CS224W Final Report
Time-variant Information Spread in Social Networks

Jingning Ji
Electrical Engineering
Stanford University
jingning@stanford.edu

Yang Zhao
Electrical Engineering
Stanford University
yzhao12@stanford.edu

Yuchen Li
Computer Science
Stanford University
yuchenli@stanford.edu

1 Introduction

In this project, we have invested several papers related to the influence spreading through a social network, and decided to study the problem of influence spreading evolution in social network where the strength of influence may change over time. Our goal is to propose models that well simulates the evolution process of time-variant information diffusion, and also develop algorithm on model to search initial set that maximize influence. We have come up with 2 suitable models and we used those models to simulate the influence-spreading process on twitter and facebook data set.

The rest of this paper is organized as follows. Section 2 reviews some previous work. The testing data set is described in Section 3. The detailed analysis of two models are in Section 4 and 5. Section 6 describes possible future works.

2 Review of relevant work

In [1] 'The spread of behavior in an online social network experiment', the author compares different hypothesis of how behavior spreads in a social network. One hypothesis is that like disease, behavior can spread with just one contact of the infected node. The other hypothesis is different from the previous one, which states that spreads only happen when a node is contacting many other infected ones.

In [2] 'Contagion', the author focused on one specific question: when do we get contagion under deterministic best response dynamics in binary action games? Given a local interaction system describing how players interact and their payoff matrix, the author analyzed the contagion on several simple infinite network types.

Although the paper did a good job in math definition and derivation, the model is still too simple. In real world the response is not deterministic, and the network structure is not that simple as described in paper.

In [3] 'The role of social networks in information diffusion', the author did a large experiment at Facebook, to analyze information diffusion in social networks with regard of share time, multiple sharing friends, and tie strength. One contribution of the paper is that it validates Mark Granovetter's predictions about the strength of weak ties. The intuitive reason is that strong ties are redundant in spreading as close friends usually share same interests, while weak ties have more diverse information. Another interesting concept shown in the paper is homophily, which is the tendency of individuals with similar characteristics to associate with one another. Therefore we can't infer whether one's activity is resulting from homophily or influence of friends. Most models don't consider this, which leaves a large space for developing new realistic models.

The models used in our project was mainly based on the models described in [4] 'Maximizing the Spread of Influence through a Social Network'

In the paper, the authors consider social network as a medium for the spread of information, ideas, and influence among its members. Diffusion models are used to simulate the dynamics of spread of information in the social network. In considering diffusion models for the spread of an idea or innovation through a social network G, represented by a directed graph, each individual node is either active (an adopter of the innovation) or inactive. This paper focuses on settings where each nodes tendency to become active increases monotonically as more of its neighbors become active. Two basic diffusion models are discussed in this paper: Linear Threshold Model and Independent Cascade Model.

In Linear Threshold Model, a node v is influenced by each neighbor w according to a weight $b_{v,w}$ such that $\sum_{\text{w, neighbour of v}} b_{v,w} \leq 1$. Each node v chooses a threshold θv uniformly at random from the interval [0, 1]; this represents the weighted fraction of v's neighbors that must become active in order for v to become active. Given a random choice of thresholds, and an initial set of active nodes A_0 (with all other nodes inactive), the diffusion process unfolds deterministically in discrete steps: in step t, all nodes that were active in step $t-1$ remain active, and we activate any node v for which the total weight of its active neighbors is at least θv.

The Independent Cascade Model also starts with an initial set of active nodes A_0, and the process unfolds in discrete steps according to the following randomized rule: when node v first becomes active in step t, it is given a single chance to activate each currently inactive neighbor w; it succeeds with a probability $p_{v,w}$ a parameter of the system independently of the history thus far. If v succeeds, then w will become active in

step $t+1$; but whether or not v succeeds, it cannot make any further attempts to activate w in subsequent rounds.

3 Data

3.1 Data source and preprocessing

The data set we are using are the social network data set from the cs224w resource website. In particular, we used Facebook and Twitter data sets from the social networks section and combined all the ego networks into a large data set for both Twitter and Facebook data set. The Twitter data set has 81306 nodes and 1768149 edges; the Facebook data set has 3963 nodes and 176312 edges. In the following section, we are providing the visualization and statistics summary of the data set we have used.

