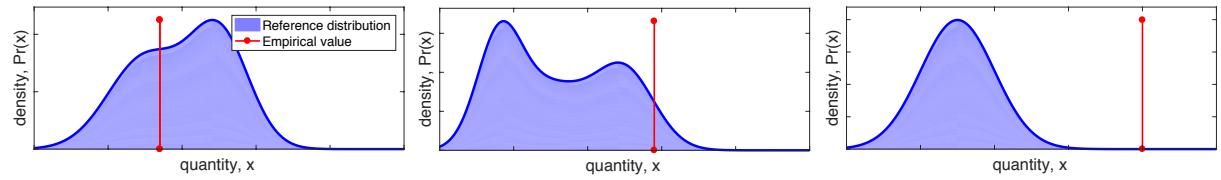
1 Random graph models

Suppose that we've calculated one or more descriptive statistics for a network G. For example, we've calculated the mean degree $\langle k \rangle$, or the degree distribution $\Pr(k)$, or the (global) clustering coefficient C, or the mean geodesic distance $\langle \ell \rangle$. How do we interpret the values these statistics take? Which values are "big" or "small," "typical" or "unusual"?

Random graph models are one way to answer these questions: they provide a reference point x or a whole reference distribution $\Pr(x)$ for assessing an empirical measurement x.[1]

For example, let's say we obtain a clustering coefficient of C_* for a gene regulatory network G. Using a random graph model M that is parameterized to be *similar* to G in some way, e.g., similar edge density, similar degree distribution, etc., we can use M to produce many "synthetic" (computer generated) networks, and calculate each of their clustering coefficients. These synthetic values represent a *reference distribution* $\Pr(C)$, which encodes how often M will produce networks with different values of C. If the empirical value C_* falls in the "middle" of $\Pr(C)$ (left-hand panel below), then we may conclude that the assumptions that underlie M are sufficient to explain the value of C_* in G. Or, if C_* tends to be on the upper end of the reference distribution, we may conclude that C_* is somewhat unusual but not unrealistic (middle panel). Or, if C_* falls far outside $\Pr(C)$, we may conclude that the assumptions of M are insufficient to explain the value of C_* (right-hand panel).



In these lecture notes, we will learn about three classes of random graph models, which represent a sequence of increasing structure imposed on the basic property of randomness.[2]

1. **Erdős-Rényi (ER) random graph**:
   the simplest random graph, in which edges are independent and identically distributed with probability p, which makes them very homogeneous.

[1] As mathematical models, random graphs are also extremely useful for analyzing the structural properties of networks in ways that are impractical for real-world networks, whose structure is messier and more complicated. Random graph models provide a way to build our intuition about how certain descriptive statistics should behave, under different assumptions about the structural randomness.

[2] A fourth class of random graph, which we will not discuss in detail, are spatial random graphs, in which the probability that an edge (i, j) exists depends on the pairwise distance $d(i, j)$ of the two nodes in some latent or explicit space, e.g., a shorter distance correlates with a higher connection probability.

2. **Configuration model** (and the Chung-Lu model):
   in which the random graph is conditioned on having a specified degree sequence $\vec{k}$ or a specified degree distribution $\Pr(k)$, which makes them more heterogeneous.

3. **Modular random graphs**:
   in which the random graph is conditioned on having a specified modular structure θ, and possibly also a specified degree sequence $\vec{k}$, etc.

1.1 Aside: random graph models vs. mechanistic models

The many models of network structure can largely be divided into two conceptual classes: *random graph* models and *mechanistic* models. Although we will treat them as being distinct, the boundaries between these classes are not sharp. The value of these conceptual classes comes mainly from highlighting their different purposes.

A mechanistic model is one that codifies or formalizes a notion of causality via a set of rules (often mathematical) that are applied repeatedly to produce a network. Identifying the underlying mechanism that explains ("causes," in this case) the existence of an empirically observed pattern allows us to better understand and predict networks — if we see that pattern again, we can immediately generate hypotheses about what might have produced it. In network models, the mechanisms are often very simple, which makes them more amenable to mathematical analysis, and a well-defined mechanistic model produces specific kinds of topological patterns in the networks it generates. One of the most well-known network mechanisms is that the model of *preferential attachment*, in which new nodes arrive in the network in some order, and each new node j attaches to an existing node i with probability proportional to i's current degree k_i. A key signature of preferential attachment is that it produces power-law degree distributions, although it is not the only mechanism that does so, and some versions of preferential attachment do not produce power laws.[3] Mechanistic models are commonly found in hypothesis-driven network analysis and modeling, where the goal is specifically focused on cause and effect.

Random graph models, on the other hand, typically represent weaker notions of causality and generate structure via a set of free parameters that may or may not have specific meanings. The simplest and best known random graph model is the Erdős-Rényi random graph, after two of its most famous investigators (sometimes also called a Poisson or Binomial random graph). In this and other, more structured random graph models, edges are random variables whose probability may depend on other variables. In the Erdős-Rényi random graph model, every edge is an iid random variable with a fixed probability p, which determines everything about the network.

[3] Hence, power-law degree distributions are not diagnostic of preferential attachment.

A key attraction of random graph models is that many questions about their structure, e.g., the network measures we have encountered so far, may be calculated analytically or numerically, after parameterizing the model to be "close" to a network of interest. Hence, random graphs can provide a reference point or reference distribution for assessing empirical measurements, as described above. For instance, let G denote a graph and let $\Pr(G)$ be a probability distribution over all such graphs that comes from our selected random graph model. The typical or expected value of some network measure is then given by

$$\langle x \rangle = \sum_G x(G) \times \Pr(G) \ ,$$

where $x(G)$ is the value of the measure x on a particular graph G. This equation has the usual form of an average, but is calculated by summing over the combinatoric space of graphs.[4] This quantity then gives us the ability to make the comparisons described above, to ask whether an observed value $\langle x_{\text{data}} \rangle$ is close to or far from the value expected from the model $\langle x_{\text{model}} \rangle$. In the former case, we can conclude that the model that produces $\Pr(G)$ is a sufficient explanation of $\langle x_{\text{data}} \rangle$, and in the latter, we can conclude that the assumptions of the model are insufficient, i.e., there is something else going on. This approach to analyzing empirical patterns from the perspective of random graph models treats the random graph as a *null model*, which is a classic approach in the statistical sciences.

