Probabilistic Diffusion of Social Influence with Incentives
Myungcheol Doo, Ling Liu Member, IEEE,

Abstract—With explosive growth of social media, social computing becomes a new IT feature. A core functionality of social computing is social network analysis, which studies dynamics of social connectivity among people, including how people influence one another and how fast information diffuses in a social network and what factors stimulate influence diffusion. One of the models for information diffusion is the heat diffusion model. Although it is simple in capturing the basic principle of social influence, there are several limitations. First, the uniform heat diffusion is no longer hold in social networks. Second, high degree nodes are most influential in all contexts is not realistic. In this paper we propose a probabilistic approach to model social influence diffusion model with incentives. Our approach has three features. First we define an influence diffusion probability for each node instead of uniform probability. Second, we categorize nodes into two classes: active and inactive. Active nodes have chances to influence inactive nodes but not vice versa. Third, we utilize a system defined diffusion threshold to control how influence is propagated. We study how incentives can be utilized to boost the influence diffusion. Our experiments show the reward-powered model is more effective in influence diffusion.

Index Terms—Social Influence, Diffusion of Information, Rewards and incentives, Heat Diffusion Model, Probabilistic Model

1 INTRODUCTION

Social network analysis research can be broadly classified into two categories: (i) applying and extending existing graph mining and machine learning algorithms to social network datasets to predict and mine features of statistical significance and (ii) developing innovative social computing-specific algorithms that can derive new insights and new values that traditional general purpose data mining algorithms may fail to deliver. We argue that social influence analysis falls into the second category and it studies the diffusion of influence and how the ideas and influences are spread and propagated through a social network. For example, the diffusion of medical innovation or the sudden spread of viruses and contagious diseases has been studied in the bioinformatics and health science domain. The effect of “word of mouth” has been studied in business marketing, and the pollution propagation in the water networks has been studied in technological settings.

Heat diffusion model [6], [25], [26], [37] is used to model and analyze social influence. Although the heat diffusion kernel is simple and straightforward in capturing the basic principle of social influence among a social network of people, there are several serious limitations of using heat diffusion to model social influence among people.

The first limitation of using heat diffusion kernel for modeling social influence is the uniform influence distribution assumption [26]. Every node receives equal amount of heat (influence) from its neighbor nodes having higher heat (influence) values, and then it propagates its heat uniformly to each of its neighbor nodes having lower heat values. The amount of heat distributed to its neighbors is computed by the out-degree of the node. For example, node \( v \) has 5 neighbors and the current heat value is \( h_v \), then \( v \) distributes \( \frac{h_v}{5} \) to each adjacent node. However, all friends of \( v \) are not equal. Some are best friends while others may be acquaintances. Edges between two nodes should be weighted by some measurable factors and the weights should be applied to the influence(heat) diffusion computation.

The second limitation is the assumption that high degree nodes are always more active in heat diffusion process. In the context of social influence, it is recognized that high degree nodes are not always active in posting articles or interacting with neighbors. Some people report that they cannot reject friend requests from their clients [34] or they accept friend requests from even vaguely recognized people [8] to increase their popularity.

Third but not the least, we argue that the heat diffusion based social influence model is not suitable to study whether and how incentives may stimulate the rate and the coverage of influence diffusion in a social network. All heat diffusion models used in [6], [25], [26], [37] consider only number of edges to compute influence. However, in a viral marketing, the number of edges is not the only factor to consider. People promote new products to their friends due to either explicit incentives such as financial incentives or simply a desire to share benefits of products with friends. Good examples are ICQ and PayPal’s marketing strategy. ICQ gives a user an option to invite user’s friends while PayPal gives monetary incentives for viral marketing and both strategies have worked well [9].

Bearing these issues in mind, in this paper we present a probabilistic approach to model social influence diffusion with multi-scale rewards as incentives. Our probabilistic diffusion approach has three unique features. First, we distinguish active nodes from inactive nodes. Active nodes represent people who adopted new products while inactive nodes represent people who did not adopt yet. Active nodes have chances to influence inactive neighbor nodes but not vice versa. If active nodes succeed in activating inactive nodes, then newly activated nodes have also chances to activate their inactive nodes, which is the same as viral marketing. Second, we compute an influence diffusion probability for each pair of nodes, which can differentiate nodes that have higher level of interactions and higher node degree from those nodes that have low or zero interactive activities. Third but not the least, we utilizes a system defined diffusion threshold, combined with the pair-wise diffusion probability, to control and manage how influence is propagated across the social network of \( n \)
nodes. Based on this probabilistic diffusion model, we formally study how incentives can be utilized as stimuli to further boost the influence diffusion rate and coverage. For each node in a social network, we compute its probability-based social influence ranking score, which is measured by the approximate influence coverage of this node. Our experiments show that the reward-powered social influence model is more effective in terms of both diffusion rate and diffusion coverage of influence.

2 Influence Diffusion: An Overview

Given that our probabilistic social influence model is an enhancement of both heat diffusion model [26] and stochastic influence models [16], in this section we briefly describe the basics of both heat diffusion model and stochastic influence model, and then outline the design principle of our approach. We will present our probabilistic social influence model (PSI) and our algorithm for computing influence rank in Section 3 and Section 4 respectively.

2.1 Heat Diffusion Kernel

Given a social graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$, the heat diffusion model diffuses heat by following four basic rules:

i. Heat transfers from a source node to its connected neighbor nodes and the amount of $v$’s heat at time $t$ is $H_v(t)$; 

ii. At time $t$, an amount of heat diffused from $v$ to its neighbors during $\Delta t$ is $DH_v(\Delta t)$; 

iii. At time $t$, an amount of heat $v$ received from its neighbors during $\Delta t$ is $RH_v(\Delta t)$; and  

iv. The heat difference of node $v$ during $\Delta t$ (from $t$ to $t + 1$) is $H_v(t) - H_v(t + 1) = RH_v(\Delta t) - DH_v(\Delta t)$.

Thus, heat diffusion equation at time $t$ for $n$ nodes is computed as follows [26]:

$$
H(t) = e^{\alpha t}K H(0)
$$

$$
= (I + \alpha t K + \frac{\alpha^2 t^2}{2!} K^2 + \frac{\alpha^3 t^3}{3!} K^3 + \ldots) H(0)
$$

$$
= I + \sum_{n=1}^{\infty} \frac{\alpha^n t^n}{n!} K^n H(0)
$$

(1)

where $H(0)$ is a $n$ by 1 column matrix that has heat values of $v \in V$ at time 0 and $\alpha$ is a heat conductivity, and $K$ is $n$ by $n$ matrix defined in Eq. 2.

$$
K(i, j) = \begin{cases} 
1 & \text{if } (v_j, v_i) \in E \\
\frac{1}{d_i} & \text{if } i = j \text{ and } d_i > 0 \\
-1 & \text{if } i = j \text{ and } d_i = 0 \\
0 & \text{otherwise}
\end{cases}
$$

(2)

where $d_i$ denotes the out-degree of vertex $v_i$.

In this heat diffusion model, given an active heat source node $v_s$, we compute $H(t)$ and see how many nodes have heat value larger than a system defined value $\theta$. If heat value of inactive node $v_t$ is greater than $\theta$, then we consider $v_t$ is activated by $v_s$. $H(t)$ is computed in three steps:

1) Set every node inactive by $H(0) = 0$;

2) Select a node $v_i$ that is not visited and mark as visited then set $(i, 1)$th element of $H(0)$ as amount of heat of $v_i$;

3) Compute $H(t)$ using Eq. 1.

Figure 1(a) shows a network of 10 nodes. Each edge has a weight computed by $\frac{1}{d_i}$. $v_9$ has two out edges, thus $E(v_9, v_10)$ has weight 0.5. $v_10$ has only one edge, $E(v_10, v_1)$, with weight 1. Figure 1(b) shows the amount of heat after $t$ period of diffusion from $v_5$ for this example social network. We observe that for nodes $v_1$ and $v_6$ that are one-hop away from the heat source (black lines), their heat values increases in the beginning at the same rate because two nodes have the same edge weight to $v_5$. In Figure 1(a) $v_1$ and $v_6$ receive the same heat from $v_5$ and thus $v_6$, a line with circle symbol, and $v_4$, a line with a plus symbol, have the same diffusion pattern over the duration. Also heat values of two nodes start decreasing at $t = 5$ differently because they have different number of 2-hop away nodes. For nodes $v_1$, $v_2$, and $v_7$, which are 4-hop away from the heat source (blue lines), their heat values continue to increase until $t = 10$. For nodes that are 2-hop away (red lines) or 3-hop away $(v_{10})$ from the heat source, their heat values increase slowly and gradually until $t = 10$ due to the topology. We call this heat diffusion model a topology-based approach because its kernel only concerns the node degree and the topological connectivity.