3.2 Statistics summary

Figure 1 shows the visualization of the Facebook data. We used this data set primarily in our Linear Threshold Model. Figure 2 shows the degree distribution of the Facebook network. We don't provide the visualization or degree distribution of the Twitter network because its size is too large.

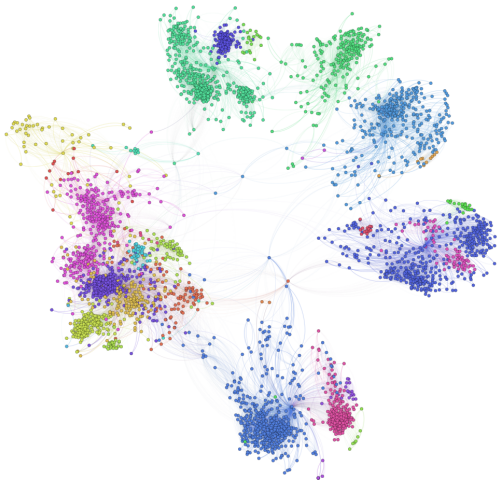

Figure 1: Visualization of Facebook data

4 Probabilistic Model for diffusion evolution

4.1 Model Description

The model considers adoption and recovery of a person in the diffusion process. A more formal description is shown below.

1. Initially there are K people adopted.
2. Every time, An adopted person has probability q to recover.

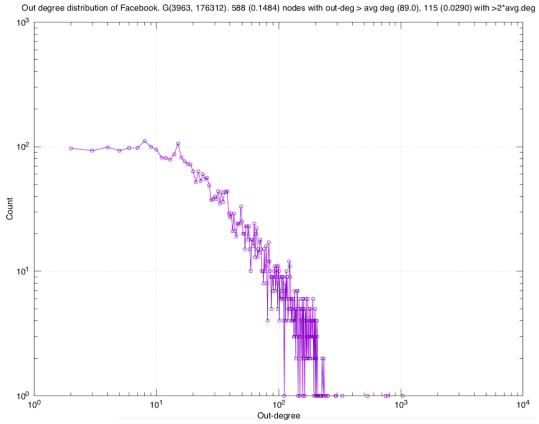

Figure 2: Out degree distribution of Facebook data

3. Every time, an adopted person has probability p to infect his neighbor.
4. All people who has adopted won't adopt again.

4.2 Mathematical Analysis on ER network

If a person is adopted, then the time of remaining adopting is distributed exponentially. $Pr(\tau = \tau_0) = (1-q)^{\tau_0 - 1}q, \tau_0 > 0$ The expected time $E[\tau] = \frac{1}{q}$. The probability of a person not infecting his neighbor is $Pr(!infect) = \sum_{\tau=1}^{\infty}(1-q)^{\tau-1}q(1-p)^{\tau} = 1 - \frac{p}{p+q-pq}$. Therefore the probability of a person infecting his neighbor is $Pr(infect) = \frac{p}{p+q-pq}$. Therefore, this model can be regarded as a basic independent cascade model with infecting probability $\frac{p}{p+q-pq}$, without considering time. If the diffusion want to spread widely, we should have $Pr(infect)d > 1$, i.e. $pd > p+q-pq$ where d is the average degree.

Then we derive the number of infected nodes by time from another perspective. Denote the number of nodes in graph by N, the probability of edge existing between two nodes by $s = d/N$, the expected number of adopted nodes at time t by S_t, the expected number of nodes having adopted at time t by H_t. Then at time t-1, the number of nodes that havent adopted is $N - H_{t-1}$. For one node that has not adopted and one node that is currently adopted, theres probability s that an edge exists and probability p that its infected given the edge. Therefore, the probability of one node infected by a specific other node is ps. Because there are S_t nodes that is adopted, the probability of an uninfected node to be infected at time t-1 is

$$Pr(IF_{t-1}) = 1 - (1-ps)^{S_{t-1}} \approx 1 - e^{-psS_{t-1}} \quad (1)$$

Therefore, the expected number of new infected node in t is $(N - H_{t-1})(1 - e^{-psS_{t-1}})$. We have the following equations

$$H_t = H_{t-1} + (N - H_{t-1})(1 - e^{-psS_{t-1}}) \quad (2)$$

$$S_t = (1-q)(S_{t-1} + (N - H_{t-1})(1 - e^{-psS_{t-1}})) \quad (3)$$

From this equation we can also estimate a condition for diffusion to spread. When H_{t-1} and S_{t-1} are very small, we have

$$S_t \approx (1-q)(S_{t-1}+NpsS_{t-1}) = (1-q)(1+pd)S_{t-1} \tag{4}$$

To let information spread, we need $S_t > S_{t-1}$, therefore $(1-q)(1+pd) > 1$, i.e. $pd > q/(1-q)$. This turns out to be too optimistic. In simulation, we need larger average degree to see the diffusion happens.

