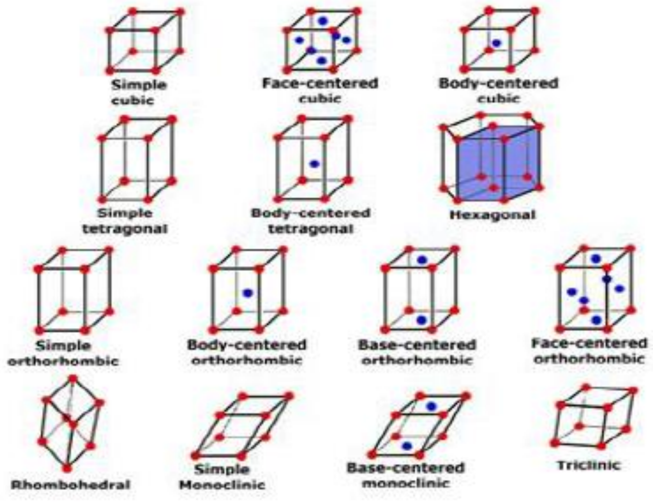
Lecture 6: Network Models I

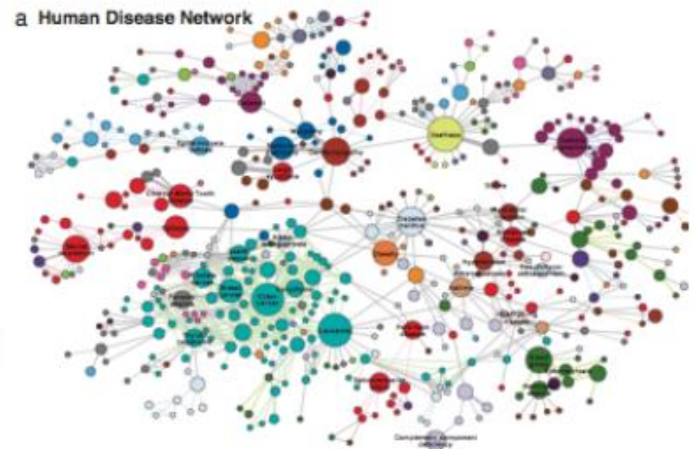
Complexity: Between Randomness and Order

Lattices



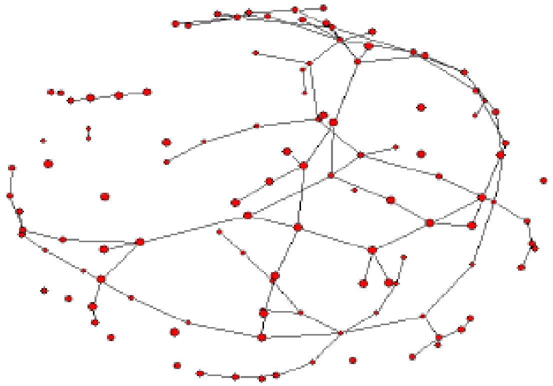
Regular networks
Symmetric

Complex Networks



Scale free networks
Small world
With communities
ENCODING
INFORMATION
IN THEIR
STRUCTURE

Random Graphs



Totally random
Binomial degree
distribution

Network Models:

A realistic network model should capture the three key properties of networks and hidden metric spaces:

Scale-free degree distribution + small-world property + high clustering

- The natural degree distribution of a random geometric graph embedded in hyperbolic space is a scale-free with characteristic degree 3…..
- The small-world property explains why hyperbolic hidden metric spaces underlying complex networks offer a natural embedding.….
- Clustering gives the clue to connect complex networks to underlying hidden metric spaces……..

Network Models:

What is a network model?

- Informally, a network model is a *process* (randomized or deterministic) for generating a graph

- Models of static graphs

  - input: a set of parameters Π, and the size of the graph n
  - output: a graph $G(\Pi, n)$

- Models of evolving graphs

  - input: a set of parameters Π, and an initial graph G_0
  - output: a graph G_t for each time t

Network Models:

- Equilibrium network models

  The number of nodes is fixed to N

  - Classical random graphs, Erdos and Renyi model
  - Watts-Strogatz model
  - Configuration model
  - ....

- Non-equilibrium network models

  The number of nodes N grows

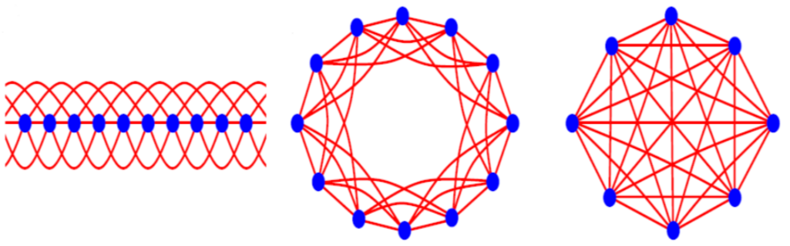
  - Classical random growing graphs
  - Preferential attachment, Barabasi-Albert model
  - ....

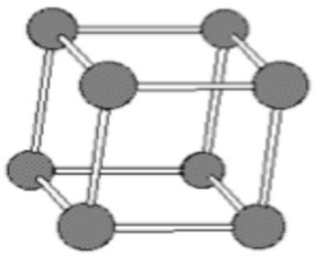
Regular Networks

Regular Networks

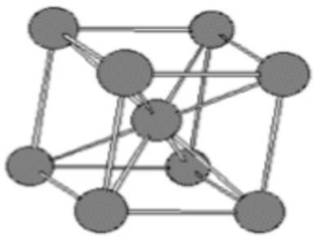
Regular graph: a graph where all nodes have the same degree.

Lattice: a regular network where all nodes are coupled to its nearest neighbor.





d dimensions



N = number of nodes
K = degree
C = clustering coefficient
d = dimension of the lattice
L = average path length

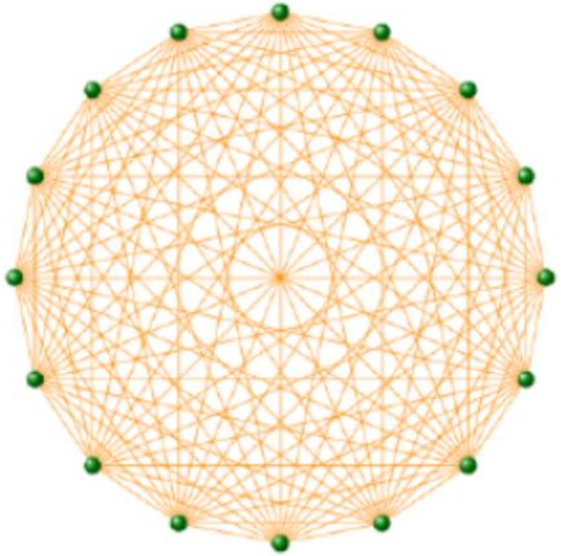
$$C = \frac{3(K-2d)}{4(K-d)}$$

(**if $K < 2N/3$)

