Spring 2015

MAXIMIZING THE SPEED OF INFLUENCE IN SOCIAL NETWORKS

Yubo Wang
San Jose State University

MAXIMIZING THE SPEED OF INFLUENCE IN SOCIAL NETWORKS

A Creative Project Report

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Computer Science

by

Yubo Wang

May 2015

The Designated Thesis Committee Approves the Thesis Titled

MAXIMIZING THE SPEED OF INFLUENCE IN SOCIAL NETWORKS

by

Yubo Wang

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2015

Dr. Robert K. Chun, Department of Computer Science Date

Dr. Teng Moh, Department of Computer Science Date

Dr. Chris Pollett, Department of Computer Science Date

ABSTRACT

MAXIMIZING THE SPEED OF INFLUENCE IN SOCIAL NETWORKS

by Yubo Wang

Influence maximization in social networks is the problem of selecting a limited size of influential users as seed nodes so that the influence from these seed nodes can propagate to the largest number of other nodes in the network. Previous studies in influence maximization focused on three areas, i.e., designing propagation models, improving algorithms of seed-node selection and exploiting the structure of social networks. However, most of these studies ignored the time constraint in influence propagation. In this paper, I studied how to maximize influence propagation in a given time, i.e., maximizing the speed of influence propagation in social networks. I extended the classic Independent Cascade (IC) model to a Continuous Dynamic Extended Independent Cascade (CDE-IC) model. In addition, I propose a novel heuristic algorithm and evaluate the algorithm using two large academic collaboration data sets from www.arXiv.org. Comparing with previous classic heuristic algorithms on the CDE-IC model, the new algorithm is 9%-18% faster in influence propagation. Furthermore, I gave solution to calculate propagation probability between adjacent nodes by exploiting the structure of social networks.

ACKNOWLEDGEMENTS

I sincerely thank my project advisor Dr. Robert Chun for giving me invaluable instructions during my completing of this project. I would like to thank my committee members Dr. Teng Moh and Dr. Chris Pollett for their great suggestions to my work.

I would also like to show great appreciate to my family for their support. Dedicated to my wife Li and two adorable daughters Arwen and Claire.

Table of Contents

List of Tables

List of Figures

INTRODUCTION

Maximizing the speed of influence propagation is crucial in marketing through social networks. In this paper, I gave a whole set of solution including improving model, designing a novel seed-node selection algorithm and calculating propagation probability. Online social networks, such as Twitter, Facebook and Pinterest, despite having different functionalities and target users, all connect people into a virtual society. Each user, represented by a node in a social network, is connected to other users based on certain relationships, such as followers on Twitter, or friends on Facebook. The communication between users is either one-way or two-way depending on the relationship. For example, if the relationship is "follower" on Twitter, a user can only follow the followee's post, thus forming a one-way communication. If the relationship is friends on Facebook, both sides can post on each other's wall, thus forming a two-way communication. The high frequency of communication together with the large number of users in social networks can lead to explosive propagation of information and provides an ideal marketing platform.

In viral marketing strategy, a company invites some initial users, i.e. the seed nodes, to try its new products or technologies. The company would give these initial users free samples and hope that they will give a positive feedback in social networks. By the power of word-of-mouth, these users may affect their neighbors in a social network. These affected neighbors may subsequently propagate the influence to their own neighbors, and

so on. The challenge in viral marketing strategy is how to select the seed nodes to maximize return of investment.

Consider the following case as an example. A Yogurt company wants to launch a two-week campaign to promote their product with a budget of less than $20k. The company would like to select 40 initial users to try their sample and ask these users to write a blog in their social networks. Since each initial user will cost resources including money, time and human labor, the company should decide carefully how to select these initial users to affect most of other potential customers.

Other companies and individuals that hope to promote their new products and new ideas by the power of word-of-mouth through social networks face the same challenge. Social networks provide a great opportunity to promote new products or ideas because of the large number of users and the high frequency of communication. In addition, the propagation of information can be fairly quick if the right seed nodes are selected. However, the large scale of social networks and their complicated structures made it challenging to select the right seed nodes. We need a solution that is efficient even when scaling up to large social networks and guarantees to maximize the number of affected nodes under this solution.

Influence maximization was first proposed as an algorithm problem by Domingos and Richardson in a study of viral marketing[1, 2]. Instead of viewing users as independent individuals and only considering the intrinsic value of each users, they made the selection based on a customer's network value, which is defined as an expected total

profit that can be achieved from all the other customers who are influenced by this customer directly or indirectly.

Kempe, Kleinberg and Tardos provided a foundation to solve the influence maximization problem[3]. They proved that the optimization problem of selecting the most influential seed nodes is NP-hard. They also presented the first provable approximation solution to this problem, which is within 63% (1- 1/e) of optimal under two different models, the Independent Cascade model (IC) and the Linear Threshold (LT) model.

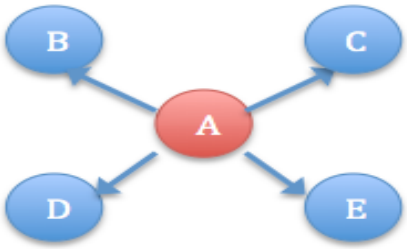
In particular, they modeled a social network as a graph, where nodes represent individuals and edges represent relationships (friends, family or followers) between users. Influence maximization can then be described as when starting from seed nodes of size k, how influence will propagate to the other nodes with a certain probability and reach a maximized total number of influenced nodes. They used a greedy algorithm named the Hill-Climbing algorithm to obtain their solution, which is an approximation result to optimum within bound 1-1/e (e is the natural logarithm base). One big drawback in this algorithm is the efficiency. Trying to calculate influence of a given size of seed nodes proves to be a difficult task. Instead of obtaining a precise value, they ran Monte-Carlo simulations on their models multiple times to obtain an accurate estimation. However, even finding a small seed nodes set in a moderately large network (e.g. 15000 nodes) would take days to finish.

Several following studies have been carried to improve the efficiency of seed-node selection algorithms. Leskovec, Krause and Guestrin proposed a nearly optimal algorithm

called Cost Effective Lazy Forward (CELF) algorithm[4]. In this algorithm, the number of nodes to be considered in each round of seed selection is greatly reduced by exploiting the submodularity property of models. This algorithm scaled well to large data sets and their experiments showed that it was 700 times faster than Hill-Climbing algorithm. Further optimization was achieved by Goyal, Lu and Lakshmanan[5]. In their paper, they proposed an algorithm called CELF++, which was 35%-55% faster than CELF. There are also several other greedy algorithms that perform similarly as the CELF++ algorithm.

Chen, Wang and Yang[6] tackled the efficiency issue of seed nodes selection from a different direction. Instead of trying to further reduce the running time of a greedy algorithm, they improved the heuristics methods. Their new heuristics method achieved a nearly matched result comparing to greedy algorithms, but with significantly reduced running time. The new heuristic method was more than six orders of magnitude faster than the existing greedy algorithms.

All the previous studies focused on the space maximization of influence, i.e. how to maximize influence propagation in a social network without time constraint. In this thesis, I will study the speed maximization of influence propagation, i.e. how to maximize influence propagation in a social network in a given time frame. The meaning of maximized speed of influence can be illustrated in the example shown in Figure 1 and Figure 2.



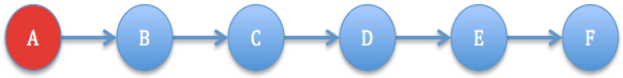


Figure 1: Star Structure Figure 2: Linear Structure

Suppose Figure 1 and Figure 2 are two subsets in a social network. We will select a seed node that will propagate its influence to all the other nodes directly or indirectly connected with it. Which node A should we choose as a seed node? From the point of view of space maximization, node A in Figure 2 is a better choice because it can propagate its influence to five other nodes, while node A in Figure 1 can only propagate its influence to four other nodes. However, from the point of view of speed maximization, node A in Figure 1 will be a better choice. With the assumption that influence can only propagate one step per unit time, node A in Figure 1 can reach more other nodes in one unit time,

This example illustrates that the previous studies focusing on space maximization ignored one important constraint, i.e., time. A hidden assumption in those models is that time of influence propagation is unlimited. The process of influence propagation in a social network would continue until no new nodes can be affected. This assumption is not always true in real life. Typically, each campaign started by a company has a valid period. If a campaign is finished, the later affected users are not counted and less meaningful. Technologies also tend to have a short shelf life. When a new technology spreads in social networks, it shall reach users as fast as possible in order to be profitable.

If speed of influence propagation is slow, other competitors will emerge to challenge its market position. For all these reasons, time is a critical constraint that must be considered.

Therefore, a similar but more important problem comparing to space maximization is how to maximize the number of influenced users in a given time, i.e. maximizing the speed of influence propagation. In this paper, I will focus on selection of initial seed nodes in a given social network in order to achieve maximized speed of influence.

In the following sections, I first introduced previous models and seed-node selection algorithms. Then I showed how I extended a classic IC model to Continuous Dynamic Extended IC (CDE-IC) model. In addition, I proposed a novel heuristic seed-node selection algorithm. I then gave formulas to decide propagation probability between adjacent nodes. In order to test my new algorithm, I compared this algorithm with three other most popular heuristic algorithms in two data sets. The result showed that even a small modification to the existing algorithms could lead to a big boost to the quality of seed nodes selection. In the end, I discussed my results and gave future directions for the study of influence maximization.

RELATED WORK

Influence maximization was first proposed as an algorithm problem by Domingos et al.[1] in their study of viral marketing. Viral marking originates from the notion that ideas spread like viruses. In contrast to direct marking that evaluates each customer independently, the viral marketing strategy exploits the network value of each customer.

Let us first consider how a company finds its potential customers. If a company's investment to a user is (I), e.g. sample or advertisement, and the expected return is (R), i.e. when user purchase product from the company, the profit (P) can be determined as $P = R - I$. Only when P is positive, will a company deem the user as a valuable customer.

Calculation of R is different in direct marking vs. viral marketing. In direct marketing, each user is independent from other users. A company only considers direct purchase action from a user and the user will decide his purchase action independently, not being affected by others' action or persuasion. Therefore, the most valuable user is the user who will purchase most products from the company in direct marketing.

