An Iterative Algorithm for Reputation Aggregation in Multi-dimensional and Multinomial Rating Systems

Mohsen Rezvani[1] Mohammad Allahbakhsh[2]
Aleksandar Ignjatovic[1] Sanjay Jha[1]

[1] University of New South Wales
{mrezvani,ignjat,sanjay}@cse.unsw.edu.au

[2] University of Zabol
allahbakhsh@uoz.ac.ir

Technical Report
UNSW-CSE-TR-201502
January 2015

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Online rating systems are widely accepted as a means for quality assessment on the web, and users increasingly rely on these systems when deciding to purchase an item online. This fact motivates people to manipulate rating systems by posting unfair ratings scores for fame or profit. Therefore, both building useful realistic rating scores as well as detecting unfair behaviours are of very high importance. Existing solutions are mostly majority based, also employing temporal analysis and clustering techniques. However, they are still vulnerable to false ratings. They also ignore distance between options, provenance of information and different dimensions of cast rating scores while building trust and rating scores. In this paper, we propose a robust iterative algorithm which leverages the information in the profile of raters, provenance of information and a decay function for the distance between options to build decent rating scores for items and trust ranks for the people. We have implemented and tested our rating method using simulated data as well as three real world datasets. Our tests demonstrate that our model calculates realistic rating scores even in the presence of massive false ratings and outperforms well-known algorithms in the area.

1 Introduction

Nowadays, the products and contents being advertised or published on the web is so tremendous that its is almost impossible to assess their quality based n the one's personal experiences. Also, millions of people generate contents or advertise products online and it is almost unlikely for a customer to have a personal experience of how trustworthiness a seller might be. One of the widely used methods to overcome this problem is relying on the feedback received from the others who have experiences of buying a product or having relations with a particular person. This is done through *online rating systems* which collects feedback and opinions from members or visitors of an online community and, based on these opinions, assigns a quality level score to every product and person in the community. The IMDb[1], Amazon[2] online market and ebay[3] are some of well-known online rating systems.

One of the big issues with the online rating systems is the credibility of the quality ranks that they produce. Such quality ranks are produced mainly based on the feedback received from others in the forms of textual or numeric opinions. Users who have posted such feedback might have different levels of expertise and experiences. They also might have different individual or group interests and benefits upon which they post unfair feedback [1, 2, 3]. An unfair feedback is a feedback that does not reflect the real opinion of a person on a product and has been posted, regardless of the real quality of a product, based on personal or group interest. Such unfair feedback while are taken in to account result in manipulated quality ranks which are not reliable anymore. Several pieces of evidence show that the online rating systems are widely subject to such attacks [2, 3]. Studies show that the collaborative attacks, also called *collusion*, are more harmful to rating systems than individual attacks [4, 5].

Several solutions have been proposed to deal with collusion in rating systems. Some studies rely on clustering techniques to analyze the behaviour of raters and find the abnormal ones, in order to detect collusion [6, 5, 7]. The main problem with these solutions is their scalability. Clustering techniques are generally based on NP-Hard graph clustering techniques and when the size of an online systems is too large, which is completely common, these techniques are not applicable anymore. The other class of solutions to collusion problems is the iterative techniques [8, 9, 10]. These techniques while perform reasonable well against collusion, are still vulnerable to knowledgeable sophisticated attacks [11, 12].

We have recently proposed an algorithm called Rating-Through-Voting (RTV) [13] and an extended version of it in [14] which outperforms the previous work in terms of detection and elimination of unfair behaviour. The RTV algorithms tries to iteratively find the community sentiment and use it as a gold standard to assess quality of products and trustworthiness of raters simultaneously [13]. The conformance to the community sentiment is the only parameter that this algorithm takes into account when calculating quality and trust scores. In [14] we proposed an extension of RTV in which the helpfulness of cast feedback is also taken into account. Although RTV and its extension outperform the other related work and show a reasonable behaviour, they have still limitations that

[1] http://www.imdb.com/

[2] http://www.amazon.com/

[3] http://www.ebay.com/

need more research and investigations.

