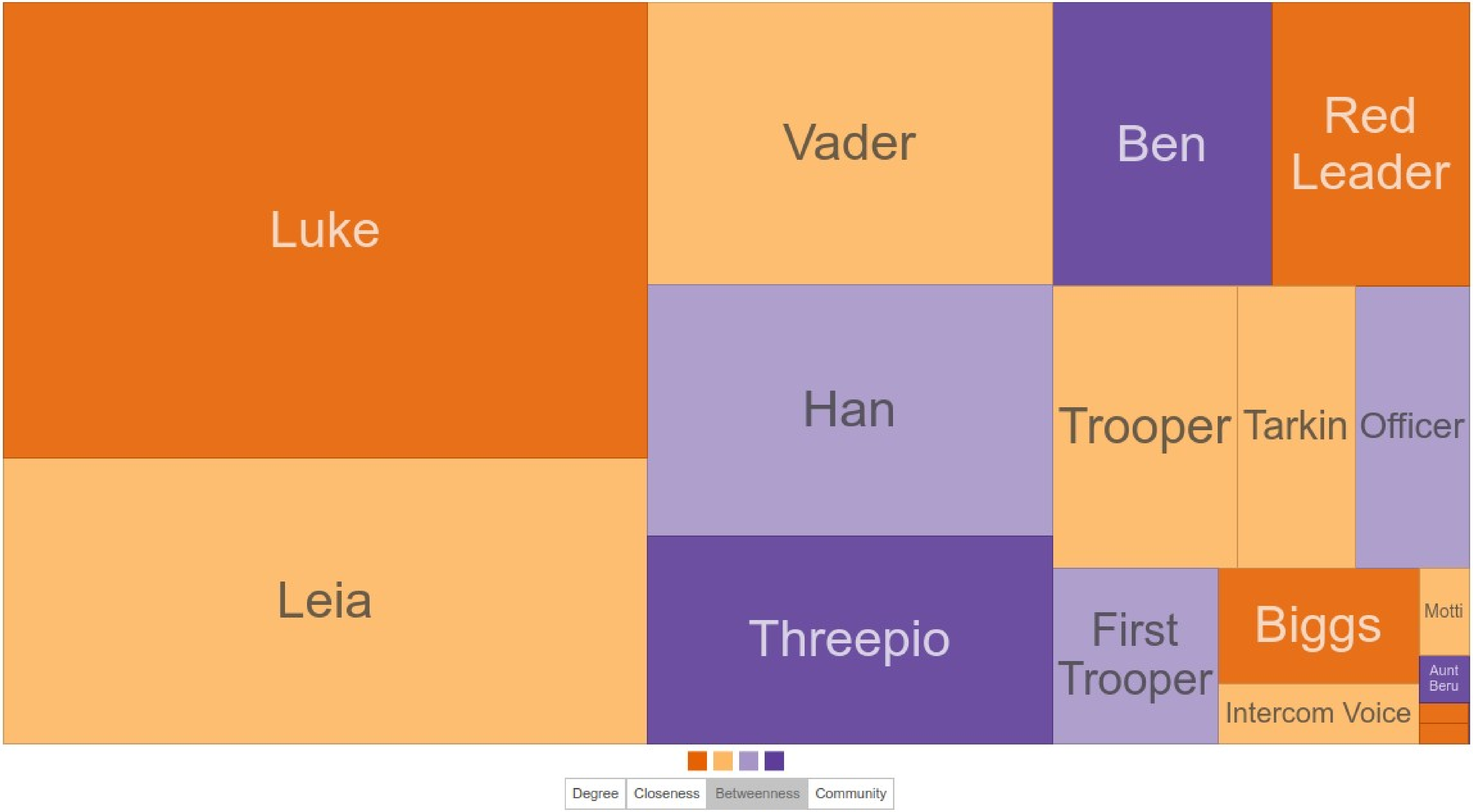
Are all nodes “equal”? How to measure their importance?

Star Wars IV Network



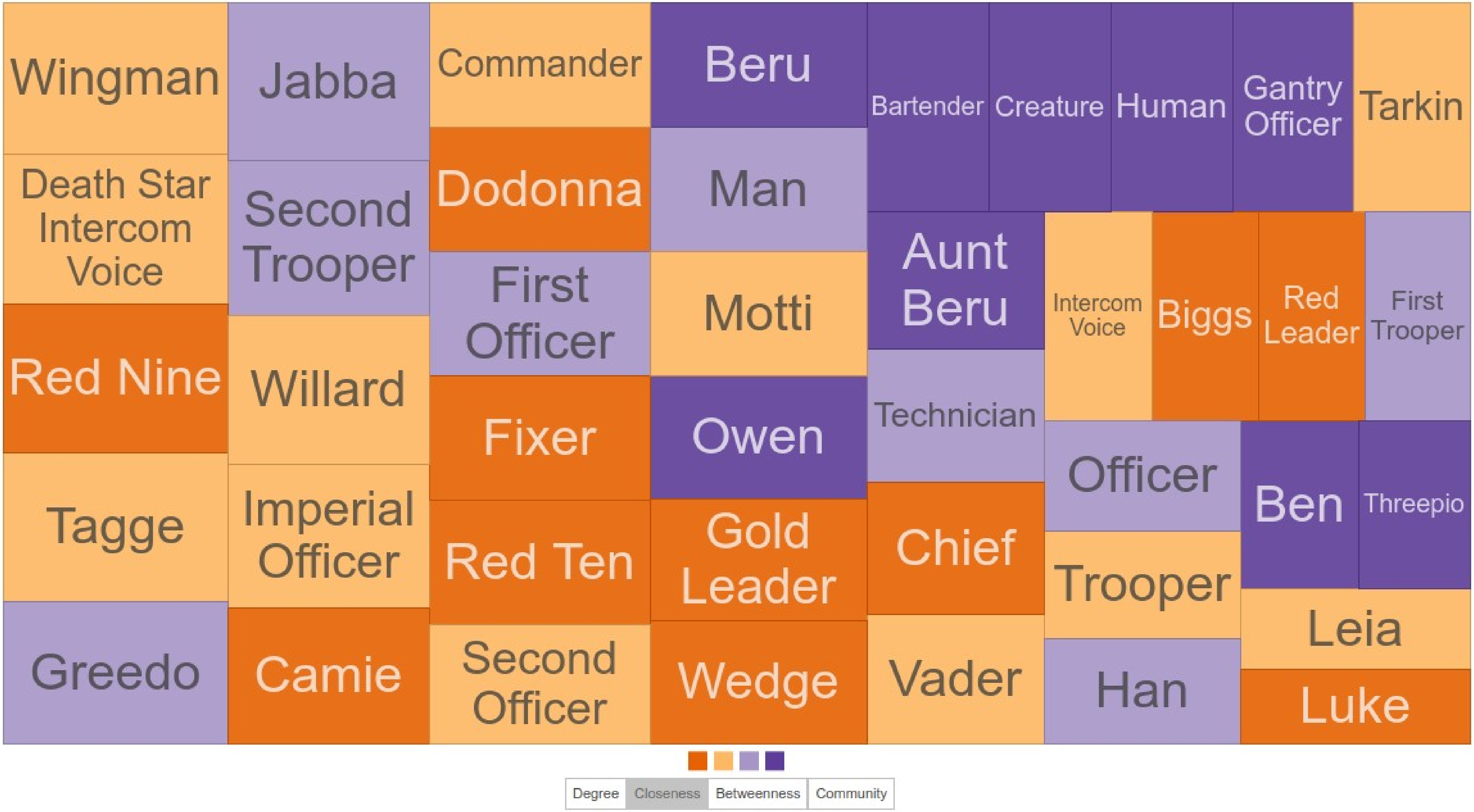
Size proportional to degree: is this the only way?

Star Wars IV Network



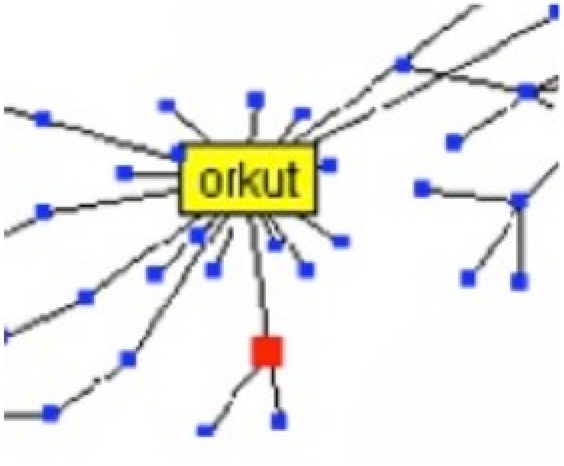
Size proportional to betweenness

Star Wars IV Network



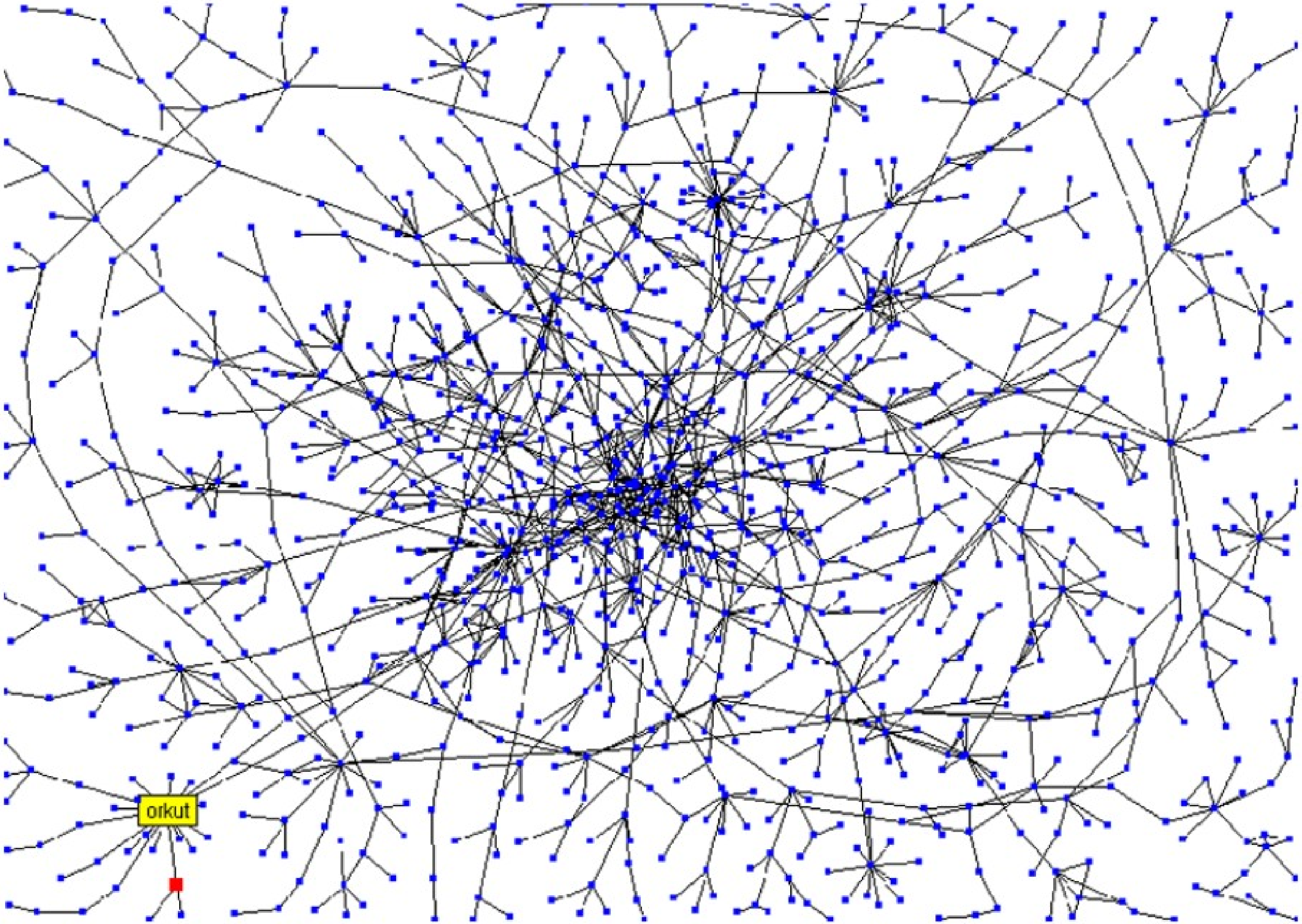
Size proportional to closeness

Why degree is not enough



Why degree is not enough

Stanford Social Web (ca. 1999)

