Friends or Foes: Detecting Dishonest Recommenders in Online Social Networks

Yongkun Li, John C.S. Lui
Department of Computer Science & Engineering, The Chinese University of Hong Kong
Email: {ykli, cslui}@cse.cuhk.edu.hk

***Abstract*—Viral marketing is becoming important due to the popularity of online social networks (OSNs) and the fact that many users have integrated OSNs into their daily activities, e.g., they provide recommendations to their friends on the products they purchased, or they make decision based on received recommendations. Nevertheless, this also opens door for "*shill attack*": dishonest users may give wrong recommendations so as to distort the normal sales distribution. In this paper, we propose a detection mechanism to discover these dishonest users in OSNs. In particular, we present two fully distributed algorithms to detect attackers in both (1) the baseline shill attack and (2) the intelligent shill attack. We quantify the performance of our algorithms by deriving the probability of false positive, probability of false negative and distribution function of time needed to detect these dishonest users. Extensive simulations are carried to illustrate the impact of shill attack and the effectiveness of our detection algorithms. The methodology we present here will enhance the security level of viral marketing in OSNs.**

***Index Terms*—Shill Attack; Online Social Networks; Performance Evaluation**

I. Introduction

In the past few years, we have witnessed an exponential growth of user population in OSNs. Popular OSNs such as Facebook, MySpace and Twitter have attracted millions of active users. Moreover, many users have integrated these sites into their daily activities, e.g., users interact with their friends frequently and they often seek or receive recommendations from their friends before they do any purchase. On the other hand, when one buys a product, she may make recommendations to her friends such that they may be influenced to do further purchase. Such *word-of-mouth effect* makes the purchase behavior spread very fast just like a virus. This phenomenon is called *viral marketing*, and it is very effective to increase sales and revenue for companies [5], [6], [9], [13].

However, viral marketing also opens door for potential security attack as people may behave maliciously to make wrong recommendations. For example, firms may hire some users in an OSN to promote their products, worse yet, they may even consider paying users in an OSN to provide misleading recommendations on their competitors' products. Such dishonest recommenders are known as *shills* [4], [8], and due to the misleading recommendations they made, even if a product is of low quality, people may still be misled to purchase that product. Furthermore, products which have high intrinsic quality may lose out since some potential buyers are diverted to other low-quality products. This is known as the *shill attack* and the aim of this paper is to address the problem of such attack in OSNs, in particular, *how can a user in an OSN discover and detect foes from a set of friends during a sequence of purchases*? The contributions of our work are:

- To the best of our knowledge, this is the first work that provides a mathematical model to describe how to detect dishonest recommenders in OSNs.
- We allow shill attackers to be *intelligent* in sense that they may probabilistically act as honest persons in the hope to avoid being detected.
- We propose fully *distributed* and *randomized* algorithms to detect shills in OSNs, and also provide analytical results on the performance of our detection algorithms.
- Via extensive simulation, we show the severe impact of shill attack and also validate the performance analysis of our detection algorithms.

The outline of the paper is as follows. In Section II, we formalize the model of recommendations in OSNs and state two forms of shill attacks. In Section III, we present two distributed detection algorithms, and derive the performance measures of the algorithms. In Section IV, we show the impact of shill attacks, as well as the effectiveness of our algorithms. Related work and conclusion are given in Section V.

II. Model for Viral Marketing and Shill Attack

In this section, we first provide the notations for viral marketing, then we explore and define the attack strategies. Since we address the problem of detecting dishonest recommenders in an OSN, we model it as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of undirected edges. Due to the fully distributed nature of our detection algorithms, we only focus on a particular node, e.g., node i, which is called the detector. Without loss of generality, we assume detector i has N neighbors, or $|\mathcal{N}^i| = N$.

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$ be a set of M *substitutable* products. Two items are substitutable if they are compatible, e.g., polo shirts from brand X and brand Y are substitutable goods from customers' points of view. In particular, firm F_i produces product P_i. These firms manufacture their products and compete in the same market. We assume people have a long term demand on these products and they will decide which product to purchase at regular time interval. Therefore, $\mathcal{P}$ may represent the set of products like: daily products, milk, cereal or distilled water, etc., in which people perform regular weekly purchase. Let q be the evaluation function that reveals

the *intrinsic quality* of a product and we assume that each product is either of high quality or of low quality. We have

$$q(P) = \begin{cases} 1 & \text{if product } P \text{ is of high quality,} \\ 0 & \text{if product } P \text{ is of low quality.} \end{cases} \quad (1)$$

We model the purchase experience of detector i as a discrete time process. We take the duration between her two continuous purchases as one round, and time proceeds in rounds $t = 1, 2, ...$ In other words, at each round, detector i only purchases once, and after she purchases a product, e.g., P_j, she knows the quality of this product, i.e., $q(P_j)$. Moreover, during the period of one round, she may also receive recommendations from her neighbors in $\mathcal{N}^i$: some may give high or low rating on a given product.