4.3 Simulation

First we simulated on a random generated directed ER graph, with N=236, E= 2478, p=0.2, q=0.2. For simplicity, we randomly infect one node in initialization, and set $S_0 = 1, H_0 = 1$. The simulation was run for 100 times and the average was taken. Then we simulated on a real network. The twitter network described before was chosen. The N,E,p,q were exactly the same. The theoretical curve on ER was also computed for comparison. The result is shown in Figure 3.

It can be seen that on ER graph, the theory fits pretty well with simulation result, but still overestimate the infected number. This is because we implicitly assume that the existence of edge is independent of whether the node is infected this turn, but actually they are dependent. This effect becomes larger when p is smaller than q.

Another finding is that the infected number on twitter data is less than that on ER graph. Notice the degree distribution of twitter data is more like power law distribution, and this is the main difference of two graphs. The intuition is that in a social network, there are more people having very few friends. If we initialize on them, then the diffusion will probably end quickly, and making average infected number to decrease.

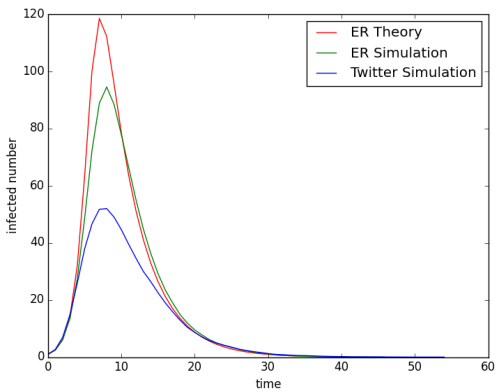

Figure 3: Infected number on ER Network and Twitter Network

4.4 Influence Maximization

4.4.1 Objective

The first thing we want to talk is what should we maximize. One objective is to maximize the number of nodes that has been infected. i.e. given an initial set I_0, the objective function is

$$f(I_0) = \lim_{\infty} H_t(I_0) \tag{5}$$

where H_t is the expected number of nodes which have been infected at time t, defined in Section 4.2. This is meaningful if a company want to have more people used the product.

Another objective is to maximize the total time of all nodes being infected, i.e. the objective function is

$$f(I_0) = \sum_{t=0}^{t\to\infty} S_t(I_0) \tag{6}$$

where S_t is the expected number of nodes being infected at time t. This will be meaningful if the product is payed by time and the company want to maximize the revenue.

The second objective is more interesting and more relevant to our model. So when we talk about influence maximization later, we refer to maximizing the total infected time. The influence maximization problem is defined as

$$I = \arg \max_{I_0, |I_0| \le k} f(I_0) \tag{7}$$

where k is the maximal number of nodes in initial set.

4.4.2 Optimization Analysis

In this simple model, since it can be regarded as independent cascade model with identical infecting time distribution, we can refer to methods used in independent cascade model.

A good choice of maximizing the influence is the greedy algorithm. In this section, we will prove that our objective function is submodular, so that the greedy algorithm can guarantee (1-1/e) performance of optimal solution.

Theorem 1. *The objective function $f(I)$ is submodular. i.e. for all $A \subset B \subset V$ and $s \in V \backslash B$, $f(A \cup s) - f(A) \ge f(B \cup s) - f(B)$*

Proof. For a fixed influence i, the objective function is equivalent to set paths by probability in advance and set active time by probability in advance. Since $A \subset B$, we have $H(A) \subset H(B)$, and thus $H(u) \cap H(A) \subset H(u) \cap H(B)$, $H(u) \setminus H(A) \supset H(u) \setminus H(B)$. Therefore, the gain of adding u to A is larger than adding u to B. Since the active time for each node is determined and is greater or equal than 0, the overall active time increase also have the property.

Since f_i is submodular, f is also submodular. □

The above proof shows that even we add time variable in the model and use time related objective function, the greedy algorithm will still guarantee good result.

