MS&E 235: Lecture 14
Prof. Ashish Goel
May 21, 2009
Notes by Rio Goodman

# Lecture 14: Network Structure and Viral Growth

Moore's Law states that computer speed doubles every 18 months. This corresponds to about a 1% increase every week. The growth associated with internet phenomena does not evolve in the same controlled manner. Instead, popular online applications experience viral growth. In this lecture we will discuss the environments necessary for viral growth.

## Viral Growth

The first application to experience viral growth was Hotmail. The reason Hotmail was able to experience viral growth was that it was built on an existing social network. An internet user who received an email from a friend that said at the bottom "join Hotmail for free by clicking this link," found it very easy to sign up for their own free (not work- or school-affiliated) email.

In December 1995 Sabeer Bhatia and Jack Smith asked for venture capital funding for Hotmail. In July 1996, Hotmail launched, and in December 1997, Hotmail had 400 million subscribers.

In contrast, the first instant messengers did not experience viral growth. The main distinguishing difference is that there was no existing network for instant messaging, whereas there was already an email network.

As these examples suggest, virality typically happens on top of an existing network. There is also typically an interesting or useful feature that also serves as an advertisement or delivery mechanism.

Another example of an application going viral is YouTube. In fact, Google Video existed before YouTube and provided essentially the same service, but YouTube was able to grow virally because of MySpace. YouTube allowed users to embed video directly into their websites. When other users saw an embedded video, they wanted one on their own websites, and thus the application spread through MySpace. Since YouTube had the existing MySpace network, it was able to experience viral growth.

Other examples include Facebook, Facebook apps, Twitter and MySpace.

## Network Structure

Let us examine a network of $N$ users, where each user has $k$ randomly chosen friends. In other words, a user pair $(u, v)$ is "friends" with probability $p = \frac{k}{N}$. Assuming that a user can convert all his/her friends to a new service, there is a threshold value for $k$ over which the network can support viral growth. The threshold value is $k \approx ln(N)$.

To get an idea of why this is the threshold value, consider one node, $v$. The probability that $v$ is *not* connected to another node, $w$, is $(1-p)$. Therefore, the probability that $v$ is not connected to *any* other nodes is $(1-p)^{N-1}$ (or approximately $(1-p)^N$).

Now if we want to make the isolation probability less than $\frac{1}{N}$, we must have the relationship

$$(1-p)^N < \frac{1}{N}$$

$$((1-p)^{1/p})^{Np} < \frac{1}{N}$$

We can then use the limit

$$\lim_{x \to 0} (1-x)^{\frac{1}{x}} = \frac{1}{e}$$

And the assumption that $p << N$, to obtain

$$e^{-Np} < \frac{1}{N}$$

$$\Rightarrow p > \frac{ln(N)}{N}$$

While this is not a rigorous proof, it shows that there is high probability that the network is connected if

$$p > \frac{ln(N)}{N}$$

This type of network is known as a Bernoulli Random Graph or Erdos-Renyi Graph. This model is abstract and does not fit any real social network. The conclusions, however, still hold for many social networks in the real world.

The relationship between $k$ and $N$ is important because of the rate at which $k$ must grow as $N$ grows. If $N$ doubles, $ln(2N) = ln(N) + C$, so $k$ increases by a constant ($C = ln(2)$). If $N$ is squared, $ln(N^2) = 2ln(N)$, and $k$ doubles.

We can see why this sort of model will not accurately capture social networks by considering Twitter. Ashton Kucher has over 1,000,000 followers, while most people have 15-30. In fact, we have seen this sort of distribution before in this course when we studied the Long Tail.

We can formulate the creation of a social network in much the same way as we created a long-tailed market. In the Preferential Attachment Model, a new node arrives at each time step, and joins one existing node in the network with probability proportional to the existing node's degree. This can be thought of as the more people you know, the more friends you will get. This is a form of "the rich get richer." The degree of the new node will be 1, and the expected degree of the first node (the node with the most neighbors) will be $\sqrt{N}$. This is analogous to the argument from the long-tailed market.

In such a network, marketers seek out nodes of high degree to spread information quickly to a large portion of the network.

## Napster, Kazaa and Skype

Napster was the first peer to peer file sharing program. It maintained a centralized index of all the files being shared, and their locations. In many ways, this system made much more engineering sense than the Kazaa system, which instead stored indexes on super nodes throughout the Kazaa network (on users' computers). The reason for Kazaa using distributed indexing was questionable. They claimed this design was beneficial because no central index was being maintained and that there was no central point of failure. But since Kazaa was serving banner advertisements to all its users, this argument is not very convincing. The Kazaa system, rather than being a good engineering system, was designed to evade the law.

On the other hand, Skype is built on the same architecture as Kazaa (by the Kazaa creators), and Skype is a well-designed system. The difference is that the Skype super nodes serve a real purpose. Instead of every user connecting to the central Skype server, every user can connect to a nearby super node. And unlike connecting directly to other users, if users are behind firewalls, the connection can still be made since the recipient is initiating contact with the super node. When a call comes through, the super node simply pings the user. This works out well for Skype because they route all the traffic through super nodes, and only need to pay local telephone charges.

This is an example of a poorly designed system working extremely well for a different purpose.