# Vulnerabilities and Countermeasures in Context-Aware Social Rating Services

QINYUAN FENG
Peking University and Georgia Institute of Technology
AND
LING LIU
Georgia Institute of Technology
AND
YAFEI DAI
Peking University

Social trust and recommendation services are the most popular social rating systems today for service providers to learn about the social opinion or popularity of a product, an item and a service, such as a book in Amazon, a seller in eBay, a story in Digg and a movie in Netflix. On one hand, such social rating systems offer great opportunities of convenience and alternative learning environment for many people and decision makers, and on the other hand, they open the door for attackers to manipulate the social rating systems by promoting or demoting some items selfishly or maliciously. Although a fair amount of efforts have been contributed to the understanding of various risks and the possible defense mechanisms to counter such attacks, most of existing work to date has been devoted to study some specific type of attacks and its countermeasure in social rating systems. In this paper, we argue that vulnerabilities in social rating systems and their countermeasures should be examined and analyzed in a systematic manner. We first give an overview of the common vulnerabilities and attacks observed in some popular social rating services. Then, we describe three types of attack strategies in two types of social rating systems, including a comprehensive theoretical analysis of their attack effectiveness and attack costs. Three context-aware countermeasures are presented: (i) hiding user-item relationships, (ii) using confidence weight to distinguish popular and unpopular items, and (iii) incorporating time windows in trust establishment. We also provide an in-depth discussion on how these countermeasures can be used effectivelyto improve the robustness and trustworthiness of the social rating services.

Categories and Subject Descriptors: H 3.3 **[Information Storage and Retrieval]** Information Search and Retrieval – Information filtering.

General Terms: Algorithm, Security.

Additional Key Words and Phrases: Rating system, trust mechanism, reputation system, recommender system.

## 1. INTRODUCTION

With ubiquitous Internet connectivity, we see a growing number of e-Commerce and Internet hosting service providers (e.g., Amazon, Netflix, eBay, Digg, iTunes Store, etc.) offering social rating services, enabling users to choose items based on online opinions of other users. Items here refer to any type of entities, such as products, sellers, transactions, digital contents, search results, applications and so on. Compared with traditional learning environments in which people learn from authority sources, such as reputable news agencies and government designated agencies, many of us today learn and make decisions based on social opinions collected from social media systems through user-generated social ratings. This new kind of learning environments presents great potential to extend our knowledge base and expedites our decision making process. For example, with one click, you can learn the recommendation of a book from the ratings provided by thousands of other customers in Amazon. You can learn the recommendation of a seller from the ratings provided by other buyers in eBay. You can learn the recommendation of a story from the

Authors' addresses: Q. Feng, Peking University, Beijing, China. Email: fqy@net.pku.edu.cn. L. Liu, Georgia Institute of Technology, Atlanta, GA, USA. E-mail: lingliu@cc.gatech.edu. Y. Dai, Peking University, Beijing, China. E-mail: dyf@pku.edu.cn.

opinions provided by other users in Digg. You can learn the recommendation of a movie provided by other viewers in Netflix. In a typical social rating system, the social recommendation of an item is calculated by aggregating the ratings provided by all or a subset of users in the system.

Social rating systems provide ratings of items or trust of users by collecting and aggregating opinions of users in a social community through online ratings or reviews. Social trust and reputation services [Caverlee et al 2010; Mobasher et al. 2007; Srivatsa and Liu 2006; Vu et al. 2010] and online recommendation services [Adomaviciusand Tuzhilin 2005; Resnick and Varian 1997; Stern et al. 2009] are the most popular social rating systems today. These forms of social rating systems have become an increasingly important platform of learning and decision making for many individuals on a wide range of subjects, since online user opinions provide valuable social intelligence to assist users in making decisions about their product selections, their medical treatment choices, their purchase options, and so forth. Such social rating systems do not only have increasing influence on consumers' learning and decision-making behavior, but also offer incentives for good behavior and positive impact on information quality.

A reputation system computes reputation-based trust scores for a set of entities, such as service providers, services, products and other types of items, within a social community such as Amazon, eBay and Facebook. Each entity receives a reputation score based on a collection of opinions that other entities in the community hold about this entity. The opinions are typically obtained initially as ratings or reviews by the reputation engine, which applies a specific reputation algorithm to dynamically compute the aggregate reputation score for each entity based on the received ratings about this entity. Thus, we also refer to the aggregate reputation score as the social trust value of the entity since the collective opinion in a community determines an entity's reputation score. Reputation systems are useful in large online communities where users may interact or share information frequently with others without prior knowledge (e.g., Twitter, Flickr, eBay, etc.). Reputation based trust utilizes the prior experiences of other users in the community to effectively assist a user in making a decision of whether to interact or share information with another user.

A reputation system computes an entity's new social trust value by aggregating the reputation scores of this entity given by those with whom this entity has interacted with in the past. In contrast, a recommender system predicts the 'rating' that a user would give to an item or another entity with which this user may have no prior experience. The rating is computed by integrating the profiles of those entities that are similar to this user in their past rating behavior. The recommendation is made to the user as a ranked list of entities or products or items that are new and interested by the given user. Recommender systems differ from one another based on how such similarity or relevance is defined. In content based recommender services, the specific attributes of the item or the entity of interest are used to predict the ratings that a user would give to the entities or items which this user has no prior knowledge of. For example, in content based seller recommender systems, a user can get a ranked list of sellers from which she may purchase an iPhone 4, based on some attributes of the sellers which are important to this user, such as the reputation trust of iPhone sellers and the unit price offered by the sellers. The concrete decision on how to balance the weights of these attributes is made based on the profile of this particular user. In comparison, Collaborative Filtering based recommender systems make recommenda-

tions about which sellers or which movies or items a user would like to choose based on the social ratings obtained from other users who have similar profiles as this user, and the seller choices they made with respect to their iPhone purchases in the past.

Although reputation systems and recommendation systems are different in terms of the roles they play in social media systems, they have one thing in common: both rely on social rating systems to provide reputation trust or recommendation services. In the rest of the paper, we refer to them as social rating systems for presentation brevity.

On one hand, social rating systems offer great opportunities of convenience and alternative learning environment for many people and decision makers. On the other hand, they also open the door for adversaries to manipulate the computation and use of social ratings by selfishly or maliciously promoting or demoting social recommendation scores of certain items of interest. Recent reports by eBay [Resnick et al. 2006] have shown that the social recommendation of an item not only reflects its popularity but also greatly affects the amount and profits of its sales in the e-Commerce sector. Thus, attackers tend to manipulate the social ratings with increasingly complicated attack strategies, for example, by adding malicious or fake ratings into the social rating systems or by strategically oscillating its attack behavior [Lam and Riedl 2004; Mobasher et al. 2007; Srivatsa and Liu 2006; Vu et al. 2010]. Although a fair amount of efforts have been contributed to the understanding of various risks and several countermeasures have been proposed and adopted in practice, most of existing work to date has been targeted at specific type of attacks and its countermeasure in social rating systems. Attackers continue exploiting new vulnerabilities of the social rating systems.

In this paper, we argue that vulnerabilities in social rating systems and their countermeasures should be examined and analyzed in a systematic manner. We first give an overview of the common vulnerabilities and attacks observed in some popular social rating services today and classify social rating systems into two categories: majority-based social rating systems and trust-enhanced social rating systems. Then, we provide a formal analysis on the effectiveness of the attacks and examine the critical contexts, which are often misused or abused by adversaries in these two representative types of social rating systems. Example contexts include majority rating principles, user-item relationships, and the timestamps of ratings, to name a few. Finally, we describe some context-aware safe guards as possible countermeasures and discuss how these countermeasures can be used effectively to improve the robustness and trustworthiness of the social rating services.

## 2. SOCIAL RATING SYSTEMS

In this section we first introduce the user-item graph as a reference model of social rating systems. Then, we describe some popular social rating systems operational today to illustrate the basic components and classification of social rating systems.

### *2.1 User-Item Graph and Social Rating Examples*

A social rating system has three basic components: users, items, and ratings of users to items. Users can be customers in an e-Commerce system or viewers of a movie. Items can be sellers, products, movies, and so on. A user shares her opinion or experience about an

item with others in the system by providing a rating to the item. Such rating score is typically represented in a numerical scale. Examples are binary (thumb up or down), ternary (such as positive, neutral and negative in eBay), and the five-star models in Amazon.

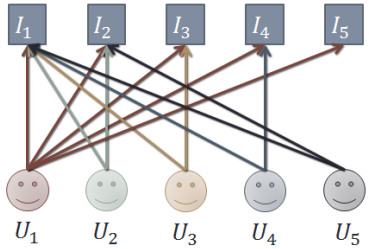



**Figure 1 User-item graph for five users and their ratings on five items**

Figure 1 shows an example relationship between users and items through ratings of users on items. This user-item graph has two types of nodes: five items represented by rectangles and five users represented by circles, and one type of edges originating from users to items, represented by ratings of users on items. A social rating system uses an aggregation function to calculate the social recommendation of an item based on the social ratings collected from the users about that item. We below use the concept of user-item graph to describe four popular social rating systems operational in real life today and discuss the key characteristics and usability of the existing social rating systems.


**Figure 2 Social ratings of a product in Amazon and a seller in eBay**

**Social rating in Amazon:** Amazon is a popular e-Commerce Internet service. When you shop with Amazon, you are provided by a list of products matching your keywords and the social recommendation of each product computed using ratings and reviews provided by other customers in Amazon. Many users find such social recommendations the most attractive feature of Amazon. Figure 2 shows an example product and its social ratings at Amazon. Among the total of 15,200 reviews collected from the customers who have purchased this product in Amazon, 9,803 gives a rating of five stars and the average rating of all 15,200 customers who entered their votes is 4.5 stars. A number of policies are enforced in Amazon to maintain the true value of its social rating based recommendation. For example, Amazon only allows customers who paid for a product to enter a review for this product. No customer can enter more than one review for one product. Therefore, users who never bought a product cannot enter reviews for the product.

**Social rating in eBay:** eBay is a popular consumer to consumer (C2C) e-Commerce platform on the Internet. Both buyer and seller can provide a rating to each other after a transaction. The rating can be positive, neutral or negative. A buyer can rate the sellers using a set of pre-defined attributes, such as "Item as described", "Communication", "Shipping time" and "Shipping and handling charges", at a scale from one star to five stars. Another feature that eBay offers is the socal recommendation aggregated over different periods of time. For example, a buyer can look at the average rating and the number of positive, neutral and negative ratings within the time interval of 1, 6, and 12 months, as shown in Figure 2. The difference between the number of positive ratings and the number of negative ratings, as well as the percentage of positive feedbacks are also given.