In this lecture, we will study the simple random graph and derive several of its most important properties.

2 The Erdős-Rényi random graph

The **Erdős-Rényi random graph model**, sometimes called an ER graph or simply $G(n, p)$, is the oldest and simplest random graph model. It was most prominently studied by mathematicians Paul Erdős (1913–1996)[5] and Alfréd Rényi (1921–1970)[6] (although it was, in fact, studied earlier).

An ER graph, denoted $G(n, p)$, is a simple graph with n vertices, where p is the probability that each simple edge exists, $(i, j) \in E$.[7] Hence, edges are independent and identically distributed (iid) random variables, and the size of the network and the edge probability fully specify the model.

[4] We may also be interested in the full distribution of x, although this can be trickier to calculate.

[5] `http://xkcd.com/599/`

[6] "A mathematician is a machine for turning coffee into theorems."

[7] A closely related random graph model is $G(n, m)$, which places exactly m edges on n vertices, meaning that m is no longer a random variable as it is under $G(n, p)$.

In an adjacency matrix representation, we say

$$\forall_{i>j} \qquad A_{ij} = A_{ji} = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$

The restriction $i > j$ appears because edges are undirected (or, the adjacency matrix is symmetric across the diagonal) and we prohibit self-loops. Furthermore, because each pair is either connected or not, this model is not a multi-graph model. That is, this is a model of a simple random graph.

Mathematically, $G(n, p)$ defines an *ensemble*, or a parametric distribution over all graphs $\Pr(G \,|\, p)$. Choosing p parameterizes this distribution, and we draw any particular graph G with probability $\Pr(G \,|\, p)$. This fact means that when we describe the properties of $G(n, p)$ below, we are typically describing *average* properties, and individual instances will tend to fluctuate around these values.[8] The extremal values of p produce extreme graphs: $p = 1$ produces a fully connected (dense) graph, or a clique, because all $\binom{n}{2}$ possible edges exist; in contrast, $p = 0$ produces an "empty" graph, because no edges exist. But, most real-world networks are what we call "sparse," meaning they have a mean degree $\langle k \rangle$ that is very small compared to n. For $G(n, p)$ to produce a sparse graph, we need only set $p = c/(n-1)$, for a desired mean degree $\langle k \rangle = c$, e.g., $c = 2$ or $c = 4$.

2.1 Properties of ER graphs

ER random graphs are not considered realistic models of networks, but they are a useful baseline random graph model. Nevertheless, for a few descriptive statistics, randomness and low density are sufficient to produce interesting structure. At a high level, ER graphs have the following properties:

- Traditionally a *simple graph*, but also easily generalized to directed edges, with or without self-loops (do you see how?).
- Connectivity is "homogeneous," and is entirely driven by the edge "density" parameter p; edges are independent and identically distributed (iid).
- The degree distribution $\Pr(k)$ is a Poisson distribution with mean $c = p(n-1)$, which is a low-variance distribution.
- The diameter and mean geodesic distance are $O(\log n)$, making ER graphs "small-world-like," so that short paths exist between most pairs of nodes.
- The clustering coefficient is $C = O(1/n)$, meaning there are very few triangles.
- The largest connected component (LCC) is proportional to the network size $O(n)$ when $c > 1$, and is vanishingly small, containing $O(1)$ nodes, when $c < 1$ (a phase transition at $c = 1$).

[8] In fact, a counter-intuitive thing about $G(n, p)$ is that so long as $0 < p < 1$, there is a non-zero probability of generating *any* graph of size n. When faced with some particular graph G, how can we then say whether or not $G \in G(n, p)$? This question is philosophically tricky in the same way that deciding whether or not some particular binary sequence, say a binary representation of all of Shakespeare's works, is "random," i.e., drawn uniformly from the set of all binary sequences of the same length.

2.2 Generating a $G(n,p)$ network

Once n and p have been chosen, the most straightforward way to draw a graph from the ensemble is to loop over pairs of nodes:

1. initialize an empty graph G with n nodes
2. for each of the $\binom{n}{2}$ pairs i,j, draw a uniformly random number r
3. if $r \leq p$, then add the undirected edge (i,j) to G.

Basically, this procedure flips $\binom{n}{2}$ coins, each with bias p. As a result, generating a network in this way takes $\Theta(n^2)$ time, and is computationally expensive when $n > 10^4$ or so.[9]

2.3 Mean degree

In the $G(n,p)$ model, every edge exists independently and with the same probability. (Technically speaking, these random variables are independent and identically distributed, or iid.) The total probability of drawing a graph with m edges from this ensemble is

$$\Pr(m) = \binom{\binom{n}{2}}{m} p^m (1-p)^{\binom{n}{2}-m} \ , \tag{1}$$

which is a binomial distribution choosing m edges out of the $\binom{n}{2}$ possible edges. (Note that this form implies that $G(n,p)$ is an undirected graph.) The mean value can be derived using the Binomial Theorem:

$$\begin{aligned} \langle m \rangle &= \sum_{m=0}^{\binom{n}{2}} m \Pr(m) \\ &= \binom{n}{2} p \ . \end{aligned} \tag{2}$$

That is, the mean degree is the expected number of the $\binom{n}{2}$ possible ties that exist, given that each edge exists with probability p.

Recall from Lecture 1 that for any network with m edges, the mean degree of a vertex is $\langle k \rangle = 2m/n$.

[9] It's possible to draw from $G(n,p)$ in only $O(n+m)$ time, using an algorithm that is only slightly more complicated than this one. See Batagelj and Brandes, *Phys. Rev. E* **71**, 036113 (2005) `http://vlado.fmf.uni-lj.si/pub/networks/doc/ms/rndgen.pdf`.