Definition *1:* A *positive recommendation* on product P ($\mathcal{R}_P(P)$) always gives high rating on P regardless of its quality. A *negative recommendation* on product P ($\mathcal{R}_N(P)$) always gives low rating on P regardless of its quality. Formally, we have

$$\mathcal{R}_P(P) = H, \mathcal{R}_N(P) = L,$$

where H means high rating and L means low rating.

Definition *2:* A *correct recommendation* on product P ($\mathcal{R}_C(P)$) gives high rating on P if it is of high quality and low rating on P if it is of low quality. A *wrong recommendation* on P ($\mathcal{R}_W(P)$) gives low rating on P if it is of high quality and high rating on P if it is of low quality. Formally,

$$\mathcal{R}_C(P)=\begin{cases} H & \text{if } q(P)=1, \\ L & \text{if } q(P)=0. \end{cases} \quad \mathcal{R}_W(P)=\begin{cases} L & \text{if } q(P)=1, \\ H & \text{if } q(P)=0. \end{cases}$$

We define $\mathcal{F}_C(t) \in \mathcal{N}^i$ as the set of neighbors of detector i who give her *correct recommendations* at round t. Similarly, we define $\mathcal{F}_W(t) \in \mathcal{N}^i$ as the set of neighbors of detector i who give her *wrong recommendations* at round t. At each round, some neighbors of i may *not* give any recommendation at all, we denote this set as $\mathcal{F}_N(t)$. Obviously,

$$\mathcal{N}^i = \mathcal{F}_C(t) \cup \mathcal{F}_W(t) \cup \mathcal{F}_N(t) \quad \forall i \in V, t = 1, 2, 3, ... \quad (2)$$

The **activities of honest users** can be described as follows. When an honest user gives recommendations to her friends, if she knows the true value of the product (e.g., she buys the product), then she gives correct recommendations. On the other hand, even if she does not know the true value (e.g., she does not buy it), she may still give recommendations based on recommendations received from friends, e.g., giving recommendations based on majority rule, i.e., if more than half of her friends give her positive (or negative) recommendations, then she also gives positive (or negative) recommendations to others, otherwise, she gives no recommendation. In this case, her given recommendation may not be correct. Therefore, if detector i receives wrong recommendations, she can not be sure whether the recommenders are dishonest or not. In other words, neighbors who give wrong recommendations are *not definitely* dishonest, but just *potentially* dishonest. Since it is also possible that dishonest neighbors do not provide any recommendation, the set of all potential dishonest neighbors at round t is:

$$D(t) = \mathcal{F}_W(t) \cup \mathcal{F}_N(t) \qquad t = 1, 2, 3, ... \quad (3)$$

Let us now define the **activities of dishonest users**. We consider two potential attack strategies. As stated before, dishonest users want to promote a particular product, e.g., product P_1, and mislead other users to purchase it. A simple strategy for dishonest users is to recommend to others that P_1 is of high quality regardless of its real quality, and at the same time, recommend other products as of low quality (or bad-mouthing). If a product which is not promoted by dishonest users is of low quality, then dishonest users have no benefit to give high rating on it, therefore, all recommendations on this product must be correct and it has no impact on our algorithms. Without loss of generality, we assume that all other products, i.e., P_2,...,P_M, are of high quality. Denote $\mathcal{R}_a$ as the recommendation of the attacker. Therefore, the **baseline shill attack** can be formally stated as:

$$\mathcal{R}_a = \mathcal{R}_P(P_1) \wedge \left[\wedge_{j=2}^{M} \mathcal{R}_N(P_j)\right]. \quad (4)$$