**Social rating in Netflix:** Netflix is a well-known Movie website which uses its social rating system to promote movie rentals and sales. A user can rate the movies with one to five stars. Each movie is associated with a social recommendation computed using the average rating score as well as the number of ratings received from users in Netflix. Given the fact that people can watch a movie with different methods, which is different from buying or selling a product, in contrast to Amazon and eBay, Netflix users can rate a movie regardless whether the user has rented or purchased the movie from Netflix. This additional context, though making Netflix more vulnerable to some attacks, is important for the Netflix's business model, as it encourages users to share their opinions on movies, which they have seen prior to becoming a member of Netflix. This enables Netflix to generate a personalized and time-sensitive list of movies, for each user at each of her visits, by incorporating both her most recent rating inputs and the most recent social recommendations of relevant movies from the Netflix user community. This personalized list contains the movies that Netflix believes that this user most likely will rent or purchase during this visit.

### *2.2 Majority-based Social Rating Systems*

Majority-based social rating systems utilize all ratings of users on an item to compute its aggregate recommendation score. Thus, items that receive ratings from a majority of users tend to have higher social recommendation scores than those that receive ratings from a smaller number of users. Most of the social rating systems operational today, such as those used in Amazon, eBay, Digg and Netflix, are using the majority rule withan aggregate function used to summarize the majority of the total ratings. Recall the discussion in the previous section, the most widely used aggregation functions are average, count and difference function.

- **Average function**: it calculates the average value of all social ratings. For example, the average star rating for a product in Amazon is calculated by using the average stars collected from customers about this product.
- **Counting function**: It counts the total number of ratings collected from users or under a given category. Examples include the number of reviews in Amazon or Netflix, the number of positive ratings in eBay, and the number of diggs for a story in Digg.

- **Difference function**: It calculates the difference between two count summaries. Recall Figure 2, the feedback score of a seller in eBay is calculated by using the number of positive feedbacks to minus the number of negative feedbacks.

The use of all three aggregation functions shares a common goal, which is to infer the social recommendation score of an item, such as a product in Amazon, a seller in eBay, a story in Digg and a movie in Netflix. This social recommendation score represents the majority opinions of the raters. By the majority rule principle, we mean that the social recommendation of an item calculated by an aggregation function should reflect the majority of the ratings of all users. Without loss of generality, we below formally define asocial rating system using the average-function based majority rule principle and a binary rating model, which we use for vulnerability analysis and countermeasure discussion in the rest of the paper.

Let $U_i$ represent a user and $I_j$ represent an item in the system. The rating from $U_i$ to $I_j$ is denoted as $R_{ij}$ with "+1"referring to good and "-1" referring to bad. To simplify our discussion, we assume that an honest user always provides an honest rating to an item. Therefore, if her true experience of an item is positive ("+1"), the honest users will give "+1" rating, and vice versa. On the other hand, a dishonest rating on an item refers to giving "-1" to an item when the user has a true positive experience with the item, and vice versa.

Now we show how to infer the social recommendation score using the average aggregate function of user ratings and the majority rule principle. Let $S_j^{+1}$ and $S_j^{-1}$ denote the set of users who rate item$I_j$ with "+1" and "-1" respectively. The aggregated rating of item $I_j$, denoted as $R_j$, is computed using the following average aggregate function.

$$R_j = \begin{cases} \frac{|S_j^{+1}|}{|S_j^{+1}|+|S_j^{-1}|}, & |S_j^{+1}| + |S_j^{-1}| > 0 \\ 0.5, & |S_j^{+1}| + |S_j^{-1}| = 0 \end{cases} \quad (1)$$

Let S denote the set of users in the system. The majority-based social recommendation score for item $I_j$ is computed using the following formula:

$$Q_j^S = \begin{cases} +1, & R_j \geq 0.5 \\ -1, & R_j < 0.5 \end{cases} \quad (2)$$

This formula calculates the social recommendation score of $I_j$ based on whether the majority of the ratings are "+1" or "-1", which is achieved by using a threshold of 0.5 in this case. It means that if there are more than 50% of users who rate $I_j$as a good item, the system will recommend it as a "good" item; otherwise, the system will recommend it as a "bad" item. This threshold varies according to the specific need of a social rating system.

It is important to note that the formula (2) is not resilient to dishonest ratings. We will show in Section 3 that when the number of dishonest ratings is relatively high, the system can easily make incorrect recommendations. Thus many researchers have proposed to use trust-enhanced social rating systems to counter such basic vulnerabilities, namely using trust to evaluate whether or not the rating behavior of a user is honest.

### *2.3 Trust-enhanced Social Rating Systems*

Ina trust-enhanced social rating system, the key component that is different from majority based rating systems is the concept of trust, namely how to define, infer, and use trust. In general, the context of **trust** is used to evaluate whether or not the rating behavior of a user are honest. Thus, the trust of a user can be defined as a measurement of the credibility of her rating. Instead of aggregation of user ratings on an item, trust enhanced social rating systems will produce the social recommendation score by using a trust-weighted aggregation of users' ratings. There are several ways to define trust [Caverlee et al 2010; Mobasher et al. 2007; Srivatsa and Liu 2006; Vu et al. 2010; Xiong and Liu 2004]. For presentation clarity, we choose to use the credibility of a user in performing her rating of items as the measurement of trust. Then, based on the calculated trust scores of the users, we can weigh the ratings of users by assigning higher weight values to the ratings of those users who are more trustworthy and lower weight values to the ratings of the users who are less trustworthy. Thus, the overall social recommendation score on an item will be computed using the trust-enhanced majority rule principle, which use the trust score weighted aggregation of social ratings to compute the overall social recommendation scores for items based on both the ratings and the rating credibility of individual users in addition to the majority rule principle. We below formalize the trust-enhanced social rating model to show how the trust-enhanced recommendation scores are computed and what types of vulnerabilities it may have. We choose to use the beta-function as the reference trust model in the rest of the paper [Feng et al. 2010; Josang and Ismail 2002] to compute the trust scores of users and infer the social recommendation scores of items.

The fundamental assumption in the majority rule based trust inference is that a user's rating of an item is considered more trustworthy if most of her ratings agree with the ratings from the majority of the raters, and vice versa.

Let $Q_j^S$ denote the majority score of the ratings on item $I_j$ and $R_{ij}$ denote the rating given by user $U_i$ to $I_j$. Under the binary rating model, the honest behavior and dishonest rating behavior of $U_i$ with respect to $I_j$ can be defined as follows: $R_{ij}$ is an honest rating if $R_{ij} = Q_j^S$ or $R_{ij} = 1$ and $R_{ij}$ is a dishonest rating if $R_{ij} \neq Q_j^S$ or $R_{ij} = 0$.

This definition implies that if the rating $R_{ij}$ agrees with the overall social recommendation of $I_j$ inferred by the system (consistent to the majority of the ratings received on item $I_j$), then this rating is counted as the honest rating of $U_i$, and, otherwise, it will be counted as a dishonest rating. In addition to the credibility of the rating $R_{ij}$, another two important factors are the set of honest ratings and the number of dishonest ratings that $\mathrm{U_i}$ has given in the system, and we use $H_i$ to represent the set of honest ratings that $U_i$ has given to $I_j$ and $D_i$ to represent the set of dishonest ratings that $U_i$ has given to $I_j$.

Based on the collected evidence on both the honest and dishonest ratings given by user $\mathrm{U_i}$, we can infer the trust ($\mathrm{T_i}$) of user $\mathrm{U_i}$ using the beta function defined in Equation (3):

$$\mathrm{T_i} = \frac{|\mathrm{H_i}|+1}{|\mathrm{H_i}|+|\mathrm{D_i}|+2} \qquad (3)$$

The beta function in Equation (3) is widely used to compute trust due to a number of useful properties it offers: When a user has given no rating in the system, its trust score is 0.5. When a user has behaved more honestly, its trust will increase. When a user has behaved

more dishonestly, its trust will decrease. A user with an increase of rating behavior in the system will have more stable trust score, as the change of its trust score is less frequent and less significant. A fair amount of trust-enhanced rating algorithms have incorporated the beta function and its mathematics foundation has also been well studied [Josang and Ismail 2002].

With the trust scores computed using the beta function we can overcome the vulnerability of the majority-based social recommendation by incorporating trust-weight aggregation function. Let$S_j^{+1}$and $S_j^{-1}$denote the set of users who rate item$I_j$ with "+1" and "-1" respectively. We can compute $R_j$ using the following trust-enhanced formula instead of (1):

$$R_j = \frac{\sum_{S_j^{+1}} T_i}{\sum_{S_j^{+1}} T_i + \sum_{S_j^{-1}} T_i} \qquad (4)$$

Equation (4) uses trust as weight values to aggregate the ratings provided by different users. It gives larger weight values to the users with higher trust scores and assigns smaller weight values to the users with lower trust scores. Clearly, the social recommendation score computed based on Equation (4) is more attack-resilient compared to the one computed based on Equation (1).

## 3. VULNERABILITIES AND ATTACKS IN SOCIAL RATING SYSTEMS

### *3.1 Attacks in Real Life*

We first describe three interesting attacks happened to Amazon. First, according to [Harmon2004], the Amazon's Canadian site mistakenly revealed the true identities of some book reviewers. Surprisingly, this incident also revealed that a sizable proportion of those exposed reviews turned out to be written by the publishers and authors of the books being reviewed to promote their own work, or by their competitors to demote the work of their competitors. Second, some latest evidence [Parsa 2009] shows that some reviewers in Amazon were paid to provide either exaggerate or fake reviews in Amazon. Furthermore, it is reported [Badger 2010] that the top-ranked reviewers in Amazon will suddenly receive many malicious votes as attackers can increase the ranking of their own faster by attacking top-ranked reviewers.

Similarly, in eBay, some sellers sold jokes at 65 cents each to increase their trust scores [Brown and Morgan 2006]. Even worse, the most popular Chinese e-Commerce website Taobao, similar to the function of eBay, is reported [Taobaozuan 2010] to have generated a new kind of grey economy, in which attackers are selling a promotion service for sellers with 66 positive feedbacks at the cost of 10 US dollars or so. [Saleh 2008] summarizes the interviews with Digg attackers in terms of the attack methods they used and the profits they can achieve. A recent report [Tran et al. 2009] shows the evidence in Digg about the Sybil attack [Douceur 2002] as well as a number of obvious bogus stories, such as advertisements, phishing articles and obscure political opinions with fake promotion in YouTube.com, political campaigns promote positive video clips and hide negative video clips by inserting unfair ratings [Hines 2007]. In Twitter, some user won the award of

"Best Producers of Short Content in Twitter" through purchasing online votes [Zarrella 2009]. One famous attack in IMDB is called from "the Batman" to "the Godfather", and attracted a lot of attention [Sciretta 2008]. Concretely, when the movie "The Dark Knight" came out, its fans launched a campaign by asking a huge number of people to rate this new movie as ten stars in IMDB, and at the same time, also rate the top-1 ranked movie "The Godfather" with one star (the lowest rating). Only after a short period of time, they succeeded in promoting this new movie to be the top-1 ranking.

All these incidents in real-world social rating systems are true stories happening right before our eyes. In order to effectively counter such attacks, we need to understand the attack model and the vulnerabilities in social rating systems today as well as the goals and methods of attackers.

