CHAPTER 13

RANDOM GRAPHS WITH GENERAL DEGREE DISTRIBUTIONS

This chapter describes more sophisticated random graph models that mimic networks with arbitrary degree distributions

IN THE previous chapter we looked at the classic random graph model, in which pairs of vertices are connected at random with uniform probabilities. Although this model has proved tremendously useful as a source of insight into the structure of networks, it also has, as described in Section 12.8, a number of serious shortcomings. Chief among these is its degree distribution, which follows the Poisson distribution and is quite different from the degree distributions seen in most real-world networks. In this chapter we show how we can create more sophisticated random graph models, which incorporate arbitrary degree distributions and yet are still exactly solvable for many of their properties in the limit of large network size.

The fundamental mathematical tool that we will use to derive the results of this chapter is the probability generating function. We have already seen in Section 12.6 one example of a generating function, which was useful in the calculation of the distribution of component sizes in the Poisson random graph. We begin this chapter with a more formal introduction to generating functions and to some of their properties which will be useful in later calculations. Readers interested in pursuing the mathematics of generating functions further may like to look at the book by Wilf [329].[1]

[1]Professor Wilf has generously made his book available for free in electronic form. You can download it from `www.math.upenn.edu/~wilf/DownldGF.html`.

13.1 GENERATING FUNCTIONS

Suppose we have a probability distribution for a non-negative integer variable, such that separate instances, occurrences, or draws of this variable are independent and have value k with probability p_k. A good example of such a distribution is the distribution of the degrees of randomly chosen vertices in a network. If the fraction of vertices in a network with degree k is p_k then p_k is also the probability that a randomly chosen vertex from the network will have degree k.

The *generating function* for the probability distribution p_k is the polynomial

$$g(z) = p_0 + p_1 z + p_2 z^2 + p_3 z^3 + \ldots = \sum_{k=0}^{\infty} p_k z^k. \tag{13.1}$$

Sometimes a function of this kind is called a *probability generating function* to distinguish it from another common type of function, the *exponential generating function*. We will not use exponential generating functions in this book, so for us all generating functions will be probability generating functions.

If we know the generating function for a probability distribution p_k then we can recover the values of p_k by differentiating:

$$p_k = \frac{1}{k!} \frac{\mathrm{d}^k g}{\mathrm{d}z^k}\bigg|_{z=0}. \tag{13.2}$$

Thus the generating function gives us complete information about the probability distribution and vice versa. The distribution and the generating function are really just two different representations of the same thing. As we will see, it is easier in many cases to work with the generating function than with the probability distribution and doing so leads to many useful new results about networks.

13.1.1 EXAMPLES

Right away let us look at some examples of generating functions. Suppose our variable k takes only the values 0, 1, 2, and 3, with probabilities p_0, p_1, p_2, and p_3, respectively, and no other values. In that case the corresponding generating function would take the form of a cubic polynomial:

$$g(z) = p_0 + p_1 z + p_2 z^2 + p_3 z^3. \tag{13.3}$$

For instance, if we had a network in which vertices of degree 0, 1, 2, and 3 occupied 40%, 30%, 20%, and 10% of the network respectively then

$$g(z) = 0.4 + 0.3\,z + 0.2\,z^2 + 0.1\,z^3. \tag{13.4}$$

As another example, suppose that k follows a Poisson distribution with mean c:

$$p_k = \mathrm{e}^{-c}\frac{c^k}{k!}. \tag{13.5}$$

Then the corresponding generating function would be

$$g(z) = \mathrm{e}^{-c}\sum_{k=0}^{\infty}\frac{(cz)^k}{k!} = \mathrm{e}^{c(z-1)}. \tag{13.6}$$