Domingos et al. argued that each customer does not exist in a society independently, but are connected to each other in a social network. The marketing decision should not be based solely on each individual's purchase action but also considering its network value. The total return a company can be expected from a user is

the sum of this individual's value plus his network value. The network value of a user is the total purchase actions of other users influenced by this user directly or indirectly. A user with a low individual value, which may be lower than the investment *I*, can still be a valuable potential customer to the company if he has a high network value in viral marketing. In viral marketing, the purchase action or opinion of a user will affect his connected users. The connected users, such as friends or families of the initial user, may obtain information from the initial user that will affect their purchase decision. People tend to trust opinions from people that they are connected with.

Studies on influence maximization in social networks are mainly focused on three areas. The first area is the selection and design of propagation models. A good model should be easy to understand, able to scale to large social networks and close to the complex structure of social networks. The second area is improving seed-node selection algorithms. There are two metrics to evaluate a selection algorithms, efficiency and quality. Efficiency measures how fast an algorithm can select a given size seed nodes, and quality measures how close the result is to optimal solution. The third area is how to reflect the structure of social networks in the propagation model, and how to decide the propagation probability between adjacent nodes. I will focus on the first two areas, i.e. propagation models and seed-node selection algorithms, in the following sections. The third area is often ignored by previous studies, and I will address it in the section of Solution Framework.

Propagation Models

In the study of the viral marketing problem, Domingos et al. proposed that users do not exist in social networks independently. Each individual's decision of whether to purchase a specific product is affected by opinions from his connected users. In other words, we should model people as networks. The two basic models of social networks are Linear Threshold (LT) model and Independent Cascade (IC) model. Many other models extended from these two basic models under different conditions. All models aim to reflect the relationships of people in real social networks.

Let us first define the parameters in a social network. G is a directed graph that represents the entire social network. V is a node set, in which each node v in V represents an individual in a social network. Each node v can have a status of "active" or "inactive". An active status means that node v has been affected, and an inactive status means it has not been affected. E is the edge set that represents all relationships between individuals. Each edge e in E can have a different weight to reflect the relationship strength between two nodes.

Influence maximization problem can be stated as the following: if we select a size of k nodes from V as seed nodes, what is the optimal seed-node selection to make the number of affected nodes maximized in the social network. The affected nodes are maximized when the influences from those seed nodes propagate through existing edges between nodes until no more nodes can be affected.

There are two constraints in the modeling of social networks. First, the change of node status is irreversible. During each step, each node is either active or inactive.

However, once a node changes its status from inactive to active, it cannot be changed back to inactive status. Secondly, the tendency of each inactive node to become an active node increases monotonically with its active neighbors. Specifically, the probability of an inactive node to become active increases as more neighbors of the inactive node become active.

Under these two constraints, the influence propagation starts from an initial active seed node set and progresses in a cascade method that each inactive node is affected by its active neighbor nodes. When more neighbors of an inactive node become active, the inactive node may become active, and subsequently propagates the influence to its inactive neighbors, until no more nodes can be affected.

Linear Threshold Model

Linear Threshold (LT) Model was proposed by Granovetter and Schelling to simulate influence propagation in social networks [7, 8]. LT model is based on node-specific threshold. The threshold represents the difficulty of switching an inactive node to an active node. A larger threshold value means a node is less likely to switch its status.

In LT model, node u is connected to a set of neighbor nodes N. Each node n in set N is connected to node u by an edge with a weight of $b_{u,n}$. The total weight of nodes in set N is no more than 1 ($\sum b_{u,n} \leq 1$). Each node u has a threshold θ_u ($0<\theta_u<1$) that defines the minimum requirement of its active neighbor nodes set. When the total weight of all active neighbor nodes is greater than θ_u, the inactive node u switches its status from inactive to active. Because of the irreversible character of a status, if at step t a node is active, it remains active at step t+1. The value of θ_u is a random

constant number between [0,1]. It can be obtained from a social network structure or set to a constant value, e.g. ½. The value of θ_u reflects the tendency of a node to adopt a new idea when it is under the influence of its neighbors.

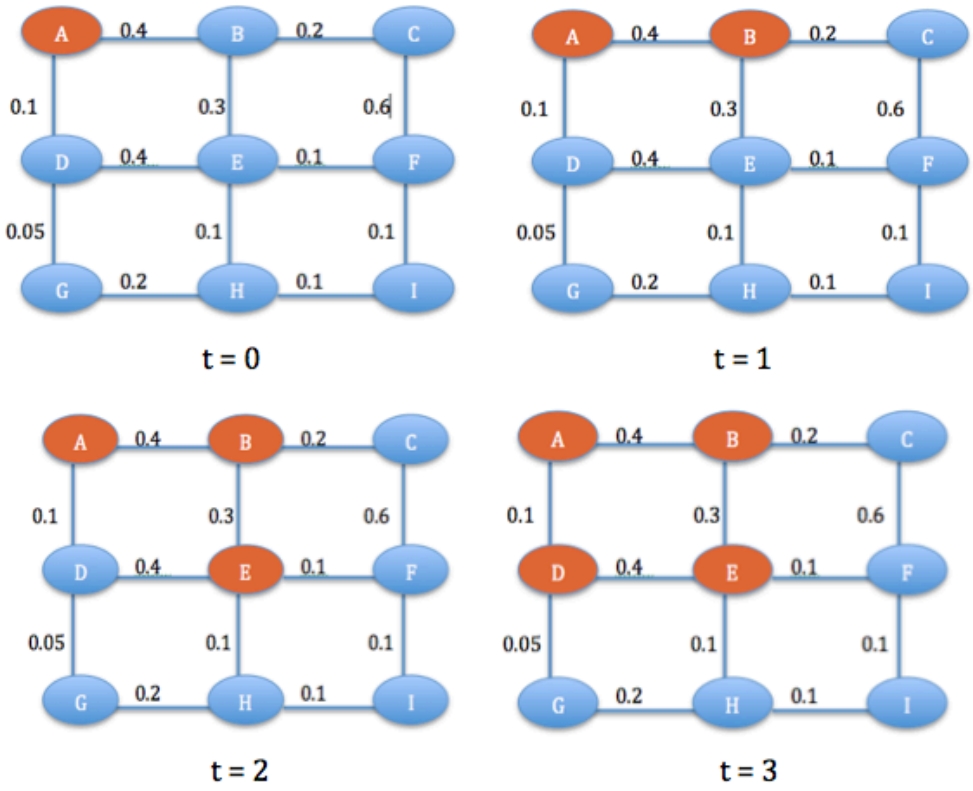


Figure 3: Influence propagation in LT model

The process of influence propagation in LT mode is shown as in Figure 3. Each node has the same threshold value of 0.3. The weight between any two nodes is labeled at the edge. At time 0, node A is selected as the seed node with an active status. Node A will try to propagate influence to its neighbor nodes of node B and node D. Edge AB has a weight value of 0.4 and edge AD has a weight value of 0.1. In this condition, only node B satisfies the condition of switching from an inactive status to an active status

as the total weight of active neighbor nodes of node B is greater than the threshold of 0.3. Therefore, node B turns into active status at time 1. The total weight of active neighbors of node D is 0.1 and node D remains inactive at time 1.

Following the same calculation, node E turns into active status under the influence of node B at time 2. Although node D fails to switch into active status at time 1, as more of its neighbor nodes become active, node D becomes more likely to switch its status as the propagation proceeds. At time 2, as node E turns into active, the total weight of active neighbor nodes of node D becomes greater than the threshold. Therefore, node D switches from inactive status to active status at time 3. After time 3, there are no more nodes that satisfy the condition to switch and the process of influence propagation stops.

There are two interesting observations in this propagation process. First, although some people will not accept new technologies at first, they are likely to change their minds as more of their neighbors accept the new technologies. These people are represented by node D in the example. Secondly, although the weight between node C and node F is higher than the threshold, they have no chance to switch to active status. Their active neighbor nodes are not powerful enough to propagate the influence to them. Node C and node F represent the users in social networks who are eager to accept new products. However, because inappropriate seed nodes are selected, the propagation process fails to discover these users. We have to carefully consider this during the seed nodes selection phase.

Independent Cascade Model

The Independent Cascade (IC) model is a dynamic cascade model based on probability theory. The simple IC model was proposed by Goldenberg, Libai, and Muller[9]. In IC model, the process of influence propagation in social networks can be illustrated as the following steps. An initial seed-node set is selected at time 0. If node A becomes active at time t, it will try to influence its inactive neighbors at time t+1 with a probability of p. Node A will only have one chance to influence its neighbors, and will not try again whether it succeeds or not. If an inactive node is connected with more than one newly active node, these newly active nodes will try to influence the inactive node in a random sequence. In addition, the result of influence propagation between two nodes is not affected by actions of other nodes. If no more nodes can be affected, the process stops.

The process of influence propagation In IC model is shown in Figure 4 and Figure 5, each representing an outcome from a single run. In both scenario, node A is selected as the seed node. For each newly activated node, it will propagate its influence to its neighbors based on the propagation probabilities between them. Each running of the simulation will obtain a different result. In Figure 4, the influence propagation reaches to 4 other nodes (D, E, F and H), while in Figure 5; the influence propagation reaches to 5 other nodes (B, C, E, F ad H). The end results are different between these two runs. Therefore, we need to run the experiment multiple times in the IC model to obtain an accurate estimation of influence propagation with specific seed node selection.

