Silence Behavior Mining on Online Social Networks

(Invited Paper)

Qingbo Hu, Guan Wang, Shuyang Lin, Philip S. Yu
University of Illinois at Chicago
Chicago, IL, U.S.A.
{qhu5, gwang26, slin38, psyu}@uic.edu

***Abstract*—Keeping silence is a behavior that widely exists in human society and has been studied in social science for a long time. After a new event occurs, instead of expressing opinions towards it immediately, individuals may choose to remain silence. Similar to a real social network, in online social networks, after observing an interesting event from their friends, users may not decide whether to share it at once due to different reasons. In influence propagation process, we observe that there are three states regarding to one's reaction on an event: *activated state* (shared), *inactivated state* (not shared) and *silent state* (take longer than usual time to make decisions). Silent state is an intermediate status before turning into inactivated or activated state. In this paper, we provide a mathematical definition of "silence" based on the length of hesitating time before a user makes decisions. However, during the hesitation period, silent users behave exactly like those users who already entered the inactivated state. In order to differentiate them in this case, we develop an iterative algorithm, Similarity Interest (SI) model, to identify possible silent users by quantifying the interest of users toward the event. Furthermore, comparing to real social networks, we reveal different behavior of silent users in online social networks and use the *Transient Influence Principle* to explain it. At last, based on experimental results, we design a new model (Diffusion with Silence (DS) model) incorporating Similarity Interest model and two widely used diffusion models (Independent Cascade (IC) model and Linear Threshold (LT) model), in order to capture the silence behavior. Our experiment shows that DS model can more accurately depict the process of information propagation than IC model and LT model do.**

KEY WORDS

Social Network, User Behavior, Information Propagation

I. INTRODUCTION

On the Internet, sharing and commenting are major ways to express a person's opinion towards a certain event. Once an individual observes an interesting event from news websites or sharing lists of his/her friends, s/he may share or comment on it. Such behavior may further impact his/her other friends and is one of the primary reasons resulting in the flourishing of online social network service. It not only increases the potential business opportunities in online social networks, but also makes them become testbeds for the research on human behavior. Previously, researchers have done reasonable amount of studies on the sharing propagation and social influence analysis [1]–[3]. Most of them assumed that there were only two states of a user's attitude towards a certain event: activated [illegible] believe that it is necessary to introduce and analyze another neglected state, which is the silent state. Fig. 1 uses a finite state automata (FSA) graph to show the difference between a traditional model and a model containing a silent state.

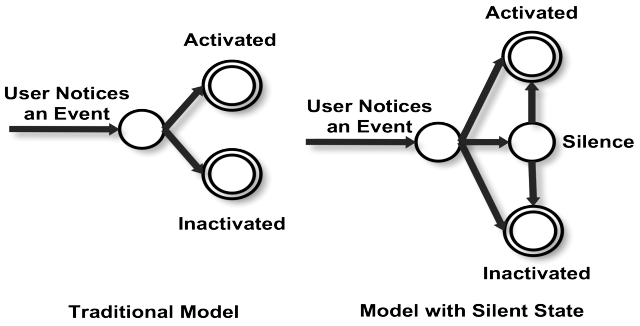


Fig. 1: Traditional Model and the Model with Silent State

A. Motivation

Studies on the silence behavior have many interesting applications in real-life. For example, during the presidential election, there are many swing voters, who are exactly those individuals taking a longer time to make decisions. Discovering them as soon as possible and designing stratagies to win their votes may change the result of the election. In our opinion, introducing the silent state into diffusion models also has both practical and theoretical values. In practice, it may improve the performance of viral marketing strategy. For example, by differentiating silent users with normal inactivated users, we can shrink the advertising cost by only focusing on the silent users. Theoretically, the diffusion model with the silent state is more consistent with empirical studies on sociology or other areas related to the silence behavior [4], [5]. Thus, such model should be more accurate in depicting the information process in real world.

B. Challenges and Solutions

Definition of Silence. Intuitively, one can define those cases when a user has longer hesitating time than his/her average responding time as "silence". Approximately, the hesitating time in the online social network, is the difference between the time when an event firstly appears in the sharing lists of

COLLABORATECOM 2013, October 20-23, Austin, United States

DOI 10.4108/icst.collaboratecom.2013.254085

a user's friends and the time when the user decides to share it or not. Unfortunately, even though we can directly observe the time when a user shares an event, we cannot be aware of the time when a user decides not to share it. To solve this problem, given the fact that a user does not share an event, we use the time when the user shares another event from his/her friends as the approximated time point when the user decides not to share the event. With such approximation, we will be able to provide a mathematical definition of "silence" in the online social network.

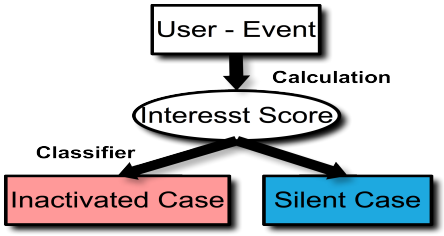


Fig. 2: Identifying Silence Behavior

Discovering Silent Users. Although we can define the silent state based on the length of time before making a decision, such post-hoc definition is not useful in real applications, due to the fact that we often need to identify silent users before they make decisions. More specifically, since silent users behave exactly like inactivated users, how to differentiate them still remains a challenge. During such period, the time when a user makes its decision is unknown, we therefore cannot directly adopt the definition of "silence" to discover a silent user. In this article, we believe the user's interest towards the event is a reasonable indicator to differentiate inactivated users and silent users (in Fig. 2), since silent users may have higher interests to the event than normal inactivated ones. If we can develop a method to quantify the interest of a user towards the event, we may further use these "interest scores" as the feature to classify whether a user is silent or inactivated. We propose an iterative algorithm, (*Similarity Interest model*), based on similarity measures to achieve that. In the experiments, we formulate a binary classification problem using two real online social network datasets to evaluate this algorithm.

