How users join a social network?
— Studies with Initial Linkage Activation Tree (ILAT)

Fangjian Guo

February 14, 2013

Contents

Introduction

It is indeed a tough task to study how large social networks evolve with time, especially when both elements are changing — nodes and links. [1] Among others, preferential attachment, homophily and triadic relations are possible mechanisms for explaining the growth of nodes and links. However, it would be hard to construct a unified framework that merges all these hypotheses. And I feel it would be quite messy and even intractable to try to incorporate all the affecting factors.

Therefore, I just look into a smaller and simpler question — how users join the network, i.e. how nodes grow in the social network. To be precise, "joining" here means making the first friendship link with another user, rather than simply "registering", as a user may stay inactive after registration.

I believe this problem is not well understood for dynamic complex networks, especially for social networks. I feel Barabasi's hypothesis

[1] Our Xiaonei SNS data is made up of about 10.6M users and 208M links, with each link labeled with its creation time. A user joins the network in the sense that his/her initial link brings him/her to the network. No removal of either node (user) or link is allowed.

of "preferential attachment" is only part of the picture, even a small part. Intuitively, one can imagine that a user joins a SNS usually because at least one of his/her friends, family members or acquaintances is using it, rather than he/she just wants to build a connection with some "highly famous" person (a node with large degree). For example, studies on Facebook have shown that users largely employ SNS to learn more about people they meet offline, and are less likely to use the site to initiate new connections. [2]

Cascaded Activation

In this study, I propose the *Cascaded Activation* theory for explaining the process of users joining a social network. The picture is very intuitive: "activation" means a new user joins the network because one or more (but we are limited to one here) old users have invited or attracted him or her to the network. For example, after I join the network, some of my friends also join because of me, and then they will activate their friends for coming to the network, etc. Such a process runs in a tree-like recursive fashion, which is called "cascaded".

I also made a key assumption in the theory: the *initial linkage* of a user, i.e. the first link that a user makes, largely reflects the *source* of activation. That is to say, if someone has been the main reason for me to join the SNS, it would be very likely that I would make my initial linkage with him/her. This heuristic also matches the way a new user adds another as a friend on mainstream SNS. After a new user has registered, he/she has to send a "friend request" to someone, and the link will not be made until that person accepts the request. Hence, it is very unlikely that one makes the "initial linkage" with a user newer than oneself.

Initial Linkage Activation Tree (ILAT) is both an algorithm and a framework for analyzing and modeling the *Cascaded Activation* process for user growth with empirical network data. ILAT reveals both the logical and temporal interconnections among users in the process by extracting and organizing *initial linkages*.

Initial Linkage Activation Tree

Tree Formation Algorithm

The algorithm for constructing ILAT from temporally labeled network data is straightforward: for every node in the network, finds the target of its *initial linkage* and assigns it as the node's *parent* in the tree, which is schematically illustrated in Figure 1.

To be precise, given an undirected temporal network $G = \{V, E, t\}$, where for any link $e \in E$, $t(e)$ is its creation time, the algorithm works

[2] Lampe, Cliff, Nicole Ellison, and Charles Steinfield. A Face (book) in the crowd: Social searching vs. social browsing. *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work.* ACM, 2006.

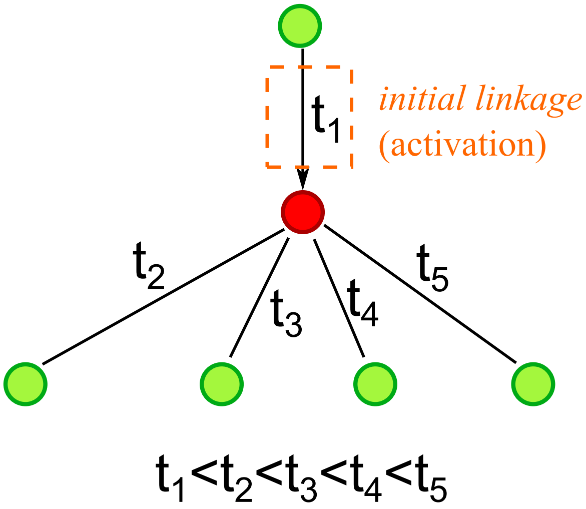

Figure 1: Initial linkage

in this way [3]:

```
Directed graph $\boldsymbol{T} = \{V, E_T, R_T\}$
Edge set $E_T \leftarrow \emptyset$
Root pair set $R_T \leftarrow \emptyset$
for $u \in V$ do
    $v = \arg\min_{v:e=(u,v)\in E} t(e)$ [4]
    add directed edge $(v \to u)$ into $E_T$
    if $(u \to v) \in E_T$ then
        add node pair $(u, v)$ to $R_T$
    end if
end for
```

The process extracts a directed subgraph from the original undirected network, as shown in Figure 2.