4.4.3 Optimization Simulation

The greedy algorithm is simulated in this section. The greatest difficulty is that computing the objective function is hard even we know the initial set. Consider the graph as a probabilistic graphical model, it is even NP-complete problem to compute the probability of a node being infected. To address this problem, we use Monte Carlo method. The influence process is simulated for 400 times and average objective value is used in optimization.

Because we simulated many times and take average, the optimization time will be very long. To speed up the greedy algorithm, lazy hill climbing is used by re-evaluating marginal increase only when it's necessary.

Besides the naive (1-1/e) upper bound, we also computed the data dependent upper bound $f(I) + \sum_{i=1}^{k} \delta(i)$ where $\delta(i)$ is the top i marginal gain.

Other two alternative algorithms are tested in the simulation for comparison. The degree algorithm is iteratively adding the the node with biggest out degree to initial set. The random algorithm is iteratively adding a random node into initial set.

Finally we do the experiment in the small twitter network, setting $p = 0.05$, $q = 0.3$, $k = 1, ..., 8$. The result is shown in Figure 4. It can be seen that greedy algorithm is the best approach. The data upper bound shows that greedy algorithm is pretty near the optimal solution. The random algorithm performs bad when $k = 1$, but much better after $k > 1$. The degree algorithm also performs pretty well in this small network.

Because computing objective function is time-consuming, it's difficult to simulate in a large network. The objective of linear threshold model is deterministic and easier to compute, but still, runtime for greedy algorithm is unacceptably long even with optimization.

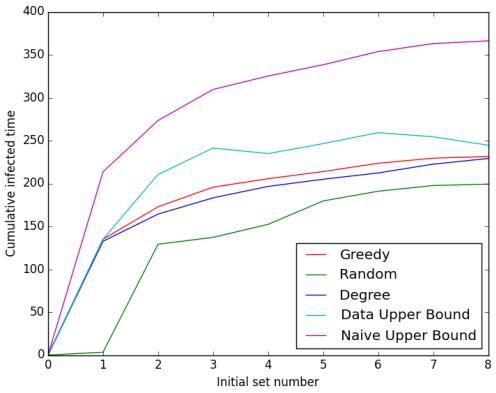

Figure 4: Total Infected Time by Initial Set Number

4.5 Extension 1 - Adopted Neighbors

4.5.1 Motivation and Introduction

Based on daily observation, the more a person's friends are using a product, the more likely he's going to continue using it. Therefore, we can set the recover rate q of a node to decrease by the number of its adopted friends. Formally, replace the second property with the following: Every time, an adopted node has probability $\frac{q}{1+n}$ to recover, where n is the number of its adopted in-link neighbors.

4.5.2 Simulations

Figure 5 showed simulations of this model with different p and fixing $q = 0.4$. Figure 6 showed simulations with different q and fixing $p = 0.2$. When p or q increases, the infection is larger. It can also be seen that the effect of q is stronger than the effect of p. When p is large enough, the infected number doesn't grow when p increases, meanwhile the infection period is shorter. Based on these two figures, the model is good to describe the infection evolution.

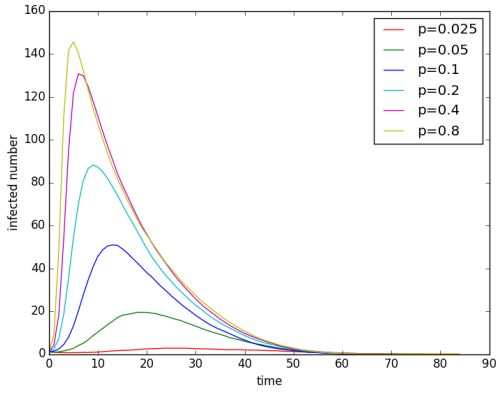

Figure 5: Infected Number by Time with Different p in Model Extension 1

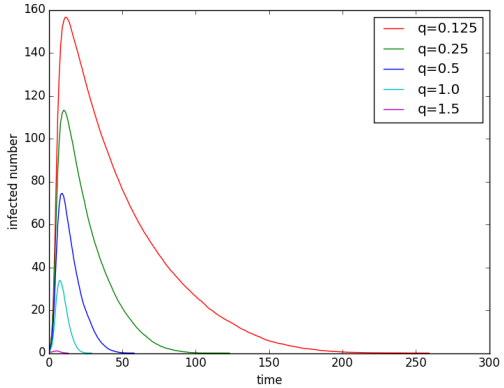

Figure 6: Infected Number by Time with Different q in Model Extension 1

4.5.3 Influence Maximization

The influence maximization for different methods are also simulated for this model at $p = 0.05$ and $q = 1.2$, in the same small directed Twitter graph, for $k = 1, ..., 16$. Surprisingly, the objective function is no longer submodular, and therefore greedy algorithm