Thus, the mean degree in $G(n,p)$ may be derived, using Eq. (2), as

$$\begin{aligned}\langle k\rangle &= \sum_{m=0}^{\binom{n}{2}} \frac{2m}{n}\Pr(m) \\ &= \frac{2}{n}\binom{n}{2}p \\ &= (n-1)p \ . \end{aligned} \tag{3}$$

In other words, each vertex has $n-1$ possible partners,[10] and each of these exists with the same independent probability p. The product, by linearity of expectations, gives the mean degree, which is sometimes denoted c.

2.4 Degree distribution

Because edges in $G(n,p)$ are iid random variables, the entire degree distribution has a simple form: it's a Binomial distribution. For a node to have degree k, when we flip the $n-1$ coins that determine which of the other $n-1$ nodes it connects to, we have to get exactly k successes and $n-1-k$ failures. And, there are $\binom{n-1}{k}$ ways to get those k successes. Hence, the degree distribution is

$$\Pr(k) = \binom{n-1}{k}p^k(1-p)^{n-1-k} \ , \tag{4}$$

with parameter p for $n-1$ independent "Bernoulli" trials (coin flips). What value of p should we choose? Commonly, we set $p = c/(n-1)$, where c is the target mean degree and is a finite value. (Verify using Eq. (3) that the expected value is indeed c under this choice for p.) That is, we choose the regime of $G(n,p)$ that produces *sparse networks*, where $c = O(1)$, which implies $p = O(1/n)$.

When p is very small, the binomial distribution may be simplified. When p is small, the last term in Eq. (4) may be approximated as

$$\begin{aligned}\ln\Big[(1-p)^{n-1-k}\Big] &= (n-1-k)\ln\left(1-\frac{c}{n-1}\right) \\ &\simeq (n-1-k)\frac{-c}{n-1} \\ &\simeq -c \ , \end{aligned} \tag{5}$$

where we have used a first-order Taylor expansion of the logarithm[11] and taken the limit of large n. Taking the exponential of both sides yields the approximation $(1-p)^{n-1-k} \simeq \mathrm{e}^{-c}$, which is exact

[10] In many mathematical calculations, we approximate $n-1 \approx n$, implying that $\langle k\rangle \approx pn$. In the limit of large n this approximation is exact.

[11] A useful approximation: $\ln(1+x) \simeq x$, when x is small.

as $n \to \infty$. Thus, the expression for our degree distribution becomes

$$\Pr(k) \simeq \binom{n-1}{k} p^k \mathrm{e}^{-c} \ , \tag{6}$$

which may be simplified further still. The binomial coefficient is

$$\begin{aligned} \binom{n-1}{k} &= \frac{(n-1)!}{(n-1-k)!\,k!} \\ &\simeq \frac{(n-1)^k}{k!} \ . \end{aligned} \tag{7}$$

Thus, the degree distribution is, in the limit of large n

$$\begin{aligned} \Pr(k) &\simeq \frac{(n-1)^k}{k!} p^k \mathrm{e}^{-c} \\ &= \frac{(n-1)^k}{k!} \left(\frac{c}{n-1}\right)^k \mathrm{e}^{-c} \\ &= \frac{c^k}{k!} \mathrm{e}^{-c} \ , \end{aligned} \tag{8}$$

which is called the Poisson distribution. This distribution has mean and variance c, and is slightly asymmetric. The left panel of Figure 1 below shows examples of several Poisson distributions, all with $c \geq 1$. Recall, however, that most real-world networks exhibit heavy-tailed distributions. The degree distribution of the random graph model decays rapidly for $k > c$ and is thus highly unrealistic.

Implications. Recall from Lecture 2 that most real-world networks have heavy-tailed distributions. In contrast, a Poisson distribution is not heavy tailed—it falls off too quickly to produce nodes with degrees much higher than the mean. Hence, ER graphs are poor models of real-world networks.

2.5 Motifs, reciprocity, and the clustering coefficient

The fact that edges in an ER graph are iid makes it easy to calculate the probability of many motifs.

For instance, if we generalize $G(n,p)$ to a directed case, in which both (i,j) and (j,i) each occur with probability p, then the reciprocity coefficient is

$$r = \frac{\text{(number of reciprocated links)}}{\text{(number of links)}} = \frac{(n^2-n)p^2}{(n^2-n)p} = p \ , \tag{9}$$

where $n^2 - n$ is the number of possible directed edges (without self-loops), and in the numerator, we require that both (i,j) and (j,i) exist, which happens with probability p^2.

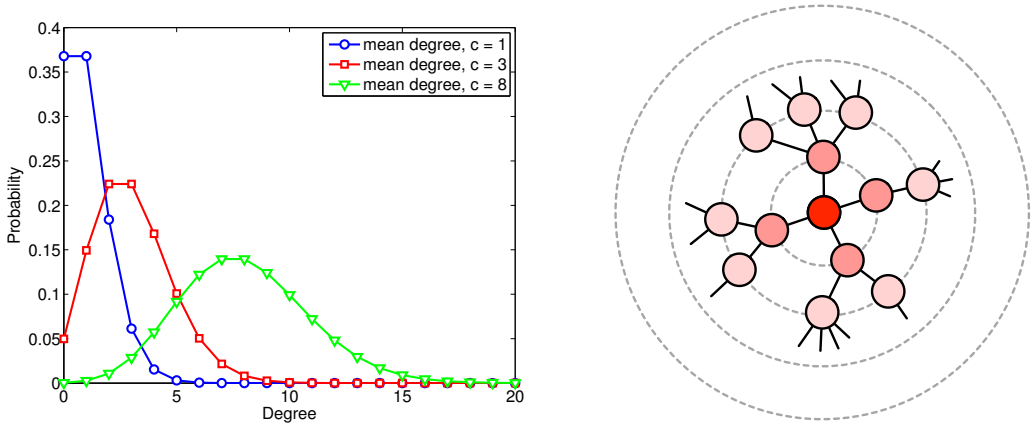


Figure 1: (left) Poisson degree distributions for a few choices of mean degree $c = \{1, 3, 8\}$, showing that as the mean degree increases, the distribution move away from the minimum value of $k = 0$, but remains tightly concentrated around the mean c. (right) A schematic illustrating a "locally tree-like" structure, in which for most nodes, very few or none of their neighbors are connected (few or no loops of short length).