$$L \sim N^{1/d}$$

Regular Networks

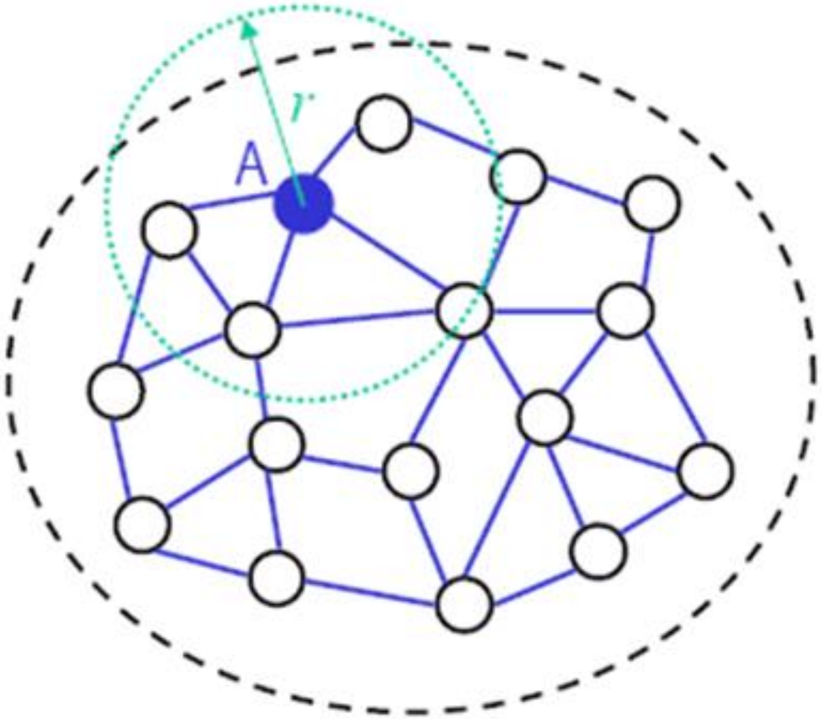
Example: Fully connected network



- In a fully connected network, every node is connected to any other nodes.
- Fully connected networks have the *LOWEST path length (L)* and *diameter (D)*:
  - $L = D = 1$
- Have *HIGHEST clustering coefficient*
  - $C = 1$
- And a PEAK degree distribution (at the largest possible constant)
- $k_{average} = N - 1\,,\ P(k) = \delta(k - N + 1)$
- Also the highest number of edges: $L = L_{max} = N(N-1)/2$
- It is a *complete graph*

Regular Networks

Example: Lattice



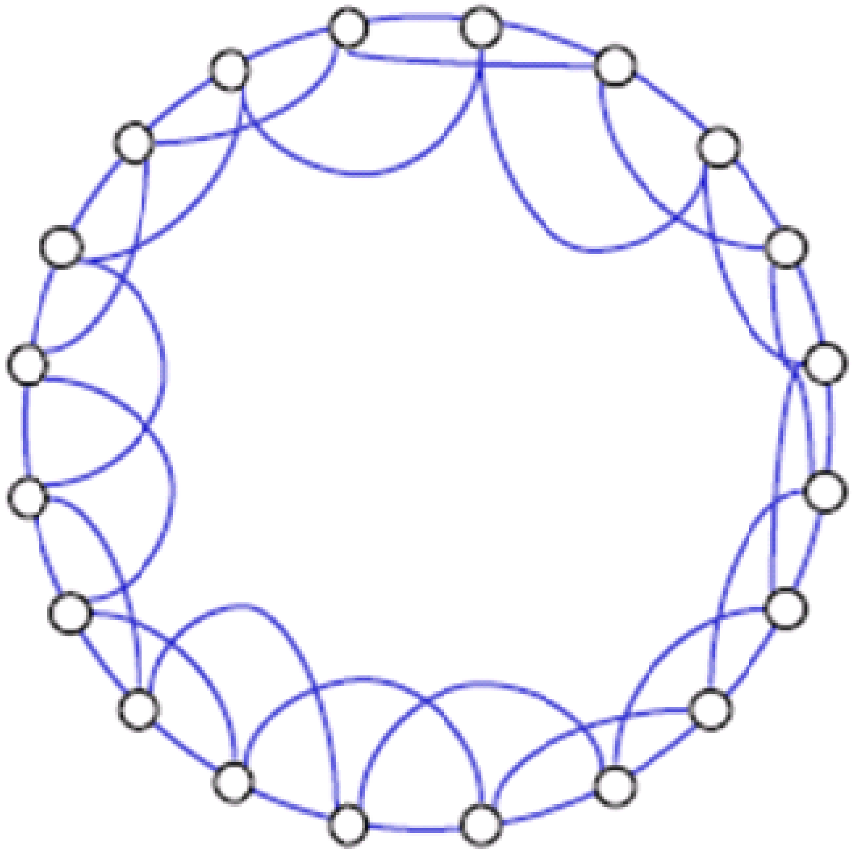
A 2-D lattice network

- A *lattice* network is generally structured against a geometric 2-D or 3-D background
- For example, each node is connected to its neighbors depending on the Euclidean distance

  $A \longleftrightarrow B \Longleftrightarrow d(A, B) \leq r$
- The radius r should be sufficiently small to stay far from a fully connected network, i.e., to keep a large diameter $D >> 1$

Regular Networks

Example: Lattice -- Ring World



A ring lattice with K = 4

- In a *ring lattice*, nodes are laid out on a circle and connected to their K nearest neighbors, with $K << N$

- *HIGH average path length:*
  $L \approx N/2K \sim N$ for $N >> 1$
  (mean between closest node $l = 1$ and antipode node $l = N/K$)

- HIGH clustering coefficient:
  $C \approx 0.75$ for $K >> 1$
  (mean between center with K edges and farthest neighbors with $K/2$ edges)

- *PEAK degree distribution (low value):*
  $k_{average} = K \quad P(k) = \delta(k - K)$