2.2 Stochastic Influence Diffusion Models

In the heat diffusion model, once a heat source is selected and values of parameters are set, the result is the same for every experiment. In contrast, the stochastic diffusion process involves some non-determinacy. Familiar examples of processes modeled as stochastic time series include stock market and exchange rate fluctuations, speech and video signals. The two basic stochastic models used to gauge social influence are Independent Cascading model (ICM) and Linear Threshold model (LTM). Both models use a directed graph $G = (V, E)$, where $V$ is a set of vertices representing users and $E$ is a set of edges expressing friendship relationship. A node $u$ is called active if $u$ adopts information from her friends, otherwise it is inactive. Initially all nodes are inactive.

Independent Cascading Model (ICM). ICM [16], [17], [21] takes advantages of user interaction. Each relationship represented by $E(u, v)$ has a probability for $u$ to activate $v$, denoted by $p_{u,v}$, which is assigned randomly by the system. When a node $u$ first becomes active at time $t$, it is given a single chance to activate its inactive neighbor $v$ with
probability $p_{u,v}$ at time $t+1$. For example, a coin with biased probability $p_{u,v}$ that is likely to turn up heads is tossed. If the output is head, then we consider $u$ is activated. If $v$ is activated by $u$ at time $t+1$, then $v$ also has one chance to activate its inactive neighbors at time $t+2$. This iterative process is started with an initial set of active nodes and stops when there is no more activation possible.

We argue that using random probability is unrealistic to study social influence in real world social networks. One important challenge is how to design a stochastic influence diffusion model that is based on meaningful attributes and relationships of social network nodes, such as different types and volumes of social interactions, which are critical to the stochastic influence diffusion process, to define the computation of the probability $p_{u,v}$.

**Linear Threshold Model (LTM).** LTM [7], [18], [27], [33], [35] considers that each node’s tendency to be active increases monotonically as more of its neighbors become active. For example, the more friends of Alice buy new iPhones, the higher desire Alice will have for buying an active. For example, the more friends of Alice buy new iPhones, the higher desire Alice will have for buying an active. Thus, at some point, $v$’s active neighbors may reach to a level of influence that can trigger an inactive neighbor, $v$, to be active. This is called node-specific threshold, $\theta_v$. In all existing LTM, this threshold is typically assigned randomly. A node $v$ is activated by each neighbor $u$ according to $w_{u,v}$ such that $\theta_v \leq \sum_{v \in E(u,v)} w_{u,v} \leq 1$, and $0 \leq w_{u,v} \leq 1$. At time 0, only one node $u$ is active and the rest are inactive. A neighbor node $v$ is activated by $u$ if the sum of the weights of all $v$’s is greater than or equal to $\theta_v$, $\sum_{v \in E(v,u)} w_{u,v} \geq \theta_v$. Thus, $\theta_v$ is a system-supplied threshold parameter in LTM. If $\theta_v$ is small, then the tendency of activation is high. For example, an iPhone5S should have a lower $\theta_v$ than the previous generation of iPhone.

Although LTM captures the tendency of information propagation, it fails to consider the similarity between a pair of nodes. Furthermore, the use of randomly generated threshold to compute the influence diffusion is inadequate. Instead, we should use the probabilistic influence diffusion model that can incorporate the dynamics of both node degree and the amount of recent and past activities as well as the rate of the information to be diffused across the network.

Furthermore, none of the existing diffusion models have studied whether and how incentives may be incorporate to create stimuli for the diffusion of influence to a broader coverage of the network at a faster rate.

Bearing the above problems in mind, we develop a probabilistic influence diffusion approach, powered with rewards as incentives.

### 3 Probabilistic Social Influence Model

We present the basic design of our Probabilistic Social Influence model (PSI), which is designed by combining the best of both ICM and LTM while removing the limitation of each. Comparing with the heat-diffusion model, our PSI model has introduced probability instead of deterministic management of influence diffusion over the given topology of social network.

First, we argue that the edge weight between a pair of nodes $u$ and $v$ should be probabilistic in nature but the probability should not be randomly assigned. Instead, the probability of the edge weight should reflect the dynamic interactions between $u$ and $v$ such that the more interactive activities from $u$ to $v$ should result in the higher weight on the edge from $u$ to $v$ and thus the higher probability of $u$ diffusing its influence to $v$.

Formally, given a social network graph $G = (V,E)$ where $V$ is a set of nodes and $E$ denotes a set of directed edges, each node $u \in V$ has node attributes, such as the number of non-interactive activities, $NA(u)$. Nodes are either inactive or active. Initially all nodes are inactive. If a node $u$ has performed some interactive activities with $v$, then an edge $E(u,v)$ is created from $u$ to $v$ with the edge weight defined by the number of interactive activities from $u$ to $v$, $IA(u,v)$. Based on this collection of information we compute the probability, $w(u,v)$, for $u$ to activate or influence $v$. We will explain how to compute the probability in the next section.

Second, we need to incorporate the probabilistic differentiation factor into our PSI model in order to determine whether and when to stop propagating information. In the real world, we witness at least three categories of social network participants [32]. Some people are really active in posting reviews about new products or new ideas, and promote others to adopt them and/or disseminate them to more people. Some people are only interested in propagating information to their close friends and receive influence from their close friends. Other people are passive participants in the social network and they may read reviews only and do not propagate the information to their friends. Unfortunately, the heat diffusion based influence model establishes the influence diffusion process by assuming that (i) every node always propagates information to their neighbor nodes (e.g., friends) and that (ii) every node uniformly diffuses its influence to all its neighbor nodes, which is unrealistic. Thus, in our PSI model, we differentiate different types of participants in a social network in terms of their influence diffusion adoption style [16], [17], [32], such as active, friend only, or non-active by introducing two parameters: $c$ as the closeness threshold and $A(u)$ as the influence adopter category. We allow each node in a social network to set its personalized threshold $\theta_c$, which captures the probabilistic characterization of a node’s interest in influencing its neighbor nodes. We use $A(u)$ to determine how an active node $u$ is in propagating information and diffusing influence to its friends. Once $u$ decides to propagate information based on $A(u)$, we use $\theta_c$ to determine which friends of $u$ accept the information or influence. We will describe in detail how to compute and set these two parameters, $\theta_c$ and $A(u)$, in Section 3.2 and Section 3.3 respectively.

Third but not the least, we introduce incentives as a way to stimulate and encourage inactive nodes to become interested and actively engaged in the diffusion process of their neighboring nodes. We will present the probabilistic social influence model with rewards as incentives in Section 3.4.
3.1 Activity-based Probability of Influence

When node $u$ becomes active at time $t$, $u$ has one chance to activate all of $u$’s inactive friends, say $v$, with the probability $w(u,v)$. The result of activation is either “active” or “inactive”. We can view the outcome of this random event as being determined by flipping a coin of bias probability $w(u,v)$: “head (active)” or “tail (inactive)”. In our PSI model, the probability of “activation” is defined by $w(u,v)$ and the probability for “being inactive” is $1-w(u,v)$. The computed probability, $w(u,v)$, is assigned to the edge $E(u,v)$.

It is important to note that the probability $w(u,v)$ should be computed based on dynamic properties of the social network graph. Also for different neighbor nodes, their probability of being activated (influenced) by $u$ should not follow a uniform distribution, since it is well known that some friends are more likely to be influenced while others may be too stubborn to be influenced. Thus $w(u,v)$, the probability for $u$ to activate its neighbor $v$, should not be simply assigned randomly or computed using $\frac{1}{d_u}$, since the degree of node $u$ ($d_u$) fails to capture the amount and the types of activities $u$ has engaged in the social network and with its neighbors.

In our PSI approach we capture both the amount and types of activities engaged and the topological connectivity by defining $w(u,v)$ as the activity-based probability of influence. Concretely, we categorize all activities of a node $u$ into two types: non-interactive activities and interactive activities. We use $NA(u)$ and to denote the number of non-interactive activities performed by $u$ and $IA(u,v)$ to denote the total number of interactive activities conducted between $u$ and its neighbor node $v$ where $v \in V, (u,v) \in E$. Examples of non-interactive activities include posting reviews about the newly purchased camera or posting tips for programing shell scripts. Thus we consider that $NA(u)$ represents how active $u$ is in performing non-interactive tasks that can influence others. Examples of interactive activities include instance chat, co-author papers between two people, commenting on postings of your friends. $NA(u)$ is open to all of $u$’s friends and $IA(u,v)$ is exclusively dedicated to a pair of users, $u$ and $v$. The topological connectivity can be captured by aggregating all $IA(u,v)$ for any $v \in V, (u,v) \in E$. Therefore, the activity-based probability of influence from $u$ to $v$ should be defined by combining these two attributes.