### *3.2 Attack Model*

The goal of attacking a social rating system can be characterized simply as the effort of minimizing the true value of an item in terms of its social recommendation score at the least cost of an attacker. Several contexts may impact on the effectiveness of an attack, including the concrete methods of how the social recommendation scores are computed and inferred (w.r.t. attack goal), the specific rating model used by the attacker, and the specific cost model of the attacker. We below first formally define the attack goal, the rating mode of attacker and attack cost based on the binary rating model. Then, we analyze three types of attacks that are detrimental to many existing social rating systems.

***The Goal of Attacker***

In principle, the goal of an attack can be transformed into a basic case, namely: to make sure the social recommendation score produced by the system is different from its true value. By true value we refer to the social recommendation score produced in the system when everyone is giving honest ratings. Using the binary rating model, we only consider the basic case of whether a user gives a dishonest rating on an item. To further simplify the modeling, we assume that all items have the true social recommendation score of"+1". We can define the goal of an attack by making the system to infer the social recommendation score of its target item $\boldsymbol{I_T}$ to be "-1".

$$\boldsymbol{Attack\ goal}: \quad \boldsymbol{Q_T^S = -1} \tag{5}$$

Under the context of majority-based social rating systems, we can transform this goal into the comparison between the number of negative raters and the number of positive raters for the target item. The attack goal is to ensure that the negative ratings prevail over the positive ratings as shown in Equation (6).

$$\boldsymbol{Attack\ Goal}: \quad \left|\boldsymbol{S_j^{-1}}\right| > \left|\boldsymbol{S_j^{+1}}\right| \tag{6}$$

Under the context of trust-enhanced social rating systems, we can transform the goal into the comparison of the summation of trust scores between the negative raters and the positive raters as shown in Equation (7), where $S_j^{+1}$ and $S_j^{-1}$ denote the set of users who rate item $I_j$ with "+1" and "-1" respectively.

$$\textbf{\textit{Attack Goal}}: \quad \sum_{U_i \in S_j^{-1}} T_i > \sum_{U_i \in S_j^{+1}} T_i \qquad (7)$$

For non-binary rating models, the attack goal can be transformed to comparisons among the summations with different scales of ratings.

***Attack model***

In order to attack a social rating system, the attacker achieves its goal by making the system giving a dishonest rating to each of its target items ($I_T$). Concretely, when using binary rating, the attacker launches an attack by changing the social recommendation score of an item in its attack target from "+1" to "-1" through providing ratings in the system. The rating model used by the attacker typically consists of two steps:

Step 1: Add some malicious users into the system.

Step 2: Use these registered malicious users to provide malicious ratings.

In the first step, an attacker legitimately registers a set of user IDs according to the user registration procedure of the system. In the second step, the attacker needs to select a set of items and provide malicious ratings to these items in order to best achieve its goal. A challenging task for attackers is to decide which items to choose as their attack target and how to rate them and other items in the system in order to achieve its goal with the least effort.

To address this question, we categorize all items in the system into four categories:

- **Target set** ($I^T$): the items that the attacker wants to change their social recommendation scores from their true values. To simplify the discussion without loss of generality, in the rest of the paper, we assume that there is only one item in the target set and we use $I_T$ to denote this target.
- **Honest set** ($I^H$): the items that do not belong to the target set and on which the attacker will provide honest ratings.
- **Dishonest set** ($I^D$): the items that do not belong to the target set and on which the attacker will provide dishonest ratings.
- **Bypass set** ($I^B$): the items on which the attacker will not provide any ratings.

The main idea of this classification is to study and compare the cost for three types of attacks. The first and most intuitive type of attacks is called the **direct attack**. In this type of attacks, an attacker directs all his effort to providing dishonest ratings only to the items in the target set. The second type of attacks is somewhat indirect and called the **disguise attack**. In addition to providing dishonest ratings only to the items in the target set, the attacker also provides honest ratings to a proper subset of other items that are not in the target set to gain the trust of other users and the system and disguise its attack goal. The third type of attacks is the most indirect and most sophisticated of all three. The attacker provides dishonest ratings to the items in both the target set and the dishonest set, in addition to provide honest ratings to the items in the honest set. Such attack strategy misguides the system not only about the true value of the targets but also the true value of the items in the dishonest set. We refer to this third type of attacks as the **misguidance attack**. We will discuss each of these three types of attacks in the subsequent sections.

***The Cost Model of Attack***

Based on the two-step rating model of an attack, the cost of an attack can be captured by two parameters: the number of malicious users needed ($N_M$) and the number of malicious ratings ($K_M$) needed to succeed an attack, namely to achieve the goal of an attack. Therefore, we define the cost of an attack in the form of a vector of two elements, denoted by $<N_M, K_M>$. This attack cost model indicates that the number of malicious users ($N_M$) and the number of malicious ratings ($K_M$) are the main cost factors of an attack. Different attack strategies will demand attackers to pay different costs in terms of these two parameters. We below discuss how to compute the cost of an attack, concretely how to infer these two parameters in each of the three types of attacks, for both majority-based social rating systems and trust-enhanced social rating systems.

Consider that in most social rating systems, it is hard to register a new user ID than providing a rating of a user on an item, we first try to minimize the number of malicious users needed and then minimize the number of malicious ratings needed using this minimal number of users. Given the space constraint of this paper, we will omit the discussion on the attack strategies where the attackers may want to first minimize the number of malicious ratings or to find a balance between the number of malicious users needed and the number of malicious ratings needed.

### *3.3 Direct Attack*

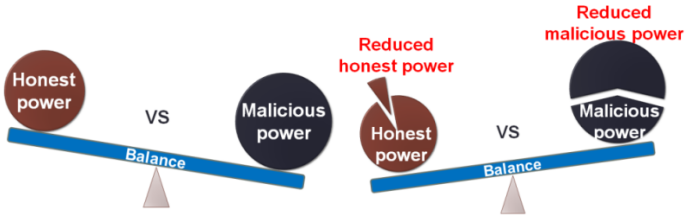
Direct attack represents the most intuitive and simplest attack strategy, in which attackers provide dishonest ratings only to the items in the attack target set. We represent this attack by using our attack model in Section 3.2, namely the four item sets that an attacker will utilize to exercise an attack: $\{I^T \neq \emptyset, I^H = \emptyset, I^D = \emptyset, I^B \neq \emptyset\}$. This attack strategy is naïve and assigns malicious ratings only on the target items of the attack.

It is interesting to note that the direct attack has been found in nearly all the real-world social rating systems which we have discussed before. The most illustrative example of the direct attack is called the **bad-mouthing** attack [Dellarocas2003; Hoffman et al 2007; Sun el al 2006]. Under a bad-mouthing attack, the attacker will register $n_M$ malicious users and use each of them to provide a dishonest rating on each item in the target set. Therefore, the number of malicious users is the same as the number of malicious ratings for the bad-mouthing attack to each item in the target set ($I^T$).

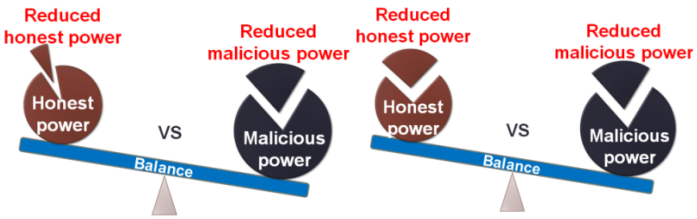
***Attack effect analysis in majority-based social rating systems***

Figure 3(a) illustrates the direct attack with Equation (6) as its attack goal in a majority-based social rating system. We can consider that the honest users and the malicious users stay at the two ends of the teeterboard in this battle. If either end can gain the majority, it will win the balance. Thus the attacker only needs to summon enough ratings to win the balance. Considering the case where only one item ($I_T$) in the attack target set and let $S_j^{+1}$ denote the number of users who rate item$I_j$ with "+1". In order to launch a successful direct attack on this item, the number of dishonest ratings that the attacker needs to provide is $|S_T^{+1}| + 1$. Thus, the attack cost can be modeled by $<|S_T^{+1}| + 1, |S_T^{+1}| + 1>$, indicating that the attacker only needs $|S_T^{+1}| + 1$ malicious users with each providing one dishonest rating on the target item. This simple example shows three important observations. First, the attack cost is defined by the number of malicious users and the number of malicious

ratings that are needed for a successful attack. Second, given a target set of items to be attacked, the dominating factor of the attack cost is the number of honest users who rate on the items in the target set. When the attack is targeted at an item that has a lot of honest raters, the attacker needs to pay much higher cost to launch a successful attack, compared to targeting at an item that has relatively smaller number of honest raters. Third, the number of ratings on an item usually follows the power-law distribution in a social rating system [Faloutsos et al 1999]. This means that only a few items will receive a lot of ratings from many users, and most of the items will receive only a few ratings from a small number of users. Thus, the attackers can easily manipulate the social recommendation score of the items that are unpopular in the social rating systems.




**a**) Direct attack on majority-based social rating systems b) Direct attack on trust-enhanced social rating systems




c) Disguise attack on trust-enhanced social rating systems d) Misguidance attack on trust-enhanced social rating systems

**Figure 3** Honest and malicious power battle under different types of attacks

***Attack effect analysis in trust-enhanced social rating systems***

Based on the above analysis, it is clear that the direct attack needs to pay much higher cost to succeed in a trust-enhanced social rating system. This is because the ratings of users on items are now aggregated by using their trust values as the weights on their ratings. Comparing with majority based social rating systems, the trust-enhanced systems are able to provide two levels of defense under direct attacks. First, it can identify and assign low trust values to those malicious users who only provide dishonest ratings on certain items (in the attack target set) since their ratings are obviously inconsistent with the rest of the ratings received by those items prior to the event of a direct attack, especially when the items under attack have received a fair amount of honest ratings. Second, by using trust-based weighted aggregation of ratings, the trust-enhanced social rating system can reduce the impact of those users under the suspicion of providing dishonest ratings during the aggregation. However, the flip side of using trust-enhanced rating aggregation to reduce the malicious power, some honest users may be mistakenly identified as behaving dishonestly when they rate only those items that are unpopular and have less raters and some of such unpopular items become the target of attack.

Figure 3(b) provides an illustration of the honest and malicious power battle under direct attack in a trust enhanced social rating system with Equation (7) as its attack goal. Consider a bad-mouthing attack (a simple form of direct attack) where the malicious users only rate items in the target set and one dishonest rating per target item. Let $S_j^{+1}$ denotes the set of users who rate item$I_j$ with "+1" and $T_i$ denotes the trust value of the user $U_i$. Since the initial trust of a malicious user (without prior honest rating history) is only 0.5 with beta-function, by Equation (7) the attacker's goal can be transformed into $0.5 \times N_M > \sum_{U_i \in S_j^{+1}} T_i$. Thus, the attack can succeed when the number of malicious users ($N_M$) satisfies the condition of $N_M > \sum_{U_i \in S_j^{+1}} 2T_i$. Therefore, we can infer the attack cost for an item $I_j$ as $<\left\lfloor \sum_{U_i \in S_j^{+1}} 2T_i \right\rfloor + 1, \left\lfloor \sum_{U_i \in S_j^{+1}} 2T_i \right\rfloor + 1>$. Let $T_H$ be the average trust score of an honest user. We can revise the attack cost as $<\lfloor 2T_H \times |S_T^{+1}| \rfloor + 1, \lfloor 2T_H \times |S_T^{+1}| \rfloor + 1>$. This means that the attack cost depends on the average trust score of honest raters and the total number of users who rate on the item $I_j$. If the trust scores of the honest users are very high, $T_H$ will approach the highest value of one, the cost of attacking the beta-function based social rating system will be doubled compared to the cost of launching the same direct attack to the average-function majority-based social rating system.

