Graph Metrics and Data Generation Models

lecture-3

Course on Graph Neural Networks (Winter Term 21/22)

Prof. Dr. Holger Giese (holger.giese@hpi.uni-potsdam.de)

Christian Medeiros Adriano (christian.adriano@hpi.de) - **"Chris"**

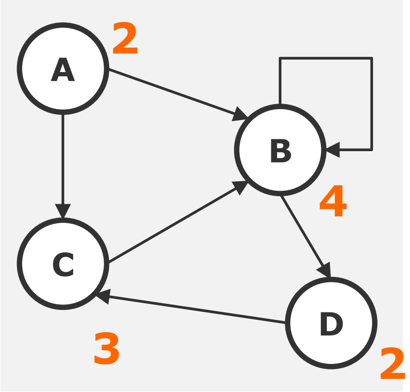
Matthias Barkowski (matthias.barkowski@hpi.de)

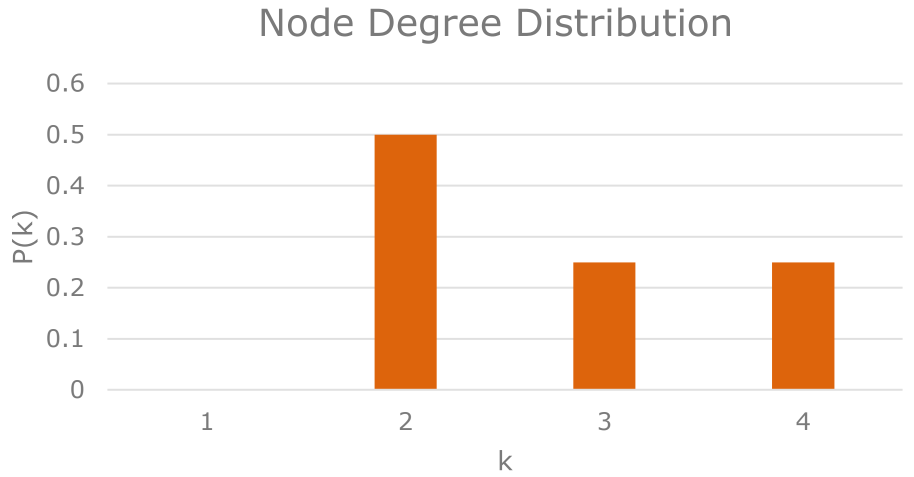
Lecture topics

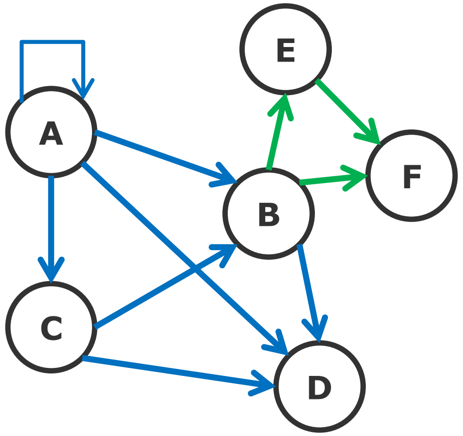
1. Network Metrics
   - Node degree distribution: $P(k)$
   - Network diameter (average shortest path length)
   - Clustering coefficient
   - Connectivity (node distribution across components)
   - Comparing Networks

2. Null Models
   - Threats to Validity
   - Random Graph model
   - Small-World model
   - Kronecker model
   - Deep Generative model

Node Degree Distribution $P(k) = N_k / N$







	A	C	D	B	E	F	∑
A	1	1	1	1	0	0	**4**
C	1	0	1	1	0	0	**3**
D	1	1	0	1	0	0	**3**
B	1	1	1	0	1	1	**5**
E	0	0	0	1	0	1	**2**
F	0	0	0	1	1	0	**2**

Network Diameter (or geodesic distance)

Network diameter $\bar{h}$: average **shortest path** length among all nodes

Path is sequence of nodes that are connected to each other

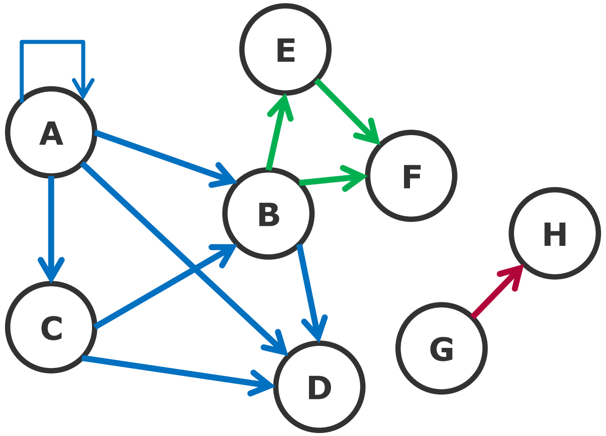
Shortest path h: is the minimal distance between nodes

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

Maximum number of edges:

$$\mathrm{E}_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$

$h_{\mathrm{i,j}}$ is the distance between nodes *i* and *j*



$$h_{A,F} = 2$$
$$h_{F,A} = \infty$$
$$h_{A,G} = \infty$$

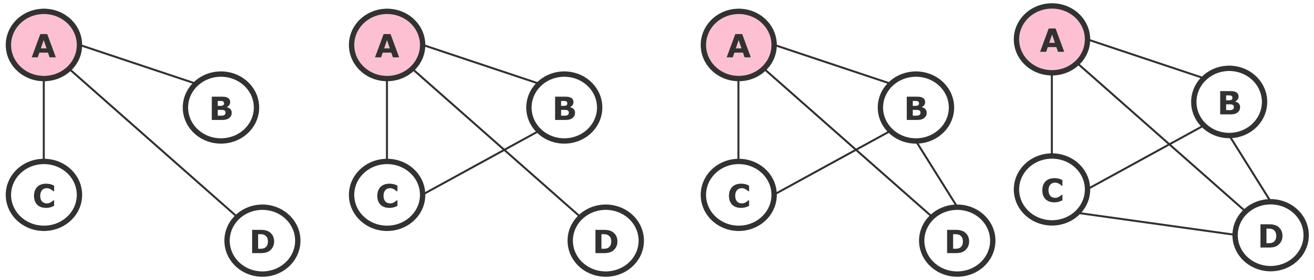
Clustering coefficient C_i

C_i Measures how many of my neighbors are connected to each other

$C_i \in [0,1]$

$$Ci = 2\frac{e_i}{\underbrace{k_i(k_i - 1)}_{\text{total possible edges}}}$$

e_i is the number of edges between neighbors of *i*



C_A	0.0	0.3	0.67	1.0
		=2*1/3*2	=2*2/3*2	=2*3/3*2

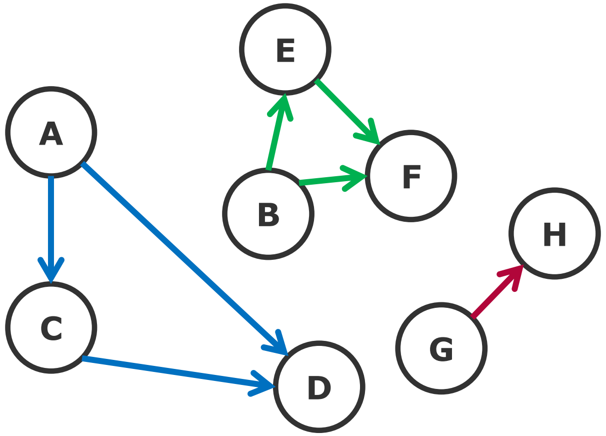
Note: clustering coefficient is undefined for nodes with $k_i = 0 \; or \; 1$

Average Clustering Coefficient

$$C = \frac{1}{N}\sum_{i \in N} C_i$$

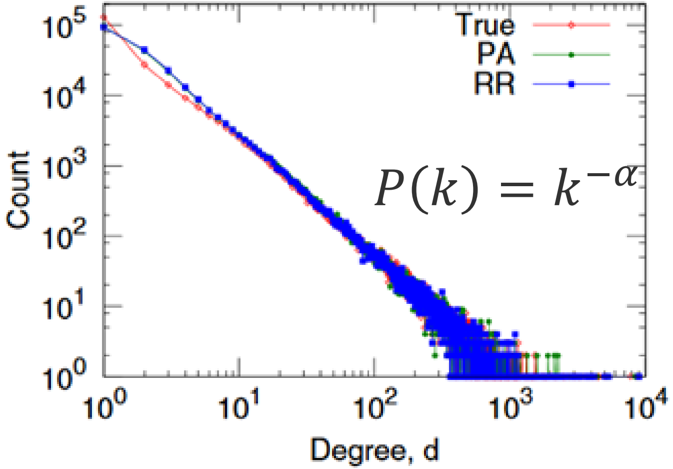
Connectivity S

Connectivity S: is the largest set of nodes that can be connected through any given path

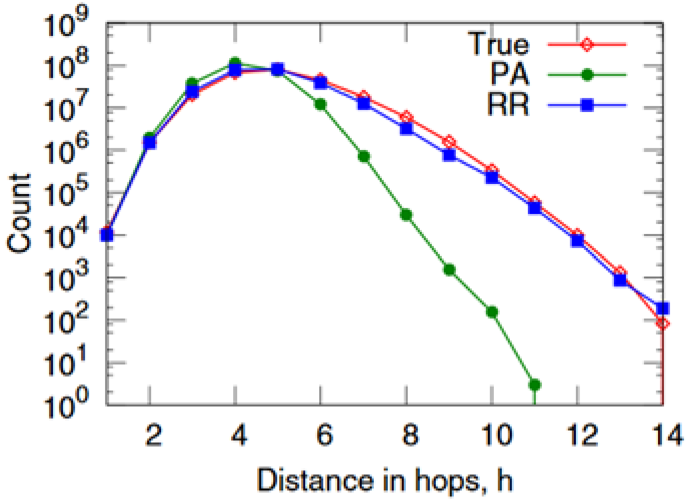


$$S_{blue} = 3$$
$$S_{green} = 3$$
$$S_{red} = 2$$

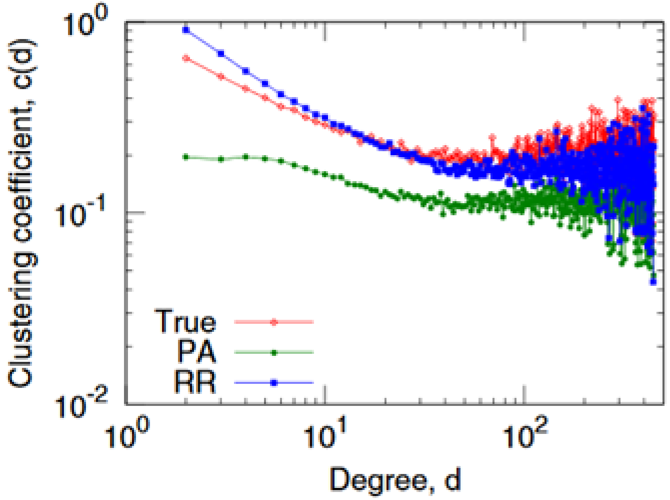
Empirical values (Flickr)



(b) Degree distribution



(c) Geodesic distance



(a) Clustering coefficient

Many nodes with low degree

Few nodes with very high degree

$P(k) = k^{-\alpha}$, i.e., follows a power-law distribution

Typically $\alpha \in [2,3]$

- Web graph [Broderet al. 00]: α in [2.1, 2.4]
- Autonomous systems [Faloutsoset al. 99]:α= 2.4
- Actor collaborations [Barabasi-Albert 00]:α= 2.3
- Citations to papers [Redner98]:α≈3
- Online social networks [Leskovecet al. 07]:α≈2