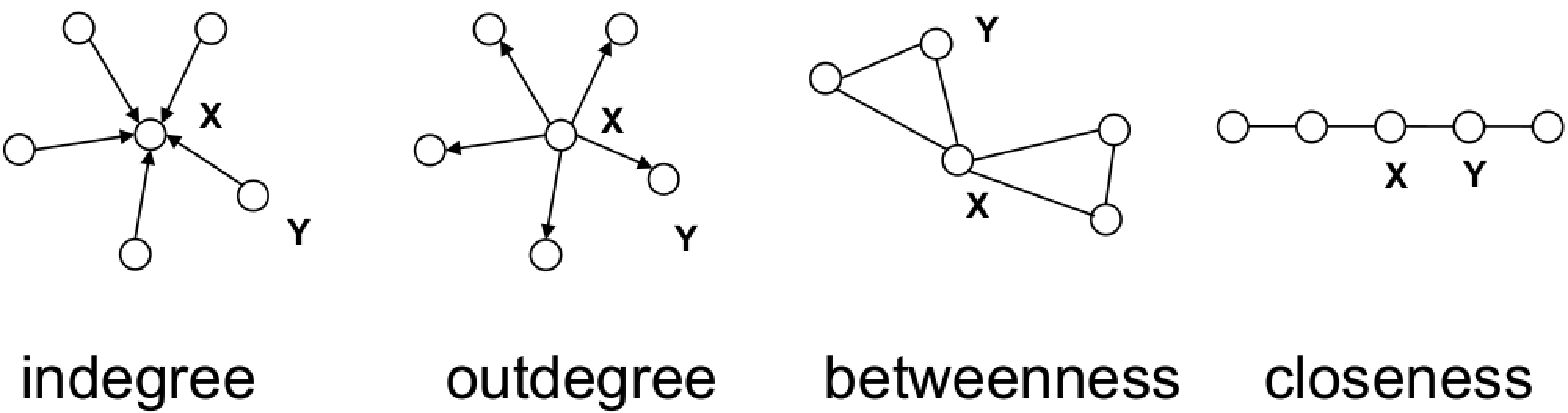


network of personal homepages at Stanford

Different notions of centrality

- **Node Centrality** measures "importance"

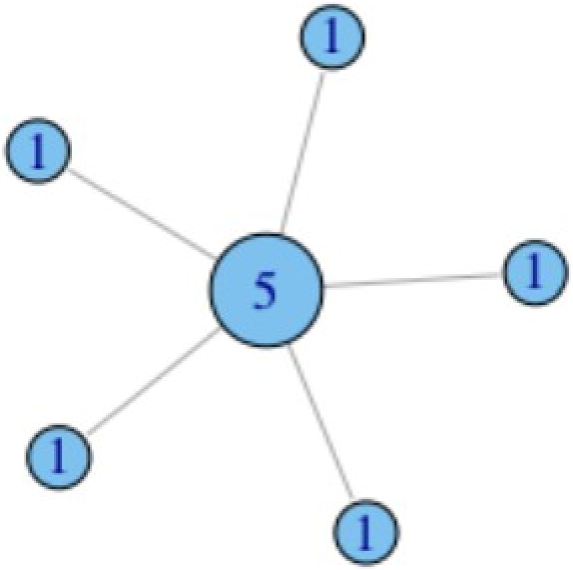
In each of the following networks, X has higher centrality than Y according to a particular measure



Node Degree

- Let's put some **numbers** to it

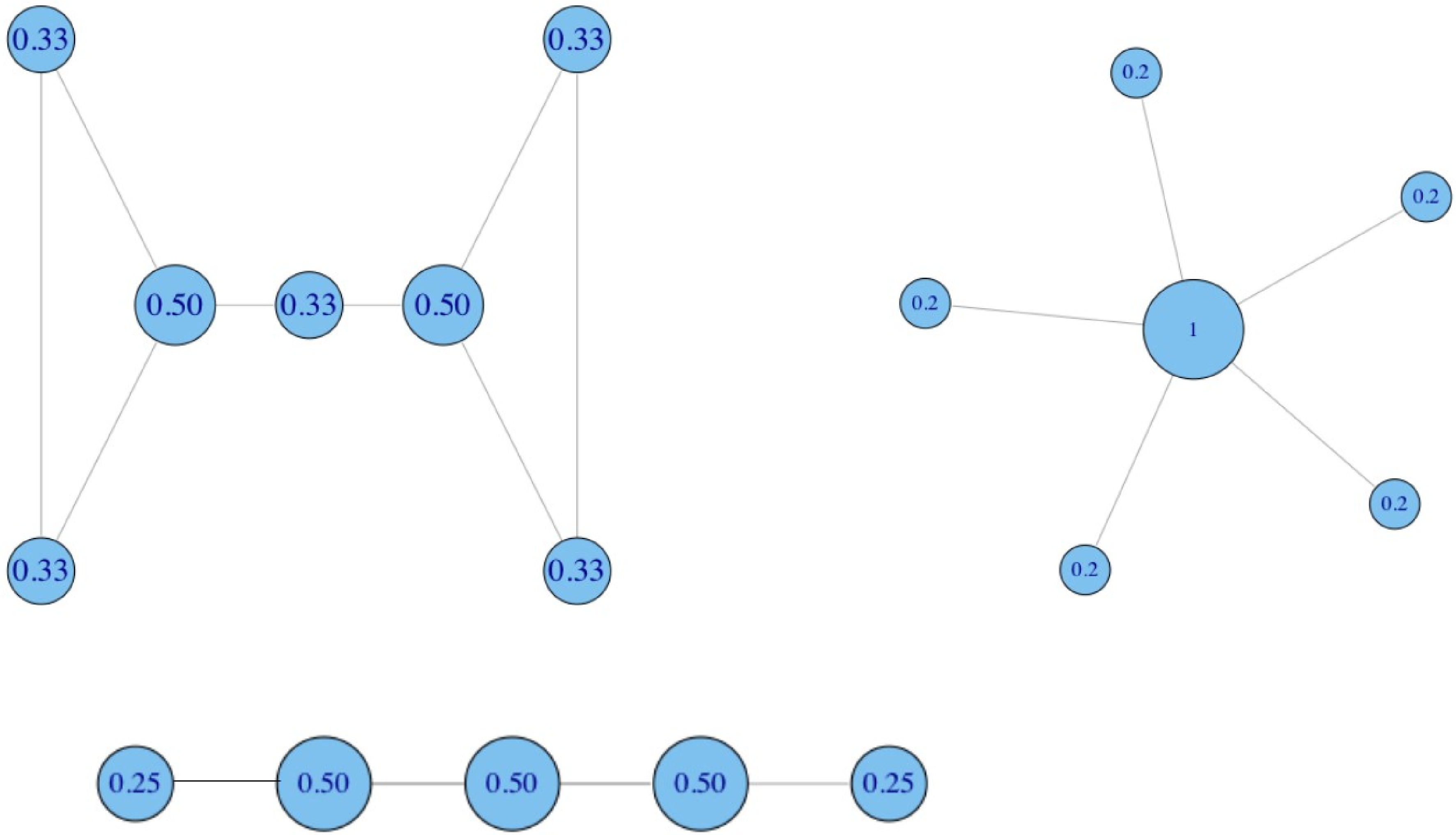
Undirected degree:
e.g. nodes with more friends are more central.



Assumption: the connections that your friend has don't matter, it is what they can do directly that does (e.g. go have a beer with you, help you build a deck...)

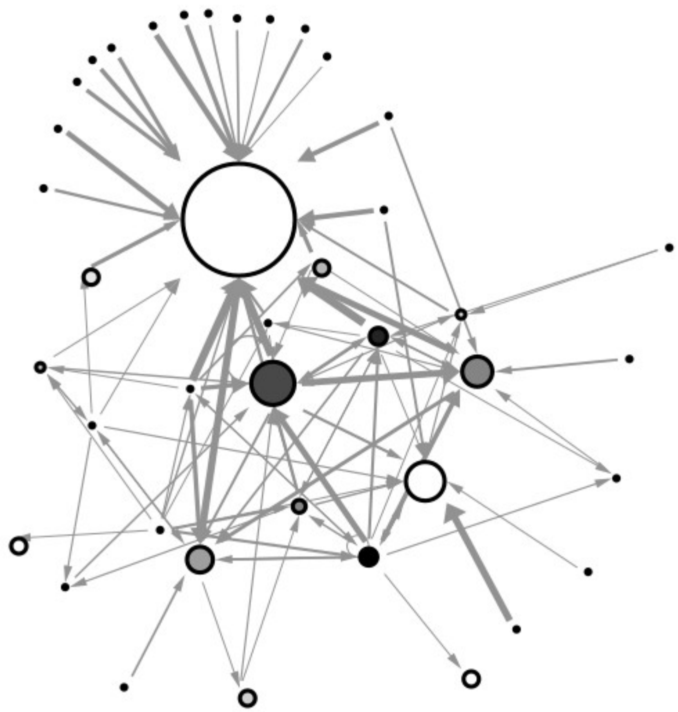
Node Degree

- **Normalization:**
  divide degree by the max. possible, i.e. (N-1)

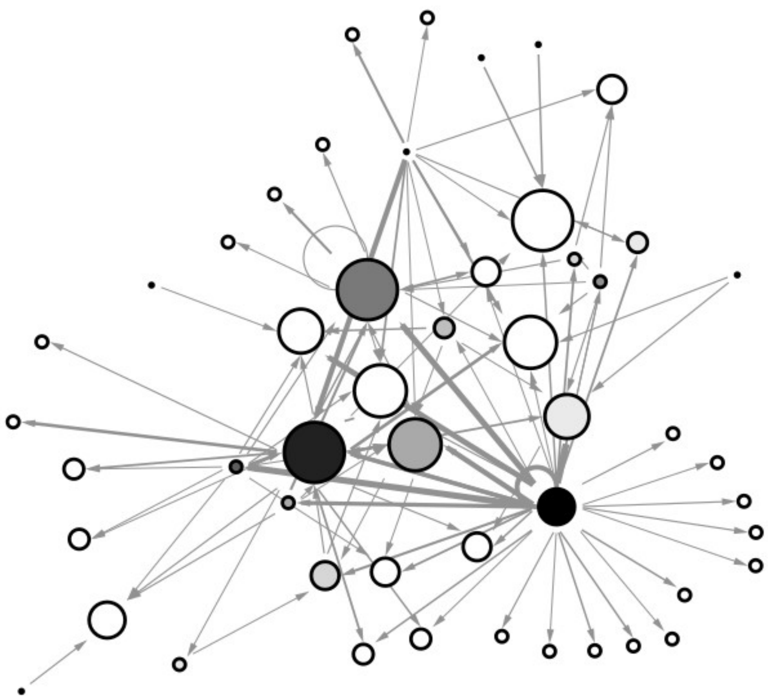


Node Degree

example financial trading networks



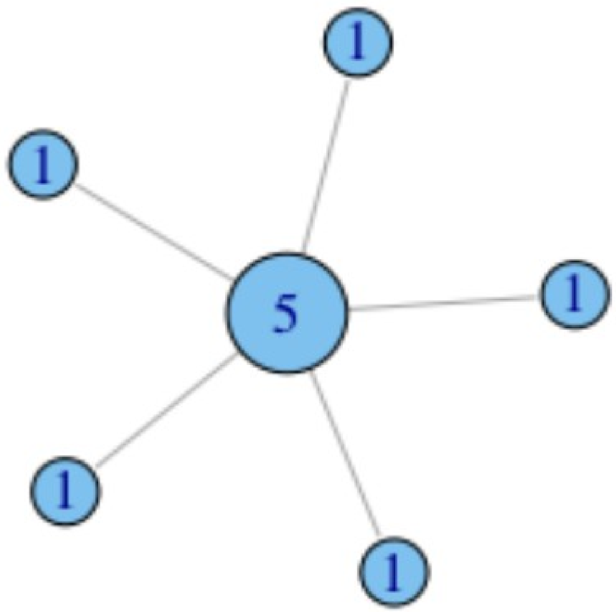
high in-centralization:
one node buying from
many others

