Complex Systems 535/Physics 508: Homework 7

*This homework is twice the length of a normal homework, but you have two weeks to complete it instead of just one. It is due in class on **Thursday, December 5**.*

1. Consider the growing network model of Price, as described in Chapter 14.

   (i) From the results given in the chapter write down an expression in terms of the parameters a and c for the expected in-degree of the ith vertex added to the network, just before the jth vertex is added, where $i < j$. You can assume that j is large.

   (ii) Hence show that the average probability of a directed edge from j to i in a network with n vertices, where $n \geq j$, is

$$P_{ij} = \frac{ca}{c+a}\, i^{-c/(c+a)} (j-1)^{-a/(c+a)}.$$

2. Consider a model of a growing directed network similar to Price's model described in Section 14.1, but without preferential attachment. That is, vertices are added one by one to the growing network and each has c outgoing edges, but those edges now attach to existing vertices uniformly at random, without regard for degrees or any other vertex properties.

   (i) Derive master equations, the equivalent of Eqs. (14.7) and (14.8), that govern the distribution of in-degrees q in the limit of large network size.

   (ii) Hence show that in the limit of large size the in-degrees have an exponential distribution $p_q = C\mathrm{e}^{-\lambda q}$ with $\lambda = \ln(1 + 1/c)$.

3. Consider a model network similar to the model of Barabási and Albert described in Section 14.2, in which undirected edges are added between vertices according to a preferential attachment rule, but suppose that the network does not grow—it starts off with a given number n of vertices and neither gains nor loses any vertices thereafter. In this model, starting with an initial network of n vertices and some specified arrangement of edges, we add at each step one undirected edge between two vertices, both of which are chosen at random in direct proportion to degree k. Let $p_k(m)$ be the fraction of vertices with degree k when the network has m edges in total.

   (i) Show that, when the network has m edges, the probability that the next edge added will attach to vertex i is k_i/m.

   (ii) Write down a master equation giving $p_k(m+1)$ in terms of $p_{k-1}(m)$ and $p_k(m)$. Give the equation for the special case of $k = 0$ also.

   (iii) Eliminate m from the master equation in favor of the mean degree $c = 2m/n$ and take the limit $n \to \infty$ with c held constant to show that $p_k(c)$ satisfies the differential equation

$$c\frac{\mathrm{d}p_k}{\mathrm{d}c} = (k-1)p_{k-1} - kp_k.$$

   (iv) Define a generating function $g(c,z) = \sum_{k=0}^{\infty} p_k(c)\, z^k$ and show that it satisfies the partial differential equation

$$c\frac{\partial g}{\partial c} + z(1-z)\frac{\partial g}{\partial z} = 0.$$

(v) Show that $g(c,z) = f(c - c/z)$ is a solution of this differential equation, where $f(x)$ is any differentiable function of x.

(vi) The particular choice of f depends on the initial conditions on the network. Suppose the network starts off in a state where every vertex has degree one, which means $c = 1$ and $g(1,z) = z$. Find the function f that corresponds to this initial condition and hence find $g(c,z)$ for all values of c and z.

(vii) Show that, for this solution, the degree distribution as a function of c takes the form

$$p_k(c) = \frac{(c-1)^{k-1}}{c^k},$$

except for $k = 0$, for which $p_0(c) = 0$ for all c.

Note that the degree distributions in both this model and the model of question 2 decay exponentially in k, implying that neither preferential attachment nor network growth alone can account for a power-law degree distribution. One must have both growth and preferential attachment to get a power law.

4. Consider a configuration model network that has vertices of degree 1, 2, and 3 only, in fractions p_1, p_2, and p_3, respectively.

   (i) Find the value of the critical vertex occupation probability ϕ_c at which site percolation takes place on the network.

   (ii) Show that there is no giant cluster for any value of the occupation probability ϕ if $p_1 > 3p_3$. In terms of the structure of the network, why is this? And why does the result not depend on p_2?