***Attack cost analysis***

We now illustrate the attack cost in terms of the minimum number of malicious users and the minimum number of malicious ratings needed to launch a successful attack in both average-function based and beta-function based (trust-enhanced) majority rating systems using the example item-user graph shown in Figure 1.

For an average-function social rating system, assuming $I_1$ is the attack target, Figure 1 shows that the total number of honest raters for item $I_1$ is 5, thus the minimum number of malicious users that the attacker needs for launching a successful direct attack should be 6, assuming that each of the 6 malicious users provides one dishonest rating to $I_1$. Therefore, the attack cost can be represented as <6,6>.

For a beta-function social rating system, we need to compute the trust scores for all five honest users based on Equation (5). Given that $U_1$ has provided 5 honest ratings and her trust score is calculated as $\frac{5+1}{5+2} = 0.86$; each of the other four users has provided 2 honest ratings and their trust score is calculated as $\frac{2+1}{2+2} = 0.75$. Therefore, the trust scores for the five honest users are 0.86, 0.75, 0.75, 0.75, and 0.75 respectively. Thus, the summation of the trust scores of the honest users is $0.86 + 0.75 \times 4 = 3.86$. Since the initial trust score for each malicious user is 0.5, the minimum number of malicious users needed for launching a successful attack will be $\lfloor 3.86/0.5 \rfloor + 1 = 8$. Thus, the attack cost is <8,8>, which means that at least 8 malicious users are needed to provide 8 dishonest ratings to $I_1$. This example also shows that compared with majority-based social rating systems, trust-enhanced social rating systems may increase the attack cost significantly.

### *3.4 Disguise Attack*

The disguise attack is a strategically more complex attack compared to the direct attack where the attacker only rates on items in the target set. The disguise attack aims at exploit-

ing trust-enhanced social rating systems and by design it has malicious users first behave honestly and gain high trust scores before they behave dishonestly. Concretely, an attacker may first register a set of malicious users and have them provide honest ratings to a chosen subset of non-target items, called the honest set $\mathrm{I^H}$, in order to hide their malicious intent by gaining higher trust and confusing the trust-enhanced social rating system. Then, the malicious users will provide dishonest ratings on all items in the attack target set. This type of attack can be represented as $\{\mathrm{I^T} \neq \emptyset, \mathrm{I^H} \neq \emptyset, \mathrm{I^D} = \emptyset, \mathrm{I^B} \neq \emptyset\}$ and both target set and honest set are non-empty. The choice of $\mathrm{I^H}$ can directly impact on the cost and effectiveness of a disguise attack. Figure 3(c) illustrates the disguise attack. When an attacker also rates some items honestly as honest users do, it is harder for the system to identify their malicious intent. Compared to the direct attack in Figure 3(b), the disguise attack leads to more malicious power with less attack cost.

**Self-promoting attack** [Dellarocas 2003; Hoffman et al 2007; Sun el al 2006] is an example disguise attack. An attacker first registered a set of malicious users and put those items that are in competition in its target set. Then, it chooses several non-target items and has the newly registered malicious users to rate them honestly with "+1" such that the trust scores of the malicious users are increased after each honest rating. When the attacker has accumulated enough trust scores for the malicious users, these malicious users start launching the attack to the target by providing dishonest ratings to items in the target set.

***Attack effect analysis***

Consider a beta-function based social rating system and we assume that the attacker chooses those items that have a majority of "+" ratings to be included in its honest set and the size of honest set is $|I^H|$. Also we assume that the attacker requires each malicious user registered in the system to give one honest rating to each of the items in the honest set $I^H$. Based on the majority rule and the beta-function, the system will mistakenly consider all malicious users honest based on the observation that all of their ratings on items in $I^H$ is consistent to the majority of ratings received and zero ratings so far are inconsistent with the majority, thus malicious users are judged by the system as honest with the trust score of $\frac{|I^H|+1}{|I^H|+2}$ according to Equation (3) with $\mathrm{H_i} = |I^H|$ and $\mathrm{D_i} = 0$. Let $N_M$ denote the number of malicious users needed. By Equation (7), we have $N_M \times \frac{|I^H|+1}{|I^H|+2} > \sum_{U_i \in \mathrm{S_j^{+1}}} T_i$, where $N_M$ denotes the number of malicious users needed. Therefore, the attacker can estimate that the minimum number of malicious users required to launch a successful disguise attack should satisfy the condition of $N_M > \sum_{U_i \in \mathrm{S_j^{+1}}} T_i \,/\, \frac{|I^H|+1}{|I^H|+2}$. In self-promotion based disguise attack, each malicious user provides one dishonest rating to the items in the attack target set. Consider the case of only one item in the target set for presentation simplicity. The attack cost is calculated by $< \left\lfloor \sum_{U_i \in \mathrm{S_j^{+1}}} T_i \,/\, \frac{|I^H|+1}{|I^H|+2} \right\rfloor + 1,\ (\left\lfloor \sum_{U_i \in \mathrm{S_j^{+1}}} T_i \,/\, \frac{|I^H|+1}{|I^H|+2} \right\rfloor + 1) \times (|I^H| + 1) >$, since each malicious user will provide $|I^H|$ honest ratings and one dishonest rating. Let $T_H$ represent the average trust score of an honest user. We can revise this attack cost function to $< \left\lfloor T_H \times |\mathrm{S_T^{+1}}| \times \frac{|I^H|+2}{|I^H|+1} \right\rfloor + 1,\ (\left\lfloor T_H \times |\mathrm{S_T^{+1}}| \times \frac{|I^H|+2}{|I^H|+1} \right\rfloor + 1) \times (|I^H| + 1) >$. Clearly, in

contrast to the bad-mouthing attack, the cost of a self-promotion attack depends on not only the size of the target set but also the size of honest set.

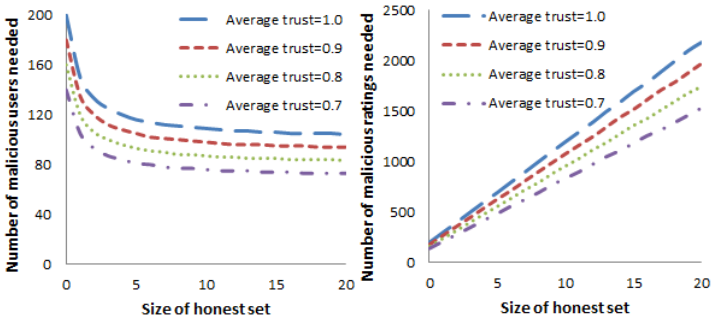



**Figure 4 Number of malicious users and malicious ratings needed for disguise attack**

Figure 4 shows the number of malicious users ($N_M$) and the number of malicious ratings ($K_M$) needed for a successful self-promotion attack respectively. In this experiment, we measure the results of $N_M$ and $K_M$ by varying the size of the honest set $I^H$ from zero to 20 (x-axis). The average trust score of honest users ($T_H$) is set with $1, 0.9, 0.8$, and $0.7$ and the total number of honest raters for the target item prior to attack is 100.

In the left figure, we measure the number of malicious users needed to launch a successful attack to the target. Each malicious user provides one dishonest rating to the target item in addition to providing an honest rating to each item in the honest set $I^H$. The four different curves measure the value of $N_M$ needed when the average trust scores of the honest users are different. Two important observations are made. First, when malicious users rate more items honestly, the number of malicious users needed for launching a successful disguise attack will be reduced. This is because the trust scores of malicious users are increased as they rate more items honestly and as a result their attack power is increased. However, this boost of power is limited by the upper bound of the maximum trust score for a user set in the trust enhanced rating systems. As a result, the number of malicious users needed ($N_M$) will not change after the honest ratings of a user reach a certain threshold. Second, when the average trust score of the honest users is increased, the number of malicious users needed for a successful disguise attack will also be increased. This is because the high trust values of honest users imply higher cost for the attacker in terms of the number of malicious users and the number of malicious ratings.

In the right figure, we vary the size of honest set (x-axis) and measure the number of malicious ratings needed ($K_M$) to successfully attack a target item (y-axis). Similarly, the four different curves refer to the four different average trust scores of honest users. We can conclude that the number of malicious ratings needed for launching a successful attack will increase linearly when we increase the size of honest set.

An important observation from these two sets of experiments (Figure 4) is that the disguise attack (self-promotion) reduces the attack cost in comparison to the direct attack (bad mouth) by using higher number of malicious ratings and relatively smaller number of malicious users, i.e., $K_M >> N_M$. Note that the honest ratings on items in $I^H$ provided by ma-

licious users are considered as malicious ratings. Assuming that an attacker wants to attack a target item that has 100 honest raters with average trust score of 0.8, when the size of the honest set is 0, it is a direct attack, and needs 161 malicious users to provide 161 malicious ratings (Figure 4 right plot) to achieve the attack goal. When the size of the honest set is 5, it is a disguise attack and each malicious user will provide honest ratings to 5 items. Thus, this attack only needs a total of 94 malicious users (Figure 4 left plot) to provide 564 malicious ratings (Figure 4 right plot) to achieve the attack goal. Furthermore, when the size of the honest set is bigger, say 20, the disguise attack will have each malicious user provide honest ratings to 20 items, and this effectively reduces $n^M$ to 84 malicious users (Figure 4 left plot), though the total of 1,764 malicious ratings (Figure 4 right plot) is needed to achieve the attack goal. The change of the attack cost from <161,161> to <94,564> and <84,1764> shows that the disguise attack can effectively reduce the number of malicious users needed for launching a successful attack at the cost of increasing the number of malicious ratings needed. Although the disguise attack is considered more powerful and harder to detect, the effect of the disguise attack, however, is still limited due to two factors: (1) The trust value of a user has an upper bound in the trust enhanced social rating systems. (2) The reduction of the number of malicious users ($N_M$) is logarithmic to the number of malicious ratings needed ($K_M$). Recall the example in Figure 4, the disguise attack will reduce the number of malicious users from 161 to 94 with an increase of 564-161=403 ratings; while the reduction of malicious users from 94 to 84 needs an increase of 1764-564=1200 ratings.

