Theoretical Network Models

Dr. Natarajan Meghanathan
Professor of Computer Science
Jackson State University, Jackson, MS
E-mail: natarajan.meghanathan@jsums.edu

Theoretical Network Models

- In this module, we will see the theoretical models using which one could generate complex networks resembling those of real-world networks.

- The theoretical models we will see are as follows
  - Random Networks
    - Erdos Renyi Model (ER Model)
  - Scale-Free Networks
    - Barabasi Albert Model (BA Model)
    - Bianconi-Barabasi (BB Model)
  - Small-World Networks
    - Watts-Strogatz Model (WS Model)

- We will see the generation of the networks based on the above models and analyze their characteristics.

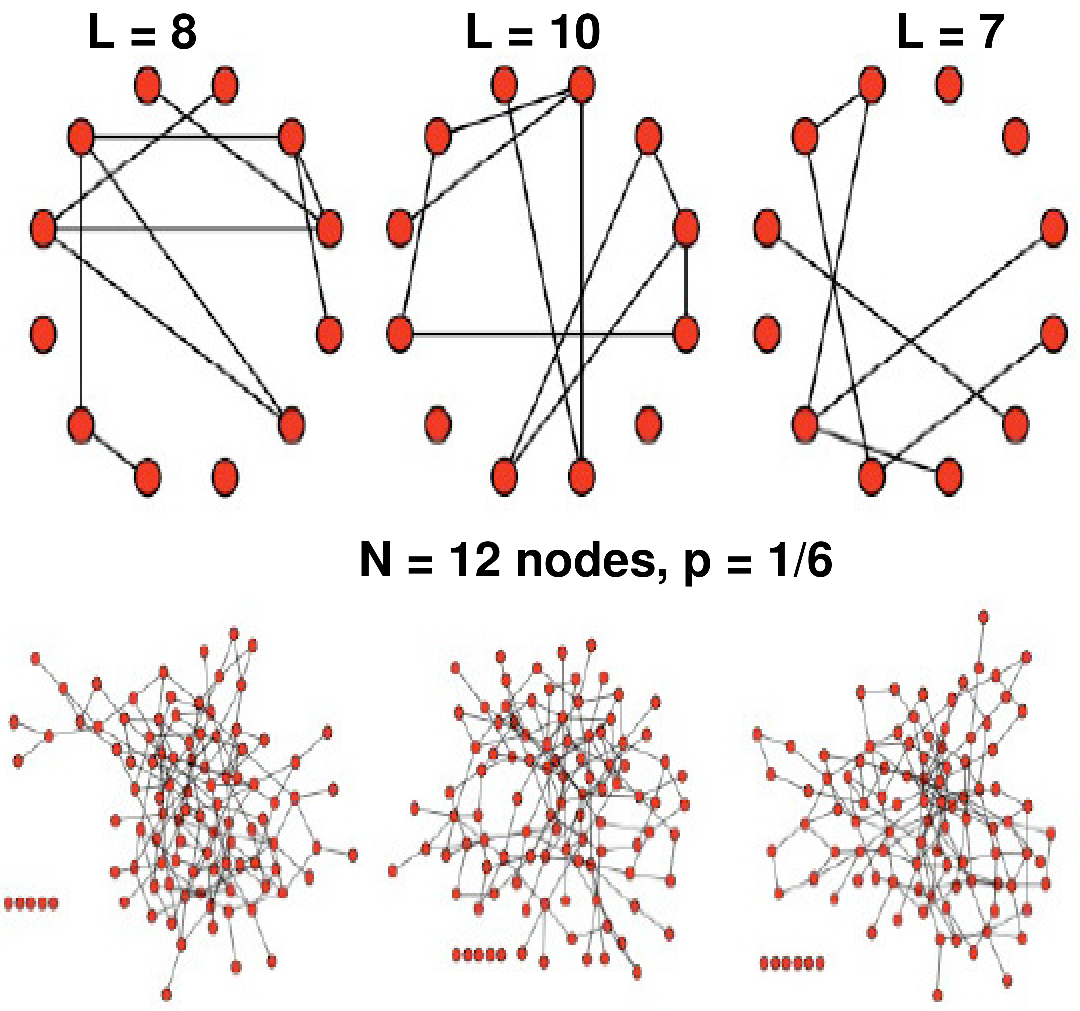
Random Networks

- A random network is the one in which there is a certain probability for a link between any two nodes in the network.
- Erdos-Renyi Model (ER) Model
  - The probability for a link between any two nodes is the same.
  - Called the G(N, p) model where N is the number of nodes and ‘p’ is the probability for a link between any two nodes
  - Highly theoretical model and it is primarily used to determine whether the links in a real-world network are formed due to random interactions of nodes or due to the preference of nodes to communicate or attach to certain nodes.

ER Model

- Step 1: Start with N isolated nodes
- Step 2: For a particular node pair (u, v), generate a random number *r*. If $r \leq p$, then, add the link (u, v) to the network.
- Repeat Step 2 for each of the N(N-1)/2 node pairs.

- Each random network we generate with the same parameters (N, p) will look slightly different.
  - The number of links L is likely to be different.



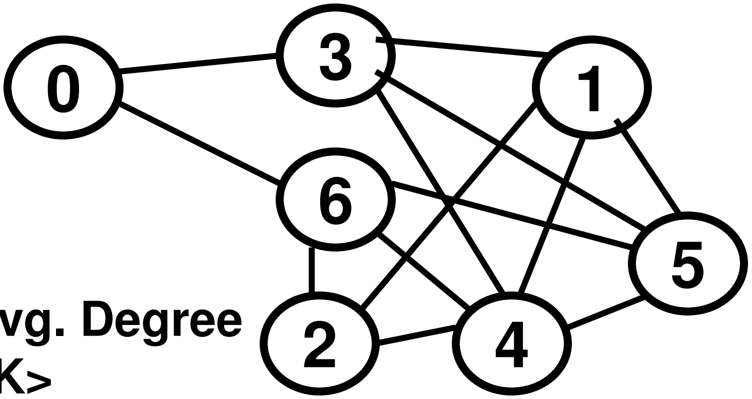
N = 12 nodes, p = 1/6

N = 100 nodes, p = 1/6

Source: Figure 3.3a: Barabasi

Generation of ER-Random Network

Let plink = 0.524



Avg. Degree
<K>
= (2 + 4 + 3 + 4 + 5 + 4 + 4) / 7
= 3.71

Index	Pairs	Random Val	Edge
1	0, 1	0.6335	N
2	0, 2	0.7478	N
3	0, 3	0.1721	Y
4	0, 4	0.9234	N
5	0, 5	0.8563	N
6	0, 6	0.3141	Y
7	1, 2	0.1594	Y
8	1, 3	0.2945	Y
9	1, 4	0.2227	Y
10	1, 5	0.0343	Y
11	1, 6	0.7621	N
12	2, 3	0.8595	N
13	2, 4	0.3091	Y
14	2, 5	0.5312	N
15	2, 6	0.1834	Y
16	3, 4	0.4194	Y
17	3, 5	0.2549	Y
18	3, 6	0.6974	N
19	4, 5	0.0968	Y
20	4, 6	0.4486	Y
21	5, 6	0.2983	Y

ER Model: Poisson Degree Distribution

- In a network of N nodes, the maximum number of links for a node with its neighbors is N-1 and each of these links can occur with a probability p.
  - Average degree of a node <K> = (N-1)p
  - Standard deviation for the degree of a node σ_K = sqrt(<K>)

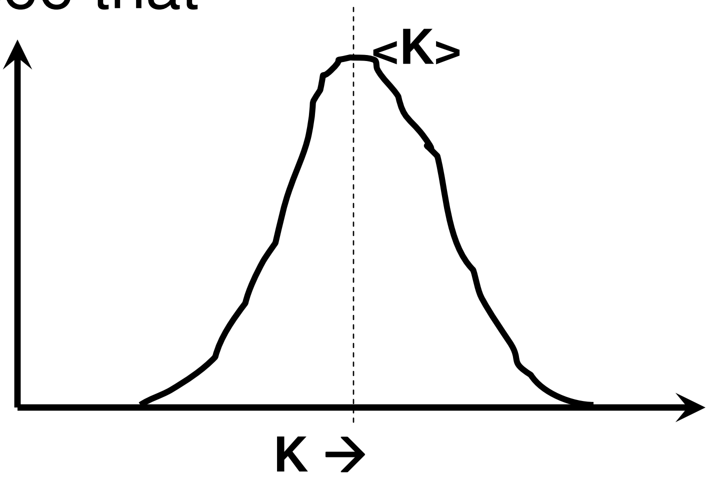
For larger networks, we simply use: <K> ~ Np

- There could be a maximum N(N-1)/2 links in a random network of N nodes and each of these links can occur with a probability p.
  - Average number of links <L> = {N(N-1)/2}*p
- From the above, we can easily see that

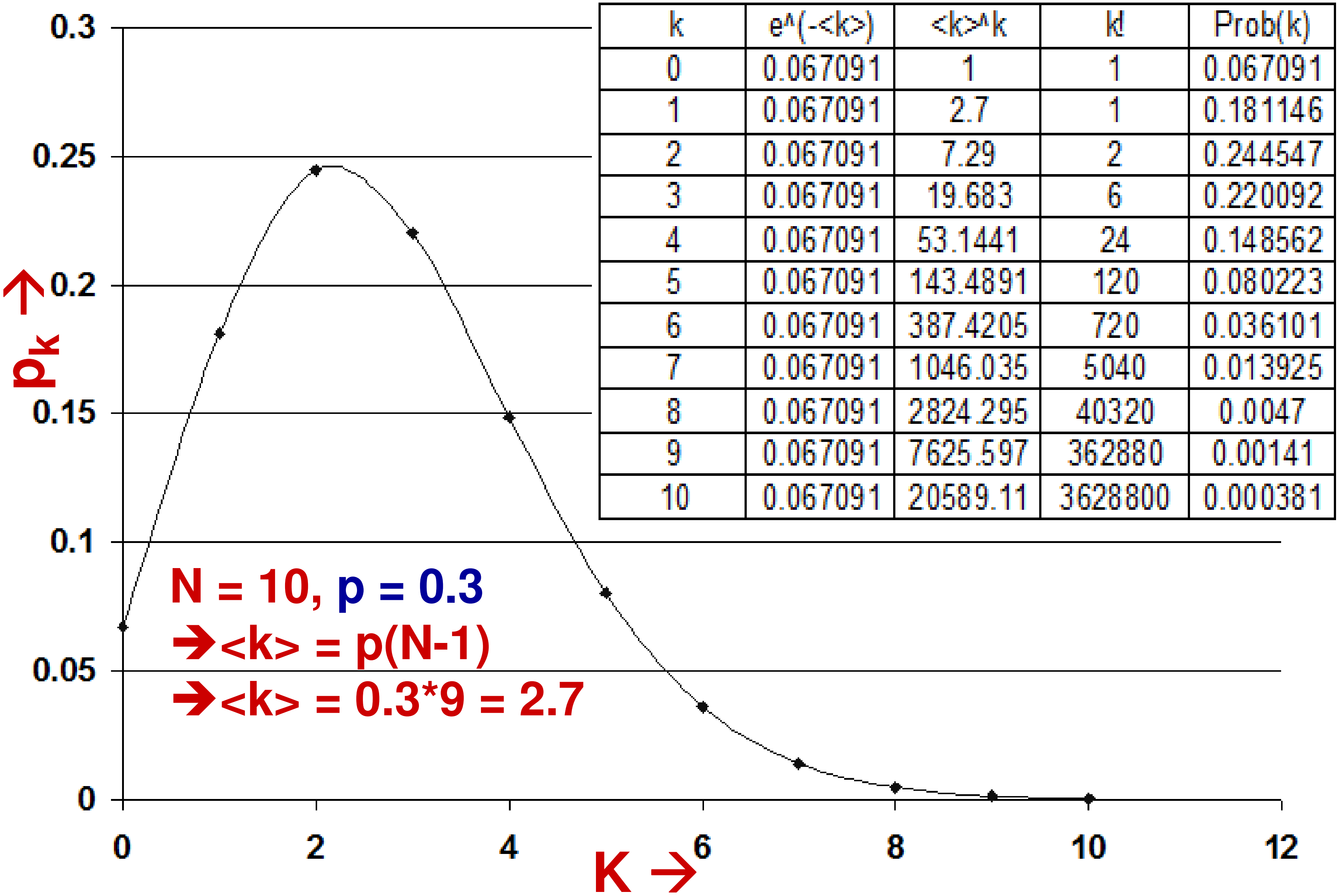
<K> = 2 * <L> / N

Poisson degree distribution

$$\boxed{p_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}}$$



Generating a Poisson Degree Distribution



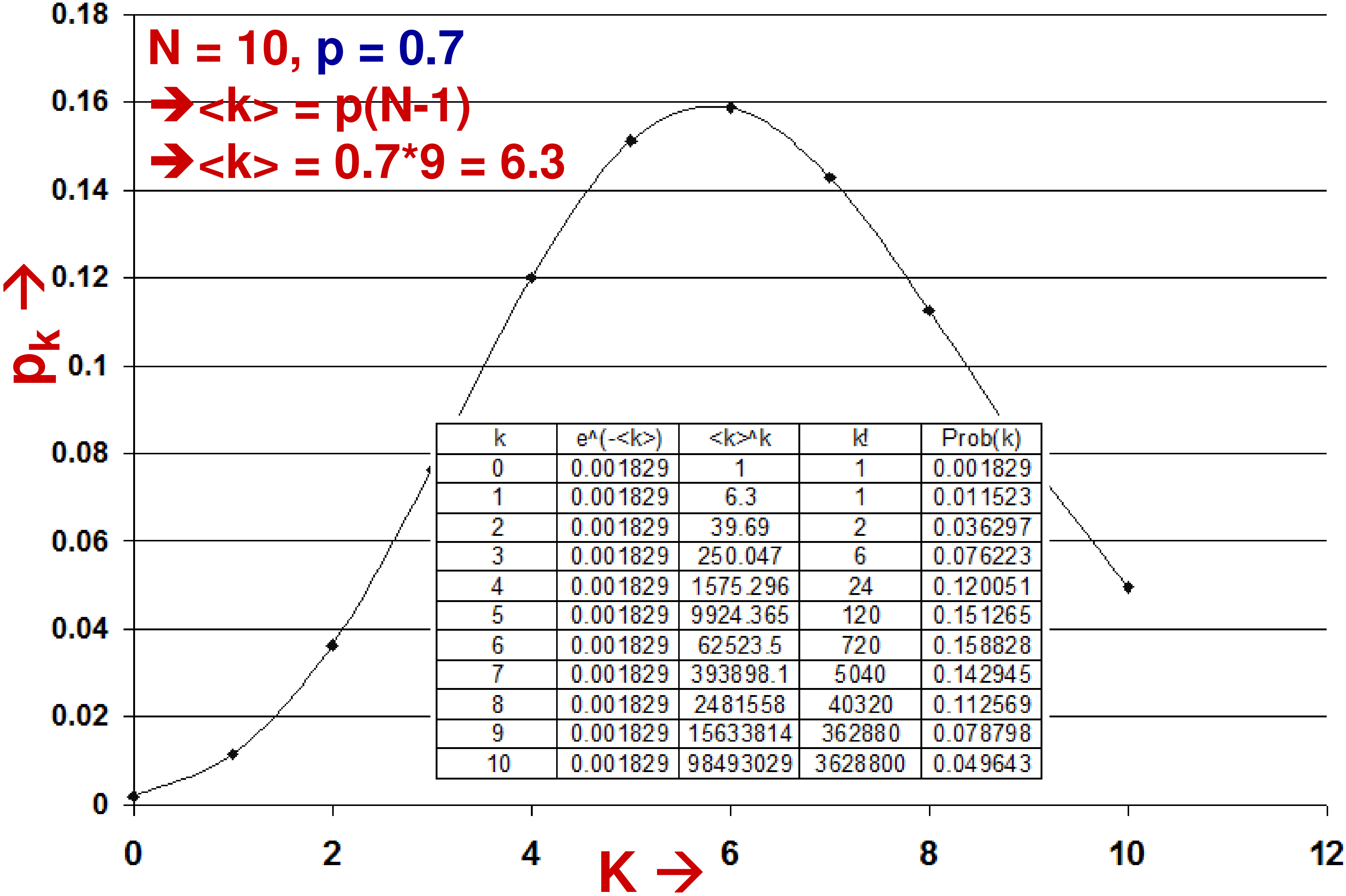
k	e^(-<k>)	<k>^k	k!	Prob(k)
0	0.067091	1	1	0.067091
1	0.067091	2.7	1	0.181146
2	0.067091	7.29	2	0.244547
3	0.067091	19.683	6	0.220092
4	0.067091	53.1441	24	0.148562
5	0.067091	143.4891	120	0.080223
6	0.067091	387.4205	720	0.036101
7	0.067091	1046.035	5040	0.013925
8	0.067091	2824.295	40320	0.0047
9	0.067091	7625.597	362880	0.00141
10	0.067091	20589.11	3628800	0.000381

N = 10, p = 0.3

→<k> = p(N-1)

→<k> = 0.3*9 = 2.7

Generating a Poisson Degree Distribution



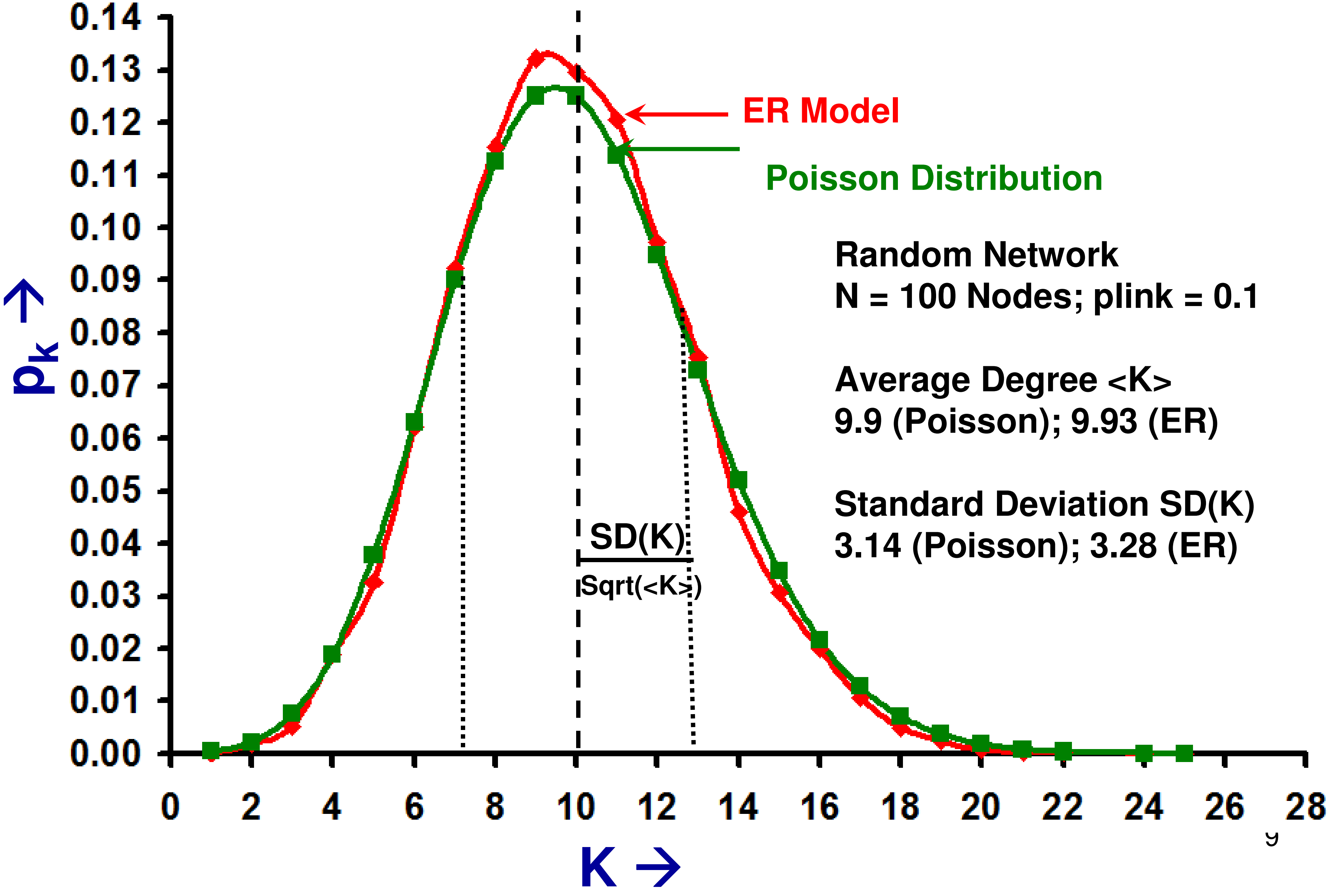
N = 10, p = 0.7

➔<k> = p(N-1)

➔<k> = 0.7*9 = 6.3

k	e^(-<k>)	<k>^k	k!	Prob(k)
0	0.001829	1	1	0.001829
1	0.001829	6.3	1	0.011523
2	0.001829	39.69	2	0.036297
3	0.001829	250.047	6	0.076223
4	0.001829	1575.296	24	0.120051
5	0.001829	9924.365	120	0.151265
6	0.001829	62523.5	720	0.158828
7	0.001829	393898.1	5040	0.142945
8	0.001829	2481558	40320	0.112569
9	0.001829	15633814	362880	0.078798
10	0.001829	98493029	3628800	0.049643

Degree Distribution: ER Model vs. Poisson



Random Network
N = 100 Nodes; plink = 0.1

Average Degree <K>
9.9 (Poisson); 9.93 (ER)

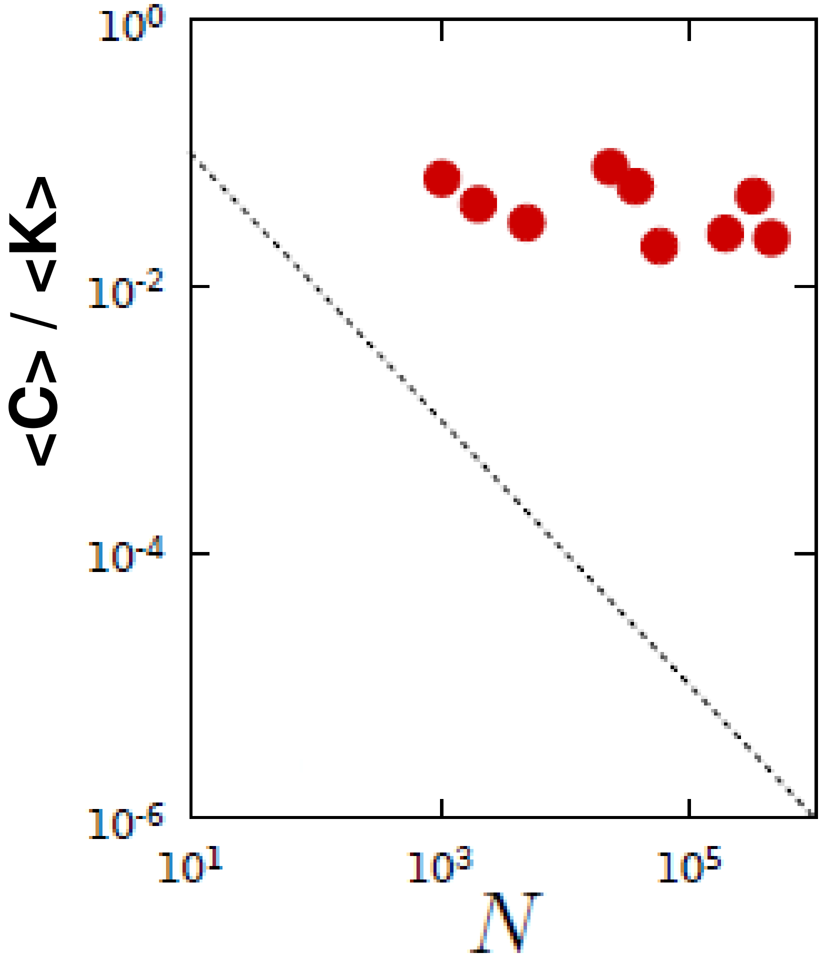
Standard Deviation SD(K)
3.14 (Poisson); 3.28 (ER)

Clustering Coefficient

- In a random network evolved under the ER: G(N, p) model:
  - For a node *i* with k_i neighbors, the expected number of links connecting the neighbors is $p*k_i(k_i -1)/2$.
  - Clustering coefficient is the ratio of the actual (also the expected value) number of links to that of the maximum number of links connecting the neighbors.
  - Thus, the average clustering coefficient <C> for an ER: G(N, p)-based random network is simply 'p' = <C> = <K> / N.
  - Unlike real-world networks, the clustering coefficient is not dependent on degree distribution.

• Networks	Actual	Random: ER-G(N, p)
Prison Friendships	0.31	0.0134
Co-authorships		
Math	0.15	0.00002
Biology	0.09	0.00001
Economy	0.19	0.00002
WWW		
Web links	0.11	0.002

Clustering Coefficients for Real Networks



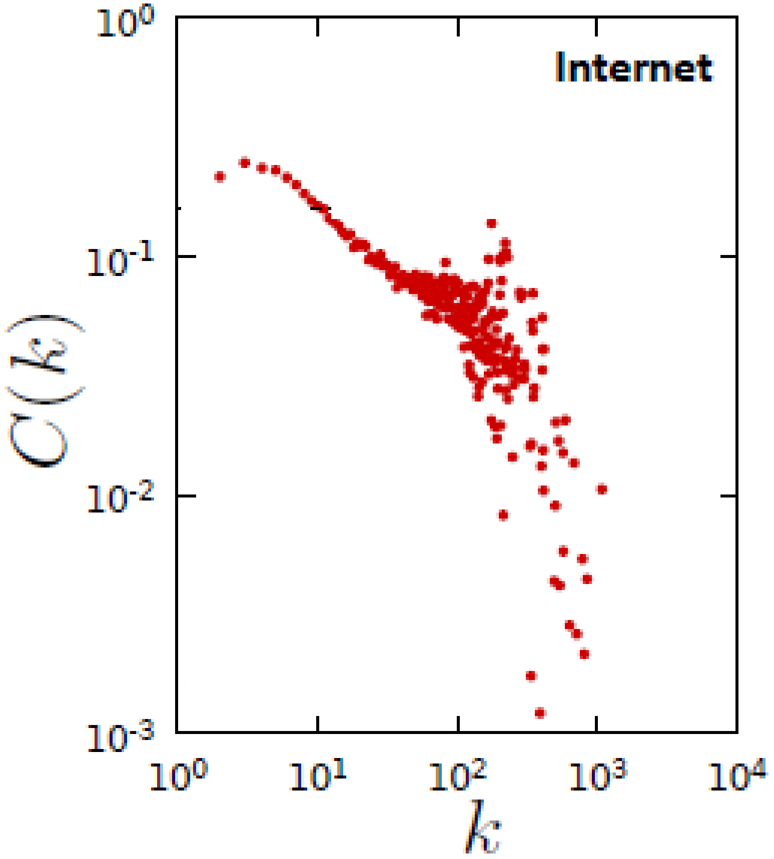
Each circle corresponds to a real network.