Diffusion Model to Capture Silence Behavior. In order to design a reasonable diffusion model to capture the silence behavior. There are two questions to be answered: (1) given an event, how can we judge whether a user is in the silent state? (2) Knowing a user is in the silent state, how to depict a silent user's evolving process? For the first question, instead of assigning uniform probability for users to become silent, we can utilize the classification results based on Similarity Interest model to be more reasonable. For the second question, we examine the factors causing silent user to transit into activated or inactivated state later. During our analysis of the relationship between activated neighbors and a silent user's final decision, we reveal a unique phenomenon (*Transient Influence Principle*) in online social networks. Such phenomenon is related to, but different from the statistical foundations in Social Threshold model [6]. Such discovery implies that when we depict a silent user's evolving process, some popular models (Linear Threshold model [1]) are more suitable than the others (like Voter model [7]). At last, we incorporate two widely used models (Independent Cascade and Linear Threshold) to describe the information propagation process that includes the silent state and its evolving process. The proposed *Diffusion with Silence* (DS) model preserves submodularity, which inherits from traditional diffusion models. Therefore, it guarantees the constant-factor approximation of greedy algorithm solving problems such as influence maximization. In order to show the value of the proposed DS model, we use it and the traditional diffusion models to simulate the propagative process of a most shared event in our dataset. The result shows that the proposed DS model is more accurate in describing the information propagation process.

C. Contributions

Compare to former work on social influence papers, we summarize our main contributions as follows:

- To the best of our knowledge, we are the first one to introduce the concept of ***silence*** from sociology into the computational research of online social networks. Moreover, we provide a formal mathematical definition of "silence".
- In order to identify possible silent users before they make decisions, we try to use user's interest towards the event to differentiate silent users from usual inactivated users. We propose an iterative algorithm, Similarity Interest model, based on similarity measures to achieve this and evaluate its performance on two real-world datasets.
- Comparing to offline social networks, we discover a unique behavior associated with silent users in online social networks. We use the Transient Influence Principle to explain the reason of such difference.
- We extend the traditional diffusion model by incorporating the silent state and its evolving process. The proposed model combines Similarity Interest model and two traditional diffusion models, which preserves submodularity. Additionally, we use experiment on the real event to demonstrate its value.

The rest of the paper is organized as follows. In Section 2, we propose the definition of silent users, as well as the *Similarity Interest model*. In Section 3, we explain our work to study the evolving process of silent users and the proposed Transient Influence Principle in online social networks. In Section 4, we introduce the proposed diffusion model, *Diffusion with Silence model*. Section 5 will present the experimental results to evaluate our work. Section 6 explains previous work in related areas. At last, Section 7 offers the conclusion.

II. Definition of Silence and Similarity Interest Model

A. Silent State

Before starting everything, we need to present the formal definition of the activated state, inactivated state and silent state:

Definition 2.1: **(Activated State)** For each user v and any event e that appears in the sharing lists of v's neighbors, we

say that the final state of user v to event e is activated, if e appears in v's sharing list as well.

Definition 2.2: **(Inactivated State)** For each user v and any event e that appears in the sharing lists of v's neighbors, we say that the final state of user v to event e is inactivated, if e does not appear in v's sharing list.

Definitions 2.1 and 2.2 are very straightforward. We further denote a user's ***response*** to an event the same as sharing the event. We define silent users as follows: on the one hand, for users who are activated to an event, a silent user takes itself a longer than usual time to respond. On the other hand, for users who are inactivated to an event, a silent user takes itself a longer than usual time to change its focus and respond to another event. Formally, we define the response duration and focus-changing duration as follows:

Definition 2.3: **(Response Duration)** Given the fact a user's final state to an event is activated, response duration d is the difference between the first time when a user's friend shares it and the first time when the user responds to it.

Definition 2.4: **(Focus-changing Duration)** Given the fact a user's final state to the event is inactivated, focus-changing duration d' is the difference between the first time when a user's friend shares it and the first time when the user responds to another event.

At last, we can have the definition of silent states:

Definition 2.5: **(Activated Silent State)** For each user v we construct a response duration vector (RDV) D_R=$\{d_1, d_2, d_3 ... d_n\}$, which includes all of v's response durations, where d_i is the response duration to the i^{th} event that v's final state is activated. Using RDV, we define that before becoming activated, a user was in the activated silent state to the event, if the corresponding response duration, say d, satisfies $d > \bar{D_R}$, where $\bar{D_R} = \frac{\sum d_i}{n}$. Let $AS(i)$ denote the set of activated silent users to the event i.

Definition 2.6: **(Inactivated Silent State)** For each user v we construct a focus-changing duration vector (FDV) D_F=$\{d'_1, d'_2, d'_3 ... d'_{n'}\}$, which includes all of v's focus-changing duration, where d'_i is the response duration to the i^{th} event that v's final state is inactivated. Using FDV, we define that before becoming inactivated, a user was in the inactivated silent state to the event, if the corresponding focus-changing duration, say d', satisfies $d' > \bar{D_F}$, where $\bar{D_F} = \frac{\sum d'_i}{n'}$. Let $IS(i)$ denote the set of inactivated silent users to the event i.

Definition 2.7: **(Silent State)** For each event i, the users in the silent state is defined as: $S(i) = AS(i) \cup IS(i)$.

One thing to mention is that if an event is never shared by the user's friend, we do not take it into consideration. In other words, we only consider the internal influence among users. The external influence is omitted, since it is usually intractable. From the Def. 2.7, it is obvious to see that during the hesitation period of a silent user, its behavior is the same as any other inactivated one. That is why we need an approach to identify them in advance. As stated previously, our solution is to use users' interest to identify possible silent users.

Before introducing the *Similarity Interest* (SI) model, we present a topological graph, *User-Event Graph* in the next subsection. User-Event graph is an abstract model on which our study is based. In the subsection C, we are going to introduce how to use Similarity Interest model to quantify a user's interest towards each event. Finally, we will provide the analysis of the SI model. TABLE I lists all the important notations we use in this section.

Notation	Definition
U	set of user nodes
V	set of event nodes
E_{UV}	$\{uv \vert u \in U, v \in V\}$
E_{UU}	$\{uu' \vert u, u' \in U\}$
$s(x, y)$	SimRank score between node x and y
$\mathcal{N}(x)$	$\{y \vert xy \in E_{UV}\}$
$i(u, v)$	SI score representing user u's interest towards event v
R	iteration times of SI algorithm

TABLE I: Notations in Similarity Interest Model

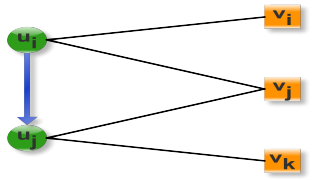


Fig. 3: User-Event Graph

B. User-Event Graph

Definition 2.8: **(User-Event Graph)** Let U, V denote the sets of two different types of nodes, which represent the users and events in an online social network, respectively. In other words, $U = \{u \vert u$ is a user in the online social network$\}$, $V = \{v \vert v$ is an event in the online social network$\}$. E_{UV} denotes the set of edges crossing between U and V, while E_{UU} denotes the set of directed edges inside U. Formally, $E_{UV} = \{uv \vert$ if user u shares event $v\}$ and $E_{UU} = \{uu' \vert$ if user u' follows $u\}$. A User-Event graph is a graph $G = (U, V, E_{UV}, E_{UU})$