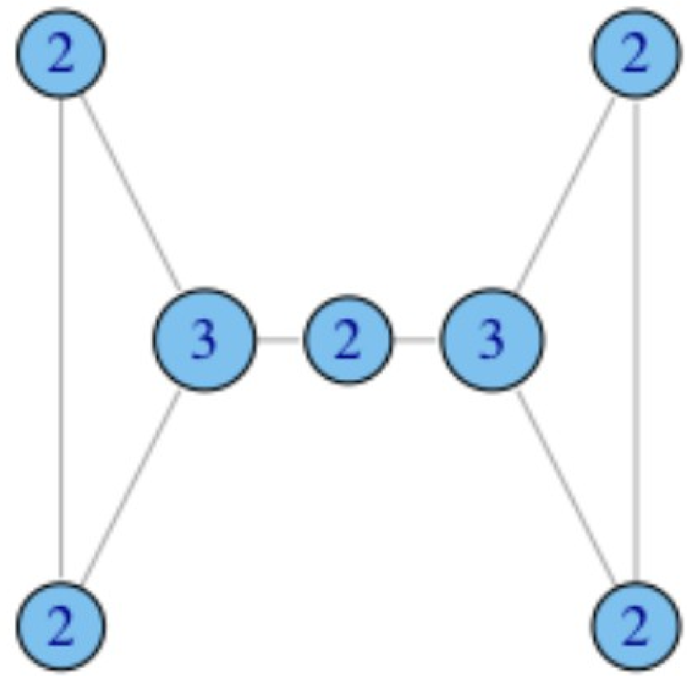


low in-centralization:
buying is more evenly
distributed

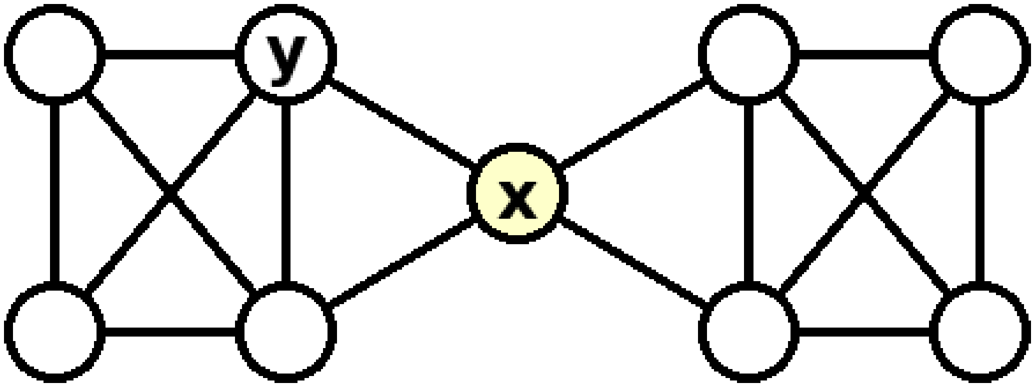
What does degree not capture?

- In what ways does degree fail to capture centrality in the following graphs?

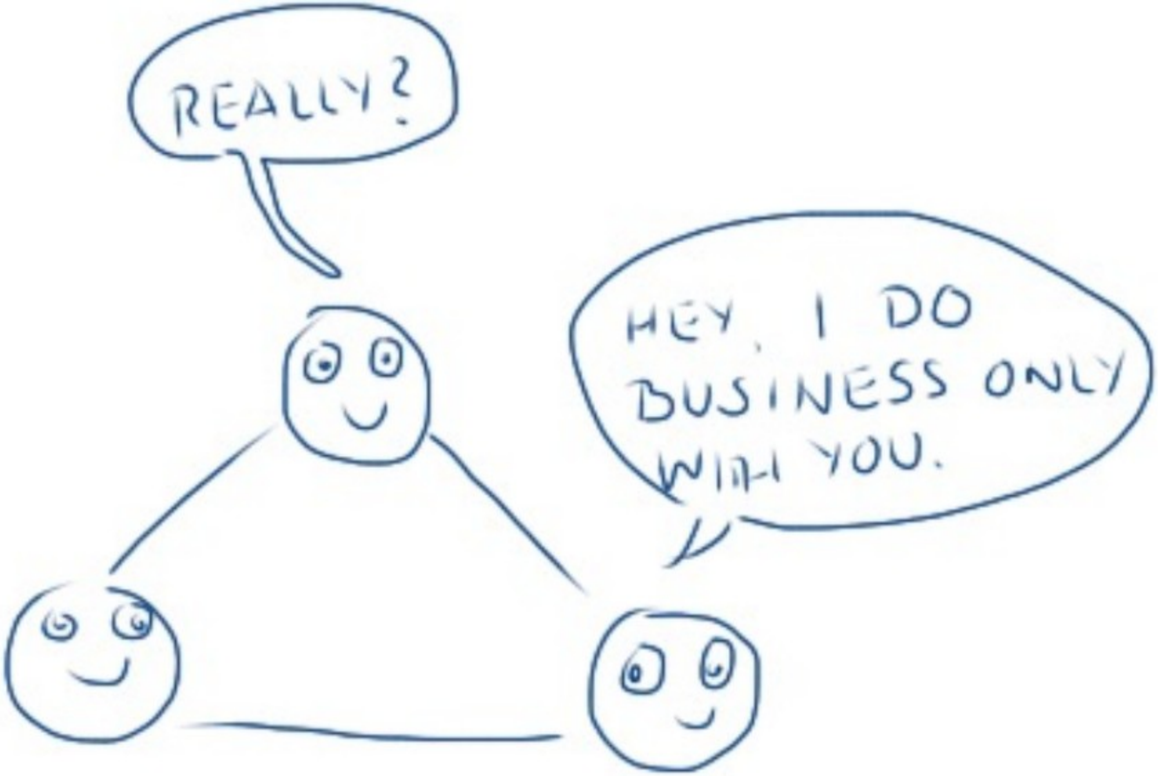




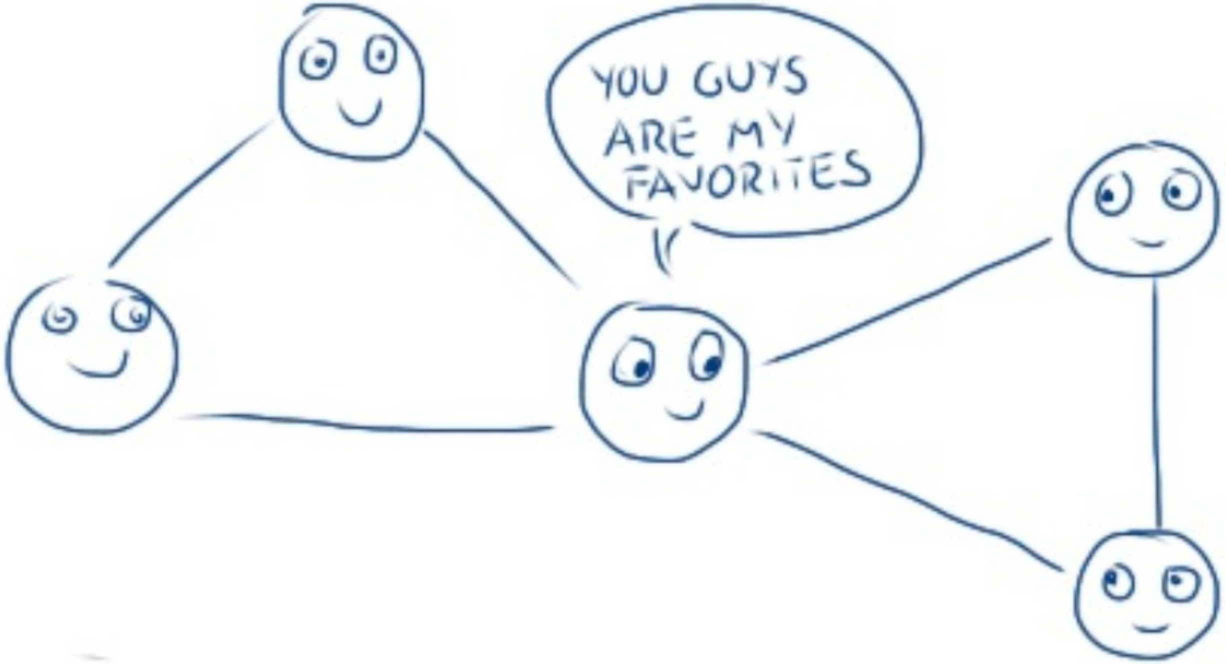
Brokerage not captured by degree



Brokerage: Concept



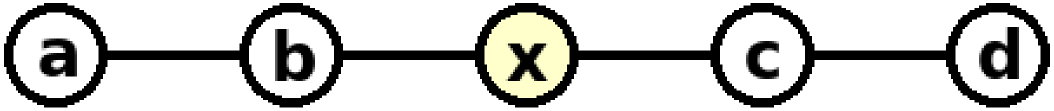
Brokerage: Concept



Capturing Brokerage

- **Betweenness Centrality:**

intuition: how many **pairs of individuals** would have to go through you in order to reach one another in the **minimum number of hops**?



Betweenness: Definition

$$C_B(i) = \sum_{j<k} \frac{g_{jk}(i)}{g_{jk}}$$

Where:

g_{jk} = the number of **shortest paths** connecting nodes *j* and *k*

$g_{jk}(i)$ = the number that node *i* is on.

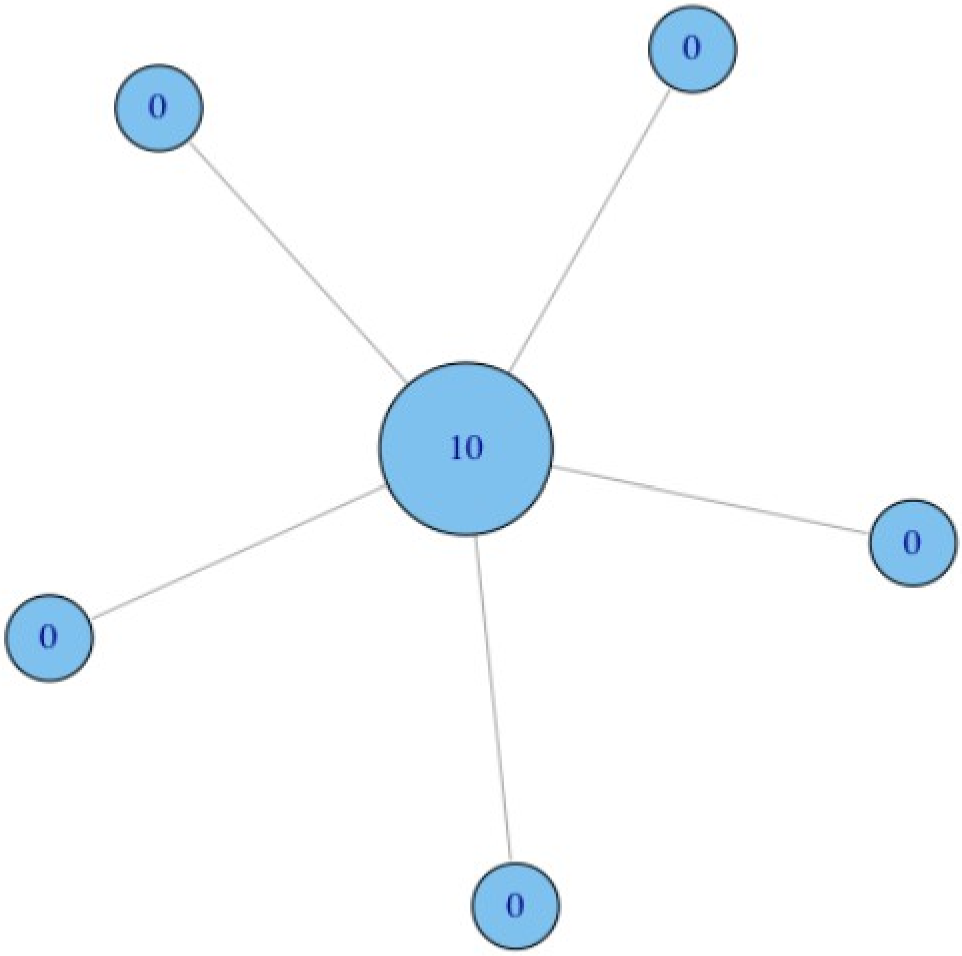
Usually normalized by:

$$C'_B(i) = \frac{C_B(i)}{\boxed{(n-1)(n-2)/2}}$$

number of pairs of vertices
excluding the vertex itself

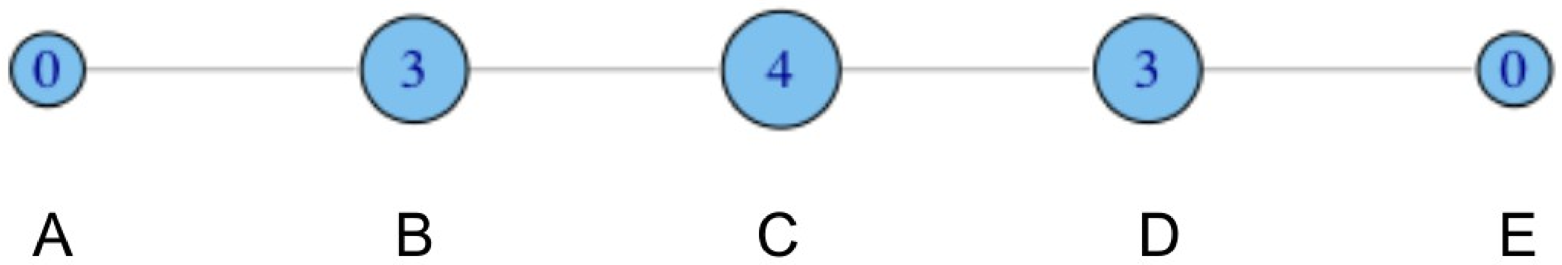
Betweenness: Toy Networks

- Non-normalized version:



Betweenness: Toy Networks

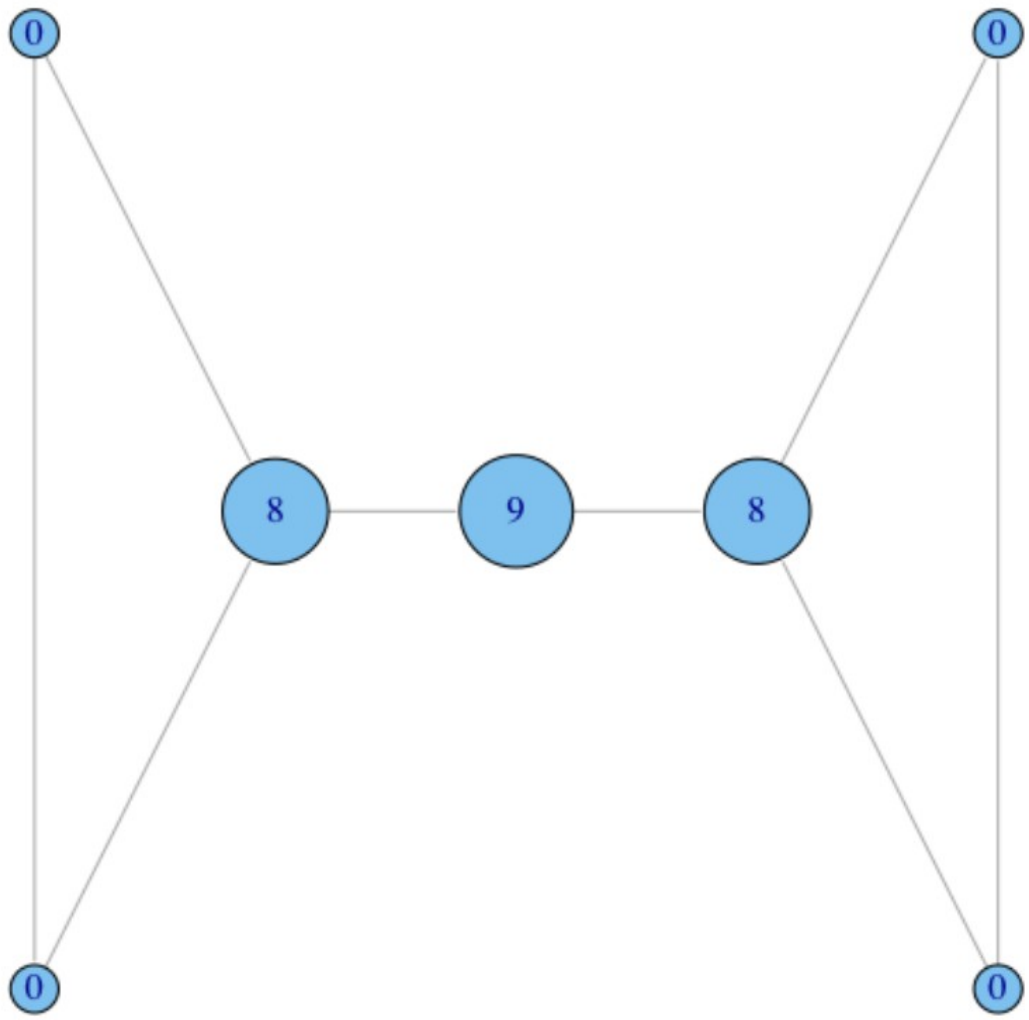
- Non-normalized version:



  - A lies between no two other vertices
  - B lies between A and 3 other vertices: C, D, and E
  - C lies between 4 pairs of vertices: (A,D),(A,E),(B,D),(B,E)
    - note that there are no alternate paths for these pairs to take, so C gets full credit

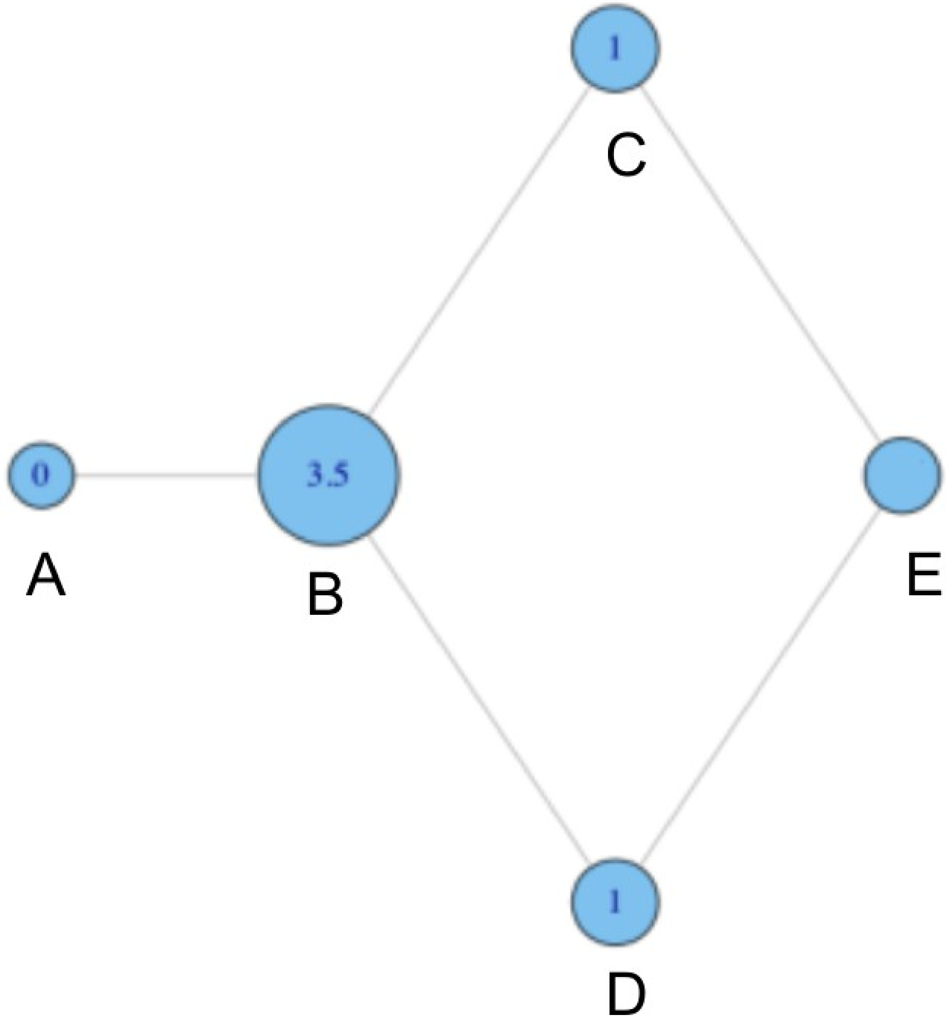
Betweenness: Toy Networks

- Non-normalized version:



Betweenness: Toy Networks

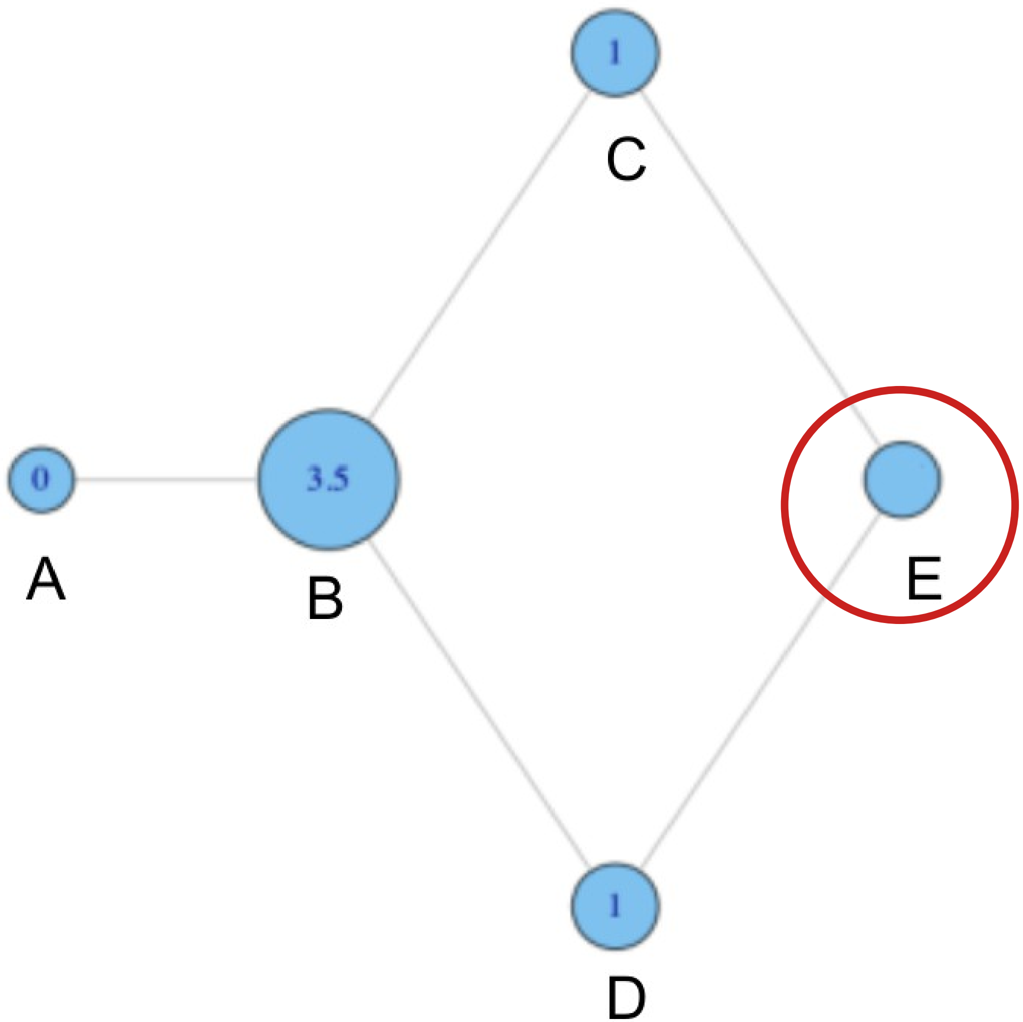
- Non-normalized version:



- why do C and D each have betweenness 1?
- They are both on shortest paths for pairs (A,E), and (B,E), and so must share credit:
  - 1⁄2+1/2 = 1