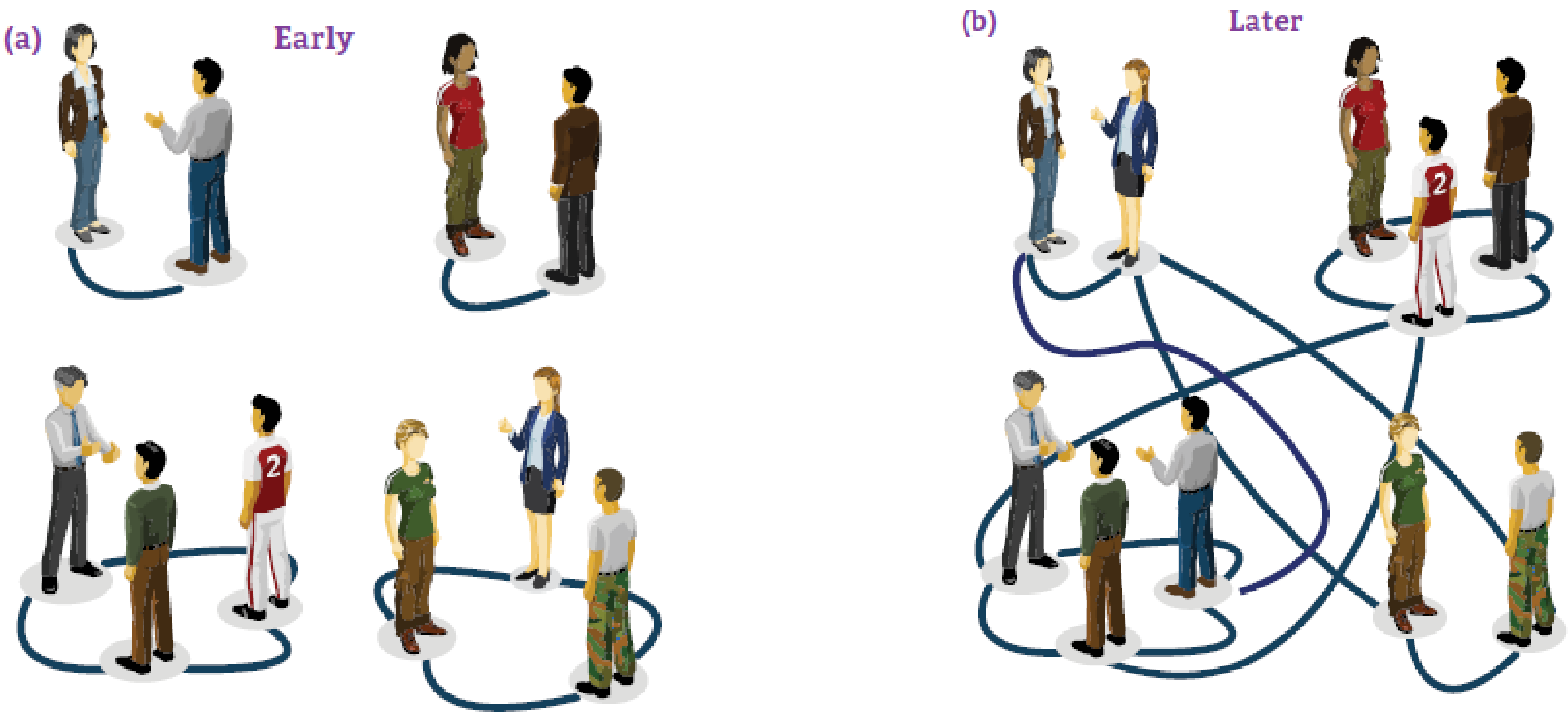
Random Networks

Random Networks

Complex networks are the outcome of
dynamical processes
that are intrinsically stochastic
but
are complex networks purely random?

What is a random network?

Random Networks

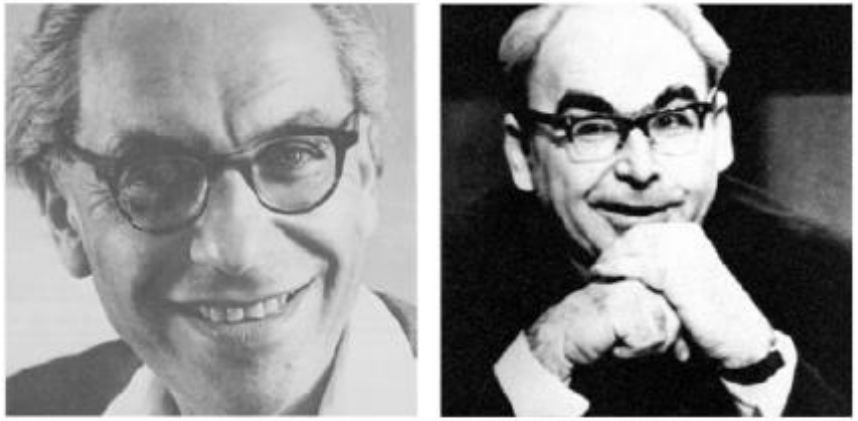


From a Cocktail Party to Random Networks
The emergence of an acquaintance network through random encounters at a cocktail party.

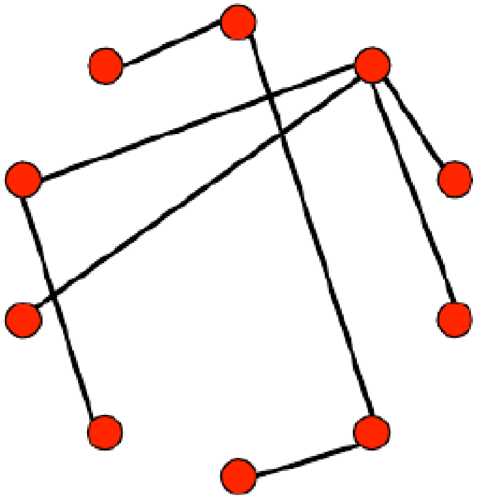
(a) Early on the guests form isolated groups.
(b) As individuals mingle, changing groups, an invisible network emerges that connects all of them into a single network.

Random Networks

Pál Erdös
(1913-1996)



Alfréd Rényi
(1921-1970)



Erdos-Renyi Model (1960)

Connect with probability *p*

In the figure, $p = 1/6$; $N = 10$

<k> ~ 1.5

Random Networks

Two versions:

Microcanonical Ensemble

$G(n, m)$ model: a graph is chosen uniformly at random from the collection of all graphs which have n nodes and m edges.

$$P(G) = \frac{1}{\binom{n(n-1)/2}{m}} \delta\left(\sum_{i<j} a_{ij}, m\right)$$

Canonical Ensemble

$G(n, p)$ model: a graph is constructed by connecting nodes randomly. Each edge is included in the graph with probability p independent from every other edge.

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

In the large N limit there is asymptotic equivalence between the ensembles!

The statistical properties of the two ensembles are the same!

Random Networks

- Some properties of the random graph $G(n, m)$ are straightforward to calculate: for instance, and the average degree is $\langle k \rangle = 2m/n$. Unfortunately, other properties are not so easy to calculate, and most mathematical work has actually been conducted on the $G(n, p)$ model, which is considerably easier to handle.

- $G(n, p)$ was first studied by Solomonoff and Rapoport (1951), but it is most closely associated with the names of Paul Erdos and Alfred Renyi, who published a celebrated series of papers about the model in the late 1950s and early 1960s. It is now commonly referred to as the "*Erdos-Renyi model*" or the "*Erdos-Renyi random graph*".

- Also sometimes called the "*Poisson random graph*" or the "*Bernoulli random Graph*", names that refer to the distributions of degrees and edges in the model.

Probability of Total Number of Links in the G(n,p) ensemble

➡ The probability of obtaining a graph with *L* Links in the *G(n,p)* ensembles is a binomial distribution

$$P(L) = \binom{n(n-1)/2}{L} p^L (1-p)^{n(n-1)/2-L}$$

Select *L* pair of nodes from *n(n-1)/2* possible pair of nodes

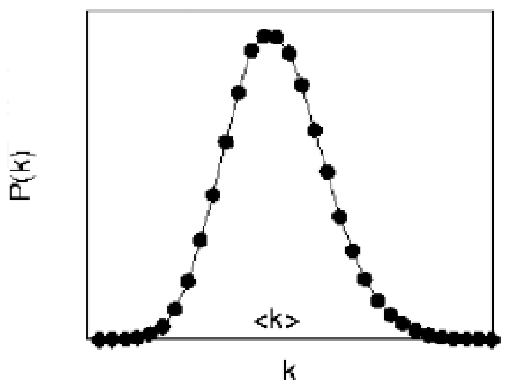
Probability of having *L* links between the selected pairs of nodes

Probability of the absence of links between the others *n(n-1)/2-L* pairs of nodes

Random Networks

Degree distribution in the G(n,p) ensemble

The degree distribution a graph in the *G(n,p)* ensembles is a binomial distribution



