1 Random graph models

A large part of understanding what structural patterns in a network are *interesting* depends on having an appropriate reference point by which to distinguish interesting from non-interesting. In network analysis and modeling, the conventional reference point is a random graph, i.e., a network in which edges are random variables, possibly conditioned on other variables or parameters. The most basic of these random graph models is the Erdős-Rényi model, which is the subject of this lecture. Before describing its form and deriving certain properties, however, we will explore what it means to be a model of a network and identify two major classes of network models: generative and mechanistic.

1.1 What are models?

There are many models of network structure, and these largely can be divided into two classes: *mechanistic* models and *generative* or probabilistic 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, generally speaking, codifies or formalizes a notion of causality via a set of rules (often mathematical) that produces certain kinds of networks. Identifying the mechanism for some 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 (particularly for mechanisms proposed by physicists), and these produce specific kinds of topological patterns in networks. We will explore examples of such mechanisms later in the semester, including the *preferential attachment* mechanism, for which the evidence is fairly strong in the domain of scientific citation networks and the World Wide Web. Mechanistic models are thus most commonly found in hypothesis-driven network analysis and modeling, where the goal is specifically focused on cause and effect (recall Section 2 of Lecture 1).

Generative 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 most basic form of probabilistic network model is called the *random graph* (sometimes also the Erdős-Rényi random graph, after two of its most famous investigators, or the Poisson or Binomial random graph). In this and other generative models, edges exist probabilistically, where that probability may depend on other variables. The random graph model is the simplest such model, where every edge is an iid random variable from a fixed distribution. In this model, a single parameter determines everything about the network. Generative models are thus most commonly found in exploratory network analysis and modeling, where the goal is to identify interesting structural patterns that deserve additional explanation (recall Section 2 of Lecture 1).

The attraction of generative models is that many questions about their structure, e.g., the network measures we have encountered so far, may be calculated analytically, or at least numerically. This provides a useful baseline for deciding whether some empirically observed pattern is interesting. For instance, let G denote a graph and let $\Pr(G)$ be a probability distribution over all such graphs. The typical or expected value of some network measure is then given by

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

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.[1] If some observed value $\langle x_{\rm data} \rangle$ is very different from the value expected from the model $\langle x_{\rm model} \rangle$, then we may conclude that the true generating process for the data is more interesting than the simple random process we assumed. This approach to classifying properties as interesting or not 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 is the "original" random graph model, and was most prominently studied extensively by the Hungarian mathematicians Paul Erdős (1913–1996)[2] and Alfréd Rényi (1921–1970)[3] (although it was, in fact, studied earlier).

This model is typically denoted $G(n, p)$ and has two parameters: n the number of vertices and p the probability that each simple edge (i, j) exists.[4] These two parameters specify everything about the model. In terms of the adjacency matrix, 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.

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

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

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

[4] Another version of this model is denoted $G(n, m)$ which places exactly m edges on n vertices. This version has the advantage that m is no longer a random variable.

The utility of this model lies mainly in its mathematical simplicity, not in its realism. Virtually none of its properties resemble those of real-world networks, but they provide a useful baseline for our expectations and provide a warmup for more complicated generative models.

To be precise $G(n, p)$ defines an *ensemble* or collection of networks, which is equivalent to the distribution over graphs $\Pr(G)$. When we calculate properties of this ensemble, we must be clear that we are not making statements about individual instances of the ensemble, but rather making statements about the typical member.[5]

2.1 Mean degree and degree distribution

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 $\langle k \rangle = 2m/n$.

[5] 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.

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

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

In other words, each vertex has $n-1$ possible partners,[6] 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.

Because edges in $G(n, p)$ are iid random variables, the entire degree distribution has a simple form

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

which is a binomial distribution with parameter p for $n-1$ independent trials. 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

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

where we have used a first-order Taylor expansion of the logarithm[7] 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 as $n \to \infty$. Thus, the expression for our degree distribution becomes

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

[6] 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.