Fig. 3 shows a user-event graph. In such a graph, edges in the E_{UV} have no directions, while the edges in E_{UU} are directed, and the direction implies how information flows between two users. Such difference between E_{UV} and E_{UU} is due to that a user shares an event is equivalent to the event shared by the user, yet the relationship between two users are usually asymmetric. For example, in Fig. 3, u_j has followed u_i, while u_i does not follow u_j. In this case, u_j can receive influence from u_i, while u_i cannot receive influence from u_j. This assumption is reasonable, since in a real online social network, friendship between users are usually directed as well. For example, in the Twitter network, if user A follows user B, s/he would see the tweets of B, but user B could not see the tweets of A unless B follows A as well.

C. Similarity Interest Model

Similarity Interest (SI) model is inspired by the framework of *nearest neighborhood model* [8] and used to quantify a user's interest towards events. However, being different from neighborhood model, we cannot observe the user's initial interest towards the event (a.k.a. user-item rate in the neighborhood model), and non-iterative neighborhood model fails to depict the propagative essence of information diffusion. We fix the

first problem by using similarity measurement to approximate the initial interest and extend the model to an iterative one to fix the second problem. The results can be used to identify possible silence behavior in the online social network. In order to ensure the computational efficiency, the similarity measure here is only based on the topology of the User-Event Graph, which does not incorporate other factors (like [9]).

The intuition of the SI model is straightforward: a user's interest towards an event is determined by the similarity between this event and his/her previously shared events. Moreover, a neighbor's interest towards the same event can also influence this user, and the power of such influence is determined by the similarity between the user and its neighbor. One should notice that such influence can be a cascade, which implies that an iterative computation must be used in order to obtain the final interest between a user and an event.

Before the model computes a user's interest towards events, we need the similarity between each user pair, as well as each event pair on G. We use SimRank algorithm [10] to compute these similarity scores. Obviously, the similarity score between two nodes of different types should be zero, since it is unreasonable to talk about the similarity between a user and an event. In fact, this can naturally be achieved by removing edges in E_{UU} when we compute the similarity score. After removing E_{UU}, G becomes an undirected, bipartite graph, resulting that the neighborhoods between two nodes of different types will never have overlap. Therefore, the similarity score between them will always be zero. As a result, combining with the SimRank algorithm, the final similarity $s(x, y)$ between two nodes x and y ($x, y \in U \cup V$) equals to $\lim_{t\to+\infty} s_t(x, y)$, while $s_t(x, y)$ is the t^{th} iteration value of $s(x, y)$ and calculated as follows:

$$s_t(x,y) = \frac{\sum_{i\in\mathcal{N}(x), j\in\mathcal{N}(y)} s_{t-1}(i,j)}{|\mathcal{N}(x)||\mathcal{N}(y)|} \tag{1}$$

where $\mathcal{N}(x) = \{x'|x'x \in E_{UV}\}$. The initial case of $s_t(x, y)$ is

$$s_0(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

SimRank algorithm will usually stop around 10 to 20 iterations [10]. However, it is still very time consuming if we compute every pair of users or events. Therefore, similar to the original SimRank, we also apply pruning during the computation. When two nodes can only be connected through a path having more than two edges, we directly assign the similarity score between them to be zero.

After obtaining similarity scores between users and those between events, respectively, SI model can start to compute user u's interest towards an event v. Unlike measuring the similarity scores, we take E_{UU} into account in this step, since we need to consider individual's interest influences his/her friends. User u's interest towards event v, which is a score $i(u, v)$ calculated according to the following equations:

$$\begin{aligned} i_0(u,v) &= \frac{\sum_{v_i\in\mathcal{N}(u)} s(v,v_i)}{|\mathcal{N}(u)|} \\ i_t(u,v) &= \frac{\sum_{uu_i\in E_{UU}} i_{t-1}(u_i,v)s(u,u_i)}{\sum_{uu_i\in E_{UU}} s(u,u_i)} \\ i(u,v) &= \sum_{t=0\ldots R} i_t(u,v) \end{aligned} \tag{2}$$

In Eq. (2), R is the iterative times of the algorithm. Eq. (2) can be interpreted as follows: a user's initial interest towards an event ($i_0(u, v)$) is the normalized sum of the similarity between this event and the previously shared events of this user. Moreover, the neighbors can further influence a user's interest at step t, which is the normalized weighted sum of the neighbors' interest at step $t-1$. The weight exactly equals to the similarity of the user and this neighbor. A user's final interest towards the event is the sum of each step's influence and its initial interest.

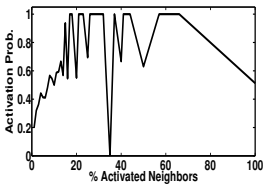
The final scores obtained from SI model are used to train classifiers to differentiate silent cases and inactivated cases. In the Evaluation section, we demonstrate that the final classifiers using this feature can achieve an overall accuracy of around 70%. This is much higher than a random classification baseline.

D. Analysis of Similarity Interest Model

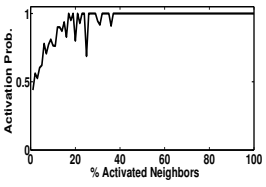
There are several interesting points that can be raised after presenting the SI model. The first one is that when a brand new user or event just joined the network, can the SI model approximate this new user's interest or the interest of existing users towards the new event immediately? Unfortunately, the answer is negative. We cannot connect the new user/event to our existing User-Event Graph, if we do not have any previous historical or related information of them. As a result, we are unable to use SimRank to compute its similarity with the other users or events, a.k.a. all the similarity scores related to this new user or event will always be zero. However, if we can approximate these similarity scores by using other related features rather than topological information, we will be able to apply the SI model to further estimate the "interest". By only using the topological features, SI model is used to approximate the interest of existing users towards existing events.

Secondly, one may ask whether we can use other methods to compute similarity scores instead of SimRank. In fact, the Similarity Interest Model we are presenting here is a framework, so the definition of "similarity" may vary from different perspectives. However, the fundamental idea behind the SI model is well described in our model: a user tends to share the events, which are similar to its previous shared events. Furthermore, its interest towards an event can influence its neighbors' interests.