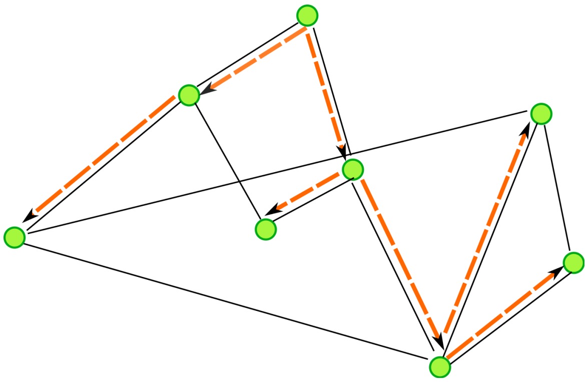

Figure 2: Tree as a directed subgraph

The output of the algorithm is a directed graph $\boldsymbol{T}$ recording the trees structure [5] and the root for each tree. As shall be seen, an ILAT tree is special in that the *root* of the tree is not a single node, but *a*

[3] While is not necessary to use directed links in a tree graph, we explicitly use *directed links* from a parent node to its child nodes to reflect the dynamics of activation.

[4] As our data is temporally labeled with *high resolution* (up to a second), there is almost always only link with earliest time stamp.

[5] $\boldsymbol{T}$ is usually a collection of trees, i.e. *a forest*, rather than a single tree.

pair of *ancestor nodes* that are mutually connected.

Tree Structure and Mathematical Properties

Why a Tree? Supposing we treat the pair of *ancestor nodes* as the root, we can see that T is a collection of trees by noticing the following properties:

1. Every node has one and only one parent node, identified by the initial linkage.
2. There is no cycle in the graph, except the length-2 cycle for root pairs.

The second property is less obvious. To see this, consider the case in Figure 3.

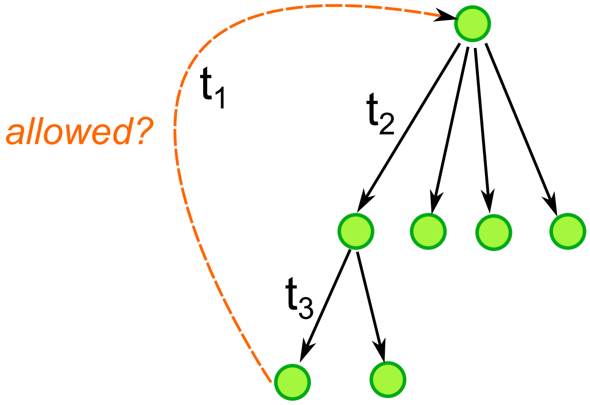

Figure 3: No cycle longer than 2 is allowed

If that cycle is allowed, then from the definition of *initial linkage*, we have $t_1 \leq t_2 \leq t_3$, but meanwhile $t_3 \leq t_1$, which is possible only $t_1 = t_2 = t_3$. [6] In other words, a cycle can happen only when all the links in the cycle occur at *precisely the same time*. However, as link creation is recorded with precision to 1 second, such occasion cannot occur in general. [7]

The complexity of the algorithm would be $O(\|E\|)$. Therefore, it is very efficient even for large networks.

Characterization of ILAT The *forest* is composed of a number of separate trees, denote as $\boldsymbol{T} = \{T_1, T_2, \cdots, T_n\}$. The structure and its corresponding structural concepts are schematically drawn in Figure 4.

By performing a *breadth-first traversal* on the tree, one can slice the tree into *levels*, for which we number the two *ancestor nodes* as level 1. The maximum level is called the tree's *depth*. A level can also be referred to as a *generation*. The number of nodes in one generation

[6] *Initial linkage* is pivotal in that it implies a natural *temporal ordering*.