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

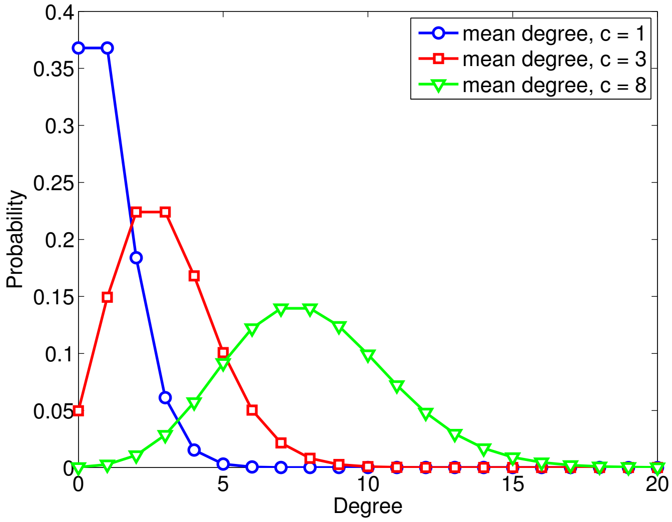
which may be simplified further still. The binomial coefficient is

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

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

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

which is called the Poisson distribution. This distribution has mean and variance c, and is slightly asymmetric. The figure 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.



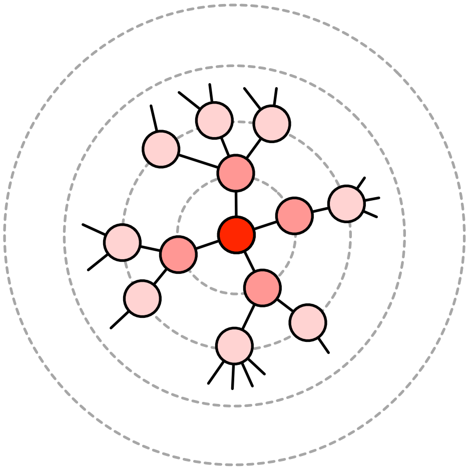
2.2 Clustering coefficient, triangles and other loops

The density of triangles in $G(n,p)$ is easy to calculate because very edge is iid. The clustering coefficient is

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

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 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. This implies that $G(n,p)$ graphs are locally *tree-like* (see figure below), meaning that if we build a tree outward from some vertex in the graph, we rarely encounter a "cross edge" that links between two branches of the tree.



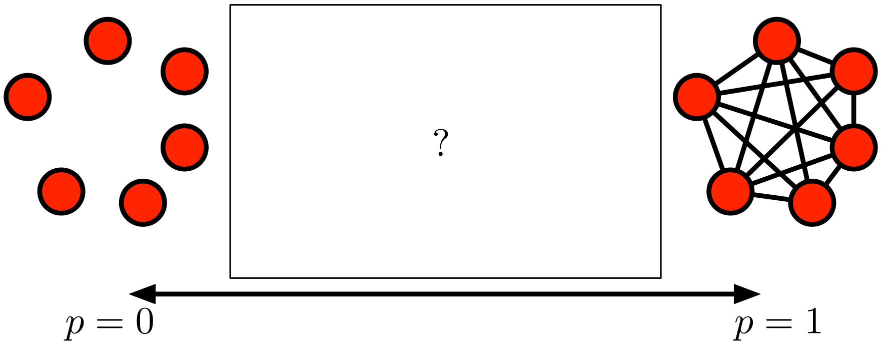
Simple random graphs are locally tree-like

This property is another that differs sharply from real-world networks, particularly social networks, which tend to have many triangles, and are thus not locally tree-like.

2.3 A phase transition in network connectedness

This random graph model exhibits one very interesting property, which is the sudden appearance, as we vary 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*.[8]

We care about phase transitions because they represents *qualitative* changes in the fundamental behavior of the system. Because they are inherently non-linear effects, in which a small change in some parameter leads to a big change in the system's behavior, they often make good models of the underlying mechanisms of particular complex systems, which often 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.[9] 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 in fact, something much more interesting happens. (When a graph is sparse, what other network measures are intensive? What measures are extensive?)

[8] 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.

[9] 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`

2.3.1 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

$$\begin{aligned} u &= (1-p+pu)^{n-1} \\ &= \left[1-\frac{c}{n-1}(1-u)\right]^{n-1} && (9) \\ &= \mathrm{e}^{-c(1-u)} && (10) \end{aligned}$$

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.[10]

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. (10) 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:

$$S = 1-\mathrm{e}^{-cS} \quad . \tag{11}$$

Note that this equation is transcendental and there is no simple closed form that isolates S from the other variables.[11]

[10] We can sidestep using the second identity by taking the logarithms of both sides of Eq. (9):

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

where the approximate equality becomes exact in the limit of large n. Exponentiating both sides of our approximation then yields Eq. (10). This should look familiar.

[11] 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 $W(z)\mathrm{e}^{W(z)}=z$.