The calculation for the clustering coefficient, on the undirected ER graph, follows a similar logic:

$$C = \frac{\text{(number of triangles)}}{\text{(number of connected triples)}} \propto \frac{\binom{n}{3}p^3}{\binom{n}{3}p^2} = p = \frac{c}{n-1} \quad . \tag{10}$$

In the sparse case, this further implies that $C = O(1/n)$, i.e., the density of triangles in the network decays toward zero in the limit of large graph.

This type of calculation can be generalized to loops of longer length or cliques of larger size and produces the same result: the density of such structures decays to zero in the large-n limit.

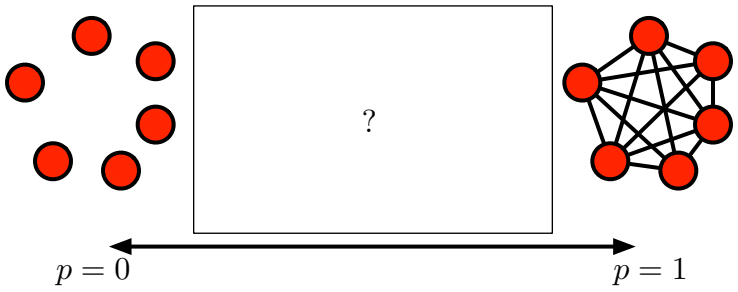
Implications. When a random graph G is sparse, both $r = O(1/n)$ and $C = O(1/n)$, meaning that the density of these motifs shrinks toward zero in the limit of large graphs. In other words, ER graphs have very few reciprocated links or triangles.[12] The latter fact has a crucial implication: sparse ER graphs are *locally tree-like*, meaning that if we build a tree outward from any particular node in the graph, we rarely encounter a "cross edge" that links between two branches of the tree. The right panel in Figure 1 illustrates this idea.

[12] In fact, Eq. (10) generalizes to a loop of $1 < \ell \leq n$ edges. Do you see how? As a result, ER graphs have few loops of any size.

2.6 Surprise 1: network connectedness phase transition

The first of two interesting properties of this random graph model is the sudden appearance, as we increase the mean degree c, of a *giant component*, i.e., a component whose size is proportional to the size of the network n. This sudden appearance is called a *phase transition*.[13]

We care about phase transitions because they embody *qualitative* changes in the fundamental behavior of the model system. They are inherently non-linear effects, in which a small change in some parameter leads to a big change in the system's behavior. As a result, they can be a good model of the underlying non-linear mechanisms of particular complex systems, which sometimes exhibit precisely this kind of sensitivity.



Consider the two limiting cases for the parameter p. If $p = 0$ we have a fully empty network with n completely disconnected vertices. Every component in this network has the same size, and that size is a $O(1/n)$ fraction of the size of the network. In the jargon of physics, the size of the largest component here is an *intensive* property, meaning that it is independent of the size of the network.

On the other hand, if $p = 1$, then every edge exists and the network is an n-clique. This single component has a size that is a $O(1)$ fraction of the size of the network. In the jargon of physics, the size of the largest component here is an extensive property, meaning that it depends on the size of the network.[14] Thus, as we vary p, the size of the largest component transforms from an intensive property to an extensive one, and this is the hallmark of a phase transition. Of course, it could be that the size of the largest component becomes extensive only in the limit $p \to 1$, but

[13]The term "phase transition" comes from the study of critical phenomena in physics. Classic examples include the melting of ice, the evaporation of water, the magnetization of a metal, etc. Generally, a phase transition characterizes a sudden and qualitative shift in the bulk properties or global statistical behavior of a system. In this case, the transition is discontinuous and characterizes the transition between a mostly disconnected and a mostly connected networked.

[14]Other examples of extensive properties in physics include mass, volume and entropy. Other examples of *intensive* properties—those that are independent of the size of the system—include the density, temperature, melting point, and pressure. See `https://en.wikipedia.org/wiki/Intensive_and_extensive_properties`

in fact, something much more interesting happens. (When a graph is sparse, what other network measures are intensive? What measures are extensive?)

2.6.1 The sudden appearance of a "giant" component

Let u denote the average fraction of vertices in $G(n,p)$ that do *not* belong to the giant component. Thus, if there is no giant component (e.g., $p=0$), then $u=1$, and if there is then $u<1$. In other words, let u be the probability that a vertex chosen uniformly at random does not belong to the giant component.

For a vertex i not to belong the giant component, it must not be connected to any other vertex that belongs to the giant component. This means that for every other vertex j in the network, either (i) i is not connected to j by an edge or (ii) i is connected to j, but j does not belong to the giant component. Because edges are iid, the former happens with probability $1-p$, the latter with probability pu, and the total probability that i does not belong to the giant component via vertex j is $1-p+pu$.

For i to be disconnected from the giant component, this must be true for all $n-1$ choices of j, and the total probability u that some i is not in the giant component is

$$\begin{aligned} u &= (1-p+pu)^{n-1} \\ &= \left[1-\frac{c}{n-1}(1-u)\right]^{n-1} \qquad (11) \\ &= \mathrm{e}^{-c(1-u)} \qquad (12) \end{aligned}$$

where we use the identity $p=c/(n-1)$ in the first step, and the identity $\lim_{n\to\infty}\left(1-\frac{x}{n}\right)^n = \mathrm{e}^{-x}$ in the second.[15]

If u is the probability that i is not in the giant component, then let $S=1-u$ be the probability that i belongs to the giant component. Plugging this expression into Eq. (12) and eliminating u in favor of S yields a single equation for the size of the giant component, expressed as a fraction of the total network size, as a function of the mean degree c:

$$S = 1-\mathrm{e}^{-cS} \quad . \qquad (13)$$

[15] We can sidestep using the second identity by taking the logarithms of both sides of Eq. (11):

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

where the approximate equality becomes exact in the limit of large n. Exponentiating both sides of our approximation then yields Eq. (12). This should look familiar.