[7] One can perform a breath-first traversal from root on a tree (taking care of the mutually connected root pairs). If the traversal terminates, then there is no cycle longer than 2. In our Xiaonei SNS data, no cycle longer than 2 is found.

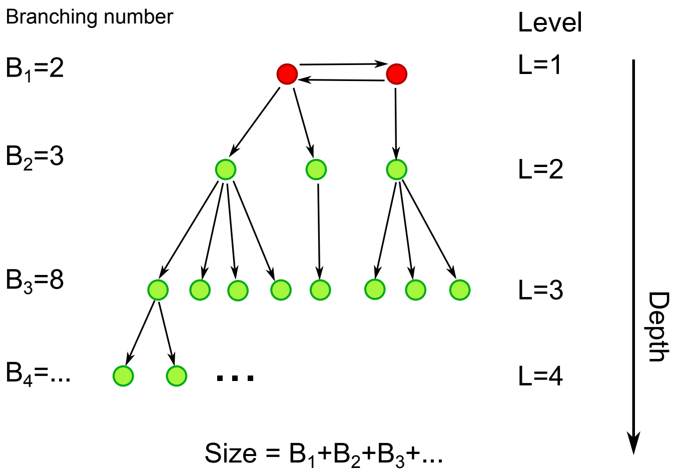

Figure 4: Structural anatomy of an ILAT tree

is called *branching number*, whereas B_i denotes the branching number in the *i*-th generation. The number of nodes in one single tree is called the tree's *size*, while the number of nodes in the *forest* is called *population*.

These concepts reflect the logical relations among nodes in the ILAT framework. Meanwhile, the temporal relations, which also describe the dynamics of tree growth, are characterized by the *birthtime* of each node. A node's birthtime is simply the creation time of its *initial linkage*.

Validating Cascaded Activation by Comparing with Null Model

While ILAT is designed to capture the structural dynamics of *Cascaded Activation* in the network growth, we have to make sure that the mechanism of *Cascaded Activation* really *exists* in empirical data before adopting the ILAT framework.

The caution is not unnecessary, as given a temporal network, the ILAT algorithm can *always* produce something. To make sure that the forest obtained by ILAT algorithm is really a meaningful structure, we have to validate the effects of *Cascaded Activation* in the network growth. This is done by comparing the output of ILAT from both empirical data with that from null-model data, which is the temporally randomized counterpart of the original data.

Null Model

There are several null models with different *orders* of randomization that can be used for comparison.

The "first order" null model would be a randomized network that destructs the temporal properties while preserving the topological properties.

The "second order" null model further randomizes both the time stamps and the wiring details, while preserving some basic structural statistics (e.g. degree sequence).

In this section, I would discuss the results obtained by comparing with the simplest "first order" null model, which is constructed by performing *random shuffling* of all time stamps.

Temporally Randomized Null Model The temporally randomized null model is produced with the original linkage data plus a randomly shuffled version of time stamps [8]. This model destroys the original *temporal order* necessary to *Cascaded Activation* (an newer node is activated by an older node).

Comparison and Results

The pictures of the ILAT *forest* from the original data and the null-model data are distinctively different. To summarize, in the original data:

1. The forest is composed of a small number of *big* trees and a large number of *small* trees.
2. *Most* of the population aggregate in those *big* trees.

On the contrary, in the randomized data, *trees cannot grow big*, making the *forest* simply a collection of a huge number of scattered small trees [9] . The details are given as follows.

Tree size distribution As shown in Figure 5, the big trees are magnitudes larger than those in the null model. The top-10 biggest trees in the original data are of sizes:
263485 182328 119724 78079 74427 67768 66645 64545 63049 56360,
while those for the null model are:
4179 922 564 465 305 271 261 233 231 229.

Tree depth distribution The trees in the original data are much deeper than those in null model, as shown in Figure 6.

Population distribution among trees Figure 7 reports the *Gini curve* for how population are distributed among trees of different sizes. As can be seen, 80% of the population "live in" the top 10.5% biggest trees in the real data, while population are distributed almost evenly in the null-model case.

Therefore, to conclude, *Cascaded Activation* is clearly validated as an evident mechanism in network growth from empirical data.

[8] One can imagine listing all the (u, v, t) tuples and randomly shuffling the third column.

[9] There are 23,121 trees from the real data, while 1,729,430 from null-model data.