can't guarantee a good result. The speeding up trick and upper bounds are also no longer valid. The simulation result is shown in Figure 7. The degree algorithm performed well, sometimes even better than greedy algorithm. The intuition is that initial infected nodes being together now has the advantage of keeping longer infected time, and getting more chances to infect others. This breaks the submodularity, and the greedy algorithm doesn't work well because it tend to look for initial infected nodes far away from each other.

After doing these simulations, it is realized that although greedy algorithm is a success in theoretical analysis, it still has some disadvantages as follows.

- The running time is too long, compared to other simple algorithms such as degree algorithm.
- The good theoretical properties are lost in a more complicated model, as is shown in this section.
- The proposed model is not exactly the same as reality, therefore the greedy algorithm is less stable than other simple algorithms in real world.
- Usually in a small network, the degree algorithm can achieve a similar performance.

Therefore in reality all factors should be taken into consideration to decide which algorithm to choose, and I think it's more appealing to use simpler methods.

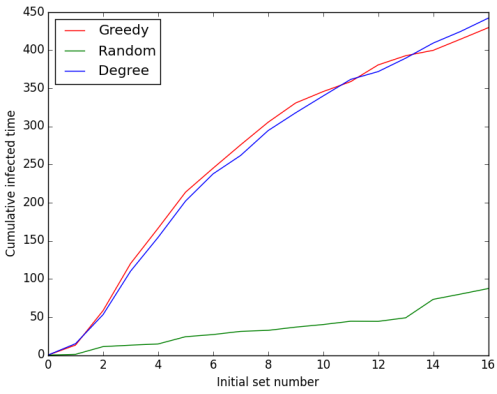

Figure 7: Influence Maximization in Model Extension 1

4.6 Extension 2 - Time Variant Rates

The following properties are added on top of the previous model:

1. As time iteration increases, p, the probability of any node being infected by its neighbor decreases.
2. Also the probability q such that n become inactive forever becomes larger as time iteration increases.

This model can better describe the situation that the influence becomes weaker as time goes forward. Also it describes that people may lose interest on the subject after certain time period, therefore more likely to stop spreading.

A simulation of this extended model over the facebook dataset is provided below:

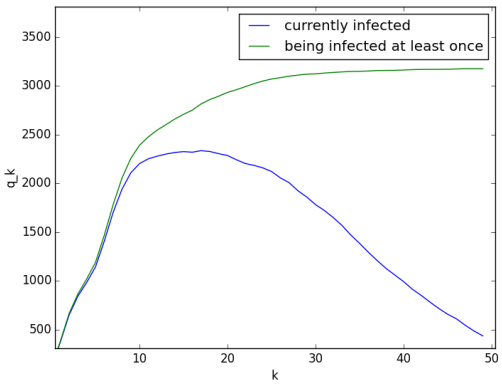

Figure 8: Infected number on Facebook network with extended model

5 Time-variant Linear Threshold Model

5.1 Model & Metric

To simulate the evolution of active nodes in social network, we propose a time-variant linear threshold model based on the linear threshold model, which is described as follows:

1. The model is based on a graph where an edge from i to j indicates i has influence on j. The influence value I_{ij} is proportional to the number of common friends (undirected graph like Facebook) or followers (directed graph like Twitter) of i and j. It's normalized so that the influence a node receives from other nodes sums up to 1, i.e. $\sum_{i \in j\text{'s neighbors}} I_{ij} = 1$.
2. Each node i has an activeness sequence, which is a non-increasing sequence of real numbers in the range $(0, 1]$, i.e. $\{A_{ik}\}, k \in \{1, 2, ..., l_i\}$ where $1 \geq A_{i1} \geq A_{i2} \geq \cdots \geq A_{il_i} > 0$. Once node i is activated, the starting activeness of the node is A_{i1}. Then, after a time step, its activeness becomes A_{i2} and so forth until its activeness reaches the end of the sequence (A_{il_i}) and then deactivated. For inactive nodes, their activeness are 0. l_i is the length of the activeness sequence of node i and let's call this active time.
3. Each node j has a threshold T_j in the range [0, 1]. At each time step, for any node j, it receives activeness from its neighbors. The activeness it receives is $\sum_{i \in j\text{'s neighbors}} A_i I_{ij}$ where A_i is the activeness of node i in the last time step. If $\sum_{i \in j\text{'s neighbors}} A_i I_{ij} > T_j$ and node j was inactive, then node j is activated.