Another more intelligent attack strategy for dishonest users is to *probabilistically* give correct recommendations on other products instead of keeping performing bad-mouthing. The reason why a dishonest node chooses such attack strategy is to confuse the detector so as to make the detection more difficult. Denote $\mathcal{R}_a^*$ as the recommendation of the attacker using such strategy. Formally, we define this **intelligent shill attack** as:

$$\mathcal{R}_a^* = \mathcal{R}_P(P_1) \wedge \left[\wedge_{j=2}^{M} \Big(\delta \mathcal{R}_C(P_j) \vee (1-\delta)\mathcal{R}_N(P_j)\Big)\right], \quad (5)$$

where $\delta\mathcal{R}_C(P_j)$ means giving correct recommendation on product P_j with probability δ, and the second term represents the bad-mouthing action. We will show in later section how δ may influence our detection mechanism.

III. Distributed Detection Algorithms

In this section, we present our detection algorithms in detail, as well as the performance analysis of the algorithms. We first present the detection algorithm for the baseline shill attack, then we generalize it to handle the intelligent shill attack.

A. Detecting Baseline Shill Attack

In this case, dishonest users give positive recommendations on the product they aim to promote, e.g., P_1, and give negative recommendations on all other products. The baseline attack is not only simple to realize by an attacker, but more importantly, we want to use it to illustrate our detection framework so that readers can gain a better understanding on the detection process. Let $\mathcal{S}^i(t)$ represent the set of potentially dishonest neighbors of detector i until round t. Initially, detector i is conservative and considers all her neighbors as *potentially* dishonest persons, i.e., $\mathcal{S}^i(0) = \mathcal{N}^i$. As time proceeds in rounds, the suspicious set shrinks, i.e., $|\mathcal{S}^i(t)| \leq |\mathcal{S}^i(t-1)|$, and after sufficient number of rounds, we expect that it only contains dishonest neighbors. Based on the above description, we formalize the detection algorithm at round t as follows.

Alg. A1: Detection Algorithm for the Baseline Shill Attack

if ($\mathcal{F}_W(t)$ is empty): /* there is no wrong recommendation */
$\mathcal{S}^i(t) \leftarrow \mathcal{S}^i(t-1)$;
else: $\mathcal{S}^i(t) \leftarrow \mathcal{S}^i(t-1) \cap D(t)$;

The rationale of this algorithm is as follows. If detector i does not receive wrong recommendation, she can not shrink the suspicious set, as her dishonest friends may also make correct recommendations, e.g., product P_1 is of high quality and i buys it. On the other hand, if detector i receives some wrong recommendations, then all her dishonest friends must be in the set $D(t)$, and $\mathcal{S}^i(t)$ can shrink to $\mathcal{S}^i(t-1) \cap D(t)$.

To quantify the correctness of our detection algorithm, we propose two performance measures: (a) *probability of false negative* $\mathcal{P}_{fn}(t)$, and (b) *probability of false positive*, $\mathcal{P}_{fp}(t)$. $\mathcal{P}_{fn}(t)$ is the probability that a dishonest node is wrongly regarded as an honest one at the end of round t. Note that, after t rounds, detector i claims that a node $j \in \mathcal{N}^i$ is dishonest if and only if $j \in \mathcal{S}^i(t)$. Therefore, $\mathcal{P}_{fn}(t)$ can be computed as the probability that a dishonest neighbors of detector i is not in $\mathcal{S}^i(t)$ after t rounds, i.e.,

$$\mathcal{P}_{fn}(t)=\frac{\text{\# of dishonest neighbors of } i \text{ that are not in } \mathcal{S}^i(t)}{\text{total \# of dishonest neighbors of detector } i}. \quad (6)$$

On the other hand, $\mathcal{P}_{fp}(t)$ characterizes the error that an honest node is wrongly regarded as a dishonest one. To formally define this measure, observe that, all neighbors of detector i are initially included in $\mathcal{S}^i(0)$. After t rounds, if an honest node still remains in the suspicious set $\mathcal{S}^i(t)$, she will be wrongly classified as a dishonest node. Therefore, we can compute $\mathcal{P}_{fp}(t)$ as the probability of an honest node not being removed from the suspicious set after t rounds, i.e.,

$$\mathcal{P}_{fp}(t)=\frac{\text{\# of honest neighbors of } i \text{ which are in } \mathcal{S}^i(t)}{\text{total \# of honest neighbors of detector } i}. \quad (7)$$