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

Select k nodes from (n-1) possible neighbors of a given node

Probability of having k links connected to the selected k nodes

Probability of not having links to the remaining (N - 1) - k nodes

Random Networks

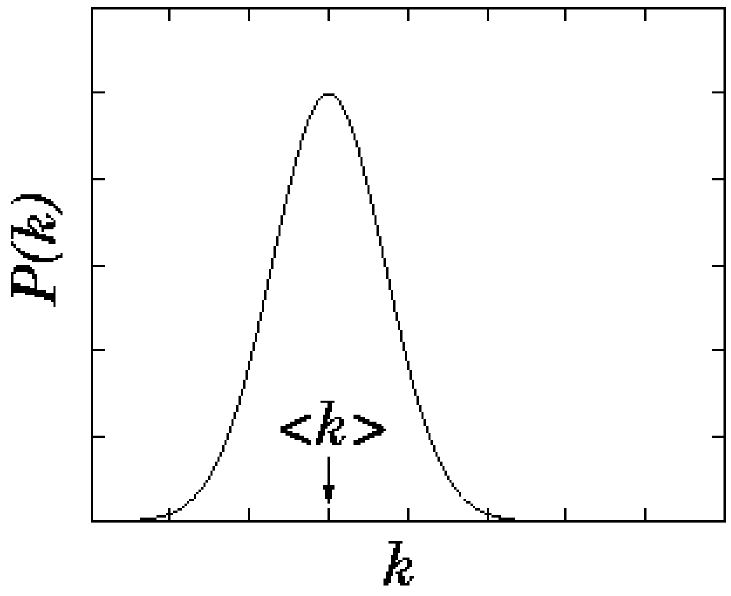
Degree distribution in the G(n,p) ensemble

Degree distribution for the whole network $$P(k) = \sum_{i=1}^{n} P_i(k)/n$$

Average degree: $< k > = c = \frac{2E}{n} = p(n-1) \approx pn$ (as $n \to \infty$)

n → ∞ such that <k> = constant ➔ Poisson distribution

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \to \frac{(np)^k e^{-np}}{k!}$$
$$= \frac{< k >^k}{k!} e^{-<k>}$$



Clustering coefficient: $C = \frac{c}{n-1}$

Random Networks

Degree distribution in the G(n,p) ensemble

Exact Result: Binomial Distribution

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

Large n limit: Poisson Distribution

$$P(k) = \frac{<k>^k}{k!} e^{-<k>}$$

MOMENTS OF THE DEGREE DISTRIBUTION

FINITE FLUCTUATIONS AROUND THE MEAN

$$\langle k \rangle = (N-1)p \approx c$$

$$\langle k(k-1) \rangle = p^2 (N-2)(N-1) \approx c^2$$

$$\sigma = \left(\langle k^2 \rangle - \langle k \rangle^2 \right)^{1/2} \approx c^{1/2}$$

$$\frac{\sigma}{\langle k \rangle} \approx \frac{1}{c^{1/2}}$$

$$\langle k \rangle = c$$

$$\langle k(k-1) \rangle = c^2$$

$$\sigma = \left(\langle k^2 \rangle - \langle k \rangle^2 \right)^{1/2} = c^{1/2}$$

$$\frac{\sigma}{\langle k \rangle} = \frac{1}{c^{1/2}}$$

Random Networks

Model a social network with a random graph

Consider a social network with average degree: <k> = *c* = 100

Assume that the degree distribution is Poisson, then the standard deviation is $\sigma = \sqrt{c} = \sqrt{100} = 10$

Observing a person with k=1000 friends implies observing an event which is

$$\frac{|k-c|}{\sigma} = \frac{1000-100}{10} = 90$$

standard deviations from the mean!!!

This is very unexpected!!

Distances in random graphs

Random graphs with $p = <k>/n$ have the small world distance property

$$< L_{max} > \cong \frac{\log n}{\log < k >}$$

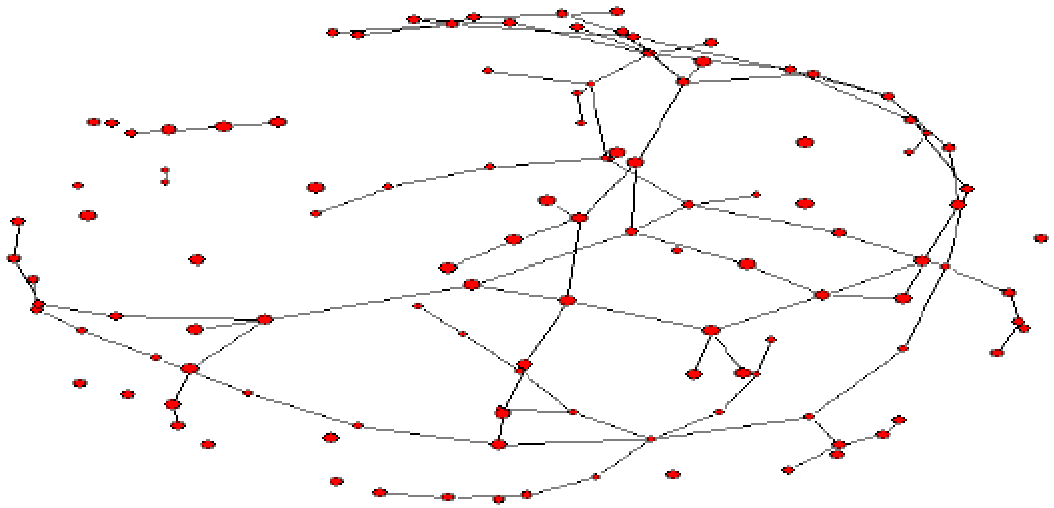
Diameter

The fact that the degree in these networks is homogeneous is not realistic for modeling most complex networks !!!

Random Networks

Giant Component

A connected component of a network is a subgraph in which any two nodes are connected to each other by at least one path and which is connected to no additional nodes of the network. The giant component is the connected component of the network which contains a number of nodes of the same order of magnitude of the total number of nodes in the network.



Giant Component

Denote by u the average fraction of vertices in the random graph that do *not* belong to the giant component. Thus if there is no giant component in our graph, we will have $u = 1$, and if there is a giant component we will have $u < 1$. Alternatively, one can regard u as the probability that a randomly chosen vertex in the graph does not belong to the giant component.

For a vertex i not to belong to the giant component it must not be connected to the giant component via any other vertex. That means that for every other vertex j in the graph either (a) i is not connected to j by an edge or, (b) i is connected to j but j is itself not a member of the giant component.

The probability of (a) is $1 - p$, the probability of not having an edge between i and j, and the probability of (b) is pu, => total probability of not being connected to the giant component via vertex j is $1- p + pu$.

Total probability of not being connected to the giant component via any of the n - 1 other vertices in the network is

$$u = (1 - p + pu)^{n-1} = [1 - \frac{c}{n-1}(1-u)]^{n-1}$$

Giant Component

Take logs of both sides, and in the large n limit

$$\ln u = (n-1)\ln\left[1 - \frac{c}{n-1}(1-u)\right] \cong -(n-1)\frac{c}{n-1}(1-u) = -c(1-u)$$