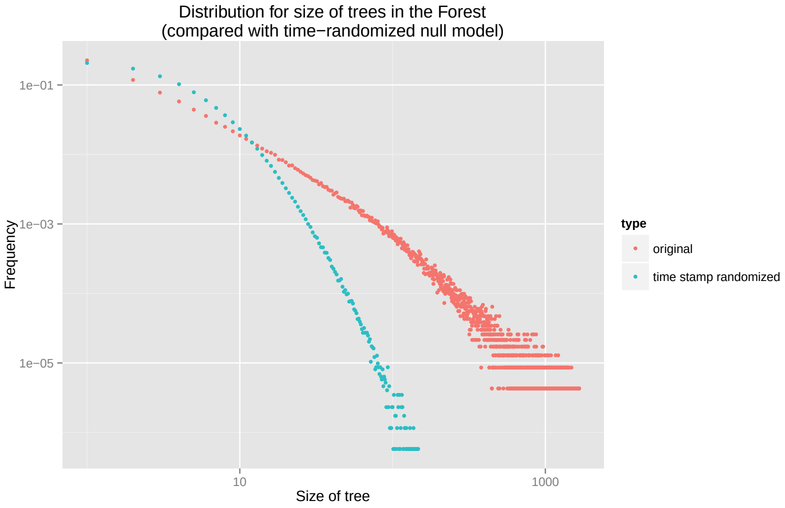

Figure 5: Tree size distribution in the forest: data and null-model

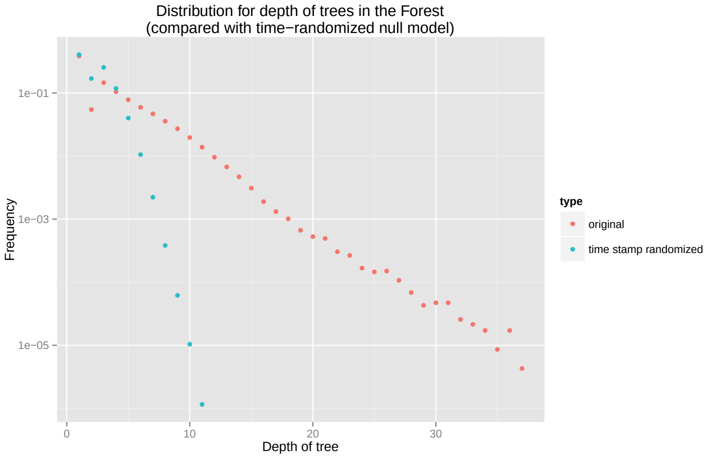

Figure 6: Tree depth distribution in the forest: data and null-model

Analysis of ILAT

Scales of Analysis

Due to the hierachical organization of ILAT structures, I suggest that the analysis of ILAT can be performed on three scales:

1. The forest scale: the distribution of trees and related quantities in the forest.
2. The tree scale: the dynamics of tree growth and how users are related and distributed among different generations.
3. The individual scale: the activation pattern of an user.

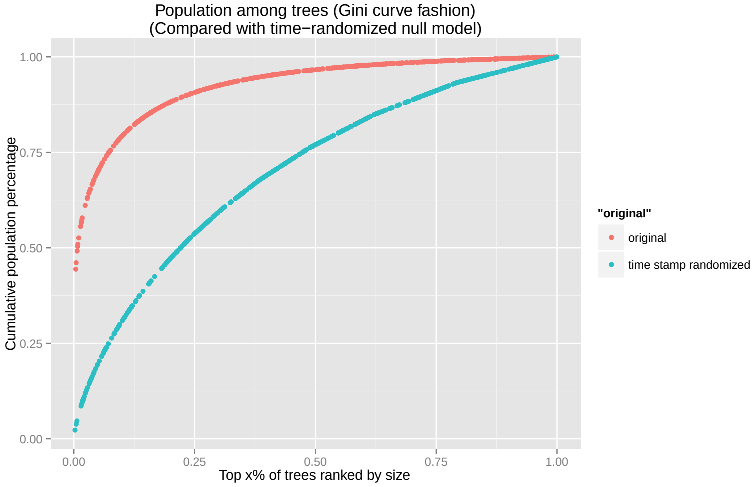

Figure 7: The distribution of population among trees: data and null-model

The Forest Scale

The distributions of tree size, depth and population have been reported in the last section.

