Inference, Models and Simulation for Complex Systems — CSCI 7000-001
Lecture 13 — 18 October 2011
Prof. Aaron Clauset

1 Models of network structure

There are now a very large number of models of network structure. Some of these models are *mechanistic*, meaning that they codify or formalize a set of rules (often mathematical) that produces certain kinds of networks. The purpose of these models is often to represent certain notions of cause and effect. Typically, these models have a single dominant mechanism, designed to produce certain specific kinds of topological patterns. For instance, the *preferential attachment* mechanism and the *duplication-mutation* mechanism (which is a kind of "vertex-copying" mechanism).

Other models are *generative*, meaning that they generate network structure, but typically without as strong a notion of causality as is the case in mechanistic models. *Random graph* models are a special kind of generative model, in which the existence of edges ranges from iid to various relaxations of the iid assumptions (for instance, independent but drawn from a mixture of different distributions). Some of the simple models of random graphs have structure that can be determined analytically. The most common types of random graph models are the Erdös-Rényi random graph (sometimes called a Poisson random graph or Binomial random graph), in which all edges are iid, and the configuration model, in which edges are independent but the graph is conditioned to have a specific degree sequence.

We won't cover all or even most of the major network models. Instead, we'll cover a few highlights that will help us think more generally about the others.

2 The Erdös-Rényi random graph

This network model is the original random-graph model, and was studied extensively by the famous Hungarian mathematicians Paul Erdös (1913–1996),[1] and Alfréd Rényi (1921–1970)[2] although it was studied earlier, as well. This model is typically denoted $G(n, p)$ for its two parameters: n the number of vertices and p the probability that an edge (i, j) exists, for all i, j. The beauty of this model lies mainly in its mathematical simplicity—almost everything about its structure can be calculated analytically—not in its realism. Virtually none of its properties resemble those of real-world networks.

[1] http://xkcd.com/599/
[2] "A mathematician is a machine for turning coffee into theorems."

To be precise $G(n,p)$ defines an *ensemble*, i.e., a collection, of networks and when we calculate properties of these, we must be clear that we are computing properties of the ensemble, not of individual instances of the ensemble.

2.1 Poisson degree distribution

Let's warm up with a few simple calculations. In $G(n,p)$, every edge exists independently and with the same probability. 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 simply a standard binomial distribution. The mean value can be derived using the Binomial Theorem:

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

That is, the mean degree is the expected number of the $\binom{n}{2}$ possible ties that exist, given flipping a coin that comes up "heads" with probability p. The mean degree of a vertex in a network with m edges is $\langle k \rangle = 2m/n$. Thus, the mean degree in $G(n,p)$ can be derived, using Eq. (2), as

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

which converges asymptotically on pn. Sometimes, the average degree $\langle k \rangle$ is called c. This result should be unsurprising as it's completely equivalent to asking what is the expected number of "heads" in $n-1$ trials when each one occurs iid with probability p.

And, because ties occur iid with probability p, the entire degree distribution is also given by a Binomial distribution

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

Recall from Lecture 1 that in a Binomial distribution, when p is small, the distribution converges on a Poisson distribution. Thus, in the sparse limit of $G(n,p)$, which is typically the limit we consider, the degree distribution is given by

$$\Pr(k) = \mathrm{e}^c \frac{c^k}{k!} \ , \tag{5}$$

where $c = p(n-1) = \langle k \rangle$. This is one way that $G(n,p)$ is unrealistic: most real-world networks exhibit more heterogeneous (broad; high-variance) degree distributions at the whole-network scale.

2.2 Vanishing transitivity

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

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

If the average degree is constant $\langle k \rangle$, then the clustering coefficient is $C = O(1/n)$. This implies that there are a constant number of triangles in a network, independent of how large the network is.

This calculation can be generalized to loops of longer length or cliques of larger size and a consequence of this generalization is that $G(n,p)$ graphs are locally *tree-like*. That is, they have very few loops of any length. This behavior is related to *expander graphs*, a concept often discussed in graph theory in computer science, logarithmic diameters and rapid mixing of stochastic processes on these networks. This is another way that $G(n,p)$ random graphs are unrealistic. Social networks, for example, have large clustering coefficients, even for extremely large n.

2.3 A giant component and a phase transition

The emergence of a giant component [of size $O(n)$] is an example of a *phase transition*.[3]

To begin, consider the two extremes of p: if $p = 0$ we have a fully empty network with n completely disconnected vertices; in contrast, if $p = 1$, every edge exists, the network is an n-clique, and every vertex is a distance $\ell = 1$ from every other vertex. In the jargon of physics, the size of the component in the latter case (complete graph) is an *extensive* property,[4] meaning that it grows with the system size n; in the former case (empty graph), it's a non-extensive property because if we make the system larger, the size of the largest component (1) remains the same. Thus, the size of the largest component goes from being an intensive property to an extensive property as we gradually increase the density of edges p in the graph.[5]

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, $u < 1$. Equivalently,

[3] 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.