We formulate the probability $w(u,v)$ as follows:

$$w(u,v) = \alpha \frac{NA(u)}{MAX(NA)} + (1 - \alpha) \frac{IA(u,v)}{\sum_{s(u,s) \in E} IA(u,s)}$$

where $\alpha \in [0,1]$ is a damping factor for balancing between non-interactive activities and interactive activities in computing the influence probability $w(u,v)$, and $MAX(NA) = MAX_{v \in V}(NA(v))$.

Note that $w(u,v)$ is high when $NA(u)$ and $IA(u,v)$ are relatively large. When $\alpha$ is set to a small value, approaching zero, large $NA(u)$ will no longer imply high probability $w(u,v)$ if $IA(u,v)$ is relatively small. Similarly, when $\alpha$ is set to a value approaching one, then large $IA(u,v)$ will no longer imply high $w(u,v)$ unless $u$ has significantly high $NA(u)$.

Let $X_{u,v}$ denote the result of activation of $v$ by $u$. Then $X_{u,v}$ can be defined as a binary mode of either “active” or “inactive”. Thus, $X_{u,v}$ can be seen as a discrete Bernoulli random variable, which outputs a “head (active)” with probability $w(u,v)$ or “tail (inactive)” with probability $1-w(u,v)$. We can formally define $X_{u,v}$ as:

$$X_{u,v} = \begin{cases} 
1, & \text{succeed in activating (head)} \\
0, & \text{fail to activate (tail), } u = v, \text{ or } (u,v) \notin E
\end{cases}$$

and

$$P_X(x) = \begin{cases} 
w_{u,v}, & x \text{ head} \\
1-w_{u,v}, & x \text{ tail}
\end{cases}$$

3.2 Closeness Threshold, $\theta_c$

In a real world social network, a node $u$ may influence different friends differently based on the level of closeness that $u$ may have with each of its friends. In order to differentiate friends of $u$ who are very close, who are simply acquaintances in the past, or who are no longer in close contact, we introduce a system defined parameter, called the closeness threshold $\theta_c$. In our PSI model, each node $u$ is given a closeness threshold $\theta_c(u)$. Concretely, $u$ has a chance to activate its friends $v$ with the activation probability $w(u,v)$ if and only if the following condition is satisfied: $w(u,v) > \theta_c(u)$. This condition implies that $u$ can only activate or influence $v$ if the activation probability $w(u,v)$ is above $u$’s closeness threshold $\theta_c(u)$. If $u$ is actively engaged in the diffusion of influence across the network, then $\theta_c(u)$ should be set to a low value. By setting a low $\theta_c(u)$, the probability that the condition of $w(u,v) > \theta_c(u)$ holds is high and thus $u$ has many chances to activate neighbors. On the other hand, if $u$ is only interested in diffusing influence to its close friend, $u$ can set its closeness threshold $\theta_c(u)$ to be higher. Thus, the probability of $w(u,v) > \theta_c(u)$ is lower than previous setting and $u$ has lower chances to activate neighbors.

For example if we set $\theta_c = 0.3$ for all nodes, $u$ has two friends $s$ and $t$, $w(u,s) = 0.25$, and $w(u,t) = 0.5$. Thus, $u$ can only activate or diffuse its influence to $t$. This is because $w(u,s)$ is lower than $\theta_c$ and $s$ is not considered as a close friend of $v$ at the diffusion time. Thus the influence diffusion from $u$ to $s$ is not successful by the current activation probability.

3.3 Adoption Probability Group, $A(u)$ and $P_A(u)$

In order to incorporate different types of nodes in terms of their influence diffusion behavior, we classify all social network nodes according to their capacity with respect to social influence diffusion, and we refer to it as adoption intent for simplicity. According to the statistical study reported in [32], people can be classified into five groups with respect to their willingness to adopt an innovation as shown in Table 1: Innovators, Early Adopters, Early Majority, Late Majority, and Laggards. Innovators are people who are often the first among their friends to adopt an innovation. They are very social and have interaction with other innovators. They propagate innovation to Early Adopters, who are the second group of people who adopt an innovation faster than the rest though they may not be the first one. They are more socially forward than Early...
Majority, the third group. People in Early Majority group tend to be slower in the adoption process and seldom hold positions of opinion leadership in a system. Thus they adopt an innovation after a varying length of time. Also the time of adoption is significantly longer than the Innovators and Early Adopters. Early Majority propagates innovation to Late Majority. People in Late Majority are typically skeptical about an innovation and very little opinion leadership. The last group is Laggards. People in this group are the last to adopt an innovation. Unlike some of the previous categories, individuals in this category show little or no leadership of opinion in influence diffusion. Laggards typically tend to be in contact with only family and close friends.

In our PSI model, we adopt above these five categories of social network nodes to capture the different interests and willingness of social network nodes with respect to information propagation and influence diffusion. For example, innovators are trend leaders. They adopt and also propagate new ideas and innovation to others actively. Thus, we set high percentage (say 90%) of them to activate others and only a small percentage (say 10%) of them may stop propagation. The percentage of stopper increases as we move to the next category. For the laggards, they are neither into adopting new things (accepting influence) nor propagating it (diffusing influence). Thus we can say that high percentage (e.g. 90%) of them are classified as stoppers. Table 1 shows the probability of propagating $P_a(u)$ and the probability to stop diffusion, $1 - P_a(u)$, in each category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Distribution Ratio</th>
<th>$P_a(u)$</th>
<th>$1 - P_a(u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Innovator</td>
<td>2.5%</td>
<td>0.90</td>
<td>0.10</td>
</tr>
<tr>
<td>Early Adopters</td>
<td>13.5%</td>
<td>0.45</td>
<td>0.55</td>
</tr>
<tr>
<td>Early Majority</td>
<td>34%</td>
<td>0.23</td>
<td>0.77</td>
</tr>
<tr>
<td>Late Majority</td>
<td>34%</td>
<td>0.10</td>
<td>0.90</td>
</tr>
<tr>
<td>Laggards</td>
<td>16%</td>
<td>0.05</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Given $P_a(u)$, a node $u$ tosses a coin with the probability to have a head $P_a(u)$. If $u$ gets a head, $u$ keeps propagating, otherwise it stops propagation. We represent this random event as $Y_u$, which is defined formally as follows:

$$Y_u = \begin{cases} 
1, & \text{decide to propagate information (head)} \\
0, & \text{decide not to propagate information (tail)} 
\end{cases}$$

and

$$P_Y(y) = \begin{cases} 
P_a(u), & y \text{ is head} \\
1 - P_a(u), & y \text{ is tail} 
\end{cases}$$

Now we discuss how $P_a(u)$ is used in conjunction with the closeness threshold $\theta_c$ and the probability of $u$ influencing $v$, $w(u, v)$.

Without incorporating the categorization of user nodes, all users are assumed to engage in the influence diffusion. The probability of $u$ activating one of its inactive neighbors, say $v$, is only dependent on $w(u, v)$ and $\theta_c$. By introducing $P_a(u)$, we are differentiating nodes that are more actively engaged in influence diffusion from nodes that are less interested in propagating influence. Thus, we determine whether a node $u$ will be propagating its influence to its inactive neighbor nodes in three steps:

1) With the probability $P_a(u)$, node $u$ decides whether to propagate its influence to its inactive neighbor nodes or not;

2) Once $u$ is in the state of propagating its influence, $w(u, v)$ and $\theta_c$ are used to determine which of its inactive neighbor nodes, $v$, will be selected for activation process;

3) For these selected candidate nodes $v$, $w(u, v)$ is used to determine whether $v$ is activated by $u$.

Given a node $u$, we still need to determine which of the five categories to which $u$ will belong. This will allow us to obtain the respective $P_a(u)$, the group specific probability for propagating influence, as shown in Table 1.

In order to categorize nodes, we propose to introduce the adoption probability group $A(u)$. A naive approach to compute $A(u)$ is to use the degree of its friends. We argue that using the degree of friends to compute $A(u)$ is not sufficient. As studied in [8], [34], some people want to have many friends just for increasing popularity, and others may agree friend requests in order not to be impolite. Thus the degree of friends may not accurately reflect the adopter probability and thus should be used as one of factors, rather than the sole factor, of activeness.