One thing we need to mention is that, detector i only knows her neighbors' behaviors in each round, i.e., what recommendations her neighbors provide. However, she does not know which neighbors are dishonest, so she cannot count the number of dishonest neighbors. In other words, $\mathcal{P}_{fn}(t)$ and $\mathcal{P}_{fp}(t)$ can not be derived by definition, i.e., Equations (6 - 7). In the following, we focus on the derivation of $\mathcal{P}_{fn}(t)$ and $\mathcal{P}_{fp}(t)$ only based on the information that detector i can gain. In each round, after the purchase, detector i knows the intrinsic value of the product she just purchased and which friends provide her which type of recommendations (positive or negative). Therefore, detector i can accurately decide every received recommendation is a correct recommendation or a wrong recommendation. Based on our detection algorithm A1, if detector i does not receive any wrong recommendation at round t, then that round is not effective in detection and we say round t is not *detectable*. We use a notation $d(t)$ to indicate whether round t is detectable or not, $d(t) = 1$ means round t is detectable, and 0 otherwise. Furthermore, detector i also knows the set $D(t)$ for round t. We use a tuple $(d(t), D(t))$ to represent the information i obtains at round t. The set of all tuples until round t constitutes the *detection history* of i, and we use notation $\mathcal{H}(t)$ to represent it, i.e., $\mathcal{H}(t) = \{(d(1), D(1)), (d(2), D(2)), ..., (d(t), D(t))\}$. Now, we can derive $\mathcal{P}_{fn}(t)$ and $\mathcal{P}_{fp}(t)$ based on the detection history $\mathcal{H}(t)$, and the results are summarized in Theorem 1.

Theorem *1: For the case of the baseline shill attack and using algorithm Alg. A1 for detection, after t rounds, we have* $\mathcal{P}_{fn}(t) = 0$ *and* $\mathcal{P}_{fp}(t) \approx \prod_{\tau=1,d(\tau)=1}^{t} \frac{|D(\tau-1)\cap D(\tau)|}{|D(\tau-1)|}$.

Proof: please refer to the technical report [10]. ∎

Remark: $\mathcal{P}_{fn}(t) = 0$ implies that dishonest recommenders will be identified with probability one. Since $\mathcal{P}_{fp}(t) \to 0$ as $t \to \infty$, it implies that all honest friends will be removed from the suspicious set eventually. Therefore, after sufficient rounds, one can claim with high probability that a node is dishonest only when she is in the suspicious set.

Now let us focus on quantifying the *efficiency* of the detection process. In particular, we seek to determine the expected number of rounds detector i needs to shrink the suspicious set $\mathcal{S}^i(t)$ until it only contains dishonest nodes. Let $\mathcal{R}$ be the random variable denoting the number of rounds needed for detection, and we have the following theorem.

Theorem *2: When the baseline detection algorithm Alg. A1 is used,* $\mathcal{R}$ *follows the distribution of* $P(\mathcal{R} = r) = \sum_{u=1}^{r} \binom{r-1}{u-1} p_d^u (1-p_d)^{r-u}((1-(1-p_{hc})^u)^{N-k} - (1-(1-p_{hc})^{u-1})^{N-k})$, *where* p_{hc} *is the probability of an honest node giving correct recommendations at each round,* p_d *is the probability of a round being detectable and* k *is the number of dishonest neighbors of the detector.*

Proof: please refer to the technical report [10]. ∎

To derive p_{hc}, we assume that, when honest people make recommendations based on their friends' recommendations, they adopt the commonly used majority rule [2], [12], which is a special case of the linear threshold model [7]. Specifically, for an honest person j, if more than half of her friends give positive (negative) recommendations to her, she also gives positive (negative) recommendations to others. Otherwise, she does not give any recommendation. Based on this majority rule, an honest person j gives correct recommendations if and only if she buys a product or more than half of her friends give her correct recommendations. By employing the local mean field technique proposed in [14], [15], we can derive p_{hc}.