Exponential Size ∼ Depth Relation As reported in Figure 8, an evident exponential relation between size and depth in trees is found. If assuming *very loosely* that every tree evolves with the same branching number, then by fitting the growth curve to an exponential function, I found that

$$\text{size} \sim 1.48^{\text{depth}}, \tag{1}$$

which gives a rough sketch that **on average, 1 user directly attracts 1.5 users to the network.**

Linear Growth of Trees Figures 9 illustrates how the size of a tree depends on the time it has been growing. Basically, the size it attained grows **linearly with the time required**. And clearly, there is **a maximal speed for tree growth, as indicated by the diagonal boundary**.

The Tree Scale: the Biggest Tree

I choose the biggest tree [10] in particular for a careful study.

Size of Each Generation Quite interestingly, as shown in Figure 10, the size vs. generation clearly lie on a bell-curve. This is due to the **trade-off between the size of the parent generation and the time required for growth**: for a generation to grow big, it needs a

[10] depth: 33, size: 263,485 (2.5% of all pupulation)

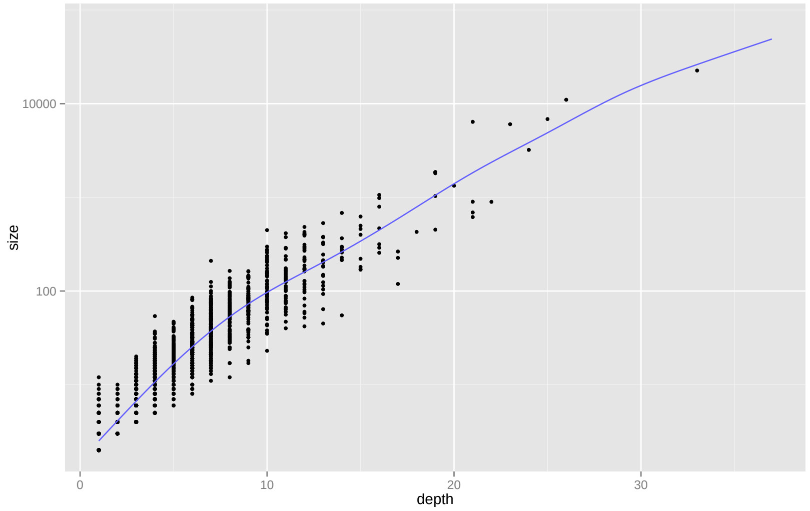

Figure 8: The exponential relation between size and depth in trees in the forest

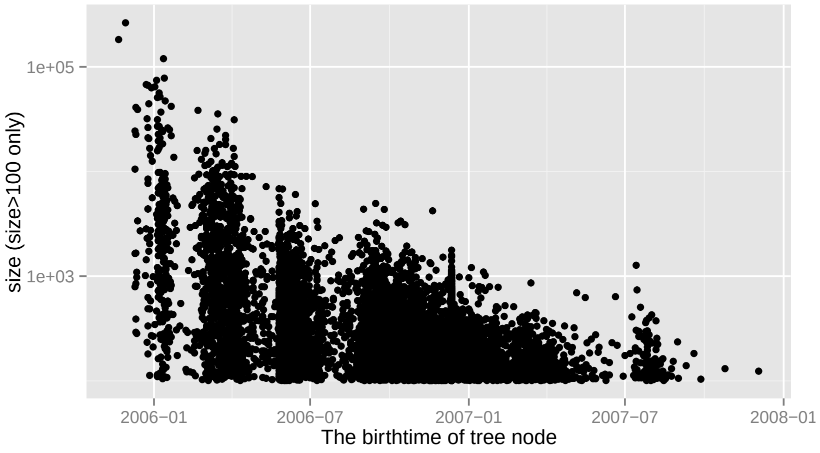

Figure 9: The size of tree plotted against the birthtime of root

large-sized parent generation, and meanwhile enough time for them to activate many children.

Accelerated Generation Activation As can been seen from Figure 11, it takes less and less time to activate a new generation, i.e. **the difference between "ages" in between two consecutive generations is decaying**.