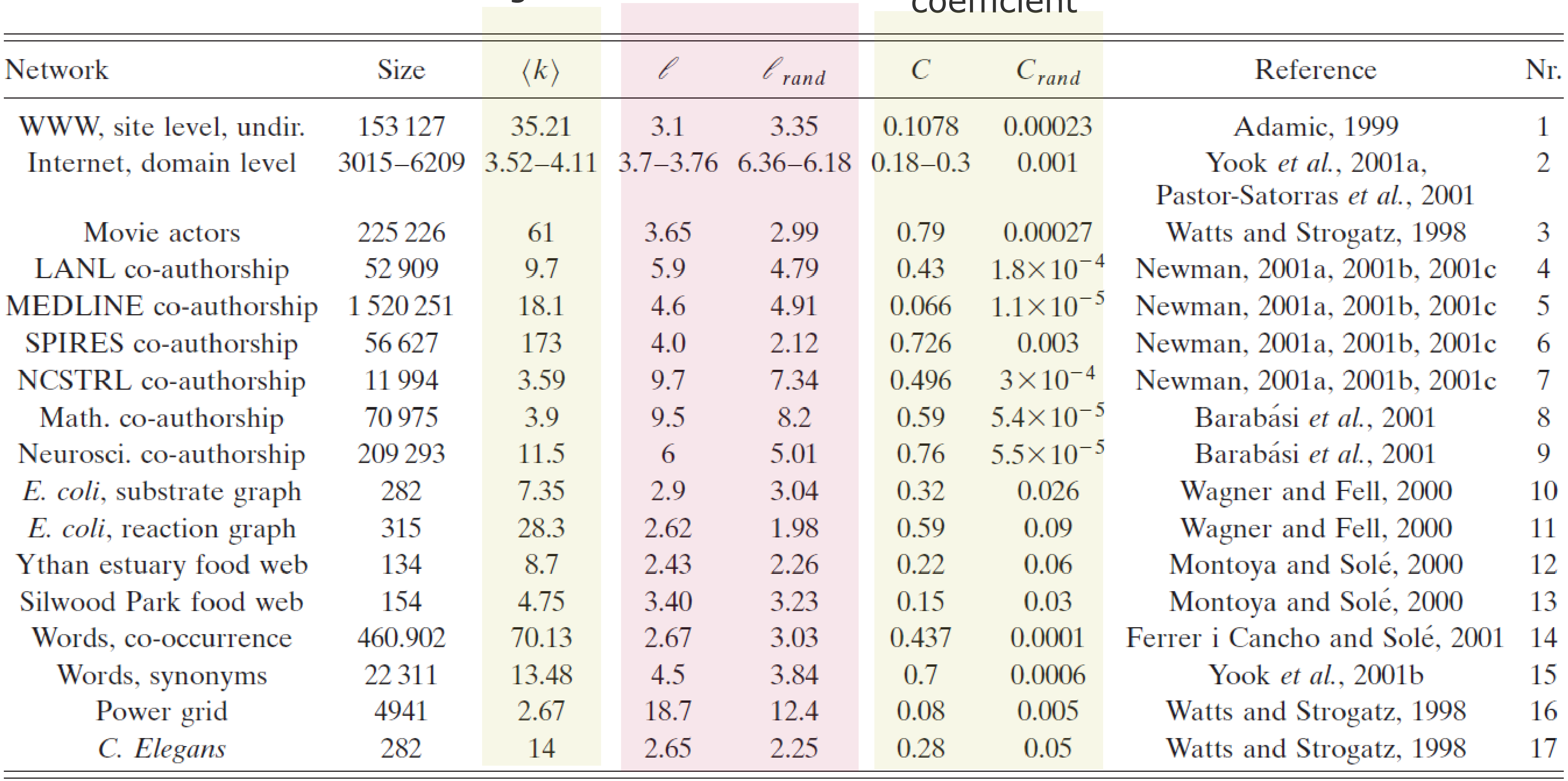
Few hops on average allow to reach anywhere in the network

Low degree nodes tend to have higher clustering

Note: remember to plot degree distributions and clustering coefficients in log-log scale

Source: Leskovec, Jure, et al. "Microscopic evolution of social networks." *Proceedings of the 14th ACM SIGKDD* 2008.

Comparing some networks



		Average degree	Network diameter		Average Clustering coefficient			
Network	Size	$\langle k \rangle$	ℓ	ℓ_{rand}	C	C_{rand}	Reference	Nr.
WWW, site level, undir.	153 127	35.21	3.1	3.35	0.1078	0.00023	Adamic, 1999	1
Internet, domain level	3015–6209	3.52–4.11	3.7–3.76	6.36–6.18	0.18–0.3	0.001	Yook *et al.*, 2001a, Pastor-Satorras *et al.*, 2001	2
Movie actors	225 226	61	3.65	2.99	0.79	0.00027	Watts and Strogatz, 1998	3
LANL co-authorship	52 909	9.7	5.9	4.79	0.43	1.8×10^{-4}	Newman, 2001a, 2001b, 2001c	4
MEDLINE co-authorship	1 520 251	18.1	4.6	4.91	0.066	1.1×10^{-5}	Newman, 2001a, 2001b, 2001c	5
SPIRES co-authorship	56 627	173	4.0	2.12	0.726	0.003	Newman, 2001a, 2001b, 2001c	6
NCSTRL co-authorship	11 994	3.59	9.7	7.34	0.496	3×10^{-4}	Newman, 2001a, 2001b, 2001c	7
Math. co-authorship	70 975	3.9	9.5	8.2	0.59	5.4×10^{-5}	Barabási *et al.*, 2001	8
Neurosci. co-authorship	209 293	11.5	6	5.01	0.76	5.5×10^{-5}	Barabási *et al.*, 2001	9
E. coli, substrate graph	282	7.35	2.9	3.04	0.32	0.026	Wagner and Fell, 2000	10
E. coli, reaction graph	315	28.3	2.62	1.98	0.59	0.09	Wagner and Fell, 2000	11
Ythan estuary food web	134	8.7	2.43	2.26	0.22	0.06	Montoya and Solé, 2000	12
Silwood Park food web	154	4.75	3.40	3.23	0.15	0.03	Montoya and Solé, 2000	13
Words, co-occurrence	460.902	70.13	2.67	3.03	0.437	0.0001	Ferrer i Cancho and Solé, 2001	14
Words, synonyms	22 311	13.48	4.5	3.84	0.7	0.0006	Yook *et al.*, 2001b	15
Power grid	4941	2.67	18.7	12.4	0.08	0.005	Watts and Strogatz, 1998	16
C. Elegans	282	14	2.65	2.25	0.28	0.05	Watts and Strogatz, 1998	17

Source:
Albert, R., & Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, *74*(1), 47.