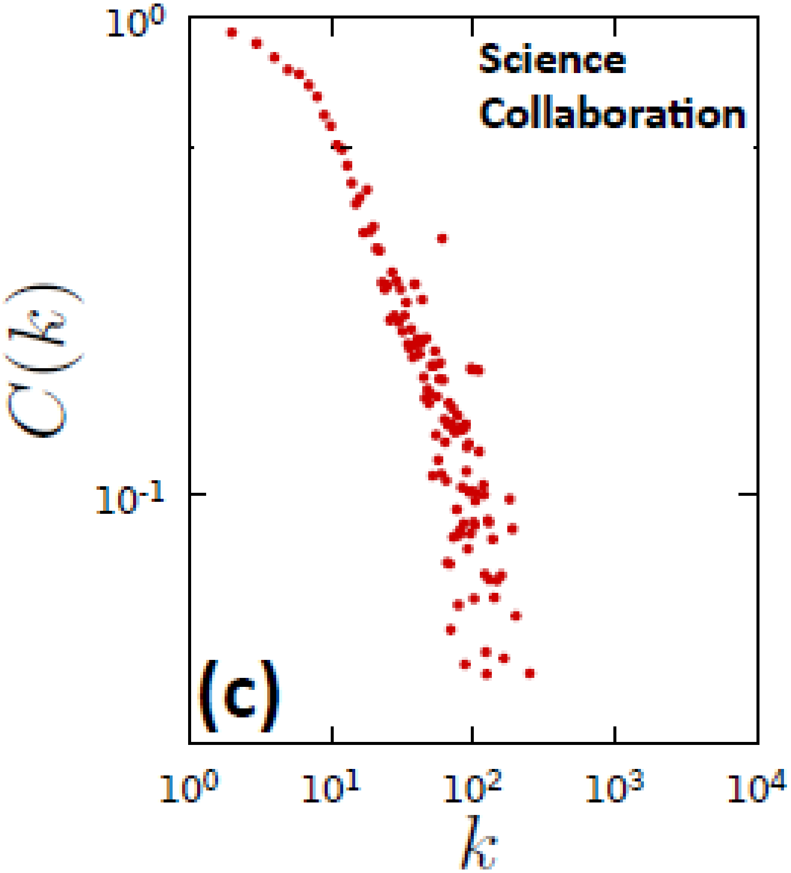
Directed networks were made undirected to calculate C.

For ER-random networks, the average clustering coefficient decreases as 1/N. In contrast, for real networks, <C> has only a weak dependence on N.

Real networks have a much higher Clustering coefficient than expected for a ER-random network of similar N and L.

Clustering for Real Networks





C(k) is measured by averaging the local clustering coefficient of all nodes with the same degree k.

According to the ER-Random Network theory model, C(k) is independent of the individual node degrees. However, we find that C(k) decreases as k increases. Nodes with fewer neighbors have larger local clustering coefficients and vice-versa

Real Networks are not ER-Random

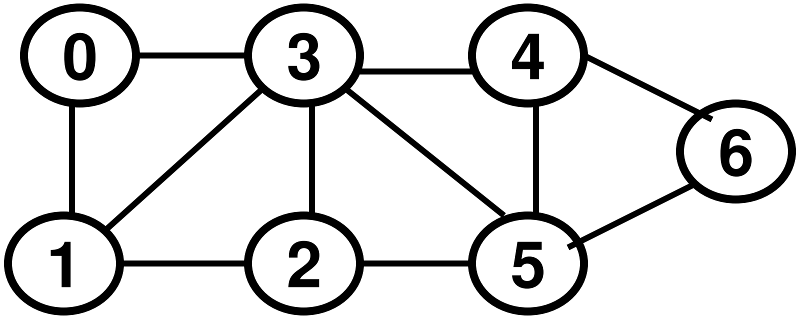
- Degree distribution:
  - ER-Random networks –Poisson distribution, esp. for k << N.
    - Highly connected nodes (hubs) are effectively forbidden.
  - Real networks: More highly connected nodes, compared to that predicted with random model.
- Connectedness:
  - ER-Random networks: One single giant component exists only if <k> > ln N (i.e., p > (ln N)/N)
  - Real networks: One single giant component exists for several networks with <k> < ln N.
- Average Path Length (small world property):
  - For both ER-random and real networks, the average path length scales as ln N / ln <k>.
- Clustering coefficient:
  - ER-Random model: Local clustering coefficient is independent of the node's degree and <C> depends on the system size as 1/N.
  - Real networks: <C> decreases with increase in node degrees and is largely independent of the system size.

Real Networks are not ER-Random

- Except for the small world property (avg. path length ~ lnN/ln<K>), the properties observed for real-world networks are not matching with that observed for ER-random networks.

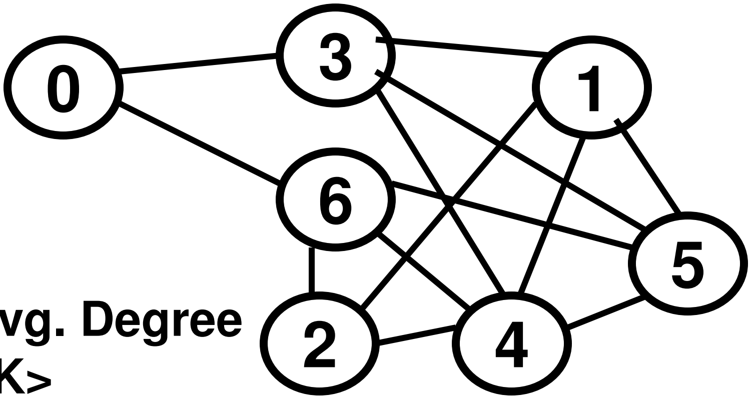
- Then why study random graph theory (ER-model)?
- If a certain property is observed for real-world networks, we can refer to the random graph theory and analyze whether the property is observed by chance (like the small world property).
- If the property observed does not coincide with that of the random networks (like the local clustering coefficient), we need to further analyze the real-world network for the existence of the property because it did not just happen by chance.
- Establish useful benchmarks (e.g., diameter, degree distribution)

Generation of ER-Random Network



Avg. Degree <K>
= (2 + 3 + 3 + 5 + 3 + 4 + 2) / 7
= 3.14

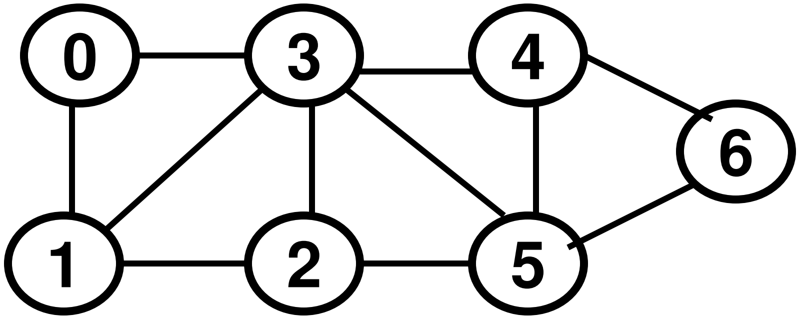
ER Model plink = <K> / (N-1)
plink = 3.14 / 6 = 0.524



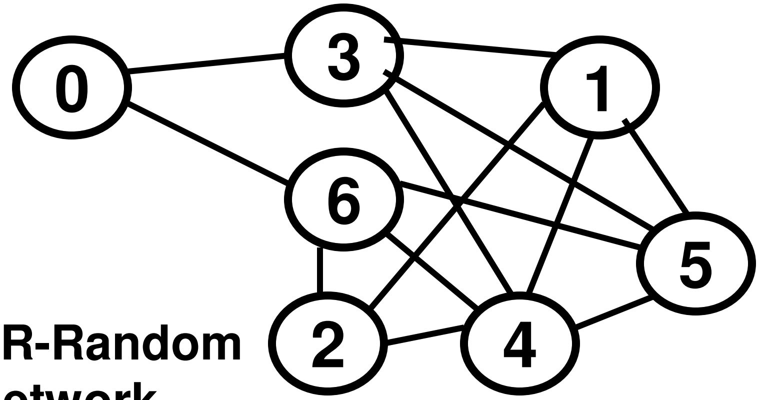
Avg. Degree
<K>
= (2 + 4 + 3 + 4 + 5 + 4 + 4) / 7
= 3.71

Index	Pairs	Random Val	Edge
1	0, 1	0.6335	N
2	0, 2	0.7478	N
3	0, 3	0.1721	Y
4	0, 4	0.9234	N
5	0, 5	0.8563	N
6	0, 6	0.3141	Y
7	1, 2	0.1594	Y
8	1, 3	0.2945	Y
9	1, 4	0.2227	Y
10	1, 5	0.0343	Y
11	1, 6	0.7621	N
12	2, 3	0.8595	N
13	2, 4	0.3091	Y
14	2, 5	0.5312	N
15	2, 6	0.1834	Y
16	3, 4	0.4194	Y
17	3, 5	0.2549	Y
18	3, 6	0.6974	N
19	4, 5	0.0968	Y
20	4, 6	0.4486	Y
21	5, 6	0.2983	Y

Generation of ER-Random Network (contd...)



Given Network



ER-Random Network

Degree	#Nodes	Prob.
2	2	2/7 = 0.286
3	3	3/7 = 0.428
4	1	1/7 = 0.143
5	1	1/7 = 0.143

Degree	#Nodes	Prob.
2	1	1/7 = 0.143
3	1	1/7 = 0.143
4	4	4/7 = 0.571
5	1	1/7 = 0.143

Degree	Nodes	LCC	Avg <LCC>
2	0, 6	1, 1	1.0
3	1, 2, 4	2/3, 2/3, 2/3	0.67
4	5	3/6	0.5
5	3	4/10	0.4

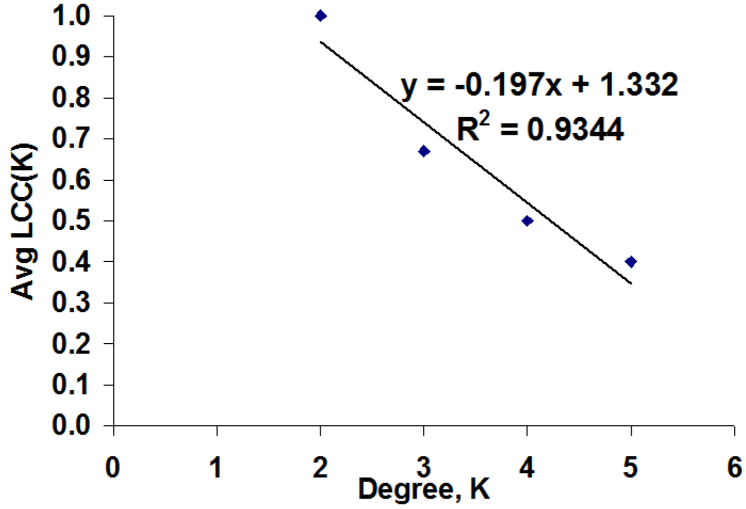
Degree	Nodes	LCC	Avg <LCC>
2	0	0.0	0.0
3	2	2/3	0.67
4	1, 3, 5, 6	4/6, 3/6 4/6, 2/6	0.542
5	4	6/10	0.6

Generation of ER-Random Network

(contd…)

Given Network

Degree	Avg <LCC>
2	1.0
3	0.67
4	0.5
5	0.4

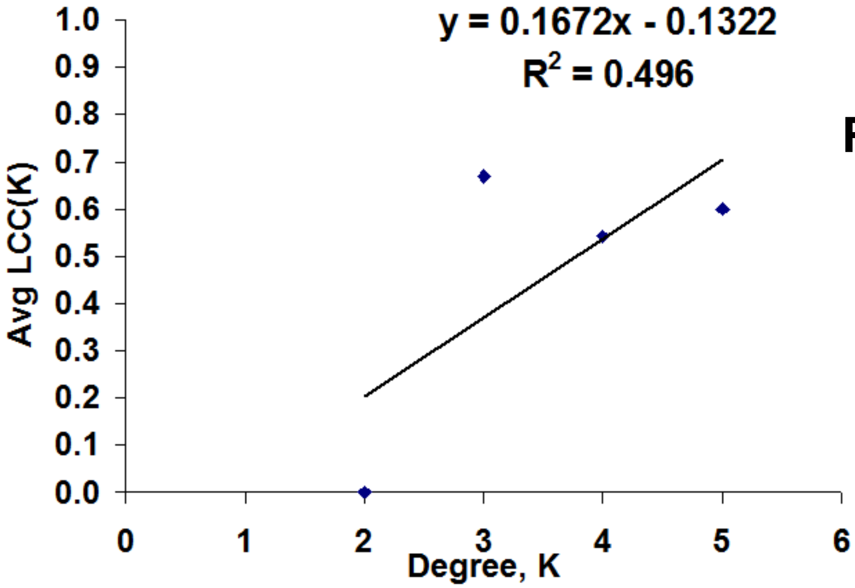


R^2 = 0.9344

K and Avg LCC(K) exhibit a strong correlation

ER-Random Network

Degree	Avg <LCC>
2	0.0
3	0.67
4	0.542
5	0.6



R^2 = 0.4960

K and Avg LCC(K) exhibit a weak correlation

Problem Example 1

- Consider a random network generated according to the G(N, p) model where the total number of nodes is 12 and the probability that there are links between any two nodes is 0.20. Determine the following:
  - The average number of links in the network
  - The average node degree
  - The standard deviation of the node degree
  - The average path length (distance between any two nodes in the network)
  - The average local clustering coefficient for any node in the network.
  - The expected local clustering coefficient for a node that has exactly 5 neighbors.

Problem Example 1: Solution

- There are N = 12 nodes
- Prob[link between any two nodes] = p = 0.2

Max. possible number of links between any two nodes is (N)(N-1)/2 = (12*11/2) = 66

(2) The average number of links in the network = p * N(N-1)/2
= 0.2 * 66 = 13.2

(3) Average node degree = p*(N-1) = 0.2 * 11 = 2.2

(4) Standard deviation of node degree = sqrt(<K>)
= sqrt(2.2) = 1.48

(5) Average path length = ln N / ln <k> = ln(12) / ln(2.2) = 3.15

(6) Avg. Local clustering coefficient for any node in the network = p = 0.2.

(7) The expected local clustering coefficient for a node in a random network is independent of its number of neighbors. Hence, the answer is 0.2

Scale-Free Networks

Dr. Natarajan Meghanathan
Professor
Department of Computer Science
Jackson State University, Jackson, MS
E-mail: natarajan.meghanathan@jsums.edu

Scale-Free Networks

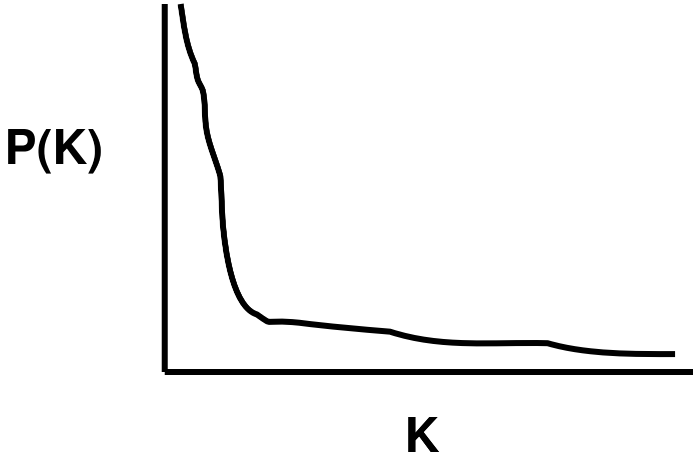
- Scale-free networks follows a Power-law distribution.
- P(k) ~ $k^{-\gamma}$, where γ is the degree exponent (> 1)
- P(k) = $Ck^{-\gamma}$, where C is the proportionality constant

Assuming the degree distribution is discrete

$$C = \frac{1}{\sum_{k=1}^{\infty} k^{-\gamma}} = \frac{1}{\zeta(\gamma)}$$

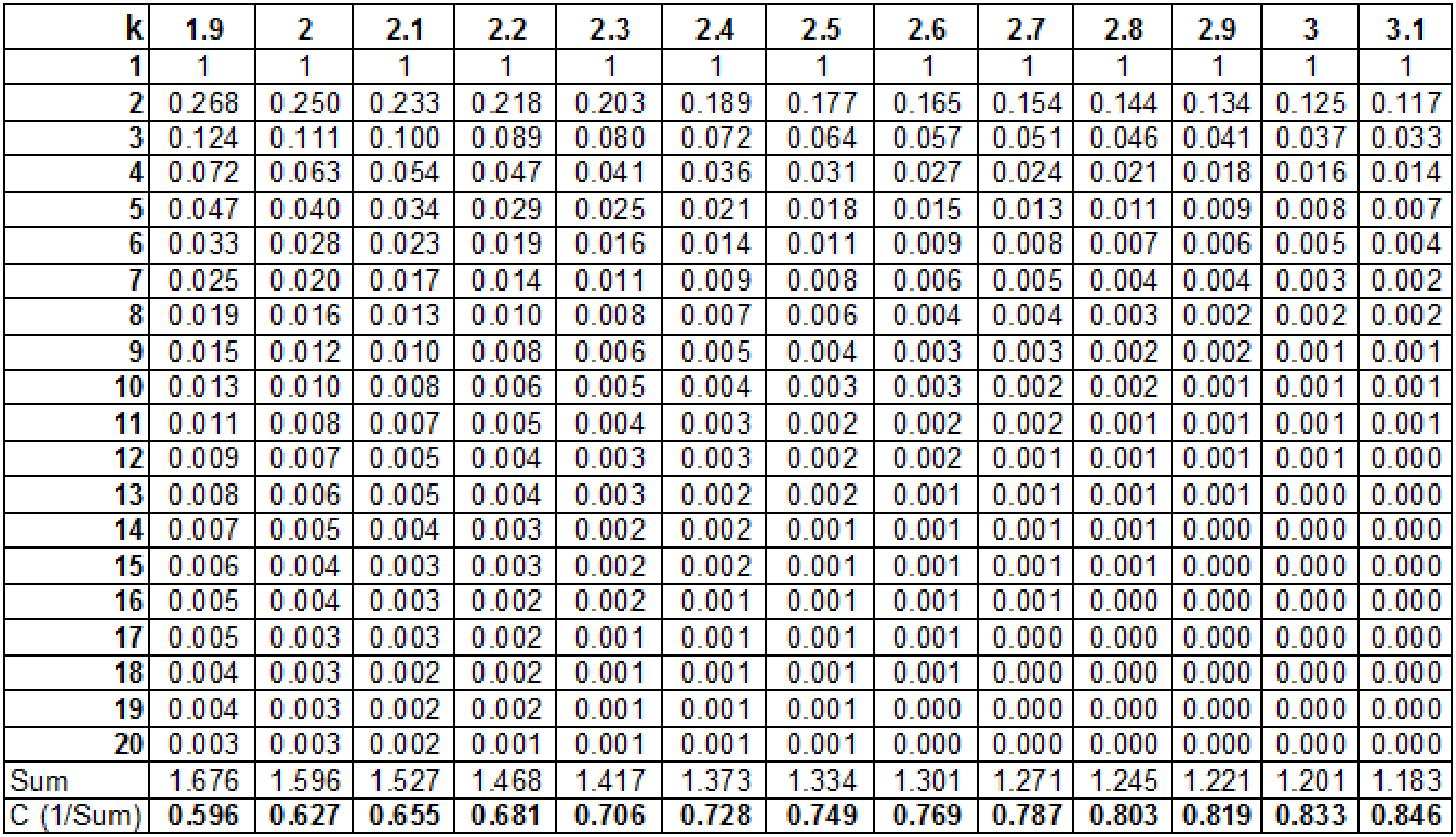
$$\zeta(\gamma) = \sum_{k=1}^{\infty} k^{-\gamma}$$

ζ(γ) is called the Riemann-Zeta Function



Proportionality Constant (Discrete)

γ



k	1.9	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3	3.1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0.268	0.250	0.233	0.218	0.203	0.189	0.177	0.165	0.154	0.144	0.134	0.125	0.117
3	0.124	0.111	0.100	0.089	0.080	0.072	0.064	0.057	0.051	0.046	0.041	0.037	0.033
4	0.072	0.063	0.054	0.047	0.041	0.036	0.031	0.027	0.024	0.021	0.018	0.016	0.014
5	0.047	0.040	0.034	0.029	0.025	0.021	0.018	0.015	0.013	0.011	0.009	0.008	0.007
6	0.033	0.028	0.023	0.019	0.016	0.014	0.011	0.009	0.008	0.007	0.006	0.005	0.004
7	0.025	0.020	0.017	0.014	0.011	0.009	0.008	0.006	0.005	0.004	0.004	0.003	0.002
8	0.019	0.016	0.013	0.010	0.008	0.007	0.006	0.004	0.004	0.003	0.002	0.002	0.002
9	0.015	0.012	0.010	0.008	0.006	0.005	0.004	0.003	0.003	0.002	0.002	0.001	0.001
10	0.013	0.010	0.008	0.006	0.005	0.004	0.003	0.003	0.002	0.002	0.001	0.001	0.001
11	0.011	0.008	0.007	0.005	0.004	0.003	0.002	0.002	0.002	0.001	0.001	0.001	0.001
12	0.009	0.007	0.005	0.004	0.003	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000
13	0.008	0.006	0.005	0.004	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000
14	0.007	0.005	0.004	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000
15	0.006	0.004	0.003	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000
16	0.005	0.004	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000
17	0.005	0.003	0.003	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000
18	0.004	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000
19	0.004	0.003	0.002	0.002	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
20	0.003	0.003	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
Sum	1.676	1.596	1.527	1.468	1.417	1.373	1.334	1.301	1.271	1.245	1.221	1.201	1.183
C (1/Sum)	**0.596**	**0.627**	**0.655**	**0.681**	**0.706**	**0.728**	**0.749**	**0.769**	**0.787**	**0.803**	**0.819**	**0.833**	**0.846**