Thirdly, $i_t(u, v)$ is usually a monotonically decreasing function in term of t. This actually has a real world meaning: the influence of a user's interest will decay with the increase of the steps. However, mathematically, we can only prove that $i_t(u, v)$ is a monotonic function with mathematical induction. Unless we make a strict assumption, such as that $i_1(u, v) \geq$

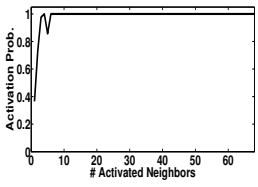


(a) Dataset 1

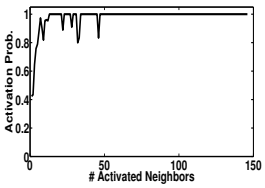


(b) Dataset 2

Fig. 4: X axis is the proportion of activated neighbors for silent users. Y axis shows the probability of silent users becoming activated (the proportion of activated silent users among all silent users who have the same proportion of activated neighbors). Missing points are filled by linear interpolation.



(a) Dataset 1



(b) Dataset 2

Fig. 5: X axis is the number of activated neighbors for silent users. Y axis shows the probability of silent users becoming activated (the proportion of activated silent users among all silent users who have the same proportion of activated neighbors). Missing points are filled by linear interpolation.

$i_0(u, v)$ holds for any u and v, we will be unable to prove that $i_t(u, v)$ is always monotonically decreasing. Therefore, in real world application, we usually use a exponential dampening factor to force $i_t(u, v)$ decreases with the increment of t. We do not apply it here, since we found that in our experiment, most of $i_t(u, v)$ are already monotonically decreasing, and the the results obtained by adding such dampening factor are very similar to the original one.

III. Transient Influence and Silence Evolution

After using the SI model to identify silent cases, we attempt to address the second question that bothers us: except for interest, what will further influence a silent user's final state? We believe that one of the reasons causing them silence is the insufficient number of activated neighbors to draw their attentions. In other words, will the number of activated neighbors influence the activation probability of silent users? We use statistical study and experiment based on our datasets to find the answer.

Intuitively, with the increase of the proportion of activated neighbors, the activated probability of silent users should also increase. This is consistent with models and findings in sociology [4], [6]. Inspired by these previous work, we conduct a statistical study on the silent cases from two datasets (the details of datasets will be introduced in the Evaluation section). We draw two figures to show the relation between the proportion of activated neighbors and the probability of a silent user to become activated in Fig. 4. Surprisingly, we can see that the curves are very unstable (especially in the first dataset). These are different from the same type of curves in sociology, which implies that in our settings, online social networks may be different from traditional real social networks.

We then draw the same figures on these two datasets. Instead of indicating the proportion of activated neighbors, X axis here means the absolute number of activated neighbors. The results are shown in Fig. 5. It is obvious to see that these two curves are much smoother and can serve a better job to show the relationship between activated neighbors and a silent user's activation probability. We use the following principle to explain such phenomenon.

Definition 3.1: **(Transient Influence Principle)** In online social networks, the influence of an activated neighbor towards a silent user is transient, i.e. the accommodated influence from neighbors depends upon the total number of activated neighbors instead of the fraction of activated neighbors.

In real-world social networks, the contacts between two individuals are usually multiple times. It implies that a silent/inactivated user's exposure times to an event are proportional to the percentage of activated neighbors. For example, the spread of virus is a perfect example to illustrate this. Once a person becomes a virus carrier, s/he will try to infect his/her friend every time when they make contact until his/her friend becomes a virus carrier as well. However, in online social networks, when an individual becomes activated to an event, s/he usually only impact his/her friends once. For example, if someone posted a new event in Twitter, it usually only appeared in the updated feed list of his/her friends for a short time and drew their attention once. Then it will be quickly overwhelmed by other newer feeds. This explains that the influence of an activated neighbor to a silent user in online social networks is transient. As a result, the exposure times to the event are approximately the same to the exact number of activated neighbors. Therefore, comparing to the proportion of activated neighbors, the exact number of activated neighbors are more related to the activation probability.

The Transient Influence Principle is important in our research for two reasons: (1) it instructs us what feature should be chosen to build the classifiers in the next paragraph to further study the factor influencing a silent's user's evolving process; (2) it also gives us the hint of choosing the appropriate model to depict silent user's behavior when we design our diffusion model in the Section 4.

In order to further demonstrate the influence of number of activated neighbors towards a silent user's final decision, we apply classification experiments on the datasets. We use the decision tree [11] algorithm to test whether the generated decision tree supports us. Intuitively, because of the Transient Influence Principle, we prefer to use the exact number instead of the fraction of activated neighbors as a feature to train the classifiers to predict the final state of a silent user. However, in order to be more persuasive, we also trained the classifiers based on the proportion of activated neighbors, and the prediction accuracy is lower than the ones using the number of activated neighbors. Therefore, the final reported decision trees are based on the exact number of activated

neighbors. More details will be presented in the Evaluation section.

IV. Diffusion with Silence Model

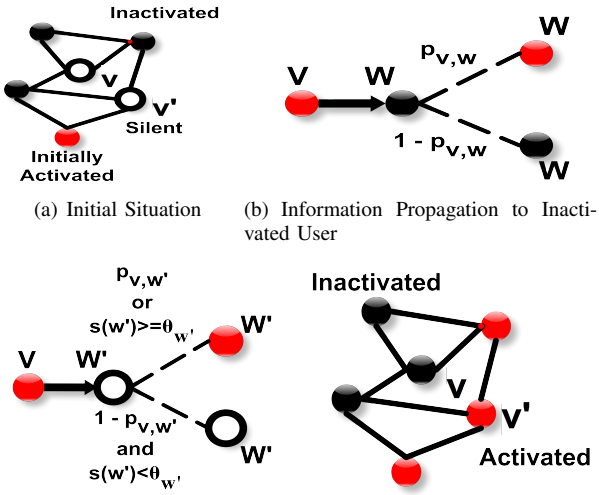
With the help of the previous section's work, we can face our final task: how to extend the traditional diffusion models so that they can depict the silent user's behavior? In this section, we firstly introduce our *Diffusion with Silence (DS)* model and then present the analysis to demonstrate both of its practical and theoretical value.