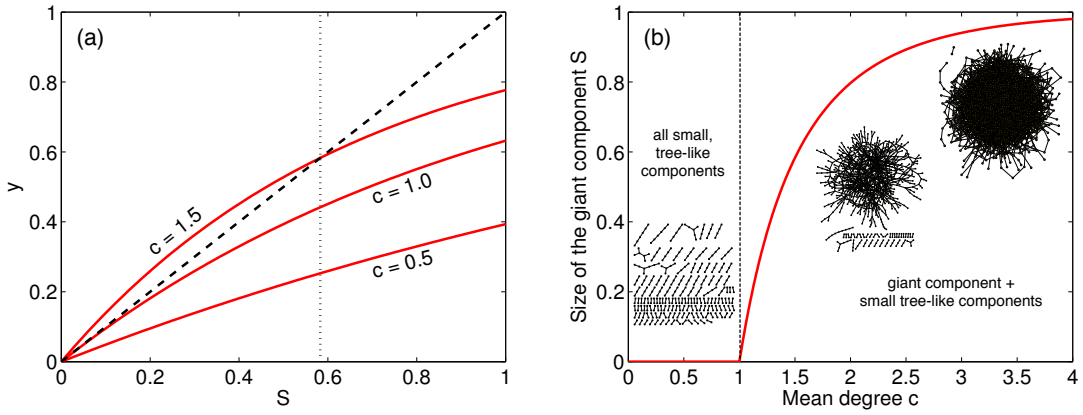


Figure 2: (a) Graphical solutions to Eq. (13), showing the curve $y = 1 - \mathrm{e}^{-cS}$ for three choices of c along with the curve $y = S$. The locations of their intersection gives the numerical solutions to Eq. (13). Any solution $S > 0$ implies a giant component. (b) The solution to Eq. (13) as a function of c, showing the discontinuous emergence of a giant component at the critical point $c = 1$, along with some examples random graphs from different points on the c axis.

Note that this equation is transcendental and there is no simple closed form that isolates S from the other variables.[16]

We can visualize the shape of this function by first plotting the function $y = 1 - \mathrm{e}^{-cS}$ for $S \in [0, 1]$ and asking where it intersects the line $y = S$. The location of the intersection is the solution to Eq. (13) and gives the size of the giant component. Figure 1 (next page) shows this exercise graphically (and Section 6 below contains the Matlab code that generates these figures). In the "sub-critical" regime $c < 1$, the curves only intersect at $S = 0$, implying that no giant component exists. In the "super-critical" regime $c > 1$, the lines always intersect at a second point $S > 0$, implying the existing of a giant component. The transition between these two "phases" happens at $c = 1$, which is called the "critical point".

2.7 Surprise 2: small world networks

The second of two interesting properties of ER random graphs is that, despite being as simple as possible, they are *small world* networks, in the sense of having a diameter that grows very, very slowly, like $O(\log n)$. Rather than derive this fact mathematically, we will go through the intuition, based on a branching process, that produces the right asymptotic behavior.

[16] For numerical calculations, it may be useful to express it as $S = 1 + (1/c)W(-c\mathrm{e}^{-c})$ where $W(.)$ is the *Lambert W-function* and is defined as the solution to the equation $W(z)\mathrm{e}^{W(z)} = z$.

2.7.1 Branching processes and percolation

Consider an empty graph $G = (V, \emptyset)$ and choose an arbitrary node i to begin. This approach uses the *principle of deferred decisions* to construct the ER graph a few edges at a time, which exploits the fact that edges are iid random variables, and hence we can defer until later in a process figuring out which edges exist and which do not.

For each of the $n-1$ possible neighbors of i, we flip a coin with bias p and place an edge (i, j) for each j that is a success. In expectation i has the average degree c neighbors. Now, note that for each of those neighbors j, we can repeat the process, but for the $n-2$ possible neighbors (since we've already flipped a coin for the adjacency i, j). This procedure of flipping coins to add edges to an otherwise empty graph is called a *branching process*, and is most commonly used to model the spread of infectious diseases and the spread of information along a network.

The key question is: how many steps ℓ can this branching process go on for before it stops? Notice that each time we take a step, we are reducing the number of possible new neighbors by 1. That is, initially, we have $n-1$ possible neighbors, then $n-2$, then $n-3$, etc. And, each time we take a step, we are lengthening the "depth" ℓ of the tree we are growing from node i. And, the tree must stop growing when all n nodes are, in expectation, in the tree.

The key insight is that the number of nodes in the tree grows like $O(c^\ell)$, where c is the expected degree and ℓ is the depth of the tree. Setting this function equal to n and solving yields $\ell = O(\log n)$. Finally, note that the diameter is 2ℓ, but we can drop that constant inside the asymptotic notation.

Hence, ER graphs diameters grow like $O(\log n)$, implying that the longest geodesic paths are *short*. This result is surprising, because a simple model like the ER graph model nevertheless can "explain" a pattern in real-world networks: they have very short path lengths between all nodes.

Phase transition redux: This branching process argument for the diameter works when $c > 1$, because then the size of the tree grows exponentially, like $O(c^\ell)$. This fact lets us see something else about the phase transition.

Recall that the giant component emerges only when $c > 1$. This fact, and the applicability of the branching process argument to calculate the diameter, are two sides of the same coin—only when $c > 1$ does the tree "percolate" and span the entire network, producing a component whose size is $\Theta(n)$. In contrast, when $c < 1$, the tree *dies out* after a finite number of steps. Hence, in the subcritical regime of $c < 1$, the components are finite in size and none is very large. In the supercritical regime of $c > 1$, the tree grows exponentially and, the larger c, the more likely its corresponding component is to span the entire network. At the critical point of $c = 1$, fluctuations drive the growth of the tree, and the size distribution of components follows a power law.