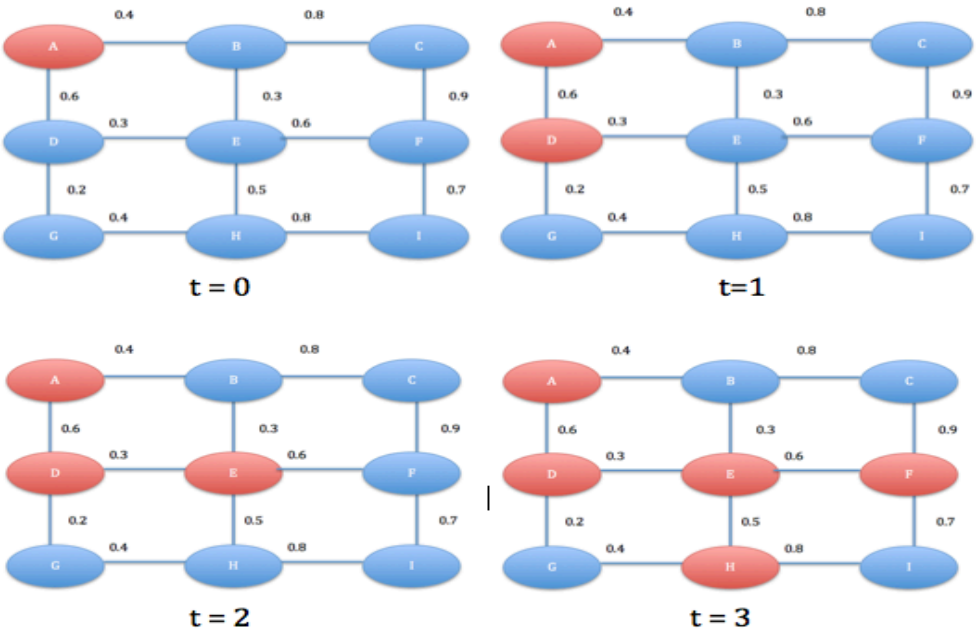


Figure 4: Influence propagation in IC first try.

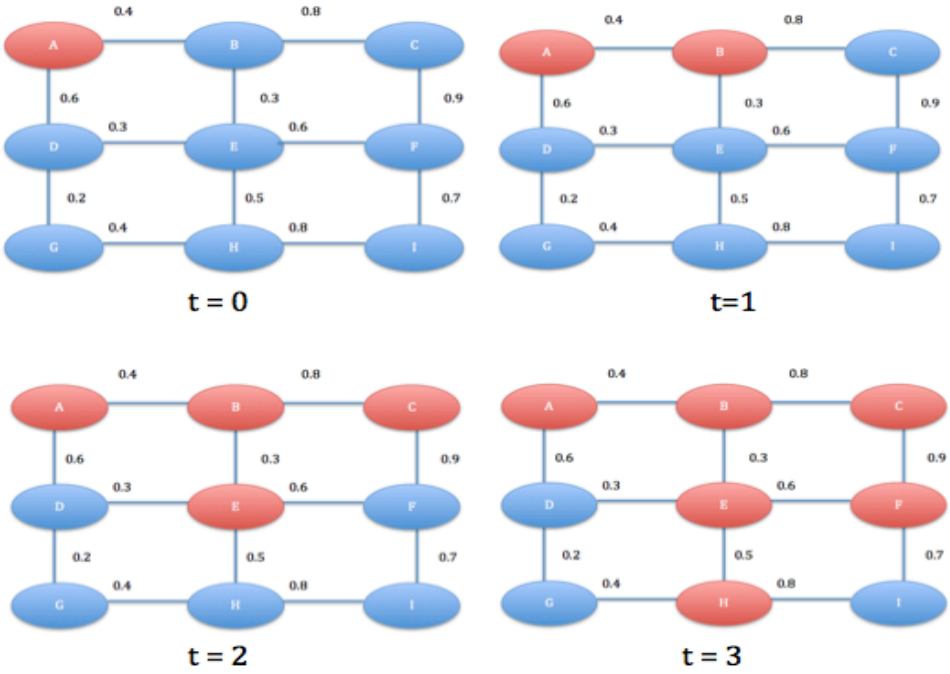


Figure 5: Influence propagation in IC second try

Extensional IC Model

LT Model and IC Model are two effective models of social networks. These two models explained a basic idea of how information is propagated through links between nodes in a social network. However, both models simplified social networks based on an assumption that whenever a node is active, it always try to propagate the influence to its neighbors. Different behaviors of users are not reflected in these two models while formalizing the propagation of information in a social network[10].

Wang, Qian, and Lu proposed an Extensional Independent Cascade (EIC) model. In the EIC model, the influence propagation between two adjacent nodes is no longer decided by only one probability. Instead, the propagation process is divided into two phases involving a spreading phase from an active node and an adopting phase from the inactive node being influenced. In the spreading phase, an active node decides whether to spread the influence to its neighbor nodes based on a spreading probability of p_s. If the active node decides to spread out the influence to its neighbor nodes, the adopting phase is similar as in the original propagation process in classic IC model. In the adopting phase, each inactive node decides independently whether to adopt the influence based on an adopting probability of p_a. Here, the concept of probability p_a is the same as the probability p in the classic IC model.

The EIC model added one more step in the process of influence propagation. The new propagation probability in EIC model should be the product of probability in spreading phase and probability in the adopting phase, such that probability of

propagation $p_p = p_a \times p_s$. The propagation probability in EIC model is no longer decided only by the adopting probability p as in the classic IC model.

To simulate in the EIC model, Wang et al. [10] used a similar greedy algorithm as used by Kempe et al. in the classic IC model[3]. In the classic IC model, the Hill-Climbing greedy algorithm tries to decide whether each edge is valid in advance. They throw a coin with bias $p_{u,w}$, where u is a newly activated node and w is its inactive neighbors. If the trial is successful, then the edge is claimed as a live edge, and if the trial is failed, the edge is claimed as a blocked edge. In the EIC model, each active node will first decide whether to propagate the influence in the spreading phase with probability p_s. If it is successful, the active node will try to activate its neighbors with the adopting probability of p_a. If the active node fails, all its out edges will be blocked, and there is no need to try the adopting phase. This is a delicate extension to the classic IC model.

The EIC model can also be illustrated in an example of transmission of infectious diseases. If someone catches flu, he may decide to constrain the virus by completely isolating himself from his connections. In this case, all his connected people will be prevented from catching the disease. If the infected person does not constrain from spreading the virus, all his connections are under the danger of being infected as well. However, some of his connections may decide to actively protect themselves from the infected person by wearing protective equipment whenever in contact with the infected person, thus reduce the probability of being infected.

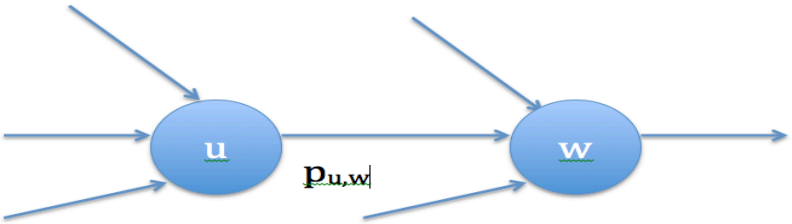


Figure 6: Classic IC model

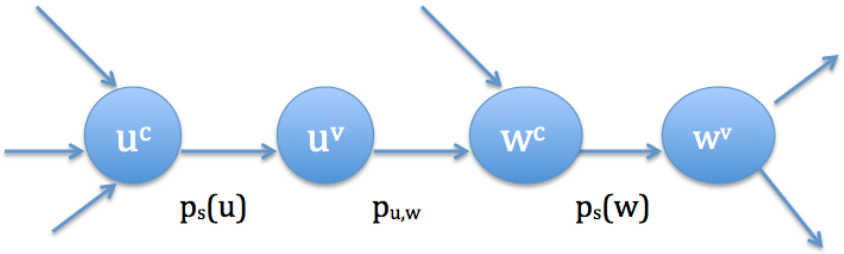


Figure 7: EIC model

The comparison of classic IC model and EIC model is illustrated in Figure 6 and Figure 7. In the classic IC model (Figure 6), node u is an active that has accepted the influence and node w is one of u's inactive neighbor nodes. There is only one probability $p_{u, w}$ that decides whether node w will be affected by node u. In contrast, in EIC model (Figure 7), each node splits into two nodes. One node is used to show if it is active and the other node is to show if it has spread the influence to its neighbor nodes. The first node has all the original node's in-neighbors and the second node has all of its out-neighbor nodes. If a newly activated node u wants to propagate its influence to its out-neighbor nodes, node u has to decide whether to spread the influence first. If node u

decides not to spread the influence, the edge between u^c and u^v is blocked. In this case, none of the out-neighbor nodes of u can be affected through path u^c --> u^v --> w^c. No mater how easily a neighbor node of u can be affected by its influence, the neighbor node will not have a chance of being influenced because node u decides not to spread the influence.

Time-Delayed IC model

The EIC model is only one example of modification to the classic IC model. There are other models that try to extend the classic IC models based on different constraints. Chen, Lu and Zhang[11] proposed a Time-Delayed IC model that handles the situation where the influence of an active node cannot reach its inactive neighbor nodes immediately.

In Time-Delayed IC model, Chen et al. proposed that the process of influence propagation in social networks has a postponed phenomenon. This postponed phenomenon is created by the fact that not all users are available on the social networks at a given time. A users need to log into social networks to check updates from his neighbors. When an active node tries to propagate influence to its inactive neighbors, the process of propagation has to pause if the inactive neighbor is offline. When an inactive user cannot see the new updates from his active neighbor, he cannot be affected by the influence propagated to him. Only after the inactive user log into the social networks can he receive the influence. The process after this initial step of postpone is the same as classic IC model. The inactive user will decide whether to be influenced based on the propagation probability between him and the active user.

In Time-delayed IC model, each edge is given a meeting probability to decide the time-delay value in addition to the normal propagation probability. Before each experiment, they first decide the time delay value by throwing a coin with bias p_m that is the probability of two users meet on the social network.

Seed-node Selection Algorithms

As discussed in the previous section, the study of influence maximization focuses on three areas, i.e. designing propagation models, optimizing seed-node selection algorithms, and exploiting the structure of social networks. I will focus on optimizing seed-node selection algorithms in this section.

Influence maximization is the selection of a seed node set in order to reach the maximum number of other nodes when influence propagates in social networks. Seed-node selection algorithms play a key role in the study of the influence maximization. There are two metrics to measure a seed node selection algorithm, efficiency and quality. Efficiency is important for a seed-node selection algorithm. A good algorithm should be able to scale up to a large data set with the ever-expanding size of popular social networks in millions or billions. If a seed-node selection algorithm can only handle a small data set, it is useless to real social network study.

Quality is the other important metrics of seed-node selection algorithm. A good algorithm is not only fast and scalable, but it also should give a correct answer. If there is an optimal solution of seed-node selection, a good algorithm should be able to find these optimal seed nodes. Less ideally, a qualified seed-node selection algorithm should return a near optimal solution. Unfortunately, influence maximization problem is NP-hard in

both the LT Model and the IC Model[3]. There is only provable approximation for the optimal solution.

