Can You Trust Online Ratings? A Mutual Reinforcement Model for Trustworthy Online Rating Systems

Hyun-Kyo Oh, Sang-Wook Kim, *Member, IEEE*, Sunju Park, and Ming Zhou

***Abstract*—The average of customer ratings on a product, which we call a reputation, is one of the key factors in online purchasing decisions. There is, however, no guarantee of the trustworthiness of a reputation since it can be manipulated rather easily. In this paper, we define false reputation as the problem of a reputation being manipulated by unfair ratings and design a general framework that provides trustworthy reputations. For this purpose, we propose TRUE-REPUTATION, an algorithm that iteratively adjusts a reputation based on the confidence of customer ratings. We also show the effectiveness of TRUE-REPUTATION through extensive experiments in comparisons to state-of-the-art approaches.**

***Index Terms*—False reputation, robustness, trust, unfair ratings.**

I. Introduction

WHILE using online shopping channels, consumers share their purchasing experiences regarding both goods and services with other potential buyers via evaluation. The most common way for consumers to express their level of satisfaction with their purchases is through online ratings. The overall buyers' satisfaction is quantified as the aggregated score of all ratings and is available to all potential buyers. In this paper, we call this aggregated score for a product its reputation. The reputation of a product plays an important role as a guide for potential buyers and significantly influences consumers' final purchasing decisions [7], [9], [17], [21].

"Is the Product's Reputation Trustworthy?" Reputation is the score of a product obtained through collective intelligence, i.e., the result of collaboration between many individuals. The trustworthiness of a reputation can be achieved when a large number of buyers take part in ratings with honesty [10], [13], [15], [25]. If some users intentionally give unfair ratings to a product, especially when few users have participated, the reputation of the product could easily be manipulated. In this paper, we define false reputation as the problem of a reputation being manipulated by unfair ratings. In the case of a newly-launched product, for example, a company may hire people in the early stages of promotion to provide high ratings for the product. In this case, a false reputation adversely affects the decision making of potential buyers of the product.

In this paper, we describe the scenarios in which a false reputation occurs and propose a general framework that resolves a false reputation. The most common way to aggregate ratings is to use the average (i.e., to assign the same weight to each rating), which may result in a false reputation. For example, a group of abusers may inflate or deflate the overall rating of a targeted product. The existing strategies [2], [4], [11], [20], [29], [33] avoid a false reputation by detecting and eliminating abusers. However, abusers cannot always be detected, and it is possible that normal users may be regarded as abusers. Consequently, existing strategies can exclude the ratings of normal users or allow the ratings of abusers to be included in the calculation of a reputation.

The proposed framework, on the other hand, uses all ratings. It evaluates the level of trustworthiness (confidence) of each rating and adjusts the reputation based on the confidence of ratings. We have developed an algorithm that iteratively adjusts a reputation based on the confidence of customer ratings. By adjusting a reputation based on the confidence scores of all ratings, the proposed algorithm calculates the reputation without the risk of omitting ratings by normal users while reducing the influence of unfair ratings by abusers. We call this algorithm, which solves the false reputation problem by computing the true reputation, TRUE-REPUTATION.

The computation of a trustworthy reputation starts by measuring the confidence of a rating. We have surveyed previous social science studies that analyzed the characteristics of reliable online information and adopted three key characteristics that are suitable for determining the confidence of a rating [6], [23]. According to previous research, the reliability of online information increases when an information producer has no bias, maintains an objective perspective (objectivity)

Manuscript received July 4, 2014; revised November 8, 2014; accepted January 19, 2015. Date of publication April 9, 2015; date of current version November 13, 2015. This work was supported in part by the National Research Foundation of Korea (NRF) through the Korean Government under Grant NRF-2014S1A3A2044046, in part by the Ministry of Science, ICT and Future Planning (MSIP), Korea, under Information Technology Research Center Support Program NIPA-2014-H0301-14-1022 supervised by the National IT Industry Promotion Agency (NIPA), in part by Semiconductor Industry Collaborative Project between Hanyang University and Samsung Electronics Company Ltd., and in part by the NRF through the Korean Government (MSIP) under Grant NRF-2014R1A2A1A10054151. This paper was recommended by Associate Editor J. Lu.

H.-K. Oh and S.-W. Kim are with the Department of Computer and Software, Hanyang University, Seoul 133-791, Korea (e-mail: wook@agape.hanyang.ac.kr).

S. Park is with the School of Business, Yonsei University, Seoul 120-749, Korea.

M. Zhou is with Microsoft Research Asia, Beijing 100080, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2015.2416126

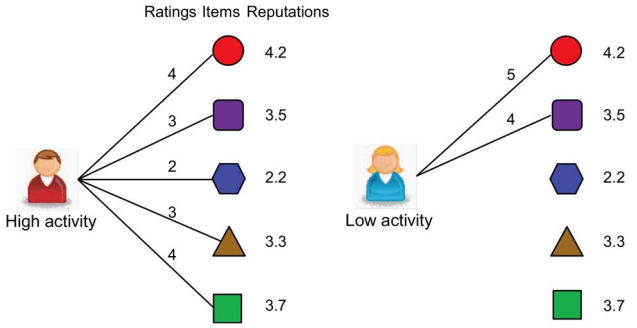

Fig. 1. Two different states of user activity.

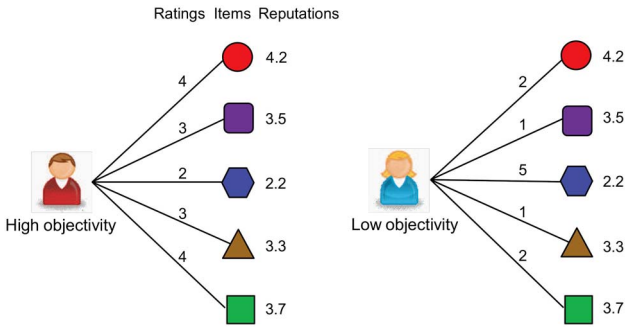

Fig. 2. Two different states of user objectivity.

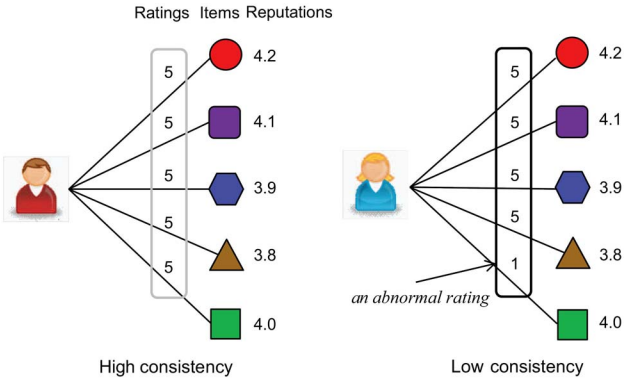

Fig. 3. Two different states of user consistency.

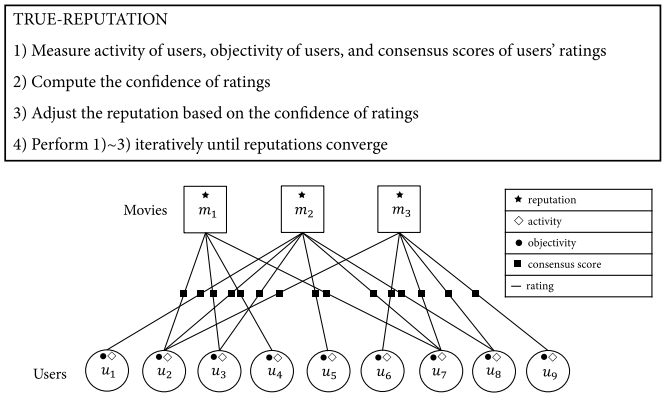

Fig. 4. General process of TRUE-REPUTATION.

and has a consistent viewpoint (consistency). In addition, the reliability of information increases when an information producer actively interacts with users who have obtained information through him (activity).

To determine the confidence of a rating, therefore, we have adopted three key factors of activity, objectivity, and consistency and defined these factors in the context of online ratings. First, the user who rates more items displays a higher level of activity. The above description of activity implies that the activity is defined by the amount of interactions between an information producer and the users obtaining his information. There exist, however, no interactions between users in an online rating system; instead, there are actions by users on products. Therefore, we measure user activity in an online rating system based on the amount of actions by the user on products (i.e., the number of products he rates). In Fig. 1, the user on the left shows a higher level of activity than the user on the right because the number of ratings by the user on the left is greater than that by the user on the right.

Second, a rating is considered more objective if it is closer to the public's evaluation (i.e., a reputation). The objectivity of a rating is defined as the deviation of the rating from the general reputation of the item. The more similar are the rating and the reputation, the higher is the objectivity of a rating; the more dissimilar they are, the lower the objectivity of a rating. Additionally, a user whose ratings exhibit higher objectivities should also have a higher level of user objectivity. The user objectivity is measured by the normalized average of the objectivities of the ratings submitted by that user. In Fig. 2, the user on the left whose ratings are similar to the reputations of the items exhibits higher objectivity than the user on the right whose ratings are quite different from the reputations of the items.