***Attack cost analysis***

The cost of a disguise attack is determined by the minimum number of malicious users and the minimum number of dishonest ratings needed. Given the attack target, one will need to examine different choices of selecting the honest set of items $I^H$ from the remaining set of non-target items in order to find the attack strategy with the minimum cost. We first illustrate the search space for attack strategy with minimum attack cost using the example in Figure 1. Assume that the self-promotion attack is targeted at item$I_1$.In order to find the concrete strategy for launching this attack at the lowest cost, we need to enumerate the set of possible ways to launch such a self-promotion attack in terms of the number of malicious users and the number of malicious ratings. The process to reach this decision is as follows: Given that the attack target is item $I_1$, which has five honest raters. Her trust score is calculated as $\frac{5+1}{5+2} = 0.86$; each of the other four users has provided two honest ratings and their trust score is calculated as $\frac{2+1}{2+2} = 0.75$. Therefore, the trust scores for the five honest users are 0.86, 0.75, 0.75, 0.75, and 0.75 respectively. Thus, the summation of the trust scores of the honest users is $0.86 + 0.75\times4 = 3.86$. Thus, the attackers should use an attack strategy that can gain a sum of trust scores larger than 3.86. If the direct attack is used, then the initial trust score for each malicious user is 0.5, and the minimum number of malicious users needed for launching a successful attack will be $\lfloor 3.86/0.5 \rfloor + 1 = 8$. The attack cost is <8,8>, namely at least 8 malicious users are needed to provide 8 dishonest ratings to $I_1$. Under the disguise attack, the number of malicious users should be smaller than 8. Consider the attack strategy of using 5 malicious users and the honest set contains one item, say $I_2$: First, two malicious users provide two honest ratings on $I_2$ to increase

their trust scores from 0.5 to $\frac{2+1}{2+2} = 0.75$ according to Equation (3). Similarly, three malicious users provide three honest ratings on $I_2$ to increase their trust scores from 0.5 to $\frac{3+1}{3+2} = 0.8$. Then, the sum of the trust scores of the five malicious users is computed as $0.75\times2 + 0.8\times3 = 3.9$, which is larger than the sum of the five honest raters of $I_1$, which is 3.86. Therefore, in a majority-based social rating system, the five malicious users will get a summation of trust scores higher than the summation of trust scores of the honest users. Thus the attacker only needs to provide five dishonest ratings to the target item to achieve his attack goal. This attack cost is $<5, 2\times2 + 3\times3 + 5\times1 = 18>$. By examining different attack strategies, we find that the cost of the most efficient disguise attack is <5, 18>, namely the minimum number of malicious users needed is 5 and the minimum number of malicious ratings needed is 18 when there are five malicious users. We omit the complete algorithm in this paper due to space limit. Readers may refer to [Feng2011] for more detail.

### *3.5 Misguidance Attacks*

The misguidance attack is the most sophisticated and most powerful attack in the social rating systems. On one hand this type of attacks is much harder to detect, and on the other hand, this type of attacks can succeed with much less resource compared to the direct attack and the disguise attack. Concretely, in addition to create malicious users to provide dishonest ratings on items in the attack target set, the misguidance attack strategically selects a subset of non-target items as the dishonest set and provide dishonest ratings on these items to boost the attack efficiency. Thus, this third type of attacks can be represented by $\{I^T \neq \emptyset, I^H \neq \emptyset, I^D = \emptyset, I^B \neq \emptyset\}$. We below discuss how to select items in both $I^H$ and $I^D$ for a given attack target set $I^T$. This new strategy aims at strategically making the system misjudge the honest rating behavior to be dishonest and the dishonest rating behavior to be honest. Consequently it will reduce the trust of honest users and increase the trust of dishonest users, effectively exploiting the vulnerabilities of many social rating systems under the context of majority voting principle. Figure 3(d) illustrates the effect of adding this new attack strategy of providing dishonest ratings on a selected subset of non-target items. Compare with Figure 3(c) we see more power reduction from honest raters and consequently an increase of malicious power through lower attack cost.

Misguidance attacks can appear in different forms. The **reputation-trap attack** reported first in [Feng et al. 2010] is a known form of the misguidance attack. We below use the reputation-trap attack to illustrate the intrinsic properties of the attack strategy, such as how an attacker selects the subset of non-target items to form the dishonest set strategically and why this attack can further increase the trust of malicious users and reduce the trust of honest users at the same time.

***Reputation-trap attack***

First of all, the goal that the attacker uses the "dishonest set" is to misguide the judgment of the system on which users are honest raters. If the attacker can manipulate the system to make the honest ratings of the users to be judged as dishonest and vice versa using the majority principle, it will be able to reduce the trust of honest users and increase the trust of

dishonest users effectively. In summary, the reputation-trap attack includes the selection of dishonest set and the selection of honest set. It will provide honest ratings to the "honest set" to gain the effect of a self-promotion attack and at the same time provide dishonest ratings to the dishonest set to set a trap that can increase the trust values of malicious users and reduce the trust of honest users.

An important challenge for the attacker is to define a strategy that can select the minimum number of malicious users and the smallest subset of non-target items to form the "dishonest set" such that the malicious users have enough power through dishonest ratings to turn around the judgment of the system on the social recommendation of each item in the "dishonest set". A straightforward approach is a three-phase iterative process. (1) We examine all items in the candidate item set and compute the trust scores for each. (2) We put the item with lowest aggregate trust score from all raters into the dishonest set of items as a trap. (3) We need to determine the minimum number of malicious users with respect to this chosen trap. This can be done by examining different number of malicious users needed in order to turn the item in the trap to honest raters on the trap item to be dishonest. This process repeats with the remaining set of items as the candidate set until the attacker gains the sufficient power to launch a successful attack on the target item.

We illustrate this algorithm using the user-item graph in Figure 1. Using the beta-function based social rating system, the trust scores of the five honest users in Figure 1 ($U_1$~$U_5$) can be calculated in the same way as done in the attack analysis of Section 3.4: $T_1 = 0.86$; $T_2 = T_3 = T_4 = T_5 = 0.75$. Therefore, we can calculate the summation of the trust scores of the honest raters for each of the five items in Figure 1: $\sum_{S_1^{+1}} T_i = 0.86 + 0.75\times4 = 3.86$; $\sum_{S_2^{+1}} T_i = 2.36$; $\sum_{S_3^{+1}} T_i = 1.61$; $\sum_{S_4^{+1}} T_i = 1.61$; $\sum_{S_5^{+1}} T_i = 0.86$. These five summation values represent the minimum summations of the trust scores that the attacker needs to achieve in order to turn the social rating system upside down and turn the honest ratings into dishonest ones. Now we need to determine the number of malicious users needed for launching a successful attack.

By examining the item $I_5$ with the lowest trust aggregation score, we observe that with two malicious users, each with the initial trust scores of 0.5, their summation of trust scores will be 1.0 higher than $\sum_{S_5^{+1}} T_i = 0.86$. Thus, with two malicious users, the attacker can make the system mistakenly judge $I_5$ as a bad item by using two dishonest ratings on $I_5$. Now we examine the remaining four items and with $I_4$ as the next item with lowest trust aggregate score. We need three malicious users to bring down $I_4$. Assuming that after examining all possibilities, we find that three malicious users is the minimum number. Thus if the attack goal is not $I_5$, the attacker can always chooses $I_5$ to be an item in its dishonest set. By first attacking $I_5$, we see four interesting consequences: (i) The dishonest ratings provided by the three malicious users on $I_5$ will be judged as honest by the system since they match the social recommendation of $I_5$, because the aggregated trust of these three malicious users (1.5) is higher than the aggregated trust of honest users on $I_5$ (0.86), which is the trust of $U_1$ since only $U_1$ has rated "+" on $I_5$. (ii) The honest rating provided by $U_1$ will now be judged as dishonest, since it does not match the social recommendation of $I_5$. (iii) The trust score of $U_1$ will be reduced from 0.86 to $T_1 = \frac{4+1}{4+1+2} = 0.71$ since the system now considers $U_1$ have provided one dishonest rating according to the majority

rule. (iv) By first rating $I_5$ dishonestly and the above three consequences, the summations of the trust scores of honest raters on $I_1$, $I_1$, $I_3$, and $I_4$ will be reduced to $\sum_{S_1^{+1}} T_i = 3.71$; $\sum_{S_2^{+1}} T_i = 2.21$; $\sum_{S_3^{+1}} T_i = 1.46$; $\sum_{S_4^{+1}} T_i = 1.46$ respectively. This implies that the attacker will need less power to attack the other four items after putting $I_5$ into a reputation trap. If the attacker continually puts some other items into the traps, he can successfully attack the hardest item $I_1$ with only three malicious users.

In formation of the above attack strategy, we need to take into account the following two factors in order to select the suitable items to be the low-cost traps.

- **The gain of the trap**: when an item is selected to be put into the "dishonest set", the attacker needs to provide the overwhelming dishonest ratings to this item relative to the current honest ratings received on this item, such that the social recommendation of this item will be turned upside down ("+" becomes "-" and vice versa). Therefore, all the honest users who rate on this item will be misjudged as conducting one dishonest rating, while all malicious users will be treated as honest simply because the malicious users form the majority raters for the trapped item and their ratings are consistent with the trust-enhanced recommendation score, even though they rate this item dishonestly. If a user, who rates on this trapped item, also rates on items in the attack target, this reputation trap will reduce the power of the honest ratings received on the target, since this user's trust will be reduced. Thus, the gain of putting an item into a trap can be inferred from the number of users who rate on both this item and the items in the target set.
- **The cost of the trap**: when we want to turn the social recommendation of an item upside down, we should provide the overwhelming number of malicious ratings in order to succeed the attack. We call the number of ratings needed the cost of this target item.

When we select the items to be included in the "dishonest set", we need to consider the trade-off between the gain and the cost of putting the item as a reputation trap. One can use the ratio of these two factors as the heuristic value to select the items. We omit the complete algorithm for constructing reputation-trap attack and refer readers to [Feng2011] for more detail.

***Attack effect analysis***

In a misguidance attack represented in the form of the reputation-trap attack, for a given minimum number of malicious users, we need to design $I^H$ and $I^D$ such that the number of malicious ratings needed is the minimum. Consider each malicious user, in addition to dishonest ratings on items in the target set, it will also provide $|I^H|$ honest ratings to a subset of non-target items and $|I^D|$ dishonest ratings to another selected subset of non-target items. Since the goal of the attacker is to exploit the vulnerabilities of the majority rating principle to make all ratings on items in $I^H$ and $I^D$ to be identified as honest by the system before it provides dishonest ratings to items in the target set. Therefore, the trust score of a malicious user is calculated by $\frac{|I^H|+|I^D|+1}{|I^H|+|I^D|+2}$ according to Equation (3). Finally, the summation of the trust scores of malicious users will be computed by $N_M \times \frac{|I^H|+|I^D|+1}{|I^H|+|I^D|+2}$.