Betweenness: Toy Networks

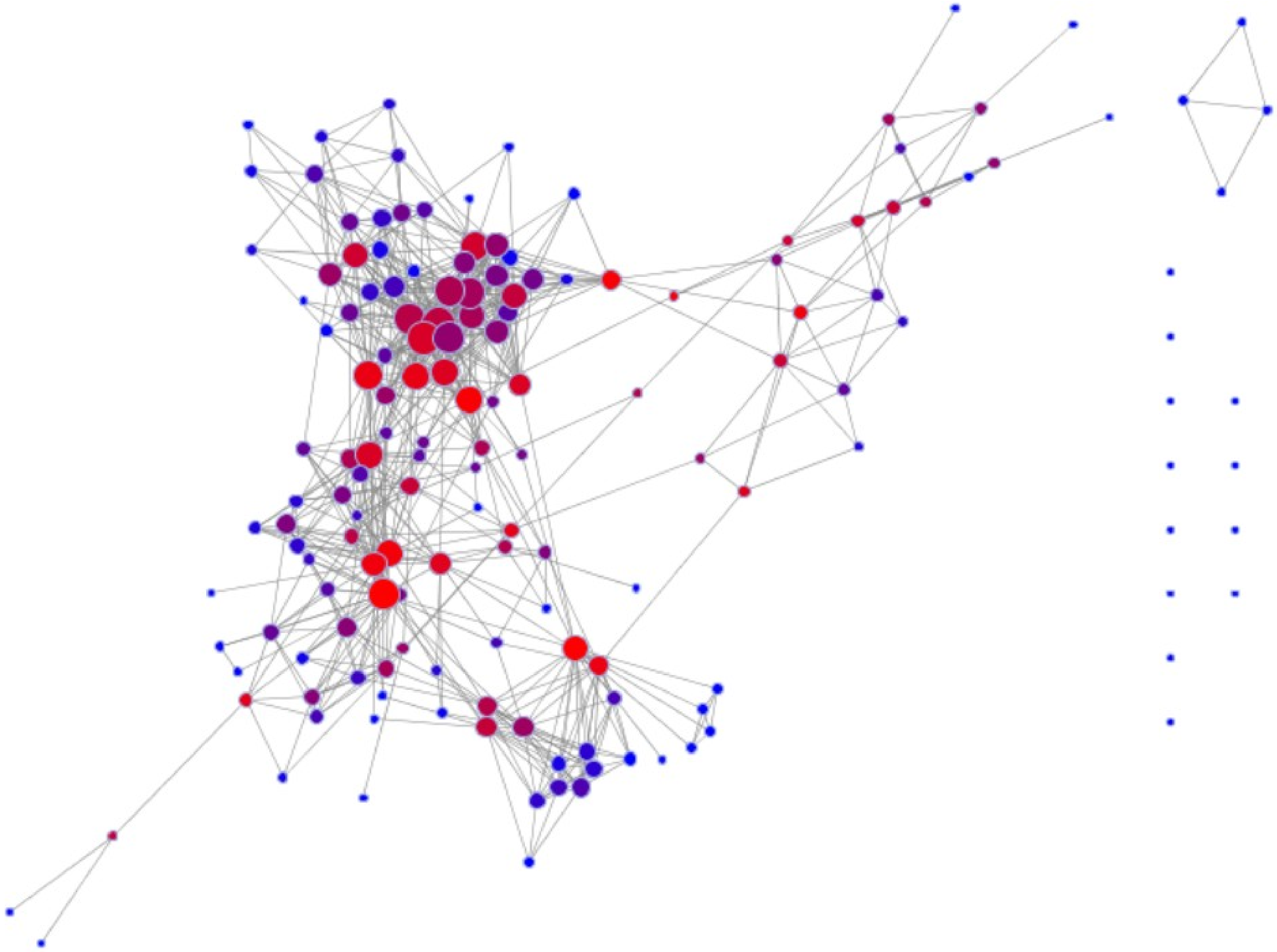
- Non-normalized version:



What is the betweenness of node E?

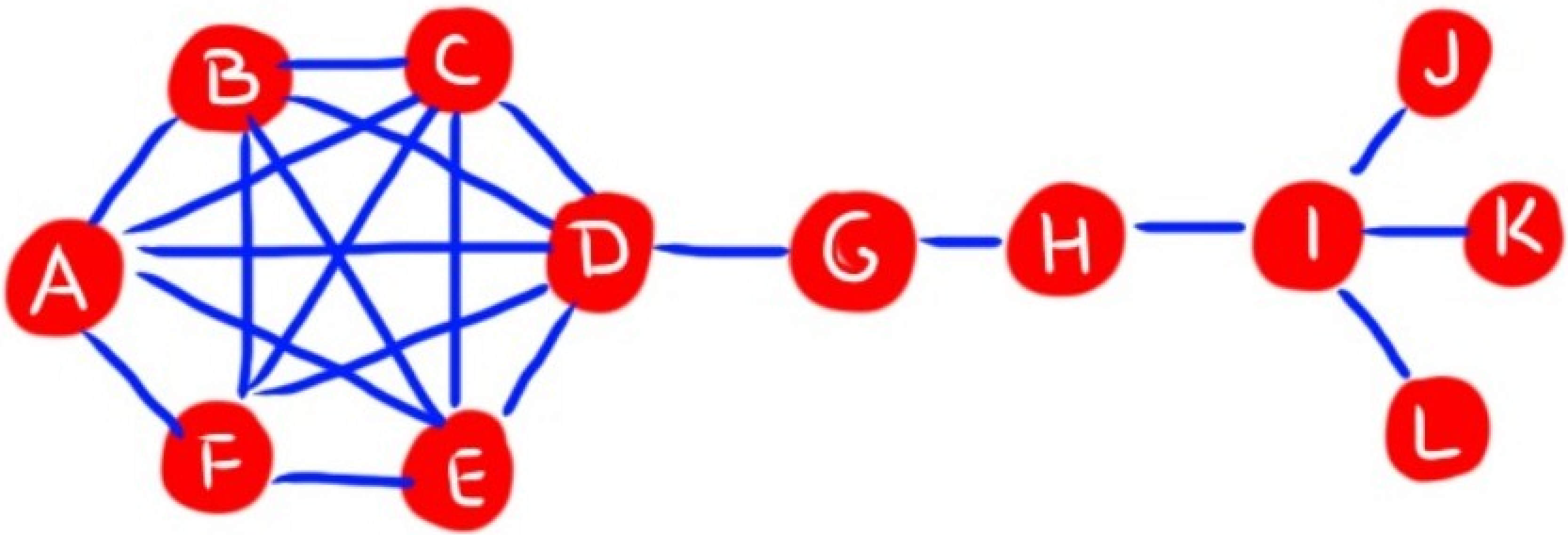
Betweenness: Real Example

- Social Network (facebook)
  nodes are sized by degree, and colored by betweenness



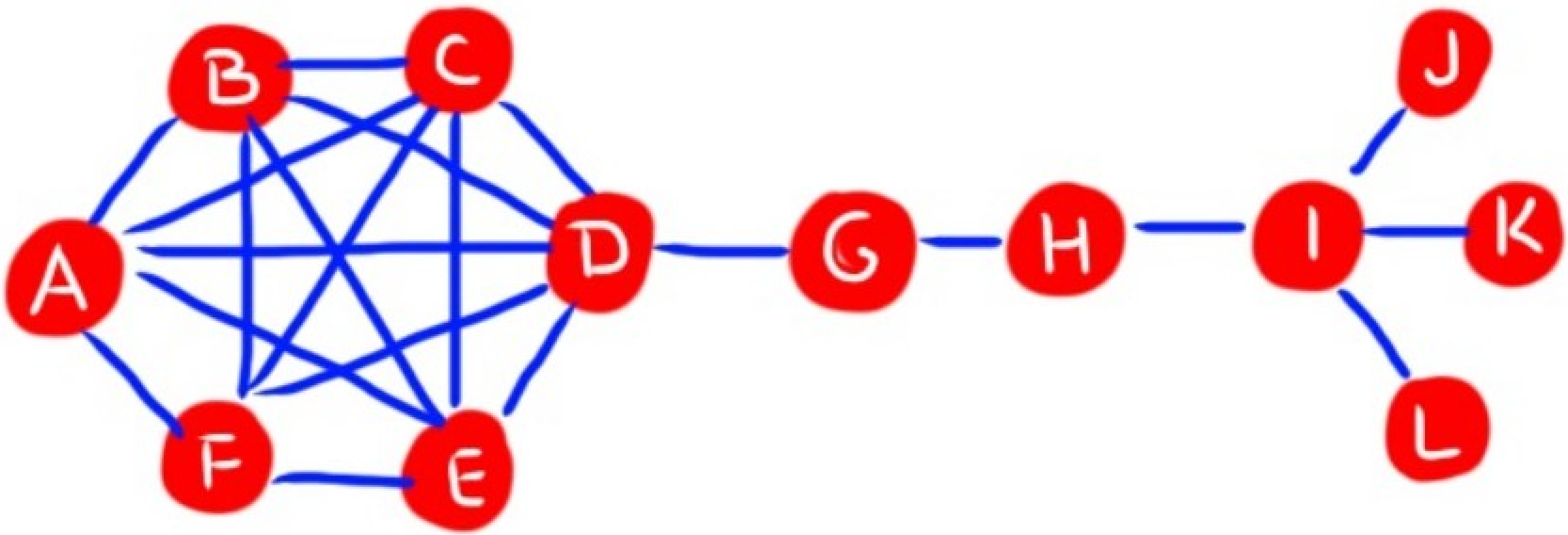
Betweenness: Question

- Find a node that has **high betweenness** but **low degree**



Betweenness: Question

- Find a node that has **low betweenness** but **high degree**

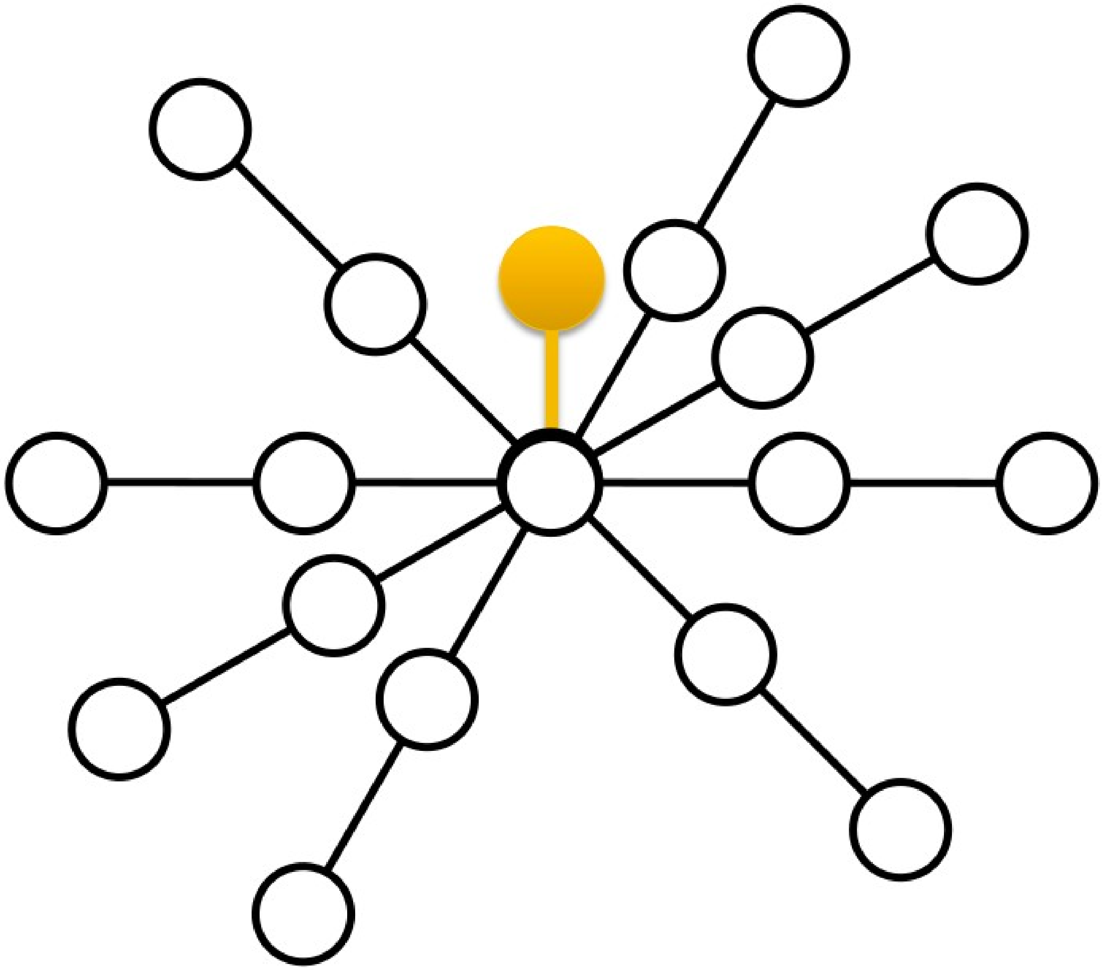


Closeness Centrality

- What if it’s not so important to have many direct friends?
- Or be “between” others
- But one still wants to be in the **“middle”** of things, **not too far from the center**

Closeness Centrality

- Need not be in brokerage position



Closeness: Definition

- **Closeness** is based on the **length of the average shortest path** between a node and all other nodes in the network

Closeness Centrality:

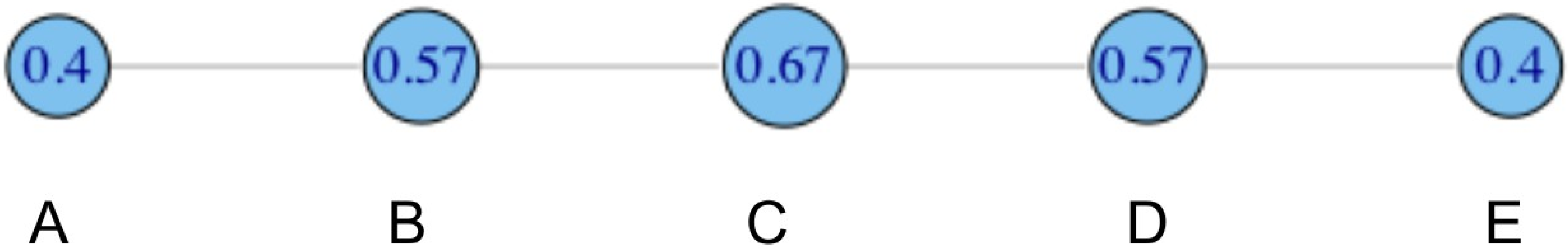
$$C_C(i) = \frac{1}{\sum_{j=1}^{N} d(i,j)}$$