Alternatively, suppose that k follows an exponential distribution of the form

$$p_k = C\,\mathrm{e}^{-\lambda k}, \tag{13.7}$$

with $\lambda > 0$. The normalizing constant is fixed by the condition that $\sum_k p_k = 1$, which gives $C = 1 - \mathrm{e}^{-\lambda}$ and hence

$$p_k = (1-\mathrm{e}^{-\lambda})\,\mathrm{e}^{-\lambda k}. \tag{13.8}$$

Then

$$g(z) = (1-\mathrm{e}^{-\lambda})\sum_{k=0}^{\infty}(\mathrm{e}^{-\lambda}z)^k = \frac{\mathrm{e}^{\lambda}-1}{\mathrm{e}^{\lambda}-z}, \tag{13.9}$$

so long as $z < \mathrm{e}^{\lambda}$. (If $z \geq \mathrm{e}^{\lambda}$ the generating function diverges. Normally, however, we will be interested in generating functions only in the range $0 \leq z \leq 1$ so, given that $\lambda > 0$ and hence $\mathrm{e}^{\lambda} > 1$, the divergence at e^{λ} will not be a problem.)

13.1.2 POWER-LAW DISTRIBUTIONS

One special case of particular interest in the study of networks is the power-law distribution. As we saw in Section 8.4, a number of networks, including the World Wide Web, the Internet, and citation networks, have degree distributions that follow power laws quite closely and this turns out to have interesting consequences that set these networks apart from others. To create and solve models of these networks it will be important for us to be able to write down generating functions for power-law distributions.

There are various forms that are used to represent power laws in practice but the simplest choice, which we will use in many of our calculations, is the "pure" power law

$$p_k = C\,k^{-\alpha}, \tag{13.10}$$

for constant $\alpha > 0$. This expression cannot apply all the way down to $k = 0$, however, or it would diverge. So commonly one stops at $k = 1$. The normalization constant C can then be calculated from the condition that $\sum_k p_k = 1$,

which gives

$$C\sum_{k=1}^{\infty}k^{-\alpha} = 1. \tag{13.11}$$

The sum unfortunately cannot be performed in closed form. It is, however, a common enough sum that it has a name—it is called the *Riemann zeta function*, denoted $\zeta(\alpha)$:

$$\zeta(\alpha) = \sum_{k=1}^{\infty}k^{-\alpha}. \tag{13.12}$$

Thus we can write $C = 1/\zeta(\alpha)$ and

$$p_k = \begin{cases} 0 & \text{for } k = 0, \\ k^{-\alpha}/\zeta(\alpha) & \text{for } k \geq 1. \end{cases} \tag{13.13}$$

Although there is no closed-form expression for the zeta function, there exist good numerical methods for calculating its value accurately, and many programming languages and numerical software packages include functions to calculate it.

For this probability distribution the generating function is

$$g(z) = \frac{1}{\zeta(\alpha)}\sum_{k=1}^{\infty}k^{-\alpha}z^k. \tag{13.14}$$

Again the sum cannot be expressed in closed form, but again it has a name—it is called the *polylogarithm* of z and is denoted $\mathrm{Li}_{\alpha}(z)$:

$$\mathrm{Li}_{\alpha}(z) = \sum_{k=1}^{\infty}k^{-\alpha}z^k. \tag{13.15}$$

Thus we can write

$$g(z) = \frac{\mathrm{Li}_{\alpha}(z)}{\zeta(\alpha)}. \tag{13.16}$$

This is not completely satisfactory. We would certainly prefer a closed-form expression as in the case of the Poisson and exponential distributions of Eqs. (13.6) and (13.9). But we can live with it. Enough properties of the polylogarithm and zeta functions are known that we can carry out useful manipulations of the generating function. In particular, since derivatives of our generating functions will be important to us, we note the following useful relation:

$$\frac{\partial\,\mathrm{Li}_{\alpha}(z)}{\partial z} = \frac{\partial}{\partial z}\sum_{k=1}^{\infty}k^{-\alpha}z^k = \sum_{k=1}^{\infty}k^{-(\alpha-1)}z^{k-1} = \frac{\mathrm{Li}_{\alpha-1}(z)}{z}. \tag{13.17}$$

We should note also that in real-world networks the degree distribution does not usually follow a power law over its whole range—the distribution

is not a "pure" power law in the sense above. Instead, it typically obeys a power law reasonably closely for values of k above some minimum value $k_{\min}$ but below that point it has some other behavior. In this case the generating function will take the form

$$g(z) = Q_{k_{\min}-1}(z) + C \sum_{k=k_{\min}}^{\infty} k^{-\alpha} z^k, \tag{13.18}$$

where $Q_n(z) = \sum_{k=0}^{n} p_k z^k$ is a polynomial in z of degree n and C is a normalizing constant. The sum in Eq. (13.18) also has its own name: it is called the *Lerch transcendent*.[2] In the calculations in this book we will stick to the pure power law, since it illustrates nicely the interesting properties of power-law degree distributions and is relatively simple to deal with, but for serious modeling one might sometimes have to use the cut-off form, Eq. (13.18).