Hill-Climbing Algorithm

Kempe et al.[3] proposed the first provable approximation solution by using Hill-Climbing algorithm. They proved that the Hill-Climbing algorithm guarantees to achieve an approximation solution with a factor $(1- 1/e – \varepsilon)$ to the optimal solution in both the LT Model and the IC Model. Here e is the natural logarithm base and ε is a small positive real number. This algorithm is based on the theory of submodular functions[12].

Kempe et al. proved their approximation by using the submodular property of the function. Any function f is submodular if it has the property of "diminishing return value". The property can be expressed as the following formula 1[3]: $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$. Here, S and T are two sets and S is a subset of T. v is a new element that does not belong to any of the two sets. The above function can be explained as the following: if we add a new element to a set S, the extra return value from the newly added element will not be less than the return value if we add it to a superset of the current set, set T. In other words, the earlier we add a new element to a set, the more return value we can obtain from this element.

The influence maximization problem has the property of "diminishing return value". We can think of seed node selection as a discrete process. Each time we add a new seed node to the set of already selected nodes. For any specific selected seed node, we will not expect more return value from it if we choose it as a seed node later.

In order to obtain an accurate estimation, Kempe et al. run Monte Carlo simulation sufficient number of times (20,000) for selecting each seed node. Chen et al. gave an implementation of the Hill-Climbing algorithm[6] (See Figure 8). If we want to select k initial seed nodes from a social network, one node per step, we need to calculate all possible selections to select the most influential node in each step. Then the time complexity of the Hill-Climbing algorithm is O (kRmn), in which m is the number of nodes, and n is the number of edges. Clearly, the greedy algorithm is not efficient. Although greedy algorithm guarantees the quality of seed nodes selection, it is not efficient enough for large-scale social networks. There are many following studies trying to improve efficiency of this algorithm.

1: initialize $S = \emptyset$ and $R = 20000$
2: **for** $i = 1$ to k **do**
3: **for** each vertex $v \in V \setminus S$ **do**
4: $s_v = 0$.
5: **for** $i = 1$ to R **do**
6: $s_v \mathrel{+}= |RanCas(S \cup \{v\})|$
7: **end for**
8: $s_v = s_v / R$
9: **end for**
10: $S = S \cup \{\arg\max_{v \in V \setminus S}\{s_v\}\}$
11: **end for**
12: output S.

Figure 8: Hill-Climbing Algorithm

CELF Algorithm

The biggest drawback of the Hill-Climbing greedy algorithm is low efficiency. Even selection of a small seed set in a moderate large social network took days[3]. Leskovec et al. tried to solve this problem using an improved greedy algorithm named

Cost-Effective Lazy Forward (CELF)[4]. By exploiting the property of submodularity of social networks, they greatly reduced the number of candidate nodes. In each step, they only evaluated a few candidate nodes to obtain an efficient solution that is scalable to a large dataset. Their experiment showed that they would get a near optimal solution while being 700 times faster than the simple Hill-Climbing greedy algorithm.

Further effort of reducing the running time of the greedy algorithm was carried by Goyal et al.[5]. They proposed an improved CELF algorithm called "CELF++" that further decreases the running time by 35%-50% compared to the CELF algorithm.

The Hill-Climbing algorithm and its improved versions can guarantee quality in seed node selection. By using greedy algorithms, we can always obtain an approximation solution to the optimal solution within a factor. However, the efficiency limited the use of greedy algorithms in large data sets.

Heuristic Algorithm

To solve the efficiency problem of greedy algorithms, heuristic algorithms were applied in the seed-node selection phase. In contrast to greedy algorithms, heuristic algorithms may not provide the best result, but they are able to obtain an acceptable result in much less time. Two widely used heuristic algorithms are Degree-Centered algorithm and Distance-Centered algorithm.

In Degree-Centered algorithm, the nodes that have a large number of connections in a social network, i.e. the high degree nodes, are deemed as influential nodes. The more out-neighbors a node has, the more influence it is believed to have in a social network. This is a simple and intuitive assumption. However, a known phenomenon in social

network is that high degree nodes tend to connect to each other. Therefore, if only high degree nodes are selected as seed nodes, these seed nodes will have a large overlap with each other. The overlapped nodes will not bring any additional value to the set of seed nodes to maximize influence propagation.

In Distance-centered algorithm, influential nodes are determined by computing the average distance from each node to other nodes in a social network. The nodes that have a smaller average distance to other nodes are the candidates for seed nodes.

The Degree-Centered algorithm can achieve better influence propagation than other heuristic algorithms[3], but it is still not as good as greedy algorithms. In general, heuristic algorithms were not studied extensively in the research field because of the low expectation of quality. However, Chen et al. proposed an improved heuristic algorithm in IC Model[6]. This improved heuristic algorithm performed comparably to greedy algorithms. Importantly, this algorithm significantly reduced the running time in the seed-node selection phase with six orders of magnitude.

In this new heuristic algorithm, Chen et al. introduced the concept of "discount" to the degree of a node. The logic is that when a node is selected as a seed node, its neighbor nodes will become less influential to the social network. Therefore, there should be a discount on the neighbor nodes. They proposed two methods to discount the degree of a node. The first method is called "Single Discount", in which the degree of all neighbor nodes of a selected seed node is reduced by one. The second method is more complicated. For each newly activated node, it will calculate the affection to its inactive neighbor nodes.

SOLUTION FRAMEWORK

In this section, I study how to maximize the speed of influence propagation in social networks. I extended the classic IC model to a Continuous Dynamic Extended Independent Cascade (CDE-IC) model. In addition, I proposed a novel heuristic algorithm and evaluate the algorithm using two large academic collaboration data sets. Comparing with previous classic heuristic algorithms on the CDE-IC model, the new algorithm achieved higher speed of influence propagation. Furthermore, I discuss how to decide the parameters in the propagation model by exploiting the structure of social networks.

Design of Propagation Model

The Extensional Independent Cascade (EIC) model extends the process of influence propagation from one phase in classic IC model to two phases involving a spreading phase and an adopting phase[10]. In EIC model, the probability of propagation (p_p) is decided by the probability of spreading ($p_{s)}$ and the probability of adopting ($p_{a)}$.

The EIC model with two phases in propagation process is a good extension to the classic IC model, and it is more close to reality in social networks. However, The EIC model has the same drawbacks as the classic IC model, in that the propagation process is one-time and static.

One-time refers to the assumption that a newly activated node will only try to propagate the influence to its neighbors once. If a node turns into active status in step t, it will try to propagate its influence to its inactive neighbor nodes in step t+1. No matter

what the result is, it will not try in later steps. Although this is simple for modeling the process of influence propagation, it does not reflect the reality in social networks. For example, if one of your friends posts a message on Facebook "Hey guys, I just got a new iPhone 6!!!!" What is your reaction? You probably will not go out and buy a new iPhone 6 right away. Then, what is your reaction if this friend keeps bragging about his new phone everyday? Now you may be more likely to purchase one as well after keeping seeing the great features of the new phone. The classics IC and EIC model will not able to simulate this situation.

Static in EIC model means that the probability of propagation does not change with time. The probability of propagation remains the same throughout the whole process of influence propagation.

In this paper, I propose a new improvement on the EIC model to take into consideration of continuous influence, and also the dynamic nature of influence propagation. In this Continuous Dynamic Extended IC (CDE-IC) Model, an active node will keep propagating influence to its neighbor nodes until there are no more neighbor inactive nodes or the process of influence propagation stops. Furthermore, the probability of propagation between two nodes will change with time. In my experiments, I will study the CDE-IC model in two steps. In the first step, I extend the EIC model to Continuous Extended IC (CE-IC) model that studies the continuous influence of an active node. In the second step, I will improve the CE-IC model to CDE-model, in which the probability of propagation will change with time.

The process of influence propagation in CDE-IC model is shown in Figure 9. Suppose all edges in Figure 9 have a probability of ½. At time 0, node A is selected as a seed node. At time 1, node B turns active under node A's influence. At time 2, node B propagates influence to its neighbor node C and node E. At time 3, although node A fails to influence node D at time 0, it gets another opportunity and succeeds. Node D becomes active, so does node F. The influence process stops at time 3.

The key difference between CE-IC Model with classic IC Model or EIC model is that an active node will have multiple chances to propagate its influence. If it fails the first time, it still has a second or a third chance to propagate. In CE-IC model, all activated nodes need to be considered in each step.

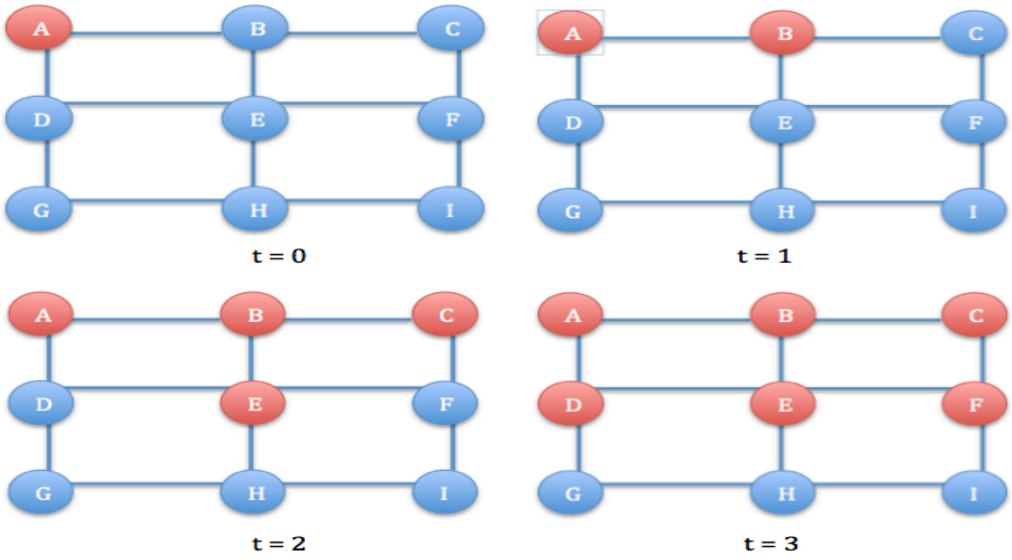


Figure 9: Influence propagation in CDE-IC model

Optimization of Seed-node Selection Algorithms

Efficiency and quality are two metrics to measure seed-node selection algorithms. The Hill-Climbing greedy algorithm and its improvements can provide guaranteed

approximation solution to optimal seed node selection, but they are not efficient enough to scale to large social networks. Traditional heuristic algorithms such as Degree-Centered algorithm and Distance-Centered algorithm are efficient in seed nodes selection, but they do not have comparable results to Hill-Climbing greedy algorithm. Chen et al. have shown that a properly improved heuristic algorithm can achieve a comparable result to greedy algorithms, but with several orders of magnitude faster[6].