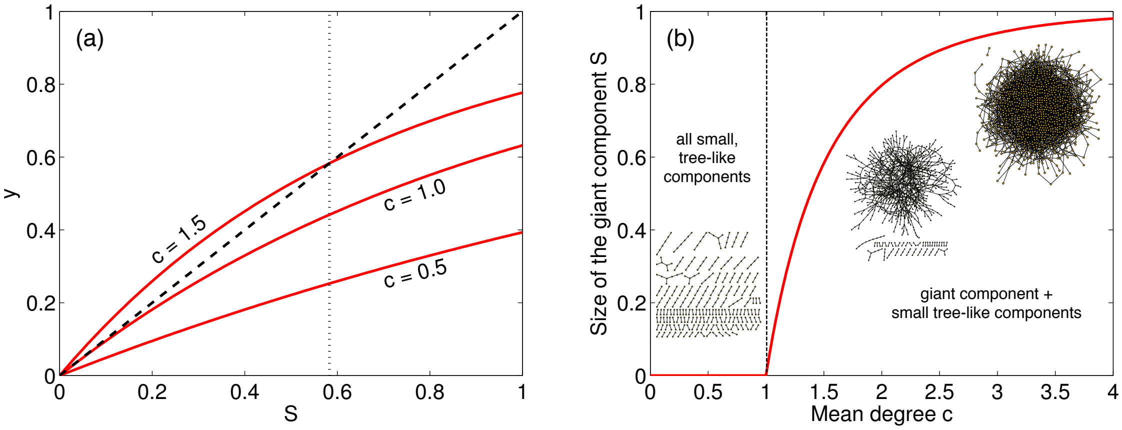


Figure 1: (a) Graphical solutions to Eq. (11), showing the curve $y = 1 - \mathrm{e}^{-cS}$ for three choices of c along with the curve $y = S$. The locations of their intersection gives the numerical solutions to Eq. (11). Any solution $S > 0$ implies a giant component. (b) The solution to Eq. (11) 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.

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. (11) 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".

2.3.2 Branching processes and percolation

An alternative analysis considers building each component, one vertex at a time, via a *branching process*. Here, the mean degree c plays the role of the expected number of additional vertices that are joined to a particular vertex i already in the component. The analysis can be made entirely analytical, but here is a simple sketch of the logic.

When $c < 1$, on average, this branching process will terminate after a finite number of steps, and the component will have a finite size. This is the "sub-critical" regime. In contrast, when $c > 1$, the average number of new vertices grows with each new vertex we add, and thus the branching process will never end. Of course, it must end at some point, and this point is when the component has grown to encompass the entire graph, i.e., it is a giant component. This is the "super-critical" regime. At the transition, when $c = 1$, the branching process could in principle go on forever, but instead, due to fluctuations in the number of actual new vertices found in the branching process, it does terminate. At $c = 1$, however, components of all sizes are found and their distribution can be shown to follow a power law.

2.4 A small world with $O(\log n)$ diameter

The branching-process argument for understanding the component structure in the sub- and super-critical regimes can also be used to argue that the diameter of a $G(n, p)$ graph should be small, growing like $O(\log n)$ with the size of the graph n. Recall that the structure of the giant component is locally tree-like and that in the super-critical regime the average number of offspring in the branching process $c > 1$. Thus, the largest component is a little like a big tree, containing $O(n)$ nodes and thus, with high probability, has a depth $O(\log n)$, which will be the diameter of the network. This informal argument can be made mathematically rigorous, but we won't cover that here.