Implications. The logarithmic diameter of ER graphs implies that information or activation will tend to spread quickly across the network, because a signal needs to only cross a chain of $O(\log n)$ edges to reach *anywhere* in the network. That is, for any pair of nodes i, j, there exists a "short" path between them. If you recall how fast the binary search algorithm is at finding a target element within a sorted array, traversing an ER graph is similarly fast—the length of the longest chain of steps in both cases is $O(\log n)$.

This fact implies that the existence of such short paths can be explained by randomness and sparse connectivity alone, without needing to refer to other structural patterns. In this sense, short paths are "automatic" if a network has some amount of randomness in its connectivity structure. This property can be useful for spreading signals quickly, e.g., to synchronize different parts of a system, but also problematic if the goal is to decouple or insulate different parts of the network.

3 What $G(n, p)$ graphs look like

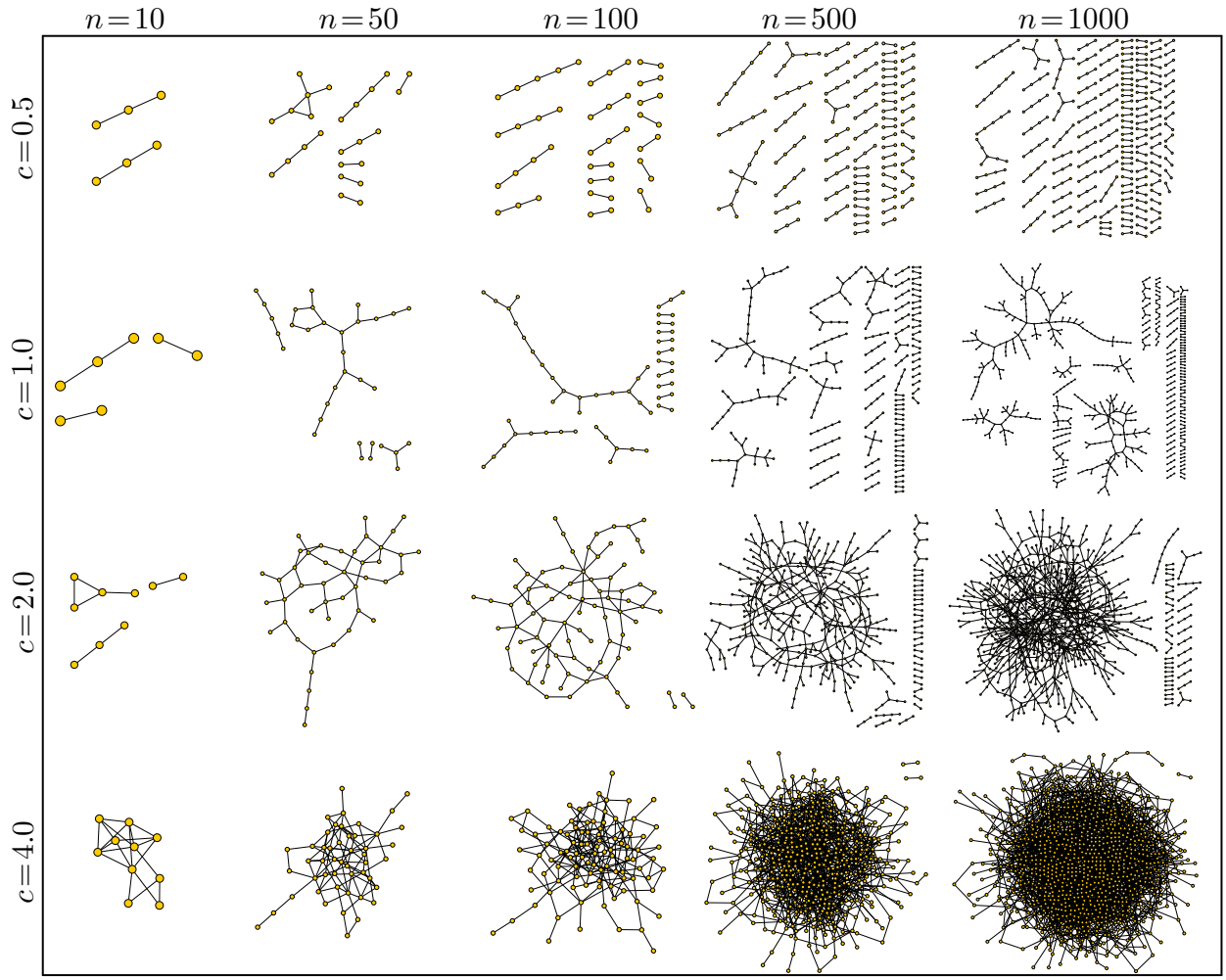
In this section, we'll build some intuition for the shape of ER graphs, and how that structure varies with mean degree c. The figure below shows simple visualizations[17] for networks of varying sizes $n = \{10, 50, 100, 500, 1000\}$ and mean degrees $c = \{0.5, 1.0, 2.0, 4.0\}$. And, to avoid visual clutter, all singleton vertices (degree $k = 0$) are omitted.

In the "ultra-sparse" case of $c < 1$, the networks are composed entirely of small or very small components, mostly perfect trees, and the largest connected component (LCC) contains $O(1)$ nodes. At $c = 1$, many of the little trees have begun to connect, forming a wide variety of larger components, most of which are still trees. However, for $c > 1$, we see the emergence of a very large connected component (in fact, a *giant component*), which will contain $O(n)$ vertices. Despite its size, when $c = 2$, we still see some "dust" components, i.e., the same small trees we saw for $c < 1$, around the giant component. In this regime, the LCC displays some interesting structure, being locally tree-like but exhibiting long cycles punctuated by tree-like whiskers.