Third, we define the user consistency as how consistent the user is in rating products; in other words, how consistently he keeps his objectivities of ratings. In Fig. 3, the user on the left has rated with consistency. The user on the right, on the other hand, was consist until she rated the last item. That is, the user on the left has higher consistency in his ratings compared to the user on the right. An abnormal rating that deviates from the user's consistency is penalized by assigning a low consensus score when computing the confidence of the rating.

The objectivity of a rating is calculated based on the deviation of the "rating" from the "reputation" of the product. The difficulty in computing a reputation lies in the fact that the reputation itself is the sum of the ratings adjusted by the confidence, and the confidence of an individual rating is computed using the objectivity of the rating, which uses the reputation in its computation. In other words, the reputation and the confidence of a rating interact with each other in mutual reinforcement. We propose TRUE-REPUTATION, an iterative method, to compute these measures.

Fig. 4 shows the general process of TRUE-REPUTATION with a mini-example dataset containing nine users (u_1–u_9) and three items (m_1–m_3). An edge represents the rating given by a user to an item. Initially, the reputation of each item (denoted by the black star) is the average of all user ratings. At each iteration, TRUE-REPUTATION computes the confidence of

each rating based on the user activity (denoted by the white diamond), the user objectivity (denoted by the black circle), and the rating consensus score (denoted by the black square). Then, TRUE-REPUTATION adjusts the reputation of each item based on the confidence of the ratings. TRUE-REPUTATION performs these two steps (computing the confidence of ratings and adjusting the reputation of items) iteratively until all reputations converge to a stable state. More details are described in Section IV.

The proposed framework does not require clustering or classification, both of which necessitate considerable learning time. Though TRUE-REPUTATION does not require any learning steps when solving a false reputation, extensive experiments show that TRUE-REPUTATION provides more trustworthy reputations than do algorithms based on clustering or classification.

The contributions of this paper are as follows. First, we have defined false reputation and categorized various real-life scenarios in which a false reputation can occur. The categorization of the false-reputation scenarios helps us design experimental scenarios similar to real-life situations. Second, we have proposed a general framework to address a false reputation by quantifying the level of confidence of a rating. The framework includes TRUE-REPUTATION, an algorithm that iteratively adjusts the reputation based on the confidence of customer ratings. Third, we have verified the superiority of TRUE-REPUTATION by comparing it with machine-learning-based algorithms through extensive experiments.

This paper is organized as follows. Section II introduces related work. Section III describes various scenarios in which a false reputation occurs. Section IV develops the computational model that quantifies the confidence of a rating based on objectivity, activity, and consistency and describes the TRUE-REPUTATION algorithm. Section V verifies the superiority of TRUE-REPUTATION in comparison with previously suggested algorithms via extensive experiments. Section VI concludes this paper with directions for future research.

II. Related Work

Numerous studies have been conducted to improve the trustworthiness of online shopping malls by detecting abusers who have participated in the rating system for the sole purpose of manipulating the information provided to potential buyers (e.g., reputations of sellers and recommended items). Especially in the fields of multiagent and recommendation systems, various strategies have been proposed to handle abusers who attack the vulnerability of the system.

Multiagent systems compute and publish the reputation scores of sellers based on a collection of buyer opinions (which can be viewed as ratings). Strategies for improving the robustness of multiagent systems can be classified into two categories. The first group of strategies is based on the principle of majority rule. Considering the collection of majority opinions (more than half the opinions) as fair, this group of strategies excludes the collection of minority opinions, viewed as biased, when calculating the reputation [2], [24], [29]. The second group of strategies computes the reputation score of the seller based on the ratings of a target buyer and the ratings of a selected group of users whose rating patterns are very similar to that of the target buyer [18], [22], [28], [31], [32]. This group of strategies considers the ratings of the buyers whose rating patterns are different from that of the target buyer as biased and excludes these ratings when calculating the reputation.

Our framework for online rating systems and the existing strategies in multiagent systems serve the same purpose in that they are trying to address unfair ratings by abusers. It should be noted that the "seller" is the object evaluated in multiagent systems, while the "item" is the object evaluated in online rating systems. In multiagent systems, a buyer can evaluate a seller multiple times since he rates a seller whenever he purchases an item. In online rating systems, on the other hand, a buyer can give only a single rating per item. Thus, the relationship between buyers and items is significantly different from the relationship between buyers and sellers; as such, the graph structure of an online rating system is very different from that of a multiagent system. This paper uses an approach that considers the relation between buyers and items.

Recommendation systems predict the preference of a user for an item (such as books or movies) that they have not yet purchased using a model based on either the characteristics of an item (content-based approaches), the user's rating history (collaborative filtering approaches), or both (hybrid approaches that combine both content-based and collaborative-filtering approaches) [5], [12], [26], [27]. These systems are known to be vulnerable to a profile injection attack (which is also called a shilling attack) where malicious users try to insert fake profiles into the recommendation systems in order to increase the popularity of target item(s) [1], [4], [8], [20], [30].

In order to enhance the robustness of recommendation systems, it is imperative to develop detection methods against shilling attacks. Major research in shilling attack detection falls into three categories: 1) classifying shilling attacks according to different types of attacks [4]; 2) extracting attributes that represent the characteristics of the shilling attacks and quantifying the attributes [1], [33]; and 3) developing robust classification algorithms based on the quantified attributes used to detect shilling attacks [11], [14], [20], [30], [33].

The purpose of our framework is the same as that of existing strategies against shilling attacks; all are trying to prevent the manipulation of ratings by abusers. The classification algorithms for detecting shilling attacks, however, may face situations where malicious users cannot be detected and/or where normal users are considered as malicious. As a result, there may be instances when a reputation is calculated without the ratings of normal users or including the ratings of malicious users. Additionally, a significant amount of time is required to collect training data and extract attributes related to the abusers. The performance of the classifier is sensitive to the choice of the training data and the attributes used.

TRUE-REPUTATION, on the other hand, uses all ratings to calculate the reputations of items without the risk of losing the ratings of normal users. Thus, it is possible to

reduce the influence of unfair ratings from malicious users by assigning a confidence score to all ratings. Furthermore, TRUE-REPUTATION, a graph-based algorithm, does not require machine-learning and thus saves time.

III. False Reputation

In an online rating system, it is almost impossible to obtain the ground-truth data because there is no way of knowing which users have caused a false reputation in a real-life database. We artificially establish various situations in which a false reputation may occur and test the performance of the proposed algorithm in these situations. In order to claim that the generated situations are likely to occur in real-life online rating systems, we list various scenarios involving a false reputation and categorize them according to the types of user and situations. The experimental scenarios in Section V are based on these scenarios.

In this section, we define dangerous users who cause a false reputation and dangerous situations leading to a false reputation. Using the definitions of dangerous users and dangerous situations, we specify the scenarios in which a false reputation occurs.

A. Dangerous Users

Based on observations of online rating systems, we identified two types of abusers who present unfair ratings regardless of the quality of the product.

1) *Planned Attacker:* A planned attacker is a user who "intentionally" manipulates the reputation of a target product(s) by giving unfair ratings. This user may be hired by a company to improve the reputation of its product or to damage the reputation of competitors' products [16], [19]. Sometimes, planned attackers act as a group to influence public opinion on a target product [19].
2) *Unplanned Attacker:* An unplanned attacker is either an extremist who evaluates the quality of a product according to "abnormal" standards or a don't-carer who "without planning" provides meaningless ratings. An example of an extremist is a user who gives an extremely high rating to an author he prefers regardless of the quality of the book. An example of a don't-carer is the user who gives a meaningless high rating to a product to receive points or freebies from an online shopping mall. The ratings given by these unplanned attackers deviate from the general tendency of users and create a distortion in a product's reputation, which in turn deteriorates the trustworthiness of its reputation.

B. Dangerous Situations

Most goods and services in online markets receive little public attention. In order to attract attention, companies attempt to generate positive public opinion about their products from the moment of, or even before, the release of products. The reputation of products at the early stage of the product life cycle (such as new movies or new books) can be easily manipulated.

TABLE I
False-Reputation Scenarios

	Product launch phase	Unpopular products
Planned attacker	Hired planned attackers manipulate the reputation of a product during the product launch phase	Hired planned attackers manipulate the reputation of an unpopular product
Unplanned attacker	Extremists give biased ratings or don't-carers give meaningless ratings to a product during product launch phase	The product is unpopular and attracts unplanned attackers who give distorted ratings

Dangerous situations in which a false reputation can occur are as follows.