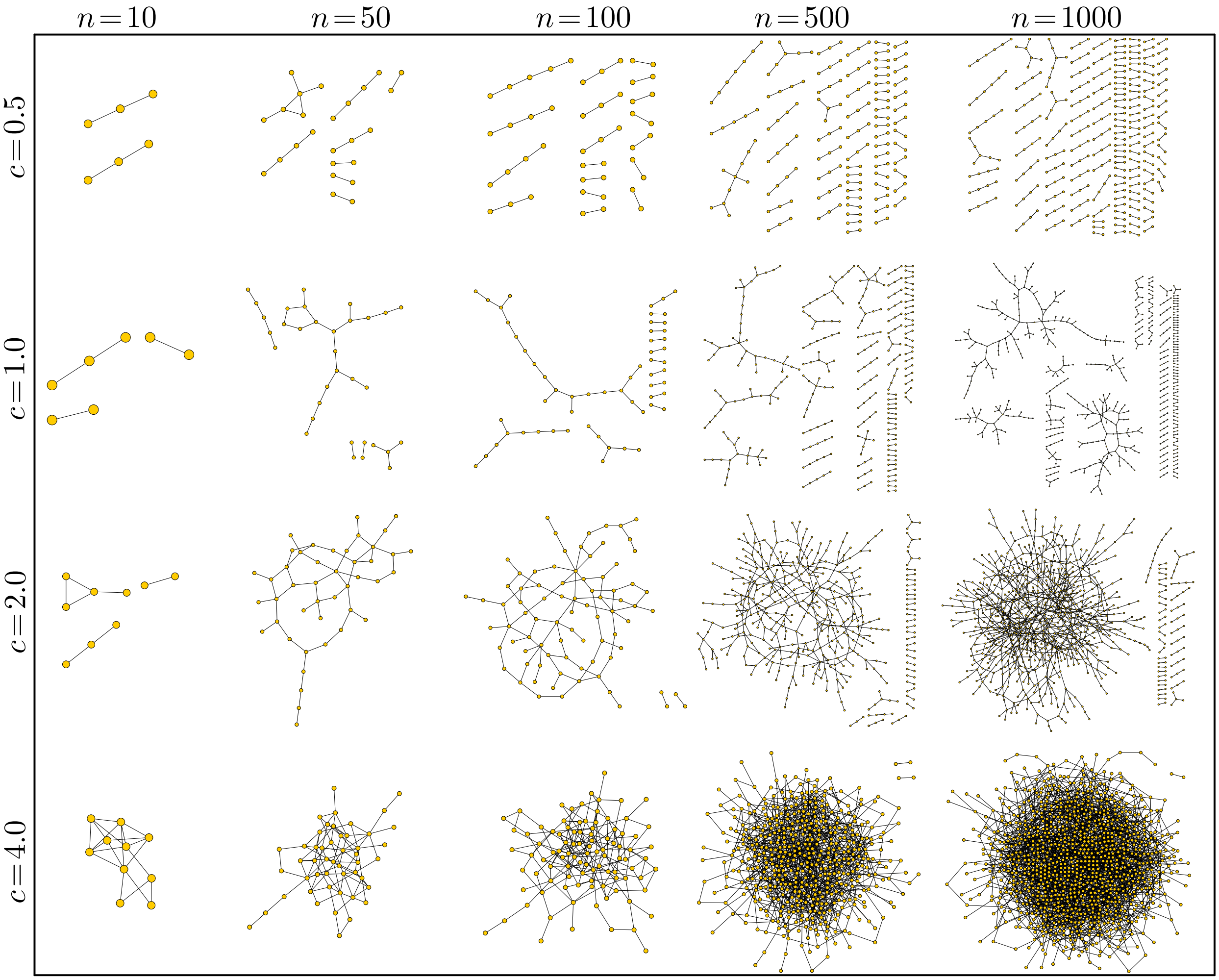
3 What $G(n, p)$ graphs look like

Generating instances of $G(n, p)$ is straight forward. There are at least two ways to do it: (i) loop over the upper triangle of the adjacency matrix, checking if a new uniform random deviate $r_{ij} < p$, which takes time $O(n^2)$; or (ii) generate a vector of length $n(n-1)/2$ of uniform random deviates, threshold them with respect to p, and then use a pair of nested loops to walk the length of the vector, which still takes time $O(n^2)$. A third way, which does not strictly generate an instance of $G(n, p)$, is to draw a degree sequence from the Poisson distribution to construct the network, which takes time $O(n + m \log m)$. In the sparse limit, the latter approach is essentially linear in the size of the network, and thus substantially faster for very large networks.

To give some intuition about what kind of shapes these simple random graphs take, the figure below shows simple visualizations (laid out on the page using a standard spring-embedder algorithm like the Fruchterman-Reingold force-directed layout algorithm) for $n = \{10, 50, 100, 500, 1000\}$ vertices with mean degree $c = \{0.5, 1.0, 2.0, 4.0\}$ (with $p = c/(n-1)$). Additionally, in these visualizations, singleton vertices (those with degree $k = 0$) are omitted.

A few things are notable about these graphs. For $c < 1$, the networks are composed of small or very small components, nearly all of which are perfect trees. At $c = 1$, many of these little trees have begun to connect, forming larger components. Most of the components, however, are still perfect trees, although a few loops appear. For $c > 1$, we see the giant component phenomenon, with nearly all vertices connected in a single large component. However, for $c = 2$ and sufficiently large graphs, we do see some "dust," i.e., the same small trees we saw for $c < 1$, around the giant component. The giant component itself 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.[12] Although one cannot see it in these visualizations, the structure is still locally tree-like.