Finally, for large mean degree (here, $c = 4$), the giant component contains nearly every vertex and has the appearance of a big hairball.[18] Although one cannot see it in these visualizations, the structure is still locally tree-like.

[17] The positions of nodes on the page were assigned via a standard spring-embedder algorithm like the Fruchterman-Reingold force-directed layout algorithm, in this case, implemented in the `yEd` software program.

[18] Visualizations of such networks are sometimes called *ridiculograms*, reflecting the fact that all meaningful structure is obscured. Such figures are surprisingly common in the networks literature.



4 Discussion

The Erdős-Rényi random graph model is a crucial piece of network science in part because it helps us build intuition about what kinds of patterns we should expect to see in our data, if the true generating process were a boring iid coin-flipping process on the edges.

Knowing that such a process produces a Poisson degree distribution, locally tree-like structure (meaning, very few triangles), small diameters, and the sudden appearance of a giant component gives us an appropriate baseline for interpreting real data. For instance, the fact that most real-

world networks also exhibit small diameters suggests that their underlying generating processes include some amount of randomness, and thus observing that some particular network has a small diameter is not particularly interesting.

4.1 Degree distributions

The degrees of vertices are a fundamental network property, and correlate with or drive many other kinds of network patterns. A key question in network analysis is thus

How much of some observed pattern is generated by the degrees alone?

We will return to this question in a later lecture, when we study the configuration random graph model, which is the standard way to answer such a question. In the meantime, we will focus on the simpler question of

How much of some observed pattern is generated by the edge density (mean degree) alone?

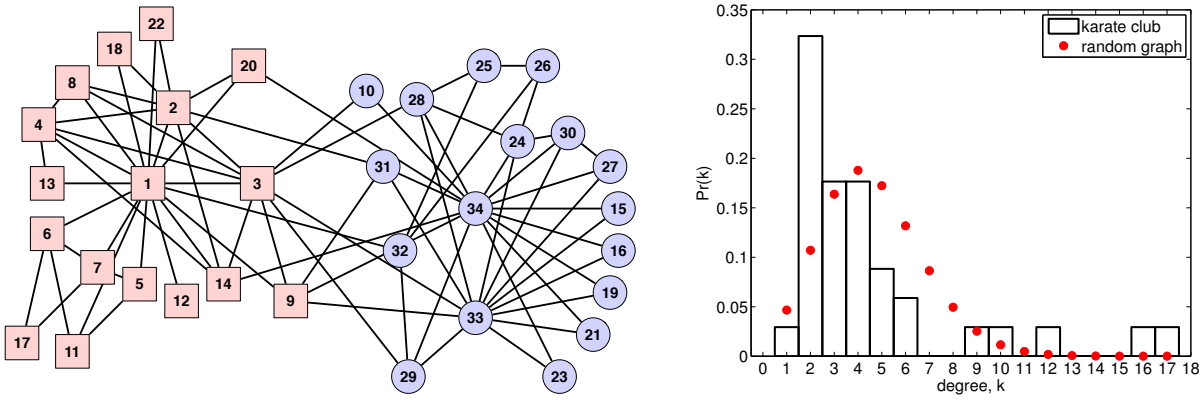
Recall that the degree distribution of the simple random graph has been claimed highly unrealistic. To illustrate just how unrealistic it is, we will consider two commonly studied social networks: (i) the "karate club" network, in which vertices are people who were members of a particular university karate club, and two people are connected if they were friends outside the club, and (ii) the "political blogs" network, in which vertices are political blogs from the early 2000s and two blogs are connected by a directed edge if one hyperlinks to the other.

4.1.1 The karate club

The left-hand figure on the next page shows the network, which has 34 vertices and 78 undirected edges, yielding a mean degree of $\langle k \rangle = 4.59$. This value is above the connectivity threshold for a random graph (recall Section 2.6), implying that we should expect this network to be well connected.

Examining the network's structure, we can see that that several vertices (1, 33 and 34) have very high degree, while most other vertices have relatively low degree. Now, we tabulate its degree distribution by counting the number of times each possible degree value occurs, and then normalizing by the number of vertices: $p_k = (\#\text{ vertices with degree } k)/n$, for $k \geq 0$. This probability mass function or distribution (pdf) is a normalized histogram of the observed degree values, which is shown in the right-hand figure, along with a Poisson distribution with parameter 4.59. That is, to compare the simple random graph with the karate club, we parameterize the model to be as close to the data as possible. In this case, it means setting their densities or mean degrees to be equal.

Notably, the degree distributions are somewhat similar. Both place a great deal of weight on the small values. However, at the large values, the distributions disagree. In fact, the Poisson distribution places such little weight on those degrees that the probability of producing a vertex with



degree $k \geq 16$ is merely 0.00000675, or about 1 chance in 15,000 random graphs with this mean degree. And in the karate club, there are 2 such vertices! The presence of these vertices here is thus very surprising from the perspective of the simple random graph.

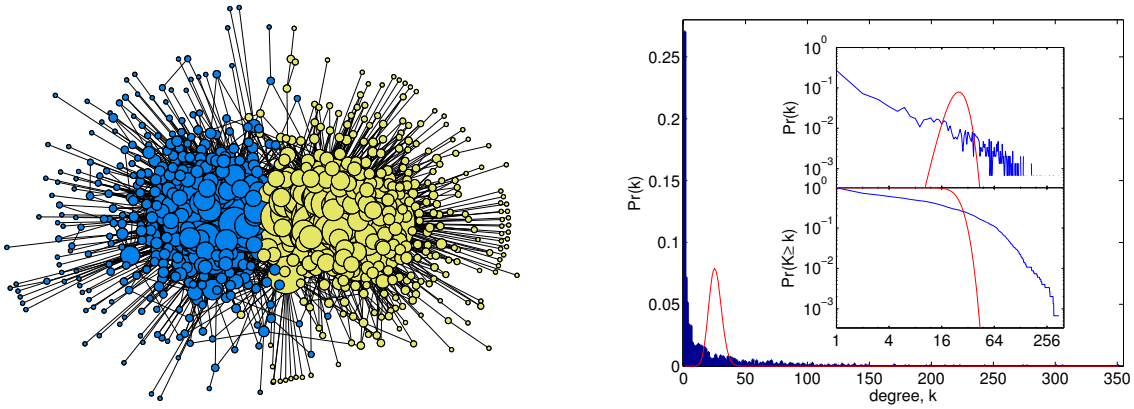
This behavior is precisely what we mean by saying that the simple random graph model produces unrealistic degree distributions. Or, to put it more mathematically, empirically we observed that degree distributions in reality are often "heavy tailed," meaning that as k increases, the remaining proportion of vertices with degree *at least* k decreases more slowly than it would in a geometric or exponential (or Poisson) distribution. That is, high-degree vertices appear much more often than we would naively expect.