Other ways of comparing networks

Known Node Correspondence Methods:

- Difference of the adjacency matrices
- DeltaCon (similarity between node pairs)
- Cut distance

Unknown Node Correspondence Methods:

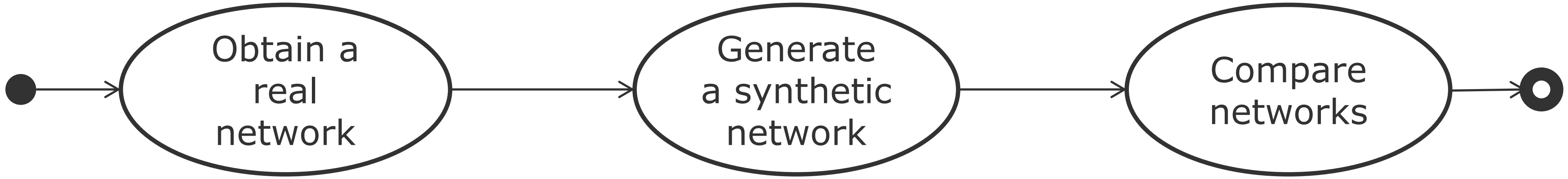
- global statistics,
- matching of subgraphs (graphlets)
  - Relative Graphlets Frequency Distance (RGFD)
  - Graphlet Degree Distribution Agreement (GDDA)
- alignment-based methods entropy measures (isomorphism)
- spectral methods (distance-based)

Source: Tantardini, M., Ieva, F., Tajoli, L., & Piccardi, C. (**2019**). Comparing methods for comparing networks. *Scientific reports*, *9*(1), 1-19.

Null Models

Motivation

We need null-models to compare our graph metrics in a principle and reproducible manner



Null-models are generate by synthetic graph generation procedures.

- Random Graph Model
- Small-World Model
- Kronecker Graph Model

Threats to validity

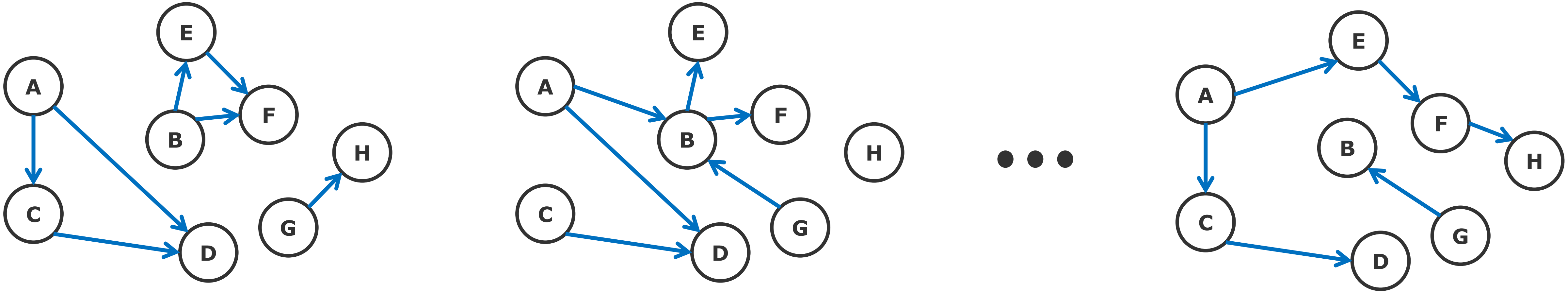
1. **Construct validity:** measurements might not be correct
2. **Conclusion validity:** instruments or methods adopted are not adequate
3. **Internal validity:** relations of cause-effect might not be true
4. **External validity:** results do not generalize to slight changes in the data or context.

Recommended readings:

- Siegmund, J., Siegmund, N., & Apel, S. (2015, May). Views on internal and external validity in empirical software engineering. In Proceedings of the 37th International Conference on Software Engineering-Volume 1 (pp. 9-19). IEEE Press.
- Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Random Graph - Erdös-Rényi (ER) networks

Stochastically connect nodes



Procedure:

For each pair of nodes decide do connect them with a probability p_k

Note: The presence or absence of an edge between two vertices is independent of the presence or absence of any other edge, probabilities p_{k_i} are independent from each other.

Source: Newman, Mark EJ, Steven H. Strogatz, and Duncan J. Watts. "Random graphs with arbitrary degree distributions and their applications." *Physical review E* 64.2 (2001): 026118.

Random Graph – Distribution of Node Degree

Distribution of Node Degrees

z = average number of edges
k = degree of an edge

$$p_k = \binom{N}{k} p^k (1-p)^{N-k} \simeq \frac{z^k e^{-z}}{k!},$$

In the limit when N is very large.
i.e., a Poisson distribution

For the binomial distribution:
Mean $\bar{k} = p(N-1)$
Variance $\sigma^2 = p(1-p)(N-1)$

Newman, Mark EJ, Steven H. Strogatz, and Duncan J. Watts. "Random graphs with arbitrary degree distributions and their applications." *Physical review E* 64.2 (2001): 026118.

Random Graph – Clustering coefficient

Clustering coefficient:

$$C_i = 2\,\frac{(Numer\ of\ Triangles\ containind\ node\ i)}{k_i(k_i - 1)}$$

$$\mathrm{E}[e] = p\,\frac{k_i(k_i-1)}{2}$$

Expected local clustering coefficient = p

$$E[C_i] = p\,\frac{k_i(k_i-1)}{k_i(k_i-1)} = p = \frac{\bar{k}}{N-1} \sim \frac{\bar{\boldsymbol{k}}}{\boldsymbol{N}}$$