In our PSI model, we propose to combine node degree with the level of activities in both NA and IA categories to compute $A(u)$, the influence adoption probability group, as follows:

$$A(u) = \frac{\beta \cdot (NA(u) + IA(u))}{MAX(NA) + MAX(IA)} + \frac{1 - \beta \cdot d_u}{MAX_{v \in V}(d_v)}$$

where $NA(u)$ is the number of non-interactive activities that $u$ has performed, $IA(u)$ is the number of interactive activities that $u$ did with her friends, $d_u$ is the out-degree of $u$ and $MAX_{v \in V}(d_v)$ is the maximum degree in the graph $G$, and $\beta$ is a balancing weight function, which carefully combines node degree and activities in computing $A(u)$. By varying $\beta$ we can focus more on activities or degree of node $u$. When $\beta$ is 0, $A(u)$ solely depends on the degree of node. On the other hand, by setting $\beta$ to 1, $A(u)$ is determined exclusively by the normalized number of interactive and non-interactive activities.

$A(u)$ represents the influence adoption probability of $u$ and thus can be used to categorize node $u$ into the appropriate influence adoption group as shown in Table 1.

3.4 PSI with Rewards as Incentives

In this section we describe how to incorporate rewards as incentives to the probabilistic social influence (PSI) model in terms of reward effects and reward targets.

3.4.1 Reward Effects

In general, companies give out rewards to people in order to stimulate the product sales through "word of mouth" effect [9]. Rewards can be given in many different forms. One way to model different forms of rewards is to define rewards in terms of benefits that a user receives and the efforts that a user would need to make in order to receive rewards. We propose to formulate reward effects in terms
of two factors: Efforts and Benefit. Some rewards require much more efforts in terms of time or monetary, whereas other rewards demand low efforts. For example, a credit card company \( C_1 \) offers 50,000 points after you spend $3,000 in first 3 months. Another company \( C_2 \) offers $10 credit back when you spend $40 at a restaurant. \( C_1 \)'s promotion requires users to spend at least $3,000 while \( C_2 \)'s promotion requires only $40. Thus we use \( E \) to represent a scale of efforts, which ranges between 1 to \( m_E \), and \( m_E \) is a positive integer defining the upper bound. Similarly, we use \( B \) to denote a scale of benefit, which is ranging between 1 to \( m_B \). \( C_1 \)'s promotion offers 50,000 points in their monetary system, which is worth of $500, thus the reward effect is \( 500/3000=1/6 \), while \( C_2 \)'s promotion gives back $10 credit, thus the reward effect is \( 10/40=1/4 \). Though the two promotions have different raw benefit and effort scales, we usually use the reward effect to represent the ratio of benefit over effort.

In PSI, we can formulate reward effects as the following formula:

\[
R = \frac{B}{E}
\]

where \( c \) is a normalized function and can be used to make \( R \) within the range between 0 and 1. For example, By setting \( c \) to \( \frac{1}{m_B} \), we can ensure \( R \) to be in the range [0, 1]. Figure 2 shows the trend of reward effects by varying \( B \) and \( E \) at the same time while we set \( m_E = 10 \) and \( m_B = 10 \). When we fix the effort scale and increase benefit scale from 1 to 10, then the reward effect also increases up to 10%. But when we increase effort scale, the reward effect decreases to as low as 0.1%

![Fig. 2. Reward Effect](image)

3.4.2 Reward Target

For a social network of \( n \) nodes, the next question that we need to address is how to select nodes to receive reward under a limited budget constraint. This is a common question when the amount of rewards to be given out for the purpose of marketing campaign or influence diffusion stimulation is limited due to the budget constraint. For example, most of companies have limited marketing budget to promote their products. The ultimate goal here is to maximize the influence of the viral marketing with a limited budget of rewards so that the maximization of the utility of rewards can lead to the largest possible number of people to buy the products. It is widely recognized that random selection of marketing targets is ineffective.

In PSI, we promote to distribute rewards by selecting only one of five groups of social network nodes at a time: Innovator, Early Adopter, Early Majority, Late Majority, and Laggards. Clearly, members in the same group have similar activation probabilities. Innovators have highest activation probability and laggards have the lowest. We will compare and determine which group is the most effective group in terms of promoting new products. Also even though people in the chosen group will be exposed to the reward but not all of people will get the reward because of the limited marketing budget. Thus we again introduce a probabilistic control such that each person in the group has one chance to take the reward or not. If the user takes the reward, then her probability to propagate the social influence will be increased from \( P_a(u) \) to \( P_a^R(u) \) by incorporating the reward effect into the Eq. (10).

The problem of choosing a subset of \( k \) nodes in a social network graph, which can provide the maximum influence coverage, is NP hard [21]. Here \( k \) is given based on the resource budget constraint, assuming that the total cost of sensing the influence at each of the \( k \) nodes should not exceed the given resource budget. Thus a greedy algorithm is often employed [26].

4 Influence Rank By Coverage

4.1 Influence Coverage (IC) and Influence Rank (IR)

One criterion to evaluate the effectiveness of an influence diffusion model over a social network is to measure the influence coverage for each of its influence sources and to aggregate influence coverage of all of its influence sources.

Let \( G = (V,E) \) denote a social network graph and \( u \in V \) is a node that is chosen as the sole influence source at time \( t = 0 \). In other words \( u \) is the only active node at \( t = 0 \). The Influence Coverage (IC) of node \( u \) over \( G \), denoted by \( IC(u) \), can be defined by the total number of nodes that \( u \) can influence through hop by hop iterative computation of \( u \)'s influence over the entire network graph \( G \). The iterative computation terminates when the graph traversal started from \( u \) has reached every node of \( G \) and thus the convergence condition is met. For PSI model, the influence coverage is computed in terms of influence probability, \( \theta_c \), and activation probability group \( A(u) \).

After we compute \( IC \) for every node in a social network graph \( G \), we can sort all nodes in \( G \) by descending order of their influence coverage score \( IC \) and produce a rank value
for each node \( v \). We refer to this rank value the Influence Rank (IR) of \( u \).

Figure 3(a) shows an example social network graph with 10 nodes and 26 edges. Each node has the number of postings in their profile page as shown in the number underlined. Each edge has a weight value defined by the number of interactions between two nodes. For example, \( v_9 \) has 23 postings and 6 interactions with \( v_{10} \) and 20 interactions with \( v_1 \). Recall that we use \( \alpha \in [0, 1] \) as a weight function to balance between non-interactive activities and interactive activities in computing the influence probability \( w(u, v) \) as the edge weight for each edge \((u, v) \in E\). If we set \( \alpha \) to be 0.5. In this example graph, \( MAX(NA) \) is 70 from \( v_1 \). Thus \( w(v_9, v_{10}) \) is computed as follows based on Eq. 3:

\[
 w(v_9, v_{10}) = 0.5 \cdot \frac{NA(v_9)}{MAX(NA)} + (1 - 0.5) \cdot \frac{IA(v_9, v_{10})}{\sum_{s \in E} IA(v_9, s)} = 0.5 \cdot \frac{23}{70} + (1 - 0.5) \cdot \frac{3}{3 + 10} = 0.28
\]

Figure 3(b) shows the graph with influence probability as edge weight upon the completion of the influence probability computation for each edge in Figure 3(a).

Now we use this example to illustrate how to compute \( IC \) and \( IR \). By using the five categories of activation probability groups and their distribution ratios in Table 1, we categorize the nodes in this example graph by computing their adopter probability category \( A(u) \) using Eq. (8) as shown in Table 2. Given that \( A(v_1) \) is included in top 2.5\% node \( v_1 \) is chosen as an innovator. Similarly, \( v_8 \) and \( v_{10} \) are considered as laggards since their adopter probability group \( A(v_8) \) and \( A(v_{10}) \) are included in the bottom 16\%. In this example, we set \( \theta_c \) to 0.5. To compute \( IC \) for \( v_1 \), we set only \( v_1 \) as the influence diffusion source at \( t = 0 \). In other words, \( v_1 \) is active and all other nodes are inactive, as shown in Figure 4(a).