In my paper, I propose a new improved heuristic algorithm called Zero-Discount. I will test four algorithms, Random, Distance-Centered, Degree-Centered and Zero-Discount in the seed-node selection phase. Distance-Centered and Degree-Centered algorithms are discussed in previous sections. Random algorithm is selecting seed nodes uniformly and randomly from a given social network. Zero-Discount is an improved Degree-Centered algorithm inspired by the work from Chen et al[6].

In Single-Discount algorithm, the degree of an inactive node is decreased by one each time one of its neighbors turns into an active status. The reason the degree of an inactive node is discounted is because when a neighbor of an inactive node is selected as a seed node, the inactive node is no longer as influential as before, at least it will not affect its newly affected active neighbor node. The reason of Degree-Centered method cannot behave as well as greedy algorithms in quality is that there is a high degree of overlapping between high degree nodes. If we only select nodes with the highest degree into the seed nodes set, they will greatly weaken each other in the process of influence propagation.

To extremely exploit the fact of overlapping between high degree nodes, I propose Zero-Discount algorithm. In this algorithm, if a node is selected as a seed node, the degree of all of its out-neighbor nodes will be set to zero. This assumption is reasonable especially in my CDE-IC model. Because the propagation process of my model is continuous, an active node will keep trying to propagate its influence to its neighbor nodes. If the probability of propagation between two nodes is p, it only needs 1/p on average attempts to succeed. Suppose the probability of propagation is 0.2, an active node needs an average of five times to succeed. In this condition, we can safely remove all out-neighbor nodes of a seed node without impairing performance significantly. In other hand, evenly distributed seed nodes increase the chance to propagate influence in social networks.

Estimation of Propagation Probability

In CDE-IC Model, the propagation probability between two nodes is determined by the spreading probability and the adopting probability. The probability can be estimated by three methods. The first method is to set the probability to a uniformed random value from [0,1], or a fixed value like ½. This method is used when it is hard or impossible to know the probability. However, setting the probability to a random fixed value does not reflect any property of real social networks.

The second method is to use data mining techniques to analyze past record in social networks. For example, we can analyze all the records between individuals on Twitter. By analyzing the historical logs in previous events, we can get an idea of how nodes interact with each other, thus a more precise probability value can be assigned.

This method has been well discussed previously[13]. However, this method is costly and slow, considering the large size of logs of a popular social network.

Because of the limitation of the above two methods, I propose a third method that assigning each edge an estimated probability value by analyzing the characters of social networks. I categorize influence sources into three classes, *Star Effect*, *Peer Pressure* and *Social Trend*. Star Effect refers to the influence coming from celebrities. They are role models to society and have many followers on social networks. You keep following their updates. However, their life is very different from yours and they may not influence you as much as you are by your friends. Peer Pressure refers to the influence coming from someone closely connected. You two not only meet in social networks, but also frequently communicate in daily life. He/she could be your families or childhood friends. These people that are closely connected to you have a strong force to affect your decision. Social Trend refers to the influence coming from the whole social network. If more and more people in social networks are under certain influence, the probability of an inactive node turning into active node will also increase.

In CDE-IC model, the propagation probability is determined by the structure of social networks. Similar to EIC model, the propagation process is divided into two phases of spreading phase and adopting phase. The probability of propagation is the product of spreading probability and adopting probability, both of which are decided by the structure of social networks.

The spreading probability of a node is the tendency of the node to spread its influence to its neighbor nodes. The more active a node is in social networks, the more

likely it will spread an influence. A user who publishes hundreds of posts in Facebook is more likely to spread new information to his friends than a user who posts once a week.

The most active users can be identified by analyzing past logs in social networks. However, this process is costly and time consuming. As a simplification, we may assume the users who have the largest number of out-neighbor nodes as the most active nodes. This is a reasonable simplification. The number of friends of a person reflects his popularity in social networks. A social person is more likely to share new technologies or information with his friends.

Based on this simplification, a different spreading probability is given to each node according to its degree in social networks. High degree nodes will have higher spreading probability than low degree nodes. This is similar as Degree-Centered heuristic method in classic IC model. A maximum spreading probability p_{max} is given to the node with the highest degree ($p_{max} \leq 1$) and a lower spreading probability is given to the node with a lower degree. An isolated node in a social network will have a spreading probability of 0. The other node with degree between 0 and the highest degree will be assigned a probability value based on a linear function, either by degree number or percentage in all nodes. The spreading probability is decided as the following formula 2: $p_s = p_{min} + (p_{max} - p_{min})\ (Degree(node) - Degree_{min}) / (Degree_{max} - Degree_{min}),$ in which degrees are out-degree of a node.

The estimation of adopting probability is more complicated. I will estimate the value of adopting probability according to the source of influence. For example, if two persons, your best friends and a celebrity, try to persuade you to try a new technology,

whom will you believe more? More likely, you will take the advice from your best friends. The reason that your friend is more persuasive is because he/she spends more time/energy with you. If the energy value of each node is set to 1, this energy of 1 will be divided among all out-neighbor nodes of the node. As a high degree node has more out neighbor nodes, each neighbor node of it will only have a small fraction of its energy.

Specifically, because a node needs a series of actions to persuade its neighbor nodes, a high degree node could not give much attention to every one of its out neighbor nodes. In contrast, a node has a relative small out-degree value could spend more time with its neighbors. A friend of yours, who does not have so many connections with others in social networks, is more likely to keep talking with you and answering your questions. For that reason, you will be more likely to accept his suggestion.

The adopting probability of each edge is estimated in the following three steps. Firstly, check all in-neighbor nodes of a node. Secondly, calculate how much energy each in-neighbor node spends on it. Thirdly, normalize the total energy. The adopting algorithm is shown in Figure 10.

```
List<Integer> inNeighbors = v.getInNeighbors();
double[] inWeightWeight = new double[inNeighbors.size()];
for(int i = 0; i < inWeightWeight.length; i++){
    int uIndex = v.getInNeighbors().get(i);
    int uTotal = vertexes.get(uIndex).getTotalWeigt();
    inWeightWeight[i] = 1.0 * v.getInWeight().get(i) / uTotal;
}
double sum = 0.0;
for(int i = 0; i < inWeightWeight.length; i++){
    sum += inWeightWeight[i];
}
for(int i = 0; i < inWeightWeight.length; i++){
    v.addpAdoption(inWeightWeight[i] / sum);
}
```

Figure 10: Estimation of Adopting Probability

In addition, it has been demonstrated that the probability of influence shows an exponential decay behavior[13]. To model this decay behavior, I introduce a time factor into the estimation of adopting probability in CDE-IC model. The decay of adopting probability is decided as in the following formula 3:

$$p_a(t) = p_a(t_0)\ e^{-(t-t_0)/\tau}$$

The adopting probability of a node will decrease with time in exponential number. τ is called the mean lifetime. The initial probability $p_a(t_0)$ is the adopting probability when this inactive node is first exposed to an influence. *Pa* is maximal at this initial time. As time goes on, the inactive node will lose interest and become less likely to adopt an influence.

As discussed above, the propagation probability is estimated by the structure of social networks. The nodes with high out-degree in social networks represent celebrities. They are active in social networks and more likely to spread new influence to the public. Such nodes have a high spreading probability. However, the celebrities are not closely connected to each of his followers. When people receive influence from such nodes, the

probability of adoption is relatively low. In contrast, the nodes with low out-degree are average people in social networks. In fact, the low out-degree nodes are the main component of most social networks. Such nodes will not publish as many posts like celebrities, but their posts have a bigger influence on its connections. For example, if your friend posts in your Facebook wall, he does want to share something with you and think you may like it as well. Although such node has a relatively lower spreading probability, the adopting probability is higher.

Social Trend also has an influence on the propagation probability. When a new technology was first introduced to the public, it is difficult for people to accept it immediately. As more and more people accept the new technology, the propagation probability will also increase with time. The effect of Social Trend can be estimated by the percentage of active nodes in a social network. The effect of Social Trend will also be added into the estimation of propagation probability.

Data Sets

In order to test algorithms on the CDE-IC model, I choose two real world data sets from www.arXiv.org. The data sets consist of academic collaboration networks that are believed to be a good simulation of social networks. In this data set, each node is an author and an edge between two nodes means there is collaboration between two authors, i.e. co-authorship on a paper. If there are more than one collaborations between two authors, the edge is given a higher value. If there are more than two co-authors in one paper, there will be edges between any two of the authors. I choose two different sized

data set, High Energy Physics (hep) data set and Physics (phy) data set, in my experiments. hep has 15k nodes and 59k edges, and phy has 37k nodes and 231k edges.

EXPERIMENTS

In this paper, I study how to maximize speed of influence propagation in social networks. I compared four heuristic methods, including three popular methods (Random, Distance-Centered, and Degree-Centered) and my improved Zero-Discount method on the Continuous Dynamic Extended IC (CDE-IC) Model. I evaluated the results on two large academic collaboration data sets obtained from www.arXiv.org.

I will study the CDE-IC model in two steps. In the first step, I extend the EIC model to Continuous Extended IC (CE-IC) model that studies the continuous influence of an active node. In the second step, I will improve the CE-IC model to CDE-IC model, in which the probability of propagation will change with time. The propagation probability is the product of spreading probability and adopting probability, which are estimated based on structure of social networks (see section of Estimation of Propagation Probability)

In order to study the speed of influence propagation in social networks, the process of influence propagation will be discrete, which means each active node can only propagate influence one step further in a unit time (a second/an hour/a day). Thus, an active node can only affect it's directed out-neighbors in a unit time. By recording how many nodes are affected in each step, we know both the instant speed at each step and the average speed of the whole process in influence propagation. Each method is run 100 times to obtain an average result. I compare the running time in seed nodes selection phase and influence propagation speed to analyze the efficiency and quality of each method.

