Graphs and Networks Lecture 15

The Giant Component

Lecturer: Daniel A. Spielman October 26, 2006

15.1 Probability Technicalities

I'd like to begin this lecture by getting some technicalities out of the way. These technicalities will allow us to apply bounds for independent random variables to variables which are not evidently independent.

The key issue I'd like to address is how we sample a graph from the distribution $\mathcal{G}(n,p)$. The most obvious way is to create a collection of independent random variables $X_{i,j}$ for each $i < j$ such that

$$\mathrm{P}\left[X_{i,j} = 1\right] = p, \quad \text{and} \quad \mathrm{P}\left[X_{i,j} = 0\right] = 1 - p.$$

We then say that edge (i,j) appears in the graph if $X_{i,j} = 1$.

But, we will consider a fancier approach to sampling that yields the same distribution. In particular, we will create a collection of independent random variables $X_{i,j}$ for all $i \neq j$. This is twice as many variables as we need, so we will only use one from each pair $(X_{i,j}, X_{j,i})$. We will then choose pairs (i,j) in some order, choose only one of $X_{i,j}$ or $X_{j,i}$ to examine, and then use the variable we examined to decide whether or not edge (i,j) should appear in the graph.

One way of visualizing this process is to think of building the adjacency matrix one step at a time. Initially, we view the adjacency matrix as being filled with the symbol "?". At each step, we choose some (i,j) for which $A(i,j) =?$, examine one of $X_{i,j}$ or $X_{j,i}$, and set $A(i,j)$ to 0 or 1 accordingly.

Even if we choose which pair (i,j) to examine based on the variables examined previously, we still sample a graph from $\mathcal{G}(n,p)$. In particular, I assert that if we choose the first pair (i,j) to examine before examining any of the variables, and then choose each consecutive pair (i,j) without ever having looked at $X_{i,j}$ or $X_{j,i}$, then the procedure samples a graph from $\mathcal{G}(n,p)$. This should be obvious. If it is not, just observe that since we only ever look at one of $X_{i,j}$ or $X_{j,i}$, the algorithm would behave exactly the same if they were equal.

When we process vertex $\pi(i)$, we look at every j for which $A_{i,j} =?$, and set $A_{i,j} = A_{j,i} = X_{i,j}$. We do not change the entries if $A_{i,j} \in \{0,1\}$.

The actual procedure that we will consider is just slightly fancier. In this procedure, we will generate variables $X_{i,j}$ for all i and j, During the course of the procedure, we will generate an ordering on the vertices, which is represented by a permutation π on $\{1,\ldots,n\}$. We initially set $\pi(1) = 1$. At the ith step, we set $A_{\pi(i),x}$ for all x such that $A_{\pi(i),x} =?$. We do this by setting $A(\pi(i),x) = A(x,\pi(i)) = X_{i,x}$. We then put all such nodes x for which $X_{i,x} = 1$ into the ordering. That is, if we have defined $\pi(1),\ldots,\pi(k)$, we then define $\pi(k+1),\ldots,\pi(k+j)$ to be the j vertices for which $A(\pi(i), x)$ was ? but becomes 1. If at the ith step $\pi(i)$ has not yet been defined, we define $\pi(i)$ to be the least index y for which the yth row of A contains a "?". That is, the least y for which there is not yet a j such that $\pi(j) = y$.

The only essential difference between this procedure and the one we considered before is that the variables $X_{i,j}$ are used to determine edges from $\pi(i)$, rather than from i. But, as we only ever examine one of $X_{i,j}$ and $X_{j,i}$, this procedure still samples from $\mathcal{G}(n, p)$.

Let me point out that this procedure is both Breadth First Search, and the "asleep-active-retired" process that I defined last class. In this terminology, a vertex y is asleep if there is no j such that $\pi(j) = y$ (which is when the yth row of A contains a ? and no 1's). A vertex y is retired if $\pi(j) = y$, and we are on step $i > j$ (which is when no entry of the yth row of A is ?). A vertex y is active if there is an i for which $\pi(i) = y$, but we have not yet reached the ith step.

15.2 Before Percolation

I'll now briefly make rigorous the proof that if $p = (1 - \epsilon)/n$, then for

$$c_1 = \frac{6(1-\epsilon)}{\epsilon^2},$$

the probability that there is a component of size more than $c_1 \ln n$ is at most $1/n$. We do this by examining the component of vertex 1, following the procedure described above. The size of the component of vertex 1 is equal to one minus the least t for which $\pi(t)$ was not defined at the start of step t. Let Y_i be the number of edges introduced between the vertex $\pi(i)$ and asleep nodes at step i. This is the number of vertices added to the ordering at step i. Then, the size of the component of vertex 1 is equal to the least t such that

$$Y_1 + Y_2 + \cdots + Y_t < t.$$

Let $Y_i' = \sum_j X_{i,j}$. Then, $Y_i \leq Y_i'$. So,

$$\mathrm{P}\left[Y_1 + \cdots + Y_t \geq t\right] \leq \mathrm{P}\left[Y_1' + \cdots + Y_t' \geq t\right].$$

As $Y_1', Y_2', \ldots, Y_t'$ are independent, we may apply the bound from the previous class:

$$\mathrm{P}\left[Y_1' + \cdots + Y_t' \geq t\right]. \leq e^{\frac{-\epsilon^2 t}{3(1-\epsilon)}}$$

to prove the result.

Having proved a probability bound of $1/n^2$ for the component of vertex 1, we obtain an upper bound of $1/n$ by considering the component of every vertex.

15.3 Giant Component, first attempt

On the problem set, you will prove that in the infinite k-ary tree with edge probability $p = (1+\epsilon)/k$, there is a constant c_ϵ so that the probability that the origin is connected to a leaf is at least c_ϵ. Moreover, c_ϵ is independent of k.