At \( t = 1 \), \( v_1 \) decides to activate her inactive neighbors with \( P_a(v_1) \) which is 0.9 according to Table 1 for innovator. This can be viewed as \( v_1 \) tossing a coin with a probability \( P_a(v_1) = 0.9 \) by Eq. 7. Let \( Y_{v_1} \) denote the result of tossing a coin. If \( Y_{v_1} \) returns 1, it decides to propagate. Then it tosses coins again with a probability \( w(v_1, v_j) \) in order to activate inactive friends as long as they pass the following test: \( w(v_1, v_j) > \theta_c \) for \( v_j : (v_1, v_j) \in E \) and \( v_j \) is inactive. From Figure 4(a), we see that all of \( v_1 \)'s outgoing edges have the probability greater than \( \theta_c = 0.5 \), and all of \( v_1 \)'s neighbors are inactive. Thus, \( v_1 \) tries to activate all of its 6 neighbors, \( v_2, v_3, v_4, v_5, v_8, \) and \( v_9 \). By the probabilistic influence diffusion, which combines a discrete Bernoulli random variable like tossing a coin (Eq. 6) with \( w(v_1, v_q), q = 2, 3, 4, 6, 8, 9 \), we may have four out of the six nodes, \( v_2, v_4, v_8, \) and \( v_9 \), activated as shown in Figure 4(b).

At \( t = 2 \), these newly activated nodes will follow their respective activation probability \( P_a(v_i) \) for \( i = 2, 4, 8, 9 \) to continue propagating or to stop propagation. Let us assume that by tossing a coin \( Y_2, Y_8, Y_9 \) are 1 and \( Y_4 \) is 0. Thus, \( v_2, v_8, \) and \( v_9 \) have a chance to activate their inactive neighbors. Given that \( v_2 \) has no inactive friends to activate, \( v_2 \) terminates the diffusion process. Thus, only \( v_8 \) and \( v_9 \), marked by black solid circles, are considered at time \( t=2 \), each has only one inactive friend. \( v_8 \) tries to activate \( v_{10} \) and \( v_9 \) tries to activate \( v_7 \). However, given that \( w(v_8, v_{10}) = 0.45 \leq \theta_c = 0.5 \) and \( w(v_9, v_7) = 0.28 \leq \theta_c = 0.5 \), thus, both nodes terminate the diffusion process. As a result, there are no newly activated nodes at \( t = 2 \).

At \( t = 3 \), we could not find any other active nodes that have not been examined and no more new nodes can be activated. Thus the diffusion process converges and we stop the influence diffusion process with \( v_1 \) as the sole influence source at \( t = 3 \), as shown in Figure 4(c). The \( IC \), for \( v_1 \) is 4 since there are four nodes \( (v_2, v_3, v_8, v_9) \) included in the coverage of \( v_1 \)’s influence.

The computation of \( IC \) and \( IR \) for the remaining nodes follows a similar procedure. Clearly the \( IC \) computation is the dominating factor in terms of the overall computation cost of \( IR \). When the size of a social network is big, this simple and straightforward algorithm for influence

---

**Table 2.** \( A(u) \) and Category of \( v_i \)

<table>
<thead>
<tr>
<th>( v_i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A(u) )</td>
<td>1.00</td>
<td>0.47</td>
<td>0.10</td>
<td>0.60</td>
<td>0.18</td>
<td>0.47</td>
<td>0.21</td>
<td>0.34</td>
<td>0.24</td>
<td>0.10</td>
</tr>
<tr>
<td>( P(u) )</td>
<td>0.90</td>
<td>0.23</td>
<td>0.05</td>
<td>0.45</td>
<td>0.10</td>
<td>0.23</td>
<td>0.10</td>
<td>0.23</td>
<td>0.10</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**Fig. 3.** Computed weight \( w(u, v) \)

(a) Graph with Activities  (b) Graph with Probability
coverage computation is not efficient. In the next section we will describe one efficient implementation of our IC algorithm.

### 4.2 Computing IC by Matrix Multiplication

An alternative and more efficient approach to computing the IC, for all nodes in a social network graph \( G = (V, E) \) is to use matrix multiplication.

Considering the running example given in Figure 3(a), we can formulate the computation of IC, for node \( v_i \) as a matrix multiplication problem.

Let \( S_i(t) \) denote the binary state of node \( v_i \) at time \( t \) and the value of \( S_i(t) \) is either 1 (active) or 0 (inactive). For example, at \( t = 0 \), only \( v_1 \) is active. Thus we have \( S_0(1) = 1 \) and \( S_0(i) = 0 \) \( (1 < i < n) \) and \( n \) denotes the size of \( V \).

\( v_i \) makes a decision to propagate influence with the probability \( P_a(v_i) \). Let \( Y(i) \) denote the binary state of node \( v_i \) by tossing a coin. If we get \( Y_1 = 1 \), it implies \( v_i \) is propagating the influence to its inactive neighbor nodes based on the threshold \( \theta \), and the probability weight \( w(v_i, v_j) \) satisfies that \( (v_i, v_j) \in E \) and \( v_j \) is an inactive neighbor of \( v_i \). This computation can be written as the multiplication of \( S_0(1) \) and \( Y_i \), namely \( Y_i \times S_0(1) \).

Assume that \( v_2 \) is activated by \( v_1 \) at \( t = 1 \). Then the state of \( v_2, S_1(2) \), is 1. Thus, the activation results of \( v_2 \) at \( t = 1 \) can be represented by \( S_1(2) = X_{1,2} \times Y_1 \times S_0(1) \) using Eq. 6.

In short, the activation process can be formulated as a three-step iterative process:

1. **Step 1**: Define the state of the source of information diffusion by selecting a node \( v_1 \) at time \( t = 0 \), denoted as \( S_0(0) \);
2. **Step 2**: Determine whether to keep propagating or not by multiplying \( Y_i \) with \( X_{i,j} \), the result of a discrete Bernoulli random variable test, namely \( Y_i \times S_0(0) \);
3. **Step 3**: Compute the state of \( v_j \) at \( t + 1 \), \( S_{t+1}(j) \), as \( X_{i,j} \times Y_i \times S_0(i) \).

Figure 4(d) shows that both \( v_1 \) and \( v_4 \) are direct neighbors of \( v_2 \) and both nodes are active at time \( t \), then both nodes will try to activate \( v_2 \) using PSI. The result of this activation can be written as \( X_{1,2} + X_{4,2} \). If one of the nodes succeeds with \( X_{1,2} + X_{4,2} \geq 1 \), then \( v_2 \) is activated; Otherwise \( v_2 \) remains as inactive. \( v_2 \)’s state at \( t + 1 \) is expressed as follows:

\[
S_{t+1}(2) = X_{1,2} \times Y_1 \times S_t(1) + X_{4,2} \times Y_4 \times S_t(4)
\]  

We can generalize the above equation to compute the state of a node \( v_i \) at time \( t+1 \) as follows:

\[
S_{t+1}(i) = \sum_{(v_j,v_i) \in E} X_{j,i} \times Y_j \times S_j(1)
\]  

Note that for any node \( v_j \) that has no direct edge connecting to \( v_i \) at time \( t \), we have \( S_t(j) = 0 \).

In order to compute IC, for all the vertices at the same time, the matrix representation is utilized. First, we use a state column vector \( S_t \) to represent the state of vertices at time \( t \). For example, if \( v_1 \) and \( v_4 \) are active and the rest of the nodes are inactive at \( t = 0 \), then \( S_0 \) is given below:

\[
S_0 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T
\]  

Let \( X_{u,v} \) denote the result of coin toss (activation). Node \( u \) has only one chance to activate its neighbor node \( v \). Thus, \( X_{u,v} \) can be pre-computed and stored as a \( n \times n \) matrix:

\[
X = \begin{bmatrix}
X_{1,1} & X_{2,1} & X_{3,1} & \cdots & X_{n,1} \\
X_{1,2} & X_{2,2} & X_{3,2} & \cdots & X_{n,2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
X_{1,n} & X_{2,n} & X_{3,n} & \cdots & X_{n,n}
\end{bmatrix}
\]

Similar to \( X_{u,v} \), \( Y_u \) is also the result of coin toss to determine if node \( u \) is willing to activate its inactive neighbors. Thus \( Y_u \) also can be pre-computed and stored as a \( 1 \times n \) matrix as follows:

\[
Y = \begin{bmatrix} Y_1 & Y_2 & Y_3 & \cdots & Y_n \end{bmatrix}
\]

By Eq. 13, 14, and 15, we can compute the state of all vertices at time \( t + 1 \) after activation at \( t \) as follows:

\[
S_{t+1} = XY \times S_t
\]

Algorithm 1 provides a sketch of the IC computation for a given node \( v_i \). We set the given node \( v_i \) as an influence source and active. Then we perform a matrix multiplication until there are no more newly activated nodes. For each iteration we count the number of activated nodes for \( v_i \)’s IC.