4. Each node i has an upper bound of times to be activated (U_i) and let's call this activation upper bound. This allows a node to be activated for more than once. In default, the activation upper bound is 1, i.e. any node can be activated at most once.

Based on the above description, by setting $\forall i, l_i = \infty, 1 = A_{i1} = A_{i2} = \cdots, U_i = 1$, this model is exactly the same as the original linear threshold model. Hence, this is a general version of linear threshold model and we can gain some insights by comparing with the original model.

In this model, we focus on the evolution of active nodes in the network instead of the total number of nodes that have been activated. To be specific, the metric for this model is the integral of number of active nodes of all time. This is more suitable for the case where reward is proportional to the length of time. For example, many online games charge players based on the playing time and thus the integral of active players is more indicative than the total number of players that have played this game for revenue estimation.

5.2 Tests

To test our model, we use the combined Facebook network data, which is an undirected graph (refer to section 3 for basic statistics and visualization of this network). For simplicity, the initial set consists of users with highest degrees (let's call them influencers). The default settings are $\forall i, T_i = 0.1, l_i = 10, 1 = A_{i1} = \cdots = A_{i10}, U_i = 1$. Following are some simulation results by tweaking one of the above parameters:

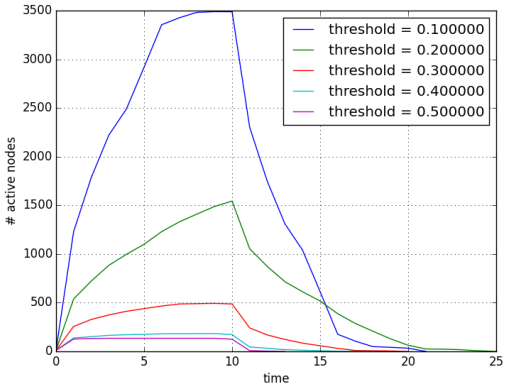

Figure 9: Plot for different thresholds

In Figure 9, we plot the number of active nodes using different thresholds (all nodes with the same threshold). In general, when threshold increases, the number of active nodes reduces, which is reasonable because nodes are more difficult to be activated.

In Figure 10, we plot the number of active nodes using different active times (activeness values are all 1). When active time increases, it takes longer before all nodes become inactive and the peak shifts right.

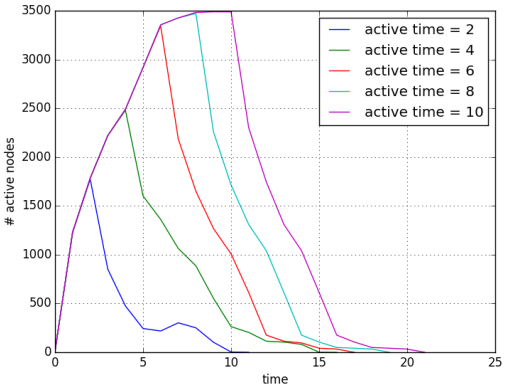

Figure 10: Plot for different active times

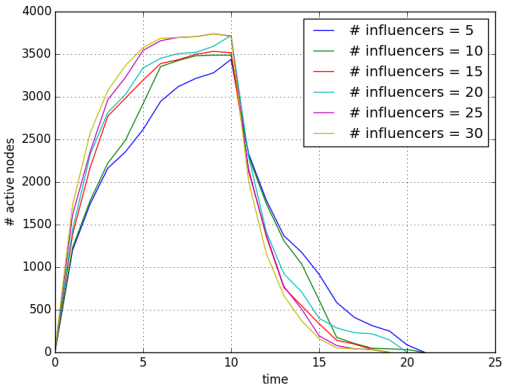

Figure 11: Plot for different sizes of initial set

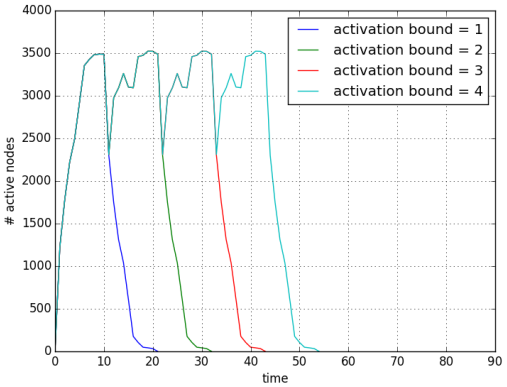

Figure 12: Plot for different activation upper bounds

In Figure 11, we plot the number of active nodes using different sizes of initial set, i.e. the number of influencers. In general, when the initial set increases, the number of active nodes increases.