Lemma *1: If honest people adopt majority rule to provide recommendations when they do not know the real value of the product, then* p_{hc} *can be derived by following equations.*

$$1-E[Y] = (1-\mu)\sum_{k=0}^{\infty}\sum_{j=0}^{\lfloor \frac{1}{2}(k+1)\rfloor} P_1(k+1)C_k^j E[Y]^j(1-E[Y])^{k-j}, \quad (8)$$

$$1-p_{hc} = (1-\mu)\sum_{k=1}^{\infty}\sum_{j=0}^{\lfloor \frac{1}{2}k\rfloor} P_0(k)C_k^j E[Y]^j(1-E[Y])^{k-j}. \quad (9)$$

where μ *is the market share of the product,* $P_0(k)$ *is the degree distribution of the social network and* $P_1(k)$ *is the degree distribution of descendant nodes in the social network.*

B. Detecting Intelligent Shill Attack

Let us consider a more complicated but more advanced case in which we allow dishonest nodes to be intelligent: they may also give correct recommendations for products $P_2, \ldots, P_M$. In particular, these dishonest nodes give correct recommendations on P_2 to P_M with probability δ at each round. As stated before, the goal of dishonest people is to attract more people to purchase P_1. Therefore, giving positive recommendations on $\{P_2, \ldots, P_M\}$ goes against their objective. The reason why a dishonest node wants to pretend as an honest node is to reduce the possibility of being detected. In this case, if detector i does not receive any wrong recommendation at one round, she faces with the same situation as in the baseline attack case, and she cannot shrink $\mathcal{S}^i(t)$. On the other hand, even if detector i receives wrong recommendations, she still faces the difficulty of distinguishing friends from foes. Because her dishonest neighbors may also give her correct recommendations on $\{P_2, \ldots, P_M\}$. Fortunately, this cannot be the long-term action of the attackers since this goes against their objective. To address the above challenge, we propose a *randomized detection algorithm* which is stated as follows:

Alg. A2: Randomized Detection Algorithm for Intelligent Shill Attack

if ($\mathcal{F}_W(t)$ is empty): $\mathcal{S}^i(t) \leftarrow \mathcal{S}^i(t-1)$;
else:
with probability p: $\mathcal{S}^i(t) \leftarrow \mathcal{S}^i(t-1) \cap D(t)$;
with probability $1-p$: $\mathcal{S}^i(t) \leftarrow \mathcal{S}^i(t-1)$;

One thing we need to emphasize is that the randomized detection algorithm can also handle other sophisticated attack scenarios.To quantify the performance of Alg. A2, we still use the same notations and performance measures defined before. The results are stated in the following theorem.

Theorem *3: When Alg. A2 runs for t rounds, $\mathcal{P}_{fn}(t) = 1-(\frac{1-\delta}{1-\delta^k(1-p_{hw})^{N-k}})^{\sum_{\tau=1}^{t} d(\tau)}$, where p_{hw} is the probability that an honest node gives wrong recommendations on $\{P_2, ..., P_M\}$ at each round, $\mathcal{P}_{fp}(t) \approx \prod_{\tau=1, d(\tau)=1}^{t} \frac{|D(\tau-1) \cap D(\tau)|}{|D(\tau-1)|}$ and $\mathcal{R}$ follows the distribution of $P(\mathcal{R} = r) = \sum_{u=1}^{r} \binom{r-1}{u-1} p_d^u (1-p_d)^{r-u} \big((1-(1-p_{hc})^u)^{N-k} - (1-(1-p_{hc})^{u-1})^{N-k}\big)$.*

Proof: please refer to the technical report [10]. ∎

Remark: We can see $\mathcal{P}_{fp}(t) \to 0$, which means that the suspicious set only contains dishonest recommenders. However, $\mathcal{P}_{fn}(t)$ does not converge to 0, which implies that the dishonest recommenders may evade the detection, but one can still effectively detect dishonest nodes (as we will show in Sec. IV) as long as $\mathcal{P}_{fn}(t)$ is not too large when $\mathcal{P}_{fp}(t)$ converges to 0. Therefore, we can identify *all* dishonest recommenders by performing the detection process multiple times.

To compute p_{hw}, note that, an honest node gives wrong recommendations only when more than half of her friends give her wrong recommendations. Moreover, all dishonest nodes give wrong recommendations on the products they do not promote. By applying the model in [14], [15], we have:

Lemma *2: If honest nodes use the majority rule to provide recommendations when they do not know the real value of the product, p_{hw} can be derived through the following equations.*

$$1-E[Y] = (1-\eta) \sum_{k=0}^{\infty} \sum_{j=0}^{\lfloor \frac{1}{2}(k+1) \rfloor} P_1 (k+1) C_k^j E[Y]^j (1-E[Y])^{k-j}, \quad (10)$$

$$1-p_{hw} = (1-\eta) \sum_{k=1}^{\infty} \sum_{j=0}^{\lfloor \frac{1}{2}k \rfloor} P_0(k) C_k^j E[Y]^j (1-E[Y])^{k-j}. \quad (11)$$

where η is the probability of a person being dishonest.