1) *Product Launch Phase:* Before the release of a new product, there is no customer experience on which to base an opinion. Online rating systems often allow users to evaluate products, such as prerelease movies, before their release. Opinions at prerelease can include vague expectations by unplanned attackers or manipulated opinions by planned attackers. Furthermore, the number of opinions about a product in the launch phase may be too limited to trust the reputation of the product.
2) *Unpopular Products:* In online shopping malls, many products are unpopular with few ratings. Because of this, the overall opinion about the unpopular product appears to be untrustworthy.

C. False Reputation Scenarios

False reputation occurs when "dangerous users" enter "dangerous situations." Table I summarizes the scenarios in which a false reputation may occur.

We establish several experimental scenarios in our experiments. First, we generate dangerous users who behave as either a planned or unplanned attacker. Second, we set up dangerous situations to simulate a product launch phase or situations involving unpopular products.

We call the planned attacker who gives unfair manipulated ratings or the unplanned attacker who gives distorted ratings rating attackers (RAs). The goal of RAs is either "push" or "nuke." Push means that the RA gives an unfairly high rating to promote a specific product, and nuke means that the RA gives an unfairly low rating to demote a certain product.

Referring to the behavior of various shilling attackers from the field of recommendation systems [20], who have the same goals as those of RAs, we generate profiles of various types of RAs, such as target-only, average, random, selected popular, reverse selected popular, and love/hate RAs. The products with ratings between 90 and 110 are considered either newly released or unpopular products targeted by RAs. The detailed descriptions of the profiles of various RAs and the features of products in dangerous situations are given in Section V.

IV. Computation Model

Reputation is computed based on ratings adjusted by confidence. The confidence of a rating is calculated based on

TABLE II
NOTATIONS OF TRUE-REPUTATION

Name	Description
$\mathbf{U}$	set of users
$\mathbf{M}$	set of items
$\mathbf{R}$	set of ratings
u	user, $u \in \mathbf{U}$
m	item, $m \in \mathbf{M}$
$\mathbf{U}_m$	set of users who have rated item m, $\mathbf{U}_m \subset \mathbf{U}$
$\mathbf{R}^u$	set of ratings by user u, $\mathbf{R}^u \subset \mathbf{R}$
$\mathbf{R}_m$	set of ratings on item m, $\mathbf{R}_m \subset \mathbf{R}$

two scores, user activity and user objectivity, and is then penalized based on its abnormality determined according to user consistency.

The user activity is quantified as the total number of his ratings for any products. The objectivity of a rating on a particular product is measured as the deviation from the mean of the product ratings. The objectivity of the user is measured as the normalized average of the objectivities of the ratings submitted by that user. Based on how much the objectivity of a rating is consistent with those of the user's other ratings, an abnormal rating is defined at one that deviates from the user's consistency in rating items. We penalize a rating with an abnormality by assigning it a low consensus score when computing its confidence. Because the confidence of a rating and the reputation of a product are mutually dependent, they are computed using an iterative method. In this section, we propose a computational model that iteratively computes the trustworthy reputation.

A. Confidence of Rating

To compute the confidence of a rating, we use three values: user activity score, user objectivity, and rating consensus score. The notations used in TRUE-REPUTATION are presented in Table II.

1) User Activity: A user who posts many ratings should be considered an active user. The activity score of user u, denoted by a_u, is quantified by the frequency of his ratings $|\mathbf{R}^u|$. Because the number of ratings by each user varies, we normalize $|\mathbf{R}^u|$ by applying the transformation function Ψ as follows:

$$a_u = \Psi\left(|\mathbf{R}^u|, \alpha, \mu\right) \tag{1}$$

where

$$\Psi\left(|\mathbf{R}^u|, \alpha, \mu\right) = \frac{1}{1 + e^{-\alpha(|\mathbf{R}^u| - \mu)}}. \tag{2}$$

The Ψ function in (1) is the sigmoid function for normalization of $|\mathbf{R}^u|$. This not only keeps the value in the range of [0, 1] but also reduces the influence of a user with many ratings. The user whose a_u is 1 is the most active user. Parameters α ($\in \mathbb{Z}$) and μ ($\in \mathbb{Z}$) determine the slope and adjust the midpoint of the curve of Ψ, respectively. In order to distribute $|\mathbf{R}^u|$ evenly in the range of [0, 1], the values of α and μ should be determined appropriately. In order that Ψ should be closer to 1 when $|\mathbf{R}^u|$ is large, α should be a positive number. In this normalization, Ψ is closer to 1 when $|\mathbf{R}^u|$ is very large, equal to 0.5 when $|\mathbf{R}^u| = \mu$, and closer to 0 when $|\mathbf{R}^u|$ is very small. Based on our preexperiments, we use $\alpha = 0.02$.[1]

Generally, the average value is used for μ; however, online rating systems often include extremely active users who provide a comparatively large number of ratings. Therefore, in our experiments, μ is set as the average number of ratings per user after excluding the top 20 percent of users in terms of the number of ratings.

2) User Objectivity: User objectivity is defined as the normalized average of the objectivities of the ratings by a specific user. The objectivity of a rating captures the deviation of the rating from the reputation on the item. On the other hand, the objectivity of a "user" indicates the aggregation of the objectivities of the ratings by that user. In other words, all the ratings' objectivities by a user are collectively used to compute his objectivity.

The objectivity of rating r for item m $(r) \in \mathbf{R}_m$ is computed based on its deviation from the aggregated score of the other buyers' ratings on the same item (i.e., reputation). The objectivity of a rating, denoted by o_r, is higher when the rating is closer to the reputation. o_r is calculated based on the reputation, denoted by $\bar{r}_m$, and the standard deviation, denoted by s_m, as follows:

$$o_r = \left|\frac{r - \bar{r}_m}{s_m}\right|. \tag{3}$$

Equation (3) indicates that the user with rating r performs a more objective evaluation on item m when o_r is closer to 0.

User objectivity, denoted by o_u, is calculated as the average of the objectivities of the ratings by that user. If this value is closer to 0, the user is more objective. We define o_u as follows:

$$o_u = \frac{1}{|\mathbf{R}^u|} \sum_{r \in \mathbf{R}^u} o_r. \tag{4}$$

In order to use o_u in calculating the confidence of a rating, we need to normalize o_u. The computational model defines the user whose normalized objectivity is "1" as the most objective user. The transformation function ψ for normalizing o_u is defined as follows:

$$o_u^* = \Psi\left(o_u, \alpha', \mu'\right). \tag{5}$$

o_u^* is normalized by the Ψ function with the α' and μ' parameters in (5). The Ψ function in (5) is the sigmoid function for normalization of o_u to keep the value in the range of [0, 1]. It also reduces the influence of a user who has a large o_u value.[2]

α' must be a negative number in order that Ψ should be close to 1 when o_u moves closer to 0. Based on our preexperiments, we set α' to be -2.5 for reasons similar to

[1] We tried various values for α and found several α values that work well and show similar trends. TRUE-REPUTATION demonstrated the best performance when α is 0.02.

[2] Note that the same sigmoid function as in (2) is used but with different parameter values (α' determines the slope and μ' controls the midpoint of the sigmoid curve), causing the two sigmoid functions to behave differently.

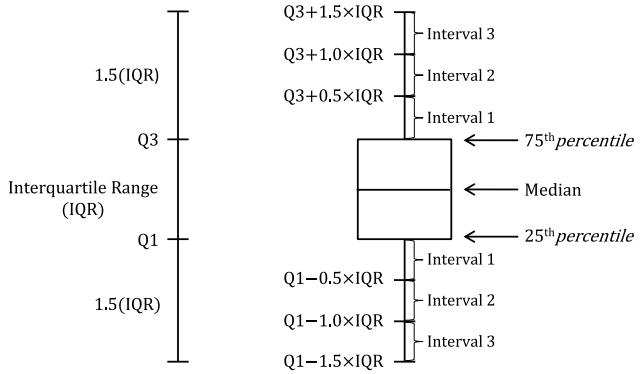

Fig. 5. Box-plot for user's consistency.

that of (2).[3] μ' is assigned to be the average of all user objectivity. In this normalization, Ψ is closer to 1 when o_u is very small, equal to 0.5 when $o_u = \mu$, and closer to 0 when o_u is very large.

3) User Consistency: By analyzing the abnormality of a rating with respect to user consistency in rating items, we assign each rating a consensus score, c_r. The goal here is to detect an abnormal rating r ($r \in \mathbf{R}_u$) that deviates from the user's normal behavior: an abnormal rating r is given a low consensus score, while a rating consistent with the user's normal behavior is given a high consensus score.