A. Definition of Diffusion with Silence Model and its Practical Value

Our proposed diffusion model, *Diffusion with Silence (DS)* model, is based on the *Independent Cascade (IC)* model [1] supplemented by the *Linear Threashold (LT)* model [1]. DS model focuses on how to depict the silence evolution process, which is neglected by traditional diffusion models. The extension has two different questions need to be answered: how to identify silent users at first and what extra factors can motivate silent users turning into activated ones. For the first question, we can utilize the results generated by Similarity Interest model. As for the second one, according to the findings in the Section 3, it is reasonable if we combine LT model to simulate the behavior of silent users. Due to the introduced Transient Influence Principle, we do not incorporate other models like Voter model [7] to simulate such behavior. More specifically, in the Voter model, a user's activation probability is proportional to the faction of its activated neighbors. However, in the LT model, a user's activation probability is directly related to the number of activated neighbors. As a result, our model is described as follows.

***Diffusion with Silence model*:** Before the information propagation starts, each user v to the event e can have two different labels: silent or inactivated. The label of v to e, is determined by the classification result using the SI score $i(v, e)$. The classifier is trained on all SI scores except $i(v, e)$ and used to predict the label of v to e according to $i(v, e)$. Furthermore, there is an edge between every two connected users v and w, and the edge has two different weight $b_{v,w}$ ($b_{v,w} \geq 0$, $\sum_v b_{v,w} \leq 1$) and $p_{v,w}$ ($0 \leq p_{v,w} \leq 1$). If v becomes activated at time step t, it will be given one chance to try to activate each of its non-activated (including inactivated and silent) neighbor w. The probability to succeed equals to $p_{v,w}$. If the trial is successful, w will become activated as well at the next time step, which is $t+1$. Otherwise, w stays in the same state in $t+1$. If the neighbor w' is a silent user, despite that v can try to activate w' directly, v will also contribute $b_{v,w'}$ to $s(w')$. $s(w')$ is the accumulated influence from all w's activated neighbors. Formally, $s(w') = \sum_{u \in \mathcal{A}(w')} b_{u,w'}$ and $\mathcal{A}(w') = \{u | u$ is an activated neighbor of $w'\}$. Once $s(w') \geq \theta_{w'}$, w' will also be activated at time step $t+1$. $\theta_{w'}$ is a real number in (0,1] assigned to w'. In the end, all the remaining silent nodes become inactivated.

We use Fig. 6 to explain the Diffusion with Silence model more vividly. In these figures, red nodes are the activated nodes, white ones are silent nodes, and black ones are inactivated nodes. Fig. 6(a) shows that before simulating an influence process, these three types of nodes all exist. Fig. 6(b) demonstrates that an inactivated user can only be directly activated by another active user with a certain probability. On the other hand, in Fig. 6(c), a silent user can become activated in two situations: directly activated by active neighbors or $s(w') \geq \theta_{w'}$, where $s(w')$ is directly dependent on the number of activated neighbors around w'. As shown in Fig. 6(d), in the end of propagation process, there are no more silent users, since they either become activated (v') or inactivated (v).



(a) Initial Situation (b) Information Propagation to Inactivated User

(c) Information Propagation to Silent User (d) The end

Fig. 6: Diffusion with Silence Model

Diffusion with Silence (DS) model combines the IC model and the LT model in a natural way. This model describes that a silent user can be activated through two different ways. Firstly, it can be immediately activated by its surrounding activated neighbors' attempts. In addition, it can also become activated directly under the influence of the number of its activated neighbors. The former process is well depicted by the Independent Cascade part of our model, while the second case is modeled by the Linear Threshold part.

Another thing needs to be mentioned is that the value of each $p_{v,w}$, $b_{v,w'}$ and $\theta_{w'}$ is not directly defined by the DS model. This is similar to the definition of the original IC model and LT model. In [1], $p_{v,w}$, $b_{v,w'}$ and $\theta_{w'}$ are all dependent on the history of the successful propagative process related to these nodes. We use the same definition of $p_{v,w}$, $b_{v,w'}$ and $\theta_{w'}$ in the proposed DS model.

The practical value behind DS model is that unlike the original IC and LT model, we will be able to identify the silent users according to their interest towards the event before simulating an event's propagation process. Actually, these silent users usually have higher interest score (related results are demonstrated in Evaluation section), and they are directly influenced by the increasing number of activated neighbors around them. Therefore, they have a higher chance to be activated than other usual users. As a result, DS model should depict the influence process in a more accurate way than the

IC or LT model. In order to show that, we simulate a real event's propagation process, which has the most shared users in our dataset. The experiment show that starting from the same seed set (early shared people), the final activated set generated by DS has higher precision and recall than the activated set generated by IC and LT. More details of this case study will be presented in the Evaluation section.

B. Analysis of the Diffusion with Silence Model and its Theoretical Value

One of the best properties DS model has is that it preserves the submodularity, which the original IC and LT model have. Submodularity is important, since it guarantees that when solving the influence maximization problem on the DS model, the greedy algorithm will always have a constant-factor approximation [1]. Such property let the DS model can be directly used in the influence maximization problem. Formally, we present submodularity as follows:

Definition 4.1: If we have a function $f(*)$ which maps from a set to a real number, $f(*)$ has submodularity property if and only if for any sets S, T, $S \subseteq T$ and any node v, the following equation always holds: $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$

The reason why the submodularity is important is that for problems like social influence maximization, it can guarantee that the result of the greedy algorithm has a lower bound of $(1 - 1/e)$ of the optimal solution [1].

In order to prove DS model preserves submodularity, we first need to prove that it is equivalent to two sequential processes. Moreover, the second process is dependent on the first one and they are running on two different networks. Formally, we have:

Lemma 1: Diffusion with Silence model is equivalent to two sequential and dependent parts. First part is a Independent Cascade model running on a network having the same structure as the Diffusion with Silence model does. And the second part is a Linear Threshold model running on a derived subnetwork.

Proof: Firstly, let us define $G = (V, E)$ is the online social network used in DS model, where V is the set of nodes representing the users, and E is the set of edges representing the connections among users. Furthermore, we define $G' = (V', E')$, where $V' = \{v' | v' \ is \ a \ silent \ user\} \cup \{u' | v' \ is \ a \ silent \ user \ and \ u'v' \in E\}$ and $E' = \{u'v' | u'v' \in E, \ u' \in V', \ v' \in V', \ and \ v' \ is \ a \ silent \ user\}$. To put it in a simpler way, V' contains all silent users and their neighbors in G, while E' contains each edge between one silent user and one of its neighbors. We define that the direction of edge in E and E' is exactly the direction how the information propagate. For example, $uv \in E$ means that v "follows" u so that v can receive influence from u. We should notice that given a set of initial activated users, S, the information propagation process depicted by the original DS model is equivalent to the following processes: **(1)** Running IC model on G started from S, and the resulting set of final activated users is S'. **(2)** Running LT model on G' started from $S' \cap V'$, and the resulting set of final activated users is S''. **(3)** The final activated users are $S' \cup S''$.