13.1.3 NORMALIZATION AND MOMENTS

Let us now look briefly at some of the properties of generating functions that will be useful to us. First of all, note that if we set $z = 1$ in the definition of the generating function, $g(z) = \sum_k p_k z^k$ (Eq. (13.1)), we get

$$g(1) = \sum_{k=0}^{\infty} p_k. \tag{13.19}$$

If the probability distribution is normalized to unity, $\sum_k p_k = 1$, as are all the examples above, then this immediately implies that

$$g(1) = 1. \tag{13.20}$$

For most of the generating functions we will look at, this will be true, but not all. As a counter-example, consider the generating function for the sizes of the small components in the Poisson random graph defined in Eq. (12.26). The probabilities π_s appearing in this generating function were the probabilities that a randomly chosen vertex belongs to a small component of size s. If we are in the regime where there is a giant component in the network then not all vertices belong to a small component, and hence the probabilities π_s do not add up to one. In fact, their sum is equal to the fraction of vertices not in the giant component.

The derivative of the generating function $g(z)$ of Eq. (13.1) is

$$g'(z) = \sum_{k=0}^{\infty} k p_k z^{k-1}. \tag{13.21}$$

[2]No, really. I'm not making this up.

(We will use the primed notation $g'(z)$ for derivatives of generating functions extensively in this chapter, as it proves much less cumbersome than the more common notation $\mathrm{d}g/\mathrm{d}z$.)

If we set $z = 1$ in Eq. (13.21) we get

$$g'(1) = \sum_{k=0}^{\infty} k p_k = \langle k \rangle, \tag{13.22}$$

which is just the average value of k. Thus, for example, if p_k is a degree distribution, we can calculate the average degree directly from the generating function by differentiating. This is a very convenient trick. In many cases we will calculate a probability distribution of interest by calculating first its generating function. In principle, we can then extract the distribution itself by applying Eq. (13.2) and so derive any other quantities we want such as averages. But Eq. (13.22) shows us that we don't always have to do this. Some of the quantities we will be interested in can be calculated directly from the generating function without going through any intermediate steps.

In fact, this result generalizes to higher moments of the probability distribution as well. For instance, note that

$$z\frac{\mathrm{d}}{\mathrm{d}z}\left(z\frac{\mathrm{d}g}{\mathrm{d}z}\right) = \sum_{k=0}^{\infty} k^2 p_k z^k, \tag{13.23}$$

and hence, setting $z = 1$, we can write

$$\langle k^2 \rangle = \left[\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)^2 g(z)\right]_{z=1}. \tag{13.24}$$

It is not hard to show that this result generalizes to all higher moments as well:

$$\langle k^m \rangle = \left[\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)^m g(z)\right]_{z=1}. \tag{13.25}$$

This result can also be written as

$$\langle k^m \rangle = \left.\frac{\mathrm{d}^m g}{\mathrm{d}(\ln z)^m}\right|_{z=1}. \tag{13.26}$$

13.1.4 POWERS OF GENERATING FUNCTIONS

Perhaps the most useful property of generating functions—and the one that makes them important for the study of networks—is the following. Suppose we are given a distribution p_k with generating function $g(z)$. And suppose

we have m integers k_i, $i = 1 \ldots m$, which are independent random numbers drawn from this distribution. For instance, they could be the degrees of m randomly chosen vertices in a network with degree distribution p_k. Then the probability distribution of the sum $\sum_{i=1}^{m} k_i$ of those m integers has generating function $[g(z)]^m$. This is a very powerful result and it is worth taking a moment to see how it arises and what it means.

