Agenda

- Random Graphs
  - giant component and small world
  - statistics
  - problems: degree distribution and triangles

Recap

Recall that a graph $G = (V, E)$ consists of a set of vertices V and a set of edges $E \subset V \times V$. We can represent the graph with an adjacency matrix A, where $A_{ij} = 1$ if $(i, j) \in E$, otherwise $A_{ij} = 0$.

Random Graph Model; parameterization 1

The simplest model for a graph we can have is the *random graph model* $G(n, p)$, where n is the number of nodes, p is the probability of an edge, and all edges form independently with probability p.

In the directed version, A_{ij} are independently distributed with a Bernoulli(p) distribution, while in the undirected case $A_{ij} = A_{ji}$ is distributed Bernoulli(p); that is, you perform one Bernoulli trial for each pair of edges, and record the result in both A_{ij} and A_{ji}.

There are no dependencies whatsoever in the random graph model, and we only have to specify one parameter (the probability p). All edges are independent and equi-probable. In a graph of n nodes, each node can have edges with up to $n - 1$ other nodes. The degree of a node i is thus the sum of $n-1$ independent Bernoulli(p) distributions, and so is Binomial($n-1$, p). Thus, with a constant p, the degree diverges to infinity as $n \to \infty$.

Random Graph Model; parameterization 2

We can also use an alternate parameterization for the random graph model, called *parameterizing by mean degree*. Let $\lambda = (n - 1)p$ (the mean degree).

In this version, we hold λ constant as $n \to \infty$ (so p is decreasing).

The different parameterizations lead to different interpretations. In the first case, we have the *dense graph (sequence) limit*, in which the expected degree increases as $n \to \infty$. In the second case, we have the *sparse graph (sequence) limit*, in which connectivity is constant, so the graph gets sparser and sparser.

In the mean degree parameterization, Degree(i) $\sim$ Poisson(λ); for this reason, we sometimes call them "Poisson random graphs" if they have the mean degree parameterization.

How many connected components?

The number of connected components depends on p; at $p = 1$, clearly there is just 1 connected component, while at $p = 0$, each node is a connected component. What is interesting is that at some magic intermediate value, there is a phase shift from many small connected components, to one CC of size $O(n)$ and a few small ones. There are several arguments to show this claim:

1. **Self-consistency argument:** suppose that such a "giant" component exists. Say that a fraction $s > 0$ of all nodes are in this giant component (GC). For any node i, the probability that it is in GC is s. Node i is not in the GC if for each of the other $n - 1$ nodes, i is not connected to that node (with probability $1 - p$) or is connected to the node and that node is not in GC (probability $p(1 - s)$). The probability i is not in the GC is thus

$$\begin{aligned} 1 - s &= (1 - p + p(1 - s))^{n-1} \\ &= (1 - ps)^{n-1} \\ &= \left(1 - \frac{s\lambda}{n-1}\right)^{n-1} \end{aligned}$$

   Hence, taking the log of both sides,

$$\log(1 - s) = (n - 1) \log \left(1 - \frac{s\lambda}{n-1}\right)$$

As long as x is small, $\log(1+x) \approx x$, so

$$\log(1-s) \approx -(n-1)\frac{\lambda s}{n-1} = -\lambda s$$

And so

$$s = 1 - e^{-\lambda s}$$

So what are the solutions for s? Clearly $s = 0$ is always a solution, but this is not a very helpful one. Can we get a solution where $s > 0$? If we plot $1 - e^{-\lambda s}$ for a range of s and see where it intersect the line $y = s$, we can find solutions. If $\lambda \leq 1$, it turns out that there is only the trivial solution. If $\lambda > 0$, then we can find a non-trivial solution $s^* > 0$. The solution s^* increases as λ increases, and the solution will always be bounded above by 1 since $s^* = 1 - e^{-\lambda s^*} \leq 1$.

2. **"Epidemic" picture for the giant component:** Consider node i in the graph. It has Z_1 first neighbors, where $Z_1 \sim \text{Poisson}(\lambda)$. each first neighbor has a random number of neighbors, as do their neighbors, and so on.