We believe that after we derived G' from G, it is straightforward to see that the DS model is equivalent to the above processes. This is because that in the DS model, LT part only works on silent users and their neighbors. In other words, we are giving a "second chance" for those silent users to become activated. Therefore, it is natural to extract this process from DS model. Of course, after such extraction, the remaining part is exactly the original IC model. ■

***Theorem 1:* (Submodularity) Diffusion with Silence model preserves submodularity property.**

Proof: In order to prove this theorem, let us first define $\sigma(*)$ as a function, which maps a set of nodes to a real number. More specifically, $\sigma(S)$ is the expected number of activated users at the end of influence process, given S as the set of initial activated users. Our job is to prove that under the DS model, $\sigma(*)$ has the submodularity property. To avoid confusion, let us denote $\sigma_{DS}(*)$ as the function on the DS model, while $\sigma_{IC}(*)$ is on the IC model and $\sigma_{LT}(*)$ is on the LT model. Then we have:

$$\begin{aligned}\sigma_{DS}(S \cup \{v\}) - \sigma_{DS}(S) = (\sigma_{IC}(S \cup \{v\}) - \sigma_{IC}(S)) + \\ (\sigma_{LT}((S' \cap V') \cup \{v\}) - \sigma_{LT}(S' \cap V'))\end{aligned} \tag{3}$$

Eq. (3) holds because of Lemma 1. From Lemma 1, we can get that the margin gain of adding a node v into the initial set S is always equal to the sum of the margin gains of the corresponding IC model and LT model. We should also notice that for any $S \subseteq T$, we have $S' \subseteq T'$. As same as the previous definition, S' and T' are the final activated sets from the IC model starting from the initial set S and T, respectively. Therefore, we also have $S' \cap V' \subseteq T' \cap V'$. The submodularity of $\sigma_{IC}(*)$ and $\sigma_{LT}(*)$ have already been proved in [1]. As a result, we have:

$$\sigma_{IC}(S \cup \{v\}) - \sigma_{IC}(S) \geq \sigma_{IC}(T \cup \{v\}) - \sigma_{IC}(T)$$

and

$$\begin{aligned}\sigma_{LT}((S' \cap V') \cup \{v\}) - \sigma_{LT}(S' \cap V') \geq \\ \sigma_{LT}((T' \cap V') \cup \{v\}) - \sigma_{LT}(T' \cap V')\end{aligned}$$

Thus, we have $\sigma_{DS}(S \cup \{v\}) - \sigma_{DS}(S) \geq \sigma_{DS}(T \cup \{v\}) - \sigma_{DS}(T)$. The DS model therefore preserves the submodularity. ■

V. EVALUATION

In this section, we present the experiment results obtained from two real-world datasets. These two datasets are generated from the raw data we crawled from the Twitter network.

A. Description of Datasets

Our datasets are retrieved through API provided by Twitter. The statistics of them are presented in Table II. The first network has users who followed the news account of a university. Unsurprisingly, most of the users are the current students, employees and alumni of the university. The second network contains all the users that the Twitter's official account

	$Dataset_1$	$Dataset_2$
Users	1,113	748
Tweets	17,075	14,432
Friendship connections	10,546	53,639
Silent User-Event Pairs	103,357 (759 activated silent)	905,500 (3,892 activated silent)
Activated User-Event Pairs (excluded activated silent pairs)	863	10,021
Inactivated User-Event Pairs (excluded inactivated silent pairs)	226,515	2,689,198

TABLE II: Statistical Results of Datasets

has followed. Most of these users are the employees or closely related people to Twitter company. Taking a look at the number of edges between these users, we can see the connections among them are very dense. In addition to the relationship network, we also collect all the tweets they have published in the year of 2011 and the time stamps associated with them.

In order to check how an event propagates through the network, we use the URL at the end of each tweet as the identifier of the tweet's content. This is the same as the method used in [12]. In other words, we consider if two tweets have the same URL, they are both talking about the same event. Those tweets containing a URL that only appears in the dataset once are removed, since there is no successful propagation related to them. In the first dataset, we have 1,113 users and 17,075 tweets. As for the second one, we have 748 users and 14,432 tweets.

B. Case Study of Diffusion with Silence Model

First of all, let us use an experiment on the datasets to show that the that the proposed Diffusion with Silence model can depict the real-world information propagation with higher accuracy than the Independent Cascade model and Linear Threshold model. We use the three models to simulate the propagation process of a real event and compare the results. In order to include as many users as possible, we select an event having the most shared users in our dataset and use different ratio of the first shared users as initial seeds. The event we have chosen is shared by several hundreds of users in our first dataset. In the experiment, we use the first 10, 15, 20, 25, 30 users who shared this event as the different initial seed set to generate the cascading process through the three different models, separately. For each seed set and each model, we compare the generated set of final activated users with the real set of activated users in the network. The results presented here are the average ones of 50 times simulations.

Of course, before the simulation, we need to first generate each $p_{v,w}$ for the IC model, and $b_{v,w}$, θ_v for the LT model. Firstly, according to the original models described in [1], we set $p_{v,w} = 1-(1-r_{v,w})^t$, where $r_{v,w}$ is a randomly generated small number between (0, 0.005], and t is the number of former successful transitions from v to w. According to the definition of $p_{v,w}$, if node v has activated w more often in the history, node v can have more powerful influence on w, which means its trial to activate w in the future will have a higher probability to succeed. The interval that we use to generate $r_{v,w}$ is the one that can generate the best result that the IC model can reach among all the different ranges we have tried. Nonetheless, since the DS model uses the same $p_{v,w}$ generated in this step, the range of $r_{v,w}$ is not relevant to the comparing result of the DS model and IC model. Secondly, $b_{v,w} = 1/|\mathcal{N}(w)|$, where $\mathcal{N}(w)$ is the set of neighbor w has followed. Under this definition, we consider each person followed by w has the same impact on w. Furthermore, θ_v is randomly generated from (0,1]. Similarly, we can also change the values of $b_{v,w}$ and θ_v, but since the DS model uses the same $b_{v,w}$ and θ_v, the changes will not influence the comparison results. At last, the DS model is generated according to our description in Section 4. Whether a user is in the silent state at the beginning is classified by the C4.5 algorithm using the Similarity Interest score. The $p_{v,w}$, $b_{v,w}$ and θ_v are the same as those in the IC and LT.