Given that our integers are independently drawn from the distribution p_k, the probability that they take a particular set of values $\{k_i\}$ is simply $\prod_i p_{k_i}$ and the probability π_s that the values drawn add up to a specific sum s is the sum of these probabilities over all sets $\{k_i\}$ that add up to s:

$$\pi_s = \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \delta\bigl(s, \textstyle\sum_i k_i\bigr) \prod_{i=1}^{m} p_{k_i}, \tag{13.27}$$

where $\delta(a,b)$ is the Kronecker delta. Then the generating function $h(z)$ for the distribution π_s is

$$\begin{aligned} h(z) &= \sum_{s=0}^{\infty} \pi_s z^s \\ &= \sum_{s=0}^{\infty} z^s \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \delta\bigl(s, \textstyle\sum_i k_i\bigr) \prod_{i=1}^{m} p_{k_i} \\ &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} z^{\sum_i k_i} \prod_{i=1}^{m} p_{k_i} \\ &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \prod_{i=1}^{m} p_{k_i} z^{k_i} = \biggl[\sum_{k=0}^{\infty} p_k z^k\biggr]^m \\ &= \bigl[g(z)\bigr]^m. \end{aligned} \tag{13.28}$$

Thus, for example, if we know the degree distribution of a network, it is a straightforward matter to calculate the probability distribution of the sum of the degrees of m randomly chosen vertices from that network. This will turn out to be important in the developments that follow.

13.2 THE CONFIGURATION MODEL

Let us turn now to the main topic of this chapter, the development of the theory of random graphs with general degree distributions.

We can turn the random graph of Chapter 12 into a much more flexible model for networks by modifying it so that the degrees of its vertices are no longer restricted to having a Poisson distribution, and in fact it is possible to

modify the model so as to give the network any degree distribution we please. Just as with the Poisson random graph, which can be defined in several slightly different ways, there is more than one way to define random graphs with general degree distributions. Here we describe two of them, which are roughly the equivalent of the $G(n,m)$ and $G(n,p)$ random graphs of Section 12.1.

The most widely studied of the generalized random graph models is the *configuration model*. The configuration model is actually a model of a random graph with a given degree *sequence*, rather than degree distribution. That is, the exact degree of each individual vertex in the network is fixed, rather than merely the probability distribution from which those degrees are chosen. This in turn fixes the number of edges in the network, since the number of edges is given by Eq. (6.21) to be $m = \frac{1}{2}\sum_i k_i$. Thus this model is in some ways analogous to $G(n,m)$, which also fixes the number of edges. (It is quite simple, however, to modify the model for cases where only the degree distribution is known and not the exact degree sequence. We describe how this is done at the end of this section.)

See Section 8.3 for a discussion of the distinction between degree sequences and degree distributions.

Suppose then that we specify the degree k_i that each vertex $i = 1 \ldots n$ in our network is to take. We can create a random network with these degrees as follows. We give each vertex i a total of k_i "stubs" of edges as depicted in Fig. 13.1. There are $\sum_i k_i = 2m$ stubs in total, where m is the total number of edges. Then we choose two of the stubs uniformly at random and we create an edge by connecting them to one another, as indicated by the dashed line in the figure. Then we choose another pair from the remaining $2m-2$ stubs, connect those, and so on until all the stubs are used up. The end result is a network in which every vertex has exactly the desired degree.

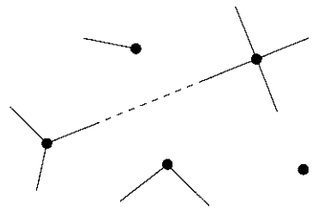

Figure 13.1: The configuration model. Each vertex is given a number of "stubs" of edges equal to its desired degree. Then pairs of stubs are chosen at random and connected together to form edges (dotted line).

More specifically the end result is a particular *matching* of the stubs, a particular set of pairings of stubs with other stubs. The process above generates each possible matching of stubs with equal probability. Technically the configuration model is defined as the ensemble in which each matching with the chosen degree sequence appears with the same probability (those with any other degree sequence having probability zero), and the process above is a process for drawing networks from the configuration model ensemble.

The uniform distribution over matchings in the configuration model has the important consequence that any stub in a configuration model network is equally likely to be connected to any other. This, as we will see, is the crucial property that makes the model solvable for many of its properties.