[4] Other examples of extensive properties in physics include mass, volume and entropy. Examples of *intensive* properties—those that are independent of the size of the system—include the density, temperature, melting point, and pressure.

[5] One possibility is that the size of the largest component is extensive only in the limit $p \to 1$, but in fact, something much more interesting happens.

we could consider u to 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

$$\begin{aligned} u &= (1-p+pu)^{n-1} \\ &= \left[1-\frac{c}{n-1}(1-u)\right]^{n-1} \qquad (6) \\ &= \mathrm{e}^{-c(1-u)} \qquad (7) \end{aligned}$$

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.[6]

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. (7) and eliminating u in favor of S yields

$$S = 1-\mathrm{e}^{-cS} \quad . \qquad (8)$$

This equation was first given by Erdös and Rényi in 1959 and tells us the size of the giant component, as a fraction of the total size of the network n, as a function of the mean degree c. Unfortunately, this equation is transcendental and has no simple closed form.[7] But, we can get an intuitive feel for its behavior by plotting the function $y = 1-\mathrm{e}^{-cS}$ for $S \in [0,1]$ and asking where it intersects the line $y = S$. The intersection is the solution to Eq. (8) and tells us the size of the giant component. Figure 1 shows this exercise graphically (see Section 4 for the Matlab code that generates these figures). In the "sub-critical" regime $c < 1$, the curves only intersect at $S = 0$, implying that

[6] We can sidestep using the second identity by taking the logarithms of both sides of Eq. (6):

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

where the approximate equality becomes exact in the limit of large n. Exponentiating both sides of our approximation then yields Eq. (7).

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

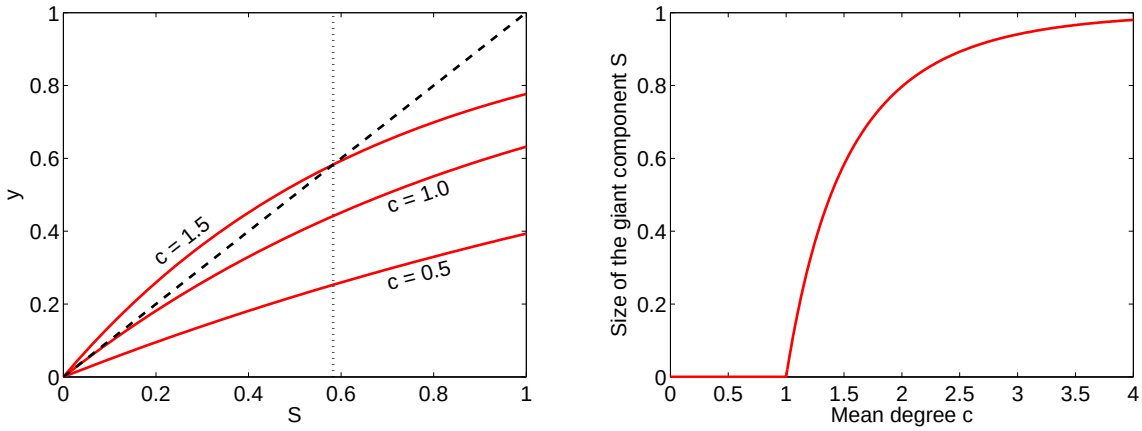

Figure 1: (a) Graphical solutions to Eq. (8), showing the curve $y = 1 - \mathrm{e}^{-cS}$ for three choices of c along with the curve $y = S$. The locations of their intersection gives the numerical solutions to Eq. (8). Any solution $S > 0$ implies a giant component. (b) The solution to Eq. (8) as a function of c, showing the discontinuous emergence of a giant component at the critical point $c = 1$.

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".

2.3.1 Branching processes and percolation

Another way to prove that the giant component exists for $c > 1$ is to model the exploration of a component as a Galton-Watson branching process (this approach is more mathematical, but has the advantage of allowing us to say much more about both the giant component and the size and shape of the components not part of the giant one). There are three cases to consider: (i) the sub-critical regime of $c < 1$, the critical regime of $c = 1$ and the super-critical regime of $c > 1$.

In general, this approach considers a "percolation" or branching process on the exploration of the component containing a vertex i. In the sub-critical regime, the average number of "offspring" or neighbors in the branching process is less than 1 and the branching process tends to die out very quickly (there is a close analogy here with Poisson processes). This leads to a network composed of mostly small trees. In the super-critical regime, the average number of offspring in the branching process is larger than 1 and the process exhibits exponential growth. At the critical point, the average number of offspring is exactly 1, and the size of the branching process is controlled by the variance rather than the average.

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.

2.5 Drawing networks from $G(n,p)$

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 and use the configuration model (see Section 3) 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.