Normalized Closeness Centrality:

$$C_C'(i) = C_C(i) \times (n-1)$$

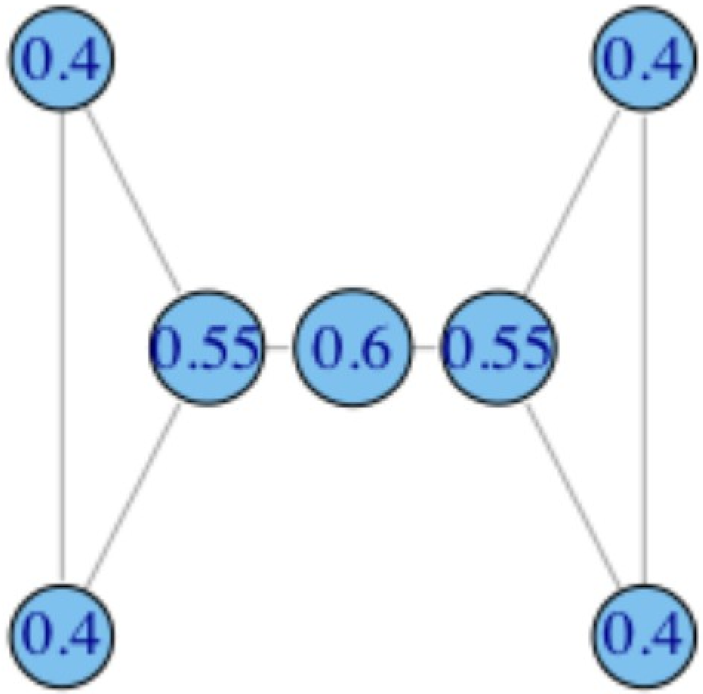
When graphs are big, the -1 can be discarded and we multiply by *n*

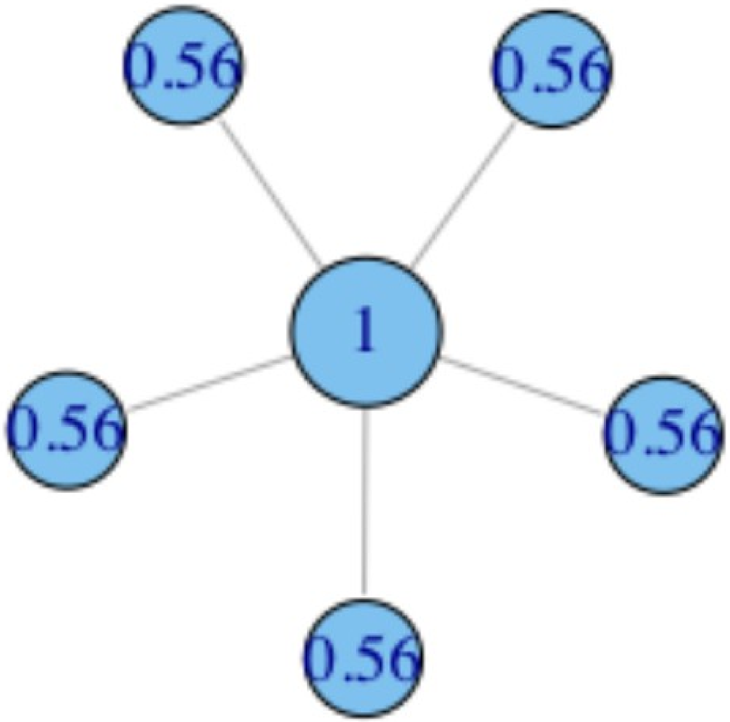
Closeness: Toy Networks



$$C_c^{'}(A) = \left[\frac{\sum_{j=1}^{N} d(A,j)}{N-1} \right]^{-1} = \left[\frac{1+2+3+4}{4} \right]^{-1} = \left[\frac{10}{4} \right]^{-1} = 0.4$$

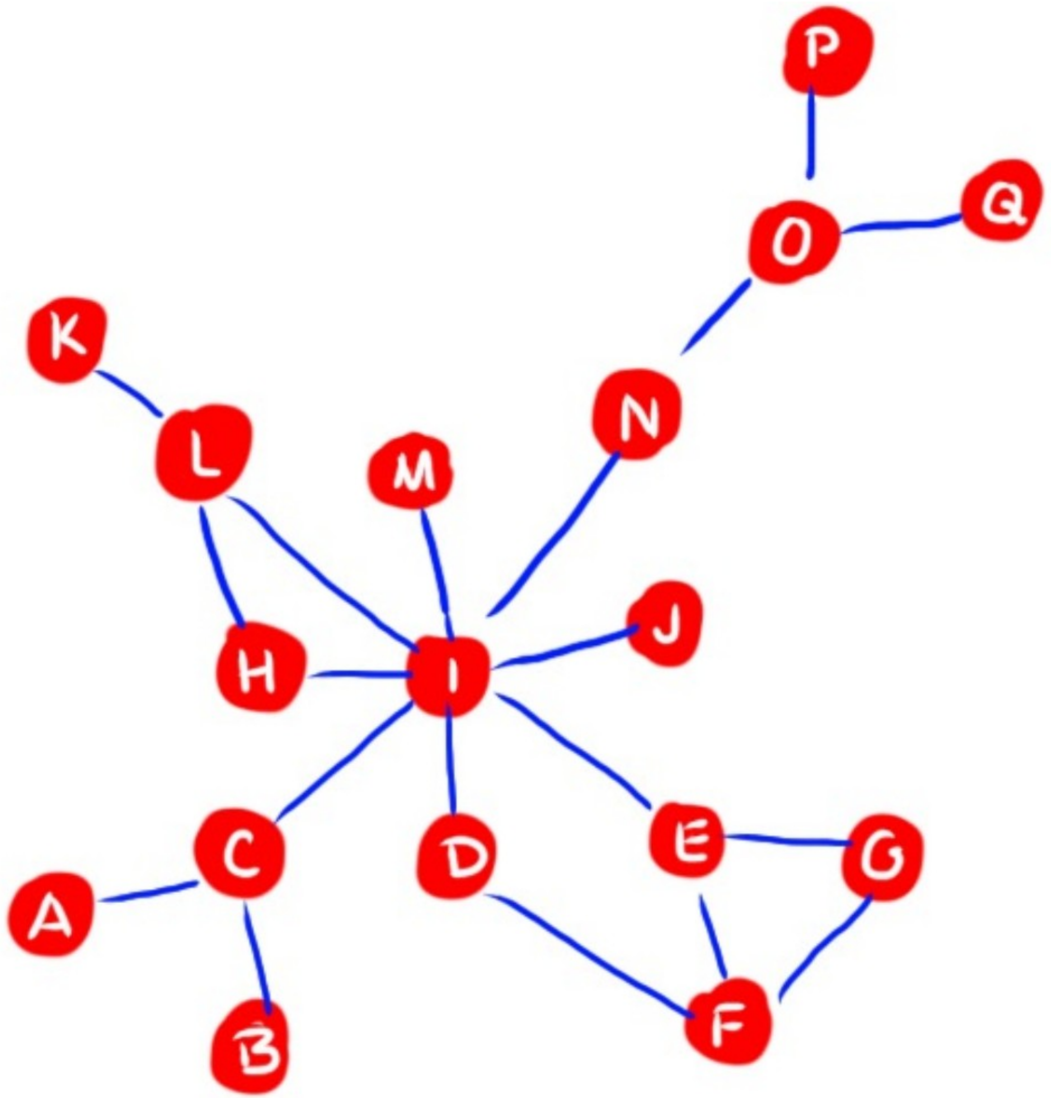
Closeness: Toy Networks





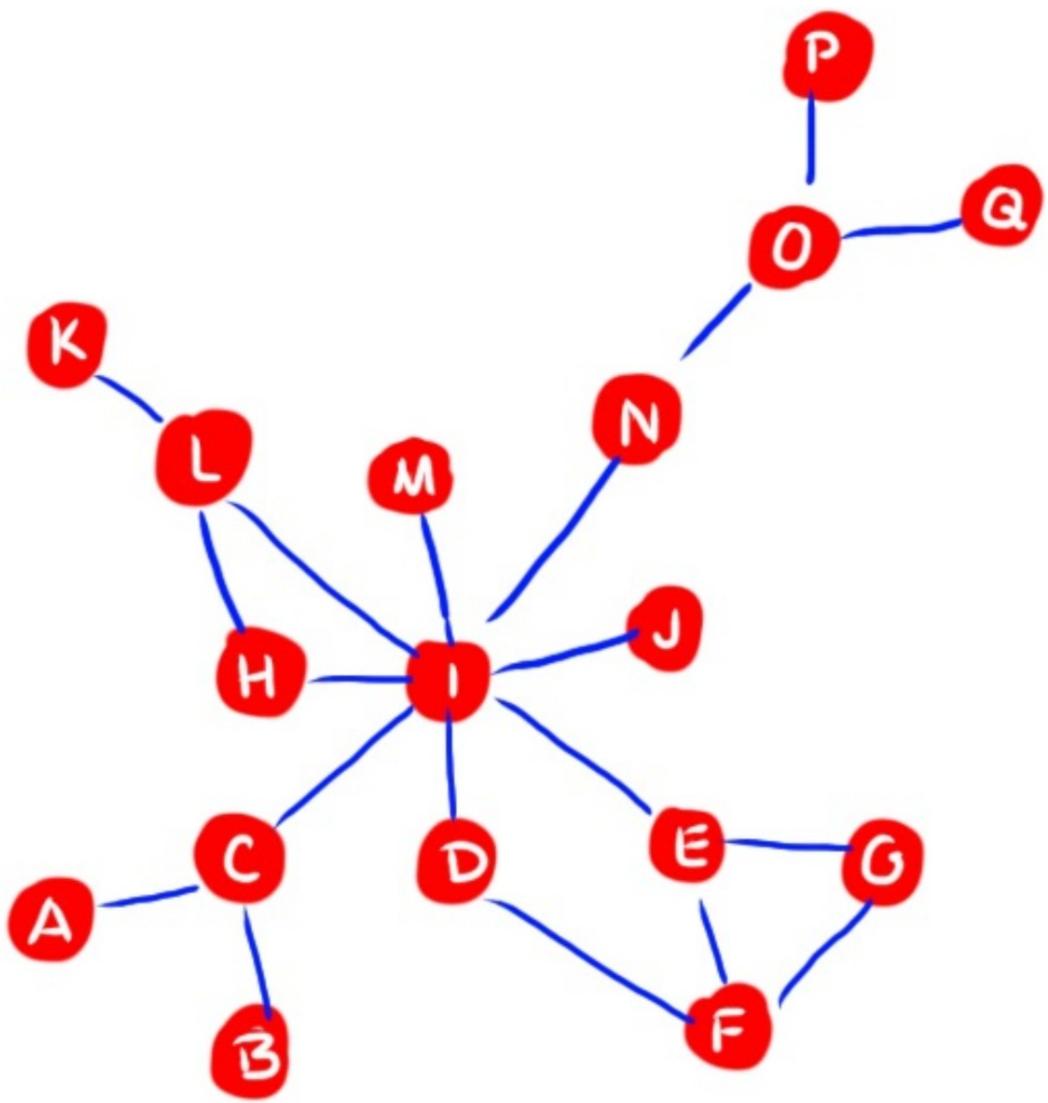
Closeness: Question

- Find a node which has relatively **high degree** but low **closeness**



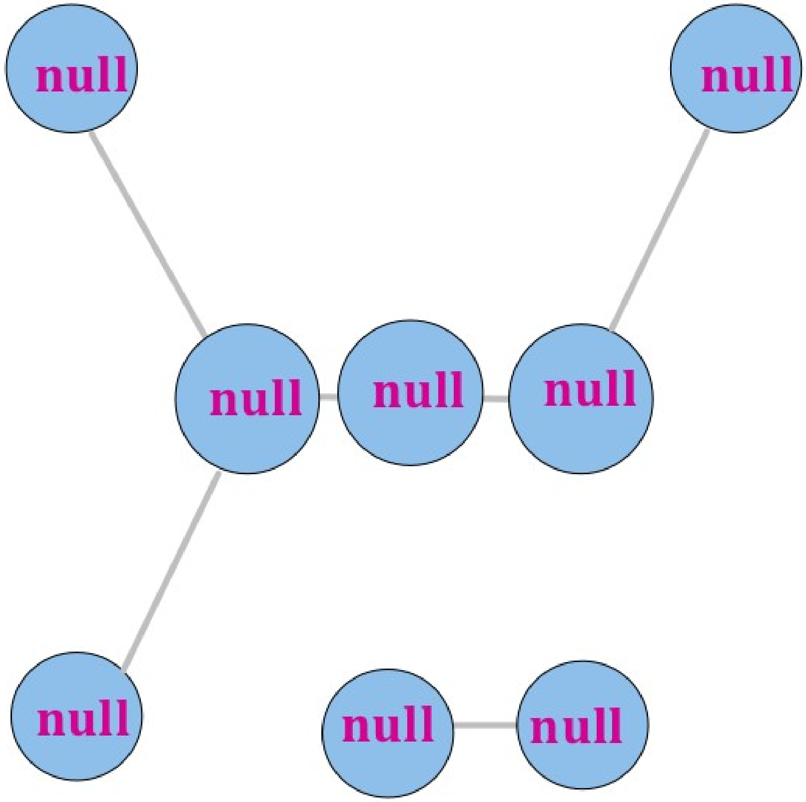
Closeness: Question

- Find a node which has **low degree** but **high closeness**



Closeness: unconnected graph

- What if the graph is **not connected**?