Meanwhile, the summation of the trust scores of the honest users of the target (assuming one item in the target set) can be calculated as $T_H \cdot |S_T^{+1}| - |I^D| \cdot A \cdot P \cdot T_A$. In this

function, $T_H \cdot |S_T^{+1}|$ gives the original summation of trust scores of the honest users, where $T_H$ denotes the average trust score of an honest user and $|S_T^{+1}|$ is the number of honest users. $|I^D| \cdot A \cdot P \cdot T_A$ gives the reduced trust score of the honest users, where $|I^D|$ is the number of items in the dishonest set, $A$ is the average number of raters for the items in the dishonest set, $P$ is the probability that a user is both the rater of an item in $I^D$ and the rater of the target item, and $T_A$ denotes the average of reduced trust score on a user when he is affected by a trap. For example, the trust of $U_1$ is reduced by 0.86-0.71=0.15 in our running example and therefore $T_A = 0.15$ for $U_1$. Thus, the attack goal can be transformed into the following condition by Equation (7):

$$N_M \times \frac{|I^H| + |I^D| + 1}{|I^H| + |I^D| + 2} > T_H \cdot |S_T^{+1}| - |I^D| \cdot A \cdot P \cdot T_A$$

We can infer the number of malicious users needed using the following formula:

$$\left\lfloor (T_H \cdot |S_T^{+1}| - |I^D| \cdot A \cdot P \cdot T_A) \cdot \frac{|I^H| + |I^D| + 2}{|I^H| + |I^D| + 1} \right\rfloor + 1$$

This function shows that a number of factors may lead to a more powerful reputation-trap attack in terms of cost and effectiveness, including a smaller average trust scores of honest users, a smaller number of honest raters on the target, a larger number of dishonest set, a larger number of the average raters on the trapped items, a larger number of the connection probability, and a larger degree of the affected trust scores.

With this function, we can further infer the number of malicious ratings needed by multiplying the number of ratings provided by each malicious user, namely $|I^H| + |I^D| + 1$. This is because every malicious user will rate on all items in the honest set, the dishonest set, and the target set. Thus, the function below computes the minimum $K_M$ needed:

$$\left( \left\lfloor (T_H \cdot |S_T^{+1}| - |I^D| \cdot A \cdot P \cdot T_A) \cdot \frac{|I^H| + |I^D| + 2}{|I^H| + |I^D| + 1} \right\rfloor + 1 \right) \cdot (|I^H| + |I^D| + 1)$$

Figure 5 shows a comparison of reputation-trap attack with self-promotion attack. It measures the number of malicious users needed (y-axis) by varying the size of honest set for self-promotion attack or the size of dishonest set for reputation-trap attack. In this experiment, we set $|S_T^{+1}| = 100, A = 50, P = 0.2$, and $T_A = 0.2$. As the size of the honest set increases, the reduction on the number of malicious users is significantly limited for self-promotion attack, since it can only increase the trust scores of malicious users to a certain degree as we have discussed before. In contrast, for reputation-trap attack, the number of malicious users needed continues to drop as the size of the dishonest set increases, since the utilization of dishonest set can also reduce the trust scores of honest raters of the target items.

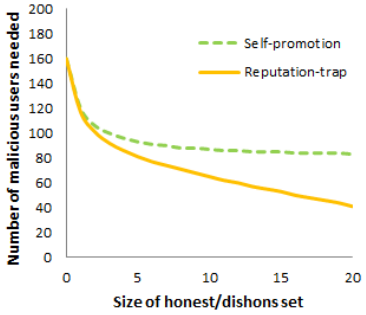



**Figure 5** Number of malicious users needed for disguise attack and misguidance attack

Consider our running example in Figure 1 again. The cost for a disguise attack is <5,18> (recall Section 3.4). In comparison, the cost for a reputation-trap attack on target item $I_1$ is <3, 13>, which means that the attacker only needs 3 malicious users and 13 malicious ratings in order to launch a successful attack to the item $I_1$. Below we illustrate how we reach the minimum number of malicious users and dishonest ratings.

To attack target item $I_1$ strategically with three malicious users at low cost, the attacker should first put $I_5$ into the dishonest set $I^D$, then $I_4, I_3, I_2$, before attacking $I_1$:

- Two malicious users rates on $I_5$dishonestly, since their summation of trust scores is 1.0 (their initial trust is 0.5) which is larger than the summation of the honest users 0.86. The system now judges the social recommendation of $I_5$ as bad ("-"). As a consequence, the trust of the two malicious users will be increased from 0.5 to 0.67, and the trust of the honest user $U_1$ will be reduced from 0.86 to 0.71.
- By using three malicious users now to rate $I_4$ dishonestly, the trust scores of malicious users will continue to increase while the trust of honest users will continue to decrease.
- Two malicious users can use two dishonest ratings (one each) to turn $I_3$ into a reputation trap, and the trust scores of the malicious users are increased and the trust scores of the honest users are reduced.
- Three malicious users can now turn $I_2$ into a reputation trap, and change the trust scores of malicious users and honest users.
- Finally, the summation of the trust scores of 3 malicious users is increased to 2.41 while the trust summation of the honest users who rate on $I_1$ is reduced to 2.29. So the attack achieves its goal with only three malicious users. In summary, the three malicious users have provided a total of 2+3+2+3+3=13 malicious ratings to succeed the attack. The attack cost is <3, 13>.

In our experimental evaluation [FENG2011], compared to bad-mouthing attack, the reputation-trap attack can reduce 63% of the malicious users needed. Compared to self-promotion attack, the reputation-trap attack can reduce 40% of the malicious users needed and 28% of the malicious ratings needed.

## 4. CONTEXT-AWARE COUNTERMEASURES AND DEFENSE METHODS

Based on our attack effect and cost analysis on the three representative types of attacks in existing social rating systems, we make two important observations. First, attacks in social rating systems utilize some common exploits of vulnerabilities in the battle between honest and dishonest ratings. Second, the definition of what considered honest and what considered dishonest is a sweet context sensitive spot that is both the corner stone for social rating systems without central authority and the detrimental weak-point that is vulnerable to strategic manipulation by attackers. Furthermore, items with controversial ratings and items that have contradict ratings between non-expert users and expert users can become the sweet spot for malicious exploits. In this section, we describe some countermeasures and context-aware safeguards that can be used as effective defense methods. We motivate our discussion primarily based on them is guidance attack, since this is the most sophisticated and yet most powerful attack among all the three types of attacks we have discussed in this paper.

### *4.1 Defense by Information Hiding*

Consider the three types of attacks discussed in the previous section, one of the common techniques that attackers use is the knowledge and information about the current state of the social rating systems. For instance, to launch a successful reputation-trap attack, the attacker needs to know which items are most vulnerable and can be manipulated with less resource. Recall our running example in Figure 1, by knowing item $I_5$ has less honest raters in comparison to other items, even though the attacker wants to bring down the social recommendation of $I_1$ as its ultimate attack target, the attacker may strategically apply the first reputation trap on $I_5$, then on $I_4$, $I_3$, and $I_2$, which lead the drop of trust values of large number of honest raters of these items. Consequently, the attacker has made the target item $I_1$ become easier to attack at lower cost. If the knowledge of who are the raters of an item is hidden from the attacker, such attack would have been much harder to exercise successfully at low cost. Clearly, the more knowledge and information context an attacker has about the social rating system, the more vulnerable the system becomes, as the attackers can exploit the system starting from its weakest spots and launch a sequence of detrimental attacks to items or users selectively at affordable cost.

A challenge in using information hiding as a context-aware countermeasure to the self-promotion and reputation-trap attacks is to identify which information should be hidden to best protect the social rating system in terms of both utility preserving and attack resilience. By utility preserving, we mean that the information we choose to hide should be important to the attackers but relatively less important to the normal users. By attack resilience, we mean that the information hiding strategy should lead to increased attack cost and make it harder for the attackers to design powerful attacks. For example, user-item relationship in terms of rating should be protected as private information. The items with controversial ratings should be protected in terms of number of raters an degree of the rating contrast. Also items that have contradict ratings between non-expert users and expert users should be carefully monitored for malicious exploits. Due to space constraint, we will focus on strategies for hiding information in the context of user-item relationship hiding technique.

***Hiding user-item relationship***

The user-item graph as shown in Figure 1 exposes three types of relationships: (i) the user-item relationship through user rating an item, (ii) the user-user relationship through rating a common item, and (iii) the item-item relationship through a common user who rated both items. If an attacker knows the user-item graph, such as the one in Figure 1, they can easily find the items that may greatly affect the honest raters of the target. Therefore, an intuitive defense method is to hide the user-item graph context from all users of the social rating system and thus the attackers. Regarding the utility of this user-item graph, it is obvious that hiding this information seldom has any impact on the fundamental function of the social rating systems, since normal users are more concerned about the overall social recommendation of the items of their interests, rather than who are the raters of those items. Alternatively, one can also use a less intrusive way of hiding information, which is to selectively hide some of the three types of relationships. For instance, one can search for the number of users rated an item or the number of items rated by a user, but not the identity or pseudo-identity of the users and who rated which items.

***Defense effect analysis***

The effectiveness of a defense method is highly dependent on the counter-measure we use, including both qualitative and quantitative measures of the amount of information hiding provided. The direct impact of the item-user relationship hiding defense is that the attacker cannot easily infer with certainty the relationship between two users in terms of the common items they have rated. Thus, even the information that the item $I_j$ has the lowest trust aggregation score is public and the attack knows that $U_i$ is among the users who have rated on the target item through some external channel, it is hard for attacker to know with certainty of whether placing a trap on item$I_j$will affect the trust of user $U_i$. This is because the attacker cannot infer whether $U_i$ is also a rater of item $I_j$. Recall the reputation-trap attack, it is clear that the attacker will not be able to infer the gain of putting an item in the reputation trap directly if the user-item relationship is hidden. Therefore, the attacker can only heuristically infer this information, for example, by using the number of raters on an item as the heuristic value, since an item with higher number of raters may have higher chance to be the raters of the items in the target set or indirectly affect the raters of the target. Thus, the relationship-hiding countermeasure will reduce the probability that an item in the dishonest set will affect an honest rater of the target. We use the parameter P to capture such a probability. Figure 6 shows our experimental measurements of both the number of malicious users and the number of malicious ratings needed when we vary the values of parameter P, the probability that a rater of an specific item in the dishonest set is also the rater of an item in the target set, from 0.2 to 0.05. We keep all the other parameters the same as in Figure 5. From Figure 6 we observe that both the number of malicious users ($N_M$) and the number of malicious ratings ($K_M$) needed will increase as we increase the probability (P) that a user who rates a specific item in the dishonest set has also rated an item in the target set. This analysis shows that the relationship hiding defense will increase the attack cost, making the attack much more difficult to succeed.

As a result, this information hiding defense will in turn increase both the number of malicious users needed and the number of malicious ratings needed for succeeding this attack.

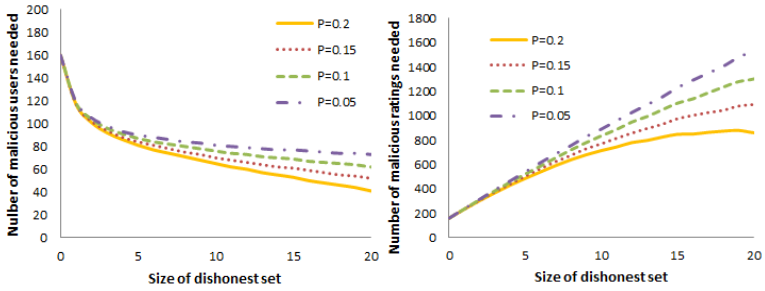



**Figure 6 The minimum attack cost $(N_M, K_M)$ with different P (probability)**

Finally, it is worth to note that the user-item relationship hiding is especially useful for social rating services in e-Commerce domain, such as Amazon and Apple App Store, since users in this domain care little about who are the reviewers and also users cannot rate on an item they have not purchased. However, such user-item relationship hiding may lose some utility in a social network based rating service, since the users in a social network tend to trust more about the ratings of their friends than other unknown raters. In summary, the effectiveness of information hiding defense in terms of attack resilience is highly dependent on the utility of social ratings in a given domain.