$\kappa^{-\gamma}$

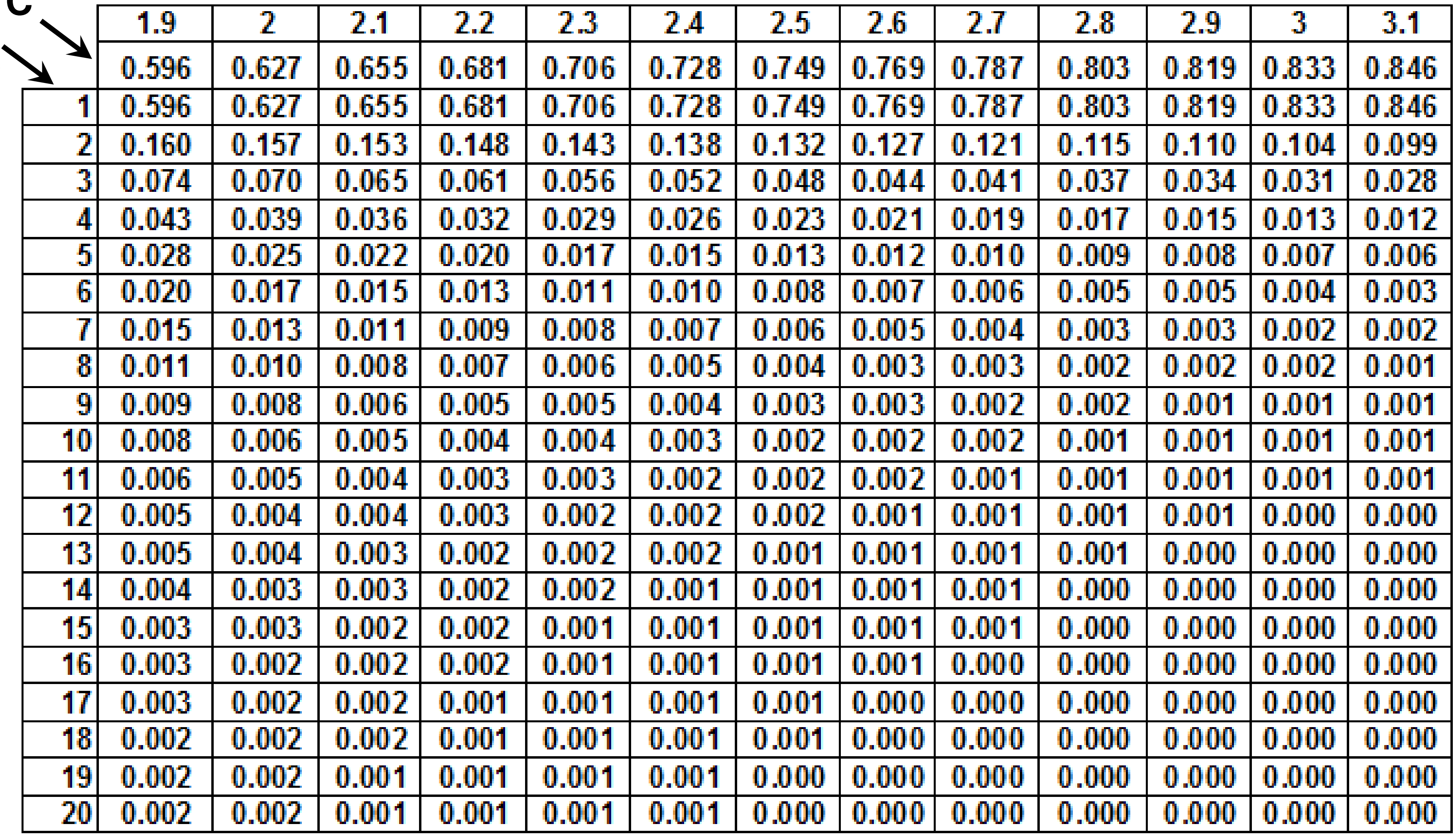
C

Computation of $P(K) = CK^{-\gamma}$ Values

γ

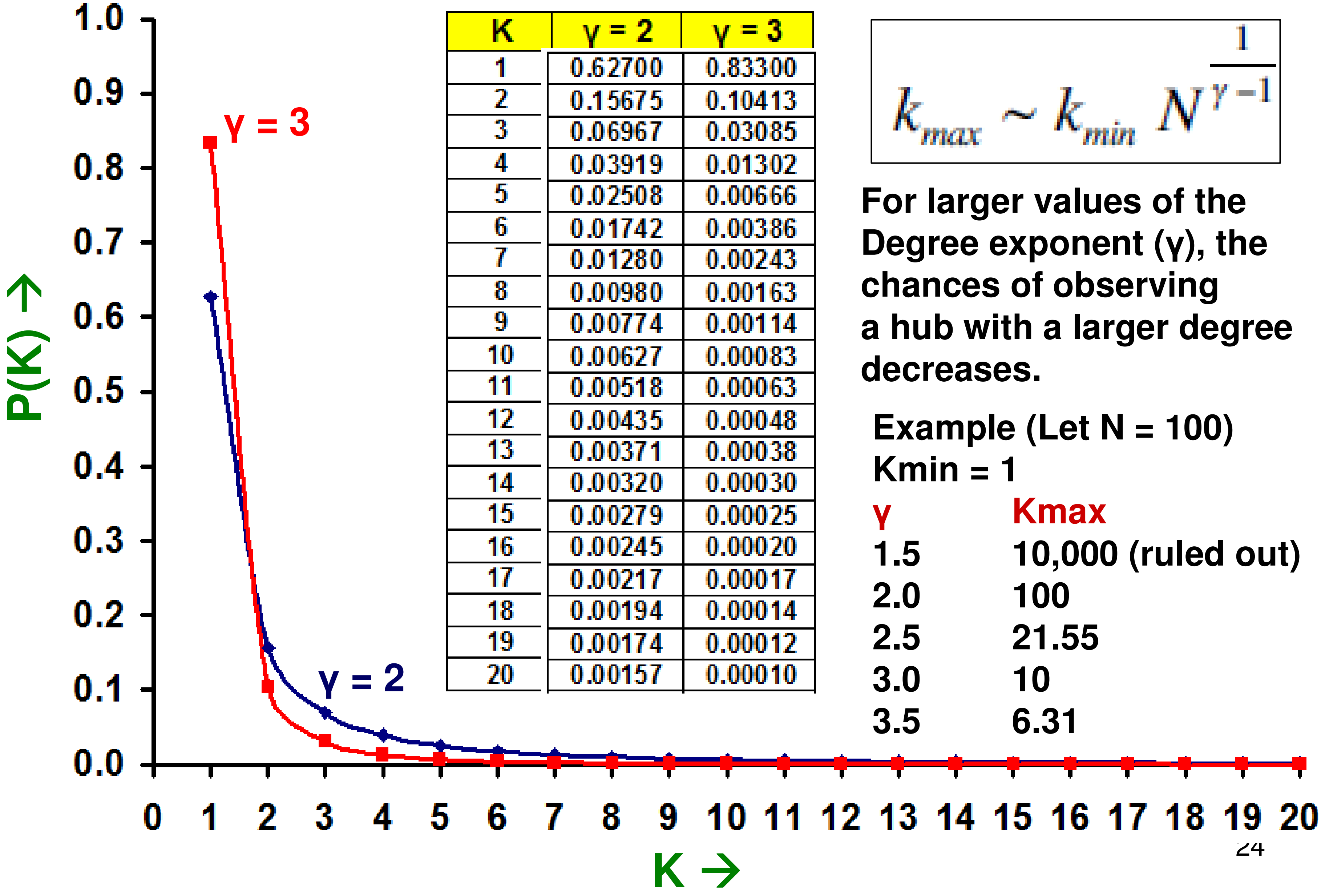
C

K



	1.9	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3	3.1
	0.596	0.627	0.655	0.681	0.706	0.728	0.749	0.769	0.787	0.803	0.819	0.833	0.846
1	0.596	0.627	0.655	0.681	0.706	0.728	0.749	0.769	0.787	0.803	0.819	0.833	0.846
2	0.160	0.157	0.153	0.148	0.143	0.138	0.132	0.127	0.121	0.115	0.110	0.104	0.099
3	0.074	0.070	0.065	0.061	0.056	0.052	0.048	0.044	0.041	0.037	0.034	0.031	0.028
4	0.043	0.039	0.036	0.032	0.029	0.026	0.023	0.021	0.019	0.017	0.015	0.013	0.012
5	0.028	0.025	0.022	0.020	0.017	0.015	0.013	0.012	0.010	0.009	0.008	0.007	0.006
6	0.020	0.017	0.015	0.013	0.011	0.010	0.008	0.007	0.006	0.005	0.005	0.004	0.003
7	0.015	0.013	0.011	0.009	0.008	0.007	0.006	0.005	0.004	0.003	0.003	0.002	0.002
8	0.011	0.010	0.008	0.007	0.006	0.005	0.004	0.003	0.003	0.002	0.002	0.002	0.001
9	0.009	0.008	0.006	0.005	0.005	0.004	0.003	0.003	0.002	0.002	0.001	0.001	0.001
10	0.008	0.006	0.005	0.004	0.004	0.003	0.002	0.002	0.002	0.001	0.001	0.001	0.001
11	0.006	0.005	0.004	0.003	0.003	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.001
12	0.005	0.004	0.004	0.003	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000
13	0.005	0.004	0.003	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000
14	0.004	0.003	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000
15	0.003	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000
16	0.003	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000
17	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
18	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
19	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.002	0.002	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Power-Law Distribution (Discrete)



K	γ = 2	γ = 3
1	0.62700	0.83300
2	0.15675	0.10413
3	0.06967	0.03085
4	0.03919	0.01302
5	0.02508	0.00666
6	0.01742	0.00386
7	0.01280	0.00243
8	0.00980	0.00163
9	0.00774	0.00114
10	0.00627	0.00083
11	0.00518	0.00063
12	0.00435	0.00048
13	0.00371	0.00038
14	0.00320	0.00030
15	0.00279	0.00025
16	0.00245	0.00020
17	0.00217	0.00017
18	0.00194	0.00014
19	0.00174	0.00012
20	0.00157	0.00010

$$k_{max} \sim k_{min} \, N^{\frac{1}{\gamma - 1}}$$

For larger values of the Degree exponent (γ), the chances of observing a hub with a larger degree decreases.

Example (Let N = 100)
Kmin = 1

γ	Kmax
1.5	10,000 (ruled out)
2.0	100
2.5	21.55
3.0	10
3.5	6.31

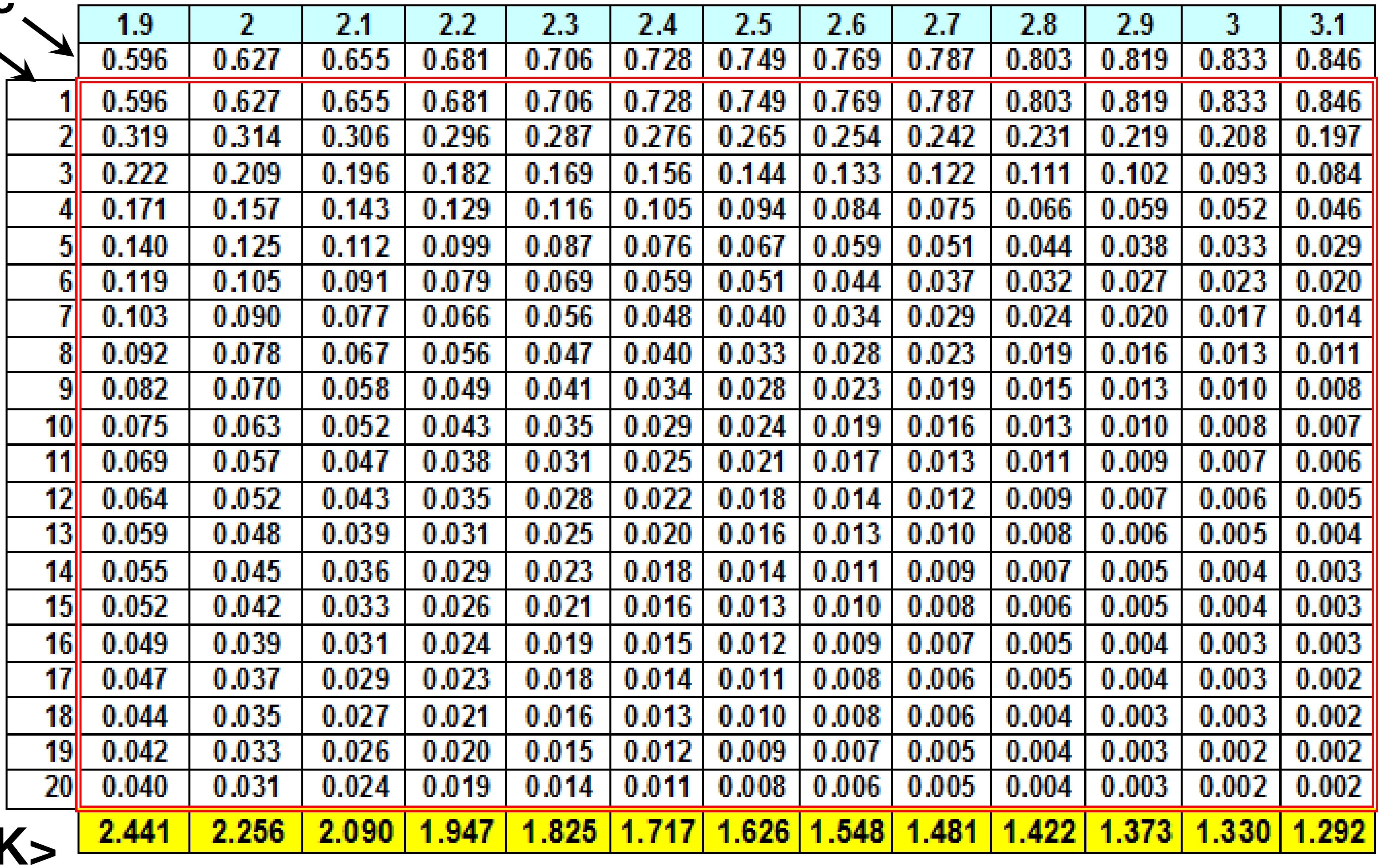
Power Law (Discrete): Avg. Degree

γ

C

K

K*P(K)



	1.9	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3	3.1
	0.596	0.627	0.655	0.681	0.706	0.728	0.749	0.769	0.787	0.803	0.819	0.833	0.846
1	0.596	0.627	0.655	0.681	0.706	0.728	0.749	0.769	0.787	0.803	0.819	0.833	0.846
2	0.319	0.314	0.306	0.296	0.287	0.276	0.265	0.254	0.242	0.231	0.219	0.208	0.197
3	0.222	0.209	0.196	0.182	0.169	0.156	0.144	0.133	0.122	0.111	0.102	0.093	0.084
4	0.171	0.157	0.143	0.129	0.116	0.105	0.094	0.084	0.075	0.066	0.059	0.052	0.046
5	0.140	0.125	0.112	0.099	0.087	0.076	0.067	0.059	0.051	0.044	0.038	0.033	0.029
6	0.119	0.105	0.091	0.079	0.069	0.059	0.051	0.044	0.037	0.032	0.027	0.023	0.020
7	0.103	0.090	0.077	0.066	0.056	0.048	0.040	0.034	0.029	0.024	0.020	0.017	0.014
8	0.092	0.078	0.067	0.056	0.047	0.040	0.033	0.028	0.023	0.019	0.016	0.013	0.011
9	0.082	0.070	0.058	0.049	0.041	0.034	0.028	0.023	0.019	0.015	0.013	0.010	0.008
10	0.075	0.063	0.052	0.043	0.035	0.029	0.024	0.019	0.016	0.013	0.010	0.008	0.007
11	0.069	0.057	0.047	0.038	0.031	0.025	0.021	0.017	0.013	0.011	0.009	0.007	0.006
12	0.064	0.052	0.043	0.035	0.028	0.022	0.018	0.014	0.012	0.009	0.007	0.006	0.005
13	0.059	0.048	0.039	0.031	0.025	0.020	0.016	0.013	0.010	0.008	0.006	0.005	0.004
14	0.055	0.045	0.036	0.029	0.023	0.018	0.014	0.011	0.009	0.007	0.005	0.004	0.003
15	0.052	0.042	0.033	0.026	0.021	0.016	0.013	0.010	0.008	0.006	0.005	0.004	0.003
16	0.049	0.039	0.031	0.024	0.019	0.015	0.012	0.009	0.007	0.005	0.004	0.003	0.003
17	0.047	0.037	0.029	0.023	0.018	0.014	0.011	0.008	0.006	0.005	0.004	0.003	0.002
18	0.044	0.035	0.027	0.021	0.016	0.013	0.010	0.008	0.006	0.004	0.003	0.003	0.002
19	0.042	0.033	0.026	0.020	0.015	0.012	0.009	0.007	0.005	0.004	0.003	0.002	0.002
20	0.040	0.031	0.024	0.019	0.014	0.011	0.008	0.006	0.005	0.004	0.003	0.002	0.002
	2.441	2.256	2.090	1.947	1.825	1.717	1.626	1.548	1.481	1.422	1.373	1.330	1.292

<K>

Why Power-Law is said to be scale-free?

- Kurtosis is a measure of how “heavy-tailed” a distribution is.
- A probability distribution is generally said to be scale-free (i.e., heavy-tailed) if its kurtosis is quite high (typically larger than 3).
- Scale-free distributions also have a standard deviation that is comparable or even larger than the mean.

$$Kurtosis[K] = \frac{E[K - <K>]^4}{\left(E[K - <K>]^2\right)^2}$$

K is the degree;
<K> is the mean degree

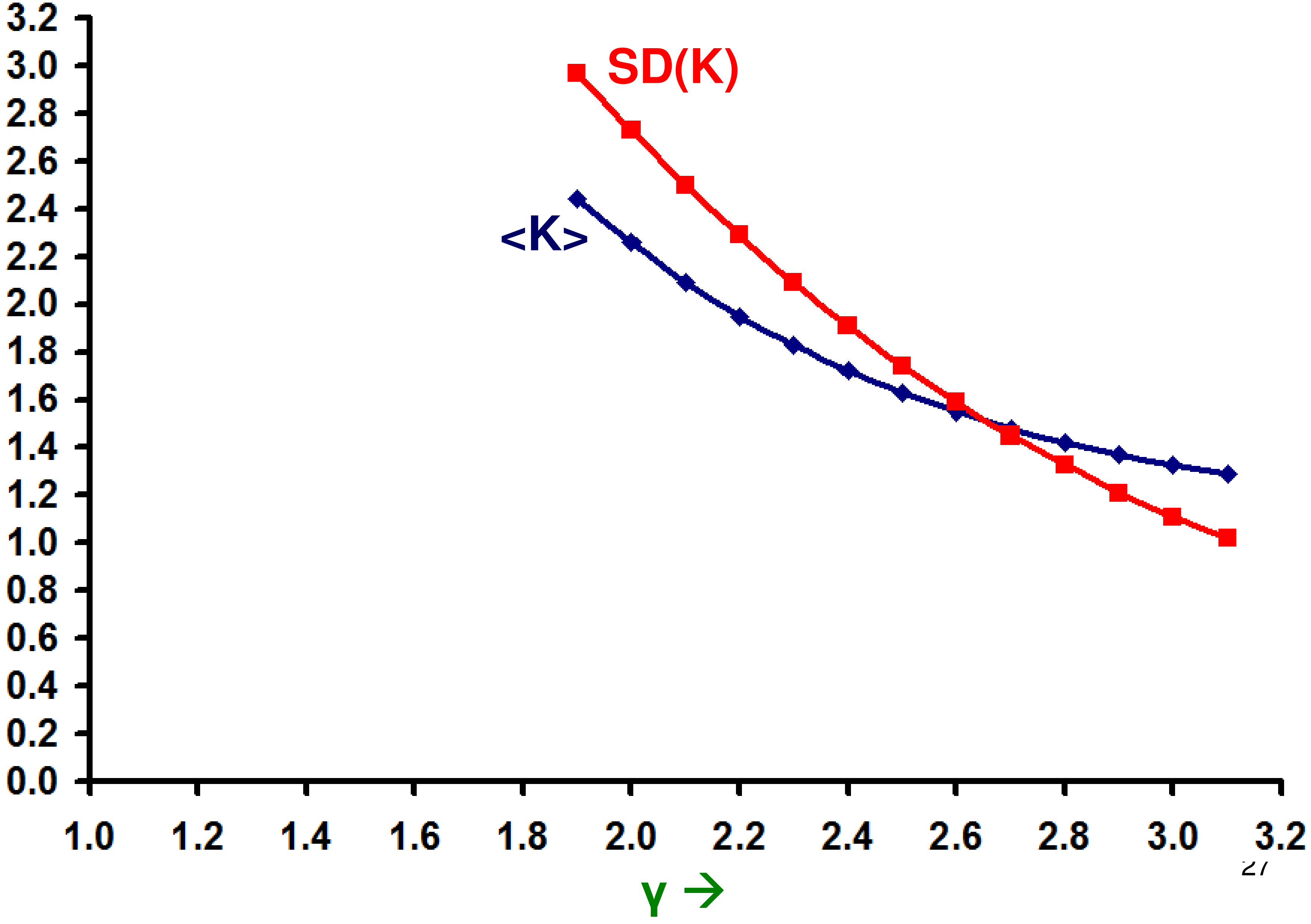
Standard Deviation, SD (K) = $\sqrt{\frac{1}{N}\sum_{K;P(K)>0}^{N}(K - <K>)^2}$

SD(K) = $\sqrt{\sum_{K} P(K) * (K - <K>)^2}$

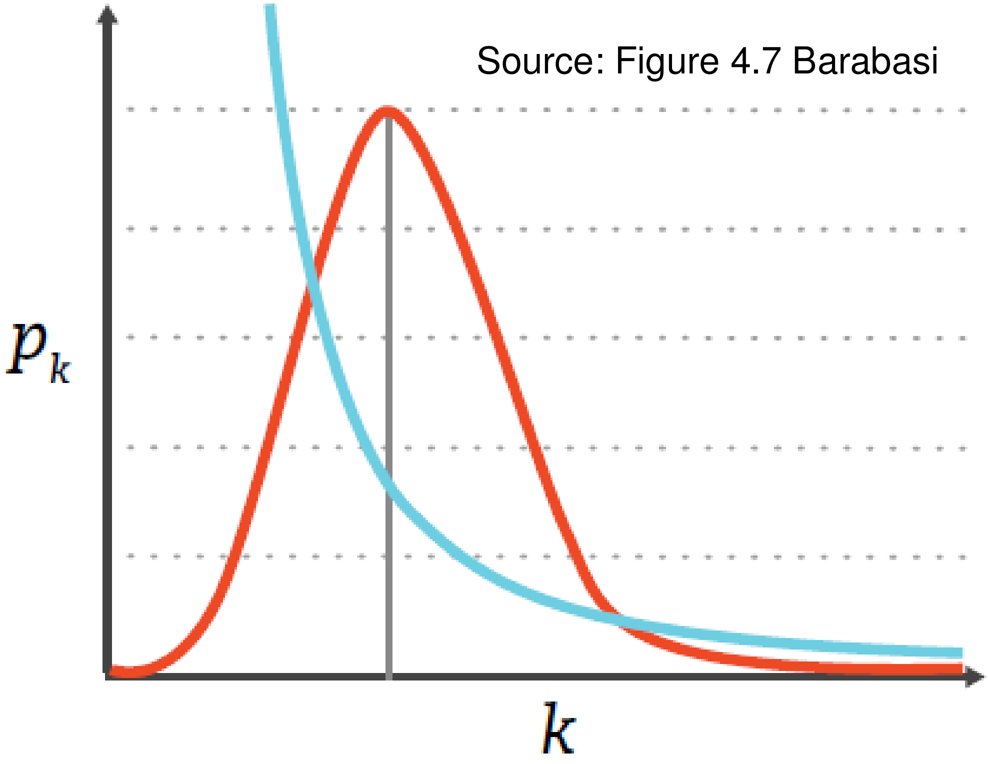
Kurtosis(K) = $\frac{\sum_{K} P(K) * (K - <K>)^4}{SD^4}$

The Kurtosis and SD formula are applied for all values (# samples: N) of K for which there is a non-zero probability of finding a vertex with the particular degree

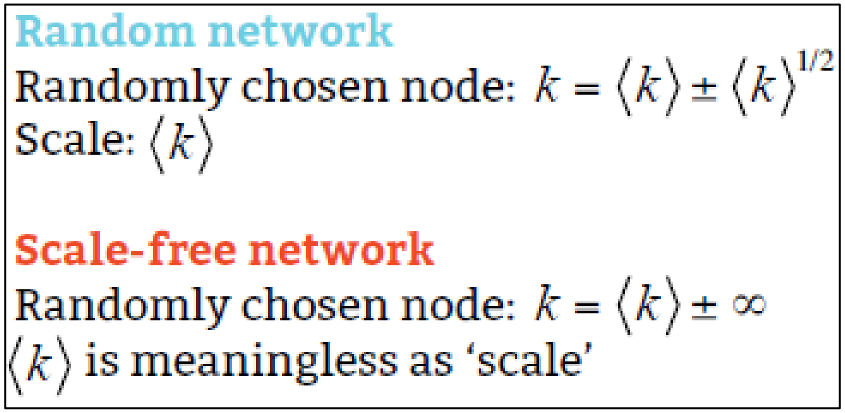
Power-Law (Discrete): Degree (Avg. and SD)



Scale-free networks lack an intrinsic scale



Source: Figure 4.7 Barabasi



Random network

Randomly chosen node: $k = \langle k \rangle \pm \langle k \rangle^{1/2}$

Scale: $\langle k \rangle$

Scale-free network

Randomly chosen node: $k = \langle k \rangle \pm \infty$

$\langle k \rangle$ is meaningless as 'scale'

- For any bounded distribution (e.g. a Poisson or
- a Gaussian distribution) the degree of a randomly chosen node will be in the vicinity of <k>. Hence <*k*> serves as the network's scale.
- In a scale-free network the second moment diverges, hence the degree of a randomly chosen node can be arbitrarily different from <*k*>. Thus, a scale-free network lacks an intrinsic scale (and hence gets its name).

Example 1: Power-law

- Consider a network modeled using the power-law, $P(K) = CK^{-\gamma}$. Determine the power-law exponent γ and the constant C if the network has approximately 4% of nodes with degree 4 and 10% of nodes with degree 3.

- Solution:
- $P(K) = CK^{-\gamma} \rightarrow \ln P(K) = \ln C + (-\gamma)\ln K$
- Given that $P(3) = 0.10$ and $P(4) = 0.04$

$\ln(0.10) = \ln C + (-\gamma)\ln(3) \rightarrow -2.303 = \ln C + (-\gamma)*1.098$

$\ln(0.04) = \ln C + (-\gamma)\ln(4) \rightarrow -3.219 = \ln C + (-\gamma)*1.386$

Solving for γ, we get $\gamma = (3.219 - 2.303)/(1.386 - 1.098) = 3.18$

Substituting for **$\gamma = 3.18$** in the Power-law equation for one of the two degrees, we get $C = P(4) / 4^{-\gamma}. = 0.04 / 4^{-3.18}$

We get **$C = 3.286$**

Example 2: Analyzing a Degree Distribution for Scale-Free Property

Given the following adjacency list for the vertices, determine whether the Degree distribution could be classified to exhibit "scale-free" property.

0	1
0	2
0	3
0	4
0	5
0	7
0	8
1	3
1	5
1	6
1	9
3	4
4	6
4	7
5	9
7	8

0 → 1, 2, 3, 4, 5, 7, 8
1 → 0, 3, 5, 6, 9
2 → 0
3 → 0, 1, 4
4 → 0, 3, 6, 7
5 → 0, 1, 9
6 → 1, 4
7 → 0, 4, 8
8 → 0, 7
9 → 1, 5

ID	Degree
0	7
1	5
2	1
3	3
4	4
5	3
6	2
7	3
8	2
9	2

Degree	#Nodes	P(K)
1	1	1/10 = 0.1
2	3	3/10 = 0.3
3	3	3/10 = 0.3
4	1	1/10 = 0.1
5	1	1/10 = 0.1
7	1	1/10 = 0.1

Avg. Degree = $\sum_{k} K * P(K)$

Avg. Degree, <K>
= (1)(0.1) + (2)(0.3) + (3)(0.3)
+ (4)(0.1) + (5)(0.1) + (7)(0.1)
= 3.2

Example 2(1): Analyzing a Degree Distribution for Scale-Free Property

<K> = 3.2

Degree (K)	P(K)	$(K-<K>)^2$	$(K-<K>)^4$	$P(K)*(K-<K>)^2$	$P(K)*(K-<K>)^4$
1	0.1	4.84	23.43	0.484	2.343
2	0.3	1.44	2.07	0.432	0.621
3	0.3	0.04	0.0016	0.012	0.00048
4	0.1	0.64	0.4096	0.064	0.04096
5	0.1	3.24	10.498	0.324	1.0498
7	0.1	14.44	208.51	1.444	20.851

SD(K) = $$\sqrt{\sum_K P(K)*(K-<K>)^2} = \sqrt{2.76}$$ = **1.661**

$$\sum_K P(K)*(K-<K>)^4$$ = 24.906

Since the Kurtosis (K) = 3.27 is greater than 3, we say the Degree distribution is heavy-tailed.

Kurtosis(K) = $$\frac{\sum_K P(K)*(K-<K>)^4}{SD^4}$$ = $\frac{24.906}{(1.661)^4}$ = **3.27**

Example 3: Analyzing a Degree Distribution for Scale-Free Property

Given the following adjacency list for the vertices, determine whether the Degree distribution could be classified to exhibit "scale-free" property.

0	1
0	2
0	3
0	4
0	5
1	2
1	4
1	6
1	9
2	3
2	8
3	6
3	8
4	5
4	7
4	9
5	7

0 → 1, 2, 3, 4, 5
1 → 0, 2, 4, 6, 9
2 → 0, 1, 3, 8
3 → 0, 2, 6, 8
4 → 0, 1, 5, 7, 9
5 → 0, 4, 7
6 → 1, 3
7 → 4, 5
8 → 2, 3
9 → 1, 4

ID	Degree
0	5
1	5
2	4
3	4
4	5
5	3
6	2
7	2
8	2
9	2

Degree	#Nodes	P(K)
2	4	4/10 = 0.4
3	1	1/10 = 0.1
4	2	2/10 = 0.2
5	3	3/10 = 0.3

Avg. Degree = $$\sum_k K * P(K)$$

Avg. Degree, <K>
= (2)(0.4) + (3)(0.1) + (4)(0.2)
+ (5)(0.3) = 3.4

Example 3(1): Analyzing a Degree Distribution for Scale-Free Property

<K> = 3.4

Degree (K)	P(K)	$(K-\langle K\rangle)^2$	$(K-\langle K\rangle)^4$	$P(K)*(K-\langle K\rangle)^2$	$P(K)*(K-\langle K\rangle)^4$
2	0.4	1.96	3.842	0.784	1.5368
3	0.1	0.16	0.0256	0.016	0.00256
4	0.2	0.36	0.1296	0.072	0.02592
5	0.3	2.56	6.5536	0.768	1.9661

SD(K) = $\sqrt{\sum_K P(K)*(K-<K>)^2} = \sqrt{1.64}$ = **1.281**

$$\sum_K P(K)*(K-<K>)^4$$ = 3.5313

Since the Kurtosis (K) = 1.31 Is lower than 3, we say the Degree distribution is NOT heavy-tailed.