   The expected number of first neighbors is $E(Z_1) = \lambda$. The number of second neighbors is the sum of all neighbors of the first neighbors, not including the initial vertex (we neglect overlapping because λ is fixed and n is large, so the probability of overlap is small):

$$Z_2 = \sum_{j=1}^{Z_1} (\text{Poisson}(\lambda) - 1)$$

   The expectation of each summand is $\lambda - 1$, so $E[Z_2] = E[Z_1](\lambda - 1) = \lambda(\lambda - 1)$. Hence, $E[Z_1 + Z_2] = \lambda^2$. After k steps, we expect about λ^k nodes: $E[Z_1 + Z_2 + \cdots + Z_k] \approx \lambda^k$. If $\lambda < 1$, this tends to a finite number. If $\lambda > 1$, this tends to infinity.

   This argument maps the connected component onto a branching process: each object i produces Z_i branches to new objects, and does this independently of the other objects. If $E[Z_i] \leq 1$, we get *subcritical*

branching; with probability 1, the population will go extinct in finite time. If $E[Z_i] > 1$, we get supercritical branching; there is a positive probability the population never goes extinct. The branching process lets us see how the giant component happens; as long as $E[Z] > 1$ ($\lambda > 1$) then it will keep growing. If $\lambda \leq 1$, it will fizzle out.

Because of the giant connected component, we get the small-world phenomenon in random graphs. The number of nodes reached after k steps is roughly $O(\lambda^k)$, and the size of the giant component is about $O(n)$. How big can k get inside the GC (how big can the diameter be)?

$$O(\lambda^k) = O(n) \Rightarrow k \log(\lambda) = \log(n) \Rightarrow k = O\left(\frac{\log n}{\log \lambda}\right)$$

Thus we get the small-world phenomenon (diamter is $O(\log(n))$ in random graphs if $\lambda > 1$.

Statistics w/ Random Graphs

We can show that the likelihood is

$$L(p) = P(A; n, p) = \prod_{i<j} p^{A_{ij}} (1-p)^{1-A_{ij}}$$

and so the log-likelihood is

$$\ell(p) = \sum_{i<j} \log(1-p) + A_{ij} \log\left(\frac{p}{1-p}\right)$$

This produces the MLE for p:

$$\hat{p}_{MLE} = \frac{\sum_{i<j} A_{ij}}{\binom{n}{2}}$$

that is, the most likely value for the density is the observed density. Furthermore, the likelihood belongs to an exponential family, so we know that the MLE is strongly consistent (converges almost surely) and efficient (which gives Gaussian CIs for large n). These are all great properties; what could go wrong?

Problems with the Random Graph Model

Turns out, the random graph model is a horrible model of any real network known to science. There are two big things that rule it out:

1. degree distributions in real networks are not binomial/Poisson (they are much too light-tailed and symmetric)

2. because edges form independently in the random graph model, there are too few triangles

Triangles and transitivity

The proportion of triples forming triangles is p^3 (independent probability for each of the three edges). Also, $P(\text{triangle}|\text{ two edges connected}) = p$. But in real social networks, $P(\text{triangle}|\text{ two edges connected}) > p$, while in some applications (like electric grids) $P(\text{triangle}|\text{ two edges connected}) < p$.

This gives us a lesson for model criticism: find things where (1) the model makes predictions and (2) you didn't fit to them, to (3) check against data.

The problem with the random graph model is that everything is independent and exchangeable; there are always independent, equal probabilities of an edge between two nodes, so there is nothing intrinsically different about nodes. In other words, the model is too symmetric. To fix the problems, we have to introduce some sort of asymmetry.

One thing we can do is make some edges dependent on each other. This produces latent-space models, exponential family models, and stochastic block models.

We could also let the probabilities vary; this gives inhomogenous random graphs, block models, P_1 models, configuration models, and β models.

Finally, we could differentiate nodes, producing covariate-based link prediction, block models, and some sorts of "degree-correction" (where we have an idea that some nodes are more popular than others).