3 The configuration model

The degree distribution of $G(n,p)$ follows a Poisson form, which isn't particularly realistic. We can improve this by considering a generalization of $G(n,p)$ to the "configuration model," which we can denote $G(n,\vec{k})$ where $\vec{k} = \{k_i\}$ is a *degree sequence* and k_i is the degree of vertex i.

In general, the degree sequence $\vec{k}$ can be just about anything and could, for instance, be drawn iid from a probability distribution $\Pr(k)$. If $k \sim \text{Poisson}(\lambda)$, we are generating something like an instance of $G(n,p)$, but we could just as easily draw $\vec{k}$ from some other distribution or set it equal to an empirical degree sequence.

Given a degree sequence, a simple construction technique, given in Section 3.2, can produce an instance of the corresponding ensemble. Notably, these graphs are not strictly simple like $G(n,p)$. Instead, they are random multi-graphs, and both self-loops and multi-edges can occur. Fortunately, their number is typically a vanishing fraction of the full network [the argument follows closely that of the vanishing clustering coefficient for $G(n,p)$] and can generally be safely ignored (or removed).

3.1 Mathematical properties

The precise mathematical properties for a configuration model random graph depend, unsurprisingly on the particular degree sequence. For many kinds of degree sequences, for example, a k-regular random graph where every degree is the same or a power-law random graph where $\Pr(k) \propto x^{-\alpha}$, many properties can be calculated exactly. All depend on the general fact that the probability of two vertices i and j being connected depends only on their respective degrees:

$$p_{ij} = \frac{k_i k_j}{2m - 1} \ . \tag{9}$$

That is, the higher the degrees of i and j, the more chances under the configuration model they have to be connected.

These kinds of random graphs also typically exhibit the vanishing clustering coefficient, emergence of a giant component and a logarithmic diameter. For graphs with highly skewed degree distributions, the high-degree vertices tend to have high centrality scores, for example, high betweenness and high closeness centralities, because these vertices are likely to be connected to each other (the probability of i and j being connected is multiplicative in their degrees, so extremely high-degree vertices are extremely likely to be connected, relative to any particular pair of low-degree vertices) and tend to be located in the middle of the network.

Another pattern in such skewed degree distributions is that of degree disassortativity (see Lecture 8), meaning that high-degree nodes tend to connect to many low-degree nodes; this behavior may sound contradictory to the previous statement, but it's not. Because high-degree nodes are relatively few in these graphs, the low-degree nodes swamp the correlation coefficient and thereby induce effective degree-degree anti-correlations.

These and many more properties are discussed in more detail in Chapter 13 of M.E.J. Newman's "Networks: An Introduction." Oxford Press (2010).

3.2 Generating networks from the configuration model

Given a degree sequence $\vec{k} = \{k_1, k_2, \ldots, k_n\}$, we can generate a random graph that has exactly this set of degrees but is otherwise random in the following way. Let v be a vector of length $2m$ and let us write the index of a vertex i k_i times in the vector v. If we then take a random permutation of the entries of v, and then take the entries sequentially in pairs, we get a random matching of the degrees or "stubs" of the vertices. The random permutation can be done easily by storing a uniform random deviate r_j with each entry in v and then sorting the values in v based on the random deviates. Using a reasonable sorting algorithm like QuickSort leads to a construction time of $O(n + m \log m)$. Figure 2 illustrates the stub-matching process schematically.

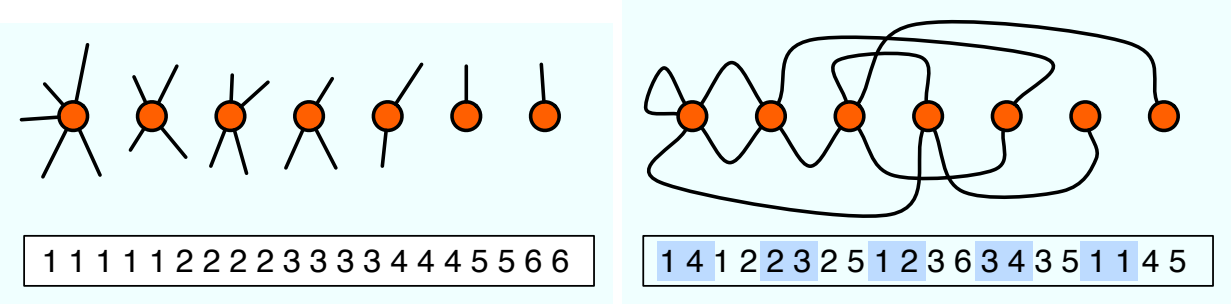

Figure 2: An example of the "stub matching" construction technique for generating an instance of the configuration model. On the left, we have written the vertices and their degrees (represented as "stubs" of "half edges"), along with the vector v filled with a number of copies of a vertex's index equal to its degree. On the right, a wired up network where the stubs have been matched randomly; v now shows a random permutation of the indices, grouped in pairs to indicate which stubs were connected. Note the appearance of self-loops and multi-edges.

4 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));
```