Kurtosis(K) = $\dfrac{\sum_K P(K)*(K-<K>)^4}{SD^4}$ = $\dfrac{3.5313}{(1.281)^4}$ = **1.31**

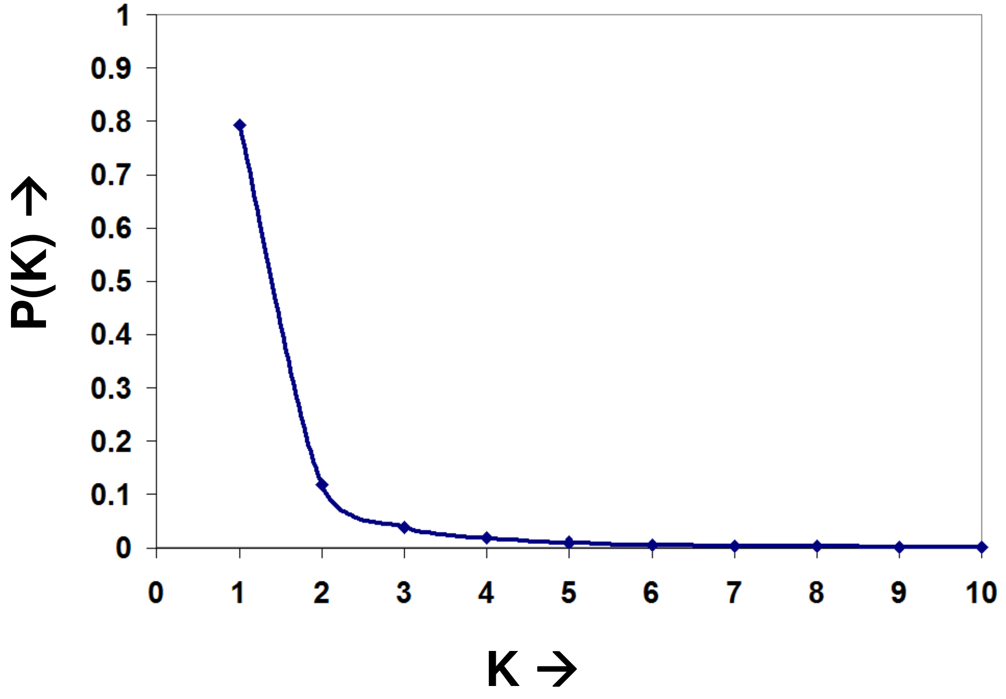
Example 4: Predicting the Nature of Degree Distribution

Given the following probability degree distribution:

1) Draw a plot of the degree distribution and determine if the degree distribution follows a power-law or Poisson?

2) Determine the parameters of the degree distribution you decided.

K	P(K)
1	0.794
2	0.119
3	0.039
4	0.018
5	0.010
6	0.006
7	0.004
8	0.003
9	0.002
10	0.001



It looks clearly like a power-law distribution.

Example 4(1): Predicting the Nature of Degree Distribution

$P(K) = C*K^{-\gamma}$

$\ln P(K) = \ln C + (-\gamma * \ln K)$: Compared to $Y = (\text{slope})*X + \text{constant}$;

slope = - γ ; constant = lnC;

We will use curve (line) fitting in Excel to find the slope and constant

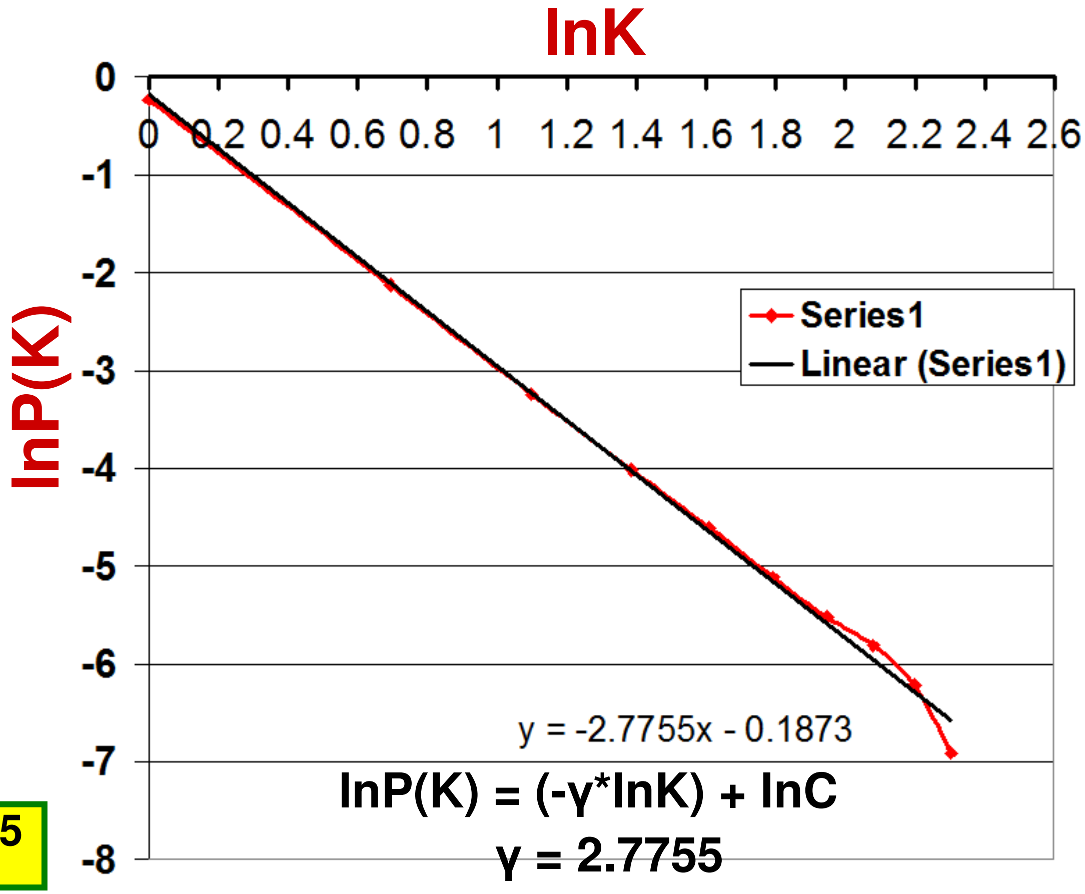
K	P(K)	lnK	lnP(K)
1	0.794	0	-0.23
2	0.119	0.69	-2.13
3	0.039	1.10	-3.24
4	0.018	1.39	-4.02
5	0.010	1.61	-4.61
6	0.006	1.79	-5.12
7	0.004	1.95	-5.52
8	0.003	2.08	-5.81
9	0.002	2.20	-6.21
10	0.001	2.30	-6.91

Constant = lnC = –0.1873

$C = e^{-0.1873} = 2.718^{-0.1873}$

C = 0.829

$\boxed{P(K) = 0.829*K^{-2.7755}}$



$\ln P(K) = (-\gamma * \ln K) + \ln C$

$\gamma = 2.7755$

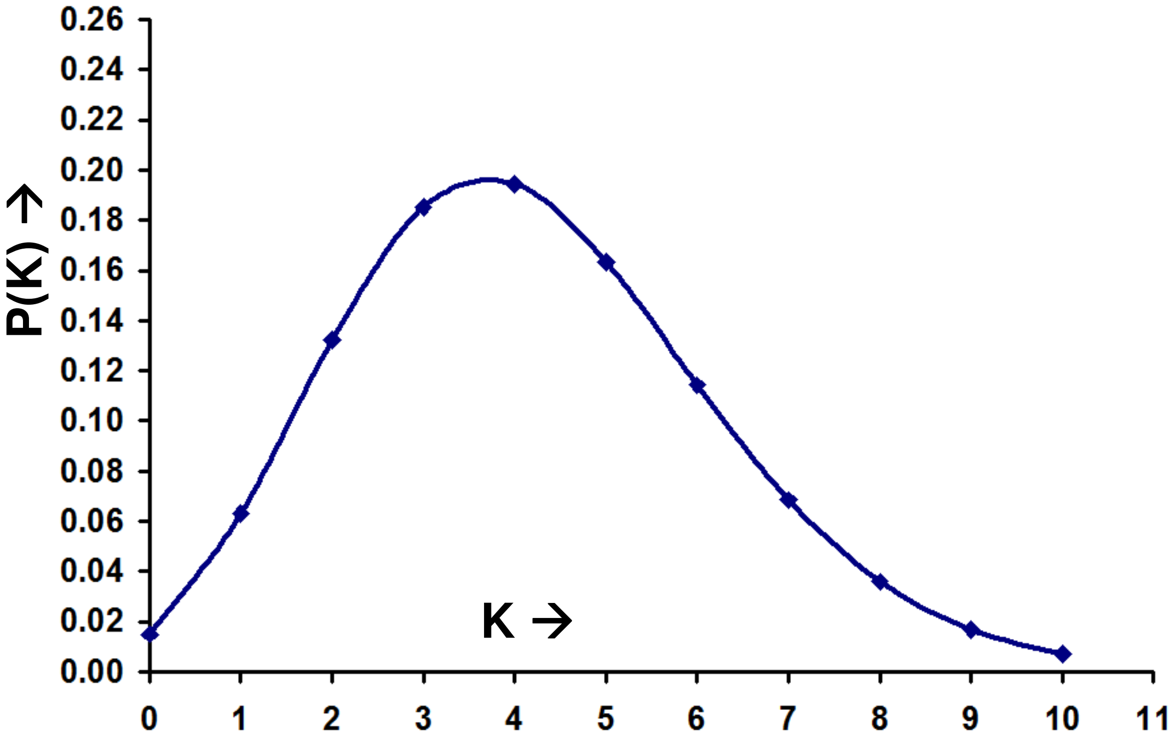
Example 5: Predicting the Nature of Degree Distribution

Given the following probability degree distribution:

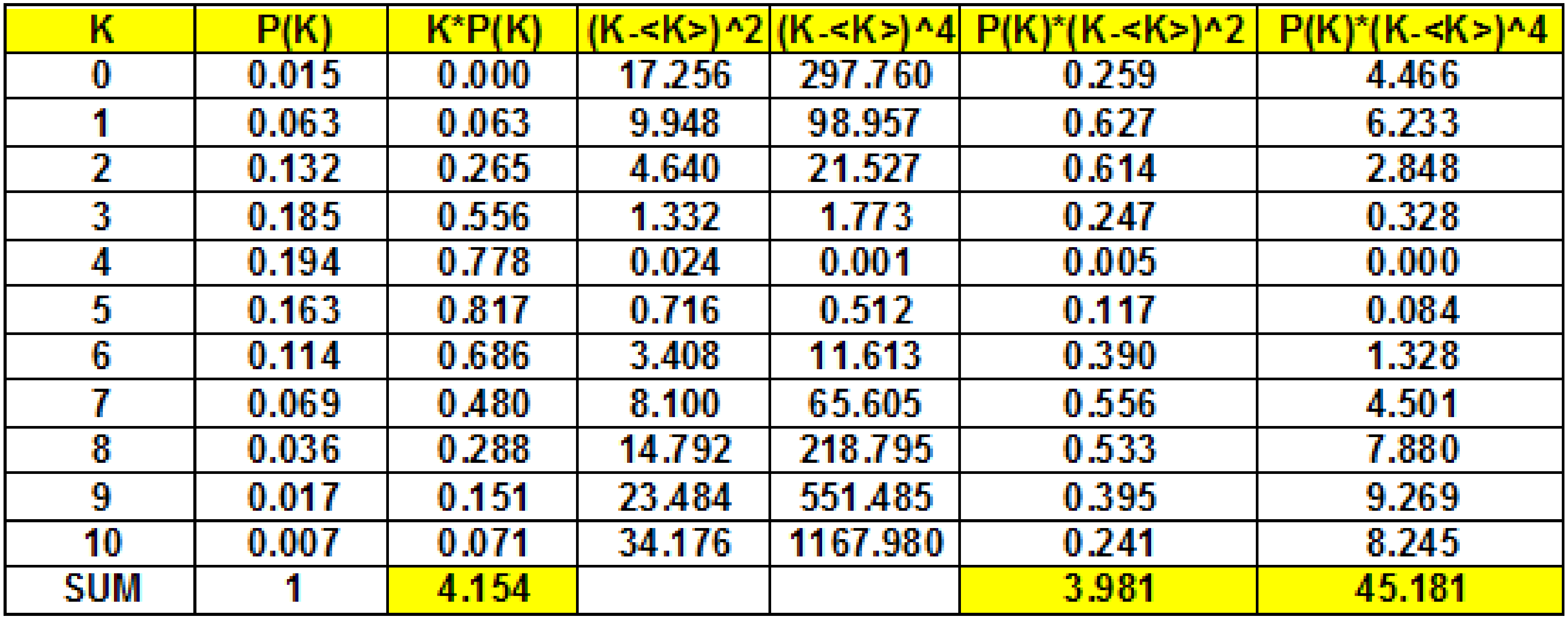
1) Draw a plot of the degree distribution and determine if the degree distribution follows a power-law or Poisson?

2) Determine the parameters of the degree distribution you decided.

K	P(K)
0	0.015
1	0.163
2	0.132
3	0.185
4	0.194
5	0.163
6	0.114
7	0.068
8	0.036
9	0.017
10	0.007



It looks clearly like a Poisson distribution.



K	P(K)	K*P(K)	(K-<K>)^2	(K-<K>)^4	P(K)*(K-<K>)^2	P(K)*(K-<K>)^4
0	0.015	0.000	17.256	297.760	0.259	4.466
1	0.063	0.063	9.948	98.957	0.627	6.233
2	0.132	0.265	4.640	21.527	0.614	2.848
3	0.185	0.556	1.332	1.773	0.247	0.328
4	0.194	0.778	0.024	0.001	0.005	0.000
5	0.163	0.817	0.716	0.512	0.117	0.084
6	0.114	0.686	3.408	11.613	0.390	1.328
7	0.069	0.480	8.100	65.605	0.556	4.501
8	0.036	0.288	14.792	218.795	0.533	7.880
9	0.017	0.151	23.484	551.485	0.395	9.269
10	0.007	0.071	34.176	1167.980	0.241	8.245
SUM	1	4.154			3.981	45.181

Avg. Degree <K> = $\sum_{k} K * P(K)$ **= 4.154**

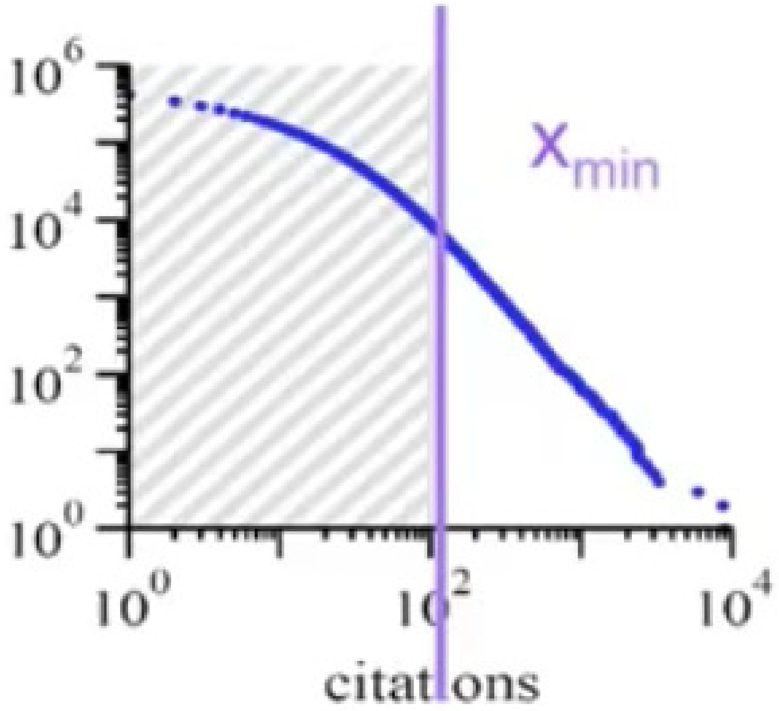
SD(K) = $\sqrt{\sum_{K} P(K) * (K - < K >)^{2}} = \sqrt{3.981}$ **= 1.995**

Kurtosis(K) = $\frac{\sum_{K} P(K) * (K - < K >)^{4}}{SD^{4}}$ $= \frac{45.181}{(1.995)^{4}}$ **= 2.852**

Kurtosis of Poisson distribution is expected to be close to 3
Kurtosis of Heavy-tailed Power-law distribution is expected to be (much) larger than 3.

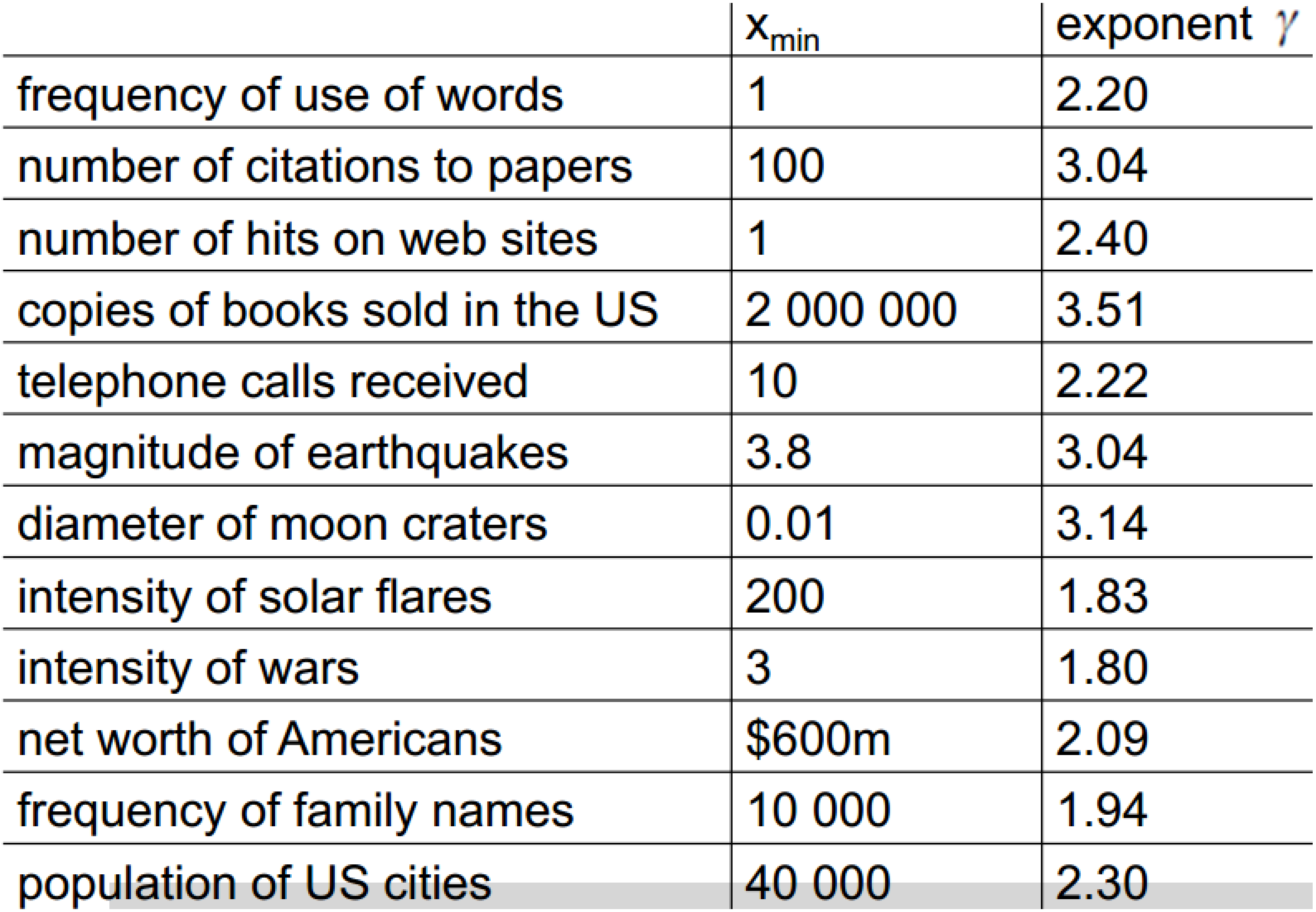
Where does the Power-Law distribution start for real networks?

- If $P(x) = C X^{-\gamma}$, then Xmin needs to be certainly greater than 0, because $X^{-\gamma}$ is infinite at X = 0.
- Some real-world distributions exhibit power-law only from a minimum value (x_{min}).



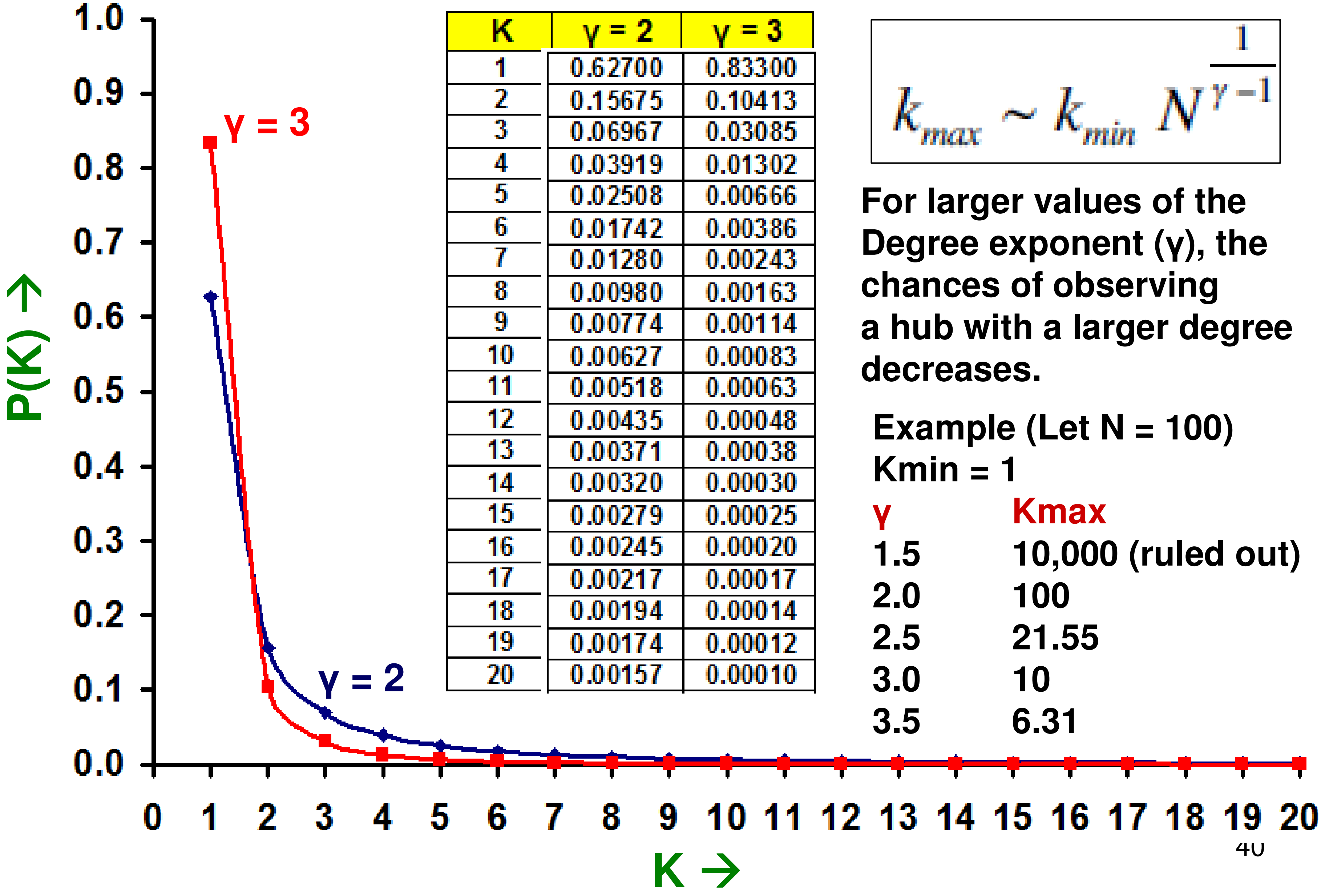
Source:MEJ Newman,
Power laws, Pareto distributions and Zipf’s law, Contemporary Physics 46, 323–351 (2005)

Some Power-Law Exponents of Real-World Data



	x_{min}	exponent γ
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

Power-Law Distribution (Discrete)



K	γ = 2	γ = 3
1	0.62700	0.83300
2	0.15675	0.10413
3	0.06967	0.03085
4	0.03919	0.01302
5	0.02508	0.00666
6	0.01742	0.00386
7	0.01280	0.00243
8	0.00980	0.00163
9	0.00774	0.00114
10	0.00627	0.00083
11	0.00518	0.00063
12	0.00435	0.00048
13	0.00371	0.00038
14	0.00320	0.00030
15	0.00279	0.00025
16	0.00245	0.00020
17	0.00217	0.00017
18	0.00194	0.00014
19	0.00174	0.00012
20	0.00157	0.00010

$$k_{max} \sim k_{min}\, N^{\frac{1}{\gamma-1}}$$

For larger values of the Degree exponent (γ), the chances of observing a hub with a larger degree decreases.

Example (Let N = 100)
Kmin = 1

γ	**Kmax**
1.5	**10,000 (ruled out)**
2.0	**100**
2.5	**21.55**
3.0	**10**
3.5	**6.31**

Average Distance: Power-Law

$$d \sim \begin{cases} \text{const.} & \text{if } \gamma = 2, \\ \dfrac{\ln \ln N}{\ln(\gamma - 1)} & \text{if } 2 < \gamma < 3, \\ \dfrac{\ln N}{\ln \ln N} & \text{if } \gamma = 3, \\ \ln N & \text{if } \gamma > 3. \end{cases}$$

Anomalous regime: Hub and spoke configuration; average distance independent of N.