The first limitation arises from the fact that in RTV algorithm the rating task is reduced to a voting task. In an election, the order of the choices is not important and the distance between the choices is not defined. For example, when a voter chooses the Nominee$_1$ in an election and another voter selects the Nominee$_2$, it does not make sense to talk about the distance between these two options. But such distances are important in rating tasks. More precisely, when the real quality of a product is, say 5, the credibility of a rater's choice , say r_1, who has chosen 3 is more than the 1 which has been chosen by r_2, because the distance between 3 and 5 is less than the distance between 1 and 5. Consequently, the credit that the r_1 receives from his vote should be more than what r_2 gains. The distance between choices is not taken into account in RTV algorithm.

Moreover, in a rating system, raters may assess quality of a product, a service or a person from different aspects. For instance, in eBay's detailed seller rating system, buyers express their opinion on the quality of a transaction form four different aspects, i.e., *Item as described*, *Communication*, *Shipping time*, and *Shipping charges* [15]. For a reputation or a rating score to be more credible, it is needed that the reputation management system aggregate the scores received for all different aspects in order to build the final reputation score. This is another limitation of the exiting algorithms.

Finally, the provenance of a rating score is another piece of information that is ignored in the existing related work. The contextual information around a cast rating score can give the system so many useful hints to adjust its weight and credibility. The profile of the rater, the time a feedback has been cast, the geographical region, etc., are examples of contextual meta data that can be taken into account in reputation computation.

In this paper we propose a novel method for calculating robust and credible reputation and rating scores. The proposed method is based on the RTV algorithm which we have proposed in [13]. The reputation computation method we propose here takes into account the distance between options in order to fairly propagate credibility from raters to options and vice versa. To do so, we use a decay function to adjust the credibility is propagated from raters to various options and from options to the trust ranks of raters. We also, consider the different dimensions of the cast rating scores and utilize them in order to build more realistic and credible reputation and rating scores for people and products. Finally, our proposed method takes advantage from the provenance of the cast feedback when calculating reputation and rating scores; and consequently computes more informative and reasonable scores. We have assessed the accuracy of the rating scores computed by our proposed method using both synthetic and three real-world datasets. The evaluation results show superiority of our method over three well-known algorithms in the area.

The rest of this paper is organized as follows. Section 2 formulates the problem and specifies the assumptions. Section 3 presents our novel reputation system. Section 4 describes our experimental results. Section 5 presents the related work. Finally, the paper is concluded in Section 6.

2 Preliminaries

2.1 Basic Concepts and Notation

Assume that in an online rating systems a set of n users cast ratings for m items. Each user may rate one or more items and each item might be rated from K different prospectives. As an example, consider e-bay in which a user can rate a seller from various prospectives such as Deliverables, Communication, and Attitude. We represent the set of cast ratings by a three dimensional matrix $A_{n\times m\times K}$ in which $A_{i,j,k}$ $(1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq K)$ is the rating cast by user i on the item j from the k^{th} prospective. We suppose that rating scores are selected according to Likert technique [16], i.e., they are selected from a discrete set of numbers each of which represent a quality level, for example 1-Start to 5-Stars.

2.2 Rating through Voting

In our previous work [13], we reduced the problem of rating to a voting task. More precisely, when a rater chooses a quality level, say 4-Starts to represent quality of a product, one can say that the rater believes that 4-Stars represents the quality of the product better than other options, so he has voted for it out of the list of 1-Star to 5-Stars options. Therefore, we called our algorithm Rating-Through-Voting (RTV) algorithm.

In RTV, we assign each quality level a credibility degree to show how credible this quality level is to represent the real quality of the item. Then we aggregate the credibility degrees of all quality levels a users has voted for to build the trustworthiness of the user. Hence, there is an interdependency between credibility degrees and trustworthiness scores. We formulated this interdependency by proposing a fixed point algorithm which converges to a credibility degrees for each quality level as well as a trust score for each user.

Assume that for each item l, there is a list of options $\Lambda_l = \{I_1^l, \dots, I_{n_l}^l\}$ and each user can choose maximum one option for each item. We define the credibility degree of a quality level I_i on list Λ_l, denoted by ρ_{li} as follow:

$$\rho_{li} = \frac{\sum_{r\,:\,r\to li} (T_r)^\alpha}{\sqrt{\sum_{1\leq j\leq n_l} \left(\sum_{r\,:\,r\to lj} (T_r)^\alpha\right)^2}} \tag{2.1}$$

where $r \to li$ denotes the fact that user r has chosen option I_i^l from list Λ_l. $\alpha \geq 1$ is a parameter. T_r is the trustworthiness of user r which is obtained as:

$$T_r = \sum_{l,i\,:\,r\to li} \rho_{li} \tag{2.2}$$

In order to compute both the credibility degrees and trust scores, we have proposed an iterative algorithm which starts with identical trustworthiness values for all users, $T_r^{(0)} = 1$. In each iteration, we first update the credibility degrees of all quality levels using Eq. (2.1) and then the trust scores of users are recomputed by Eq. (2.2). This iteration stops when the credibility scores converge to a fixed point, i.e., when there is no considerable changes for the credibility degrees. We also presented a proof for the convergence of the algorithm. Given the credibility degrees by the iterative algorithm, we proposed a weighted averaging to obtain the aggregate rating score of each item l, denoted by $R(\pi_l)$ as follow:

$$R(\pi_l) = \sum_{1 \leq i \leq n_l} \frac{i \times \rho_{li}^p}{\sum_{1 \leq j \leq n_l} \rho_{lj}^p} \tag{2.3}$$

where $p \geq 1$ is a parameter for controlling the averaging affect.

3 Reputation Aggregation System

In this section, we propose a new reputation aggregation technique based on our previous work, the RTV algorithm [13]. we consider the rating provenance as well as credibility propagation in a multi-dimensional rating system. To this end, we first define a decaying function to formulate the distance between quality levels. We then leverage such distance formulation to extend our basic equations for computing credibility levels and users' trusts. We also define the concept of rating provenance and extend our computations to consider such provenance. Finally, we proposed a method to obtain the final reputation values in a multi-dimensional rating system.

3.1 Distance Between Nominal Values

In most of social rating systems, such as eBay 5-star feedback system, there is a numerical distance between the existing options. In order to take into account such distances in our reputation propagation method, we formulate the distance using a decaying function.

One can use any decreasing function, symmetric around the origin, i.e., such that $d(x) = d(-x)$. Here we define the distance of two options i and j as $d(i,j) = q^{|i-j|}$, where q is the *base distance*, $0 < q < 1$ and is defined as the distance value between two continuous options. Figure 3.1 shows how the distance value between two options increases exponentially using distance function $d(i,j)$. We assume that there is a limited range for the ratings in the rating system. The main condition is that the sum of all distances must be equal to a constant value, we call it *propagation parameter* and denoted as b. The propagation parameter is a positive value which controls the proportion of credibility propagation among options. By taking into account this condition, we have

$$\begin{aligned} q + q^2 + \cdots + q^{n_l - j} + q + q^2 + \cdots + q^{j-1} = b \ &\Leftrightarrow \\ q(1 + q + \cdots + q^{n_l-j-1}) + q(1 + q + \cdots + q^{j-2}) = b \ &\Leftrightarrow \\ q(\frac{1 - q^{n_l-j}}{1-q}) + q(\frac{1-q^{j-1}}{1-q}) = b \ &\Leftrightarrow \\ 2 - q^{j-1} - q^{n_l-j} = b\frac{1-q}{q} & \end{aligned} \tag{3.1}$$

Considering the condition $0 < q < 1$, Eq. (3.1) has only one real answer for each value of b. For example, such equation makes q equal to 0.34 when $b = 0.5$, and

0.52 when $b = 1.0$. In Section 4.2 we investigate the impact of various values of propagation parameter b over the accuracy of our reputation system.

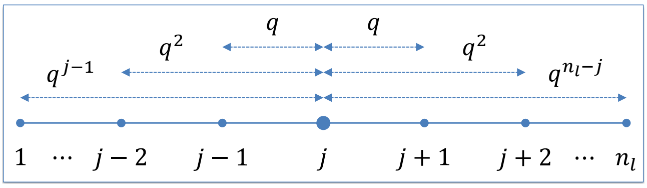


Figure 3.1: Exponential increasing trend in the proposed distance function.

