

Lecture 2

Probability and Random Networks



Figure 2.1: Probability of three different possible outcomes.

2.1 Basic Probability Theory

Before we move on to exploring stochastic network models, let us make a quick review of basic probability theory. In the frequentist approach to probability, the probability of an outcome x is the fraction of times that outcome x occurs if we were to repeat the same process an infinite number of times.

Let us consider a simple example with three possible outcomes. We divide a line of unit length into three pieces of length p_1 , p_2 and p_3 , as shown in Fig. 2.1. If we randomly throw darts at it, each dart will hit each piece a number of times proportional to the length of each section. If the game we are playing is such that I win whenever the dart hits length 1 or 2, what is the probability that I win? The total winning length is $p_1 + p_2$, so if I randomly throw each dart, the probability of a winning throw is:

$$\frac{p_1 + p_2}{p_1 + p_2 + p_3} = p_1 + p_2$$

where the equality stems from the fact that the total length is equal to 1.

What if instead of being interested in the outcome of a single throw, I'm interested in two consecutive throws? In this case, we can represent all possible outcomes by the square shown in Fig. 2.2, with the xx axis corresponding to the first throw, and the yy axis to the second throw. One should note that the total area of this square is still one and it contains all possible outcomes, so we can interpret the area as a probability. For example, the probability that a throw in area 1 is followed by a throw in area 3 is given by the top orange area, $p_1 \cdot p_3$. Of course, the probability of the opposite sequence is $p_3 \cdot p_1$, as represented by the bottom orange area. The total probability of hitting one and three in two consecutive throws is then:

$$2 \cdot p_1 \cdot p_3$$

In general, one can calculate the total probability of a series of events (one and three) by multiplying all the respective probabilities together **times** the total number of possible combinations of events (two in our simple example).

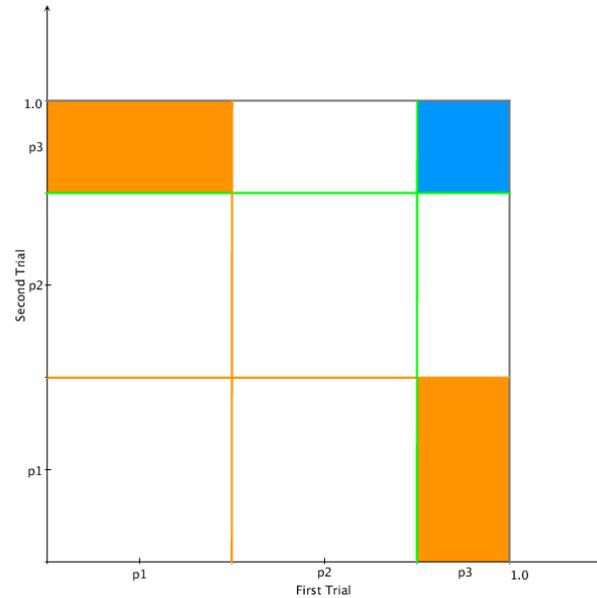


Figure 2.2: Probability of all possible outcomes after two trials.

2.2 Erdős-Rényi Model

Consider a simple set of N nodes. Between these nodes, one can create a maximum of $N(N-1)/2$ edges. If we assume that each possible edge is present in the system with probability p , then the average number of edges present in the system is:

$$E = \frac{p}{2} N(N-1)$$

From our definitions of last time, the average degree is:

$$\langle k \rangle = \frac{2E}{N} \approx Np$$

where we define λ . If we wish to build a network with a given average degree $\langle k \rangle$ and with N nodes, then the probability p should be:

$$p = \frac{\langle k \rangle}{N}$$

For a given node to have degree k , it must be chosen as the endpoint of an edge k and not be chosen exactly $N-k$ times. Since the order in which the edges are added is irrelevant we must multiply by the total number possible combinations to find the degree distribution:

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

as you would expect for a Binomial Process. Expanding the combinatorial factor, we obtain:

$$P(k) = \frac{(N-1)!}{k!(N-k-1)!} p^k (1-p)^{N-k-1}$$

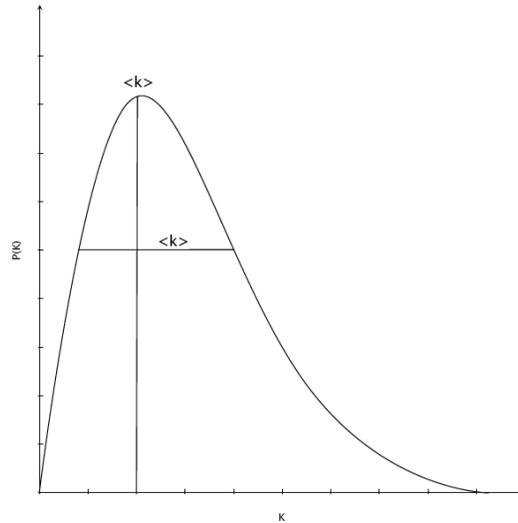


Figure 2.3: Poisson distribution.