In Figure 12, we plot the number of active nodes using different trigger upper bounds. We can find the periodical change in number of active nodes in cases where the activation upper bound is larger than 1 and the period is 10, which is exactly the length of our default activeness sequence.

5.3 Analysis & Algorithms

In this new model, our goal is to maximize the integral of number of active nodes over different initial sets with a given size. In the original linear threshold model, the metric is the total number of nodes that have been activated, which is proved to be a submodular function of the initial set [4]. Thus, greedy algorithm guarantees $(1 - 1/e)$ performance of the optimal solution [4].

However, the integral of number of active nodes is not submodular in the initial sets and hence greedy algorithm is not guaranteed to perform well. What's more, the scale of our data doesn't allow us to apply greedy algorithm (3963 nodes and 176312 edges) even for the singleton initial set (initial set containing only one node) because we need to enumerate all nodes, which means 3963 simulations where each one takes a few seconds.

We need to come up with other efficient algorithms to do this. Unfortunately, mathematical analysis for this problem is limited and therefore, we propose some heuristics and compare their performances in the simulation section.

The first one is the top degree heuristic, i.e. choosing nodes with highest degrees as the initial set, which we already mentioned in the previous section. This heuristic is too raw and it doesn't take the influences between nodes into consideration. For node i, define sum influence as $I_i = \sum_{j \in i\text{'s neighbors}} I_{ij}$, which is the sum of influences of i on its neighbors. Intuitively, when I_i is higher, node i is more important in information spread. Based on this, we propose the top sum influence heuristic, which chooses nodes with highest sum influences as the initial set. We find that the top 10 nodes based on sum influence is exactly the 10 egos in the ego-Facebook network data, which shows that top influence heuristic is able to find the true influencers. However, top sum influence heuristic doesn't take time-related parameters into consideration. For any node i, during the period when i is active, the total influence that i spreads to its neighbors is $I_i \cdot \sum_{k=1}^{l_i} A_{ik}$ and let's call this cumulative influence. The top cumulative influence heuristic is choosing top nodes based on cumulative influence as the initial set.

5.4 Simulation Results

Based on the three heuristics proposed in the previous section, we run the simulation and compare their performances. For any node i, we set activation upper bound $U_i = 1$ and active time l_i to a random integer in the range $[5, 15]$ where activeness values are all ones. For each of following simulation results, we average over 100 different models to reduce the effect of randomness (the only difference is the active time for nodes).

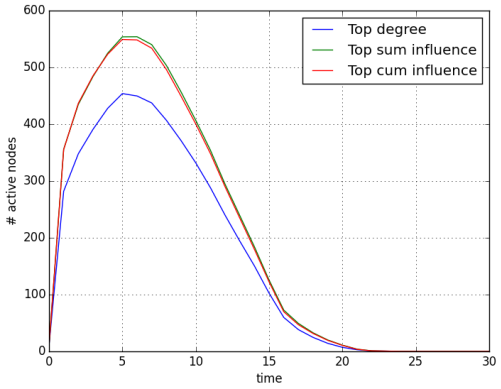

Figure 13: An example comparison of three heuristics

In Figure 13, we show an example simulation of three heuristics, from which we can see top sum influence heuristic and top cumulative influence heuristic outperforms the top degree heuristic. What's more, the difference between top sum influence heuristic and top cumulative influence heuristic is negligible.