3.2 Provenance-Aware Credibility Propagation

Given the distance function $d(i, j)$ for computing the numerical distance between options i and j, we can update our computation equations for the credibility degree as well as users' trustworthiness. Firstly, we define β_{li} as the non-normalized credibility degree of quality level li. Considering the idea of credibility propagation among the options, the credibility degree of a quality level is obtained not only from the raters who have selected that particular level, but also from all raters who chosen such item with proportion of the distance of their choices from such level. In other words, we define the credibility degree for a quality level in an item as amount of credibility which such level can obtain from all raters who rated such an item. Therefore, we reformulate the Eq. (2.1) for computing the non-normalized credibility degree of quality level li as follows:

$$\beta_{li} = \sum_{j,r\,:\,r \to lj} (T_r)^{\alpha} d(i, j) \tag{3.2}$$

Some rating systems provide contextual information about the ratings, we call it *rating provenance*. It contains attributes such as watching duration in a movie ratings system and educational level of raters in a student feedback system, which provides more information about either the raters or the environment of ratings. Thus, a reputation system needs to take into account these contextual attributes as meta data about the quality of ratings. Clearly, each rating system has its own list of contextual attributes. Thus, in this paper we propose our provenance model based on the attributes provided by a student feedback system which includes two contextual attributes: staff/non-staff and watching behaviour of students. We will show the details and evaluation results of our algorithm over a dataset from such system in Section 4.6. We note that the approach can be easily adopted for other contextual attributes. Moreover, the proposed provenance model is based on the approach proposed in [17], albeit this approach has been proposed in the context of participatory sensing.

The main idea of our provenance model is to define a weight function for considering the contextual attributes provided by the rating system. To this end, we define a weight function for each attribute and then we aggregate all the weight values from these functions using the simple product of the weights to obtain the provenance weight. Such provenance weight is used to update the computation of credibility level as well as users' trustworthiness in our system.

In the student feedback system, students are asked to rate to the movies of an online course. For each rating, the system provides the staff status of the rater as well as the spent time for watching the movie by such rater. We utilized both the staff information and watching duration as two contextual attributes to create the rating provenance. To this end, we consider an slightly higher credibility for the staff raters. Thus, we define the *staff weight*, denoted as w_s, which is set $w_s = 0.98$ for staff raters and $w_s = 0.95$ for non-staff raters. Moreover, we take into account the watching time due to the fact that a student who spends enough time to watch a whole movie can provide higher quality ratings. We denote the watching time provided for each rating and the original duration of its corresponding movie as T_r and T_v, respectively. Thus, we compute the gap between them by $|\min\{T_r, T_v\} - T_v|$. Now, we define the *watching time weight*, denoted as w_t:

$$w_t = e^{-|\min\{T_r,T_v\}-T_v|\times\beta} \tag{3.3}$$

where $0 \leq \beta \leq 1$ is the *duration sensitivity* parameter which controls the watching time weight. Note that Eq. (3.3) makes w_t equal to 1 when the time gap between the watching and duration is 0 and w_t approaches 0 when such gap is large. Given both staff and watching time weights, we define *provenance weight*, denoted as w_p through aggregating these two weights as:

$$w_p = w_s \times w_t \tag{3.4}$$

Note that in general the provenance weight can be define as the product of the weight values for all contextual attributes, where such weights are in the range of [0,1]. Given the provenance weight, we re-write Eq. (3.2) as follows:

$$\beta_{li} = \sum_{j,r\,:\,r\to lj} (T_r)^{\alpha} \times d(i,j) \times w_p \tag{3.5}$$

For normalizing the credibility degree, we use the same method used in our previous approach which is:

$$\rho_{li} = \frac{\beta_{li}}{\sqrt{\sum_{1\leq j\leq n_l} (\beta_{lj})^2}} \tag{3.6}$$

Given the credibility degree for all quality levels of items, we can update the trustworthiness of users. Such trustworthiness for a user is the weighted sum of all credibility degrees from all quality levels of items which has been rated by such user. The weight here is the distance between the chosen level by such user and the credible level. Thus, we have

$$T_r = \sum_{l,i\,:\,r\to li} \sum_{1\leq j\leq n_l} \rho_{li} \times d(i,j) \times w_p \tag{3.7}$$

Note that we formulated the uncertainty in rating systems through both credibility propagation among options and rating provenance. Thus, we considered them in computing both credibility degrees and users' trustworthiness.