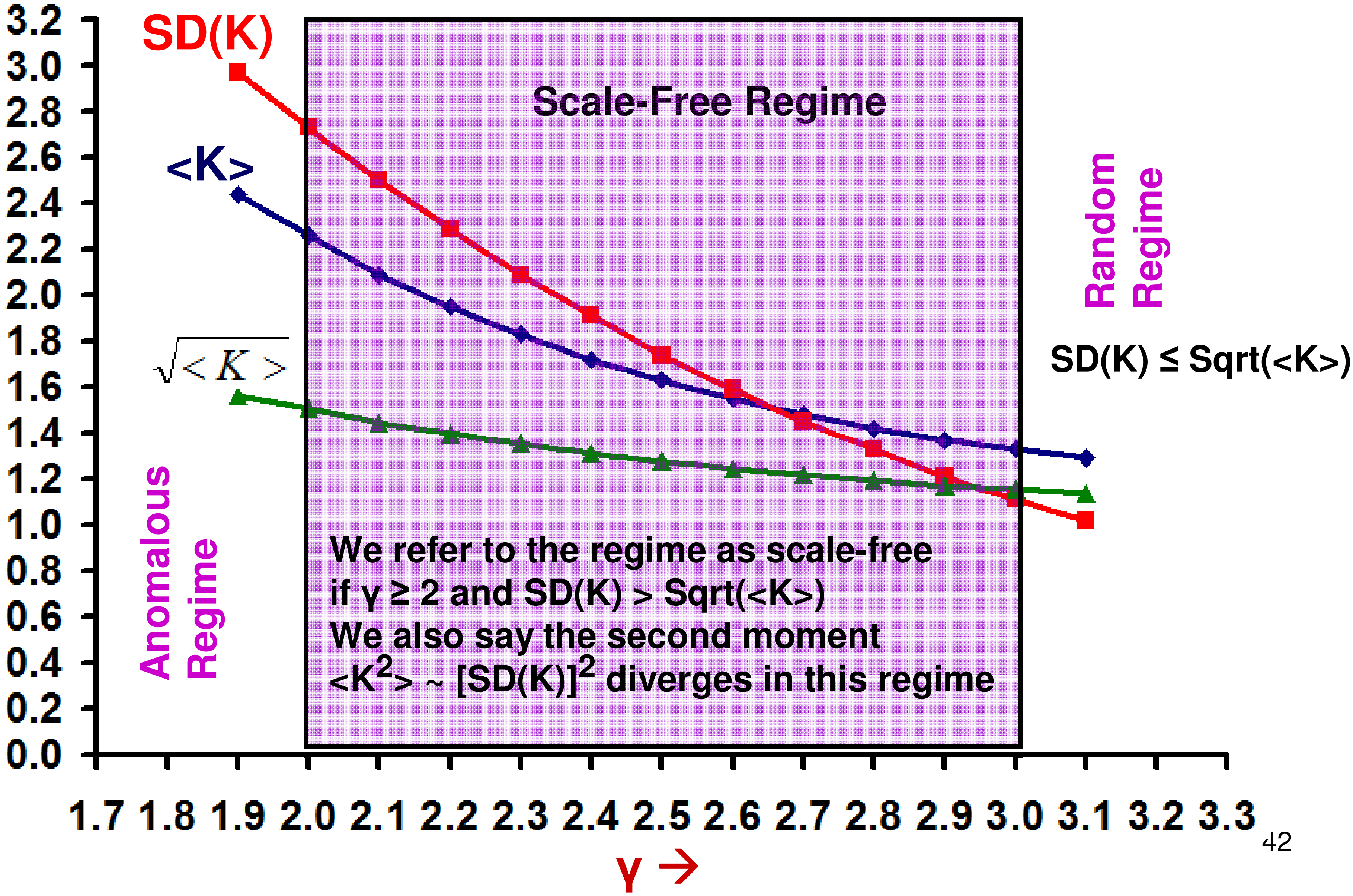
Ultra small world regime
Hubs still reduce the path length

the lnN dependence on N (as in random networks) starts

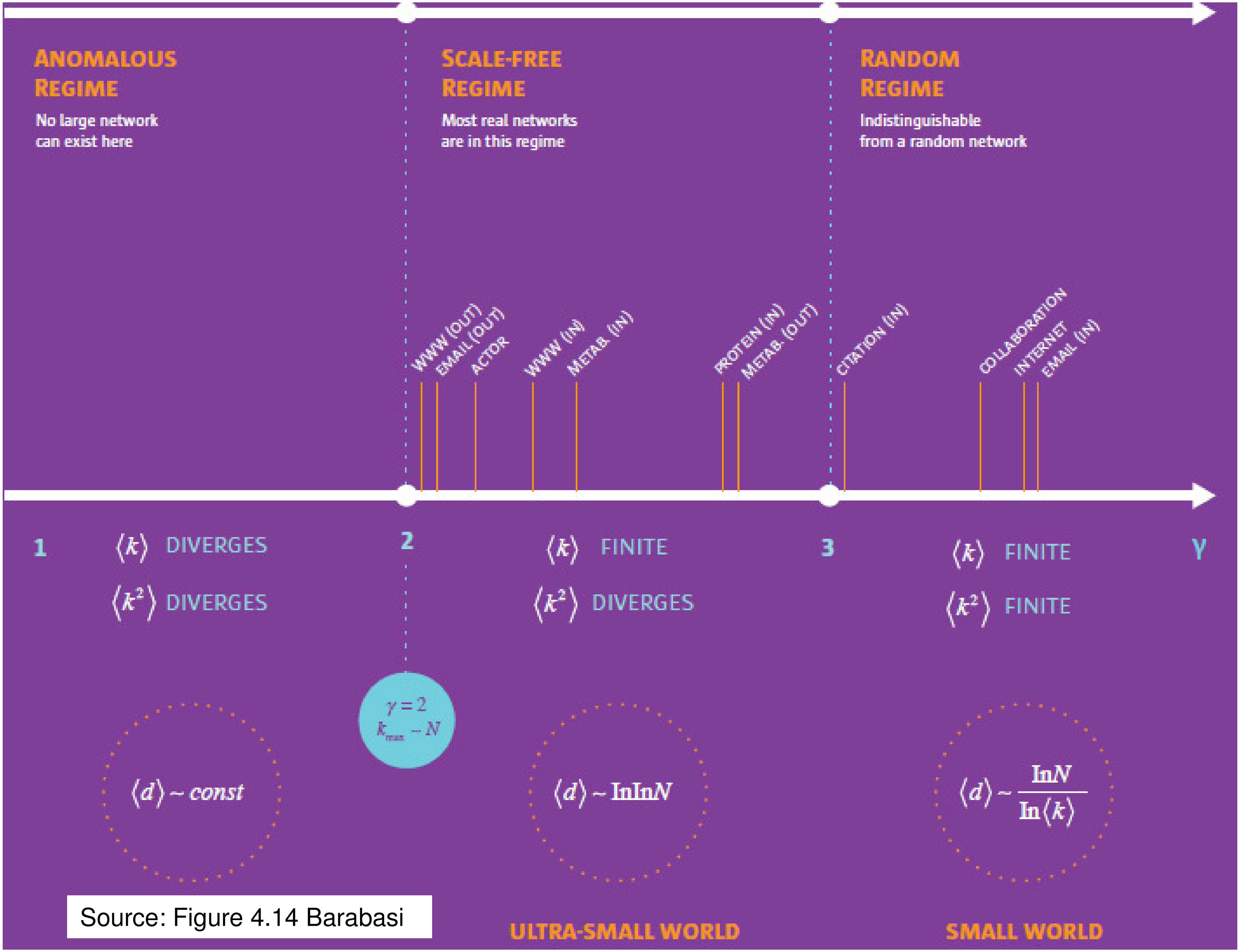
Small world property: Hubs are not sufficiently large and numerous to have impact on path length

The scale-free property shrinks the average path lengths as well as changes the dependence of <d> on the system size. The smaller γ, the shorter are the distances between the nodes.

Network Regimes based on SD(K)

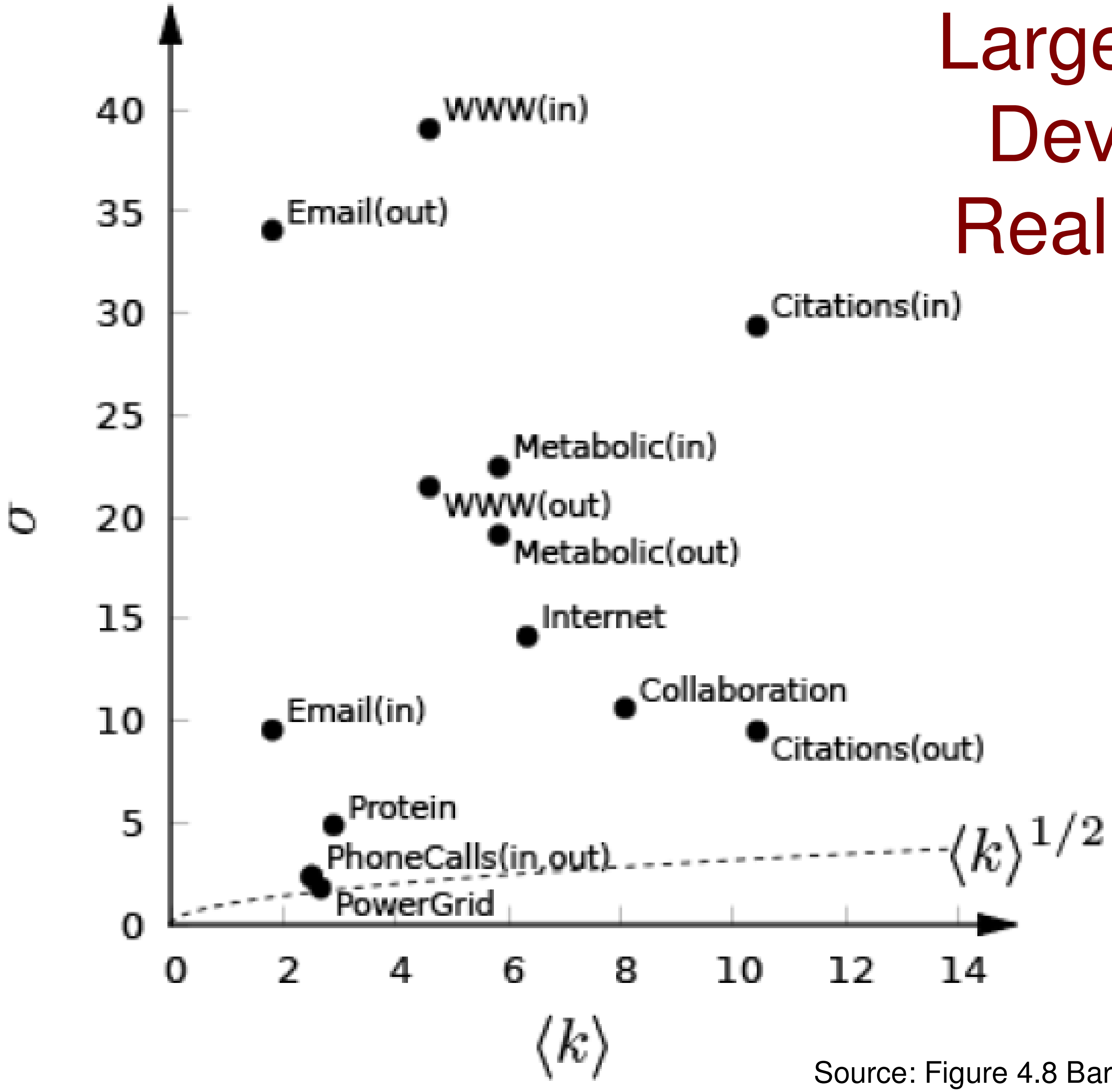


We refer to the regime as scale-free if γ ≥ 2 and SD(K) > Sqrt(<K>)
We also say the second moment $<K^2> \sim [SD(K)]^2$ diverges in this regime


ANOMALOUS REGIME
No large network can exist here
SCALE-FREE REGIME
Most real networks are in this regime
RANDOM REGIME
Indistinguishable from a random network
WWW (OUT)
EMAIL (OUT)
ACTOR
WWW (IN)
METAB. (IN)
PROTEIN (IN)
METAB. (OUT)
CITATION (IN)
COLLABORATION
INTERNET
EMAIL (IN)
1
2
3
γ
$\langle k \rangle$ DIVERGES
$\langle k^2 \rangle$ DIVERGES
$\langle k \rangle$ FINITE
$\langle k^2 \rangle$ DIVERGES
$\langle k \rangle$ FINITE
$\langle k^2 \rangle$ FINITE
$\gamma = 2$
$k_{max} \sim N$
$\langle d \rangle \sim const$
$\langle d \rangle \sim \ln \ln N$
$\langle d \rangle \sim \frac{\ln N}{\ln \langle k \rangle}$
ULTRA-SMALL WORLD
SMALL WORLD

Source: Figure 4.14 Barabasi

Large Standard Deviation for Real Networks



Source: Figure 4.8 Barabasi

Example: Power-Law, Avg. Path Length

- Consider a scale-free network of N = 100 nodes modeled using the power-law, P(K) = CK$^{-\gamma}$. The minimum and maximum degrees of the nodes in the network are kmin = 3 and kmax = 60 respectively. Find the power-law exponent (γ), the power-law constant C and the average path length.

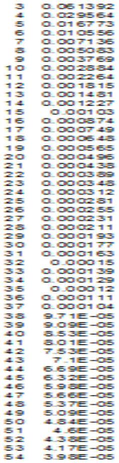
$$\boxed{k_{max} \sim k_{min}\, N^{\frac{1}{\gamma - 1}}}$$

60 = 3 * (100)$^{(1/\gamma-1)}$
1/(γ-1) = ln(60/3) / ln(100)
γ-1 = ln(100) / ln(20) = 1.54
γ = 2.54

$$C = \frac{1}{\sum_{k=1}^{\infty} k^{-\gamma}} = \frac{1}{\zeta(\gamma)}$$

In this case:

$$C = \frac{1}{\sum_{k=3}^{60} k^{-\gamma}}$$



3	0.061392
4	0.029564
5	0.016773
6	0.010556
7	0.007136
8	0.005083
9	0.003769
10	0.002884
11	0.002264
12	0.001815
13	0.001481
14	0.001227
15	0.00103
16	0.000874
17	0.000749
18	0.000648
19	0.000565
20	0.000496
21	0.000438
22	0.000389
23	0.000348
24	0.000312
25	0.000281
26	0.000255
27	0.000231
28	0.000211
29	0.000193
30	0.000177
31	0.000163
32	0.00015
33	0.000139
34	0.000129
35	0.00012
36	0.000111
37	0.000104
38	9.71E-05
39	9.09E-05
40	8.53E-05
41	8.01E-05
42	7.53E-05
43	7.1E-05
44	6.69E-05
45	6.32E-05
46	5.98E-05
47	5.66E-05
48	5.37E-05
49	5.09E-05
50	4.84E-05
51	4.6E-05
52	4.38E-05
53	4.17E-05
54	3.98E-05

Using an Excel Spreadsheet to calculate the value of the summation, we get:

$$\sum_{k=3}^{60} k^{-\gamma} \quad = 0.15333$$

C = 1/0.1533 = 6.52

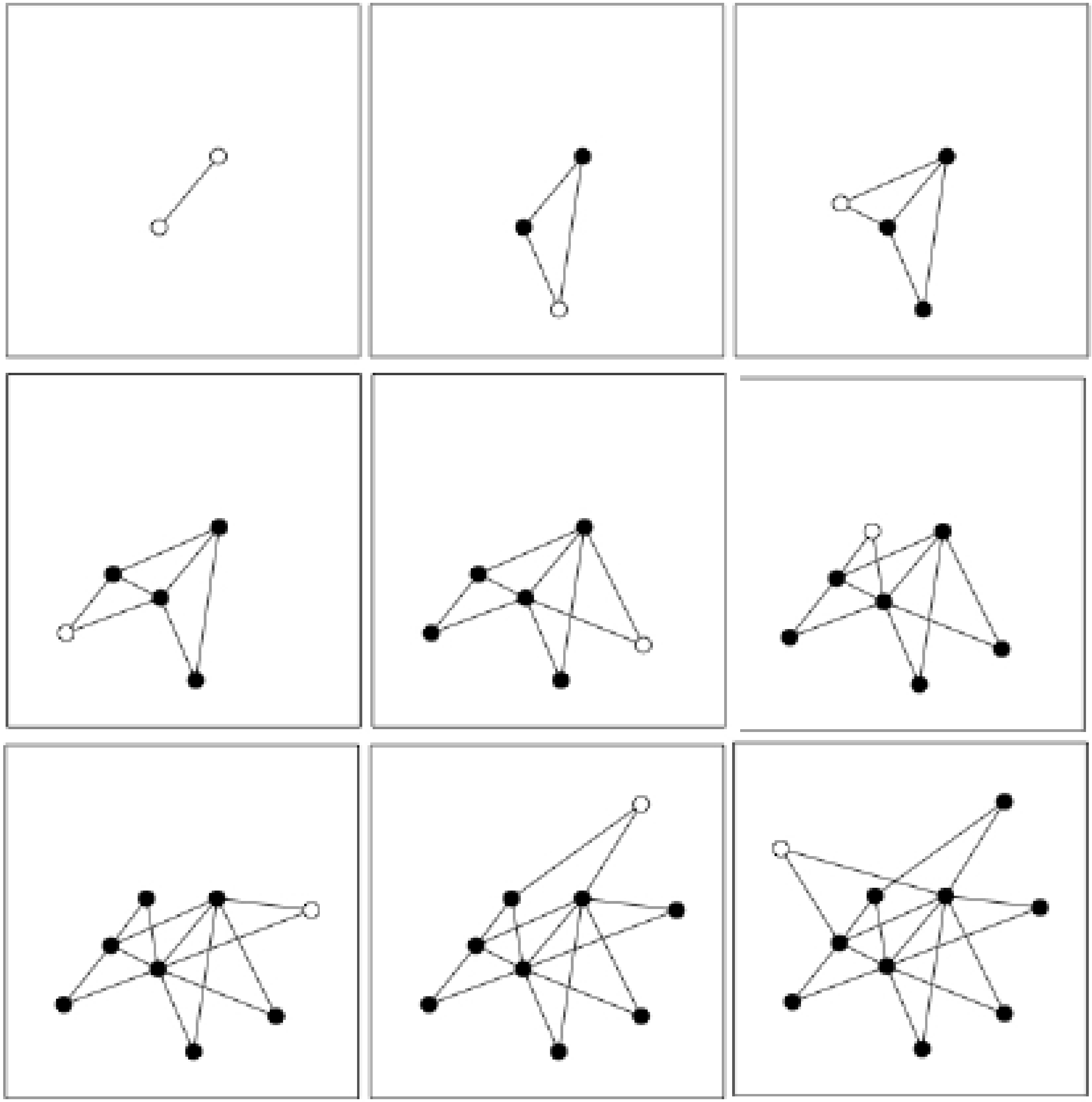
Avg. Path Length
= ln ln(N) / ln (γ-1)
= ln ln(100) / ln(2.54-1)
= 3.54

Barabasi Albert (BA) Model

- BA model is a model for generating networks with power-law degree distribution.
- The model is defined as follows:
  - We start with m_0 nodes, the links between which are chosen arbitrarily, as long as each node has at least one link.
  - The network develops as per the following growth and preferential attachment properties:
    - Growth: At each time step, we add a new node with m ($\leq m_0$) links that connect the new node to m nodes already in the network.
    - Preferential Attachment: The probability π(k) that one of the links of the new node connects to node *i* depends on the degree k_i of node *i* as:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

a node with larger degree has good chances of getting connected to even more nodes.



BA Model Example (m = 2)

Source: Figure 5.4 Barabasi

BA Model: Time-dependent Degree of a Node

- In the BA model, a node has a chance to increase its degree each time a new node enters the network.
- Let k_i be a time-dependent continuous real variable (k_i is the degree of node i that enters the network at time t_i)
- The degree of node i at any time instant $t \geq t_i$ is given by:

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{\beta}$$

where $\beta = ½$ is called the network's dynamical exponent.

Observations:

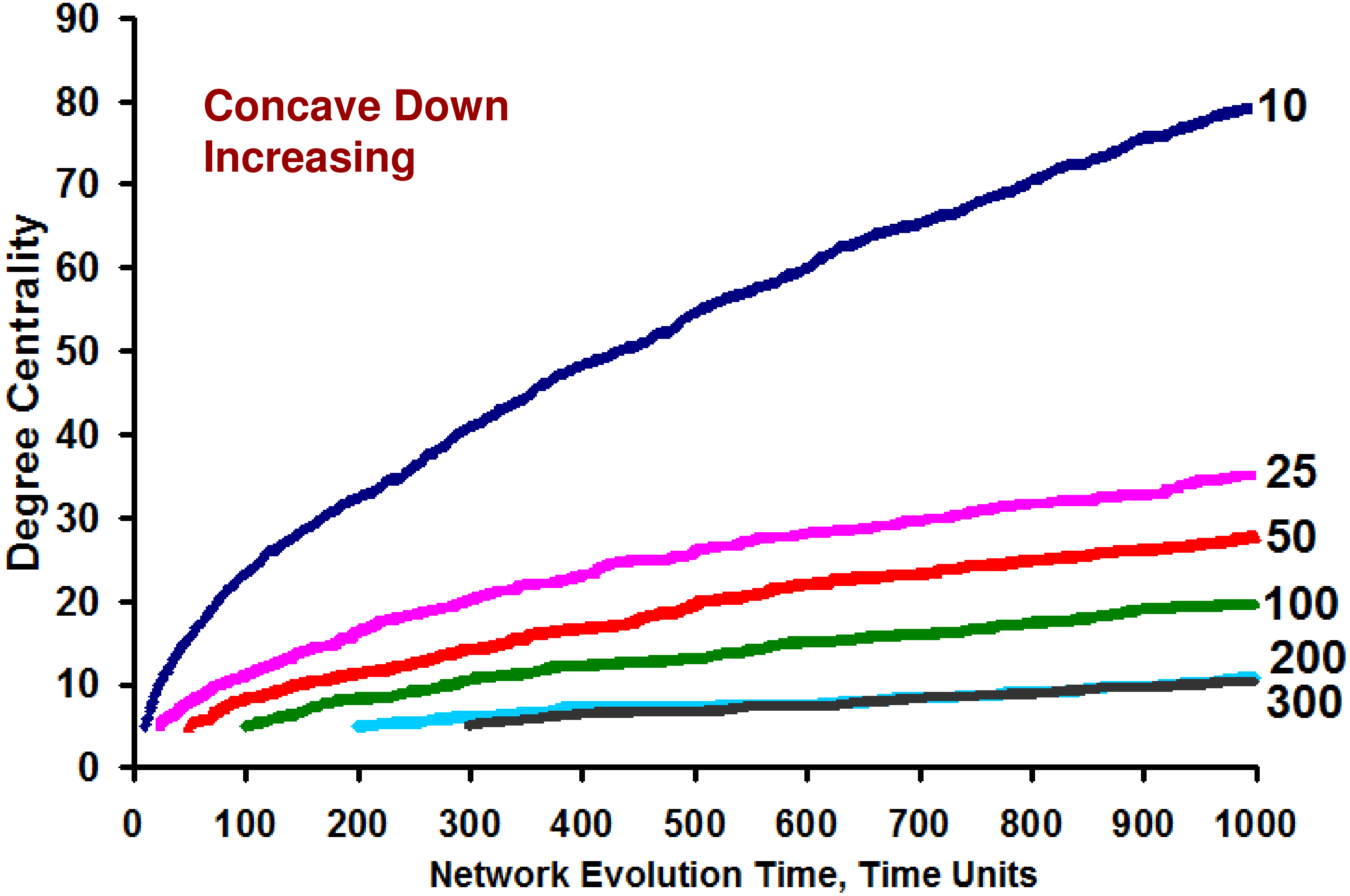
1) The degree of each node increases following the above power law.
2) Each new node has more nodes to link than the previous nodes.

In other words, with time, each node competes for links with an increasing pool of nodes.

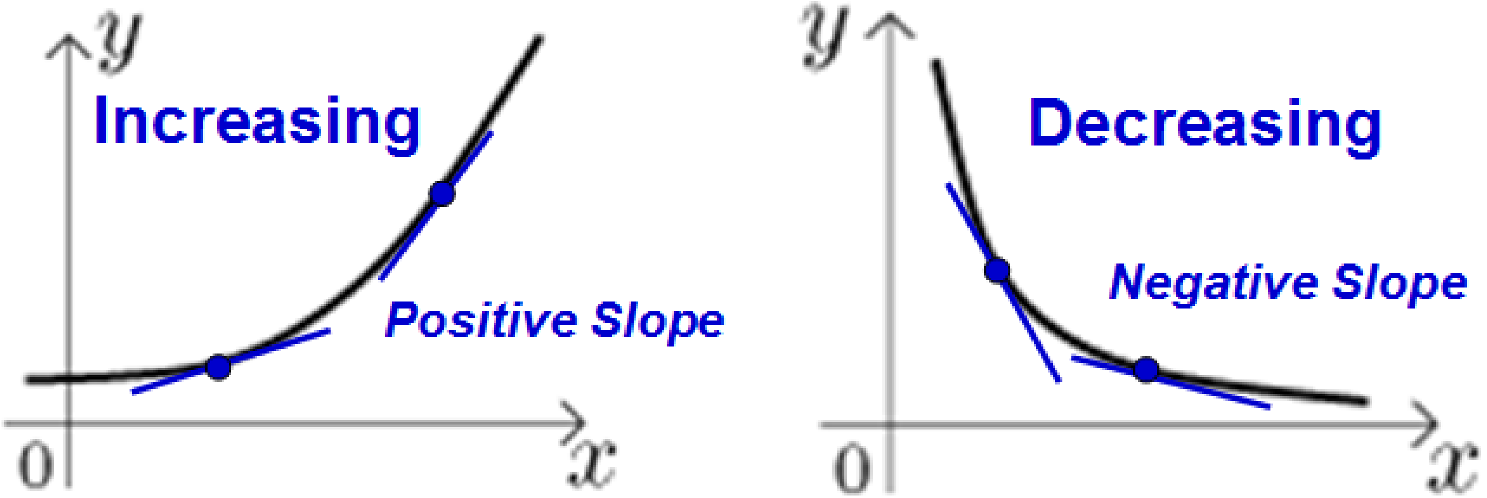
Time Dependent Degree of a Node

- The earlier node *i* was added, the higher is its degree $k_i(t)$.
  - Hence, hubs are large not because they grow faster, but because they arrived earlier.
  - The growth in the degrees is sub linear ($\beta < 1$).
- The rate at which node i acquires new links is given by the derivative: $$\frac{dk_i(t)}{dt} = \frac{m}{2}\frac{1}{\sqrt{t_i t}}$$
- Indicating that older nodes acquire more links in a unit time (as they have smaller t_i), as well as that the rate at which a node acquires links decreases with time as $t^{-1/2}$. Hence, less and less links go to a node with time.
- Thus, the BA model offers a dynamical description of a network's evolution: in real networks, nodes arrive one after the other, connecting to the earlier nodes.
  - This sets up a competition for links during which the older nodes have an advantage over the younger nodes, eventually turning into hubs.