Box-plot analysis [34] is used to analyze the distribution of o_rs for each user and to detect abnormal ratings. Fig. 5 shows the box-plot analysis. We sort the objectivities of the ratings (o_r) in ascending order. The first quartile ($Q1$), median, and third quartile ($Q3$) values are used for analysis. In the box-plot analysis, the range between $Q3$ and $Q1$ is called the interquartile range (IQR), and values less than $Q1-(1.5\times\text{IQR})$ or greater than $Q3 + (1.5 \times \text{IQR})$ are considered outliers [3]. In this paper, values within the IQR are considered normal behavior, with a consensus score of 1, and outliers are considered abnormal behavior, with a consensus score of 0. A rating outside the range of the IQR is more highly penalized as o_r moves farther from the median. More specifically, rating r will be given consensus score c_r according to the following[4]:

$$c_r = \begin{cases} 0, & \text{if } o_r > Q3 + 1.5\text{IQR} \\ & \text{or } o_r < Q1 - 1.5\text{IQR} \\ 0.5, & \text{if } o_r \le Q3 + 1.5\text{IQR} \quad \text{and } o_r > Q3 + 1.0\text{IQR} \\ & \text{or } o_r \ge Q1 - 1.5\text{IQR} \quad \text{and } o_r < Q1 - 1.0\text{IQR} \\ 0.7, & \text{if } o_r \le Q3 + 1.0\text{IQR} \quad \text{and } o_r > Q3 + 0.5\text{IQR} \\ & \text{or } o_r \ge Q1 - 1.0\text{IQR} \quad \text{and } o_r < Q1 - 0.5\text{IQR} \\ 0.9, & \text{if } o_r \le Q3 + 0.5\text{IQR} \quad \text{and } o_r > Q3 \\ & \text{or } o_r \ge Q1 - 0.5\text{IQR} \quad \text{and } o_r < Q1 \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

[3]Although several other α' values performed well, TRUE-REPUTATION has demonstrated the best performance when α' is -2.5.

[4]Although other sets of consensus scores were found to work well, the consensus score of 0.5 for the ratings in interval 1, 0.7 for the ratings in interval 2, and 0.9 for the ratings in interval 3 have demonstrated the best performance in our framework.

B. Iterative Computation

TRUE-REPUTATION is an iterative algorithm. The initial reputation of an item is set to be the arithmetic mean of all ratings provided; thus, all ratings have the same confidence scores. At each iteration, TRUE-REPUTATION recomputes the confidence of each rating and adjusts the reputation of each item based on the recalculated confidence scores of all ratings. The algorithm stops when the computation converges to a stable state.

First, TRUE-REPUTATION computes the a_u for all users. Since a user's activity level does not change, a_u is computed only once. Next, it computes o_r ($r \in \mathbf{R}_m$) for all ratings of all items, which is then used to compute both o_u and c_r. After computing o_u for all users, the value is normalized to o_u^* using the Ψ function. The consensus score c_r of a rating r ($r \in \mathbf{R}^u$) is also computed. The confidence of a rating t_r is computed as follows:

$$t_r = a_u \times o_u^* \times c_r, \ \ r \in \mathbf{R}^u. \tag{7}$$

The reputation of each item is then adjusted based on the confidence of the ratings on the same item as follows:

$$\bar{r}_m = \frac{\sum_{r\in\mathbf{R}_m}(r \times t_r)}{\sum_{r\in\mathbf{R}_m} t_r}. \tag{8}$$

At each iteration, TRUE-REPUTATION recomputes o_u^* and c_r based on $\bar{r}_m$, adjusted at the previous iteration and recomputes t_r using existing the a_u and newly computed o_u^* and c_r. Then, the reputation of each item is adjusted using the newly computed t_r. The process of computing t_r and adjusting $\bar{r}_m$ continues until the values reach a stable state.

The stability of TRUE-REPUTATION is measured by the marginal change in reputations between iterations. Let the vector $\vec{r}$ represents the reputation of all items. The kth element $\bar{r}_k$ in the vector $\vec{r}$ represents the reputation of the kth item in $\mathbf{M}$ ($|\mathbf{M}| = l$). The vector $\vec{r}$ is represented as $\vec{r} = [\bar{r}_1, \ldots, \bar{r}_l]$. The difference between the old vector $\vec{r}^*$ and the new vector $\vec{r}$ is measured as one minus the cosine similarity between $\vec{r}^*$ and $\vec{r}$ as follows:

$$d(\vec{r}^*, \vec{r}) = 1 - \frac{\vec{r}^* \cdot \vec{r}}{\| \vec{r}^* \| \| \vec{r} \|}. \tag{9}$$

If $d(\vec{r}^*, \vec{r})$ is less than some preset value δ (0.000001 in our experiments), TRUE-REPUTATION stops. The overall algorithm is presented below.

V. Experiments

In our experiments, we used the MovieLens 100k[5] dataset, which consists of 100 000 ratings using a scale from 1 (bad) to 5 (excellent) for 1682 movies submitted by 943 users. Each user rated at least 20 movies. Demographic data, such as age, gender, and occupation, were provided in the dataset.

A. Rating Attack Models

The proposed framework provides potential buyers with trustworthy reputations by reducing the influence of RAs. To verify the performance of the proposed framework,

[5]http://www.grouplens.org/node/73

Algorithm 1 TRUE-REPUTATION

Input: set of users $\mathbf{U}$, set of items $\mathbf{M}$, set of ratings $\mathbf{R}$
Output: reputations of all items

```
for u ∈ U do
  Compute a_u
end for
repeat
  for m ∈ M do
    for r ∈ R_m do
      Compute o_r
    end for
  end for
  for u ∈ U do
    Compute o_u
    Compute o_u^* from o_u
  end for
  for u ∈ U do
    for r ∈ R^u do
      Compute c_r
      Compute t_r
    end for
  end for
  for m ∈ M do
    r̄_m = (Σ_{r∈R_m} (r × t_r)) / (Σ_{r∈R_m} t_r)
  end for
  r⃗* = r⃗
  r⃗ = [r̄_1, ..., r̄_l]
until cosine similarity of r⃗ and r⃗* is less than δ
```

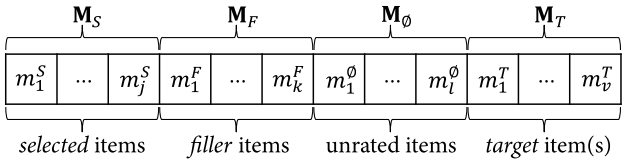

Fig. 6. General form of the RA profile adopted from [15].

we generated various types of RAs based on different attack profiles and constructed the scenario described in Section III-C where a false reputation could occur by injecting the RAs into MovieLens.

We adopted shilling attack profiles from a recommendation system [20]. The generic form of an RA profile is shown in Fig. 6. The RA profile consists of four item sets: a set of target items ($\mathbf{M}_T$) which are the targets of RAs, a set of selected items ($\mathbf{M}_s$) which are selected differently depending on the types of an attack, a set of filler items ($\mathbf{M}_F$) which are chosen randomly, and a set of unrated items ($\mathbf{M}_\emptyset = \mathbf{M} - (\mathbf{M}_T \cup \mathbf{M}_s \cup \mathbf{M}_F)$). How the items of $\mathbf{M}_s$ are selected and how the items in $\mathbf{M}_T$, $\mathbf{M}_s$, and $\mathbf{M}_F$ are rated, various attack models can be generated.

1) Push RA Models: The simplest strategy to cause a false reputation is for RAs to generate unfair (maximum or minimum) rating(s) to targeted item(s) only. An attacker may behave more strategically by trying to camouflage himself as an ordinary user; he gives ratings close to the reputations of items other than the targeted item(s), to which he gives the maximum or minimum rating(s). The RA models with the purpose of push are summarized as follows.

a) *Target-Only RA:* The profile of target-only RA is comprised of the ratings assigned to the items in $\mathbf{M}_T$. For push, the maximum rating (in our experiments, five) is assigned. $\mathbf{M}_s$ and $\mathbf{M}_F$ are an empty set.
b) *Average RA:* The profile of average RA consists of the maximum rating assigned to the item[6] in $\mathbf{M}_T$ and the ratings close to or the same as the reputation of the items in $\mathbf{M}_F$. $\mathbf{M}_s$ is an empty set.
c) *Random RA:* The profile of random RA is the same as that of average RA but the ratings assigned to items in $\mathbf{M}_F$ are randomly selected from the normal distribution with the mean of the average reputation for all items in the system and the standard deviation of the reputations for all items in the system.
d) *Selected Popular RA:* $\mathbf{M}_s$ in a selected popular RA profile consists of popular items, which receive many ratings from users and have high reputation scores. The profile of selected popular RA comprises the maximum rating for the targeted item in $\mathbf{M}_T$, randomly chosen ratings for filler items in $\mathbf{M}_F$, and maximum ratings assigned to popular items in $\mathbf{M}_s$.

2) Nuke RA Models: Note that target-only RA, average RA, and random RA, can also be used with nuke if the maximum rating(s) assigned to item(s) in $\mathbf{M}_T$ is (are) replaced by the minimum rating(s) (in our experiments, one). In addition, the following describes RA models designed particularly for nuke.

a) *Love/Hate:* The profile of love/hate RA consists of the minimum rating assigned to the targeted item in $\mathbf{M}_T$ and the maximum ratings for items in $\mathbf{M}_F$.
b) *Reverse Selected Popular RA:* Reverse selected popular RA, a variation of selected popular RA, gives low ratings to the items in $\mathbf{M}_s$, unpopular items that have received many ratings but have low reputation scores.