Taking exponentials of both sides,

$$u = e^{-c(1-u)}$$

Denote the fraction of vertices that are in the giant component to be $S = 1 - u$, the above equation becomes

$$S = 1 - e^{-cS}$$

Giant Component

The transition between the two regimes corresponds to the middle curve in the figure in the next slide and falls at the point where the gradient of the curve and the gradient of the dashed line match at $S = 0$. That is, the transition takes place when

$$\frac{d}{dS}(1 - e^{-cS}) = 1$$

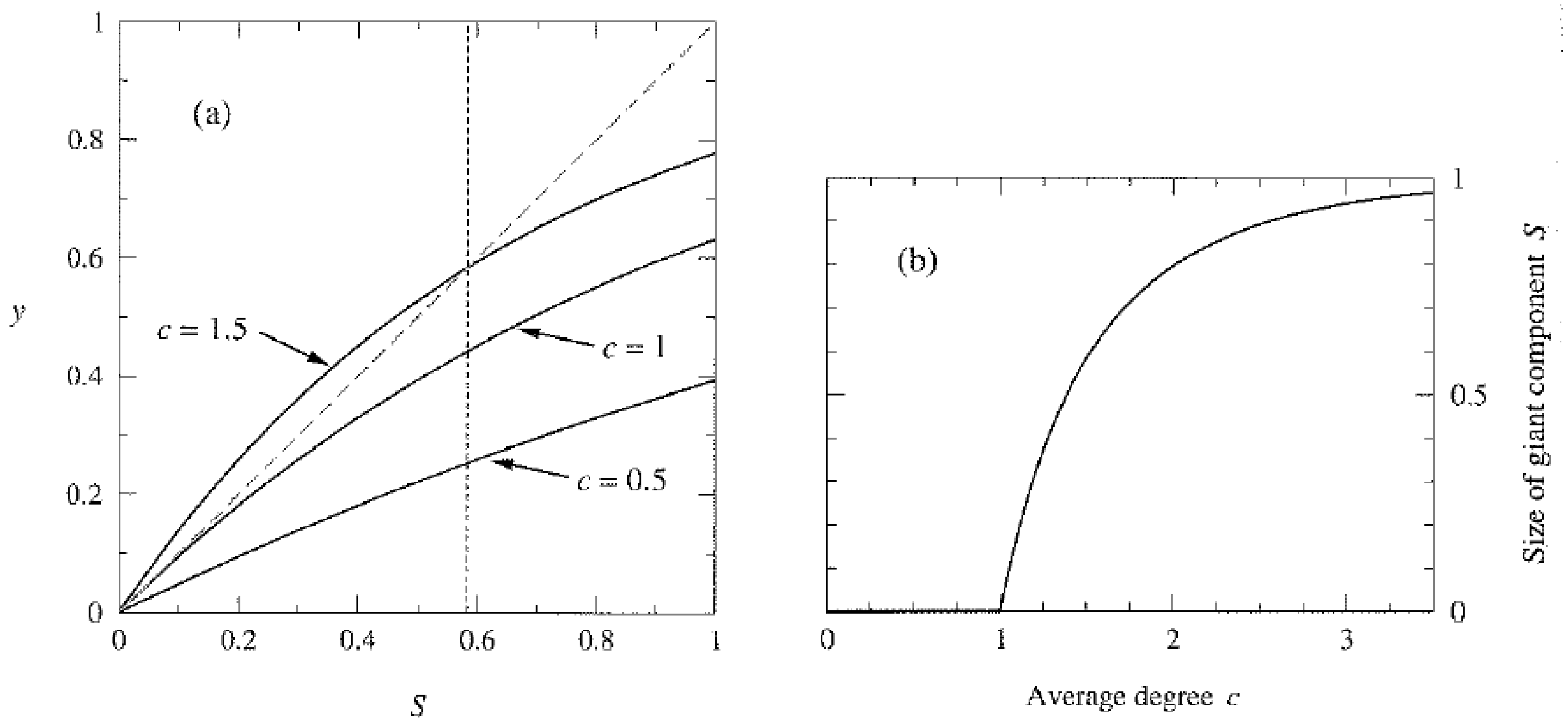
or,

$$ce^{-cS} = 1.$$

Setting $S = 0$, => $c = 1$.

Random Networks

Giant Component



Graphical solution for the size of the giant component. (a) The three curves in the left panel show $y = 1 - e^{-cS}$ for values of c as marked, the diagonal dashed line shows $y = S$, and the intersection gives the solution to the equation, $S = 1 - e^{-cS}$. For the bottom curve there is only one intersection, at $S = 0$, so there is no giant component, while for the top curve there is a solution at $S = 0.583$ ... (vertical dashed line). The middle curve is precisely at the threshold between the regime where a non-trivial solution for S exists and the regime where there is only the trivial solution $S = 0$. (b) The resulting solution for the size of the giant component as a function of c.

Random Networks

Real-World Networks / Simulated Random Graphs

	Original Network				Simulated Random Graph	
Network	*Size*	*Average Degree*	*Average Path Length*	C	*Average Path Length*	C
Film Actors	225,226	61	3.65	0.79	2.99	0.00027
Medline Coauthorship	1,520,251	18.1	4.6	0.56	4.91	1.8×10^{-4}
E.Coli	282	7.35	2.9	0.32	3.04	0.026
C.Elegans	282	14	2.65	0.28	2.25	0.05

$$\langle l \rangle \propto \ln N$$

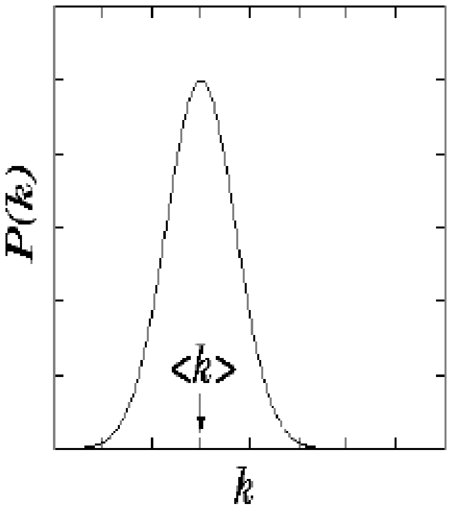
	Shortest path	Clustering
Random networks	Short	Low
Real networks	Short	High
Regular-topology networks *	Long	High *

* [Watts & Strogatz 1998]

Random Networks

Random vs Real-World Networks

Degree distributions



Poison distribution

$$P(k) = \frac{\langle k \rangle^k}{k!} e^{-\langle k \rangle}$$

[Barabási & Albert, 1999]

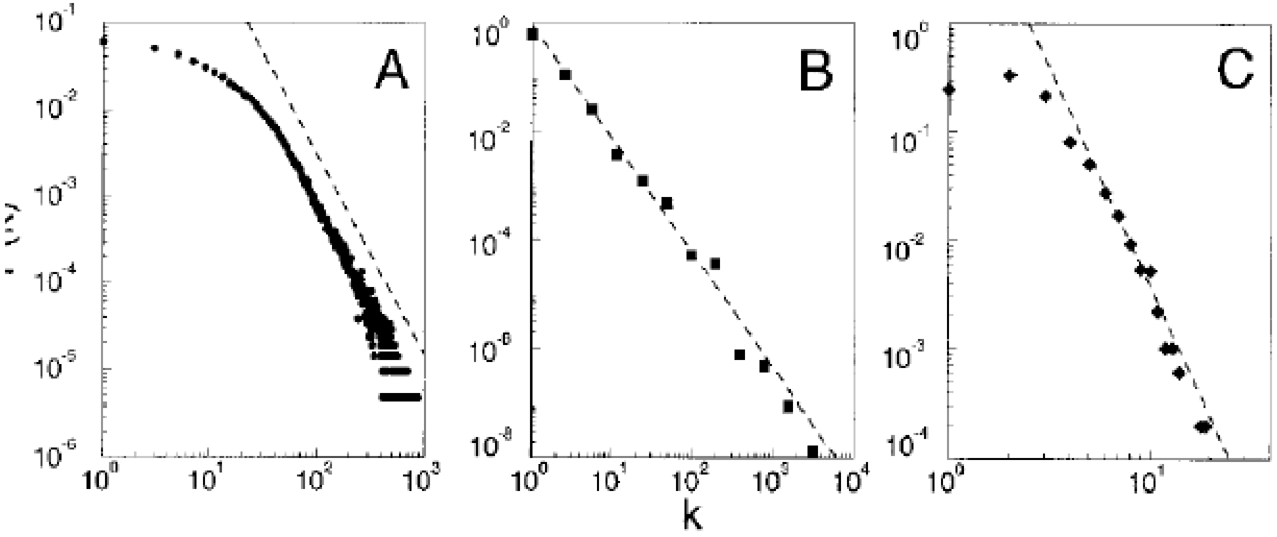


Fig. 1. The distribution function of connectivities for various large networks. (**A**) Actor collaboration graph with $N = 212{,}250$ vertices and average connectivity $\langle k \rangle = 28.78$. (**B**) WWW, $N = 325{,}729$, $\langle k \rangle = 5.46$ (6). (**C**) Power grid data, $N = 4941$, $\langle k \rangle = 2.67$. The dashed lines have slopes (A) $\gamma_{actor} = 2.3$, (B) $\gamma_{www} = 2.1$ and (C) $\gamma_{power} = \mathbf{4}$.