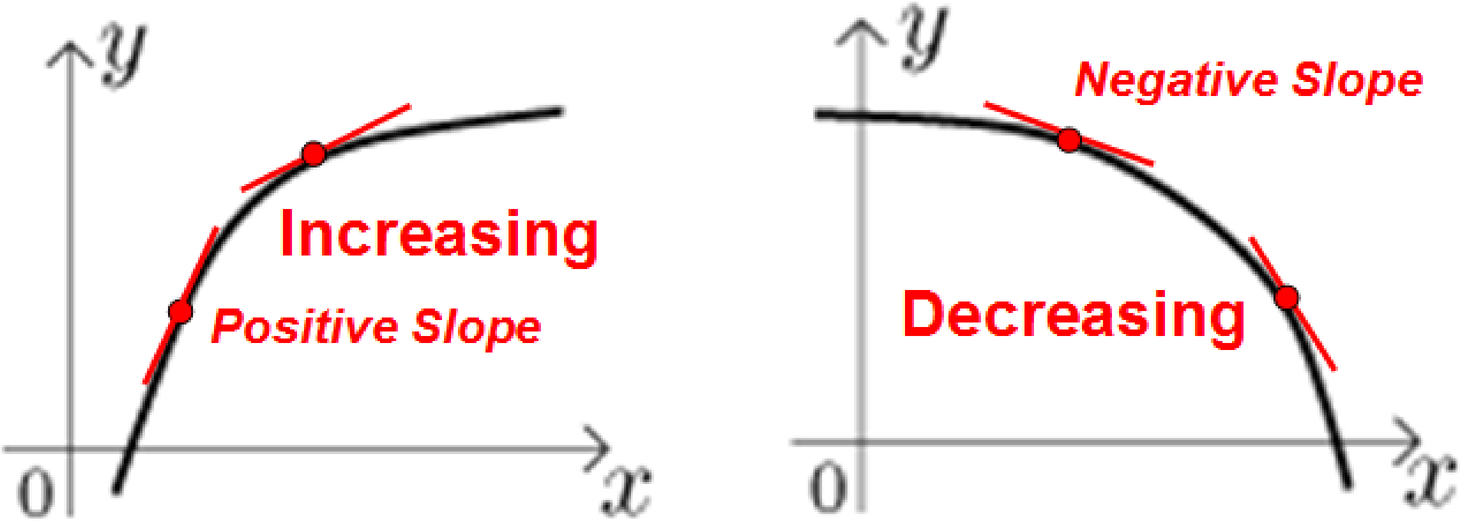
Time-Dependent Variation of Degree Centrality



Nature of Functions

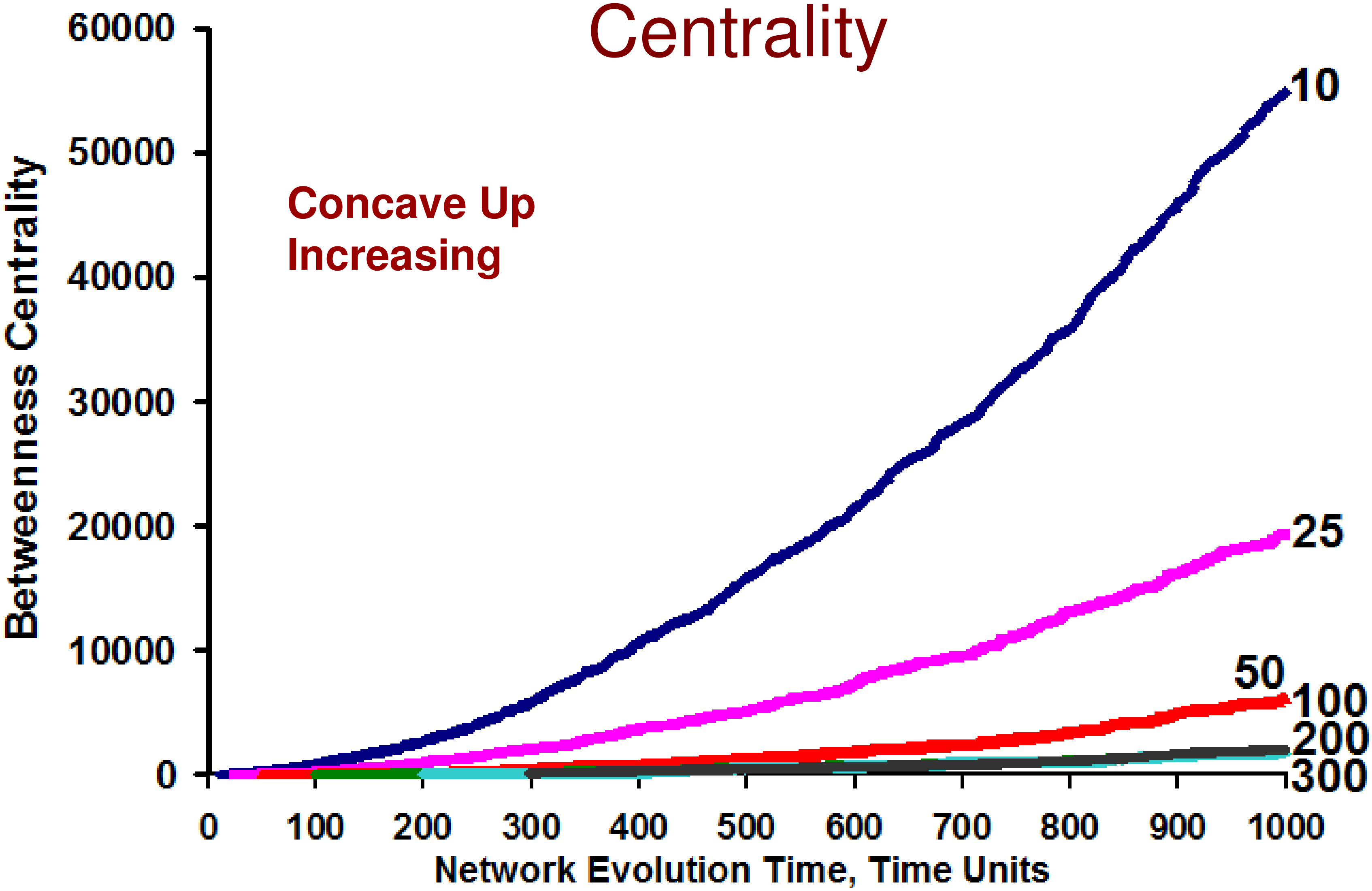


Concave Up (the rate of increase or decrease increases with time)



Concave Down (the rate of increase or decrease decreases with time)

Time-Dependent Variation of Betweenness Centrality



Concave Up Increasing

N. Meghanathan, "Time-Dependent Variation of the Centrality Measures of the Nodes during the Evolution of a Scale-Free Network," *Journal of Networks*, vol. 10, no. 7, pp. 431-442, July 2015.
http://ojs.academypublisher.com/index.php/jnw/article/view/jnw1007431442/10670

Bianconi-Barabasi (BB) Model Motivation

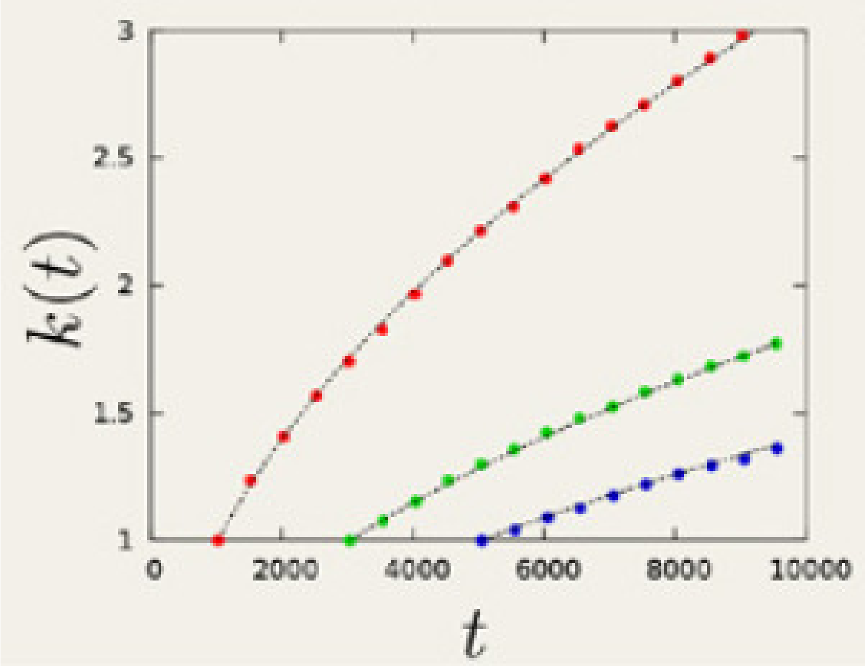
- The Barabasi-Albert model leads to a scenario where the late nodes can never turn into the largest hubs.
- In reality, a node's growth does not depend on the node's age only.
  - Instead web pages, companies or actors have intrinsic qualities that influence the rate at which they acquire links.
    - Some show up late and nevertheless grab most links within a short timeframe.
    - Example: Though, Facebook came later than Google, Facebook is the most linked node in the Internet.
- The goal of this model is to understand how the differences in the node's ability to acquire links, and other processes not captured by the Barabasi-Albert model, like node and link deletion or aging, affect the network topology.

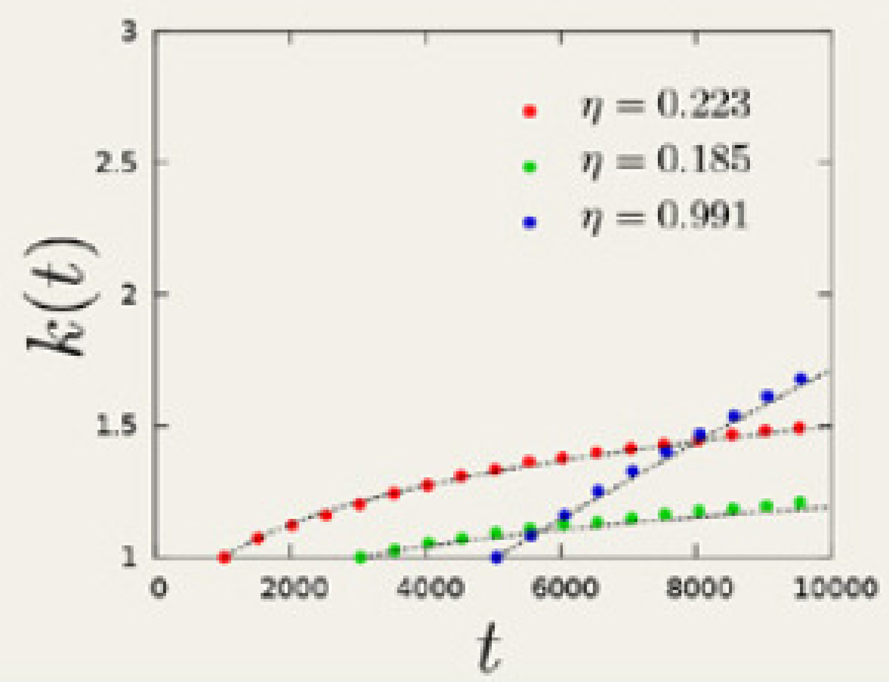
Bianconi-Barabasi (BB) Model

- Fitness – the intrinsic property of a node that propels more nodes towards it.
- The Barabasi-Albert model assumed that a node's growth rate is determined solely by its degree.
- The BB model incorporates the role of fitness and assumes that preferential attachment is driven by the product of a node's fitness, *η*, and its degree *k*.
- Growth: In each timestep, a new node j with m links and fitness η_j is added to the system, where η_j is a random number chosen from a distribution ρ(η) [for example: uniform distribution].
  - Once assigned, a node's fitness does not change.
- Preferential Attachment: The probability that a link of a new node connects to a pre-existing node *i* is proportional to the product of node i's degree k_i and its fitness η_i.

$$\Pi_i = \frac{\eta_i k_i}{\sum_j \eta_j k_j}$$

BA Model vs. BB Model





<u>BB Model</u>

$$\frac{\partial k_i}{\partial t} = m \frac{\eta_i k_i}{\sum_k \eta_j k_j}$$

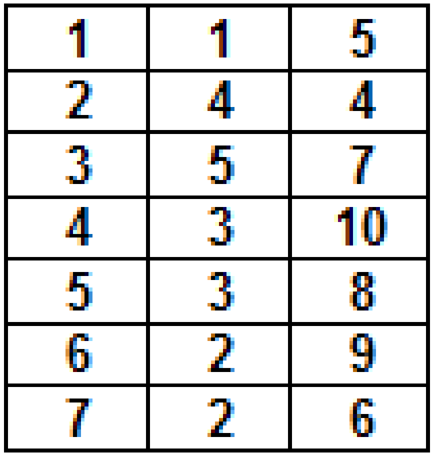
$$k_{\eta_i}(t, t_i) = m \left(\frac{t}{t_i} \right)^{\beta(\eta_i)}$$

where $\beta(\eta_i)$ is a fitness-dependent dynamic exponent of node i.

A node with a higher fitness will increase its degree faster.

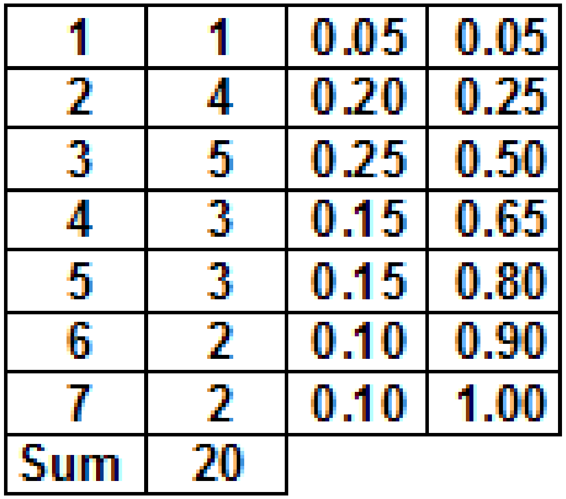
Example-1: BA & BB Model

- Consider the following degree distribution of the nodes and their fitness.
- Determine the probability with which each node is likely to get the first link with a newly joining node under the BA and BB models.
- Let a new node join the network with 2 links under the BA and BB models. Determine which nodes are likely to get connected to the new node.



ID	Degree	Fitness
1	1	5
2	4	4
3	5	7
4	3	10
5	3	8
6	2	9
7	2	6

BA Model (First Link)



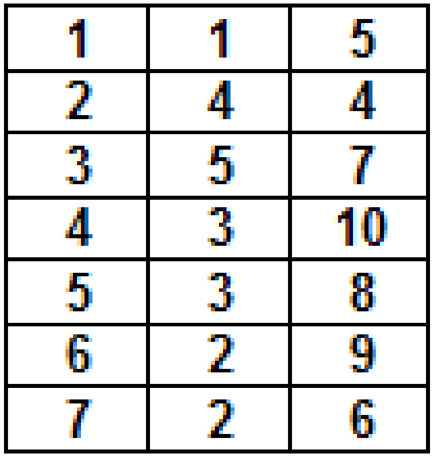
ID	Degree	Probability of a node getting the first link	Cumulative Probability
1	1	0.05	0.05
2	4	0.20	0.25
3	5	0.25	0.50
4	3	0.15	0.65
5	3	0.15	0.80
6	2	0.10	0.90
7	2	0.10	1.00
Sum	20		

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}.$$

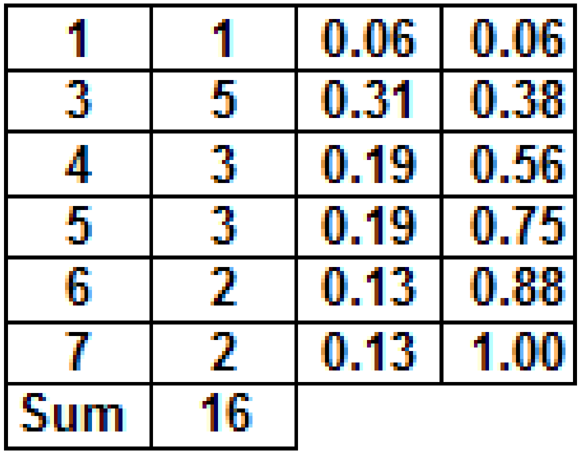
Generate a random number I got 0.2429 using the Random number generator Program I gave you.

Node 2 gets selected for the first link

Example-1: BA & BB Model



ID	Degree	Fitness
1	1	5
2	4	4
3	5	7
4	3	10
5	3	8
6	2	9
7	2	6



ID	Degree	Probability of a node getting the second link	Cumulative Probability
1	1	0.06	0.06
3	5	0.31	0.38
4	3	0.19	0.56
5	3	0.19	0.75
6	2	0.13	0.88
7	2	0.13	1.00
Sum	16		

BA Model (Second Link)

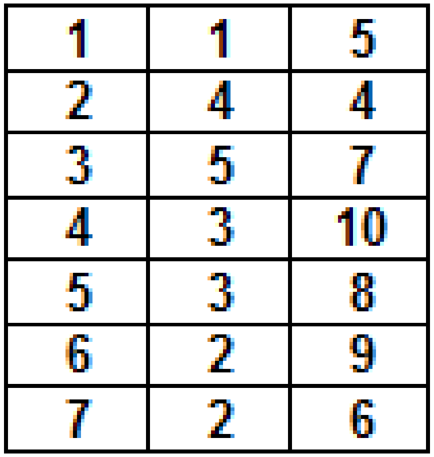
Generate a random number I got 0.0022 using the Random number generator Program

Node 1 gets selected for the second link

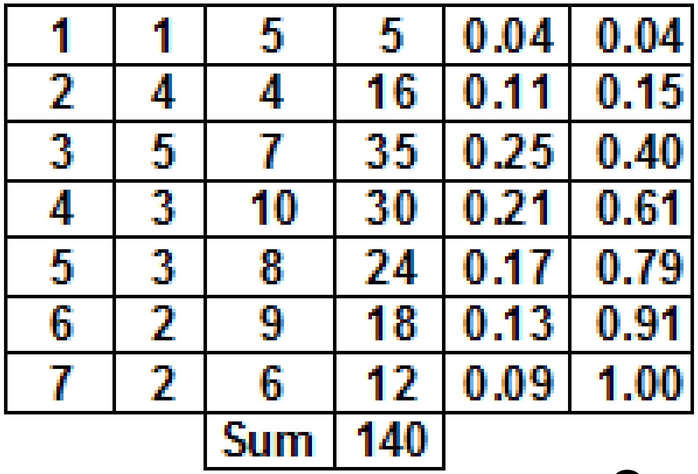
The newly joining node gets connected to nodes 2 and 1 (first and second link respectively)

Example-1: BA & BB Model

BB Model (First Link)



ID	Degree	Fitness
1	1	5
2	4	4
3	5	7
4	3	10
5	3	8
6	2	9
7	2	6



Probability of a node getting the second link

Cumulative Probability

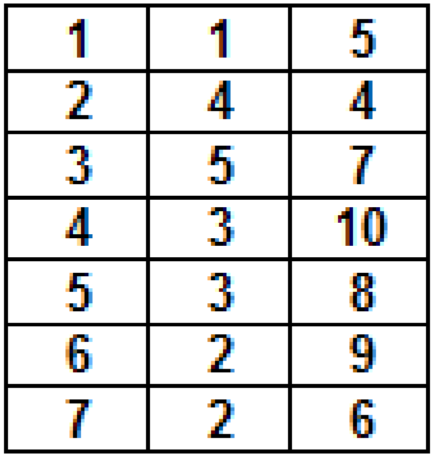
ID	Degree	Fitness	Degree * Fitness		
1	1	5	5	0.04	0.04
2	4	4	16	0.11	0.15
3	5	7	35	0.25	0.40
4	3	10	30	0.21	0.61
5	3	8	24	0.17	0.79
6	2	9	18	0.13	0.91
7	2	6	12	0.09	1.00
		Sum	140		

Generate a random number I got 0.2584 using the Random number generator Program I gave you.

Node 3 gets selected for the first link

Example-1: BA & BB Model

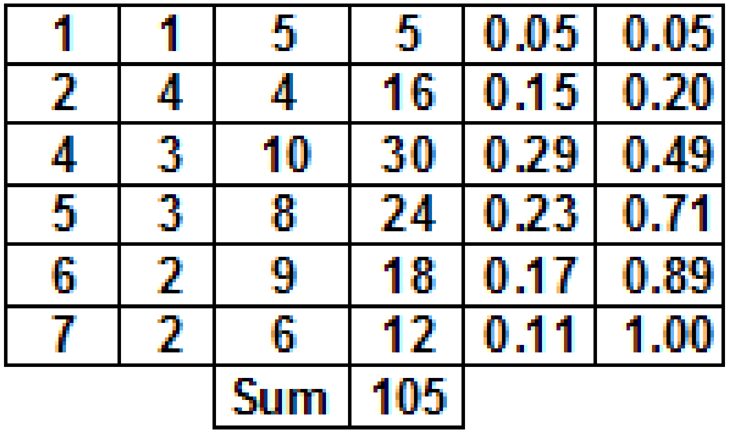
BB Model
(Second Link)



ID	Degree	Fitness
1	1	5
2	4	4
3	5	7
4	3	10
5	3	8
6	2	9
7	2	6

Probability of a node getting the second link

Cumulative Probability



ID	Degree	Fitness	Degree * Fitness		
1	1	5	5	0.05	0.05
2	4	4	16	0.15	0.20
4	3	10	30	0.29	0.49
5	3	8	24	0.23	0.71
6	2	9	18	0.17	0.89
7	2	6	12	0.11	1.00
		Sum	105		

Generate a random number I got 0.4885 using the Random number generator Program I gave you.

The newly joining node gets connected to nodes 3 and 4 (first and second link respectively)

Node 4 gets selected for the second link

Example-2: BA Model

- At some time unit t, if the degree of a node that joined the network at time 10 units is 50, compute the degree of the node that joined the network at time 100 units.
- Solution:
- $K_{10}(t) = m(t/10)^{1/2} = 50$

  ➔ $m*t^{1/2} = 50 * 10^{1/2} = 158.11$
- $K_{100}(t) = m(t/100)^{1/2} = mt^{1/2} / 100^{1/2}$

  $= 158.11/10 = 15.81$

Example-3: BA Model

- Consider a scale-free network that has evolved according to the BA model. Let there be two nodes P and Q such that the rate at which node P acquires new links is twice the rate at which node Q acquires new links. If node P joined the network at time 100 units, find the time at which node Q joined the network.
- Solution:

$$\frac{dk_i(t)}{dt} = \frac{m}{2}\frac{1}{\sqrt{t_i t}}$$

$$\boxed{\frac{dKi(t)}{dt}}_{P} = \boxed{2*\frac{dKi(t)}{dt}}_{Q}$$

$$\frac{m}{2}\frac{1}{\sqrt{t_P t}} = 2*\frac{m}{2}\frac{1}{\sqrt{t_Q t}}$$

$$\frac{1}{\sqrt{100}} = 2*\frac{1}{\sqrt{t_Q}}$$

$$\sqrt{t_Q} = 2*\sqrt{100}$$

$$t_Q = 4*100 = 400$$

Time at which node Q joined the network is 400 units.

Example-4: BB Model

- Consider the BB model for scale-free networks .
- Let the parameter $\beta(\eta i)$ for any node *i* be equal to the fitness of node *i*, *ηi*. Consider two nodes A and B such that the fitness of node B is twice the fitness of node A.
- Node A joins the network at time 10 units and node B joins the network at time 100 units.
- If the degree of the nodes increase for every time unit (when a new node joins), **what is the *minimum* value of the time unit starting from which the degree of node B would always be greater than the degree of node A**?

$$k_{\eta_i}(t,t_i) = m\left(\frac{t}{t_i}\right)^{\beta(\eta_i)}$$

KA (t, 10) = m $(t/10)^{\eta A}$
KB (t, 100) = m $(t/100)^{\eta B}$
Given that: ηB = 2*ηA
We want to find the minimum value of time instant t for which KB(t,100) > KA(t,10)

m $(t/100)^{\eta B} >$ m $(t/10)^{\eta A}$

$t^{\eta B} / 100^{\eta B} > t^{\eta A} / 10^{\eta A}$

$t^{\eta B} / t^{\eta A} > 100^{\eta B} / 10^{\eta A}$

$t^{2\eta A} / t^{\eta A} > 100^{2\eta A} / 10^{\eta A} = 10^{4\eta A} / 10^{\eta A}$

$t^{\eta A} > 10^{3\eta A}$

Hence, $t > 10^3$ ➔ t > 1000 time units

Example-5: BB Model

- Consider the BB model for scale-free networks .
- Let the degree of a node A be 50 at time 100 units. If the fitness of node A is 2, compute the degree of node A at time 400 units.

$$k_{\eta_i}(t, t_i) = m\left(\frac{t}{t_i}\right)^{\beta(\eta_i)}$$

KA(100, tA) = 50 = m (100 / tA)^2
KA(400, tA) = ?

KA(400, tA) = m (400 / tA)^2

From KA(100, tA) = 50
We get,

$$\frac{m}{t_A^2} = \frac{50}{100^2}$$

KA(400, tA) = $\dfrac{m}{t_A^2} * 400^2$

KA(400, tA) = $\dfrac{50}{100^2} * 400^2$

KA(400, tA) = 50*16 = 800

Example-6: BA Model

- At time 500 units, the following is the degree distribution of the nodes that joined at the time units indicated below. Determine the number of links added per node introduction (m) and the network's dynamical exponent (β). Estimate the degree of a node that joined the network at time 40 units.