From the relation between percolation and branching processes discussed last class, this implies that if $Y_1, Y_2, \ldots$ is a sequence of variables with distribution $B(k, (1+\epsilon)/k)$, then the probability that there is a t for which $Y_1 + \cdots + Y_t < t$ is at most $1 - c_\epsilon$.

We will use this now to show that if $p = (1+2\epsilon)/n$, then the probability that vertex 1 is in a component of size at least $\epsilon n/(1+2\epsilon)$ is at least c_ϵ. Let me begin with the intuitive argument. Let S_i be the number of nodes that are asleep at the beginning of step i. Then, Y_i is distributed according to $B(S_i, p)$, which is to say it is the sum of S_i 0/1-random variables with probability p of being 1. If $S_i \geq n - \epsilon n/(1+2\epsilon)$, then $p \geq (1+\epsilon)S_i$. So, we know that the probability that there is a $t \leq i$ for which $Y_1 + \cdots + Y_t < t$ is at most $1 - c_\epsilon$. On the other hand, once $S_i < n - \epsilon n/(1+2\epsilon)$, we have seen at least active variables $\epsilon n/(1+2\epsilon)$, which provides our giant component.

There are only two issues with making this argument complete rigorous:

(a) As S_i depends on $Y_1, \ldots, Y_{i-1}$, we cannot really say that Y_i is independent of $Y_1, \ldots, Y_{i-1}$, and

(b) there's also something fishy about running the arguement up to the point where $S_i < n - \epsilon n/(1+2\epsilon)$, when we don't know what that i will be in advance.

Both of these issues can be made to dissappear using tricks like those from the first section. For those who really care, I'll outline how. At step i, instead of assigning $A(\pi(i), x) = X_{i,x}$, we let $\rho(x)$ be the index of the xth smallest asleep node, and set $A(\pi(i), x) = X_{i,\rho(x)}$. So, the variables $X_{i,1}, \ldots, X_{i,S_i}$ are the variables that are used at this step.

We then set $Y_i' = X_{i,1} + \cdots + X_{i,n(1-\epsilon/(1+2\epsilon))}$. Let $\gamma = \epsilon/(1+2\epsilon)$. Then, each Y_i' has distirbution $B((1-\gamma)n, p)$, and they are all independent. So, we know that the probability there is an $t < \gamma n$ for which $Y_1' + \cdots + Y_t' < t$ is at most $1 - c_\epsilon$. On the other hand, for all t such that $S_t \geq n - \gamma n$, $Y_t \geq Y_t'$. So, if $Y_1' + \cdots + Y_t' \geq t$ for all $t < \gamma n$, then $Y_1 + \cdots + Y_i \geq i$ for all i such that that or $S_i \leq n - \gamma n$, which implies that there is a t such that $Y_1 + \cdots + Y_i \geq i$ for all $i < t$, and $Y_1 + \cdots + Y_t \geq \gamma n$, and so the component of 1 has size at least γn.

But, we've only proved this with probability c_ϵ.

15.4 The Giant Component

To prove that the Giant component actually does appear with high probability, we will have to consider components of verices other than the first. This is why we defined our "asleep-active-retired" process to activate a sleeping node when it gets stuck.

To prove that a giant component should appear, we will take a closer look again at the branching process, and show that it is very unlikely that the process dies out given that it reaches a large enough size.

That is, let $Y_1, Y_2, \ldots$ be a sequence of variables with distribution $B(k, (1+\epsilon)/k)$. Let $Z_t = Y_1 + \cdots + Y_t$. We want to show that, as t_0 grows:

$$\mathrm{P}\left[\exists t > t_0 : Z_t < t\right] \rightarrow 0.$$

To bound $\mathrm{P}\left[\exists t > t_0 : Z_t < t\right]$, we first bound

$$\mathrm{P}\left[Z_t < t\right].$$

As $\mathbf{E}\left[Z_t\right] = t(1+\epsilon)$, we can apply a Chernoff bound to show that

$$\mathrm{P}\left[Z_t < t\right] \leq e^{-\delta^2 t(1+\epsilon)/2},$$

where $\delta = \epsilon/(1+\epsilon)$, so

$$\mathrm{P}\left[Z_t < t\right] \leq e^{-\epsilon^2 t/(1+\epsilon)2} = \left(e^{-\epsilon^2/(1+\epsilon)2}\right)^t.$$

Now, setting $x = e^{-\epsilon^2/(1+\epsilon)2}$, we find that

$$\mathrm{P}\left[\exists t > t_0 : Z_t < t\right] \leq \sum_{t > t_0} x^t = \frac{x^{t_0}}{1-x}.$$

And, this goes to zero exponentially quickly at t_0 goes to infinity.

Returning to the graph case, this tells us that there is a constant c_2 so that the probability that the component of node 1 has size greater than $c_2 \ln n$ and less than γn is at most $1/n^2$. So, with probability at least $1 - 1/n$ the graph has no component with size between $c_2 \ln n$ and γn.

So, all we need to do now is rule out the possibility that all the components have size less than $c_2 \ln n$. Consider the first couple of components we discover. As long as the total number of vertices we have found so far is less than $\gamma n - c_2 \ln n$, we can apply the argument from the previous section to argue that the probability the component we find has size less than $c_2 \ln n$ is at most $1 - c_\epsilon$. So, provided that $k(c_2 \ln n) < \gamma n$, the probability that the first k components we find all have size at most $c_2 \ln n$ is at most $(1 - c_\epsilon)^k$, which goes to $1/n$ for $k = O(\ln n)$. So, for n sufficiently large, this argument tells us that the probability that we do not find a giant component is at most $1/n$.