   (iii) Find the size of the giant cluster as a function of ϕ. (Hint: you may find it useful to remember that $u = 1$ is always a solution of the equation $u = 1 - \phi + \phi g_1(u)$.)

5. Consider the spread of an SIR-type disease on a network in which some fraction of the individuals have been vaccinated against the disease. We can model this situation using a joint site/bond percolation model in which a fraction ϕ_s of the vertices are occupied to represent the vertices not vaccinated, and a fraction ϕ_b of the edges are occupied to represent the edges along which contact sufficient for disease transmission takes place.

   (i) Show that the fraction S of individuals infected in the limit of long time is given by the solution of the equations

   $$S = \phi_s[1 - g_0(u)], \qquad u = 1 - \phi_s\phi_b + \phi_s\phi_b g_1(u),$$

   where $g_0(z)$ and $g_1(z)$ are the generating functions for the degree distribution and excess degree distribution, as usual.

   (ii) Show that for a given probability of transmission ϕ_b the fraction of individuals that need to be vaccinated to prevent spread of the disease is $1 - 1/[\phi_b g_1'(1)]$.

6. Recall the "acquaintance immunization" process we discussed in class: instead of vaccinating random people, you choose random people and get them to nominate a friend, then you vaccinate the friend. Because your friends tend to be the popular people, this has the beneficial effect of vaccinating people with many contacts.

   Consider the acquaintance immunization process on the configuration model.

(i) With the configuration model, when you follow an edge to a friend, you arrive at vertex i with probability $k_i/2m$. Thus the probability that a person with degree k gets vaccinated in the acquaintance immunization scheme is proportional to k, or, in the notation used in the book, $1 - \phi_k = Ak$ for some constant A, where ϕ_k is the probability that a vertex of degree k is "occupied," i.e., not vaccinated. If the total fraction of unvaccinated individuals is $\bar{\phi}$ (which is equal to $\sum_k p_k \phi_k$), then find the value of the constant A in terms of $\bar{\phi}$ and hence show that

$$\phi_k = 1 - (1 - \bar{\phi}) \frac{k}{\langle k \rangle}.$$

(ii) Show that the generating function $f_0(z)$ defined in Eq. (16.32) is given by $f_0(z) = g_0(z) - (1 - \bar{\phi}) z g_1(z)$, where $g_0(z)$ and $g_1(z)$ are the generating functions for the degree distribution and excess degree distribution, as usual. Hence find an expression for the function $f_1(z)$ from Eq. (16.36).

(iii) Using Eq. (16.41), show that in order to eradicate a disease on a configuration model network using acquaintance immunization, the fraction $1 - \bar{\phi}$ of vaccinated individuals must satisfy

$$1 - \bar{\phi} > \frac{\langle k^2 \rangle \langle k \rangle - 2\langle k \rangle^2}{\langle k^3 \rangle - \langle k^2 \rangle}.$$

This is one of the very few examples I know of in which the third moment of the degree distribution enters a calculation.

7. **Extra credit:** Write a computer program in the language of your choice to simulate the model of Barabási and Albert and calculate its degree distribution using the fast algorithm described in class. (You won't need to store the actual edges of the network itself—only the degrees are needed to answer this question.) Use your program to generate the degree sequence for a Barabási–Albert network of ten million vertices with $c = 2$ and hence make a plot similar to Fig. 14.4 in the book of the average degree of a vertex as a function of time τ, where $\tau = i/n$ for the ith vertex. (You could do this for instance by calculating an average in a sliding window, or simply by dividing the vertices into groups of, say, 1000 each and averaging the degrees within each group.)

   Turn in your plot and a copy of your program to get full credit. Extra points may be awarded for particularly elegant programs. (Hint: If you do it right, it should not be a complicated program. In my own program to solve the problem, which is written in C, the main loop that does the actual simulation consists of five lines of code and the program only takes about two seconds to generate a network of ten million vertices.)