Our comparison is on the quality of the generated final activated users starting from different initial seed set and under the simulation of three diffusion models. We compare the precision and recall of each set of generated activated users given the ground truth of the final activated users. According to the detailed results shown in Table III, starting from every initial seed set, the DS model always generate final activated users with the highest precision and recall. This is especially the case for smaller number of initial seeds. For example, with 10 seeds, the recall under DS is 25% higher than the IC model and 64% higher than the LT model. The reason behind this is already explained in the previous section: the silent user identified by the DS model has a potential interest towards the event, so that they can be more easily activated than normal users. The improvement of DS model comes from two parts: SI model to identify silent cases, and transient influence principle to discover that silent users can be directly activated because of the number of its activated neighbors. Therefore, we further introduce the experiments to evaluate these two parts separately.

C. Evaluation of Similarity Interest Model

In order to evaluate whether the SI model can be used to identify silent cases, we train classifiers using $i(u, v)$ obtained by the SI model as a feature to predict the label of user-event pair u, v (silent or inactivated). Obviously, we need to first extract two kinds of user-event pairs: silent user-event pairs according to Def. 2.7 and inactivated user-event pairs excluding inactivated silent case (for simplicity, we refer them as inactivated user-event pairs later). It is straightforward to get all silent user-events, since we already have the timestamp of each tweet. As for the inactivated cases, we initially extract all the user-event pairs according to Def. 2.2, and then exclude the inactivated silent pairs among them according to Def. 2.6. As shown in Table II, the number of activated silent, inactivated silent, inactivated cases, and activated cases are highly unbalanced. This is caused by a common phenomenon in the online social network datasets, which is the observable successful propagations among users are rare comparing to the unsuccessful ones. Therefore, in order to demonstrate reasonable prediction results, we have to control the number of the three types of cases to prevent silent cases being overwhelmed by inactivated ones or the activated silent cases

	10 Seeds		15 Seeds		20 Seeds		25 Seeds		30 Seeds	
	Precsion	*Recall*	*Precsion*	*Recall*	*Precsion*	*Recall*	*Precsion*	*Recall*	*Precsion*	*Recall*
DS model	0.481	0.512	0.466	0.496	0.486	0.523	0.485	0.571	0.478	0.534
IC model	0.440	0.409	0.439	0.429	0.447	0.442	0.452	0.454	0.446	0.459
LT model	0.379	0.312	0.382	0.332	0.405	0.382	0.417	0.386	0.409	0.486

TABLE III: Comparison Results of the Final Activated Sets of Different Diffusion Models

	Precision		*Recall*		*F-Measure*	
Dataset	*#1*	*#2*	*#1*	*#2*	*#1*	*#2*
Silent cases	0.829	0.89	0.533	0.445	0.649	0.593
Inactivated cases	0.631	0.629	0.879	0.945	0.735	0.755
Weighted Average	0.735	0.76	0.698	0.694	0.69	0.674

TABLE IV: Detailed Prediction Results of Two Datasets

	Dataset 1		*Dataset 2*	
	Mean	Median	Mean	Median
Silent cases	0.968	0.337	0.214	0.173
Inactivated cases	0.171	0.083	0.138	0.137

TABLE V: Statistics of SI Scores of the Datasets

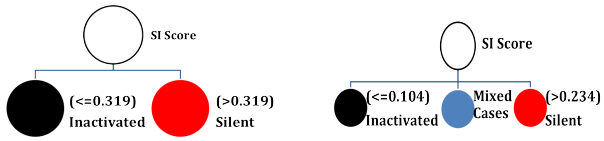
being overwhelmed by the inactivated silent ones. Thus, we use down-sampling to force the number of inactivated silent user-event pairs to be approximately the same as activated silent ones. Similarly, we down sample the inactivated user-event pairs to be approximately equal to the silent ones (union of inactivated and activated silent cases).

In the experiment, we set R, the iterations of Similarity Interest algorithm, to 30, since the results obtained by increasing R are similar to what we present here. We use the decision tree (C4.5) and the obtained SI score as the sole feature to predict the label of user-event pairs. Since we do not have any previous models to compare with, we use the random classifier as the baseline. The results are obtained by using 10-fold cross validation, and the accuracy in the first Dataset is 69.77%, while 69.43% in the second dataset. Other details of this prediction task are shown in Table IV. From these details, we can see that the weighted precisions, recalls and F-Measures are all around 70%. Moreover, the results demonstrate that we obtain higher precision than recall for the silent cases. However, the recall is higher than the precision for inactivated ones.

Furthermore, we draw the generated decision trees from these two datasets in Fig. 7. By looking at the details in these two trees, we find that among all the instances contained in the right node (who has higher SI score) of Fig. 7(a), 85.3% are silent user-event pairs. Similarly, 91% of the cases contained in the rightmost node (who has highest SI score) of Fig. 7(b) are silent cases. Together with the statistical results in Table V, we can conclude that user's interest towards the event in the silent case is usually higher than the one in the inactivated case. This confirms the intuition that silent users may be more interested to the incident than normal inactivated users.

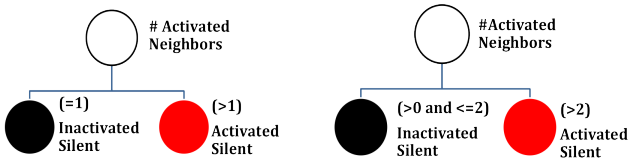
D. Activated Neighbors and Silence Evolution

As introduced in the Section 3, other than the intrinsic interest of the user towards the event, another factor motivating a silent user to become activated is the increasing number of the activated neighbors. In addition to the supporting statistical result in Section 3, we use decision trees generated by a binary classification task to further demonstrate that. In the datasets of



(a) Dataset 1 (b) Dataset 2

Fig. 7: Decision Tree Based on SI Score



(a) Dataset 1 (b) Dataset 2

Fig. 8: Decision Tree Based on Activated Neighbors' Number

previous experiments to evaluate Similarity Interest model, we already have the balanced number of activated silent user-event pairs and inactivated silent ones. Actually, these two types of cases are silent user-event pairs ending in different final states (activated or inactivated) after the evolving process. Therefore, we can directly use activated silent state and inactivated silent state as the two different labels to formulate the classification problem. Furthermore, according to the Transient Influence Principle, we use the number of activated neighbors instead of the proportion of activated neighbors (the prediction accuracy is much lower) as the sole feature to train the decision tree.