4.1.2 The political blogs network

The left-hand figure below shows a visualization of the political blogs network,[19] which has $n = 1490$ vertices and $m = 33430$ (ignoring edge direction), yielding a mean degree of $\langle k \rangle = 44.87$. Just as we saw with the large connected instances of $G(n, p)$ in Section 3, visualizing this network doesn't tell us much. To understand its structure, we must rely on our network analysis tools and our wits.

The right-hand figure shows this network's degree distribution in three different ways: as a pdf on linear axes (outer figure) and on log-log axes (upper inset), and as the complementary cdf on log-log axes (lower inset). The log-log axes make it easier to see the distribution's overall shape, especially in the upper tail, where only a small fraction of the vertices live. The complementary

[19]Network image from Karrer and Newman, *Phys. Rev. E* **83**, 016107 (2011) at `arxiv:1008.3926`. Vertices are colored according to their ideological label (liberal or conservative), and their sizes are proportional to their degree. Data from Adamic and Glance, *WWW* Workshop on the Weblogging Ecosystem (2005).



cdf, defined as $\Pr(k \geq K)$,[20] and meaning the fraction of vertices with value at least some K, is useful for such distributions because the shape of the pdf becomes very noisy for large values of k (because there are either zero or one (usually zero) vertices with that value in the network), while the complementary cdf smooths things out to reveal the underlying pattern. Finally, in each case, a Poisson distribution with the same mean value is also shown, to illustrate just how dramatically different the degree distributions are.

The lower inset in the figure (the ccdf) reveals a fairly smooth shape for the degree distribution and reveals some interesting structure: the curvature of the ccdf seems to change around $k = 64$ or so, decreasing slowly before that value and much more quickly after. Furthermore, about 11% of the vertices have degree $k \geq 64$, making the tail a non-trivial fraction of the network. Furthermore, the density of edges alone explains essentially nothing about the shape of this degree distribution.

4.1.3 Commentary on degree distributions

The shape of the degree distribution is of general interest in network science. It tells us how skewed the distribution of connections is, which has implications for other network summary statistics, inferences about large-scale structural patterns, and the dynamics of processes that run on top of networks. The degree distribution is also often the first target of analysis or modeling: What pattern does the degree distribution exhibit? Can we model that pattern simply? Can we identify a social or biological process model that reproduces the observed pattern?

[20] Mathematically, $\Pr(K \geq k) = 1 - \Pr(K < k)$, where $\Pr(K < k)$ is the cumulative distribution function or cdf. The complementary cdf, or ccdf, always begins at 1, as all vertices have degree at least as large as the small value. As we increase k, the ccdf decreases by a factor of $1/n$ for each vertex with degree k, until it reaches a value of $1/n$ at $k = \max(k_i)$, the largest degree vertex in the network. The ccdf is typically plotted on doubly-logarithmic axes.

This latter point is of particular interest, as in network analysis and modeling we are interested not only in the pattern itself but also in understanding the process(es) that produced it. The shape of the degree distribution, and particularly the shape of its upper tail, can help us distinguish between distinct classes of models. For instance, a common claim in the study of empirical networks is that the observed degree distribution follows a *power law* form, which in turn implies certain types of exotic processes. Although many of these claims end up being wrong, the power-law distribution is of sufficient importance that we will spend the rest of this lecture learning about their interesting properties.

5 At home

1. Chapter 12 (pages 397–425) in *Networks*

6 Matlab code

Matlab code for generating Figure 2a,b.

```
% Figure 1a
c = [0.5 1 1.5]; % three choices of mean degree
S = (0:0.01:1);  % a range of possible component sizes

figure(1);
plot(0.583.*[1 1],[0 1],'k:','LineWidth',2); hold on;
plot(S,1-exp(-c(1).*S),'r-','LineWidth',2); % c = 0.5 curve
plot(S,1-exp(-c(2).*S),'r-','LineWidth',2); % c = 1.0 curve
plot(S,1-exp(-c(3).*S),'r-','LineWidth',2); % c = 1.5 curve
plot(S,S,'k--','LineWidth',2); hold off     % y = S   curve
xlabel('S','FontSize',16);
ylabel('y','FontSize',16);
set(gca,'FontSize',16);
h1=text(0.7,0.26,'c = 0.5'); set(h1,'FontSize',16,'Rotation',14);
h1=text(0.7,0.47,'c = 1.0'); set(h1,'FontSize',16,'Rotation',18);
h1=text(0.2,0.32,'c = 1.5'); set(h1,'FontSize',16,'Rotation',38);

% Figure 1b
S  = (0:0.0001:1);  % a range of component sizes
c  = (0:0.01:4);    % a range of mean degree values
Ss = zeros(length(c),1);
for i=1:length(c)
    g     = find(S - (1-exp(-c(i).*S))>0, 1,'first'); % find the intersection point
    Ss(i) = S(g);                                     % store it
end;

figure(2);
plot(c,Ss,'r-','LineWidth',2);
xlabel('Mean degree c','FontSize',16);
ylabel('Size of the giant component S','FontSize',16);
set(gca,'FontSize',16,'XTick',(0:0.5:4));
```