Heavy tail distributions
(often power law in log axes) $P(k) \propto k^{-\gamma}$

Random Networks

Summarize:

- Degree distribution: Poisson (degrees of all nodes close to average)

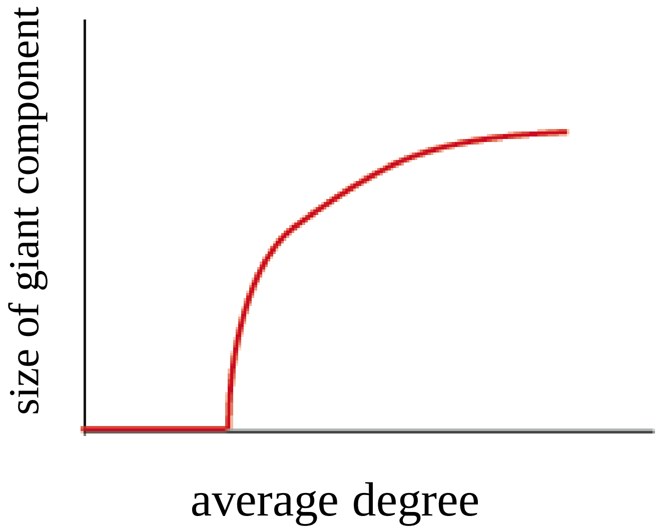
- No correlations, all edges exist independently of each other

- Path lengths grow logarithmically with system size n, $<l> \sim \ln(n)$

- Connectivity depends on average degree $<k>$
  - small $<k>$ => several disjoint components
  - high $<k>$ => giant connected component
  - there is a percolation transition phase
  - (from a fragmented to a connected)

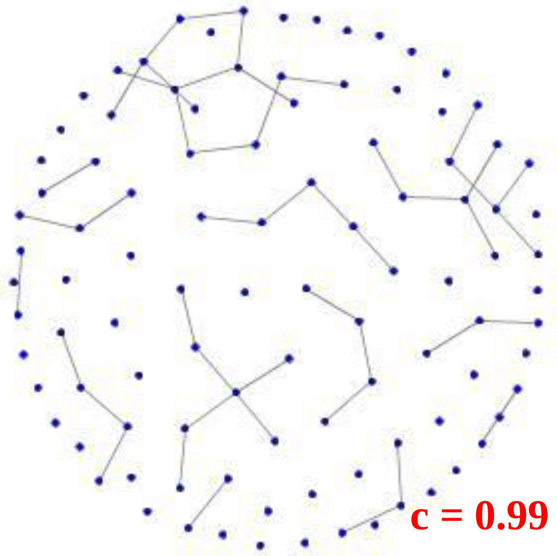
- Very “homogeneous” networks

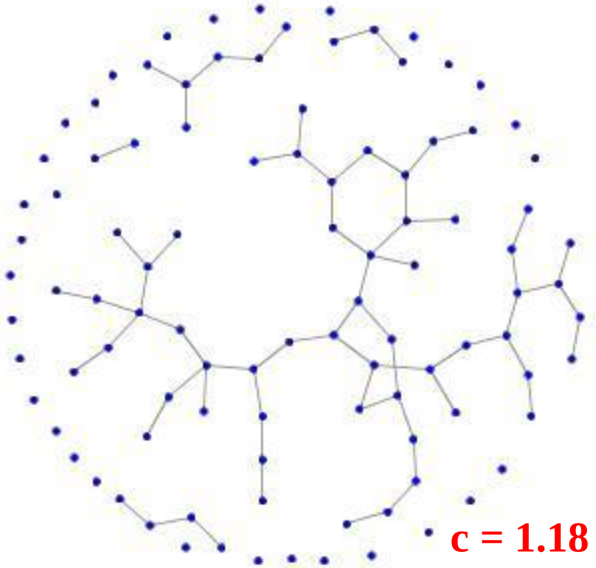
Percolation threshold in Erdos-Renyi Graphs

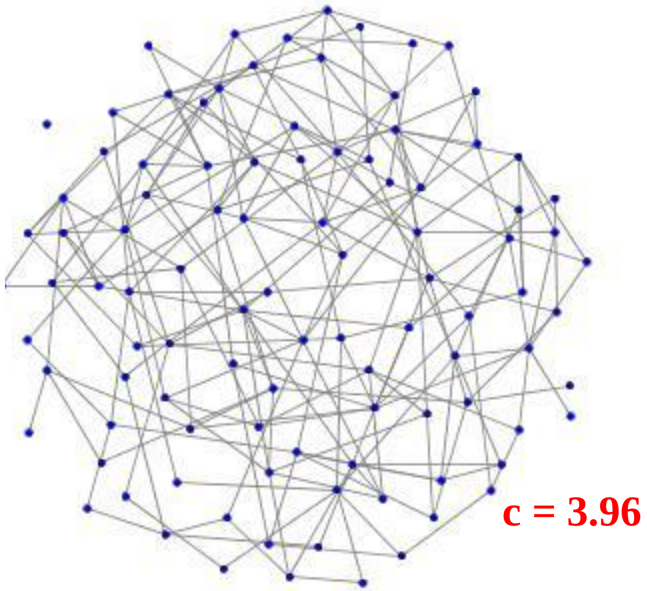


Percolation threshold: how many edges need to be added before the giant component appears?

As the average degree increases to $c = 1$, a giant component suddenly appears







Phase Transition in Erdos-Renyi Graphs

- *Phase Transition:* the point where diameter value starts to shrink in a random graph

- The phase transition happens when the average node degree $c = 1$ (or $p = 1/(n\text{-}1)$)

- At the Transition point:
  1. The giant component, just started to appear and grow
  2. The diameter just reached its maximum value and starts to decrease

Phase Transition in Erdos-Renyi Graphs

One can now distinguish the *ER* graphs into four topologically distinct regimes as one varies $\langle k \rangle$.

- Subcritical Regime: $0 < \langle k \rangle < 1$, no giant component
- Critical Point: $\langle k \rangle = 1$, giant component begins to form. It separates the regime where there is no giant component from the regime where there is one.
- Supercritical Regime: $\langle k \rangle > 1$. This regime is most relevant to real systems. The giant component continues to grow as $\langle k \rangle$ increases. Just above and close to the critical point, the giant component N_G has a size

$$\frac{N_G}{N} \sim \langle k \rangle - 1 \, .$$

- Connected Regime: $\langle k \rangle > \ln N$. This is the regime where all nodes in the network are connected as one giant component.