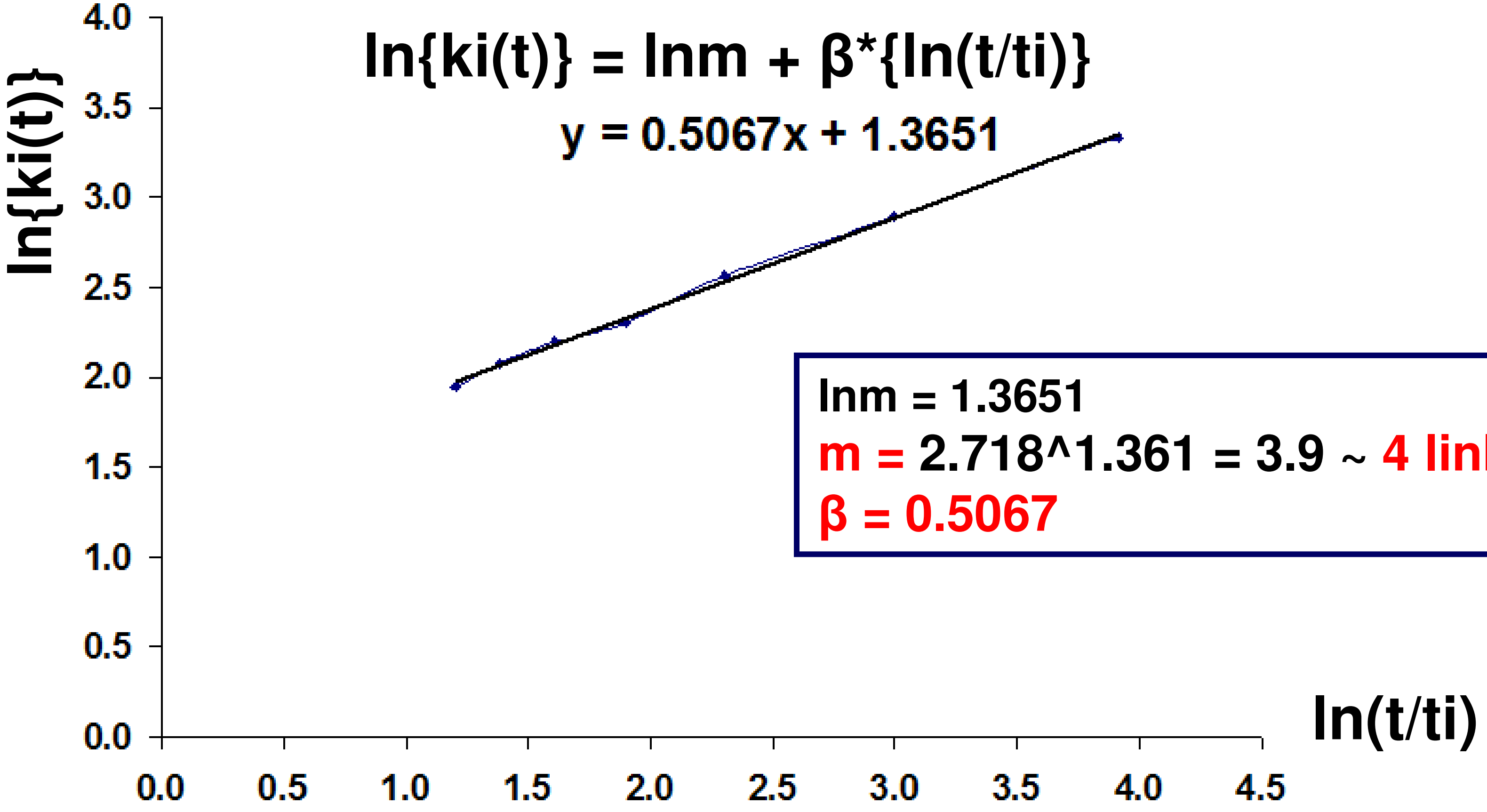
Node joining	**Degree at**	**ln(t/ti)**	**ln{ki(t)}**
Time, ti	**Time t = 500**	**ln(500/ti)**	
10	**28**	**3.912**	**3.332**
25	**18**	**2.996**	**2.890**
50	**13**	**2.302**	**2.565**
75	**10**	**1.897**	**2.302**
100	**9**	**1.609**	**2.197**
125	**8**	**1.386**	**2.079**
150	**7**	**1.204**	**1.946**

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{\beta}$$

ln{ki(t)} = lnm + β*{ln(t/ti)}

Y = Q + (slope)*X

Example-6 (1): BA Model



ln{ki(t)} = lnm + β*{ln(t/ti)}

y = 0.5067x + 1.3651

lnm = 1.3651
m = 2.718^1.361 = 3.9 ~ 4 links
β = 0.5067

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{\beta}$$

At time t = 500 units,
Degree of the node that joined the network at ti = 40 units
= 4 * (500/40)^0.5067 = 14.38 ~ 14.

Example-7: BB Model

- A node joined the network at time 10 units. Given below is the degree of the node at various time units. Determine the number of links added per node introduction and the fitness of the node. Under the BB model of evolution, assume the dynamical exponent value for a node is equal to the fitness of the node itself. Estimate the degree of the node at time 250 units.

Time Unit	Degree at	ln(t/ti)	ln{ki(t)}
t	Time t	ln(t/10)	
50	52	1.609	3.951
75	93	2.015	4.533
100	142	2.302	4.956
125	196	2.526	5.278
150	256	2.708	5.545
175	320	2.862	5.768
200	388	2.996	5.961

Given ti = 10

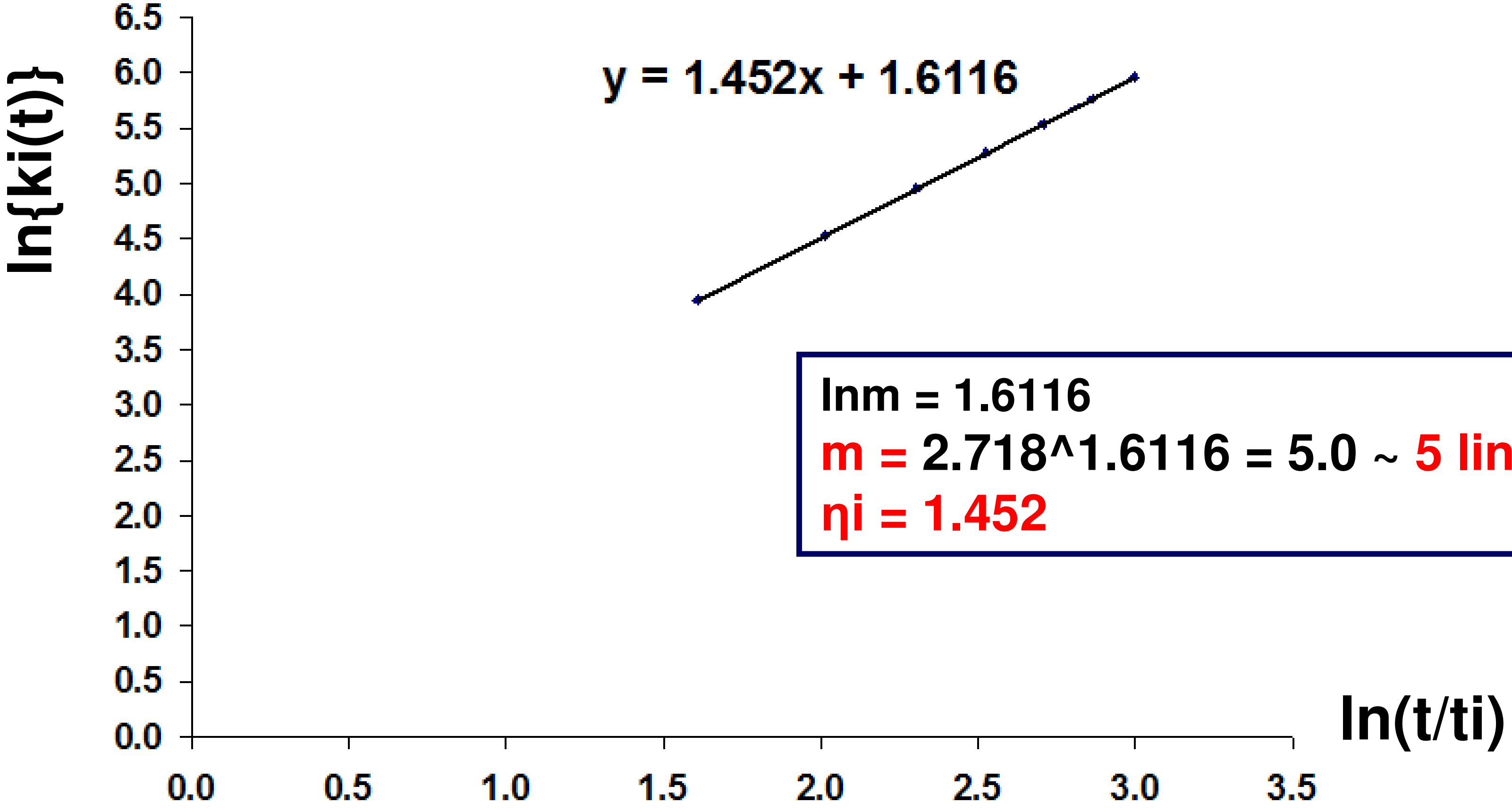
β(ηi) = ηi

$$k_{\eta_i}(t, t_i) = m\left(\frac{t}{t_i}\right)^{\beta(\eta_i)}$$

ln{ki(t)} = lnm + ηi*{ln(t/ti)}

Y = Q + (slope)*X

Example-7 (1): BB Model



lnm = 1.6116

m = 2.718^1.6116 = 5.0 ~ 5 links

ηi = 1.452

$$k_{\eta_i}(t,t_i) = m\left(\frac{t}{t_i}\right)^{\beta(\eta_i)}$$

At time t = 250 units,
Degree of the node that joined the network at ti = 10 units
= 5 * (250/10)^1.452 = 535.5 ~ 536.

Small World Networks

Dr. Natarajan Meghanathan
Professor of Computer Science
Jackson State University, Jackson, MS
E-mail: natarajan.meghanathan@jsums.edu

Small-World Networks

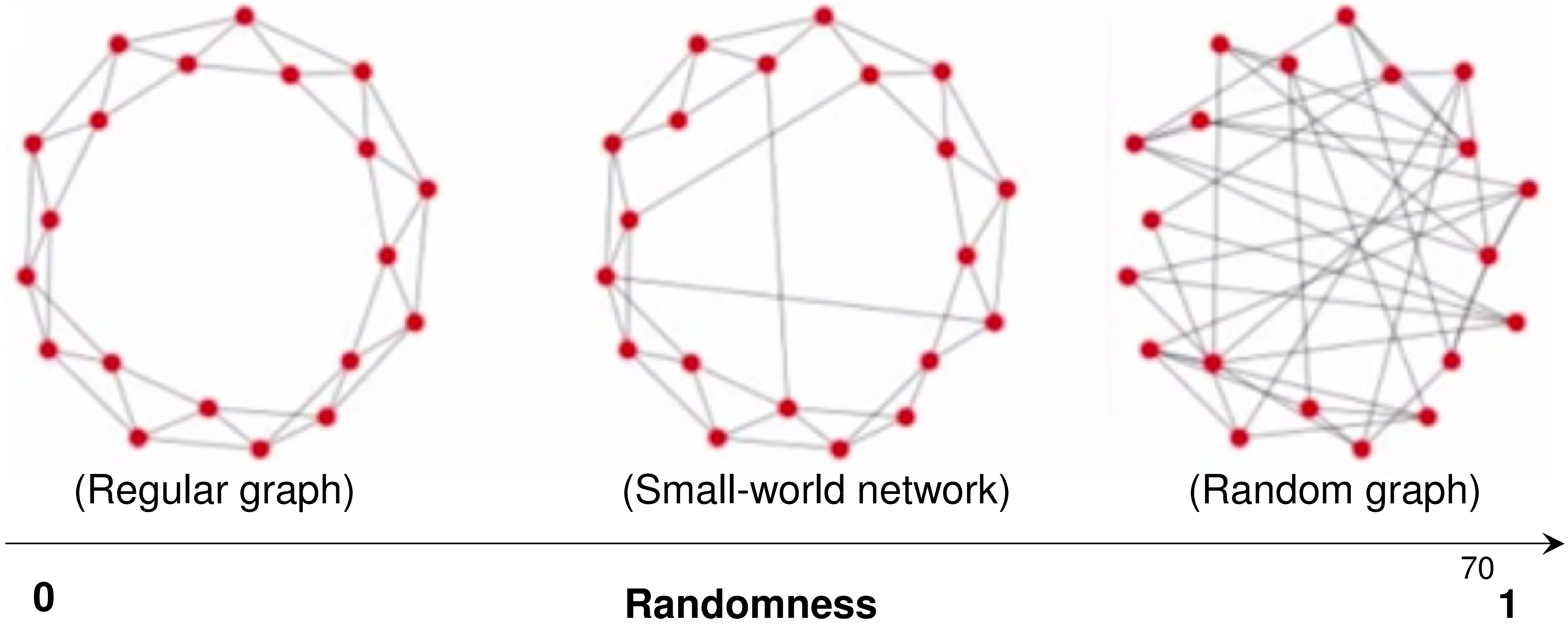
- A small-world network is a type of graph in which most nodes are not neighbors of one another, but most nodes can be reached from every other by a small number of hops.

- Specifically, a small-world network is defined to be a network where the typical distance L (the number of hops) between two randomly chosen nodes grows proportionally to the logarithm of the number of nodes in the network.

- Examples of Small-World Networks:
  - Road maps, food chains, electric power grids, metabolite processing networks, networks of brain neurons, voter networks, telephone call graphs, gene regulatory networks.

Small Worlds

- Two major properties of small world networks
  - High average clustering coefficient
    - The neighbors of a node are connected to each other
    - Nodes’ contacts in a social network tend to know each other.
  - Short average shortest path length
    - Shorter paths between any two nodes in the network



(Regular graph)

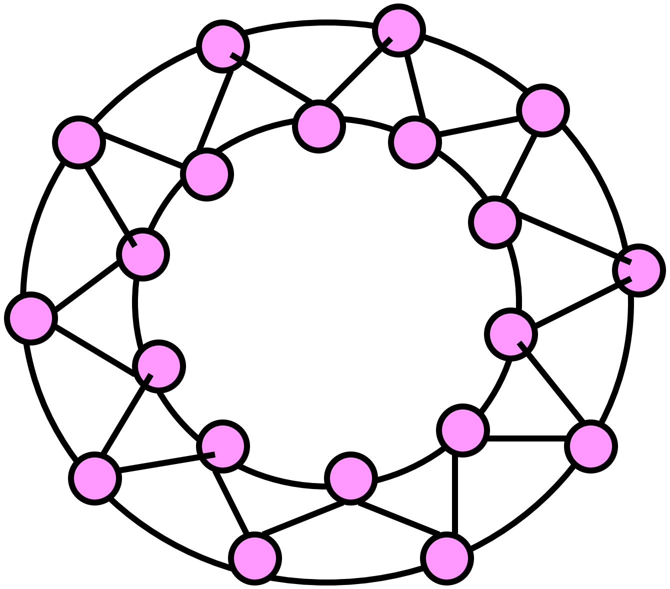
(Small-world network)

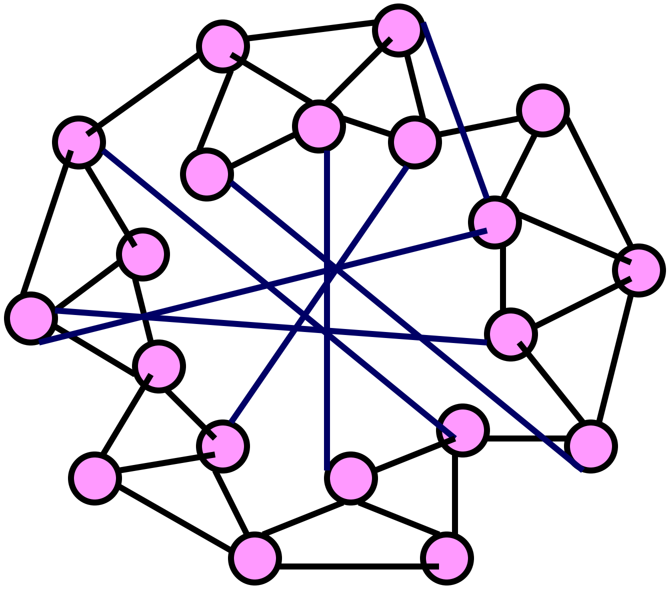
(Random graph)

Small Worlds

- Note that for the same number of nodes and edges, we could create:
  - Random graphs (with edges arbitrarily inserted between any two nodes) and
  - Regular graphs (with some specific pattern of how edges are inserted between nodes)
- Regular graphs tend to have relatively high average clustering coefficient
- Random graphs tend to have relatively low average shortest path length
- We could bring the best of the two graphs by generating a small world network as follows:
  - Remove a small fraction of the edges in a regular graph and re-insert them between any two randomly chosen nodes. This will not appreciably affect the average clustering coefficient of the nodes; but would significantly lower the average lengths of the shortest paths.

WS Model

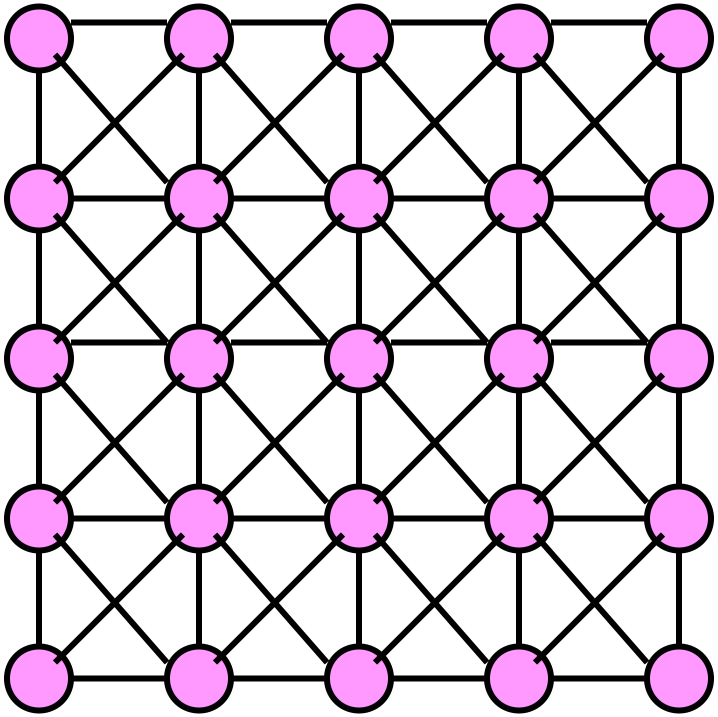


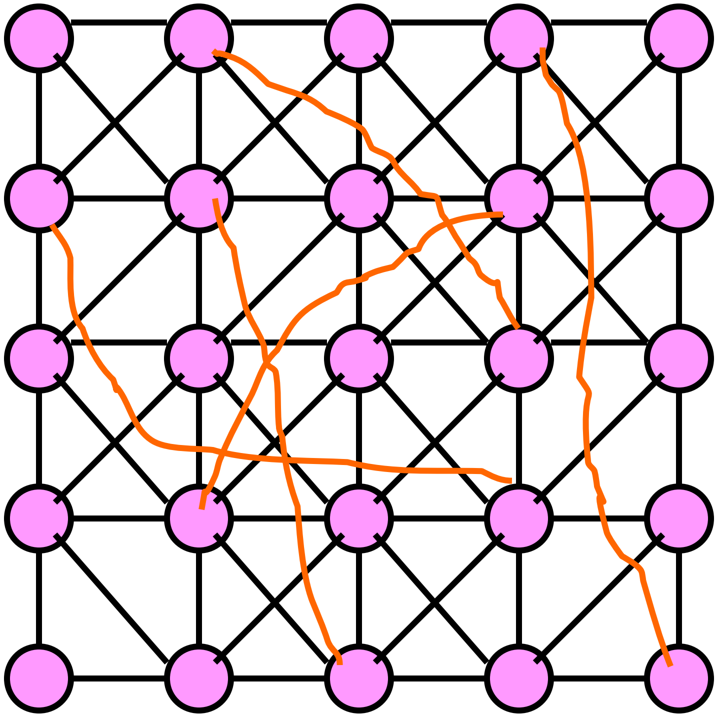


- Watts and Strogatz (WS) Model: The WS model interpolates between an ER graph and a regular ring lattice.
  - Let N be the number of nodes and K (assumed to be even) be the mean degree.
  - Assume N >> K >> ln(N) >> 1.
  - There is a rewiring parameter β (0 ≤ β ≤ 1).
  - Initially, let there be a regular ring lattice of N nodes, with K neighbors (K/2 neighbors on each side).
  - For every node $n_i = n_0, n_1, \ldots, n_{N-1}$, rewire the edge (n_i, n_j), where $i < j$, with probability β. Rewiring is done by replacing (n_i, n_j) with (n_i, n_k) where n_k is chosen uniform-randomly among all possible nodes that avoid self-looping and link duplication.

β = 0 ➔ Regular ring lattice
β = 1 ➔ Random network

WS Model for Grid Lattice (2-dim)

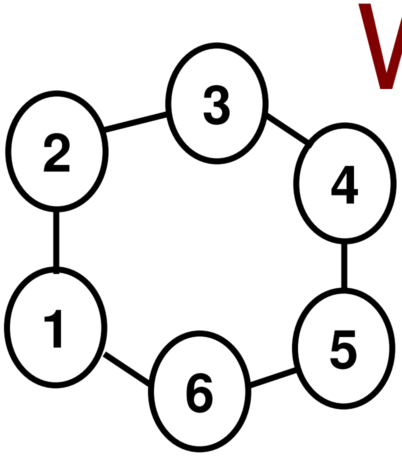




We have a nxn two-dimensional grid. The nodes are identified with lattice points i.e., a node v is identified with the lattice point (i, j) with i, j = {1, 2, …, n}

For every node u, we remove one of its associated edges to the neighbor nodes with a probability β and connect the node to a randomly chosen node v, as long as there is no self-loops and link duplication.

WS Model Illustration



Link	Random #	Selected
1 – 2	0.933	NO
1 – 6	0.907	NO
2 – 3	0.541	YES
3 – 4	0.346	YES
4 – 5	0.835	NO
5 – 6	0.361	YES

We select a link for rewiring if the random number generated for the link is less than or equal to the rewiring probability

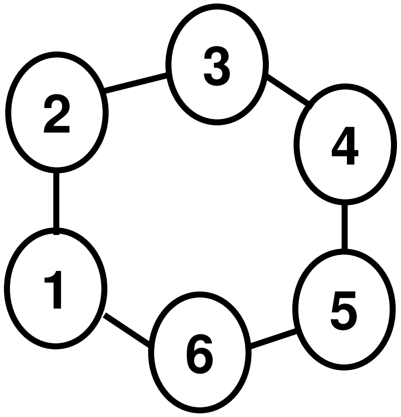
Given Network

Links (u, v): u < v

1 – 2
1 – 6
2 – 3
3 – 4
4 – 5
5 – 6

Let the rewiring Probability (β) be 0.7

Rewiring (#1): Link 2 – 3



We cannot pick vertices 3 and 1 as they are connected to vertex 2 in the original graph.

The candidates are: 4, 5, 6

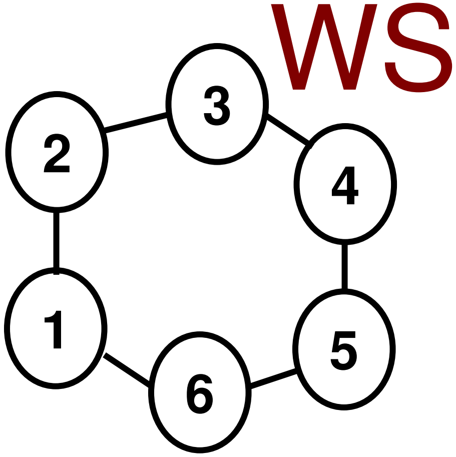
There is a 1/3 probability for each of them to be considered.

Cand.	Prob.	Cum Prob.
4	1/3	1/3 = 0.33
5	1/3	2/3 = 0.67
6	1/3	3/3 = 1.00

Generate a random Number = 0.422 →

Vertex 2 is rewired to Vertex 5.

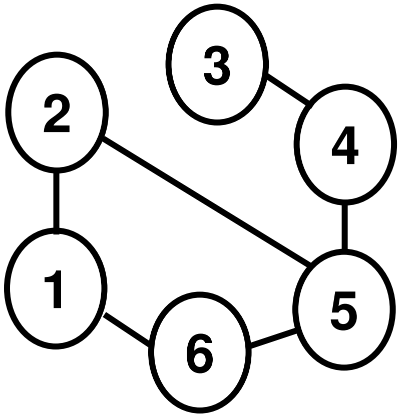
WS Model Illustration (1)



Given Network

Link	Random #	Selected
1 – 2	0.933	NO
1 – 6	0.907	NO
2 – 3	0.541	YES
3 – 4	0.346	YES
4 – 5	0.835	NO
5 – 6	0.361	YES

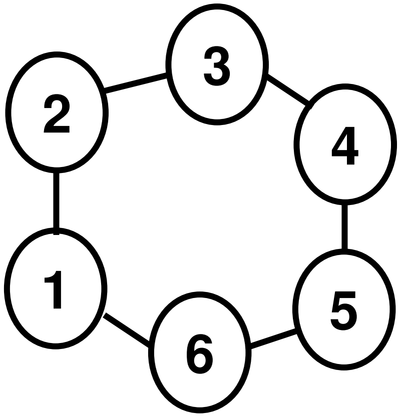
We select a link for rewiring if the random number generated for the link is less than or equal to the rewiring probability



After Rewiring # 1

Rewiring (#2): Link 3 – 4

We cannot pick vertices 4 and 2 as they are connected to vertex 3 in the original graph.
The candidates are: 1, 5, 6
There is a 1/3 probability for each of them to be considered.

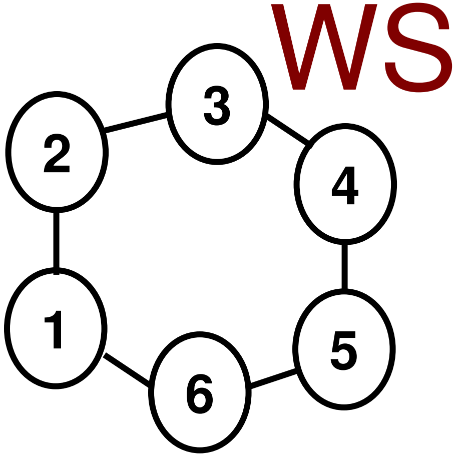


Cand.	Prob.	Cum Prob.
1	1/3	1/3 = 0.33
5	1/3	2/3 = 0.67
6	1/3	3/3 = 1.00

Generate a random Number = 0.665 →

Vertex 3 is rewired to Vertex 5.