Seed-node Selection Algorithms Comparison in CE-IC Model

I use two data sets (hep and phy) obtained from www.arXiv.org to test my CDE-IC model and novel Zero-Discount heuristic algorithm. These data sets are academic collaboration results that are believed to be a good simulation of social networks. Each node in the data sets represents an author, and two of co-authors of a paper are connected by an edge. If they have co-authored more than one paper, a different weight value is given to edge. If there are more than two co-authors in one paper, any two of them are connected by an edge. There are 15233 nodes and 58891 edges in hep data set, and 37154 nodes and 231584 edges in phy data set.

Comparison of Efficiency

Efficiency and quality are two metrics to evaluate seed-node selection algorithms. The efficiencies of four algorithms (Random, Distance-Centered, Degree-Centered, and Zero-Discount) are analyzed by comparing the running time in seed-node selection phase. Seed nodes size is 20 and the running time unit is millisecond.

Table 1. Running time of Seed-node Selection Algorithms (ms)

	hep	**phy**
Random	1	1
Distance-Centered	13613	186193
Degree-Centered	45	70
Zero-Discount	143	230

Random method is the fastest among the four methods. As Random algorithm does not consider any characters of social networks, the running time of Random selection is only related to size of seed nodes, and is not affected by number of nodes and edges in a social network. One approach to randomly select seed nodes is to randomly sort the nodes first, and then select the first k nodes as seed nodes. The running time of this approach is O (m). Alternatively, considering the seed node size is very small comparing to the number of total nodes, we may also just randomly select a node and put it in the seed nodes set. If the node is already in seed node set, we can simply select another node until we have k nodes from the social networks. Random algorithm of this approach is shown in Figure 11. The running time is O(k).

```
S = ϕ
While S.size < k
        nextIndex = Random(1, m)
        S.add(nextIndex)
```

Figure 11: Random Seed-node Selection Algorithm

Distance-Centered is the slowest in these four heuristic methods and does not scale well with large data set. The number of nodes, and more so the number of edges, greatly affects the efficiency of Distance-Centered method. This method has to select each node as a root in social networks to calculate the average distance to all other nodes. If two nodes are directly connected, the distance is one. If the root cannot reach another node, the distance between them is marked as n (n is the total number of edges). A

distance with value n is the maximum possible value between any two nodes. It takes O(n) for a root to reach all other nodes. The total time to calculate average distance of all nodes is O(mn). We can use a quick selection algorithm to find k seed nodes with the smallest average distance to all other nodes in social networks, which takes O(m). So the total running time of Distance-Centered selection algorithm is O(mn). The Distance-Centered algorithm is shown in Figure 12.

```
i = 1
While i ≤ m
        Select node(i) as root
        Calculate average_distance(node(i))
Quick_selection(V, k)
```

Figure 12: Distance-Centered Seed-node Selection Algorithm

Degree-Centered algorithm is much more efficient than the Distance-Centered algorithm. Time complexity of is O(m + n), much efficient than Degree-Centered algorithm. In Degree-Centered algorithm, each node does not need to reach all other nodes in social networks, as in Distance-Centered algorithm.

My novel Zero-Discount heuristic algorithm is an improvement over the Degree-Centered algorithm. It runs slower than Degree-Centered algorithm to check more nodes, but it is still much efficient than Distance-Centered algorithm. In Zero-Discount algorithm, if a node is selected as a seed node, the degree of all its out-neighbor nodes

will be set to zero. Actually, there is no need to sort the nodes according to out-degree again. Instead, all out-neighbor nodes of a seed node is set to zero degree and put to the end of queue at each step. Since the size of seed nodes is much smaller than the total number of nodes, we can maintain a heap of size k. Each time we select a new node, we will check if it has a larger degree than the nodes in the heap, and also if it is an out-neighbor of any node in the heap. The time complexity of Zero-Discount is O (mlgk+n). The Zero-Discount algorithm is shown in Figure 13.

```
minDegree = m
heap = {}
for each node in V
        if heap.size < k, put current node into heap
                reset minDegree if needed
        else if out-degree(current node) < minDegree
                discard current node
        else if current node is out-neighbor of any seed nodes
                discard current node
        else
                put current node in heap
                reset minDegree if needed
```

Figure 13: Zero-Discount Seed-node Selection Algorithm

In summary(as Table 2), Random algorithm is the most efficient algorithm. Distance-Centered algorithm is the least efficient algorithm since it needs to reach as

many other nodes as possible from each node. My novel Zero-Discount algorithm is slower than Degree-Centered algorithm, but significantly faster than Distance-Centered algorithm.

Table 2. Time Complexity of Seed-node Selection Algorithms

Algorithm	Time Complexity
Random	O(k)
Distance-Centered	O(mn)
Degree-Centered	O(m+n)
Zero-Discount	O (mlogk+n)

The experiments in seed-node selection phase show that my novel Zero-Discount algorithm has a comparable efficiency to the Degree-Centered algorithm, and significantly better efficiency than Distance-Centered algorithm. Zero-Discount is about 100 times faster than Distance-Centered method. It can also be scaled up to larger social networks.

Comparison of Quality

In this section, I will compare the quality of seed-node selection algorithms by analyzing how fast influence propagate in a social network. I run experiments on two data sets with seed nodes of size 20 and 50 respectively. Each experiment will run in 40 units time. The total number of active nodes at the end of step 40 reflects the average

speed over the process of influence propagation. The number of newly activated nodes in each step reflects the instant speed of influence propagation.

In the first step, I extend the EIC model to Continuous Extended IC (CE-IC) model that studies the continuous influence of an active node. In the second step, I will improve the CE-IC model to CDE-model, in which the probability of propagation will change with time. In addition, a correction factor to propagation probability will be introduced to take into account the effect of Social Trend.

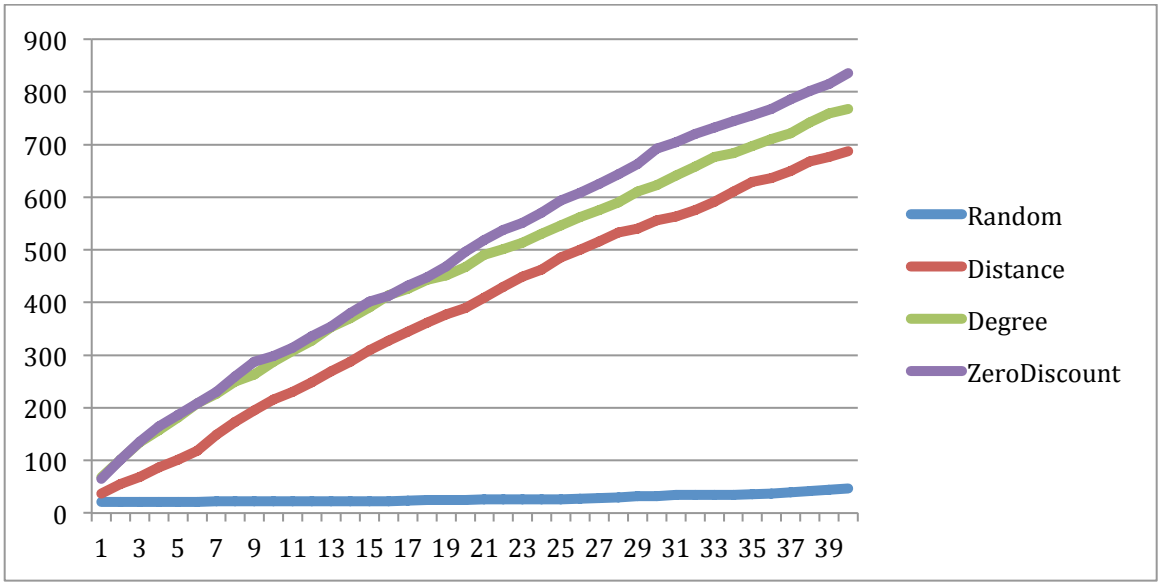


Figure 14: Average Propagation Speeds in CE-IC Model (hep k=20)

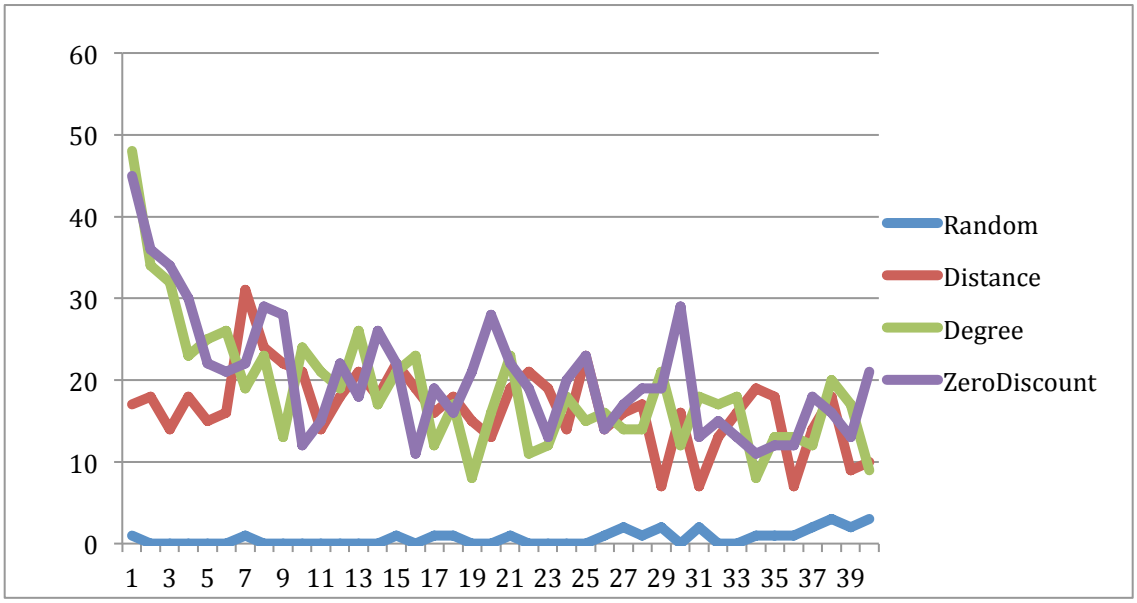


Figure 15: Instant Propagation Speeds in CE-IC Model (hep k=20)

Comparison of the average speed of influence propagation using four heuristic algorithms is shown in Figure 14. The size of seed node is 20. I used hep data set to test

the four algorithms in CE-IC model. As expected, Random algorithm has the slowest average propagation speed because Random algorithm does not consider any characters of social networks. Although randomly selecting seed nodes is fast in the seed node selection phase, this method is useless in promoting influence propagation. Degree-Centered is regarded as the best heuristic algorithm and is widely used. It is about 15% faster than Distance-Centered algorithm (Figure 14). My Zero-Discount algorithm has the fastest average propagation speed among the four algorithms. Although Zero-Discount algorithm is slower in the seed node selection phase than Degree-Centered algorithm, it increased the propagation speed by 9%.