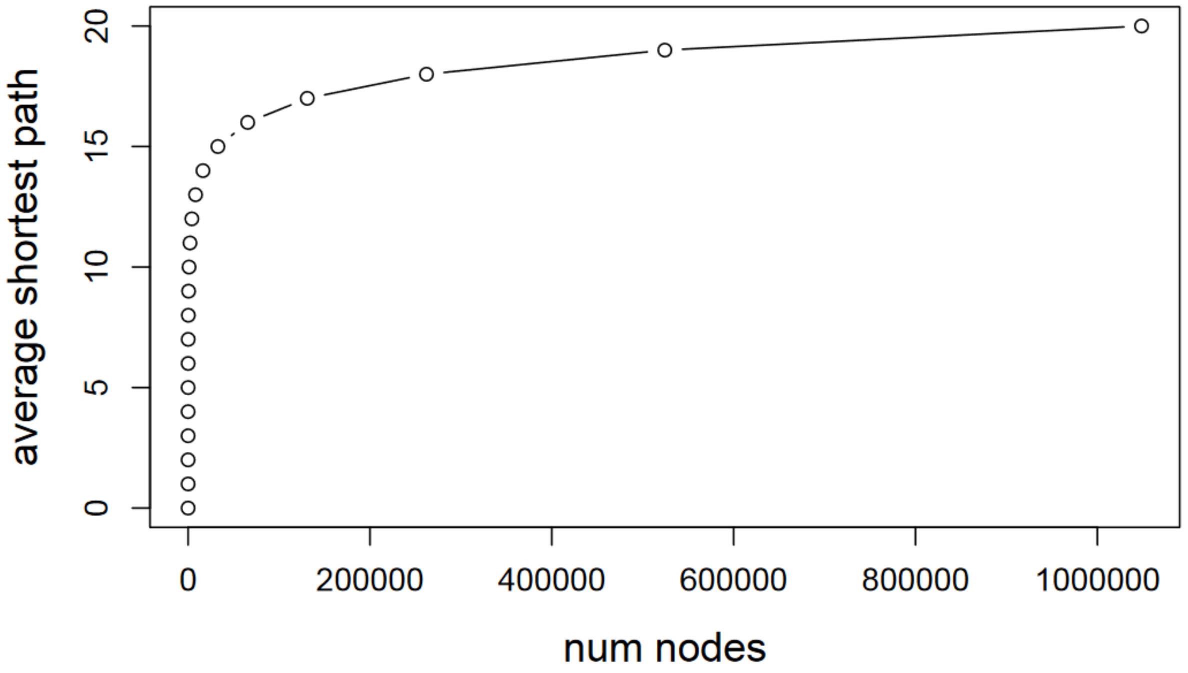
This means that the clustering coefficient of a random graph is small.

Because as we generate bigger random graphs with a fixed average degree *k*, *i.e.*, *we* set p=k·1/N), C will decrease with the graph size N

Random Graph – Network Diameter

Network Diameter (avg shortest path)

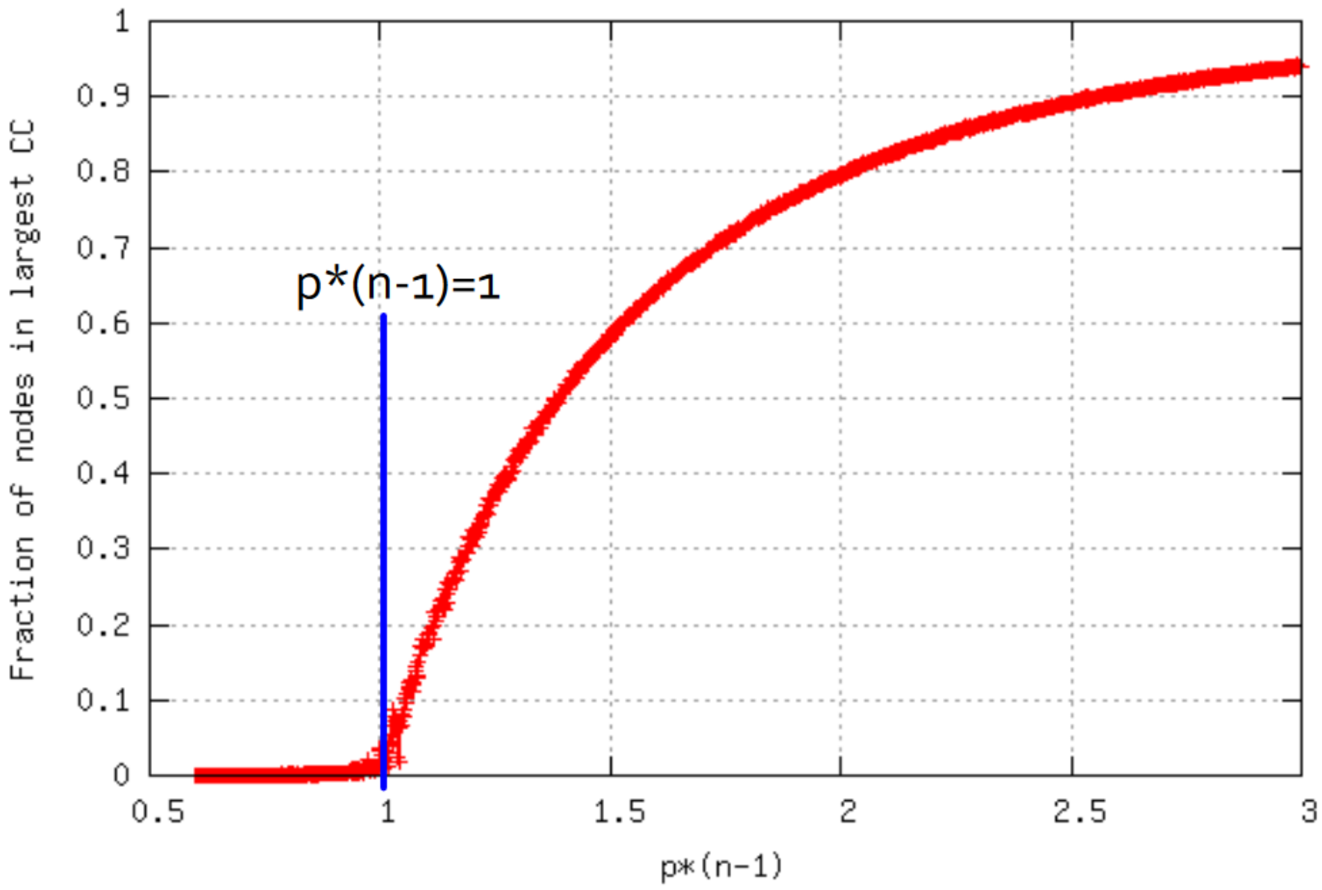
$= O(\log n)$



[Leskovec 2019]