**Algorithm 1: IC(vi)**

```
1. \( S_0 \leftarrow \begin{bmatrix} n \times n \end{bmatrix} \text{zero matrix; } t = 0; \) /\* \( v_i \) becomes first active \*/
2. \( S_0(i) \leftarrow 1; \) /\* initialize IC \*/
3. \( IC_t \leftarrow 0; \) /\* initialize \*
4. \( \text{while New active nodes exist do} \)
5. \( \text{\( S_{t+1} = XY \times S_t; \)} \)
6. \( \text{\( \text{if } \forall \text{\( S_{t+1}(j) \geq 1 \text{ then} \)} \)} \)
7. \( \text{\( S_{t+1}(j) \leftarrow 1; \)} \)
8. \( \text{\( \text{end;} \)} \)
9. \( IC_t \leftarrow IC_t \cup \{\text{Newly activated nodes}\}; t \leftarrow t + 1; \)
10. \( \text{end;} \)
```

### 4.3 IR Computation Algorithms for Rewards

We have described a simple way to compute the influence coverage IC for each node in a social network graph when setting this node as the sole influence source. We call this approach the independent IC since the influence coverage is computed for each individual node independently. However, the ultimate goal to incorporate rewards into our PSI model is to find the subset of \( k \) nodes as the recipients of the rewards, which can provide the maximum aggregate influence coverage, given \( k \) as the total resource budget for the rewards.

There are many candidate subsets of \( k \) nodes, but we want only the one with the maximum aggregate influence coverage. If we simply sort all nodes in a descending order of their individual IC scores, and select the top \( k \) nodes with the highest individual IC scores, we may not find the subset of \( k \) nodes that provide the maximum aggregate influence coverage when many of the \( k \) nodes have large overlapping in their IC, i.e., a large number of common nodes included in their influence coverage.

Given the inherent problems with the independent IC based method, in this section we consider another two mechanisms to compute the aggregate influence coverage by the top \( k \) nodes: locally minimal overlap IC, and globally minimal overlap IC. Both are considered common
mechanisms to compute the aggregate influence coverage by a given subset of \( k \) nodes, which improves the independent IC in terms of achieving maximum aggregate influence coverage.

In PSI with rewards, regardless which of the three mechanisms we use to compute the maximum aggregate influence coverage, we use the same algorithmic structure in our design and implementation of the respective IC algorithms.

- Step 1: Compute activation matrix \( X \). Result of activation is the same as coin toss. Once a probability is given, the result can be pre-computed and stored as a matrix \( X \).
- Step 2: Compute IC\( _i \). For each node \( v \), we compute corresponding IC\( _i \). IC\( _i \) can be computed individually or together with other active nodes.
- Step 3: Sort IC\( _i \) and select top-\( k \) nodes order by IC\( _i \).

### Independent IC

Given a social network composed of \( n \) nodes, Algorithm 2 shows the steps of how to compute IR\( _i \) using independent IC. The basic steps of this algorithm is first to calculate IC\( _i \) of each node \( v \) and then sort nodes in descending order of their IC. The advantage of this algorithm is the simplicity in terms of conceptualization and implementation. However, the big weakness of this algorithm is the problem of potentially large overlapping of the top-\( k \) influential sets, one by each of the top \( k \) nodes.

Table 3 shows IC\( _i \) for 5 nodes: \( v_1, v_2, v_3, v_4, v_5 \). Based on Algorithm 2, top-2 influential nodes are \( v_1, v_2 \). But \( v_2 \)’s coverage is completely overlapping with \( v_1 \)’s, leading to poor results in terms of maximum aggregate information coverage. This motivates us to consider alternative ways to compute the aggregate influence coverage, which can minimize the IC, overlap.

### Algorithm 2: Independent IC

1. Compute \( Y \).
2. Compute \( X \).
3. foreach \( v \in V \) do
   4. Compute IC\( _i \).
5. end
6. Compute IR by sorting \( \{IC\_i | v \in V \} \) by set size;

### Locally Minimal Overlap IC

This is a localized greedy algorithm. For each node \( v \), the algorithm first computes IC\( _i \). Then instead of simply choosing the top \( k \) nodes with highest individual IC as the top \( k \) influence ranked nodes, it selects the top \( k \) most influential nodes iteratively as follows: At first, node \( v \) whose IC\( _v \) is the largest is selected as the first seed of top-\( k \) influential nodes. For each remaining node \( u \), we compute the locally minimal overlap IC, LC\( _u \), which is the difference between two coverages, the chosen universal coverage, UIC, from a set of previously added highest IR nodes and the remaining individual coverage.

### Algorithm 3: Locally Minimal Overlap IC

1. Line 1 to 5 in Algorithm 2
2. \( UIIC \leftarrow \emptyset; \ V_{local} \leftarrow \emptyset; \ G' \leftarrow G\ V_{local} \leftarrow \emptyset; \)
3. while \( G' \neq \emptyset \) do
   4. foreach \( v \in (G \ V_{local}) \) do
      5. \( LC_i \leftarrow IC_i \setminus \ UIIC; \)
   6. end
7. Select \( v \) that has max(LC\( _i \));
8. \( V_{local} \leftarrow V_{local} \cup \{v\}; \ G' \leftarrow G \setminus v_{global}; \)
9. \( IC_i \leftarrow IC_i \cup IC_i; \)
10. end
11. Compute IR by sorting \( v \in V_{local}; \)

### Globally Minimal Overlap IC

Comparing to independent IC, Algorithm 3 generates a larger aggregate influence coverage for a given \( k \) nodes and thus a better quality of influence rank IR. However, due to the nature of a local greedy algorithm, it does not consider cases where multiple sources of diffusion exist. Thus locally minimal overlap may still produce large overlapping among the top \( k \) influential sets. For example, on a launch of a new iPhone, multiple bloggers post their own reviews. The information that one person receives may come from several diffusion sources. People may read multiple reviews before deciding to purchase a new product (the new iPhone). Thus, we need to devise a global greedy algorithm to model the multiple sources of information diffusion, which will enable us to find the globally minimal overlap IC.

### Algorithm 4: Globally Minimal Overlap IC

1. Line 1 to 5 in Algorithm 2
2. \( UIIC \leftarrow \emptyset; \ V_{global} \leftarrow \emptyset; \ G' \leftarrow G\ V_{global} \leftarrow \emptyset; \)
3. while \( G' \neq \emptyset \) do
   4. foreach \( v \in (G \ V_{local}) \) do
      5. \( S_0 \leftarrow n \times n \) zero matrix;
   6. end
   7. \( S_{0}(i) \leftarrow 1; /\ast \) heat sources \( \ast/\)
   8. end
9. foreach \( v \in (V \ V_{global}) \) do
   10. \( S_{1}(i) \leftarrow 1; \ IC \leftarrow IC \cup \{v\}; \)
   11. \( GC_i \leftarrow IC \setminus \ 0 \ IC; \)
   12. end
   13. Find \( v \) that has max(GC\( _i \));
   14. \( V_{global} \leftarrow V_{global} \cup \{v\}; \ G' \leftarrow G \setminus v_{global}; \)
   15. end
16. Computing IR by sorting \( v \in V_{global}; \)

The first portion of Algorithm 4 is similar to that of Algorithm 3. It computes individual IC\( _i \), then selects a node \( v \) whose IC\( _i \) is the largest (Line 2). Then we set \( v \)
as an initial seed of source of information diffusion (Line 3). We denote UIC as the universal IC, for the set of influential nodes $V_{\text{global}}$. Now we use Hill-climbing algorithm to simulate multiple source of influence diffusion. First, for each node $v_i$ not in $V_{\text{global}}$, we add temporarily $v_i$ to $V_{\text{global}}$, then compute $IC_i$, which simulates multiple sources of information diffusion (Lines 9-12). Given two coverages, $IC_i$ and $UIC_i$, we define globally minimal overlap $GC_i$ as a difference between $IC_i$ and $IC$. After computing $GC_i$ for all remaining nodes, we select $v_i$ that has the largest $GC_i$ as the next highest IR, node.

In the next section we will evaluate our PSI with rewards, including comparing these three different mechanisms to compute IC and thus the influence rank used to select the top $k$ most influential nodes to receive rewards as incentives.