### *4.2 Defense by Utilizing Confidence Weight*

We have shown that trust-enhanced social rating systems are vulnerable to both disguise attacks and misguidance attacks, though providing resilience to the direct attack, because the trust of a user is evaluated based on whether her ratings on an item can match the overall social recommendation of the item computed based on the majority of the users and their ratings received on the item. One of the most vulnerable spots in such social rating systems is the unpopular items that receive a relatively small number of user ratings. Attackers can start attacking unpopular items to drop the trust score of the corresponding users who rated on unpopular items. If some of these users also rated the items in the attack target set, by dropping the trust values of those users who rated a target item, the attacker can succeed in bringing down the social recommendation of the target item.

One way to counter such an attack is to minimize the exposure of unpopular items. When we infer the trust of a user, we should give more confidence on the ratings which are given to popular items while give less confidence on the ratings which are given to unpopular items, Such confidence weighting scheme increases the robustness of the system against malicious manipulation by minimizing the impact of making several unpopular items into traps on the trust scores of the honest users.

The design of this confidence-weight based defense is based on the observation that the more raters an item has, the harder for the attacker to manipulate the social recommendation of this item, since it costs the attackers larger amount of resource. Therefore, we want to link the confidence on the social recommendation of an item to the total number of raters on this item. However, the function transforming the number of raters to the confidence of its social recommendation should meet the following properties. First, an item

with a larger number of honest raters should have a higher confidence score. Second, the confidence on a popular item should not overwhelm the other evidences. Third, the increasing speed of the confidence should be reduced as the number of raters increases.

With these requirements in mind, we introduce a log function here to calculate the confidence on the social recommendation of $I_j$:

$$C_j = \log_2 (|S_j^{+1}| + |S_j^{-1}|) \qquad (8)$$

Based on (8), we can further revise Equation (3) by using confidence scores as weight values when inferring the trust of a user. Concretely, the set of honest behavior (ratings) $H_i$ and the set of dishonest behavior (ratings) $D_i$ can be used by combing the confidence on the honest/dishonest rating behavior.

$$T_i = \frac{\sum_{R_{ij} \in H_i} C_j + 1}{\sum_{R_{ij} \in H_i} C_j + \sum_{R_{ij} \in D_i} C_j + 2} \qquad (9)$$

***Defense effect analysis***

In order to understand the effect of this confidence-weight based defense, we first illustrate the effect by a concrete example. Assume that $U_1$ has rated 8 items, the number of raters for each of the eight items is 1024, 1024, 1024, 1024, 4, 4, 4, 4. There are 4 malicious users, and thus they can only set reputation traps on the four unpopular items with 4 raters. Without using confidence-based safeguard, the trust score of $U_1$ is $\frac{8+1}{8+2} = 0.9$ before the attack, and the trust score is reduced to $\frac{4+1}{4+4+2} = 0.5$ after the attack. By incorporating the confidence-weight based defense into the social rating system, the trust score of $U_1$ is $\frac{4\times \log 1024+4\times \log 4+1}{4\times \log 1024+4\times \log 4+2} = 0.98$ before the attack, and the trust score is only reduced to $\frac{4\times \log 1024+1}{4\times \log 1024+4\times \log 8+2} = 0.76$ after the attack. Therefore, the amount of reduction in trust score is much lower when we turn on the confidence-weight based safeguard (0.98-0.76=0.22), compared to the social rating system without confidence-weight based defense (0.9-0.5=0.4). This means that this confidence weight based defense can reduce $T_A$, which is the amount of reduction in trust due to an attack. A comprehensive evaluation of this defense can be found in [Feng et al. 2010].

In summary, the confidence-weight based defense makes it much harder for the attacker to launch the reputation trap attack at low cost by exploiting the unpopular items.

### *4.3 Defense by Incorporating Time Windows*

We observe that the time dimension of the social rating system, such as when a user rates an item, can provide rich information to evaluate the utility of honest ratings, detect dishonest ratings and increase the attack resilience of the system against malicious users. In this section, we briefly describe some of our initial development on incorporating the time dimension of the user-item relationship in social rating services to build effective attack resilient defense mechanisms.

There are many ways to utilize the context of temporal information in developing countermeasures and defense methods [Yang et al. 2009]. The time-window based defense method can be seen as a representative defense mechanism. It aggregates the trust values of users by taking into account the time-window constraints of the ratings provided by the

users. Concretely, the social recommendation score of an item is produced by the system based only on the ratings valid in the given time window as well as the social recommendation scores valid in the past time-windows considered.

In designing the time-window based defense, we first divide the time dimension into time windows with a system defined time interval, such as one week. This time interval can also be determined non-uniformly based on a fixed number of ratings received on an item. Within each time window, we calculate the time-window based social recommendation score for each item by using a rating aggregation algorithm (e.g., the ones presented in Section 3). Therefore, for the item $I_j$, we can get a sequence of temporal recommendation scores based on the sequence of time windows considered, each score is inferred within the corresponding time window. Assuming that there are $l$ time windows, the sequence of $l$ temporal recommendation scores are denoted as$Q_i^{S_1}, Q_i^{S_2}, \dots, Q_i^{S_l}$. By utilizing these temporal recommendation scores of an item, we can introduce two levels of aggregations of user ratings: The first level is the aggregation of ratings for an item based on the majority principle within a single time window. The second level of aggregation is to perform another round of majority rating along time windows to infer the final social recommendation score of this item.

Formally, let $S_{j,k}^{+1}$ and $S_{j,k}^{-1}$denote the set of users who rate item$I_j$ with "+1" and "-1"within the window $k$ respectively. The temporal recommendation score of $I_j$ within the window $k$ is calculated by aggregating the ratings collected within that window.

$$R_j^{S_k} = \frac{\left|S_{j,k}^{+1}\right|}{\left|S_{j,k}^{+1}\right| + \left|S_{j,k}^{-1}\right|} \qquad (10)$$

$$Q_j^{S_k} = \begin{cases} +1, & R_j^{S_k} \geq 0.5 \\ -1, & R_j^{S_k} < 0.5 \end{cases} \qquad (11)$$

Then, we compute the numbers of windows ($W^{+1}$) with temporal recommendation scores of "+1" and the numbers of windows ($W^{-1}$) with temporal recommendation scores of"-1" respectively as shown in Equation (12) and (13). We use Equation (14) to compare between the two numbers to determine the final social recommendation score based the majority principle.

$$W^{+1} = \left|\{Q_i^{S_j} | Q_i^{S_j} = +1\}\right| \qquad (12)$$

$$W^{-1} = \left|\{Q_i^{S_j} | Q_i^{S_j} = -1\}\right| \qquad (13)$$

$$Q_j^{S} = \begin{cases} +1, & \text{if} \quad W^{+1} \geq W^{-1} \\ -1, & \text{if} \quad W^{+1} < W^{-1} \end{cases} \qquad (14)$$

Equation (14) amounts to say that the social recommendation score of an item represents the majority opinions within the given time windows.

Figure 7 gives an example of three time windows of an item. In the first round, we compute the social recommendation scores of the item in each of the three time windows. The first two windows both have the good scores since there are only good ratings in these two windows. In the third window, we see two good ratings and seven bad ratings, and thus the social recommendation score for this time window is negative. In the second round of social recommendation aggregation, we summarize the results from different

windows. Since two windows produce positive rating on the item and one window presents the negative score, the final recommendation score will be good based on the majority of votes that this item has received over the sequence of time windows. Incorporating time dimension not only helps us detect dishonest ratings more easily and consistently but also strengthens the attack resilience of the social rating systems.

Finally we should mention that the time window is useful when an item's quality is relative stable over time. Otherwise we will need to combine confidence weight with time windows to balance between present and historical time windows.


Figure 7 Example of time windows of an item

***Defense effect analysis***

The main goal of our time-window based defense method is to increase the complexity of applying the majority rating principle and to make it harder for an attacker to exploit the context of majority rating as displayed in reputation-trap attack and self-promotion attack (recall Section 3.3 and Section 3.4). With this defense method, the attacker has to consider not only which items to attack and which items to apply reputation traps to, but also when and which time windows the attack should be exercised. The attacker has to win at least half of the time in term of per-window based aggregation in order to change the final social recommendation score of an item. If we use the fading memory model to weigh the sequence of temporal aggregations based on the sequence of time windows, the attacker will have to maintain its wining position in most recent consecutive time windows, which is much harder to achieve. Thus, the time-window based defense makes the attacker work significantly harder if he wants to make the malicious ratings become the majority in the majority of the time windows considered. Other factors that affect the effectiveness of time-window based defense methods include the length of the time needed for an attacker to attack a target, the impact of the time interval choices, the window size choices, the sequence of windows, and the fading memory effect on the attack success rate and defense effectiveness. We conjecture that the time-window based defense mechanisms have introduced a new battlefield to fight against attacks.

### *4.4 Discussion*

We have described three context-aware defense methods. We summarize them in Table 1 for reference convenience. These three types of defense methods utilize the context of user-item graph information, the confidence weight of raters, and time dimension of user ratings respectively. In comparison, the hiding of user-item graph information based defense is the easiest in terms of implementation and deployment, though it may, to some extent, affect some of the user experiences. Although the user-item information hiding, confidence weight, and time windows are effective countermeasures against attacks in social

rating systems. Several directions of efforts are required to deploy these context-aware safe guards in real-world social rating systems.

**Table 1 Summary of context-aware defenses**

| Type | Assumption | Approach | Effect |
|---|---|---|---|
| Information based | The attackers need to know some specific information to design a powerful attack. | Hide the information important to attackers but relatively less important to users, such as user-item relationships. | Reduce P |
| Confidencebased | Some evidences are easier to be manipulated while others are harder. | Evaluate the confidence on each evidence and use it as weight when inferring the trust of a user | Reduce $T_A$ |
| Time based | The rating time can provide rich information to detect and defense against attacks. | The social recommendation of an item depends on two levels of rating aggregations: per-time-window based and a sequence of time windows based. | The attackers need to consider the rating time |

## 5. RELATED WORK AND DISCUSSION

In this section, we give a brief review of the related work from the perspectives of social network, personalized trust, system design and expert opinions and provide a brief discussion on those related work that are directly related to the social trust countermeasures presented in this paper.