Average degree constant

Random Graph – Connectivity



Tend to have giant components

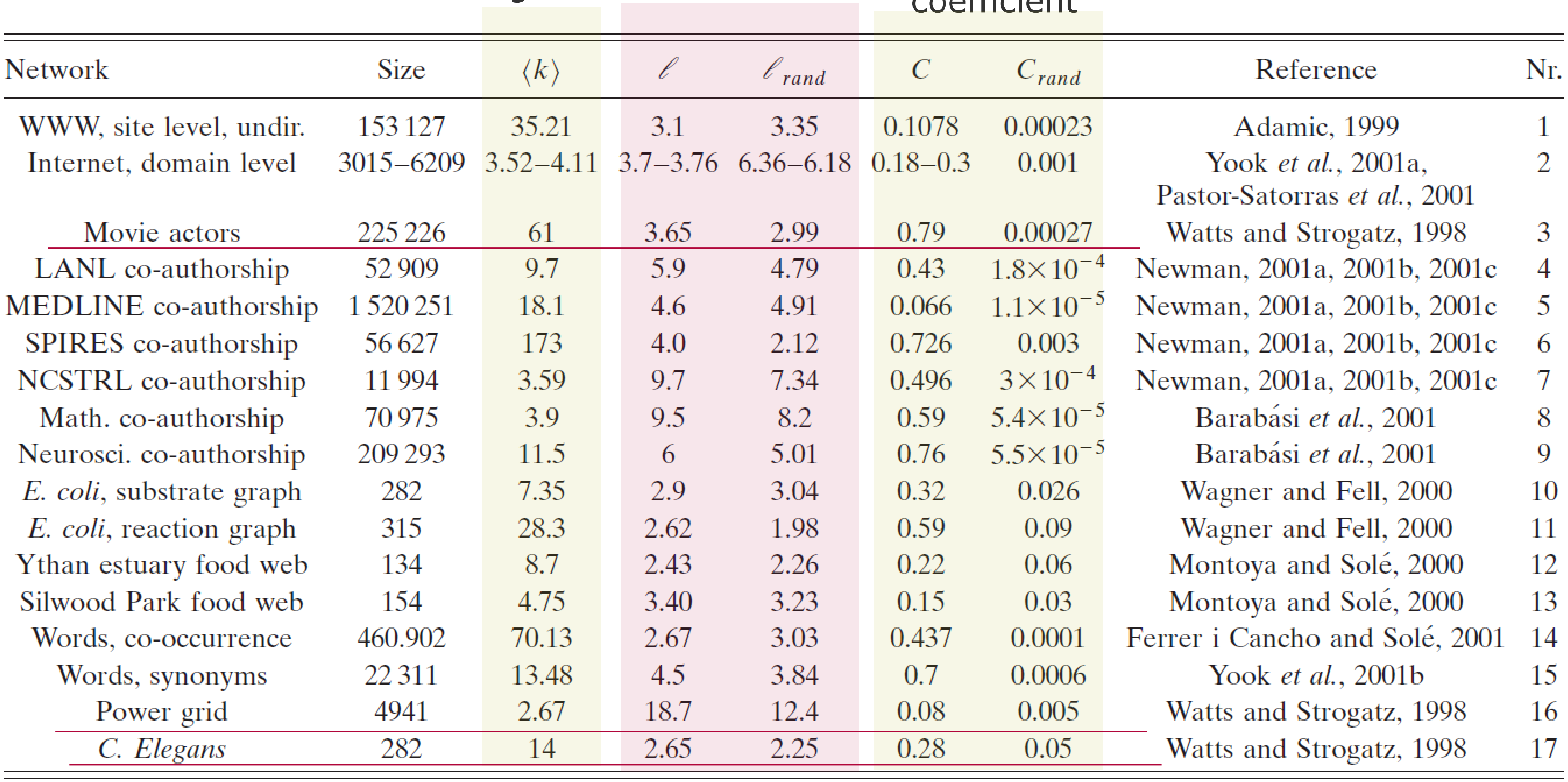
Everything gets connected.