The values of accuracy of both datasets in this prediction task are around 70%, and the decision trees are drawn in Fig. 8. One should notice that the reason we do not have a branch of activated neighbors equaling to 0 in Fig. 8 is that all the events here are shared by at least one of user's neighbors. This is because of the data here are generated according to the definitions in Section 2. The generated decision tree in dataset 1 shows that 86.9% of the silent case in the rightmost leaf (has the largest number of activated neighbors) will transit to activated state later on, and the percentage in the second dataset is 81.4%. This confirms that the number of activated neighbors is one of the motivations that the silent user becomes activated.

VI. Related Work

Silence is a behavior that has been widely studied in management research and other areas [4], [5]. Some researchers from these fields are very interested in the consequences of an individual's silence behavior [5], while articles like [4] focus on the reason why people would choose to remain silence.

However, we have found no paper in the data mining area explicitly modeling this behavior when describing the social influence process. Different from the empirical studies in [4] and [5], we focus on providing the quantifiable definition of "silence" and how to incorporate the silent concept into social influence process.

To identify the silence behavior in online social networks, we need an algorithm to approximate the user's interest towards an event. Collaborative filtering is a major area that studies event recommendation for users. Generally speaking, these algorithms can mainly be divided into user-based one [13], item-based one [14], and the combination of them [15]. Unfortunately, none of them is suitable for our task because these models do not contain the relationships among users. Unlike these studies, *Nearest Neighborhood model* proposed in [8] utilized rates from friends to predict a user's own rate to the same item. Its non-iterative calculation is suitable to predict the rate, but it can not be directly applied in our setting. This is due to that the cascading essence of interest diffusion process can not be captured. Inspired by the neighborhood model, we design an iterative algorithm named $Similarity\ Interest\ (SI)$ model to depict the user's interest towards the event, and the similarities in this model are computed by SimRank [10].

The social influence process is a popular topic that has been studied for many years. It inspired many valuable applications, such as viral marketing [16]. Social Influence Maximization problem is the key algorithmic problem behind the viral marketing, which has been shown that several greedy or heuristic methods can provide an approximately good result [1], [17], [18]. The most widely used diffusion model in solving this problem is *Independent Cascade (IC)* and *Linear Threshold (LT)* [1]. However, their generality will not be able to depict specific real-world phenomena, such as silence behavior. As complementary work, there are several papers trying to extend these models to describe more specific phenomena in the influence process. The model from [19] can depict the responding delay after one user is activated by its neighbors. In this model, after each user is activated, it will have a delay of time t to actually respond to the event. However, this model is unsuitable to depict the silence behavior, since all these "delayed" users will still become activated later. Silent users, on the other hand, can turn into inactivated state. We fix this problem by introducing an extra silent state into diffusion model and use the combination of the SI model, IC model, and LT model to capture the silent user's evolving process.

VII. Conclusion

In this paper, we introduce the silence concept from social science into computational area by providing a mathematical definition of "silence". Furthermore, in order to extend diffusion models, we accomplish two prerequisite tasks: silent user identification and examination of silent users evolving process. To accomplish the first task, we design the Similarity Interest (SI) Model to estimate the interest of a user towards the event and further use it to discover silent users. In order to demonstrate how Similarity Interest Model identifies possible silence behavior, we conduct experiments on two real-world datasets. As for the second task, we use both statistical and experimental results to show that the number of activated neighbors can be a motivation to increase the activation probability of silent users. At last, based on the experimental studies for these two tasks, we extend the traditional diffusion models. The proposed Diffusion with Silence (DS) model, includes an extra silent state and incorporates the Similarity Interest model, Independent Cascade (IC) model and Linear Threshold model. The proposed DS model preserves the submodularity inherited from the IC and LT models. To show that it depicts the actual social influence process more precisely, we use the DS model and two baseline models (IC and LT) to simulate the propagation process of a most shared real event in our dataset. Starting from a same seed set, the set of final activated users generated by DS model has a higher precision and recall.

Acknowledgement

This work is supported in part by NSF through grants CNS-1115234, DBI-0960443, and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, and Huawei Grant.

References

[1] D. Kempe and etc., "Maximizing the spread of influence through a social network," in *KDD '03*, 2003, pp. 137–146.
[2] W. Chen and etc., "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *KDD '10*, 2010, pp. 1029–1038.
[3] C. T. Li and etc., "Influence propagation and maximization for heterogeneous social networks," in *WWW '12*, 2012, pp. 559–560.
[4] E. W. Morrison and etc., "Speaking up, remaining silent: The dynamics of voice and silence in organizations," *Journal of Management Studies*, vol. 40, pp. 1353–1358, 2003.
[5] Y. M. Kalman and etc., "Online pauses and silence: Chronemic expectancy violations in written computer-mediated communication," *Communication Research*, vol. 38, pp. 54–69, 2011.
[6] T. W. Valente, "Social network thresholds in the diffusion of innovations," *Social Networks*, vol. 18, no. 1, pp. 69 – 89, 1996.
[7] T. M. Liggett, "Stochastic models of interacting systems," *The Annals of Probability*, vol. 25, no. 1, pp. 1–29, 1997.
[8] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *KDD '08*, 2008, pp. 426–434.
[9] G. Wang and etc., "Influence and similarity on heterogeneous networks," in *CIKM '12*, 2012, pp. 1462–1466.
[10] G. Jeh and etc., "Simrank: a measure of structural-context similarity," in *KDD '02*, 2002, pp. 538–543.
[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., 1993.
[12] E. Bakshy and etc., "Everyone's an influencer: quantifying influence on twitter," in *CIKM '11*, 2011, pp. 65–74.
[13] J. Herlocker and etc., "An empirical analysis of design choices in neighborhood-based collaborative ltering algorithms," *Information Retrieval*, vol. 5, no. 4, pp. 287–310, 2002.
[14] G. Linden and etc., "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
[15] J. Wang and etc., "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *SIGIR '06*, 2006, pp. 501–508.
[16] J. Leskovec and etc., "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, May 2007.
[17] J. Leskovec and etc., "Cost-effective outbreak detection in networks," in *KDD '07*, 2007, pp. 420–429.
[18] W. Chen and etc., "Efficient influence maximization in social networks," in *KDD '09*, 2009, pp. 199–208.
[19] K. Saito and etc., "Selecting information diffusion models over social networks for behavioral analysis." in *ECML/PKDD (3)*, 2010, pp. 180–195.