**Attacks in Social Networks**. In the context of social networks, many researchers have proposed to use the social rating information to strengthen the defense for Sybil attacks, represented by SybilGuard [Yu et al 2006]. The SybilGuard protocol is based on the social network among user identities, where an edge between two identities indicates a human-established trust relationship. They assume that malicious users can create many identities but few trust relationships. Therefore, the connection among the Sybil nodes and honest nodes will be very small. Based on this assumption, the SybilGuard protocol can bound the number of identities a malicious user can create. The time window based defense and the confidence weight based defense to some extend compliment the SybilGuard protocol such that the system can limit the resources that the attacker can use and thus without obtaining sufficient connections to honest raters, making the disguise attack (self-promotion) and the misguidance attack (reputation trap) harder to succeed.

**Personalized Trust.** Personalized trust is another way to build defense against attacks. [Caverlee et al 2010; Walter 2009; Xiong et al 2004] have proposed personalized and dynamic trust in social networks or decentralized networks. This line of work is to some extent analogous to some personalized variants of PageRank. By requiring attackers to gain the trust of different individual users, it greatly increases the cost of attacks. However, the use of the transitivity of trust to compute the indirect trust between two users who are not direct neighbors of one another may open doors for the attackers to exploit the trust of

powerful users to increase their impact. Thus, incorporating our hiding of user-item relationship defense can be beneficial.

**Probabilistic Trust Inference**. Kuter and Golbeck [KUTER 2010] have proposed to use probabilistic confidence models for trust inference in web-based social networks. It first uses probabilistic sampling to separately estimate trust and confidence, and then computes an estimate of trust based on the information sources with the highest confidence estimate. [KASNECI 2011] has proposed CoBayes, which operates on a collection of statements, a set of deduction rules, a set of users, and a set of truth assessments of users about statements. They use a joint probabilistic model of truth values of statements and the expertise of users for assessing statements. A common latent knowledge space is introduced to determine the probability that a user's assessment is correct. This model could be applied to the social rating systems to model the uncertainty in the recommendations themselves and the features of the user which are indicators of their trustworthiness, such as how long they have been a member of the system, the geo lookup of their IP address and so on. This will make the attackers even harder to launch a successful disguise or misguidance attack.

**Expert v.s Non-expert ratings.** In the basic majority based social rating systems, central authority (CA) and expert ratings [AMATRIAIN 2009] are not considered. However, the use of CA and expert ratings can be one type of defense methods for social rating systems. For example, by introducing experts such as professional reviewers and pre-trusted central authority to provide ratings, the systems can choose to use their ratings as the authorized recommendation and complimented by the majority based social recommendation scores. When the expert rating is in conflict with the non-expert ratings, the social rating system can make the final call in terms of choosing the expert ratings as the ultimate authority for recommendation. In this case, the expert ratings will reduce the effectiveness of all three types of attacks discussed in this paper. Especially for those unpopular items, expert ratings will serve well and thus making the reputation trap attack ineffective. A social rating system can also choose to use both the ratings from experts and normal users to infer the quality of items, but give more weight on ratings from experts.

**System design based countermeasures**. Another effective defense to the three types of attacks discussed in this paper is to use system design based countermeasures. For instance, Amazon only allows reviews after a user made a purchase in Amazon. Some systems will introduce system-design based counter-measures. In Apple App Store, a user needs to provide a valid credit card in order to register and only the users who have bought the application can rate on it. Although this type of system design based countermeasure may reduce the number of ratings we can obtain but it will also limit the resources that the attackers can get, making it hard to launch misguidance attacks and disguise attacks. Other system-design based counter-measures include reCAPTCHA [VON 2008] with which a user needs to successfully recognize several characters before register.

## 6. CONCLUSION

We have discussed the fundamental vulnerabilities and countermeasures in context-aware social trust and recommendation services. We argue that understanding such vulnerabilities and countermeasures in a systematic manner is critical to the healthy growth of con-

text aware web services. Two representative classes of social rating systems are presented: basic majority-based rating systems and trust-enhanced social rating systems. We have provided theoretical analysis on three types of representative attacks to social rating systems, in terms of attack model, attack effect, and attack costs. The three types of attacks are direct, disguise, and misguidance attacks. The direct attack is common in the basic majority rating systems, whereas the disguise attacks and misguidance attacks are representative in trust-enhanced social rating systems. We also analyzed some critical and widely used contexts in many social rating systems, which can be abused or misused by adversaries without context-aware safe guards. Finally, we presented some context-aware countermeasures, including user-item relationship-hiding methods, utilizing confidence-weight to distinguish popular and unpopular items, and incorporating time-window in trust establishment. We discussed how these countermeasures could effectively improve the robustness and trustworthiness of the social rating services.

**Acknowledgement**: This work is initiated while the first author was a visiting PhD student at the College of Computing, Georgia Tech. The second author is partially sponsored by grants from NSF CISE NetSE program, CyberTrust program, Crosscutting program, an IBM SUR grant, an IBM faculty award and a grant from Intel research council. The third author is partially supported by 973 Program (2011CB302305) and NNSF (61073015). Finally, we would like to thank the guest editors and reviewers for their helpful comments and suggestions.

## Reference

ADOMAVICIUS, G. AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17, 6, 734–749.

Amatriain, X., Lathia,N., M. Pujol, J., Kwak, H., and Oliver.N. 2009. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09). ACM, New York, NY, USA, 532-539.

BADGER, D. 2010.Amazon's Top Reviewers: The Bookstore That Corrupted Hadleyburg, http://www.dancingbadger.com/amareview.htm

BROWN, J. and MORGAN, J.2006.Reputation in Online Markets: Some Negative Feedback. IBER Working Paper, UC Berkeley.

CAVERLEE J, LIU L., WEBB S. 2010. The SocialTrust framework for trusted social information management: Architecture and algorithms. Inf. Sci. 180(1): 95-112 (2010)

DELLAROCAS, C. 2003. The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. Manage. Sci. 49, 10, 1407-1424.

DOUCEUR, J. R. 2002.The sybil attack. In Proceedings of IPTPS.

FALOUTSOS, M., FALOUTSOS, P. and FALOUTSOS, C. 1999.On power-law relationships of the Internet topology. SIGCOMM Comput. Commun. Rev. 29, 4, 251-262.

FENG, Q., SUN, Y., LIUL., YANG, Y. and DAI, Y. 2010.Voting Systems with Trust Mechanisms in Cyberspace: Vulnerabilities and Defenses. IEEE Transactions on Knowledge and Data Engineering. pp. 1766-1780.

FENG, Q. 2011.Research on Malicious and Multi-Attribute Problems in Recommender Systems, PhD Dissertation.

HARMON, A. 2004.Report: Glitch IDs anonymous Amazon reviewers. http://web.archive.org/web/20080309051211/http://www.cnn.com/2004/TECH/internet/02/14/glitch.reveals.ap/index.html. CNN.com.

HINES, M. 2007. Scammers gaming youtube ratings for profit, InfoWorld. http://www.infoworld.com/article/07/05/16/cybercrooks_gaming google_1.html

HOFFMAN, K.,ZAGE, D. and NITA-ROTARU, C. 2007.A survey of attack and defense techniques for reputation systems. Technical Report CSD TR #07-013, Purdue University.

JOSANG, A. and ISMAIL, R. 2002. The beta reputation system. In Proceedings of the 15th Electronic Commerce Conference.

KASNECI, G., GAEL, J.V., STERN, D. and GRAEPEL, T. 2011. CoBayes: bayesian knowledge corroboration with assessors of unknown areas of expertise. Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11). 465-474.

KUTER, U. and GOLBECK, J. 2010. Using probabilistic confidence models for trust inference in Web-based social networks. ACM Trans. Internet Technol. 10, 2, Article 8 (June 2010).

LAM, S. K. AND RIEDL, J. 2004.Shilling recommender systems for fun and profit. In Proceedings of the 13th international conference on World Wide Web.393–402.

MOBASHER, B., BURKE, R.,BHAUMIK,R. and WILLIAMS, C. 2007.Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transaction on Internet Technology.7, 4, Article 23.

PARSA, A. 2009.Belkin's Development Rep is Hiring People to Write Fake Positive Amazon Reviews. http://www.thedailybackground.com/2009/01/16/exclusive-belkins-development-rep-is-hiringpeople-to-write-fake-positive-amazon-reviews/2009.

RESNICK, P. and VARIAN, H. R. 1997. Recommender systems. Commun. ACM 40, 3, 56-58.

RESNICK, P.,ZECKHAUSER,R.,SWANSONAND, J. and LOCKWOOD, K.2006. The value of reputation on eBay: A controlled experiment, Experimental Economics. Vol9, No.2, 79-101.

SALEH,K.2008. An interview with Digg top user.http://www.invesp.com/blog/social-media/an-interview-with-digg-top-user.html.Social Media.

SCIRETTA, P.2008. IMDb Watch, Are Dark Knight Fanboys Burying The Godfather? http://www.slashfilm.com/2008/07/28/imdb-watch-are-dark-knight-fanboys-burying-the-godfather/

SRIVATSA, M. and LIU, L.2006.Securing decentralized reputation management using TrustGuard. J. Parallel Distrib. Comput. 66, 9, 1217-1232.

STERN, D. H., HERBRICH, R., AND GRAEPEL, T. 2009. Matchbox: large scale online bayesian recommendations. Proceedings of 18th international conference on World wide web. 111–120.

SUN, Y. L. HAN, Z. YU, W. LIU, K. J. R. 2006, A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. Proceedings of IEEE INFOCOM.

TAOBAOZUAN, 2010, http://www.taobaozuan.com.

TRAN, N., MIN, B., Li, J. and SUBRAMANIAN, L. 2009. Sybil-resilient online content voting. In Proceedings of the 6th USENIX symposium on Networked systems design and implementation. USENIX Association, Berkeley, CA, USA, 15-28.

Von Ahn, L., Maurer ,B., McMillen, C., Abraham, D.,and Blum , M. reCAPTCHA: Human-Based CharacterRecognition via Web Security Measures. Science, Vol. 321 no. 5895 pp. 1465-1468, September 2008.

VU, L., PAPAIOANNOU, T. and ABERER, K. 2010.Impact of Trust Management and Information Sharing to Adversarial Cost in Ranking Systems. IFIP Advances in Information and Communication Technology, Trust Management IV, Volume 321/2010, 108-124.

WALTER,F. E., BATTISTON, S., and SCHWEITZER, F. 2009. Personalised and dynamic trust in social networks. In *Proceedings of the third ACM conference on Recommender systems*. ACM, New York, NY, USA, 197-204.

XIONG, L. and LIU, L. 2004. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities, IEEE Transactions on Knowledge and Data Engineering, Vol.16, No. 7 (2004): 843-857.

YANG, Y., FENG, Q., SUN, Y. and DAI, Y. 2009. Dishonest Behaviors in Online Rating Systems: Cyber Competition, Attack Models, and Attack Generator. J. Comput. Sci. Technol. 24(5): 855-867.

YU, H., KAMINSKY, M., GIBBONS, P. B., and FLAXMAN,A. 2006.SybilGuard: defending against sybil attacks via social networks. *SIGCOMM Comput. Commun. Rev.* 36, 4 (August 2006), 267-278.

ZARRELLA, D. 2009. Not Everything That Can be Counted Counts. http://pistachioconsulting.com/shortyawards-gaming/.