Size of Ego-Trees As tree is itself a recursive structure, even if a node is not one of the upmost *ancestor nodes*, we can still treat it as the root of the ego-tree. An ego-tree of a node refers to the activation tree made up of itself plus all the *offspring* nodes that it directly or

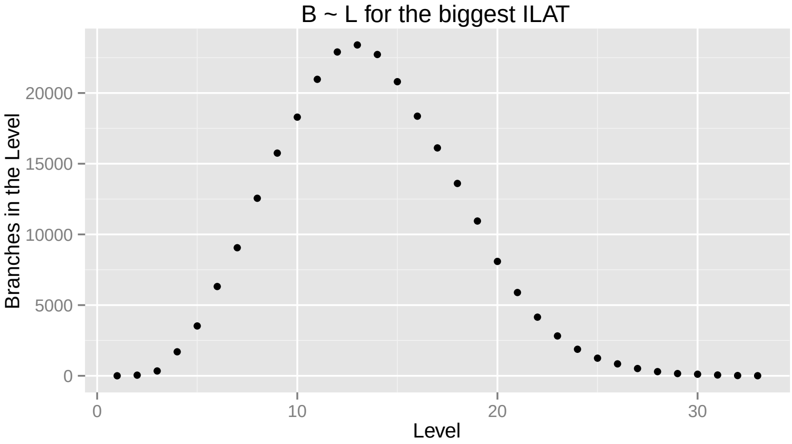

Figure 10: Number of nodes in each generation for the biggest tree

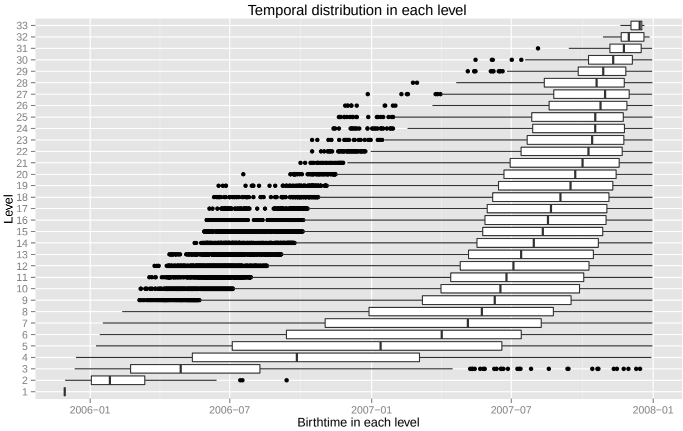

Figure 11: The boxplot for birthtime within each generation for the biggest tree

indirectly activated.

As reported in Figure 12, **the distribution for the size of all the ego-trees (corresponding to all the nodes) precisely follows a power-law with $\alpha = 2$.**

I feel it is more than an accident. For example, one can consider an arbitrary tree with any constant branching number bigger than 1, then the resulting distribution should still be a power-law, but with α being precisely 1.

So, where does $\alpha = 2$ come out? I guess it can be explained by assuming the *linear activation* of each node. [11] An ego-tree constructed by ILAT is not “fully-grown”, but still branching and growing instead. We have to take the dynamics into consideration.

[11] An intuition now, not tested yet.

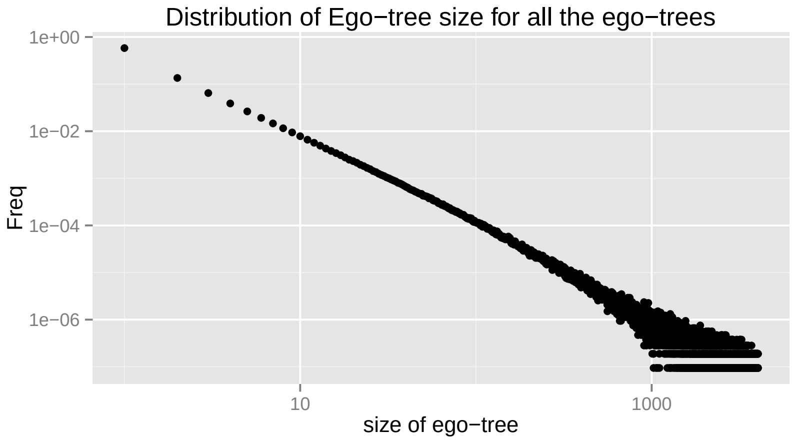

Figure 12: The distribution of the size of all the ego-trees in the forest

The Individual Scale

Two-Phase Linear Activation Pattern The activation pattern refers to how a user *activates* (directly) his/her child nodes temporally. The activation pattern of a user can be reflected with a plot, where every point corresponds to the an activation, with x-axis marking the time, y-axis counts the number activations in the user's history.