We get null score for all nodes,
if the graph is not connected!

$$C_C(i) = \frac{1}{\sum_{j=1}^{N} d(i,j)}$$

instead of *null*, we could also interpret it as 0 if *infinity* is the distance between unconnected nodes

Harmonic: Definition

- Replace the average distance with the **harmonic mean** of all distances

Harmonic Centrality:

$$C_H(i) = \sum_{j \neq i} \frac{1}{d(i,j)} = \sum_{d(i,j)<\infty,\, j \neq i} \frac{1}{d(i,j)}$$

- Strongly correlated to closeness centrality
- Naturally also accounts for nodes *j* that cannot reach *i*
- Can be applied to graphs that are not connected

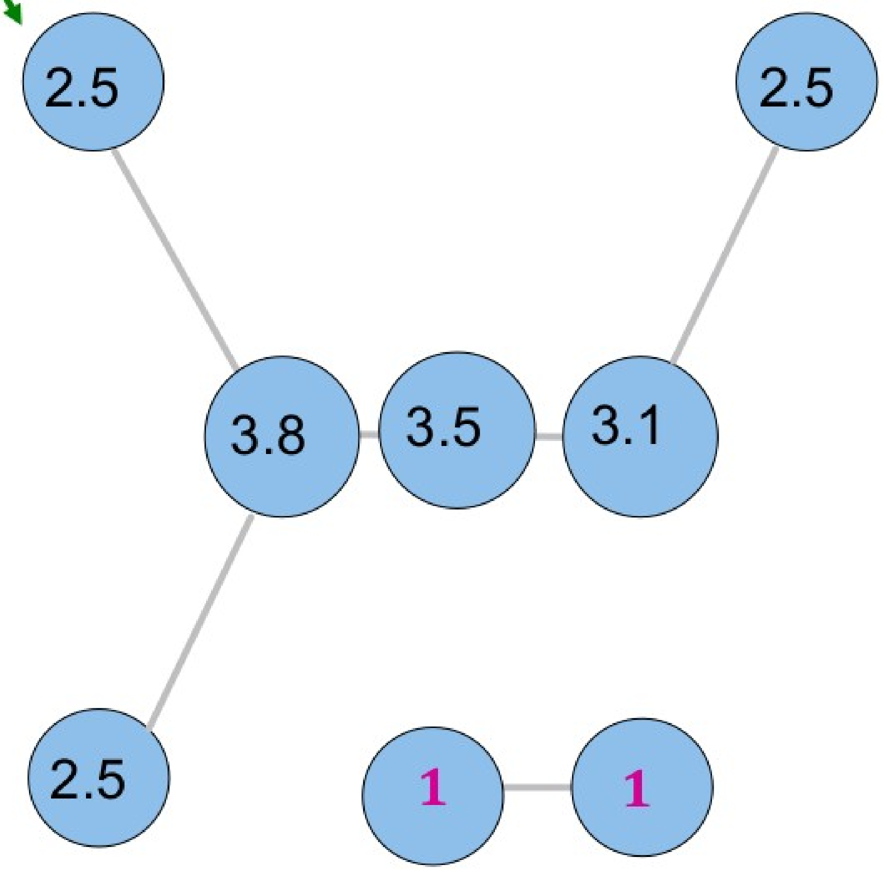
Normalized Harmonic Centrality:

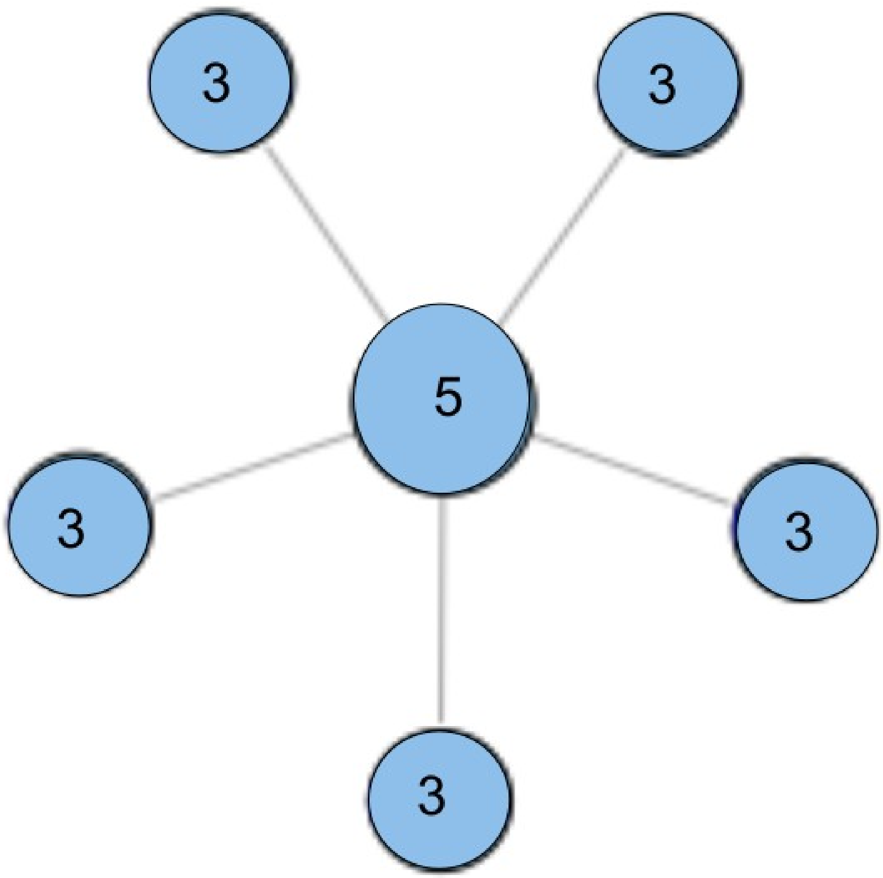
$$C'_H(i) = C_H(i)/(n-1)$$

Harmonic: Toy Networks

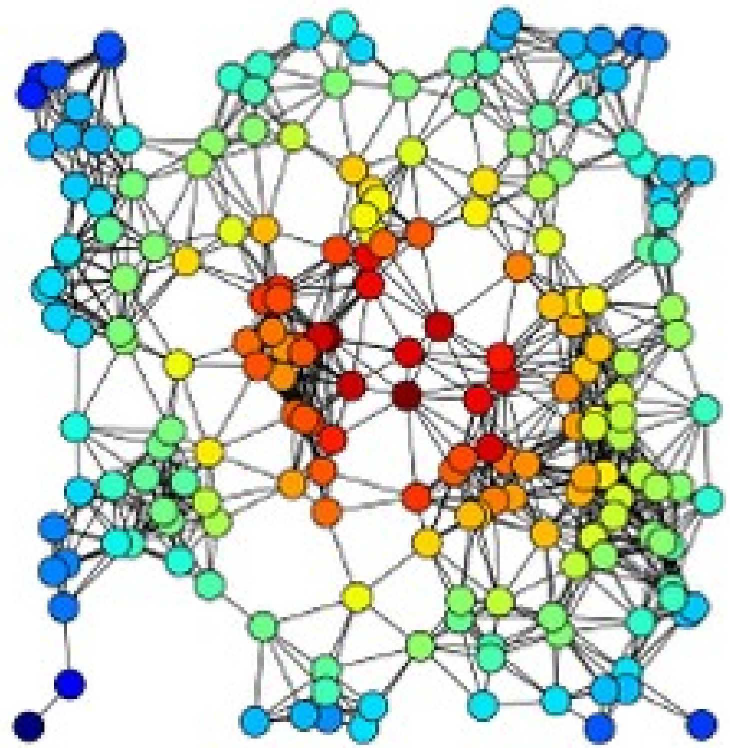
- Non-normalized version:

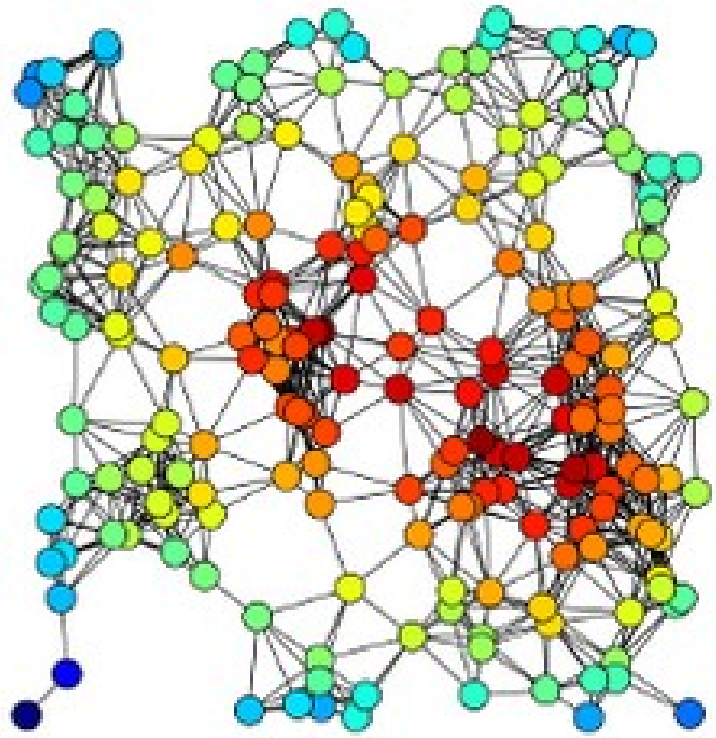
$$c_{harm} = \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.5$$





Closeness vs Harmonic





Closeness Centrality

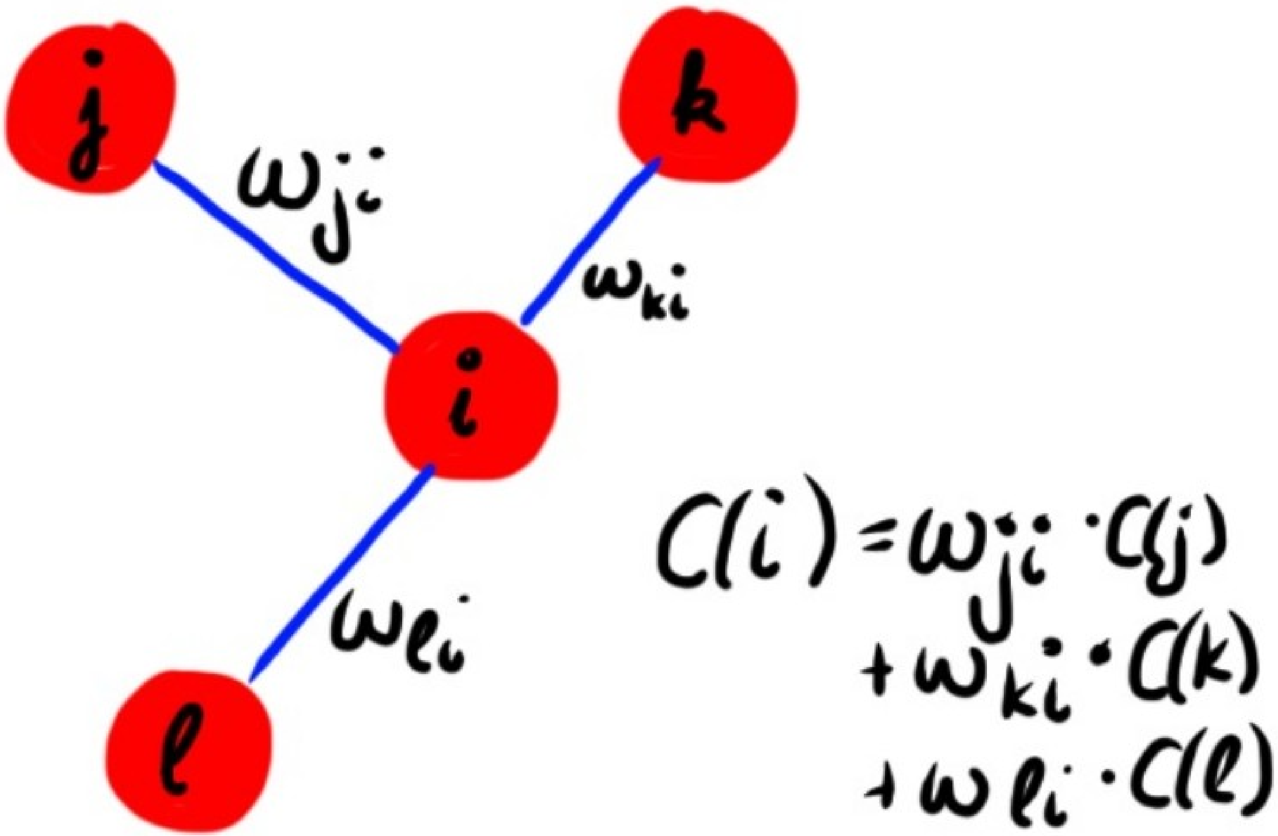
$$C_C(i) = \frac{1}{\sum_{j=1}^{N} d(i,j)}$$