3.3 Iterative Vote Aggregation

Given equations (3.5), (3.6) and (3.7), we have interdependent definitions for credibility degree and trustworthiness. Clearly, the credibility degree of a quality level in an item depends on the trustworthiness of users who rated to such item. on the other hand, the trustworthiness of a user is dependent on the credibility degree of the options in the items which has been rated by such user. In other words, there is an interdependency in computation of the credibility degree of options and users' trust scores. Thus, we propose an iterative algorithm to compute both the credibility degrees and trust scores. We denote the non-normalized credibility, normalized credibility and trustworthiness at iteration l as $\beta_{li}^{(l)}$, $\rho_{li}^{(l)}$ and $T_r^{(l)}$, respectively. Therefore, equations (3.5), (3.6) and (3.7) can be re-written as:

$$\beta_{li}^{(l+1)} = \sum_{j,r\,:\,r\to lj} \left(T_r^{(l)}\right)^{\alpha} \times d(i,j) \times w_p \tag{3.8}$$

$$\rho_{li}^{(l+1)} = \frac{\beta_{li}^{(l)}}{\sqrt{\sum_{1\leq j\leq n_l} \left(\beta_{lj}^{(l)}\right)^2}} \tag{3.9}$$

$$T_r^{(l+1)} = \sum_{l,i\,:\,r\to li} \sum_{1\leq j\leq n_l} \rho_{li}^{(l)} \times d(i,j) \times w_p \tag{3.10}$$

Algorithm 1 shows our iterative process for computing the credibility degree and trustworthiness values. One can see, the algorithm starts with identical trust scores for all users, $T_r^{(0)} = 1$. Then in each iteration, the algorithm first compute the non-normalized credibility degree β_{li}. After obtaining the normalized credibility degree ρ_{li} for all options, the trustworthiness for all users are updated. The iteration will stop when there is no considerable changes for the credibility degrees.

3.4 Multi-dimensional Reputation

Examples from eBay's multi-categories feedback system and student course evaluation in educational systems suggest that a reputation system needs to consider the correlation among raters' perceptions among multiple categories. A traditional approach is to apply the trust computation method over the ratings of each category, separately. However, since there is an implicit correlation among the ratings of different categories [18], taking into account such correlation can improve the accuracy of the reputation system for obtaining the users' trustworthiness.

In Eq. (2.3) we proposed a aggregation method for single category rating system. In this method, the final reputation of an item is obtained from an aggregate of the credibility values of different options for such item. In order to extend this method to multi-dimensional rating systems, we first aggregate the weights obtained for each user by applying Algorithm 1 over each category. Clearly, we have K weight values for each user when we have K categories in the rating system. Then, we aggregate the weights using simple averaging to obtain

Algorithm 1 Iterative algorithm to compute the credibility and trustworthiness.

```
 1: procedure CredTrustComputation($A, b, \alpha, n_l$)
 2:     Compute $q$ using (3.1)
 3:     for each $1 \leq i, j \leq n_l$ do
 4:         $d(i, j) \leftarrow q^{|i-j|}$
 5:     end for
 6:     $T_r^{(0)} \leftarrow 1$
 7:     $l \leftarrow 0$
 8:     repeat
 9:         for each level $i$ and item $l$ do
10:             Compute $\beta_{li}$ using (3.8)
11:         end for
12:         for each level $i$ and item $l$ do
13:             Compute $\rho_{li}$ using (3.9)
14:         end for
15:         for each use $r$ do
16:             Compute $T_r$ using (3.10)
17:         end for
18:         $l \leftarrow l + 1$
19:     until credibilities have converged
20:     Return $\vec{\rho}$ and $\vec{T}$
21: end procedure
```

the final users' trustworthiness. After that, we employ a weighted averaging method to compute the final reputation of item l in category k, as follows

$$R(\pi_{lk}) = \frac{\sum_{i,r\,:\,r \to lik} i \times (\hat{T}_r)^p}{\sum_{i,r\,:\,r \to lik} (\hat{T}_r)^p} \tag{3.11}$$

where $r \to lik$ denotes the fact that user r has chosen option I_i^l from list Λ_l for category k. $\hat{T}_r$ is the average of weights of user r obtained by applying Algorithm 1 over the ratings of different categories. Moreover, constant $p \geq 1$ is a parameter for controlling the averaging affect.

3.5 Algorithm Complexity

Since the rating matrix is a really sparse matrix, we evaluate the time complexity of our reputation system based on the number of ratings, denoted as L which $L \ll n \times m$ (see Table 4.1 for a similar observation in the MovieLens dataset). The initial part of Algorithm 1, lines 2-7 take a constant time as this part is independent from the number of ratings. The complexity of the iterative part of the algorithm depends on the complexity of credibility, normalized credibility and trust computations which are in $O(L \times n_l)$, $O(m \times n_l)$, and $O(L \times n_l)$, respectively. Since n_l is a constant value, each iteration in the algorithm requires a total $O(L)$ time. Thus, the time complexity of Algorithm 1 is in $O(k \times L)$, where k is the number of iterations.