The instant propagation speed is shown in Figure 15. Random selection is still the slowest with an instant speed of close to 0 at most steps. There are two important features in this graph. Firstly, both Degree-Centered and Zero-Discount algorithms have significantly fast instant speeds at the initial steps of influence propagation. This initial fast instant speed is generated by inclusion of high degree nodes in the seed nodes and is the major contributor to the fast average speeds of Degree-Centered and Zero-Discount algorithms seen in Figure 14. At the later steps of influence propagation, the instant speed of Degree-Centered algorithm declined and was similar to Distance-Centered algorithm.

Secondly, although the instant speeds of Zero-Discount algorithm also declined at later steps, there were several minor spikes during influence propagation such like data point 20 and data point 30. The difference in instant propagation speed between Degree-Centered and Zero-Discount algorithm is that Degree-Centered algorithm only selects high degree nodes, and there is a large overlapping between the high degree nodes. As

the influence propagation proceeds, these high degree nodes add little additional value to the propagation process. In contrast, Zero-Discount algorithm tries to diversify seed nodes selection by eliminating neighbor nodes of already selected seed nodes, and adding high degree nodes from isolated sub-networks as seed nodes. As influence propagation proceeds, these small isolated sub-networks can be connected. That is why Zero-Discount algorithm performs better than Degree-Centered algorithm at the later steps of influence propagation.

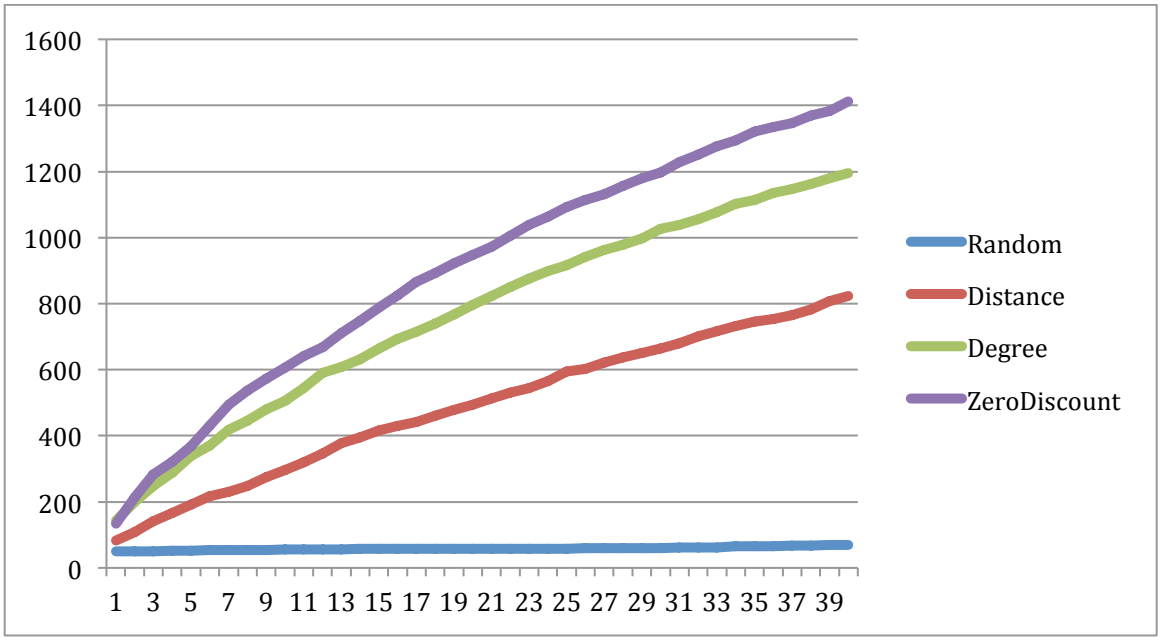


Figure 16: Average Propagation Speeds in CE-IC Model (hep k=50)

Next, I explored how seed-node size affects the speed of influence propagation. Seed-node size of 50 was used for the same data set hep. The average propagation speeds of the four algorithms are shown in Figure 16. Obviously, increase of seed node size significantly increased propagation speed of all algorithms (Figure 16 vs. Figure 14). Similar to results obtained with a seed-node size of 20, Zero-Discount algorithm has the

fastest average propagation speed among the four algorithms. In addition, the average propagation speed of Zero-Discount algorithm is about 18% faster than Degree-Centered algorithm, even better than the result with the seed node size of 20. This result indicates that the drawback of Degree-Centered algorithm is more prominent with increasing of seed-node set. The Zero-Discount algorithm has more advantage over Degree-Centered algorithm in larger seed-node selection.

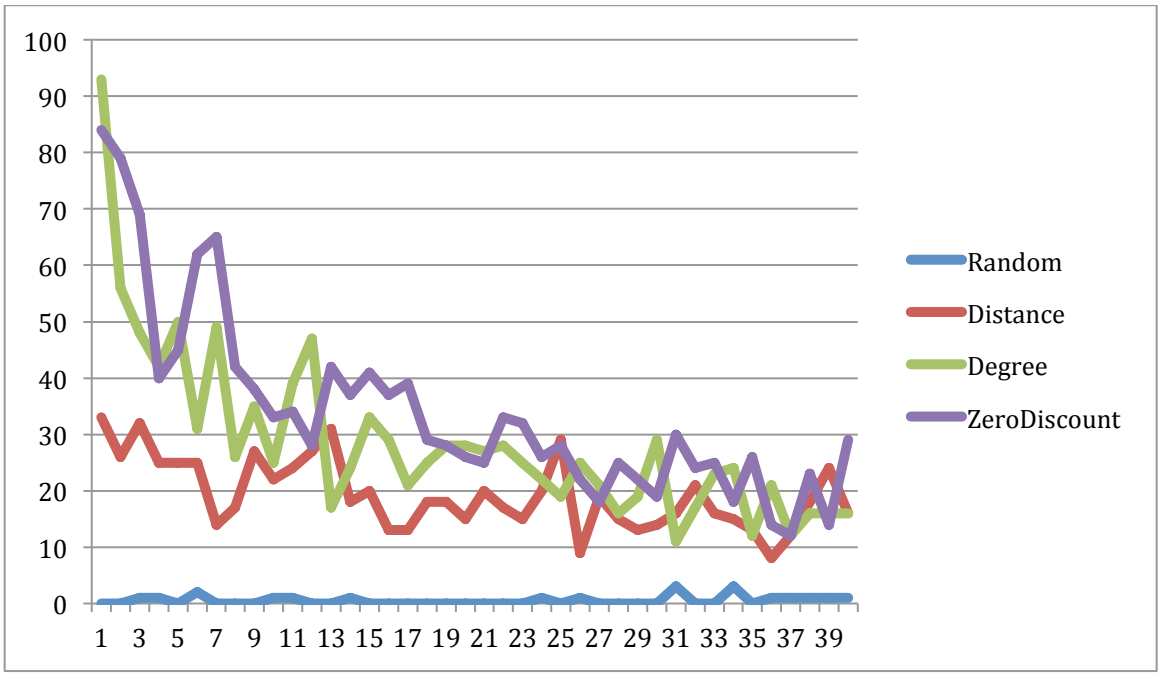


Figure 17: Instant Propagation Speeds in CE-IC Model (hep k=50)

The instant propagation speed is shown in Figure 17. It is more obvious that even Zero-Discount loses more speed at the first step, however, Zero-Discount can almost beats Degree-Centered in the following steps.

To test if the faster propagation speed of Zero-Discount algorithm is also true for other data sets, similar experiment were run in a larger data set, the phy data set. The

average and instant propagation speeds are shown in Figure 18 and Figure 19, respectively. Consistently, Zero-Discount algorithm has the fastest average and instant propagation speeds among all four algorithms. Distance-centered algorithm performed much worse in this experiments. One reason of this low performance could be that the selected seed nodes have a lower propagation probability.

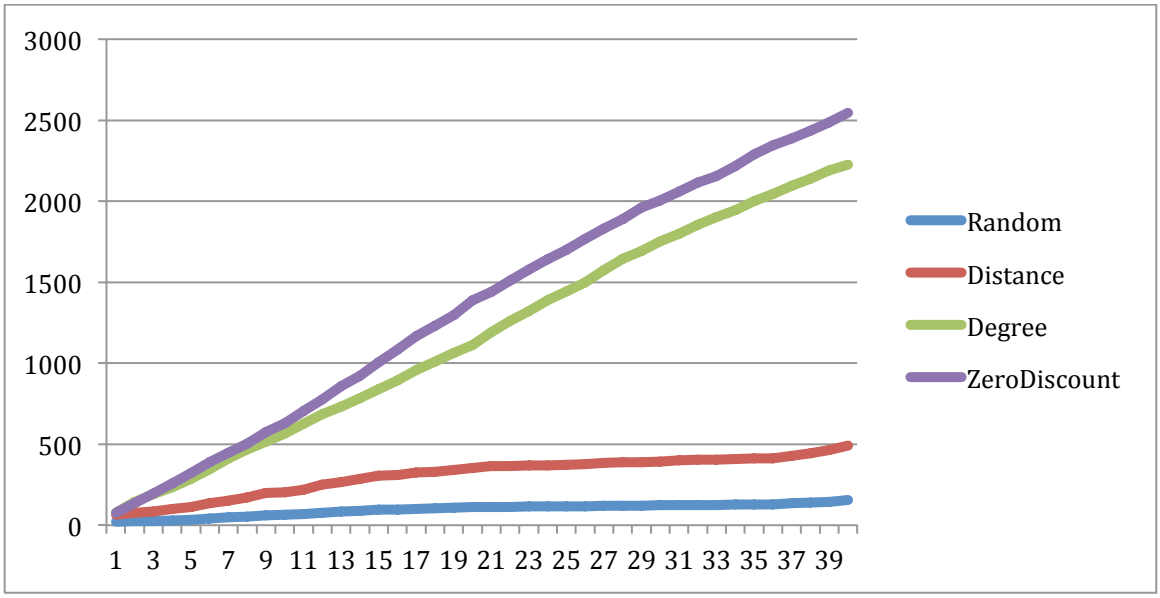


Figure 18: Average Propagation Speeds in CE-IC Model (phy k=20)