Table III summarizes the RA models. Each cell shows the items selected for each set and the ratings assigned to the items.

B. Experimental Setup

First, we need to select the target movies to be attacked by RAs in order to build false reputation scenarios. We analyzed all the movies in MovieLens and the relationships between the number of ratings for each movie and its reputation. Table IV shows the statistics, where each cell represents the number of movies.

MovieLens consists of 1682 movies. Among them, 1089 movies have been rated by less than 50 users, and more than half of the 1089 movies have been rated by less than 10 users. Note that the movies with a small number of ratings can be easily manipulated by RAs. It is impossible to convey a trustworthy reputation when the majority of users who have rated a

[6] $\mathbf{M}_T$ of all the types of RA consists of only a targeted item while that of target-only RA consists of several targeted items.

TABLE III
SUMMARY OF RA MODELS

Attack Model	Attack Type	$\mathbf{M}_S$	$\mathbf{M}_F$	$\mathbf{M}_\emptyset$	$\mathbf{M}_T$
Target-Only	Push/nuke	Not used	Not used	The items of $\mathbf{M}$ that are not in $\mathbf{M}_T$; no ratings	Target items, r_{max}/r_{min}
Average	Push/nuke	Not used	Filler items, ratings assigned with normal distribution around item mean	The items in $\mathbf{M}$ that are not in the union of $\mathbf{M}_T$ and $\mathbf{M}_F$; no ratings	The target item, r_{max}/r_{min}
Random	Push/nuke	Not used	Filler items, ratings assigned with normal distribution around system mean	The items in $\mathbf{M}$ that are not in the union of $\mathbf{M}_T$ and $\mathbf{M}_F$; no ratings	The target item, r_{max}/r_{min}
Selected Popular	Push	Popular items, r_{max}	Filler items, ratings assigned with normal distribution around system mean	The items in $\mathbf{M}$ that are not in the union of $\mathbf{M}_T$ and $\mathbf{M}_F$; no ratings	The target item, r_{max}
Love/Hate	Nuke	Not used	Filler items, r_{max}	The items in $\mathbf{M}$ that are not in the union of $\mathbf{M}_T$ and $\mathbf{M}_F$; no ratings	The target item, r_{min}
Reverse Selected Popular	Nuke	Widely disliked items, r_{min}	Filler items, ratings assigned with normal distribution around system mean	The items in $\mathbf{M}$ that are not in the union of $\mathbf{M}_T$ and $\mathbf{M}_F$; no ratings	The target item, r_{min}

TABLE IV
STATISTICS OF THE MOVIES IN THE MOVIELENS 100K DATASET

Number of Ratings	Reputation			
	1-2-	2-3-	3-4-	4-5-
1-50	129	426	454	80
50-100	0	51	180	28
100-150	0	14	100	19
150-200	0	4	65	15
200-250	0	3	35	11
>250	0	1	44	23

TABLE V
NUMBER OF RATINGS IN MOVIELENS

Number of ratings						
20-50	50-100	100-150	150-200	200-250	250-300	>300
380	202	135	78	57	38	53

movie are RAs, because reputation adjustment depends on the ratings of ordinary users. Movies with few ratings, therefore, are excluded from targets by RAs. In our experiments, movies with ratings between 90 and 110 are considered as movies in the dangerous situation (newly released movies or unpopular movies) and are selected as targets for RAs.[7] Among these, movies with a reputation score greater than 3.53 (the average reputation score of all movies in MovieLens) are targeted for push while the rest are used for nuke. We assume that the users in MovieLens are ordinary users who rate the movies fairly.

Second, in our experiments, we varied three factors: 1) the number of RAs inserted to the MovieLens dataset; 2) the RA models used; and 3) the rating frequencies of RA.

First, we varied the number of RAs attacking the targeted movie, from 5% of its total number of ratings to 30%, in increments of 5%. Second, we used six different RA models (target-only RA, average RA, random RA, selected popular RA, love/hate RA, and reverse selected popular RA). Third, we varied the rating frequencies based on observations from the MovieLens dataset.

The analysis of MovieLens reveals that the average number of ratings per user is 108, although more than 60% of users have fewer than 100 ratings. Following the characteristics of the majority of users in MovieLens, we assigned the number of ratings per RA to not exceed 100. Table V shows the number of users in each category from MovieLens.

In the case of target-only RA, we chose 32 target movies and inserted target-only RAs. The number of ratings by a target-only RA is set to be 2, 4, 8, 16, or 32. In the case of other RA models, we choose ten target movies and inserted each type of RAs. The number of strategic ratings for each type of RA is set to be 50, 75, or 100.

Third, to evaluate the effectiveness of the three key factors of activity, objectivity, and consistency, we compared a baseline algorithm with the three variations of the TRUE-REPUTATION algorithm, which differs in the computation of the rating confidence.

1) *ARITHMETIC-MEAN:* The baseline algorithm that assigns the same confidence to all ratings.
2) *ACTIVITY:* The algorithm that uses user activity to determine the confidence of ratings.
3) *OBJECTIVITY:* The algorithm that uses user objectivity to determine the confidence of ratings.
4) *ACTIVITY&OBJECTIVITY (A&O):* The algorithm that uses the product of user objectivity and activity to determine the confidence of ratings.

[7]In our preexperiments, we evaluated the performances of TRUE-REPUTATION on the target movies having various numbers (i.e., 30–300) of ratings. The difference among them was insignificant. In this paper, we showed the results for the target movies whose number of ratings was around 100 (i.e., 90–110).

Fourth, we compared the performance of TRUE-REPUTATION with those of two existing algorithms by modifying them as reputation-adjustment algorithms.

1) *iCLUB′:* The clustering-based algorithm from multiagent systems.
2) *MOBASHER:* The classification-based algorithm from recommendation systems.

The performance of iCLUB [18], based on the Density-based spatial clustering of applications with noise (DBSCAN) algorithm, has been shown to be superior to that of existing algorithms, such as Bayesian Reputation Systems [29] and TRAVOS [28]. In order to compare iCLUB with TRUE-REPUTATION, we created a variation of iCLUB called iCLUB′. iCLUB′ divides all users into similar user groups, including RAs, by their similarity using DBSCAN. The DBSACN algorithm in iCLUB′ uses a cosine similarity measure, and the parameters are set as follows: radius = 10 and MinPts = 1.[8] In iCLUB′, all ratings for the target movies are replaced with the new ones and the reputations of the movies are adjusted using these new ratings. A user rating for the target movie is replaced with a new one, which is the average of ratings from the users who have rated the target movie and have been in the same user group. If a user has no similar users, the user does not replace his ratings.

MOBASHER is a modified version for reputation adjustment of the classifier proposed by Mobasher *et al.* [20]. MOBASHER uses a classifier to detect abusers in ratings systems and adjusts reputations by excluding the ratings of the abusers. Generally, the performance of classification algorithms for detecting abusers depends on the way the training dataset is built. In our experiments, the users, including RAs who rated all the targeted movies, served as a test set and the other users as a training set. The classifier is constructed after injecting all different types of RAs introduced in Section V-B into the training set. Each user has fifteen features, as proposed by Mobasher *et al.* [20].

Finally, the performance of each algorithm is evaluated as the difference between the reputation with RAs and the reputation without RAs, as shown in (9). We expect that when using ARITHMETIC-MEAN, the reputation would increase or decrease at a constant rate according to the increase in the number of RAs. We expect, on the other hand, when using TRUE-REPUTATION, the reputation would change very little when using TRUE-REPUTATION due to the reduced influence of RAs

$$\text{Reputation Change Rate} = \frac{|\text{Reputation with RAs} - \text{Reputation w/o RAs}|}{\text{Reputation w/o RAs}}. \quad (10)$$

C. Experimental Results

Through experiments, we answer the following questions.

1) Are the three key factors (user activity, user objectivity, and consensus score derived from user consistency) meaningful in measuring the confidence of the rating?
2) Is the proposed framework superior to the existing approaches?
3) How do change in the number of RAs, rating frequencies by RAs, purpose of RAs, and type of RAs, each influence the performance of various reputation adjustment algorithms?

[8]iCLUB has shown the best performance with these values [18].