Harmonic Centrality

$$C_H(i) = \sum_{j \neq i} \frac{1}{d(i,j)}$$

Eigenvector Centrality

- How “central” you are depends on how “central” your neighbors are



Eigenvector Centrality

Eigenvector Centrality:

$$C_E(i) = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ji} \times C_E(j)$$

where λ is a constant and
A_{ij} the adjacency matrix (1 if *(i,j)* are connected, 0 otherwise)

(with a small rearrangement) this can we rewritten
in vector notation as in the eigenvector equation

$$Ax = \lambda x$$

where *x* is the eigenvector, and its *i*-th component is the centrality of node *i*

In general, there will be many different eigenvalues λ for which a non-zero eigenvector solution exists. However, the additional requirement that all the entries in the eigenvector be non-negative implies (by the Perron–Frobenius theorem) that only the greatest eigenvalue results in the desired centrality measure

Bonacich eigenvector centrality

also known as Bonacich Power Centrality

$$c_i(\beta) = \sum (\alpha + \beta c_j) A_{ji}$$

- α is a normalization constant
- β determines how important the centrality of your neighbors is
- $\mathbf{A}$ is the adjacency matrix (can be weighted)

Bonacich eigenvector centrality

also known as Bonacich Power Centrality

small β ➔ high attenuation

only your immediate friends matter, and their importance is factored in only a bit

high β ➔ low attenuation

global network structure matters (your friends, your friends' of friends etc.)

β = 0 yields simple degree centrality

$$c_i(\beta) = \sum_j (\alpha \qquad) A_{ji}$$

Eigenvector Variants

- There are other **variants** of eigenvector centrality, such as:
  - **PageRank**
    - (normalized eigenvector + random jumps) [link analysis]
  - **Katz Centrality**
    - (connections with distant neighbors are penalized)

$$C_{\mathrm{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ji}$$

Centrality in Directed Networks

- **Degree:**
  - in and out centrality
- **Betweenness:**
  - Consider only directed paths: $$C_B(i) = \sum_{j \neq k} \frac{g_{jk}(i)}{g_{jk}}$$
  - When normalizing take care of ordered pairs

$$C'_B(i) = \frac{C_B(i)}{\boxed{(n-1)(n-2)}}$$

number of ordered pairs is
2x the number of unordered

- **Closeness**
  - Consider only directed paths
- **Eigenvector** (already prepared)

Centrality in Weighted Networks

- **Degree:**
  - Sum weights *(non-weighted equals weight=1 for all edges)*
- **Betweenness and Closeness:**
  - Consider weighted distance
- **Eigenvector**
  - Consider weighted adjacency matrix

Node Centralities: Conclusion

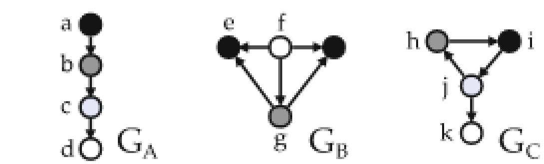
- There are other node centrality metrics, but these are the **“quintessential”**

Finding Dominant Nodes Using Graphlets

David Aparício(✉), Pedro Ribeiro, Fernando Silva, and Jorge Silva

CRACS & INESC-TEC and the Department of Computer Science,
Faculty of Sciences, University of Porto, 4169-007 Porto, Portugal
{daparicio,pribeiro,fds}@dcc.fc.up.pt, jorge.m.silva@inesctec.pt

$$D(o) = \left(\lambda \times \sum_{o_i \in \mathcal{I}(o)} \beta^{k-d(o,o_i)}\right) - \left((1-\lambda) \times \sum_{o_j \in \mathcal{S}(o)} \beta^{k-d(o_j,o)}\right)$$



	a	b	c	d
a	-	1	2	3
b	-	-	1	2
c	-	-	-	1
d	-	-	-	-

	e	e	f	g
e	-	-	-	-
e	-	-	-	-
f	1	1	-	1
g	1	1	-	-

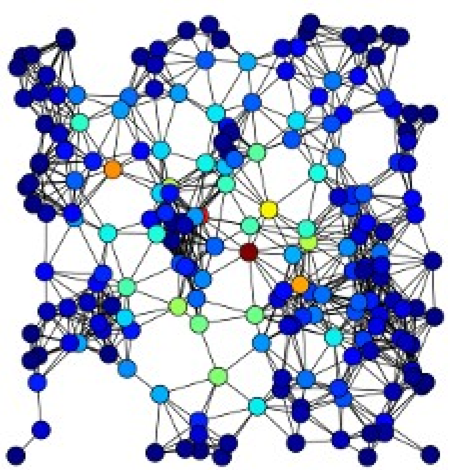
	h	i	j	k
h	-	1	2	3
i	2	-	1	2
j	1	2	-	1
k	-	-	-	-

A subgraph-based ranking system for professional tennis players

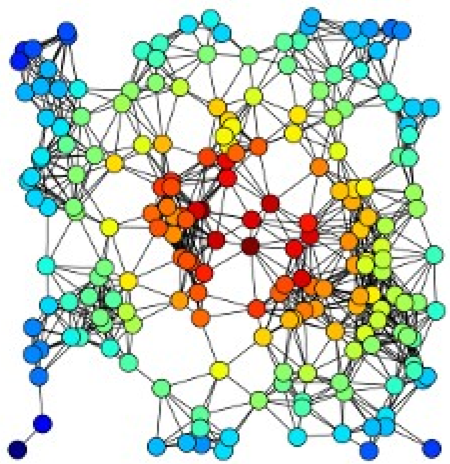
David Aparício, Pedro Ribeiro and Fernando Silva

- Which one to use depends on **what you want to achieve or measure**
  - Worry about understanding the concepts
  - They enlarge your graph vocabulary

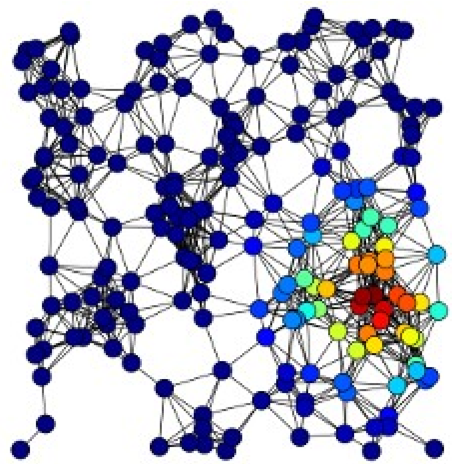
Node Centralities: Conclusion



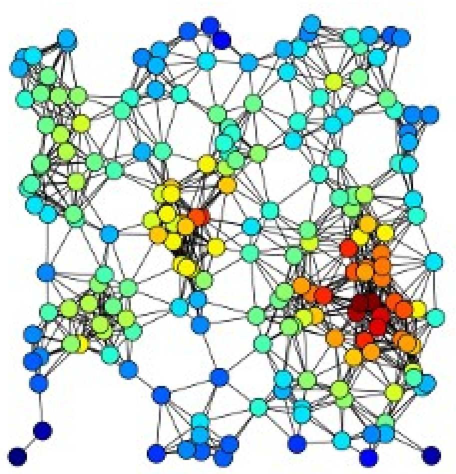
Betweenness



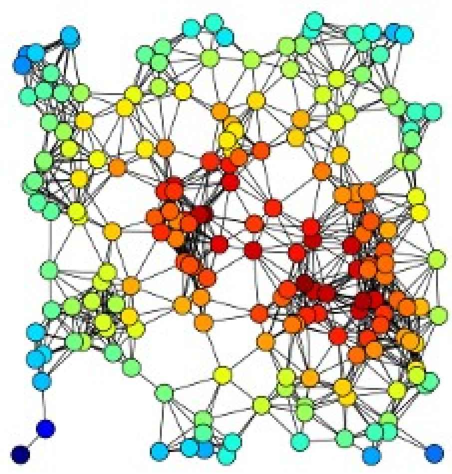
Closeness



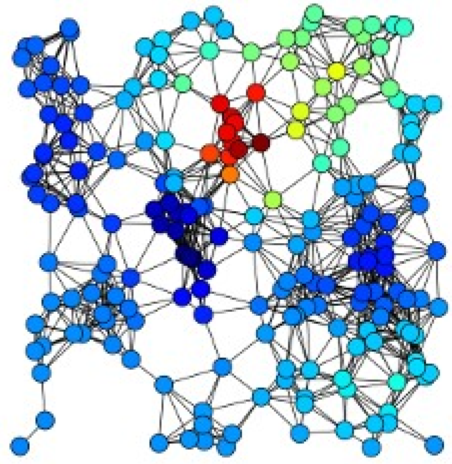
Eigenvector



Degree



Harmonic



Katz

Node Centralities: Conclusion

- All (major) network analysis packages provide them:

neo4j

The #1 Database for Connected Data

Centrality algorithms are used to determi
includes the following centrality algorithn

- Production-quality
  - Page Rank
  - Betweenness Centrality
- Alpha
  - ArticleRank
  - Closeness Centrality
  - Harmonic Centrality
  - Degree Centrality
  - Eigenvector Centrality
  - HITS

NetworkX
Network Analysis in Python

Centrality

Degree

`degree_centrality` (G)	Compute the degree centrality for nodes.
`in_degree_centrality` (G)	Compute the in-degree centrality for nodes.
`out_degree_centrality` (G)	Compute the out-degree centrality for nodes.

Eigenvector

`eigenvector_centrality` (G[, max_iter, tol, ...])	Compute the eigenvector centrality for the graph `G`.
`eigenvector_centrality_numpy` (G[, weight, ...])	Compute the eigenvector centrality for the graph G.
`katz_centrality` (G[, alpha, beta, max_iter, ...])	Compute the Katz centrality for the nodes of the graph C
`katz_centrality_numpy` (G[, alpha, beta, ...])	Compute the Katz centrality for the graph G.

Closeness

`closeness_centrality` (G[, u, distance, ...])	Compute closeness centrality for nodes.
`incremental_closeness_centrality` (G, edge[, ...])	Incremental closeness centrality for nodes.

Current Flow Closeness

`current_flow_closeness_centrality` (G[, ...])	Compute current-flow closeness centrality for nodes.
`information_centrality` (G[, weight, dtype, ...])	Compute current-flow closeness centrality for nodes.

(Shortest Path) Betweenness

`betweenness_centrality` (G[, k, normalized, ...])	Compute the shortest-path betweenness centrality for r

igraph

8. Centrality Measures

8.1. `igraph_closeness` — Closeness centrality calculations for some vertices.
8.2. `igraph_harmonic_centrality` — Harmonic centrality for some vertices.
8.3. `igraph_betweenness` — Betweenness centrality of some vertices.
8.4. `igraph_edge_betweenness` — Betweenness centrality of the edges.
8.5. `igraph_pagerank_algo_t` — PageRank algorithm implementation
8.6. `igraph_pagerank` — Calculates the Google PageRank for the specified vertices.
8.7. `igraph_personalized_pagerank` — Calculates the personalized Google PageRank for the specified vertices.
8.8. `igraph_personalized_pagerank_vs` — Calculates the personalized Google PageRank for the specified vertices.
8.9. `igraph_constraint` — Burt's constraint scores.
8.10. `igraph_maxdegree` — The maximum degree in a graph (or set of vertices).
8.11. `igraph_strength` — Strength of the vertices, weighted vertex degree in other words.
8.12. `igraph_eigenvector_centrality` — Eigenvector centrality of the vertices

- Also all (major) network analysis and visualization platforms:

Gephi makes graphs handy

Cytoscape