

Lecture 3 (October 17, 2005)

prepared by Paul Constantine

1. Polya's Urn and Preferential Attachment

Let's review the connection between Polya's urn and the Barabasi-Albert preferential attachment model. Recall that Polya's urn begins with r red balls and b blue balls. At each step, choose one ball from the urn and replace it with a ball of the same color plus k copies. As the steps go to infinity, and the distribution of the balls goes to a beta distribution.

$$P(\text{ratio of red balls to the total number of balls} = x) = \frac{(b+r)!}{(b-1)!(r-1)!} x^{r-1} (1-x)^{b-1} \frac{1}{n}$$

In the preferential attachment model, we study the ratio of the degree of the k th vertex over the sum of the degrees of vertices $1, \dots, k$, where by " k th vertex", we mean the vertex that arrived at time k . This ratio converges to a beta distribution $\beta(1, 2k-1)$. Given the distribution of the ratios, we can get the probability of degrees of each vertex.

Define $\phi(k) = \beta(1, 2k-1)$. Divide the interval $[0, 1]$ into n pieces where the length of the k th interval is

$$\psi(k) = \phi(k) \prod_{j=k+1}^n (1 - \phi(j)).$$

One can show that this quantity is the limit of the distribution of the degree of the k th vertex over the sum of the degrees of vertices $1, \dots, k$. See Noam Berger, Christian Borgs, Jennifer T. Chayes, Amin Saberi: *On the spread of viruses on the internet*. SODA 2005: 301-310.

There is an equivalent way to study this ratio much more suited to analysis. The former point of view is chalked full of dependencies that can be difficult to deal with, while the following approach removes most of those pesky dependencies. Consider splitting the interval from 0 to 1 into n disjoint intervals, where the length of interval k is $\psi(k)$. Take a walk along the subintervals beginning at 0. Upon reaching the k th subinterval, draw a line from the k th subinterval backwards into the interval 0 to $\sum_{i=1}^{k-1} \psi(i)$. If this line lands in the j th subinterval, then we connect vertex k to vertex j . Using this method, it is easier to argue that the diameter of the large connected component is $\mathcal{O}(\log n)$.

Problem 3.1 Use definition of ψ to prove that distribution of degrees in the preferential attachment model is a power law.

Problem 3.2 The probability of choosing a red ball from Polya's urn at the first step is $\frac{r}{r+b}$ where r is the number of red balls and b is the number of blue balls. Prove that the probability of choosing a red ball at the k th step is also $\frac{r}{r+b}$.

2. Modeling the Internet

Suppose you are talking to a person trying to model the Internet graph. What tools will they use to generate this graph? There are three well known applications for this type of simulation:

- **GT-ITM:** <http://www.cc.gatech.edu/projects/gtitm/>
- **Inet:** <http://topology.eecs.umich.edu/inet/>
- **BRITE:** <http://www.cs.bu.edu/brite/>

Inet is based on a simpler idea than preferential attachment. It first generates a degree sequence d_1, \dots, d_n according to a power law. Then it connects vertex i to vertex j with probability proportional to $d_i d_j$. But this brings us to an interesting topic. Suppose you are given a degree sequence d_1, \dots, d_n . How might you generate a graphs uniformly at random from all possible graphs with that degree sequence? Of course there are many more sequences of integers than there are graphs, i.e. given a degree sequence, there may be no graphs with such a sequence. For example, the sum of the degrees in any graph is even, so the sequence 1, 1, 3 would not generate a graph. Also no graph exists for the sequence 50, 1, 1; the maximum degree in any sequence must be less than or equal to $n - 1$. Fortunately there are nice theorems (Erdos-Gallai and Havel-Hakimi) that give characterizations for degree sequences that can generate graphs. But generating one of those graphs uniformly at random is still an open problem.

Theorem 3.1 (Havel-Hakimi) *Given a degree sequence $d_1 \geq d_2 \geq \dots \geq d_n$, apply the following procedure. Connect the vertex with the highest degree to its neighbors of highest degree. This creates a new degree sequence by reducing d_1 and d_2 by one. Repeat this procedure on the new sequence. If you get stuck during this algorithm, then no graph exists with the original degree sequence.*

Generating graphs uniformly at random with a given degree sequence is sharp P hard. But can we find approximate algorithms that get close to accomplishing this? How do the network simulators do it? Actually, they do something less ambitious. Given a degree distribution, they find a graph whose degree distribution is approximate.

There are two methods for generating graphs with an expected degree distribution. One is a generalization of $G(n, p)$: Given n vertices, let the expected degree of vertex i be d_i , where

$1 \leq d_i \leq \sqrt{n}$. Put an edge between vertex i and j with probability

$$\frac{d_i d_j}{\sum_{k=1}^n d_k}$$

These graphs are studied by Molloy and Reed, and Newman, Chung, and Lu. Another method involves grouping d_1 mini-vertices, d_2 mini-vertices, up through d_n mini-vertices. Then simply do a random matching between the groups of mini-vertices and collapse the mini-vertices into single vertices.

These models are not as interesting mathematically, but they are used a lot in many application areas. Let's examine the branching process in this graph. We want to answer questions like: (i) is there a giant connected component? (ii) is the graph connected, or (iii) is there a phase transition? In other words to these models have the characteristics that we observe in the Internet graphs?

Given the degree sequence d_1, \dots, d_n , let p_k be the ratio of vertices of degree k to the total number of vertices n . How many children does a vertex v with degree k have?

$$\sum k p_k \equiv \langle k \rangle$$

Now examine a child u of the first branching of v . What is the expected number of children of u ? We know that the probability of choosing a vertex with degree k in the first branching is

$$\frac{k p_k}{\sum_{j=1}^n j p_j}$$

Let q_k be the probability that vertex u has k children. Then

$$q_{k+1} = \frac{k p_k}{\sum_j j p_j}$$

Create a sequence of random variables:

$$\begin{aligned} T_1 &= \langle k \rangle \\ T_2 &= \langle k \rangle \sum k q_k \\ &= \langle k^2 \rangle - \langle k \rangle \\ &= \sum_{k=1}^n (k^2 - k) p_k \\ T_3 &= T_2 \sum k q_k \\ \vdots &= \\ T_k &= \begin{pmatrix} z_2 \\ z_1 \end{pmatrix}^{k-1} z_1 \end{aligned}$$

where $z_1 = T_1, z_2 = T_2$.

This sequence describes the behavior of the graph at the phase transition. The process survives if $z_2 > z_1$ which implies

$$\sum_k (k^2 - k) p_k > \sum k p_k$$

or

$$\sum_k (k^2 - 2k)p_k > 0$$

This is the necessary and sufficient condition for a random graph with the given degree distribution to have a giant connected component.

To wrap up this section on modelling, we note that there are other models commonly used to examine properties of the Internet graph. For example, a *copying model* is a model where every new page is simply a perturbed copy of an existing page. These models have clustering properties similar to the Internet graph. For more on these models see Durrett's book, chapter 4. However there are still some characteristics that none of these models describe, which means there are plenty of available topics for your projects.