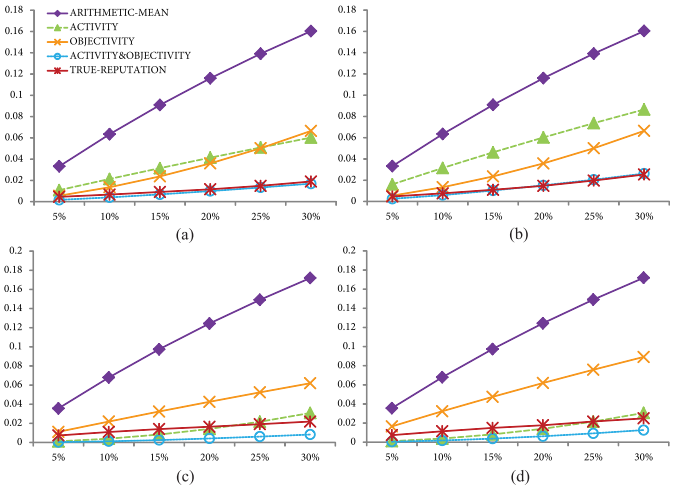

Fig. 7. Evaluation for effectiveness of activity, objectivity, and consistency. Push RA (a) rating frequency = 2 and (b) rating frequency = 32. Nuke RA (c) rating frequency = 2 and (d) rating frequency = 32.

First, by comparing TRUE-REPUTATION to the three variations of the TRUE-REPUTATION, ACTIVITY, OBJECTIVITY, and A&O, we examined the effectiveness of the three key factors. Fig. 7 shows the rate of change in a reputation according to the number of inserted RAs. The reputation-change rate of ARITHMETIC-MEAN was used as the baseline against which to compare those of the other proposed algorithms.

As shown in Fig. 7, the reputation-change rate of TRUE-REPUTATION is remarkably less than that of ARITHMETIC-MEAN. When the number of target-only RAs attacking the targeted movie is 30% of its total number of ratings, the reputation-change rate of TRUE-REPUTATION is less than 0.03 while that of ARITHMETIC-MEAN is greater than 0.16. The results indicate that TRUE-REPUTATION is robust against target-only RAs, while ARITHMETIC-MEAN remains vulnerable. TRUE-REPUTATION reduces the influence of RAs using the confidence of ratings, while ARITHMETIC-MEAN gives equal weight to all ratings. The performance of OBJECTIVITY and ACTIVITY is better than that of ARITHMETIC-MEAN, but A&O shows superior performance to both ACTIVITY and OBJECTIVITY. In the case of target-only RAs, the results of A&O are slightly superior to that of TRUE-REPUTATION because the ratings of a target-only RA are constantly unfair (either good or bad) and so receive high consensus scores.

By comparing Fig. 7(a) and (c) to Fig. 7(b) and (d), we verified that the impact of RAs is more prominent with an increase in frequency of ratings by them. Since this trend is similar in all the different rating frequencies, we showed the results of the smallest (2) and largest rating frequencies (32).

We conducted a series of two-sample *t*-tests at a 95% confidence level between TRUE-REPUTATION and the four algorithms (ARITHMETIC-MEAN and the three variations of

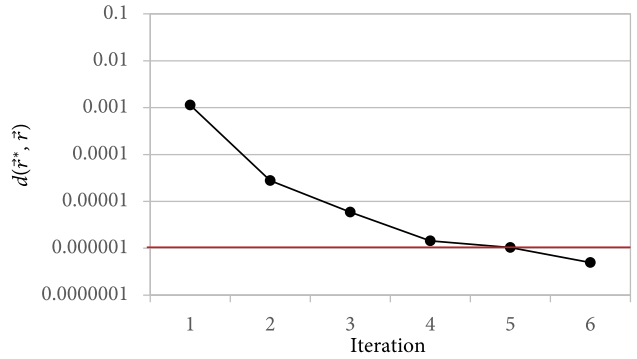

Fig. 8. Stability of TRUE-REPUTATION.

TRUE-REPUTATION). When push target-only RAs are present the rating frequency of the RAs is 32, and the number of ratings by the RAs is 30% of the target movie's total number of ratings,[9] the t-test results show that the p values between TRUE-REPUTATION and ACTIVITY, OBJECTIVITY, or ARITHMETIC-MEAN are below 0.05, which indicates that there are statistically significant differences between TRUE-REPUTATION and the three algorithms. The only exception is the case between TRUE-REPUTATION and A&O where the p-value is greater than 0.05. This indicates that although the result reported in Fig. 7 indicates that A&O is better than TRUE-REPUTATION, there is no statistically significant difference between the two.

Fig. 8 shows the stability in reputations over iterations when attacked by push target-RAs. The x-axis indicates the number of iterations, and the y-axis indicates the stability at each iteration on log scale. Stability is defined as the distance (i.e., 1 − cosine similarity) between the old vector $\vec{r}^*$ and the new vector $\vec{r}$, as described in (9). The distance between $\vec{r}^*$ and $\vec{r}$ decreases as iterations proceed and the stopping criterion, indicated by the horizontal line in Fig. 8, is met only after the fifth iteration. Similar trends in stability are displayed when attacked by other types of RAs.

The results shown in Fig. 7 confirm that user activity and user objectivity are useful for reputation adjustment, especially when used simultaneously. In the following experiments, we compared the two proposed algorithms, A&O and TRUE-REPUTATION, and the existing algorithms, iCLUB′ and MOBASHER.

Fig. 9 shows the results comparing the baseline with the four reputation adjustment algorithms. iCLUB′ shows the worst performance amongst the four algorithms. Its performance is similar or even inferior to that of baseline regardless of the number of target-only RAs. Since DBSCAN in iCLUB′ groups the users by their similarity, RAs are grouped separately from ordinary users. Since a user is supposed to consult the users in his group when adjusting reputations, iCLUB′ behaves almost the same as the baseline in which the user uses his own information only. The performance of iCLUB′ confirms that it cannot reduce the impact of target-only RAs and cannot adjust reputations properly. The performance of MOBASHER,

[9]For t-tests, we fixed the number of ratings by RAs to be 30% of the target movie's total number of ratings.

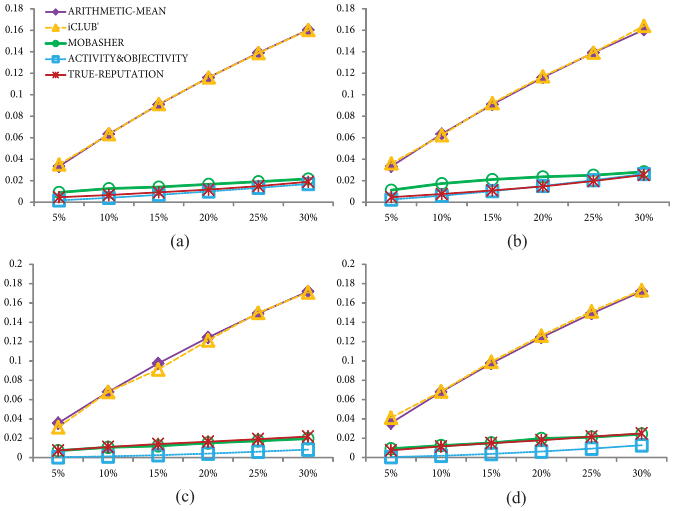

Fig. 9. Reputation-change rates by target-only RA. Push RA (a) rating frequency = 2 and (b) rating frequency = 32. Nuke RA (c) rating frequency = 2 and (d) rating frequency = 32.

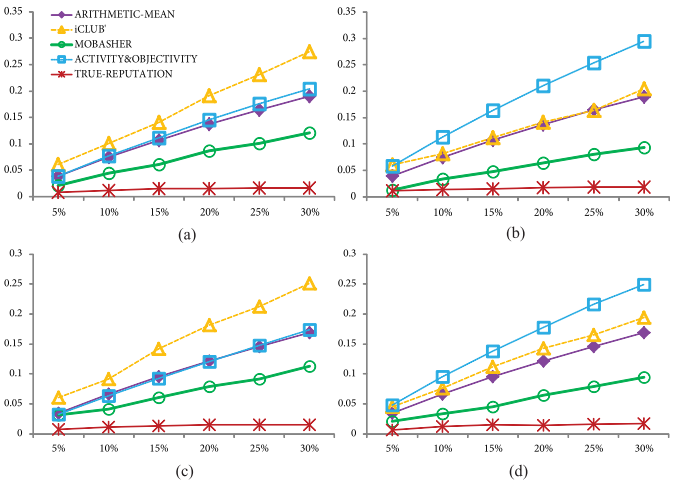

Fig. 10. Reputation change rates by average RA. Push RA (a) rating frequency = 50 and (b) rating frequency = 100. Nuke RA (c) rating frequency = 50 and (d) rating frequency = 100.

on the other hand, is similar to that of the proposed algorithms, because the classifier of MOBASHER is able to detect target-only RAs with a high probability.

As shown in Fig. 9, the performance of iCLUB′ worsens with an increase in the number of RAs, while that of MOBASHER remains strong although not as good as those of A&O and TRUE-REPUTATION. When push target-only RAs are present and the rating frequency of the RAs is 32, the t-test results show that all the p values between TRUE-REPUTATION and A&O, or MOBASHER are greater than 0.05, which indicates that the algorithms are statistically indifferent.