It can be shown that:

$$\lim_{N \rightarrow \infty} \frac{n!}{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n} = 1$$

which is typically written as:

$$n! \approx \sqrt{2\pi n} n^n e^{-n}$$

and is known as “Stirling’s approximation” for the factorial function. Using this approximation, we rewrite the degree distribution as:

$$P(k) = \sqrt{\frac{2\pi N}{2\pi(N-k)}} \frac{N^N e^{-N}}{k! (N-k)^{N-k} e^{-N+k}} \left(\frac{\langle k \rangle}{N}\right)^k \left(1 - \frac{\langle k \rangle}{N}\right)^{N-k}$$

In the limit of large N (where Stirling’s approximation is valid), we have:

$$\lim_{N \rightarrow \infty} \sqrt{\frac{2\pi N}{2\pi(N-k)}} = 1$$

$$\lim_{N \rightarrow \infty} \left(1 - \frac{\langle k \rangle}{N}\right)^N = e^{-\langle k \rangle}$$

And after some algebraic manipulations, we find:

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

known as the “Poisson distribution” that we plot in Fig. 2.3.

This distribution is peaked around the average value, $\langle k \rangle$, with a variance also given by $\langle k \rangle$. This implies that any node will be unable to have a number of edges that is much larger than the average. For example, if $\langle k \rangle = 4$, the probability that a node has degree $10 \cdot \langle k \rangle = 40$ is 2.7×10^{-26} . In such a network no hubs (highly connected nodes) are possible.

In principle, one can build an Erdos-Renyi network with any given average degree. However, if the average degree is too small, we might not have enough edges to fully connect the network. In fact, there are three possible cases of interest:

- $\langle k \rangle < 1$, we simply don't have enough edges to connect the network. The largest connected component will have $O(\log N)$ nodes.
- $\langle k \rangle = 1$, we have a Giant Connected Component of size $N^{\frac{2}{3}}$ and several other smaller components
- $\langle k \rangle > 1$, there is a GCC that takes over the overwhelming majority of the nodes.

2.3 Exponential Growing Network

As a simple variation on the ER network, let us consider an Exponential Growing Network. Our model of growth is a simple one: at each time step t , we add a new node and randomly connect it to one of the previously existing ones. If we label each node by the time s at which it joined the network, then the probability $p(k, s, t)$ that node s has degree k at time $t + 1$ is given by:

$$p(k, s, t + 1) = \frac{1}{t} p(k - 1, s, t) + \left(1 - \frac{1}{t}\right) p(k, s, t) \quad (2.1)$$

since there are only two possibilities: Either node s has degree $k - 1$ at time t and it receives a new edge, or it has degree k and keeps the same degree. It is easy to see that if we can solve this recursing equation, then we can calculate the total degree distribution by taking the average over all nodes of degree k :

$$P(k, t) = \frac{1}{t} \sum_{s=1}^t p(k, s, t) \quad (2.2)$$

However, we can cheat and use this relation immediately to try to simplify our recursion. Summing both sides of Eq. 2.1 over all values of s , we find:

$$\sum_s p(k, s, t + 1) = \frac{1}{t} \sum_s p(k - 1, s, t) + \left(1 - \frac{1}{t}\right) \sum_s p(k, s, t)$$

Now we observe that we have run into a small problem. We are summing over all values of s at time t , but are trying to define the function at $t + 1$. Expanding Eq. 2.2 for $t + 1$:

$$P(k, t + 1) = \frac{1}{t + 1} \sum_{s=1}^{t+1} p(k, s, t + 1) = \frac{1}{t + 1} \left[\sum_{s=1}^t p(k, s, t + 1) + p(k, t + 1, t + 1) \right]$$

Fortunately, we know what is the value of $p(k, t + 1, t + 1)$ from the very definition of our generative process. Each node that joins the network has degree 1, so node $t + 1$, that joins the network at time $t + 1$ and hasn't yet had the possibility of receiving any other edges must have degree one. In other words:

$$p(k, t + 1, t + 1) = \delta_{k,1}$$

where $\delta_{k,1}$ represent the Kroenecker delta that is 1 when $k = 1$ and 0 otherwise. Using these definitions, we can now write:

$$(t + 1) P(k, t + 1) - \delta_{k,1} = \frac{1}{t} t P(k - 1, t) + \left(1 - \frac{1}{t}\right) t P(k, t)$$

Expanding:

$$(t + 1) P(k, t + 1) - t P(k, t) = P(k - 1, t) - P(k, t) + \delta_{k,1}$$

Since we are interested in the stationary distribution we can ignore the time dependency of $P(k, t)$. In particular:

$$P(k, t + 1) \equiv P(k, t) \equiv P(k)$$

Using this definition, we obtain:

$$(t + 1) P(k) - t P(k) = P(k - 1) - P(k) + \delta_{k,1}$$

and simplifying:

$$2P(k) = P(k - 1) + \delta_{k,1}$$

which already contains all the information we need to find the final form of the degree distribution $P(k)$. This expression tells us that as k increases, the probability $P(k)$ decreases by half. Or, in other words:

$$P(k) \propto 2^{-k}$$

2.4 Homework

Use the techniques exemplified above to calculate the degree distribution of a growing network where each incoming node connects to one other node. However, instead of choosing randomly among all previously existing nodes as in the case of the Exponential random graph, the new node prefers to connect to older nodes.