To estimate p_d, the probability of a round being detectable, note that, for Alg. A1, a round is detectable when $\mathcal{F}_W(t) \neq \emptyset$. But for Alg. A2, a detectable round happens only when detector i receives wrong recommendations, i.e., $\mathcal{F}_W(t) \neq \emptyset$, and it is not ignored by the randomized algorithm. Therefore,

$$p_d = \begin{cases} P(\mathcal{F}_W \neq \emptyset) & \text{for Alg. A1,} \\ P(\mathcal{F}_W \neq \emptyset) p & \text{for Alg. A2.} \end{cases} \quad (12)$$

Note that, one can estimate $P(\mathcal{F}_W \neq \emptyset)$ via the detection history, e.g., approximate it as the fraction of rounds wherein detector i receives wrong recommendations.

C. Overall Detection Algorithm

In previous subsections, we present the detection algorithms at round t for different attack scenarios. Observe that, when we execute the detection algorithm for t rounds, we claim that all nodes in $\mathcal{S}^i(t)$ are dishonest nodes. This claim is accurate with very high probability as long as the algorithm runs for enough rounds. To balance the detection accuracy and detection efficiency, we terminate the algorithm when probability of false positive $\mathcal{P}_{fp}(t)$ is lower than a predefined threshold $\mathcal{P}_{fp}^*$. The termination criterion can be justified because even if the computation of $\mathcal{P}_{fp}(t)$ is not an exact result, it is a good approximation, which is also shown by simulation.

Alg. A3: Overall Detection Algorithm

$t \leftarrow 0$;
do {
$t \leftarrow t+1$;
execute the detection algorithm at round t;
compute the probability of false positive $\mathcal{P}_{fp}(t)$;
} **while** ($\mathcal{P}_{fp}(t) > \mathcal{P}_{fp}^*$)
blacklist all people in $\mathcal{S}^i(t)$ from the neighbor list;

IV. Simulation and Model Validation

Our model aims to detect dishonest nodes who give wrong recommendations in OSNs. Since each user in an OSN performs her own activities continuously, i.e., purchasing a product, providing recommendations to her friends and making decisions on what to purchase based on received recommendations, the OSN evolves dynamically. We first synthesize a dynamically evolving social network which simulates the behaviors of users, then we examine the impact of shill attack and validate the performance analysis of our detection algorithms based on the synthetic dynamic network.

A. Synthetic Dynamically Evolving OSNs

In this subsection, we synthesize a dynamic OSN to simulate the behaviors of users. To achieve this, we need to make assumptions on (1) how people make recommendations to their friends, (2) how people make decisions on purchasing which product, and (3) how fast the recommendations spread.

Firstly, only two types of users exist, i.e., honest users and dishonest users, and their activities are defined in Section II. Specifically, dishonest users follow the baseline attack strategy or the intelligent attack strategy. For an honest user, if she buys a product, she gives correct recommendations, otherwise, she adopts *majority rule*, i.e., if more than half of her neighbors give her positive (or negative) recommendations, then she also gives a positive (or negative) recommendation to others, otherwise, she gives no recommendation.

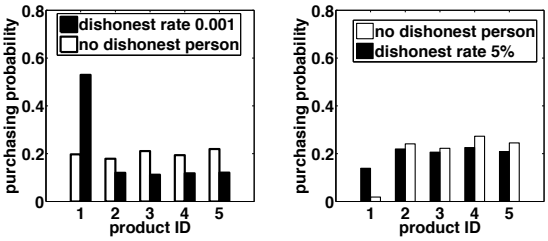
Secondly, to simulate the behaviors of people on deciding to purchase which product, we assume that, when an honest user decides to purchase, she will buy the product which is recommended with the maximum number of effective recommendations. We define the number of effective recommendations as the number of positive recommendations minus the number of negative recommendations. The rationale is that one buys the product on which people give high rating as many as possible and give low rating as few as possible.

Lastly, we assume that recommendations broadcast much faster than users' purchasing rate. Specifically, after one gives a positive/negative recommendation to her friends, they update their numbers of received positive/negative recommendations, if the numbers satisfy majority rule, then they further make recommendations to their friends, and this process continues until no one in the system can make a recommendation, and the whole process finishes before the next purchase instance in the whole system.