Fig. 10 shows that A&O, the most robust algorithm in target-only RA, is more vulnerable to average RA. The reputation-change rate of A&O is greater than 0.3, when the number of average RAs is 30% of the targeted movie's total number of ratings and the frequency of the ratings by average RAs is 100. A&O is not able to reduce the influence of the average RAs because most ratings by average RAs are regarded as fair. The

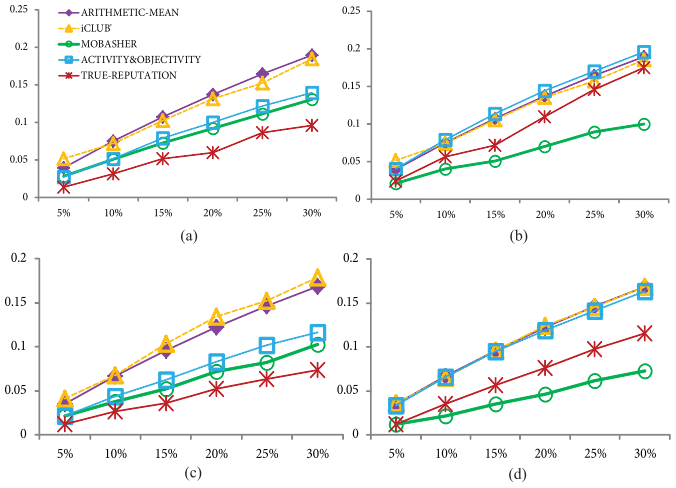

Fig. 11. Reputation change rates by random RA. Push RA (a) rating frequency = 50 and (b) rating frequency = 100. Nuke RA (c) rating frequency = 50 and (d) rating frequency = 100.

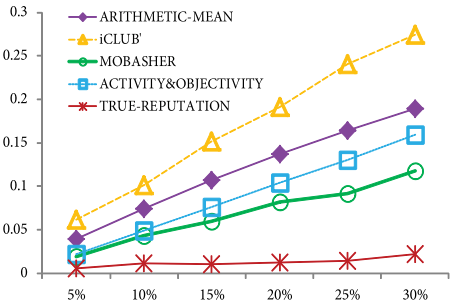

Fig. 12. Reputation change rates by selected popular RA (push, rating frequency = 50).

performance of A&O, dramatically deteriorates when the number of average RAs increases. In contrast, TRUE-REPUTATION, which considers user consistency, performs better regardless of the number of RAs and the frequency of the ratings by RAs. The reputation-change rate of TRUE-REPUTATION is always less than 0.02, which indicates TRUE-REPUTATION is able to reduce the influence of average RAs.

In addition, it should be noted that iCLUB′, which performs worse than baseline, is not suitable for fending off average RAs. MOBASHER performs better than iCLUB′, though not as well as TRUE-REPUTATION. When push average RAs are present and the rating frequency of the RAs is 100, the *t*-test results show that all the *p* values between TRUE-REPUTATION and ARITHMETIC-MEAN, iCLUB′, MOBASHER, or A&O are less than 0.05, which indicates there are statistically significant differences between TRUE-REPUTATION and the four algorithms.

Fig. 11 shows that the performance of all algorithms deteriorates when the number of random RAs increases. Random RAs are most difficult to detect, since most of their ratings, chosen with a normal distribution around the system, are meant to look similar to those of ordinary users. As a result, not only A&O, but also TRUE-REPUTATION perform relatively poorly in the case of random RAs. Again, iCLUB′ shows the worst performance, trailed by MOBASHER.

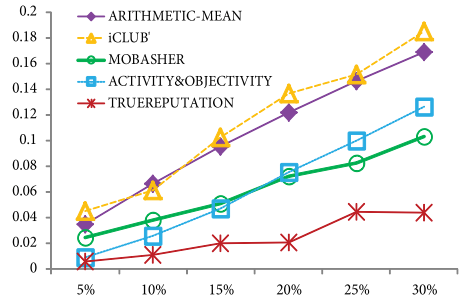

Fig. 13. Reputation change rates by reverse selected popular RA (nuke, rating frequency = 50).

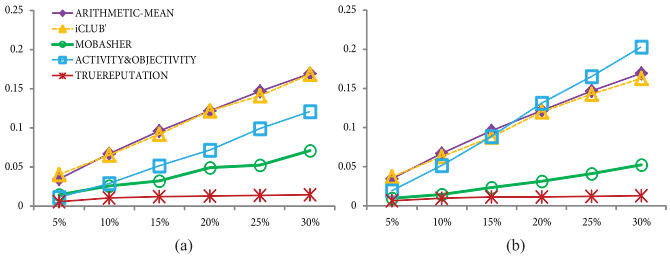

Fig. 14. Reputation change rates by love/hate RA. Nuke UR (a) rating frequency = 50 and (b) rating frequency = 100.

Figs. 12 and 13 show the performance of the algorithms in the presence of selected popular RA and reverse selected popular RA, respectively. In our experiments, the number of selected items in $\mathbf{M}_S$ is set at 40, and the number of filler items in $\mathbf{M}_F$ is set at 10. Because the selected popular RA and the reverse selected popular RA disguise themselves as ordinary users by giving fair ratings, submitting strategic high ratings to popular items and low ratings to unpopular items, respectively, A&O is not able to reduce their influence. On the other hand, TRUE-REPUTATION is able to reduce the influence of these two RAs. Compared to TRUE-REPUTATION whose reputation-change rate is always less than 0.05 regardless of the number of these RAs, the other algorithms perform worse with the increase in the number of RAs.

In the case of selected popular RA, all *p* values are less than 0.05, which indicates that there are statistically significant differences at the 95 percent confidence level between TRUE-REPUTATION and the four algorithms. In the case of reverse selected popular RA, the *t*-test results show that there are statistically significant differences between TRUE-REPUTATION and ARITHMETIC-MEAN, iCLUB′, and A&O. The only exception is the case between TRUE-REPUTATION and MOBASHER, where no statistically significant difference is found at the 95 percent confidence level.

Fig. 14 shows the performance of the love/hate RA with an increase in the number of RAs and rating frequency. Remember that love/hate RA gives the maximum ratings to filler items and the minimum rating to the target item. Because the love/hate RA gives the maximum ratings consistently, TRUE-REPUTATION, which considers the user's consistency, shows better performance than others. The reputation-change rate of TRUE-REPUTATION is always less than 0.02 regardless of the number of love/hate RAs.

As shown in Fig. 14, TRUE-REPUTATION is able to reduce the influence of the love/hate RA. MOBASHER also shows the good performance, though not as good as that of TRUE-REPUTATION. In the case of love/hate RAs, all p values are less than 0.05 which indicates that there are statistically significant differences between TRUE-REPUTATION and the four algorithms.

VI. Conclusion

This paper defines the false reputation problem in online rating systems and categorizes various real-life situations in which a false reputation may occur. The understanding of why and when a false reputation occurs helps us establish experimental situations. In order to solve the false reputation problem, we proposed a general framework that quantifies the confidence of a rating based on activity, objectivity, and consistency. The framework includes TRUE-REPUTATION, an algorithm that iteratively adjusts the reputation based on the confidence of user ratings. Through extensive experiments, we showed that TRUE-REPUTATION can reduce the influence of various RAs. We also showed that TRUE-REPUTATION is superior to the existing approaches that use machine-learning algorithms such as clustering and classification to solve the false reputation problem.

There are more factors (other than those addressed in this paper) known to be elemental in assessing the trust of users in the field of social and behavioral sciences. We plan to study how to incorporate them into our model to compute the reputation of items more accurately. In the e-market place such as Amazon.com and eBay.com, buyers give ratings on items they have purchased. We note, however, that the rating given by a buyer indicates the degree of his satisfaction not only with the item (e.g., the quality) but also with its seller (e.g., the promptness of delivery). In a further study, we plan to develop an approach to accurately separate an item score and a seller score from a user rating. Separating the true reputation of items and that of sellers would enable customers to judge items and sellers independently.

References

[1] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Classification features for attack detection in collaborative recommender systems," in *Proc. 12th Int. Conf. Knowl. Disc. Data Min. (KDD)*, Philadelphia, PA, USA, 2006, pp. 542–547.

[2] M. Brennan, S. Wrazien, and R. Greenstadt, "Using machine learning to augment collaborative filtering of community discussions," in *Proc. 9th Int. Joint Conf. Auton. Agents Multiagent Syst. (AAMAS)*, Toronto, ON, Canada, 2010, pp. 1569–1570.

[3] V. Barnett and T. Lewis, *Outliers in Statistical Data*, 3rd ed. Chichester, U.K.: Wiley, 1994.

[4] P. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proc. 7th Annu. ACM Int. Workshop Web Inf. Data Manage. (WIDM)*, Bremen, Germany, 2005, pp. 67–74.