Disconnected graph

Fully connected graph

Average degree $\bar{k}$

Again - Comparing some networks

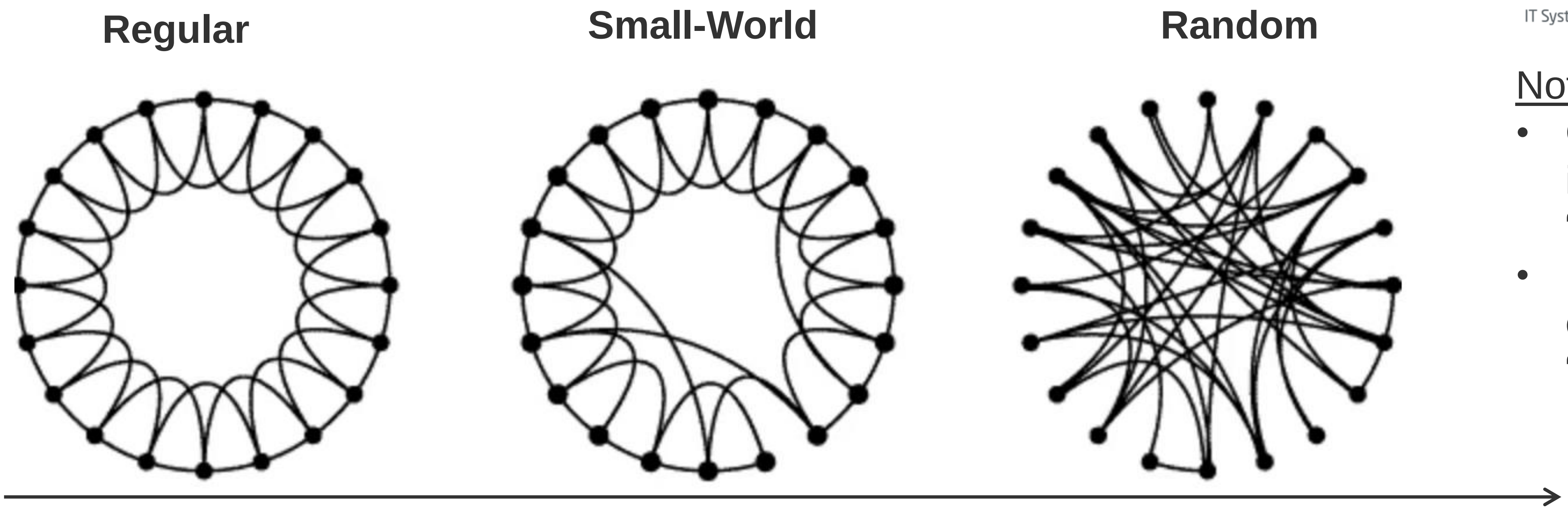


		Average degree	Network diameter		Average Clustering coefficient			
Network	Size	$\langle k \rangle$	ℓ	ℓ_{rand}	C	C_{rand}	Reference	Nr.
WWW, site level, undir.	153 127	35.21	3.1	3.35	0.1078	0.00023	Adamic, 1999	1
Internet, domain level	3015–6209	3.52–4.11	3.7–3.76	6.36–6.18	0.18–0.3	0.001	Yook *et al.*, 2001a, Pastor-Satorras *et al.*, 2001	2
Movie actors	225 226	61	3.65	2.99	0.79	0.00027	Watts and Strogatz, 1998	3
LANL co-authorship	52 909	9.7	5.9	4.79	0.43	1.8×10^{-4}	Newman, 2001a, 2001b, 2001c	4
MEDLINE co-authorship	1 520 251	18.1	4.6	4.91	0.066	1.1×10^{-5}	Newman, 2001a, 2001b, 2001c	5
SPIRES co-authorship	56 627	173	4.0	2.12	0.726	0.003	Newman, 2001a, 2001b, 2001c	6
NCSTRL co-authorship	11 994	3.59	9.7	7.34	0.496	3×10^{-4}	Newman, 2001a, 2001b, 2001c	7
Math. co-authorship	70 975	3.9	9.5	8.2	0.59	5.4×10^{-5}	Barabási *et al.*, 2001	8
Neurosci. co-authorship	209 293	11.5	6	5.01	0.76	5.5×10^{-5}	Barabási *et al.*, 2001	9
E. coli, substrate graph	282	7.35	2.9	3.04	0.32	0.026	Wagner and Fell, 2000	10
E. coli, reaction graph	315	28.3	2.62	1.98	0.59	0.09	Wagner and Fell, 2000	11
Ythan estuary food web	134	8.7	2.43	2.26	0.22	0.06	Montoya and Solé, 2000	12
Silwood Park food web	154	4.75	3.40	3.23	0.15	0.03	Montoya and Solé, 2000	13
Words, co-occurrence	460.902	70.13	2.67	3.03	0.437	0.0001	Ferrer i Cancho and Solé, 2001	14
Words, synonyms	22 311	13.48	4.5	3.84	0.7	0.0006	Yook *et al.*, 2001b	15
Power grid	4941	2.67	18.7	12.4	0.08	0.005	Watts and Strogatz, 1998	16
C. Elegans	282	14	2.65	2.25	0.28	0.05	Watts and Strogatz, 1998	17

- Actors
- Power grid
- C. elegans

Source:
Albert, R., & Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, *74*(1), 47.

Small-World Model = high clustering + short paths



	p =0		p=1
Clustering	High	High	Low
Diameter	High	Low	Low

p = probability of reconnecting uniformly

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

Procedure:
1. start with a ring of *n* vertices, each connected to its *k*-nearest neighbors by undirected edges.
2. choose a node and the edge that connects it to its nearest neighbor in a clockwise.
3. reconnect with probability p this edge to a node chosen uniformly over the entire ring

Source: Watts, Duncan J., and Steven H. Strogatz. "Collective dynamics of ‘small-world’networks." *nature* 393.6684 (1998): 440-442.

Kronecker Model

Small-World model captures the structure of many realistic networks

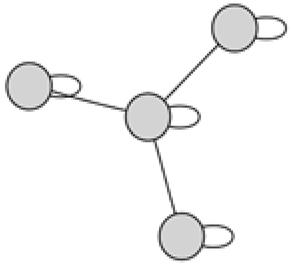
However, it does not produce the correct degree distribution

Solution: use the idea do self-similarity (the whole is in the parts)

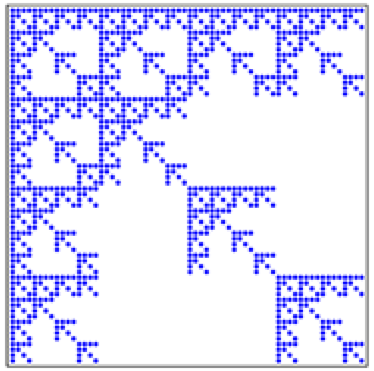
Initiator K_1

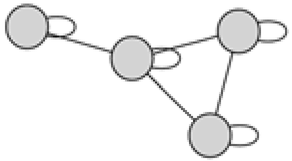
K_1 adjacency matrix

K_3 adjacency matrix

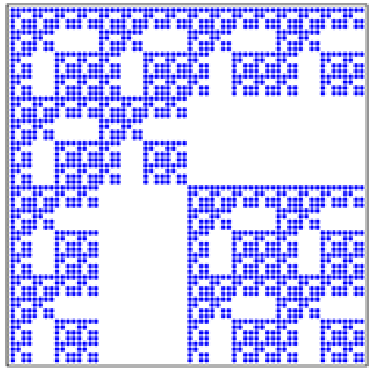


1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1





1	1	1	1
1	1	0	0
1	0	1	1
1	0	1	1



Code to generate:
https://github.com/BenjaminDHorne/Stochastic-Kronecker-Generator

Source: Leskovec, Jure, et al. "Kronecker graphs: an approach to modeling networks." *Journal of Machine Learning Research* 11.2 (2010).