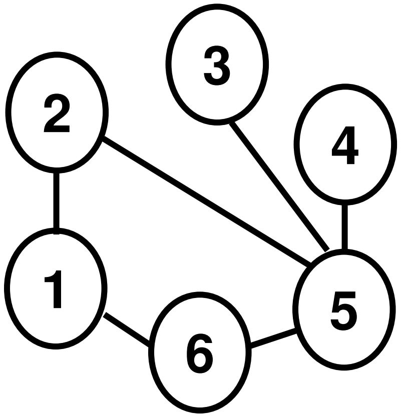
WS Model Illustration (2)



Given Network

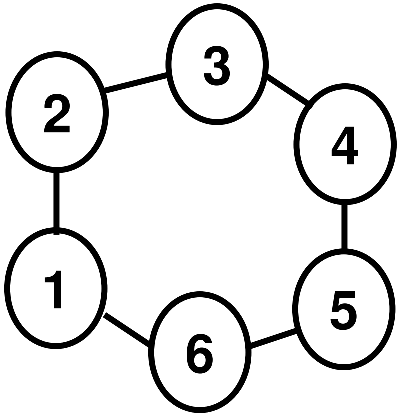
Link	Random #	Selected
1 – 2	0.933	NO
1 – 6	0.907	NO
2 – 3	0.541	YES
3 – 4	0.346	YES
4 – 5	0.835	NO
5 – 6	0.361	YES

We select a link for rewiring if the random number generated for the link is less than or equal to the rewiring probability



After Rewiring # 2

Rewiring (#3): Link 5 – 6

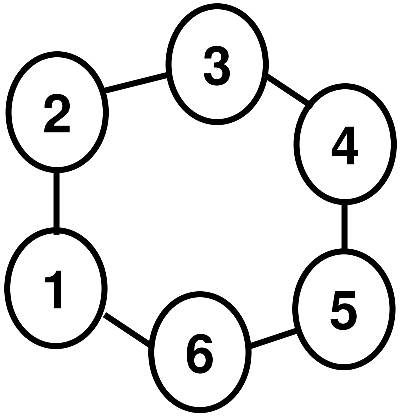


We cannot pick vertices 6 and 4 as they are connected to vertex 5 in the original graph as well cannot pick vertices 2 and 3 as they are connected to vertex 5 in the rewired graph. Hence, the only candidate vertex available is vertex 1.

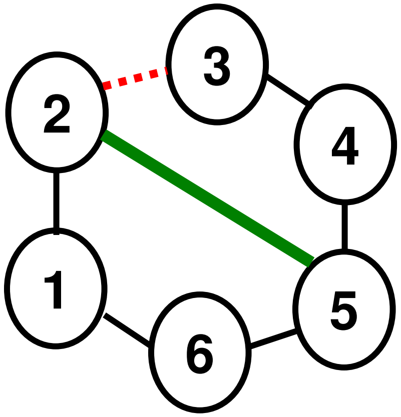
Cand.	Prob.	Cum Prob.
1	1/1	1/1 = 1.0

Vertex 5 is rewired to Vertex 1.

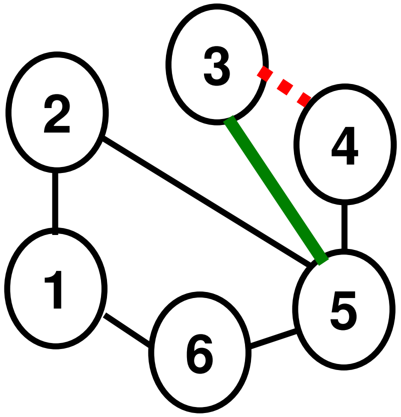
Generate a random Number = 0.763



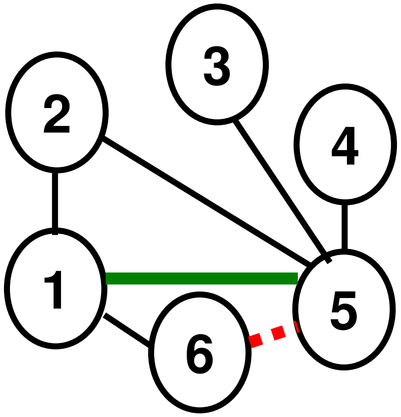
Given Network



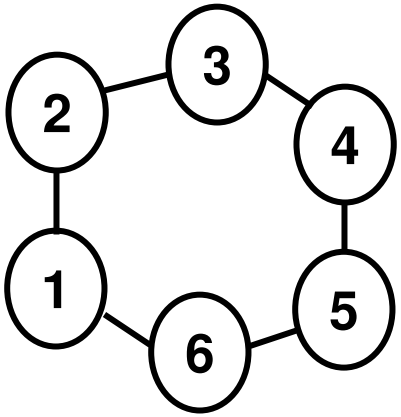
Rewiring # 1



Rewiring # 2

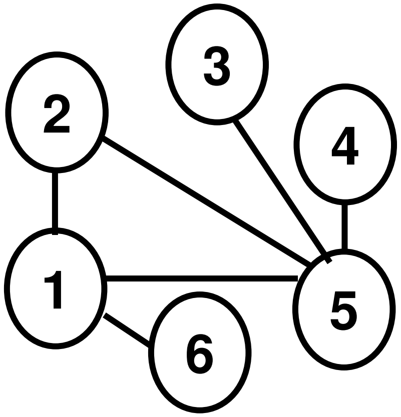


Rewiring # 3



Avg. Path Length = 54/ (6*5) = 1.8

	1	2	3	4	5	6	Sum
1	0	1	2	3	2	1	9
2	1	0	1	2	3	2	9
3	2	1	0	1	2	3	9
4	3	2	1	0	1	2	9
5	2	3	2	1	0	1	9
6	1	2	3	2	1	0	9

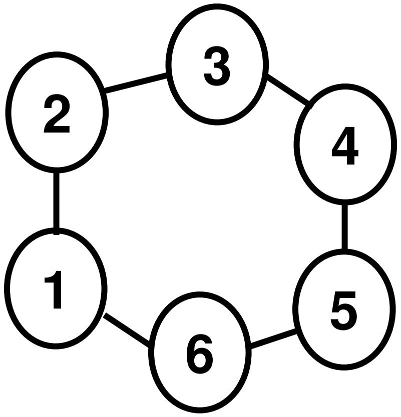


Avg. Path Length = 52/ (6*5) = 1.73

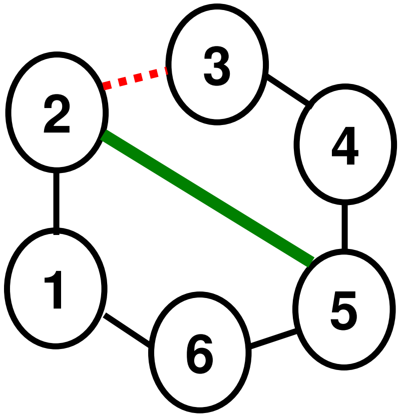
	1	2	3	4	5	6	Sum
1	0	1	2	2	1	1	7
2	1	0	2	2	1	2	8
3	2	2	0	2	1	3	10
4	2	2	2	0	1	3	10
5	1	1	1	1	0	2	6
6	1	2	3	3	2	0	11

Limitations of the WS Model

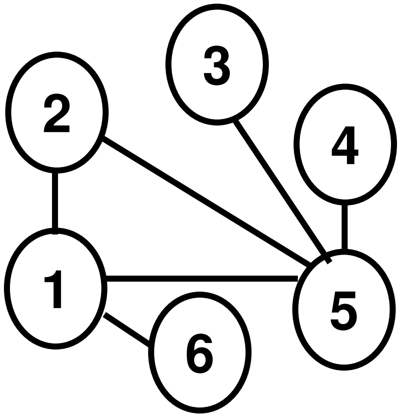
- The WS model introduced the notion of random edges to infuse shorter path lengths amidst larger clustering coefficient.
- However, the long-range edges span between any two nodes in the network and do not mimic the edges of different lengths seen in real-world networks (like in the US road map or airline map).
  - Path lengths could not be as small as they are in real networks.
  - Need some edges to nodes that are few hops away, rather than edges to some arbitrarily chosen nodes.



Given Network



After Rewiring # 1



After Rewiring # 3

Avg. Path Length = 52/ (6*5) = 1.73

Vertex 5 is chosen for rewiring to Vertex 2 even though the two vertices are three hops away. There were two Candidate Vertices (4 and 6) two hops away.

The number of paths of length 3 (i.e., 3 hops) reduced from 3 to 2
Original Net: (1, 4); (2, 5); (3, 6)
Rewired Net: (3, 6); (4, 6)

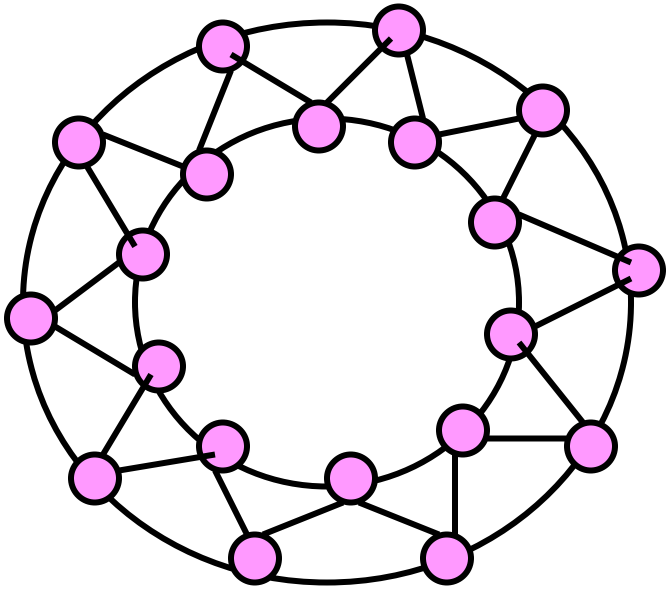
Small-World Network: WS Model

- The underlying lattice structure of the model produces a locally clustered network, and the random links dramatically reduce the average path lengths
- The algorithm introduces about (βNK/2) non-lattice edges.
- Average Path Length (β):
  - Ring lattice L(0) = (N/2K) >> 1
  - Random graph L(1) = (ln N / ln K)
  - For 0 < β < 1, the average path length reduces significantly even for smaller values of β.
- Clustering Coefficient (β):

$$C(0) = \frac{3(K-2)}{4(K-1)} \qquad C'(\beta) = C(0) * (1-\beta)^3$$

  - For 0 < β < 1, the clustering coefficient remains close to that of the regular lattice for low and moderate values of β and falls only at relatively high β.
- For low-moderate values of β, we thus capture the small-world phenomenon where the average path length falls rapidly, while the clustering coefficient remains fairly high.

Avg. Path Length and Clus. Coeff.



N = 20 nodes; K = 4

<u>Avg. Path Length</u>

Ring Lattice (Regular Net)
= N / 2K = 20 / (2*4) = 2.5

Random Network
= ln N / ln K = ln(20)/ln(4) = 2.16

$$C(0) = \frac{3(K-2)}{4(K-1)}$$

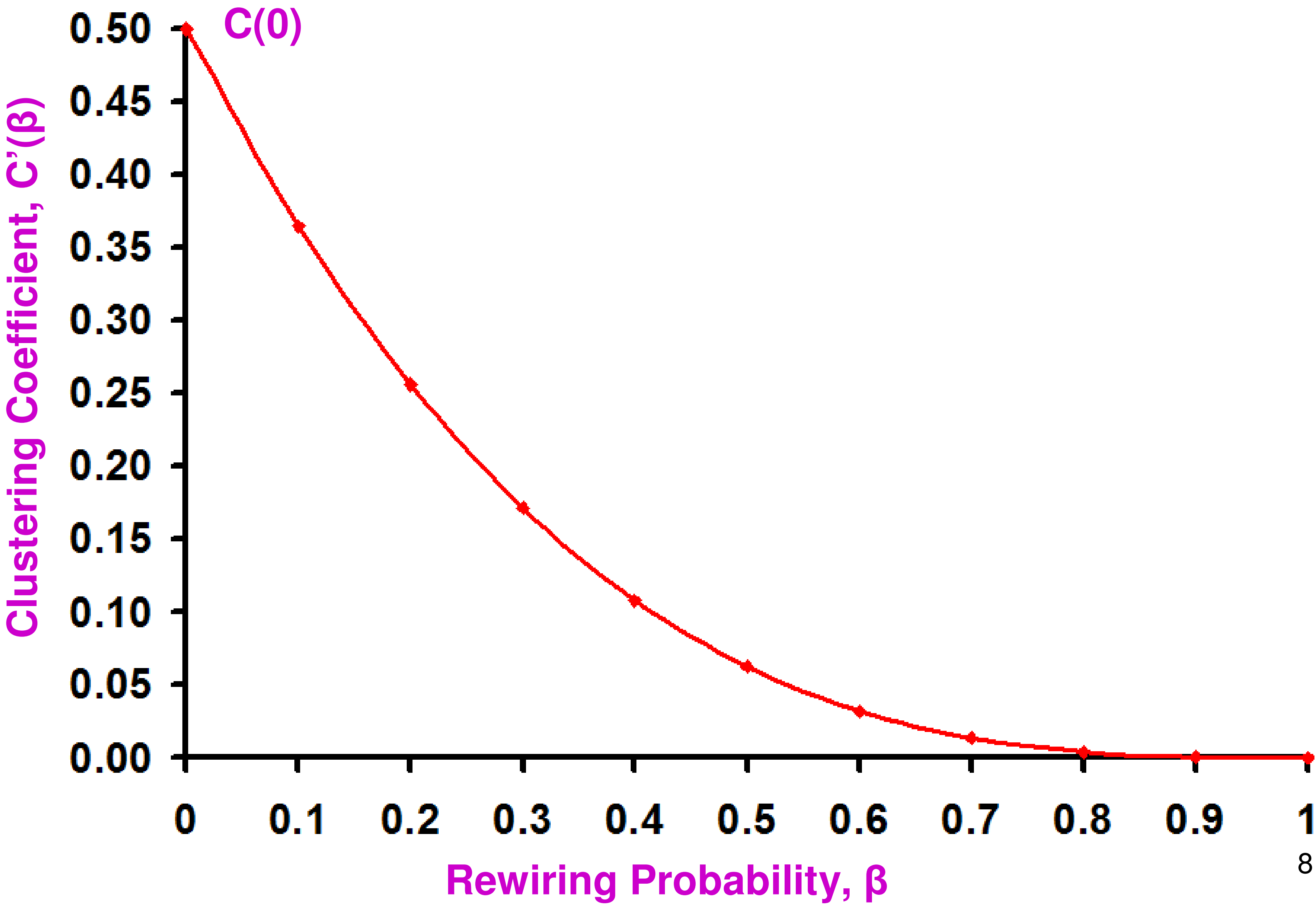
$$C'(\beta) = C(0) * (1-\beta)^3$$

C(0) = 3 (4 – 2) / 4 (4 – 1)

C(0) = 3 * 2 / 4 * 3 = 0.5

C’(0.1) = 0.5 * (1 – 0.1)^3 = 0.3645
C’(0.2) = 0.5 * (1 – 0.2)^3 = 0.256
C’(0.5) = 0.5 * (1 – 0.5)^3 = 0.0625
C’(0.9) = 0.5 * (1 – 0.9)^3 = 0.0005

Rewiring Prob. Vs. Clus. Coeff.



Enhancement to the WS Model

- In addition to the re-wiring parameter β, another parameter called the clustering exponent (q) is introduced.
- An (u, v) edge is selected for re-wiring with a probability β. After being selected, we do not randomly re-wire u with a node w. Instead, we pick a pair (u, w) for re-wiring with a probability (weight) of $[d(u, w)^{-q}]$ / 2logn, where
  - For optimal results, *q* must be the dimensionality of the network modeled. **For a ring lattice, q = 1.**
  - *n* is the number of nodes in the network.
  - d(u, w) is the minimum number of hops between u and w in the original network layout (before enhancement)
    - The ring lattice is a single-dimension network
    - A grid is a two-dimensional network.

Example 1: Small-World Model

- Consider a regular ring lattice of degree 8 for every node. This regular graph is transformed to a small-world network by arbitrarily re-wiring the edges with probability β. Let the clustering coefficient of the small-world network generated out of this re-wiring be 0.4. **Determine the re-wiring probability β**.

$$C(0) = \frac{3(K-2)}{4(K-1)}$$

$$\mathbf{C(0) = \frac{3\,(8-2)}{4\,(8-1)} = \frac{3*6}{4*7} = 0.643}$$

$$C'(\beta) = C(0) * (1-\beta)^3$$

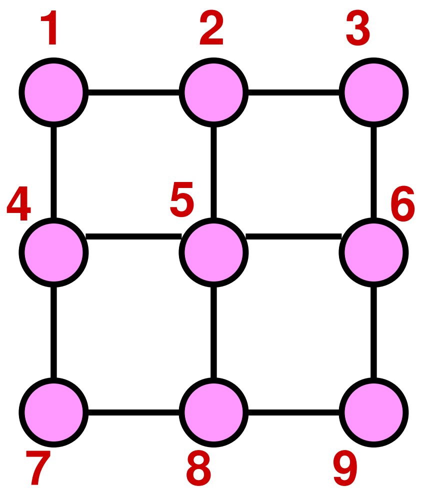
$$\mathbf{C'(\beta) = 0.4 = 0.643 * (1-\beta)^3}$$

$$\mathbf{(1-\beta)^3 = 0.4/0.643 = 0.622}$$

$$\mathbf{1-\beta = (0.622)^{1/3}}$$
$$\mathbf{1-\beta = 0.854}$$

$$\mathbf{\beta = 0.146}$$

Ex -2: Enhanced Small-World Model



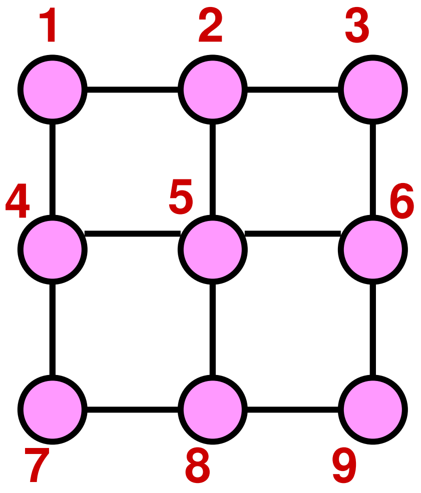
	1	2	3	4	5	6	7	8	9
1	0	1	2	1	2	3	2	3	4
2		0	1	2	1	2	3	2	3
3			0	3	2	1	4	3	2
4				0	1	2	1	2	3
5					0	1	2	1	2
6						0	3	2	1
7							0	1	2
8								0	1
9									0

Weight for a pair (u, w) at a distance d(u, w) is
$[d(u, w)^{-q}] / 2logn$

We have q = 2
and n = 9

d(u, w)	$[d(u, w)^{-q}] / 2logn$
1	(1^-2) / (2 * log 9) = 0.524
2	(2^-2) / (2 * log 9) = 0.131
3	(3^-2) / (2 * log 9) = 0.058
4	(4^-2) / (2 * log 9) = 0.033

Dist.	Orig. Weight	Pairs
1	0.524	(1, 2); (1, 4); (2, 3); (2, 5); (3, 6); (4, 5); (4, 7); (5, 6); (5, 8); (6, 9); (7, 8); (8, 9)
2	0.131	(1, 3); (1, 5); (1, 7); (2, 4); (2, 6); (2, 8); (3, 5); (3, 9); (4, 6); (4, 8); (5, 7); (5, 9); (6, 8); (7, 9)
3	0.058	(1, 6); (1, 8); (2, 7); (2, 9); (3, 4); (3, 8); (4, 9); (6, 7)
4	0.033	(1, 9); (3, 7)



Dist.	Orig. Weight	Pairs
1	0.524	(1, 2); (1, 4); (2, 3); (2, 5); (3, 6); (4, 5); (4, 7); (5, 6); (5, 8); (6, 9); (7, 8); (8, 9)
2	0.131	(1, 3); (1, 5); (1, 7); (2, 4); (2, 6); (2, 8); (3, 5); (3, 9); (4, 6); (4, 8); (5, 7); (5, 9); (6, 8); (7, 9)
3	0.058	(1, 6); (1, 8); (2, 7); (2, 9); (3, 4); (3, 8); (4, 9); (6, 7)
4	0.033	(1, 9); (3, 7)

Assume we want to rewire the edge 1 – 2
The pairs that could be considered are
(everything except (1, 2) and (1, 4)):
(1, 3); (1, 5); (1, 6); (1, 7); (1, 8); (1, 9)

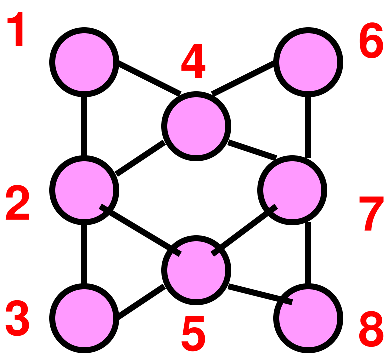
Scaled Prob. = Orig. Weight / 0.542

Pair	Orig. Weight	Scaled Prob.	Cum. Scaled Prob.
1 – 3	0.131	0.242	0.242
1 – 5	0.131	0.242	0.484
1 – 6	0.058	0.107	0.591
1 – 7	0.131	0.242	0.833
1 – 8	0.058	0.107	0.940
1 – 9	0.033	0.060	1.000
Sum	0.542		

Generate a random number: 0.2525

We will rewire node 1 to node 5.

Ex -3: Enhanced Small-World Model



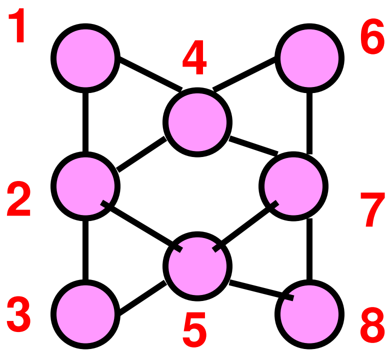
	1	2	3	4	5	6	7	8
1	0	1	2	1	2	2	2	3
2		0	1	1	1	2	2	2
3			0	2	1	3	2	2
4				0	2	1	1	2
5					0	2	1	1
6						0	1	2
7							0	1
8								0

Weight for a pair (u, w) at a distance d(u, w) is
$[d(u, w)^{-q}] / 2\log n$

We have q = 1
and n = 8

d(u, w)	$[d(u, w)^{-q}] / 2\log n$
1	(1^-1) / (2 * log 8) = 0.554
2	(2^-1) / (2 * log 8) = 0.277
3	(3^-1) / (2 * log 8) = 0.185

Dist.	Orig. Weight	Pairs
1	0.554	(1, 2); (1, 4); (2, 3); (2, 4); (2, 5); (3, 5); (4, 6); (4, 7); (5, 7); (5, 8); (6, 7); (7, 8)
2	0.277	(1, 3); (1, 5); (1, 6); (1, 7); (2, 6); (2, 7); (2, 8); (3, 4); (3, 7); (3, 8); (4, 5); (4, 8); (5, 6); (6, 8)
3	0.185	(1, 8); (3, 6)



Dist.	Orig. Weight	Pairs
1	0.554	(1, 2); (1, 4); (2, 3); (2, 4); (2, 5); (3, 5); (4, 6); (4, 7); (5, 7); (5, 8); (6, 7); (7, 8)
2	0.277	(1, 3); (1, 5); (1, 6); (1, 7); (2, 6); (2, 7); (2, 8); (3, 4); (3, 7); (3, 8); (4, 5); (4, 8); (5, 6); (6, 8)
3	0.185	(1, 8); (3, 6)

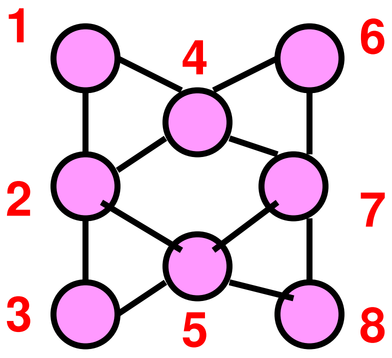
Assume we want to rewire the edge 3 – 5
The pairs that could be considered are
(1, 3); (3, 4); (3, 6); (3, 7); (3, 8)

Scaled Prob. = Orig. Weight / 1.293

Pair	Orig. Weight	Scaled Prob.	Cum. Scaled Prob.
1 – 3	0.277	0.214	0.214
3 – 4	0.277	0.214	0.428
3 – 6	0.185	0.143	0.571
3 – 7	0.277	0.214	0.785
3 – 8	0.277	0.214	0.999
Sum	1.293		

Generate a random number: 0.9338

We will rewire node 3 to node 8.



Dist.	Orig. Prob.	Pairs
1	0.554	(1, 2); (1, 4); (2, 3); (2, 4); (2, 5); (3, 5); (4, 6); (4, 7); (5, 7); (5, 8); (6, 7); (7, 8)
2	0.277	(1, 3); (1, 5); (1, 6); (1, 7); (2, 6); (2, 7); (2, 8); (3, 4); (3, 7); (3, 8); (4, 5); (4, 8); (5, 6); (6, 8)
3	0.185	(1, 8); (3, 6)

Assume we want to rewire the edge 5 – 7
The pairs that could be considered are
(1, 5); (4, 5); (5, 6)

Scaled Prob. = Orig. Weight / 0.831

Pair	Orig. Weight	Scaled Prob.	Cum. Scaled Prob.
1 – 5	0.277	0.333	0.333
4 – 5	0.277	0.333	0.667
5 – 6	0.277	0.333	1.000
Sum	0.831		

← Generate a random number: 0.9113

We will rewire node 5 to node 6.

Ex-4: Enhanced WS Model

- Consider the enhanced WS model for small-world networks. Let there be a regular graph that is transformed to a small-world network. For every edge (*u*, *v*) selected for re-wiring, the probability that a node *w* of distance 2 hops to *u* is picked for re-wiring is 0.2 and the probability that a node *w′* of distance 4 hops to *u* is picked for re-wiring is 0.08. Find the value for the **parameter *q*** in the enhanced WS model.

Prob. for a pair (u, w) at a distance d(u, w) is

$[d(u, w)^{-q}] / 2\log n$

Given: $P(u, w) = 0.2 = 2^{-q} / 2\log n$ …… (1)

$P(u, w') = 0.08 = 4^{-q} / 2\log n$ …. (2)

$$\frac{(1)}{(2)} \rightarrow \frac{0.2}{0.08} = \frac{2^{-q}}{4^{-q}} \rightarrow \frac{4^{q}}{2^{q}} = 2.5$$

$$\frac{(2^{2})^{q}}{2^{q}} = \frac{2^{2q}}{2^{q}} = 2^{q} = 2.5$$

$q = \ln(2.5) / \ln(2) = 1.322$