[5] M. Eirinaki, M. D. Louta, and I. Varlamis, "A trust-aware system for personalized user recommendations in social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 409–421, Apr. 2014.

[6] M. Eisend, "Source credibility dimensions in marketing communication—A generalized solution," *J. Empir. Gener. Market. Sci.*, vol. 10, no. 2, pp. 1–33, 2006.

[7] S. Grazioli and S. L. Jarvenpaa, "Perils of Internet fraud: An empirical investigation of deception and trust with experienced Internet consumers," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 4, pp. 395–410, Jul. 2000.

[8] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: A comprehensive survey," *Artif. Intell. Rev.*, vol. 42, no. 4, pp. 767–799, 2014.

[9] G. Häuubl and V. Trifts, "Consumer decision making in online shopping environments: The effects of interactive decision aids," *Market. Sci.*, vol. 10, no. 1, pp. 4–21, 2000.

[10] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.

[11] N. Hurley, Z. Cheng, and M. Zhang, "Statistical attack detection," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, Vienna, Austria, 2009, pp. 149–156.

[12] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User Model. User-Adapt. Interact.*, vol. 22, nos. 1–2, pp. 101–123, 2012.

[13] C. Leadbeater, *WE-THINK: Mass Innovation, Not Mass Production*. London, U.K.: Profile Books, 2008.

[14] J.-S. Lee and D. Zhu, "Shilling attack detection—A new approach for a trustworthy recommender system," *INFORMS J. Comput.*, vol. 24, no. 1, pp. 117–131, 2012.

[15] P. Levy, *L'Intelligence Collective: Pour Une Anthropologie du Cyberspace*. Paris, France: La Découverte, 1997.

[16] E. P. Lim, V. A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Toronto, ON, Canada, 2010, pp. 939–948.

[17] M. Limayem, M. Khalifa, and A. Frini, "What makes consumers buy from Internet? A longitudinal study of online shopping," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 4, pp. 421–432, Jul. 2000.

[18] S. Liu, J. Zhang, C. Miao, Y. Theng, and A. Kot, "iCLUB: An integrated clustering-based approach to improve the robustness of reputation systems," in *Proc. 10th Int. Joint Conf. Auton. Agents Multiagent Syst. (AAMAS)*, Taipei, Taiwan, 2011, pp. 1151–1152.

[19] A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal, "Detecting group review spam," in *Proc. 20th Int. Conf. World Wide Web (WWW)*, Hyderabad, India, 2011, pp. 93–94.

[20] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Trans. Internet Technol.*, vol. 7, no. 2, pp. 1–40, 2007.

[21] The Nielsen Company, *Trends in Online Shopping*, Global Nielsen Consum. Rep., Feb. 2008. [Online]. Available: http://www.freshgraphics.net/BlogLinks/GlobalOnlineShoppingReportFeb08.pdf

[22] Z. Noorian, S. Marsh, and M. Fleming, "Multi-layer cognitive filtering by behavioral modeling," in *Proc. 10th Int. Joint Conf. Auton. Agents Multiagent Syst. (AAMAS)*, Taipei, Taiwan, 2011, pp. 871–878.

[23] S. Y. Rieh and D. Danielson, "Credibility: A multidisciplinary framework," *Annu. Rev. Inf. Sci. Technol.*, vol. 41, no. 1, pp. 307–364, 2007.

[24] E. Santos and D. Li, "On deception detection in multiagent systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 2, pp. 224–235, Mar. 2010.

[25] J. Surowiecki, *The Wisdom of Crowds: Why the Many are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations*. New York, NY, USA: Doubleday, 2004.

[26] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, no. 4, pp. 1–20, Aug. 2009

[27] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Providing justifications in recommender systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1262–1272, Nov. 2008.

[28] W. Teacy, J. Patel, N. Jennings, and M. Luck, "TRAVOS: Trust and reputation in the context of inaccurate information sources," *Auton. Agents Multi-Agent Syst.*, vol. 12, no. 2, pp. 183–198, 2006.

[29] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in Bayesian reputation systems," *Icfain J. Manage. Res.*, vol. 4, no. 2, pp. 48–64, 2005.

[30] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," in *Proc. 18th Int. Conf. Knowl. Disc. Data Min. (KDD)*, Beijing, China, 2012, pp. 985–993.

[31] B. Yu and M. P. Singh, "Detecting deception in reputation management," in *Proc. 2nd Int. Joint Conf. Auton. Agents Multiagent Syst. (AAMAS)*, Melbourne, VIC, Australia, 2003, pp. 73–80.

[32] J. Zhang and R. Cohen, "Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach," *Electron. Commer. Res. Appl.*, vol. 7, no. 3, pp. 330–340, 2008.

[33] S. Zhang, A. Chakrabarti, J. Ford, and F. Makedon, "Attack detection in time series for recommender systems," in *Proc. 12th Int. Conf. Knowl. Disc. Data Min. (KDD)*, Philadelphia, PA, USA, 2006, pp. 809–814.
[34] R: Box-Plot Statistic, *R Manual*, 2011. [Online]. Available: http://en.wikipedia.org/wiki/Box_plot

Hyun-Kyo Oh received the B.S. degree from Hanyang University, Seoul, Korea, in 2008, and the M.S. degree in electronics and computer engineering from Hanyang University, in 2010, where he is pursuing the Ph.D. degree in computer and software.

He was a Visiting Scholar with the Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, in 2013. He was a Research Intern with Microsoft Research Asia, Beijing, China, from 2014 to 2015. His current research interests include data mining, social network analysis, computational trust, and trust management.

Sang-Wook Kim (M'00) received the B.S. degree in computer engineering from Seoul National University, Seoul, Korea, in 1989, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1991 and 1994, respectively.

He was a Visiting Researcher with the Department of Computer Science, Stanford University, Stanford, CA, USA, in 1991. From 1995 to 2003, he served as an Associate Professor of the Division of Computer, Information, and Communications Engineering, Kangwon National University, Chuncheon, Korea. From 1999 to 2000, he was a Post-Doctorate with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. In 2003, he joined Hanyang University, Seoul, where he is currently a Professor with the Department of Computer Science and Engineering and the Director of the Brain-Korea-21-Plus Research Program. From 2009 to 2010, he was a Visiting Research Professor with the Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests include databases, data mining, multimedia information retrieval, social network analysis, recommendation, and Web data analysis. He has authored over 110 papers in refereed international journals and international conference proceedings.

Dr. Kim was a recipient of the Best Poster Presentation Award from the ACM International Conference on Information and Knowledge Management (ACM CIKM) in 2013 and the Best Paper Award from the 29th ACM International Symposium on Applied Computing (ACM SAC), the Outstanding Service Award from ACM SIGAPP, and the Outstanding Contributions Award from Database Society of Korea in 2014. He is currently an Associate Editor of *Information Sciences*. He was a General Co-Chair of the 6th International Conference on Computational Collective Intelligence Technologies and Applications in 2014, a Program Co-Chair of ACM International Conference on Ubiquitous Information Management and Communications in 2014 and the International Conference on Emerging Databases in 2013, and a Track Chair of the Social Network and Media Analysis Track in ACM SAC'14. He served for the Program Committees of over 80 international conferences, including the IEEE International Conference on Data Engineering, Very Large Databases, WWW, and ACM CIKM. He is a member of ACM.

Sunju Park received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, and the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA.

She has served on the faculties of Management Science and Information Systems, Rutgers University, New Brunswick, NJ, USA. She is a Professor of Operations, Decisions and Information with the School of Business, Yonsei University, Seoul. Her current research interests include analysis of online social networks, multiagent systems for online businesses, and pricing of network resources. Her publications include *Computers and Industrial Engineering*, *Electronic Commerce Research*, *Transportation Research*, *IIE Transactions*, the *European Journal of Operational Research*, the *Journal of Artificial Intelligence Research*, *Interfaces*, *Autonomous Agents and Multiagent Systems*, and other leading journals.

Ming Zhou received the degree from Chongqing University, Chongqing, China, in 1985, and the Ph.D. degree in computer science and engineering from the Harbin Institute of Technology, Harbin, China, in 1991.

He was an Associate Professor with Tsinghua University, Beijing, China, responsible for its NLP Group. He joined Microsoft in 1999 and established the Microsoft Research Asia-Natural Language Computing (MSRA-NLC) Group as one of its founders. He was a Researcher with the MSRA, Beijing, China, in 1999. He was a Post-Doctorate Researcher with Tsinghua University from 1991 to 1993. He joined the faculty of Tsinghua University, as an Associate Professor in 1993. He visited Kodensha Ltd., Tokyo, Japan, a famous machine-translation software maker in Japan from 1996 to 1999 to lead the Research and Development on Chinese-Japanese Machine Translation. He is a Principal Researcher and a Manager of the NLC Group at MSRA, where he is responsible for the research of input method editor, machine translation, natural language understanding, text mining, question-answering, chat-bot, multilanguage knowledge base, and computer poetry and riddles.