Deep Generative Models

Given a distribution $P_{data}(X)$

1. Learn a model of this data $P_{model}(X;\theta)$
2. Generate new graphs by sampling from this $P_{model}(X;\theta)$

1. How to learn $\boldsymbol{P_{model}}(X;\theta)$?
Optimize the parameters θ to approximate the $P_{data}(X)$

Maximum Likelihood: $\theta^* = \arg\max_{\theta} E_{x \sim P_{data}} \log P_{model}(X|\theta)$,

which means to find parameters θ^* so that for the observed datapoints $x_i \sim P_{data}(X)$, the $\sum_i \log P_{model}(x_i;\theta^*)$ has the highest value, among all possible choices of θ

2. How to sample from $\boldsymbol{P_{model}}(X;\theta)$?

2.1 sample from a normal distribution

$$z_i = N(\mu = 0, \sigma = 1)$$

2.2 transform the noise z_i via a function f
$x_i = f(z_i, \theta)$, so x_i will follow a complex function f.

How to determine f?
Use a deep neural network to train it, for instance and Recurrent Neural Network (auto-regressive model)

Deep Generative Models - Challenge

The goal of learning generative models of graphs is to learn a distribution $P_{model}(G)$ over graphs,

Based on a set of observed graphs $G \ = \{G1, \ldots, Gs\}$ sampled from data distribution $P(G)$,

Where each graph G_i may have a different number of nodes and edges.

When representing G ∈ set of G, we further assume that we may observe any node ordering π with equal probability, i.e., $P(\pi)$ = 1/n! ,∀π ∈ Π.

Therefore, the generative model needs to be capable of generating graphs even when each graph could have exponentially many representations,

This is clearly distinct and more challenging than previous generative models for images, text, and time series…

Deep Generative Model

How to sample from $P\_model(X;\theta)$?

In auto-regressive model $P\_model(X,\theta)$ is used for density estimation and for sampling

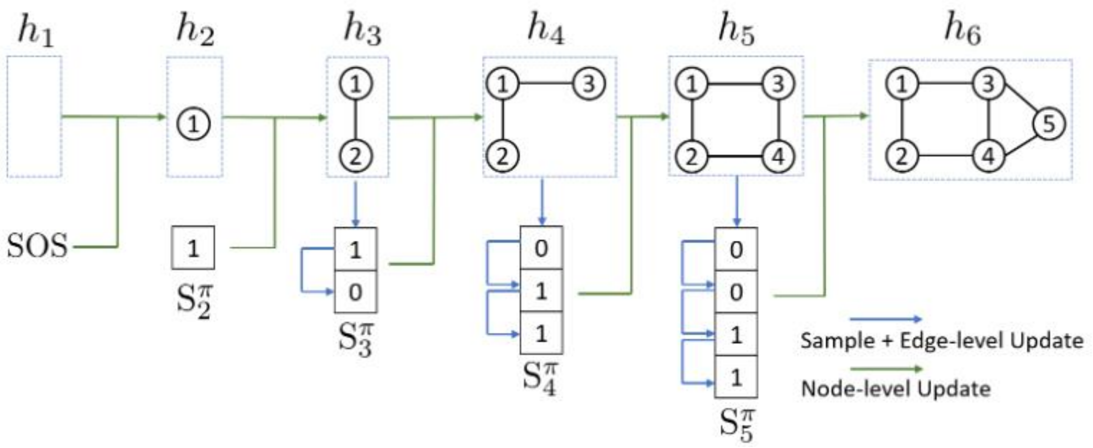
Relies on the Chain Rule

$$P_{model(X,\theta)} = \prod_{t=1\ in\ N} P\_model(x_t | x_1, \dots, x_{t-1}; \theta)$$

where x is a vector and t is the *t-th* dimension, for instance, if x is a sentence, x_t is *t-th* word.

In the case of graph generation, x is an action of adding a node or an edge.

Graph RNN [You et al. 2018]



A common way to represent a graph is using an adjacency matrix A. This requires a node ordering π that maps nodes to rows/columns of the adjacency matrix.

More specifically, π is a permutation function over nodes V $(i.e., (\pi(v1), \dots, \pi(vn))$ is a permutation of (v1,...,vn)).

Graph $G \sim p(G)$ with n nodes under node ordering π

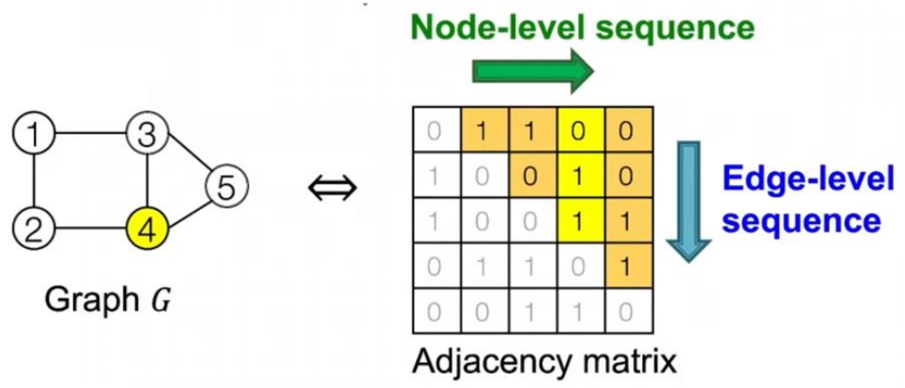
Graph Sequence definition:

$$S^\pi = f_S(G, \pi) = (S_1^\pi, \dots, S_n^\pi),$$

where each element $S_i^\pi \in \{0,1\}^{i-1}, i \in \{1, \dots, n\}$ is an adjacency vector representing the edges between node $\pi(v_i)$ and the previous nodes $\pi(v_j), j \in \{1, \dots, i-1\}$ that are already in the graph

$$S_i^\pi = (A_{1,i}^\pi, \dots, A_{i-1,i}^\pi)^T, \forall i \in \{2, \dots, n\}.$$

Adjacency matrices



0	1	1	0	0
1	0	0	1	0
1	0	0	1	1
0	1	1	0	1
0	0	1	1	0

Adjacency matrix

Sources:
- You, J., et al., 2018, GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models, in proc. of the 35th International Conference on Machine Learning
- Jures Leskovec, slides CS224W: Machine Learning with Graphs | 2021 | Lecture 15.2

Next steps

Remember to:

- Browse over a few datasets
- Try one or two examples using Snap or NetworkX
- Look at how networks compare w.r.t. metrics, do you see something surprising?

END