Generalized Random Model

The Generating Function

Suppose we have a probability distribution for a non-negative integer variable, e.g., a distribution of the degrees of randomly chosen vertices in a network. If the fraction of vertices in a network with degree k is p_k then p_k is also the probability that a randomly chosen vertex from the network will have degree k.

The *generating function* for the probability distribution p_k is the polynomial

$$g(z) = p_0 + p_1 z + p_2 z^2 + p_3 z^3 + \cdots = \sum_{k=0}^{\infty} p_k z^k \quad ; \quad p_k = \frac{1}{k!} \frac{d^k g}{dz^k} \Big|_{z=0}$$

The generating function gives us complete information about the probability distribution and vice versa. The distribution and the generating function are really just two different representations of the same thing. The moments of the distribution are given by

$$\langle k^n \rangle = \left(z \frac{d}{dz} \right)^n g \Big|_{z=1}$$

Generalized Random Model

The Generating Function

Examples:

Example 1:
Suppose our variable k takes only the values 0, 1, 2, and 3, with probabilities respectively, and no other values, and vertices of degree 0, 1, 2, and 3 occupied 40%, 30%, 20%, and 10% of the network respectively then the *generating function* for the probability distribution is

$$g(z) = 0.4 + 0.3z + 0.2z^2 + 0.1z^3 \,.$$

Example 2:
Suppose k follows a Poisson distribution with mean c

$$p_k = e^{-c}\frac{c^k}{k!}$$

The corresponding *generating function* would be

$$g(z) = e^{-c}\sum_{k=0}^{\infty}\frac{(cz)^k}{k!} = e^{c(z-1)}$$

Generalized Random Model

The Generating Function

Examples:

<u>*Example 3:*</u>
Suppose *k* follows an exponential distribution of the form

$$p_k = Ce^{-\lambda k}$$

with λ > 0. The normalization constant is $\sum_k p_k = 1$, which gives $C = 1 - e^{-\lambda}$. Then

$$p_k = (1 - e^{-\lambda})e^{-\lambda k} \ .$$

The corresponding *generating function* would be

$$g(z) = (1 - e^{-\lambda}) \sum_{k=0}^{\infty} (e^{-\lambda} z)^k = \frac{e^{\lambda} - 1}{e^{\lambda} - z}$$

as long as $z < e^{\lambda}$.

Generalized Random Model

The Generating Function

Examples:

<u>*Example 4:*</u> *Power-Law distribution*

Suppose k now follows a power-law distribution of the form

$$p_k = Ck^{-\alpha}; \quad \alpha > 0$$

This expression would diverge when $k = 0$, so commonly one stops at $k = 1$. The normalization constant C can then be calculated from the condition that , $\sum_k p_k = 1$, which gives

$$C \sum_{k=1}^{\infty} k^{-\alpha} = 1$$

The sum cannot be performed in closed form, but is common enough to be called the ***Riemann zeta function***, denoted by $\zeta(\alpha)$,

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

Generalized Random Model

The Generating Function

Examples:

Example 4: Power-Law distribution (cont'd)
Thus, we can write $C = 1/\zeta(\alpha)$ and

$$\begin{cases} 0 & \text{for } k = 0, \\ \dfrac{k^{-\alpha}}{\zeta(\alpha)} & \text{for } k \geq 1. \end{cases}$$

The generating function now takes the form

$$g(z) = \frac{1}{\zeta(\alpha)} \sum_{k=1}^{\infty} k^{-\alpha} z^k$$

The sum cannot be expressed in closed form, but again has a name called the ***polylogarithm*** of *z* and denoted $\mathrm{Li}_\alpha(z)$,

$$\mathrm{Li}_\alpha(z) = \sum_{k=1}^{\infty} k^{-\alpha} z^k \quad \longrightarrow \quad g(z) = \frac{\mathrm{Li}_\alpha(z)}{\zeta(\alpha)}$$

Generalized Random Model

The Generating Function

Examples:

<u>*Example 4:*</u> *Power-Law distribution (cont'd)*

Properties of the *polylogarithm* and *Zeta* functions are known that we can carry out useful manipulations of the generating function. For example, the derivative of a *Zeta* function gives

$$\frac{d\mathrm{Li}_\alpha(z)}{\partial z} = \frac{\partial}{\partial z}\sum_{k=1}^{\infty} k^{-\alpha}\, z^k = \sum_{k=1}^{\infty} k^{-(\alpha-1)}\, z^{k-1} = \frac{\mathrm{Li}_{\alpha-1}(z)}{z}$$

In real-world networks, the degree distribution does not usually follow a power law over its whole range, but instead, it typically obeys a power law reasonably closely for values of *k* above some minimum value k_{min} but below that point it has some other behavior. In this case the generating function will take the form

$$g(z) = Q_{k_{min}-1}(z) + C\sum_{k=k_{min}}^{\infty} k^{-\alpha}\, z^k; \qquad Q_n(z) = \sum_{k=0}^{n} p_k z^k$$

The sum in the above equation also has its own name: the ***Lerch transcendent***.

Generalized Random Model

The Configuration Model

The *Configuration Model* is a model of a random graph with a given degree *sequence* $\{k_i\} = (k_1, k_2, k_3, \ldots, k_n)$ where k_i is the degree of node *i*=1, 2, …, *n*, rather than degree distribution. That is, the exact degree of each individual vertex in the network is fixed, rather than merely the probability distribution from which those degrees are chosen. This in turn fixes the number of edges *m* in the network, where $m = \frac{1}{2}\sum_i k_i$. Therefore, this model is in some ways analogous to *G (n, m)*, which also fixes the number of edges.

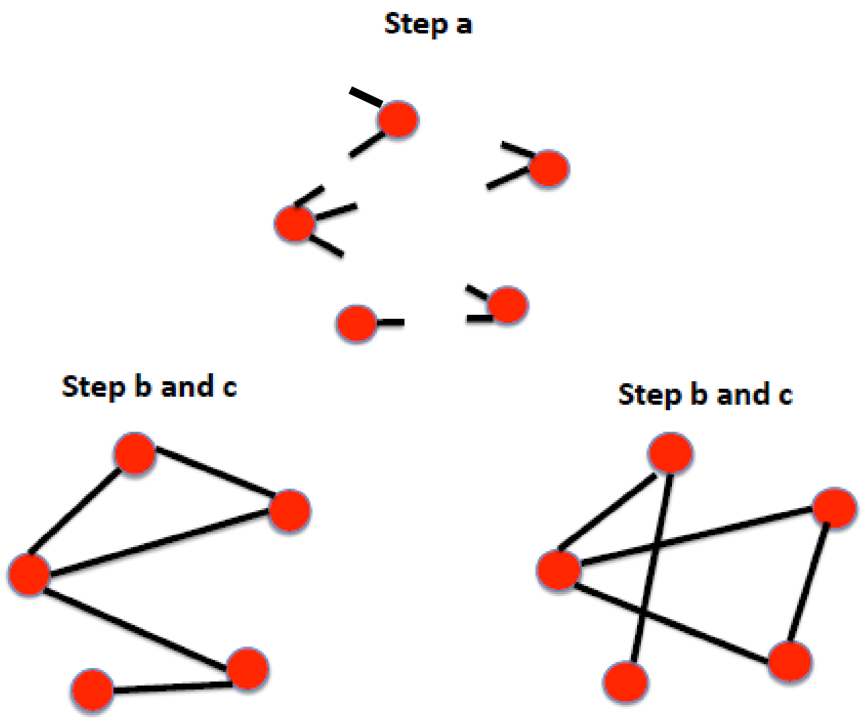
A network in the configuration model can be generated by the following recursive procedure: Given a degree sequence $\{k_i\} = (k_1, k_2, k_3, \ldots, k_n)$ with an ***even*** $\sum_i k_i$

a) We place k_i stubs on each node *i* of the network.
b) We match each stub of the network with another stub of the network.
c) We repeat step b) until all the stubs of the network are matched.