5 Experiments

In this section we report our experimental evaluation of the performance and effectiveness of our PSI model. Our experiments are conducted with two objectives. First, we want to show the effects of parameters that we present in our PSI models such as $\alpha$, a weight function for balancing between $NA$ and $IA$, $\theta_u$, closeness threshold, and $\beta$, a balancing weight function between the sum of $NA$ and $IA$ and degree. Each parameter affects the number of activated nodes. These parameters should be carefully chosen for different types of SNS. Our experiments will be a good guidance in selecting those parameters. Second, we want to evaluate the performance of our PSI models with or without rewards against topology-based and naive activity-based approach. We show that probability and incentive approach has up to 7 times bigger activation coverage than previous approaches.

5.1 Datasets

Three datasets are used in this experiments such as DBLP [3], Epinions [1], and and Facebook [2]. DBLP datasets consist of paper authors as nodes and their co-authorship as edges. For example, two authors $u$ and $v$ wrote two papers, and $u$ and $w$ wrote three papers, then $u$ and $v$ are connected by $E(u, v)$ and $u$ and $w$ by $E(u, w)$. Because $u$ wrote two papers with $v$ and three papers with $w$, we set $IA(u, v)$ as 2 and $IA(u, w)$ as 3. Note that $u$ wrote total 5 papers with $u$ and $v$. Writing five papers can be considered not only as interactive activities but also non-interactive activities. Therefore, we set $NA(u)$ as 5. DBLP dataset has 4,768 nodes and 32,020 edges.

Facebook is a social network service that provides a profile page for each user. Users can update their profile page, post photos, and leave comments on their posting or friends’ postings. When a user $u$ posts something on her page, we consider it as a non-interactive activities. If $u$ leaves comments on her friend’s posting, it is considered as an interactive activity. By counting $NA$ and $IA$, we compute $NA(u)$ and $IA(u, v)$. We launched a Facebook app and in total 273 users used the app. Once a user $u$ allows us to use the private information, we extract their friends relationship information. For example, one user may have 400 friends. Then from one user we can create 401 user nodes. Some users have more than 1,000 friends. By doing this, we can create 76,954 user nodes and 1,121,861 friendship relationships from 273 users.

Massa [28] collected data from Epinions, a website for consumer reviews and trust networks. Epinions provide a system that users who bought products can leave reviews on them. Then potential buyers read reviews and determine if they buy the product or not. The potential buyers do not solely rely on the reviews but reviews have influence on users. Therefore Epinions is a good dataset to gauge influence. On Epinions, users post reviews. We consider this reviews as $NA$. For $IA$, we use a trust list. For example, if users $u$ and $v$ posted some reviews. When a user $w$ likes $u$’s reviews and does not $v$’s reviews, then $w$ creates a trust list by adding $u$ and does a block list by adding $v$. Next time when $w$ visits Epinions, reviews from $w$’s trust list will be shown and one from $w$’s block list will be filtered out. We create $E(u, v)$ from the trust link. Epinions dataset 49,288 nodes and 487,002 edges.

5.2 Effects of $\beta$

The first set of experiments focus on $\beta$, which is used to compute $A(u)$ and plays a role of balancing a weight between the number of activities and the degree of $u$. Figure 5 shows the adopter probability category $A(u)$ by varying $\beta$. $x$-axis is the value of $A(u)$ and $y$-axis is the cumulative density function. We vary the balancing weight $\beta$ from 0 to 1.

DBLP data and Facebook data show that when we increase $\beta$, $A(u)$ decreases. Greater value of $\beta$ means we consider activities are more important factor in computing $A(u)$ than degree of a node. Activity information consists of two values; $NA$ and $IA$. The sum of two values are normalized divided by the sum of two values $MAX(IA) + MAX(NA)$. In order to get high $A(u)$, both $NA$ and $IA$ should be also large. But not all nodes have two large values. Therefore when we increase $\beta$, $A(u)$ is decreasing. On the other hand, Epinions dataset has different property. If we increase $\beta$, $A(u)$ also increases. Epinions dataset has $NA$ which is the number of reviews and does not have $IA$. Thus, when we increase and set $\beta$ to be 1, the value of $A(u)$ is computed by $NA(u)$ only. However, if we set low $\beta$ such as 0, then $A(u)$ completely depends on $d_u$. Compared to $NA(u)$, $d_u$ is lower and the standard deviation is high such as mean 9 and std 32, respectively. Big difference in $d_u$ means lower value of $\frac{d_u}{MAX_{u \in V}(d)}$. From now on, we set $\beta$ as 0.5 to weight on evenly both the number of activities and the degree of $u$. 

Fig. 5. Adopter Probability Category $A(u)$
5.3 Effects of $\alpha$

The parameter $\alpha$ is used in computing $w(u,v)$, the probability for $u$ to activate $v$. $\alpha$ is a balancing weight between $NA$ and $IA$ in computing $w(u,v)$. However, Epinions dataset has only $NA$ values, thus we designed an experiment for DBLP and Facebook datasets only. In this experiments, we set $\theta_c$ as 0.05 uniformly for all of nodes and $\beta$ as 0.5 to evenly weight on activities and the degree of node. Due to the value of $\theta_c$, nodes with $w(u,v)$ less than 0.05 are excluded for activating process.

Figure 6 shows the cumulative density function of $w(u,v)$ for each dataset. When we increase $\alpha$, a weight parameter balancing between $NA$ and $IA$, $w(u,v)$ decreases. In computing $w(u,v)$ we normalize $NA(u)$ by dividing from $MAX(NA)$ and $IA(u,v)$ by dividing from $\sum IA(u)$. In other words, $NA(u)$ is normalized over the entire $NA$ values while $IA(u,v)$ is normalized among $u$’s $IA$ values. Then the difference between $NA(u)$ and $MAX(NA)$ might be larger than the difference between $IA(u,v)$ and $\sum IA(u)$. The bigger difference, the smaller fraction value, which means $\frac{NA(u)}{MAX(NA)}$ might be smaller than $\frac{IS(u,v)}{\sum IA(u)}$. Thus, when we increase $\alpha$, $w(u,v)$ is also decreasing as shown in Figure 6(a). Note that Epinions dataset has no $IA$ information and $w(u,v)$ completely weight on $NA$. Thus, when we set $\alpha$ to be 0, then $w(u,v)$ is also 0. As we increase $\alpha$, $w(u,v)$ also increases because $w(u,v)$ weights more on $\frac{NA(u)}{MAX(NA)}$ as shown in Figure 6(c).

Note that Figure 6(b) shows that more than 90% of nodes have $w(u,v) < 0.01$. For collecting Facebook dataset, we request 273 users to take a part in our experiments. Thus, for each node $u$ in these 273 nodes, we can get $NA(u)$, $IA(u)$, and $d_u$ and we extract $u$’s friend network which results in 76,954 friends and 1,121,861 friendship. For each node $v$ in 76,954 extracted nodes, we have no $NA(v)$ and very limited information of $IA(v)$ and $d_v$. For example $u$ is one of 273 Facebook app users, $v$ is not a Facebook app user. If $v$ leaves a comment on $u$’s photo, we create two nodes $u$ and $v$, edges $E(u,v)$ and $E(v,u)$, and set $IA(v,u)$ as 1 and $NA(v)$ as 0. Due to the way of constructing the social network graph, some nodes have $NA$, while other do not have $NA$. Also some nodes have high $IA$, while others have lower $IA$. This distribution results in lower $w(u,v)$ for node $u$ that is not one of 273 Facebook App users. Therefore, more than 90% of $w(u,v)$ are lower than 0.01.

Fig. 6. Probability $w(u,v)$ for different $\alpha$

Figure 7 shows the effect of $\alpha$. $x$-axis shows the number of top-$k$ users and $y$-axis shows the number of activated nodes. In both datasets, when we set $\alpha$ to be higher, the number of influenced people tends to be also higher. In order to have a fair comparison we will set $\alpha$ as 0.6 in the next experiments.

Fig. 7. Effects of $\alpha$

5.4 Effects of $\theta_c$

The parameter $\theta_c$ is a value used for filtering out acquaintances. If $w(u,v)$ is less than $\theta_c$, we consider $u$ and $v$ is not close enough for activation process. Figure 8 shows the effect of $\theta_c$ for all three datasets. We vary $\theta_c$ from 0.01 to 0.05. If we increase $\theta_c$ then more people are excluded for activating process because there will be more edges with $w(u,v) < \theta_c$. Once the number of nodes to be target of activation is decreased, the number of activated nodes also decreases. In the following experiments we set $\theta_c$ as 0.05.