I examined the activation pattern for many users. I found that, quite surprisingly, **there distinctively exists a common "two-phase linear activation pattern"**, two typical cases of which are given by Figure 13 and 14. Most users seem to exhibit a two-phase pattern in activating child users, i.e. **a fast phase abruptly followed by a slow one**. In each phase (especially the fast phase), users make activations approximately linearly with time, i.e. **the time intervals between two consecutive activations fluctuate around a value with relatively small variance**.

A few observations:

1. The activation speed in the second phase is much slower than the first.
2. The turning point between the two phases is individual-specific. [12]
3. The speed of activation in both phases is individual-specific.
4. Some users do not have the slow-phase. [13]

Steps Ahead: Modeling User Growth with Branching Process

Based on the ILAT framework and aforementioned analysis, I am considering modeling the user growth with a *branching process*. However,

[12] I have not checked the distribution of turning point yet.

[13] Not coming yet?

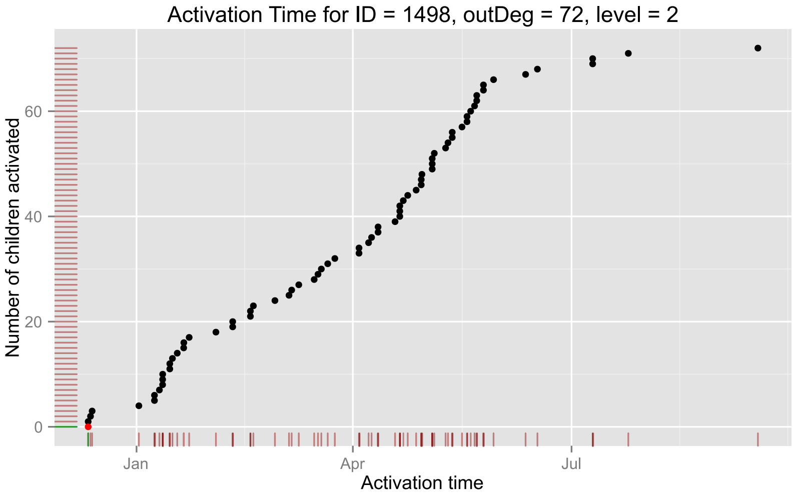

Figure 13: A typical "Two-Phase Linear Activation Pattern". The red dot marks the birthtime of the node itself

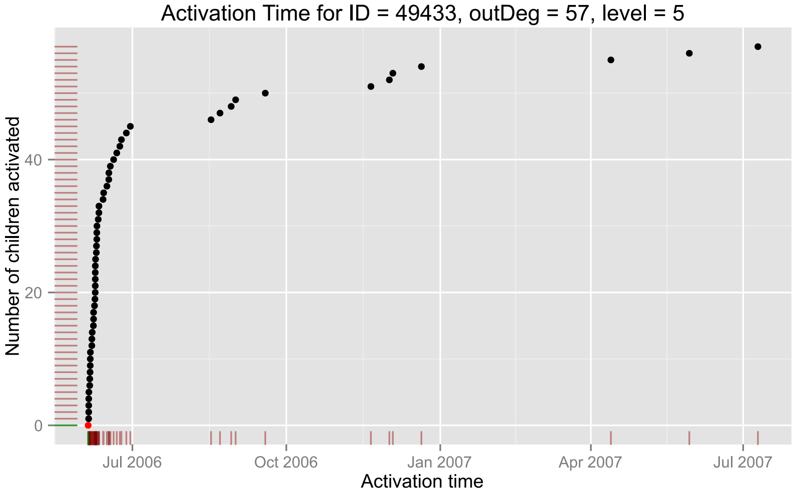

Figure 14: Another "Two-Phase Linear Activation Pattern"

it is definitely **more than a simple branching process**, as, in our theory, the *linear activation pattern* of each node must be considered. In other words, in our model, every generation would possibly continue to activate child nodes at any time, which is different from the occasion in branching process, where only one generation is "active" at each time step.