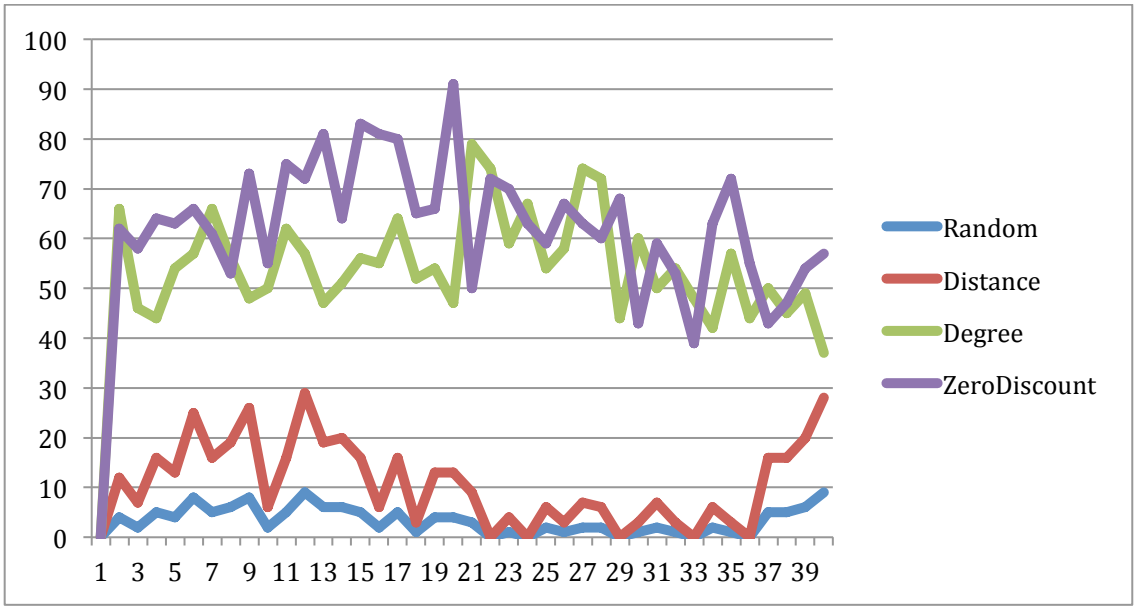


Figure 19: Instant Propagation Speeds in CE-IC Model (phy k=20)

Seed-node Selection Algorithms Comparison in CDE-IC Model

In the above experiments, Zero-Discount algorithm always has the fastest propagation speed among the four heuristic algorithms in the CE-IC model. Next step, I will add dynamic property into model to test if Zero-Discount is still the best in CDE-IC model when propagation probability changes with time. According to formula 3, the adoption probability of a node will decrease with time in exponential number. I set the half time period to 20, which is half of experiment units. I will also record at which step each active node tries to propagate influence at initial time for relative edges. I will test CDE-IC model with seed nodes set of size 50.

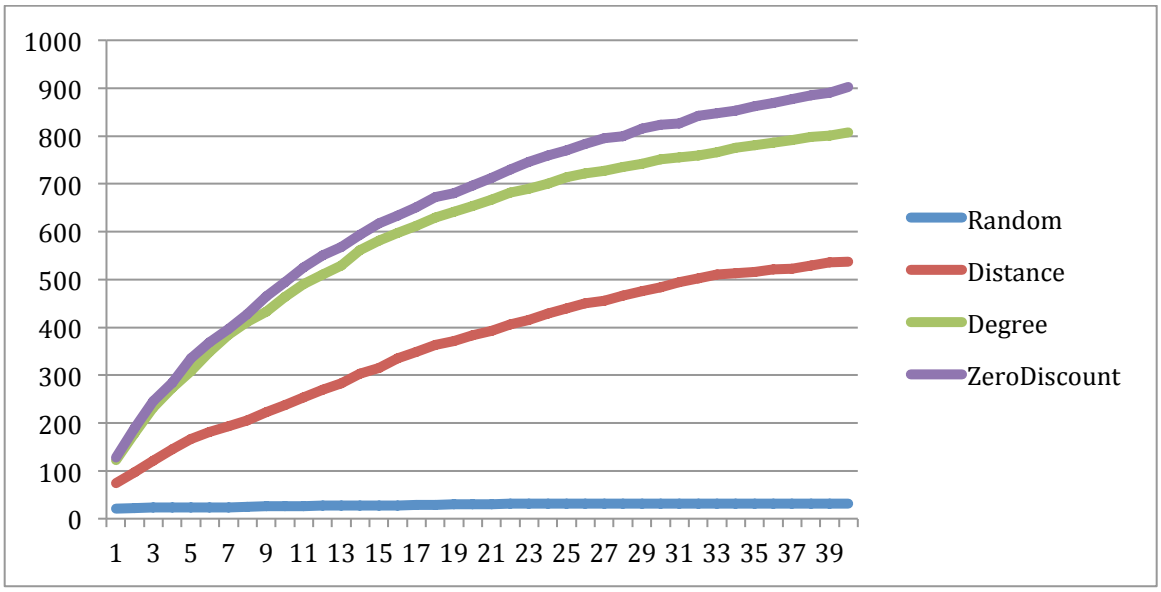


Figure20: Average Propagation Speeds in CDE-IC Model (hep k=50)

In Figure 20, we can infer similar conclusion as in CE-IC model. We can also see clearly the effect of exponential decreasing in adoption probability. First, the total

number of affected nodes in CDE-IC model is less than CE-IC model with the same conditions. In CE-IC model, Zero-Discount method can reach 1413 nodes on average. However, in CDE-IC model, only 902 nodes on average are under influence, which is about 1/3 less. Second, the curves in CE-IC model is more like straight line, which means the newly affected nodes is increasing at steady pace. Contrastingly, we can see the decrease in trend of the curve in CDE-IC model.

Both CE-IC model and CDE-IC model show that my novel Zero-Discount can always beat all the other heuristic methods with best quality of seed nodes set. Using my Zero-Discount method, we can expect a higher speed of influence propagation in social networks.

Finally, we can add a small correction factor to probability if considering Social Trend. However, this value is significant only when a large part of the social networks is under influence. Since we are studying speed of influence propagation, which is more meaningful in a short period, when not so many nodes under influence, we can safely ignore it.

CONCLUSION

In this paper, I studied how to maximize speed of influence propagation in social networks. I proposed a new Continuous Dynamic Extended IC (CDE-IC) Model, which is an improved modification of the Extensional Independent Cascade (EIC) Model. The original EIC model has two drawbacks: first, an active node can only try to propagate its influence to its neighbors once; second, the propagation probability does not change between nodes. Both of these problems are solved in my CDE-IC model.

I ran four algorithms in CDE-IC Model, Random, Distance-Centered, Degree-Centered and my novel Zero-Discount method. Experiments on two data sets with different sizes of seed nodes all showed that my Zero-Discount method performed better than any other heuristic methods. The previous best method, Degree-Centered, was 9%-18% slower than my method.

In order to give a reasonable probability between two nodes, I tried to recognize the source of influence. Depending on whether the influence is from Peer Pressure or Star Effect, I calculated the probability differently, reflecting the structure of social networks.

FUTURE WORK

Future directions in the study of maximizing of speed of influence in social networks may include: First, I only studied how to extend a classic IC model. It can also be extended LT model. Second, in the current model, a node can only change from inactive status to active status. It can be modified to allow changes in both directions to model a negative effect. Third, greedy algorithm can be compared with heuristic algorithm.

REFERENCE

1. Domingos P, Richardson M: **Mining the network value of customers**. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* San Francisco, California: ACM; 2001: 57-66.
2. Richardson M, Domingos P: **Mining knowledge-sharing sites for viral marketing**. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* Edmonton, Alberta, Canada: ACM; 2002: 61-70.
3. Kempe D, Kleinberg J, #201, Tardos v: **Maximizing the spread of influence through a social network**. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* Washington, D.C.: ACM; 2003: 137-146.
4. Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N: **Cost-effective outbreak detection in networks**. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* San Jose, California, USA: ACM; 2007: 420-429.
5. Goyal A, Lu W, Lakshmanan LVS: **CELF++: optimizing the greedy algorithm for influence maximization in social networks**. In: *Proceedings of the 20th international conference companion on World wide web.* Hyderabad, India: ACM; 2011: 47-48.
6. Chen W, Wang Y, Yang S: **Efficient influence maximization in social networks**. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* Paris, France: ACM; 2009: 199-208.
7. Granovetter M: **Threshold Models of Collective Behavior**. *The American Journal of Sociology* 1978, **83**(6):23.
8. Schelling TC: **Miromotives and Macrobehavior**. 1978.
9. Jacob Goldenberg BL, Eitan Muller: **Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth**. *Marketing Letters* 2001, **12**(3):12.
10. Wang Z, Qian Z, Lu S: **A probability based algorithm for influence maximization in social networks**. In: *Proceedings of the 5th Asia-Pacific Symposium on Internetware.* Changsha, China: ACM; 2013: 1-7.
11. Chen WL, Wei;Zhang Ning: **Time-Critical Influence Maximization in Social Networks with Time-Delayed Diffusion Process**. 2012.
12. Gerard Cornuejols aLF, George L Nemhauser: **Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms**. *Management Science* 1977, **23**(8):21.

13. Goyal A, Bonchi F, Lakshmanan LVS: **Learning influence probabilities in social networks**. In: *Proceedings of the third ACM international conference on Web search and data mining.* New York, New York, USA: ACM; 2010: 241-250.