Fig. 8. Effects of $\theta_c$

Note that in Facebook dataset the lines for $\theta_c = 0.03$ and $\theta_c = 0.05$ are the same. We explained why more than 90% of nodes in Facebook data have lower $w(u,v)$ in section 5.3. 95% of nodes in Facebook dataset has $w(u,v)$ less than 0.01. Thus when we set $\theta_c$ to be larger than 0.01, most of nodes do not take a part in information diffusion and the number of activated nodes are extremely low. Note that when we increase $\theta_c$ larger than 0.01, the number of nodes with $w(u,v)$ larger than $\theta_c$ is the same. Therefore, the number of activated nodes for $\theta_c = 0.03$ and $\theta_c = 0.05$ are the same. From now on we set $\theta_c$ as 0.05 except for Facebook dataset ($\theta_c$ as 0.01).

5.5 Effects of Reward Effect, $R$

Next set of experiments show how the reward effect $R$ affects the number of activated nodes. In this experiment we set $\alpha$ as 0.6, $\beta$ as 0.5, and $\theta_c$ as 0.05. For each dataset we vary $R$ from 0.01 to 0.2. If we set $R$ to be 0.01 then it boosts the $P_\alpha(u)$ 1%. If we set $R$ to be 0.2, then $P_\alpha(u)$
is increased 20%. We also varied the marketing target. For each experiment we select only one group as a marketing candidate. Individuals in the candidate group have chance to get the reward. If the individual accepts the reward, then $P_a(u)$ is boosted by $R$ we set. Therefore the impact of rewards is to decrease the probability of $u$ to become a stopper.

$x$-axis shows the number of top-$k$ users and $y$-axis shows the number of activated nodes. For each dataset, we performed the five experiments. For each experiment, one group is selected as a marketing candidate. We give individuals in the selected group a chance to take reward. For example, Innovator line in the chart shows the total number of influenced nodes when we assign incentives to only individuals in innovator group.

In DBLP datasets as shown in Figure 9, when we target innovators, the number of activated nodes are highest. Early adopters may be the alternative target for marketing but the effect of rewards for the early adopters is only slightly bigger than other groups. Remaining three groups, early majority, late majority, and laggards, are not appropriate targets for reward. Although $R$ increases the probability to be active, $P_a(u)$, and $u$ becomes active, $u$ may still have a very low $w(u,v)$. Innovators and Early Adopters usually have higher $w(u,v)$ because they have large $NA(u)$ and $IA(u)$. However, Early Majority, Late Majority and Laggards may have lower $w(u,v)$ due to their low activities. Therefore, regards less of $R$’s boosting in $P_a(u)$, low values in $w(u,v)$ result in the low number of activated nodes.

In Epinions dataset as shown in Figure 10, we do not have $IA$ information. Therefore $w(u,v)$ values are also low. Even Innovators may have lower $w(u,v)$. Due to the lower $w(u,v)$, $R$ is not effective for all types of nodes when $R$ is lower than 0.1. Like DBLP dataset, individuals in the innovator group is the most effective marketing target for rewards.

For Facebook datasets as shown in Figure 11, the number of activated nodes are the same for all five marketing target groups. More than 90% of $w(u,v)$ values in Facebook datasets are lower than $\theta_c$. This is due to the way of constructing social graph. $w(u,v)$ is computed using both $NA(u)$ and $IA(u)$ but only 273 users have $NA$ and some users have $IA$. Thus, $w(u,v)$ is extremely low. Although rewards boosts the probability to participation, extremely low $w(u,v)$ diminishes rewards effects. These figures may mislead that rewards are not effective at all. For Facebook dataset, we modified reward effect so that $R$ can boost both $P_a(u)$ and $w(u,v)$. Boosting $w(u,v)$ is computed as follows:

$$w(u,v) = w(u,v) + (1 - w(u,v))R$$

A node $u$ who agrees to get a reward will have a boosted $P_a(u)$, which makes $u$ more actively participate in propagating information diffusion, and increased $w(u,v)$, which allows $u$ to have higher chance to succeed in activating a neighbor node $v$. We applied this modified Eq. 17 and did the same experiments over Facebook dataset. Figure 12 shows the result of the experiment. Similar to other datasets, innovators respond more actively over rewards.

---

**5.6 Comparisons**

Lastly, we conducted an experiment to show the performance of heat diffusion model (Topology), PSI model without rewards (Activity), and PSI with rewards (Reward). We set $\alpha$ as 0.6, $\beta$ as 0.5, and $\theta_c$ as 0.05. For Facebook datasets, we use modified Eq. 17 so that we make $R$ effective. $x$-axis is the number of top-$k$ influential users and $y$-axis is the number of activated nodes. In all three datasets, heat diffusion model (Topology) has the lowest number of influenced people. As explained in Section 1, heat diffusion model sets the uniform probability to activate friends. When a degree is high, then all of friends may not be activated because $\frac{1}{\theta_{ui}}$ can be very low. But when we
consider activity information, we differentiate $w(u,v)$ so that some of close friends are activated and this activation continues to friends of friends.

On top of probability and activity-based approach, we select Innovators as marketing target and give them a chance to accept rewards. Figure 13 shows that our PSI model with and without rewards have the larger number of influenced nodes, especially for DBLP dataset, the number of influenced nodes by PSI with rewards is 7 times more than the number of influenced nodes by topology-based approach.

Fig. 13. Comparisons of Three Approaches

6 RELATED WORK

Social influence analysis as marketing techniques has received increased attention over the last decade [5], [10], [14], [19], [21], [24], [31], [38]. It holds the potential to increase brands or products awareness through word-of-mouth promotion. [21], [31] pioneered the concept of social influence by modeling the selection of influential sets of individuals in a social graph as a discrete optimization problem. It utilizes the provable greedy approximation algorithm for maximizing the spread of influence in a social network. [31] proposed a cascading viral marketing algorithm, which tries to find such a subset of individuals that if these individuals adopt a new product or innovation, then they will trigger a large cascade of further adoptions.

[26] proposed a heat-diffusion based viral marketing model with top $k$ most influential nodes which utilizes the heat diffusion theory from Physics to describe the diffusion of innovations and help marketing companies divide their marketing strategies into several phases.

Another relevant area is social influence rankings [12], [15], [23], [36], which measures and rank nodes in a social network by their social influence ranks, similar to PageRank, using the number of followers and/or social network topology.

In addition, a number of research projects focus on issue of finding the top $k$ influential people in SNS in the context of viral marketing so that the selected people maximize the spread of influence under certain influence propagating models [4], [11], [12], [20], [22], [29], [36], [38]. [13] proposed a new heuristic influence maximization algorithm to maximize the spread of influence under certain influence cascade models.

Recently, [30] presented a model in which information can reach a node via the links of the social network or through the influence of external sources. [39] model social influence patterns based on self-influence and co-influence similarity by incorporating statistical significance in selected node attributes and devise a social influence based distance metric for clustering large graphs demonstrating higher quality and higher efficiency compared to existing approaches. Most of the research on this subject also focused on topology of social networks or static attributes of node state without probabilistic feature and reward schemes.
7 CONCLUSION

We have presented a probabilistic social influence diffusion model (PSI) with incentives. Comparing with previous approaches, our PSI approach has three novel features. First, we argue that social influence is sensitive to dynamic properties of social network nodes and we define an activity-based influence diffusion probability for each pair of nodes instead of uniform distribution of influence based solely on topology of social network. We categorize nodes into two classes: active and inactive. Active nodes can have one chance to influence inactive nodes but not vice versa. Second, in order to express the real world more accurately, we introduce some system parameters, such as a weight function for balancing between $N A$ and $I A$ for computing $w(u, v)$, a $\beta$ damping factor for balancing between the number of activities and the degree of node $u$, and a closeness threshold $\theta_c$ to allow probabilistic propagation of influence across the social network of $N$ nodes. Third but not the least, we incorporate multi-scale incentives into our PSI model as stimuli to further boost the influence diffusion rate and coverage. Finally we conduct an extensive series of experiments on various parameters and to evaluate the performance of our approach. Our experiments show that our PSI model with rewards is more effective in terms of both diffusion rate and diffusion coverage of influence. Although the PSI model presented in this paper centered on quantitative information about activities and a single homogeneous social network, one of our ongoing research efforts is to study how qualitative information about activities, such as context and user profile, may further impact on social influence diffusion rate and coverage. We are also interested in studying social influence diffusions in multi-tier heterogeneous information networks.

ACKNOWLEDGMENTS

The authors acknowledge the partial support from grants under NSF CISE NetSE, SaTC and I/UCRC and Intel ISTC on cloud computing.

REFERENCES