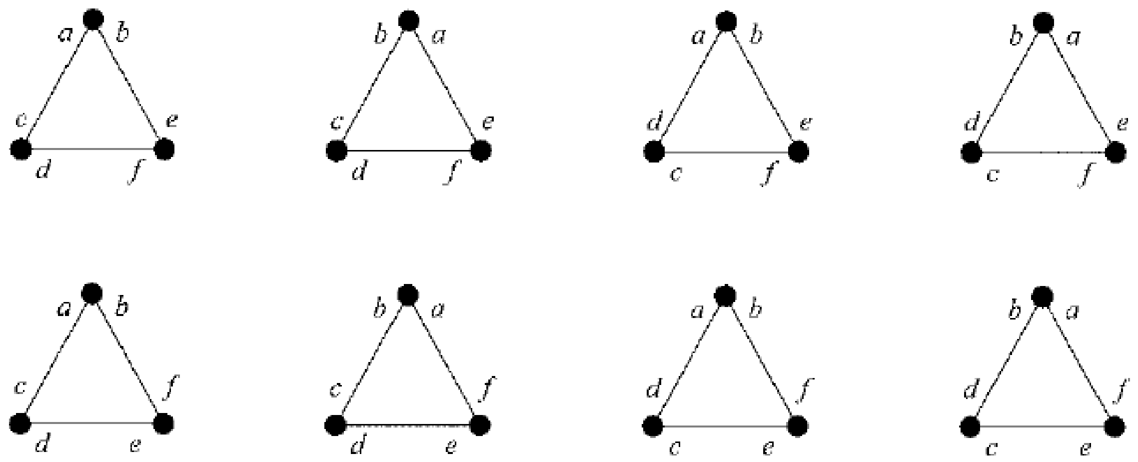
***** *There could be self-edges or multiedges. One way to avoid this is to repeat steps b) and c).***

Generalized Random Model

The Configuration Model



Construction of a network in the Configuration Model. In step a) the k_i stubs are placed on each node i of the network. In step b) and c) all the stubs are repeatedly matched until the full network is formed. In the above figure we show two possible networks generated by the configuration model starting from the same degree distribution. Starting from the degree distribution in this figure, one can construct 6 different simple networks.



Eight stub matchings that all give the same network. This small network is composed of three vertices of degree two and hence having two stubs each. The stubs are lettered to identify them and there are two distinct permutations of the stubs at each vertex for a total of eight permutations overall. Each permutation gives rise to a different matching of stub to stub but all matchings correspond to the same topological configuration of edges, and hence there are eight ways in which this particular configuration can be generated by the stub matching process.

Generalized Random Model

The Configuration Model

Edge Probability in the Configuration Model

Consider any one of the stubs that emerges from node *i*. What is the probability that this stub is connected by an edge to any of the stubs of node *j* ?

There are 2*m* - 1 stubs, excluding the stub from *i*, exactly k_j of them are attached to node *j*. Therefore the probability that our particular stub is connected to any of those around node *j* is $k_j/(2m-1)$. The total probability of a connection between *i* and *j* is

$$p_{ij} = \frac{k_i k_j}{(2m-1)} \sim \frac{k_i k_j}{2m}$$

Generalized Random Model

The Configuration Model

Edge Probability in the Configuration Model

One can use the above result, for example, to calculate the probability of having two edges between the same pair of vertices. The probability of having a second edge is given by the above equation but with k_i and k_j each reduced by one: $(k_i - \mathrm{l})(k_j - 1)/2m$. Therefore, the probability of having at least two edges is given by $k_i k_j (k_i - 1)(k_j - 1)/(2m)^2$. Summing this probability over all vertices and dividing by two (to avoid double counting of vertex pairs), we find that the expected total number of multiedges in the network is

$$\frac{1}{2(2m)^2} \sum_{ij} k_i k_j (k_i - 1)(k_j - 1) = \frac{1}{2\langle k \rangle^2 n^2} \sum_i k_i (k_i - 1) \sum_j k_j (k_j - 1) = \frac{1}{2} \left[\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \right]^2$$

where

$$\langle k \rangle = \frac{1}{n} \sum_i k_i \quad ; \quad \langle k^2 \rangle = \frac{1}{n} \sum_i k_i^2 \quad ; \quad 2m = \langle k \rangle n$$

Thus the expected number of multiedges remains constant as the network grows larger, so long as $\langle k^2 \rangle$ is constant and finite, and the density of multiedges-the number per vertex vanishes as 1/ n.

Generalized Random Model

The Configuration Model

Edge Probability in the Configuration Model

We can similarly write down the probability of a vertex *i* having a self-edge (self-loop)

$$p_{ii} = \frac{k_i(k_i - 1)}{4m}$$

Summing over all vertices *i*

$$\sum_i p_{ii} = \sum_i \frac{k_i(k_i - 1)}{4m} = \frac{\langle k^2 \rangle - \langle k \rangle}{2\langle k \rangle}$$

This expression remains constant as $n \to \infty$ provided $\langle k^2 \rangle$ remains constant, and hence, as with the multiedges, the density of self-edges in the network vanishes as 1/ n in the limit of large network size. The expected number of common neighbors of *i* and *j* is given by

$$n_{ij} = \sum_l \frac{k_i k_l}{2m} \frac{k_j(k_l - 1)}{2m} = \frac{k_i k_j}{2m} \frac{\sum_l k_l(k_l - 1)}{n\langle k \rangle} = p_{ij} \frac{\langle k^2 \rangle - \langle k \rangle}{2\langle k \rangle}$$

Thus the probability of sharing a common neighbor is equal to the probability p_{ij} of having a direct connection times a multiplicative factor that depends only on the mean and variance of the degree distribution.

Generalized Random Model

The Configuration Model

Clustering Coefficient for the Configuration Model

Consider a vertex *v* that has at least two neighbors, which we will denote *i* and *j*. Being neighbors of *v*, *i* and *j* are both at the ends of edges from v, and hence the number of other edges connected to them, k_i and k_j are distributed according to the excess degree distribution. Recall from the above, the probability of an edge between *i* and *j* is then $k_i k_j / 2m$. The clustering coefficient is then given by

$$C = \sum_{k_i k_j = 0}^{\infty} q_{k_i} q_{k_j} \frac{k_i k_j}{2m} = \frac{1}{2m} \left[\sum_{k=0}^{\infty} k q_k \right]^2$$

where $q_k = \frac{(k+1)p_{k+1}}{\langle k \rangle}$ represents the probability of a vertex that connects to vertex *v* having *k* edge ends left to connect to other vertices. Thus

$$C = \frac{1}{2m\langle k \rangle^2} \left[\sum_{k=0}^{\infty} k(k+1)p_{k+1} \right]^2 = \frac{1}{2m\langle k \rangle^2} \left[\sum_{k=1}^{\infty} (k-1)kp_k \right]^2 = \frac{1}{n} \frac{[\langle k^2 \rangle - \langle k \rangle]^2}{\langle k \rangle^3}$$

C goes as 1/*n* for fixed degree distribution and vanishes in the limit of large system size.