To model the evolution of OSN, we assume that it starts from the "uniform" state in which all products have the same market share, and during one detection round of detector i, products are purchased $10\%|V|$ times where $|V|$ is the total number users in the system, i.e., between two successive purchases of detector i, $10\%|V|$ purchases happen. We need to emphasize that the assumptions we make in this subsection are just for simulation purpose.

B. Impact of Shill Attack

In this subsection, we explore the impact of shill attack based on the synthetic system. We employ the GLP model proposed in [3] which is based on preferential attachment [1] to generate a scale-free graph with power law degree distribution and high clustering coefficient. The generated graph has around 8,000 nodes, 70,000 edges and clustering coefficient of around 0.3. The system starts from "0" state where no one has purchased any product and evolves until 10,000 purchases are made. We assume that there are five products, $\{P_1, \ldots, P_5\}$. Our objective is to count what is the fraction of purchases for each product out of the total 10,000 purchases. We run the simulation multiple times to take the average value, and results are shown in Figure 1.



(a) Malicious Competition Case (b) Malicious Cheating Case

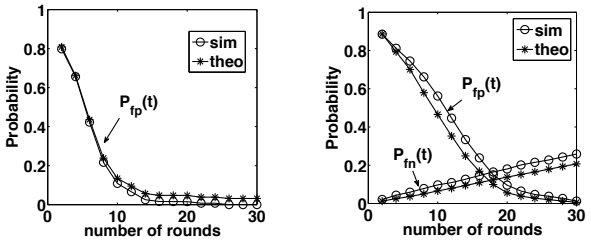
Fig. 1: Shill Attack: dishonest people recommend P_1

We first consider the case where P_1 is of high quality, which is called *malicious competition case*, and it is shown in Figure 1a. Observe that, if all users behave honestly, all five products will be purchased with similar probability, or 0.2. However, if 0.1% of the population are dishonest users, and they just simply employ baseline attack strategy to promote product P_1, then P_1 is purchased with a much *higher probability* which is over 0.5. Figure 1b corresponds to the case where the promoted product P_1 is of low quality, which is called the *malicious cheating case*. Firstly, if there is no dishonest users to disturb the market, P_1 is only purchased with a small non-zero probability. The reason why the probability is not zero is that if a person does not receive any recommendation, she just randomly purchases a product. However, if 5% of the population are dishonest users, then P_1 will be purchased with probability greater than 0.15. In summary, shill attack can distort the normal sales distribution severely.

C. Analysis Validation

In this subsection, we run simulation based on the synthetic system to validate our model and performance analysis. We carry out the simulation many times and take the average value as the final result. We first focus on the performance measures of $\mathcal{P}_{fn}(t)$ and $\mathcal{P}_{fp}(t)$. The results are shown in Figure 2. In both figures, the horizontal axes are the number of detection rounds and the vertical axes represent probability. For the baseline detection algorithm (Alg. A1), since the dishonest user cannot evade the detection, i.e., $\mathcal{P}_{fn}(t) = 0$, we only show the probability of false positive $\mathcal{P}_{fp}(t)$ in Figure 2a. From this figure, we can see that, after only a small number of rounds (< 15), the probability of false positive quickly converges to 0, which means that the detected users are really dishonest with very high probability after a small number of rounds. We have similar result for randomized detection algorithm, which is shown in Figure 2b. Intuitively, if detectors cooperate with each other to share their detection histories, probability of false positive must converge much faster. Therefore, our detection algorithm is very effective in detecting foes and at the same time, quite efficient since it only takes a small number of rounds. However, for the case of intelligent attack, the probability of false negative may not be 0, which means that only a part of dishonest people

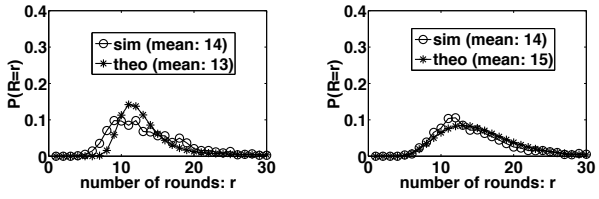
are detected in one detection experiment. One way to solve this problem is to run our detection algorithm multiple times, and at each time, remove a subset dishonest users who are detected before proceeding to the next experiment. Eventually, all dishonest people can be detected with high probability. In our simulation, by running the detection algorithm three times, we are able to detect all dishonest neighbors of the detector. Another way is to adjust the shrinkage probability p in the randomized detection algorithm so as to decrease $\mathcal{P}_{fn}(t)$, however, when p decreases, the expected number of total rounds needed for detection may increase, which shows the tradeoff between the detection accuracy and efficiency.



