

3.4 Hybrid Models

August 27, 2018

1 3.4 Hybrid Models

A hybrid model spans between uniform at random models and the preferential attachment model

1.1 Simple Hybrid

Fraction a uniformly at random, $1-a$ via searching neighborhoods of friends - What we're going to do is we're going to have one fraction formed uniformly at random(a) and then the rest($1-a$) fraction formed via meeting friends of friends, the second part is going to look like preferential attachment. So, that will give us an explanation for how preferential attachment might actually be working.

1.2 Meeting 'Friends of Friends'

- Find new nodes via others: Network-based search
- New node meets am nodes uniformly at random and directs links to them
- Meets $(1-a)m$ of their neighbors and attaches to them too
- The distribution of neighbors' nodes is not the same as the degree distribution - even with independent link formation
- A neighbor is more likely to be higher degree

1.3 Relation to Preferential Attachment:

- In a network with *half* degree k and half degree $2k$ individuals:
- randomly select a link and then a node on one end of it - $2/3$ chance that it has degree $2k$, $1/3$ chance that it has degree k
- Suppose in a network with half degree k individuals and half degree $3k$ individuals, you randomly select a link and then a node on one end of it. What is the probability that the node found has degree $3k$?
- Following the procedure described, the chance of finding a node with degree $3k$ is $3k/k = 3$ times of the chance of finding a node with degree k . Hence, the probability of finding a node with degree $3k$ is $3/(3+1) = 0.75$
- Randomly find a node, randomly pick one of the nodes it attached to(that's the sort of finding of friends of friends bit), chance of finding some node via the second part of this procedure is proportional to its degree: find it if find one of its neighbors...

1.4 Simple Hybrid

- Fraction a uniformly at random, $1-a$ via preferential attachment:

$$dd_i(t)/dt = am/t + (1-a)d_i(t)/2t$$

and starting condition: $d_i(i) = m$, and the solution of this differential equation is:

$$d_i(t) = (m + 2am/(1-a))(t/i)^{(1-a)/2} - 2am/(1-a)$$

What this does is tell us what an expected degree is, at any point in time for a given node.

1.5 Degree distribution

Nodes that have expected degree less than d at some time t are those i such that

$$(m + xam)(t/i)^{1/x} - xam < d \text{ where } x = 2/(1-a)$$

critical i is such that

$$i/t = [(m + xam)/(d + xam)]^x \text{ where } x = 2/(1-a)$$

$$F(d) = (t - i)/t$$

$$F(d) = 1 - ((m + amx)/(d + amx))^x \text{ where } x = 2/(1-a)$$

1.6 Spans Extremes

$$F(d) = 1 - ((m + amx)/(d + amx))^x \text{ where } x = 2/(1-a)$$

- As a near 1, so that all of the links are being formed uniformly at random, then this is nearly a negative exponential distribution.
- As a near 0, then the x just becomes a 2, and we're going to get our $1 - m/d$ raised to the 2 power the derivative, the density function, this looks exactly like preferential attachment

1.7 Problem Set 3

Consider the "Hybrid Model" described in lecture 3.4, with $m = 10$. Let us compare the cases of $a = 0.8$, and $a = 0.2$

At data $t = 10000$, consider a node born at data $i = 20$ and a node born at data $i = 9999$.

Recall that the approximation of the expected degree at time t of a node born at time i is:

$$d_i(t) = (m + 2am/(1-a))(t/i)^{(1-a)/2} - 2am/(1-a)$$

Which option makes the following statement correct?

At data $t = 10000$,

a node born at data $i = 9999$ has a _____ expected degree with $a = 0.8$ than with $a = 0.2$; and

a node born at data $i = 20$ has a _____ expected degree with $a = 0.8$ than with $a = 0.2$

[Note: After this problem set is complete, there are two optional quizzes that you may follow.

One will have you use some software to analyze real network data ('Quiz: Optional: Empirical Analysis of Network Data '), and the other will have you use R to estimate an ERGM, (Quiz: Optional: Using Statnet in R to Estimate an ERGM']

```

In [18]: function expected_degree(m, a, t, i)
          ed = (m + 2*a*m/(1-a)) * (t/i)^((1-a)/2) - 2*a*m/(1-a)
          return ed
        end

Out[18]: expected_degree (generic function with 1 method)

In [21]: [expected_degree(10, 0.8, 10000, 9999), expected_degree(10, 0.2, 10000, 9999)]

Out[21]: 2-element Array{Float64,1}:
          10.0009
          10.0006

In [22]: [expected_degree(10, 0.8, 10000, 20), expected_degree(10, 0.2, 10000, 20)]

Out[22]: 2-element Array{Float64,1}:
          87.5481
          175.169

```