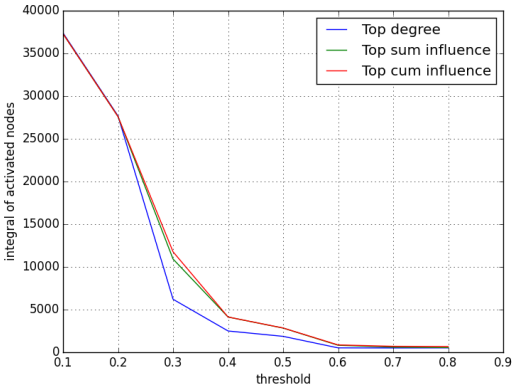

Figure 14: Performance plot using different thresholds with init set size 50

In Figure 14, we fix the initial set size to 50 and compare the performances using different thresholds for all nodes. As expected, when the threshold increases, the integral of activated nodes reduces. The gap between top degree heuristic and the other two algorithms is the largest when threshold is 0.3.

In Figure 15, we fix the threshold to 0.5 and compare the performances using different initial set sizes. We can see there's a gap between top degree heuristic and the other two algorithms while the difference between

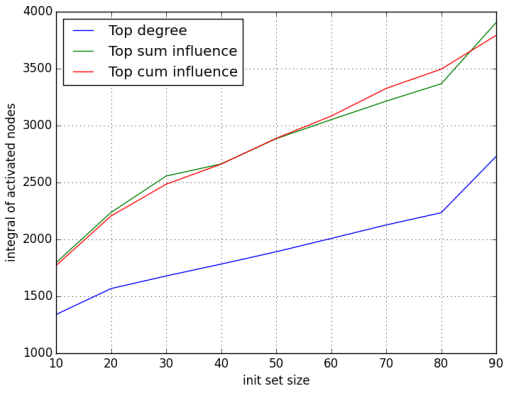

Figure 15: Performance plot using different initial set sizes with threshold 0.5

the other two algorithms is still very small.

5.5 Conclusion

The theoretical result for basic probabilistic model is analyzed and works well in cases when infection is large. For influence maximization in basic probabilistic model, the greedy algorithm performs well both in theory and practice. However, when performed in extended model or larger network, the disadvantages of greedy algorithm start to appear, most importantly, the long runtime and the break of submodularity.

From the simulation results in linear threshold model, we can conclude that among the three heuristics, the top degree heuristic is the worst, which only involves degree information. The performance of the other two algorithms are very close and they outperform the top degree heuristic. To make use of the time-related parameters, we can further generalize the top cumulative influence heuristic to an algorithm that chooses top nodes based on $f(I_i, \sum_{k=1}^{l_i} A_{ik})$ where $f(\cdot)$ could be any function. We can consider I_i and $\sum_{k=1}^{l_i} A_{ik}$ as features and treat this as a machine learning problem and we should achieve an algorithm that outperforms the top sum influence heuristic because this algorithm is also a special case of the above general algorithm. In the future, we need more mathematical analysis for this problem and achieve better algorithms other than heuristics.

6 Future works

First, the real world information spread is complicated, so it's appealing to propose more model extensions to reflect it.

Since the degree algorithm takes too much time, a stable and fast algorithm is desired, outperforming basic heuristic algorithms.

For time-variant linear threshold model, as we discussed in Section 5, we can consider I_i and $\sum_{k=1}^{l_i} A_{ik}$ as features and treat this as a machine learning problem and we should achieve a better algorithm. We also need more mathematical analysis for this problem to get better algorithms.

Also, the simulation result should be compared with real world influence evolution data. This will need a much more complicated data collection process, but benefits a lot for the study.

7 References

[1] Damon Centola, The spread of behavior in an online social network experiment, Science, 2010.

[2] S. Morris. Contagion. Review of Economic Studies 67, 57-78, 2000.

[3] Bakshy, Rosenn, Marlow, and Adamic, The role of social networks in information diffusion, Proc. WWW, 2012

[4] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the Spread of Influence Through a Social Network, KDD'03.

[5] C. Asavathiratham, S. Roy, B. Lesieutre, G. Verghese. The Influence Model. IEEE Control Systems, Dec. 2001.

[6] J. Goldenberg, B. Libai, E. Muller. Using Complex Systems Analysis to Advance Marketing Theory Development. Academy of Marketing Science Review 2001.

[7] S. Wasserman, K. Faust. Social Network Analysis. Cambridge University Press, 1994.

[8] G. Nemhauser, L. Wolsey, M. Fisher. An analysis of the approximations for maximizing submodular set functions. Mathematical Programming, 14(1978), 265294.

[9] M. Granovetter. Threshold models of collective behavior. American Journal of Sociology 83(6):1420-1443, 1978.