(a) Baseline detection alg. (b) Randomized detection alg.

Fig. 2: Probability of false positive and false negative.

Let us now consider the distribution of $\mathcal{R}$, which is the number of rounds needed for detection until the suspicious set does not contain honest nodes. The simulation results and theoretical results are shown in Figure 3. Figure 3a shows the results of the baseline attack case where Alg. A1 is used, while Figure 3b corresponds to the intelligent attack case where Alg. A2 is used. In both figures, the horizontal axes are the number of rounds needed for detection and the vertical axes indicate corresponding probabilities. One can observe that our analytical results capture the probability density functions closely and the average estimate, $E[\mathcal{R}]$, is also very accurate.



(a) Baseline detection alg. (b) Randomized detection alg.

Fig. 3: Probability mass function of $\mathcal{R}$.

V. Related Work and Conclusion

Viral marketing is becoming very popular due to the large population base in OSNs. Various work shows its importance and the information spread effect in OSNs [5]–[7], [9], [13]–[15]. However, it opens the door for shill attacks in which malicious users make wrong recommendations to distort the sales market. In [4], [8], authors discuss the impact of misleading comments in recommendation systems, in which there is a *centralized agent* to determine the weights and correctness of all recommendations. In this paper, we defend against shill attack in OSNs based on the idea of shrinking suspicious set [11], and we consider both the baseline shill attack and the intelligent shill attack. We develop a set of distributed and randomized detection algorithms to identify dishonest users who give misleading recommendations in OSNs. Our detection algorithm allows each honest user to independently perform the detection so as to discover her dishonest friends. We provide mathematical analysis on our detection algorithm to quantify the effectiveness and efficiency of our detection mechanism. We also validate our models via extensive simulation. Our detection framework can be viewed as a valuable tool to maintain the viability of viral marketing.

Acknowledgement: this research is supported by the RGC grant 415310 and SHIAE 8115032.

References

[1] A.-L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 1999.

[2] E. Berger. Dynamic Monopolies of Constant Size. *J. Comb. Theory Ser. B*, 83(2):191–200, 2001.

[3] T. Bu and D. Towsley. On Distinguishing between Internet Power Law Topology Generators. *INFOCOM*, 2002.

[4] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions. *CHI Letters*, 5:585–592, 2003.

[5] P. Domingos and M. Richardson. Mining the Network Value of Customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM.

[6] G. J. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 12:211–223(13), 2001.

[7] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the Spread of Influence Through a Social Network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM.

[8] S. K. Lam and J. Riedl. Shilling Recommender Systems for Fun and Profit. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 393–402, New York, NY, USA, 2004. ACM.

[9] J. Leskovec, L. A. Adamic, and B. A. Huberman. The Dynamics of Viral Marketing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 228–237, New York, NY, USA, 2006.

[10] Y. Li and J. C. Lui. Friends or Foes: Detecting Dishonest Recommenders in Online Social Networks, available at http://www.cse.cuhk.edu.hk/%7ecslui/icccn2011tr.pdf. *Technical report*.

[11] Y. Li and J. C. Lui. Stochastic Analysis of a Randomized Detection Algorithm for Pollution Attack in P2P Live Streaming Systems. *Performance Evaluation*, 67(11):1273 – 1288, 2010. Performance 2010.

[12] D. Peleg. Local Majority Voting, Small Coalitions and Controlling Monopolies in Graphs: A Review. Technical report, Israel, 1996.

[13] M. Richardson and P. Domingos. Mining Knowledge-sharing Sites for Viral Marketing. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70, New York, NY, USA, 2002. ACM.

[14] B. Q. Zhao, Y. Li, J. C. Lui, and D. M. Chiu. On Modeling Product Advertisement in Large Scale Online Social Networks. *submitted and under revision*.

[15] B. Q. Zhao, Y. Li, J. C. Lui, and D. M. Chiu. Mathematical Modeling of Advertisement and Influence Spread in Social Networks. *ACM Workshop on the Economics of Networked Systems (NetEcon)*, 2009.