[12] 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?

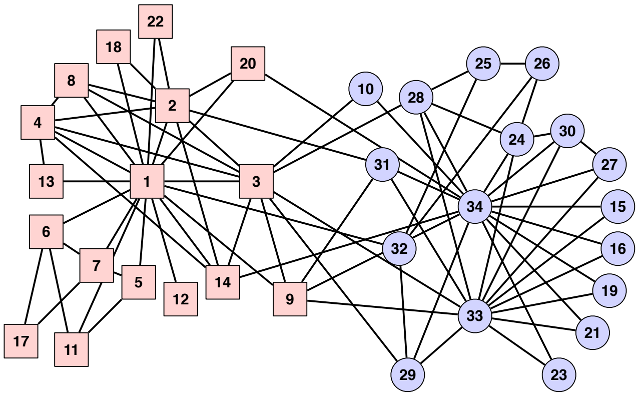
We will return to this question in two weeks, when we study the configuration random graph model, which is the standard way to answer such a question. In the meantime, we will focus on the simpler question of asking how much of some observed pattern is generated by the *density* (mean degree) alone, which is the one parameter of the simple random graph model.

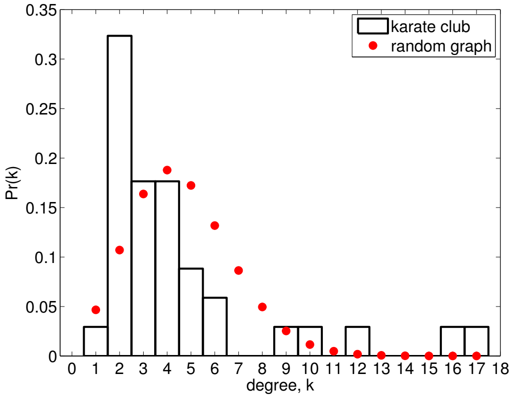
Recall that the degree distribution of the simple random graph has been claimed highly unrealistic. To illustrate just how unrealistic it is, we will consider two commonly studied social networks: (i) the "karate club" network, in which vertices are people who were members of a particular university 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 2.3), 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.





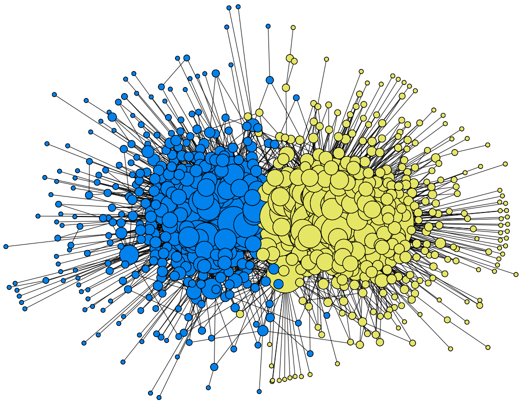
Notably, the degree distributions are somewhat similar. Both place a great deal of weight on the small values. However, at the large values, the distributions disagree. In fact, the Poisson distribution places such little weight on those degrees that the probability of producing a vertex with degree $k \geq 16$ 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.

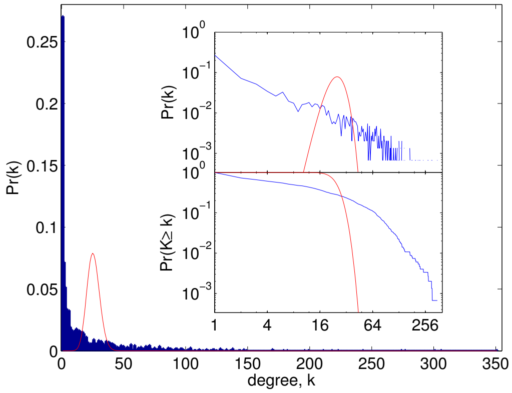
This behavior is precisely what we mean by saying that the simple random graph model produces unrealistic degree distributions. Or, to put it more mathematically, empirically we observed that degree distributions in reality are often “heavy tailed,” meaning that as k increases, the remaining proportion 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,[13] 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 cdf, defined as $\Pr(k \geq K)$,[14] 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

[13] 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).

[14] 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.

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?

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 1a,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));
```