4 Experiments

In this section, we detail the steps taken to evaluate the robustness and effectiveness of our reputation system in the presence of faults and false data injection attacks.

4.1 Experimental Environment

Although there are a number of real world datasets for evaluating reputation systems such as MovieLens[1] and HetRec 2011 [19], none of them provides a clear ground truth. Thus, we conduct our experiments by both real-world datasets and generating synthetic datasets.

In order to generate our synthetic datasets, we used the statistical parameters of the MovieLens 100k dataset. These parameters are presented in the Table 4.1. In this table, two statistical distribution for the number of votes per movie and number of votes per user for the dataset was determined by using MATLAB distribution fitting tools. We generate our synthetic datasets by using these probability distributions for the number of rates. Moreover, we set both the minimum number of ratings for each user and minimum number of ratings for each movie to 20. The quality of each movie has been uniformly randomly selected from the range [1,5]. In addition, we consider a zero mean Gaussian noise for ratings of each user with different variance values for the users. All ratings are also rounded to be discrete values in the range of [1,5]. For each experiment which is based on synthetic datasets, we perform the algorithms over 100 different synthetically generated datasets, and then results are averaged. The program code has been written in MATLAB R2012b.

Table 4.1: MovieLens 100k dataset statistics.

Parameter	MovieLens 100k
Ratings	100,000
Users	943
Movies	1682
Rating range	discrete, range [1-5]
# of votes per movie	Beta($\alpha = 0.57, \beta = 8.41$)
# of votes per user	Beta($\alpha = 1.32, \beta = 19.50$)

In all experiments, we compare our approach against three other IF techniques proposed for reputation systems. For all parameters of other algorithms used in the experiments, we set the same values as used in the original papers where they were introduced.

The first IF method considered computes the trustworthiness of users based on the distance of their ratings to the current state of the estimated reputations [9]. Two proposed discriminant functions consist of g$(\vec{d}) = \vec{d}^{-1}$ and g$(\vec{d}) = 1 - k_l\vec{d}$, we call them as *dKVD-Reciprocal* and *dKVD-Affine*, respectively. We recently [12, 11] introduced a collusion attack against the *dKVD-Reciprocal* function and showed that an attacker can compromise such function using its pole in the point $d = 0$. Thus, we consider the *dKVD-Affine* function for our comparative experiments as such function is more robust against the attack [12].

[1] In this paper, we used the MovieLens dataset which was supplied by the GroupLens Research Project. http://grouplens.org/datasets/movielens/

Table 4.2: Summary of different IF algorithms.

Name	**Discriminant Function**
dKVD-Reciprocal	$w_i^{l+1} = (\frac{1}{T}\left\Vert\mathbf{x}_i - \mathbf{r}^{l+1}\right\Vert_2^2)^{-1}$
dKVD-Affine	$w_i^{l+1} = 1 - k\frac{1}{T}\left\Vert\mathbf{x}_i - \mathbf{r}^{l+1}\right\Vert_2^2$
Zhou	$w_i^{l+1} = \frac{1}{T}\sum_{i=1}^{T}\left(\frac{x_i^t - \bar{\mathbf{x}}^t}{\sigma_{\mathbf{x}_i}}\right)\left(\frac{r^t - \bar{\mathbf{r}}}{\sigma_r}\right)$
Laureti	$w_i^{l+1} = (\frac{1}{T}\left\Vert\mathbf{x}_i - \mathbf{r}^{l+1}\right\Vert_2^2)^{-\frac{1}{2}}$

The second IF method we consider is a correlation based ranking algorithm proposed by Zhou et al. [10]. In this algorithm, trustworthiness of each user is obtained based on the correlation coefficient between the users ratings and the current estimate of the reputation values. In other words, this method gives credit to users whose ratings correlate well with the reputation values. The authors employed Pearson correlation coefficient [20] between users ratings and the current estimated reputation values. We call this method as *Zhou*.

The third algorithm is the pioneer IF algorithm proposed by Laureti et al. [8] and is an IF algorithm based on a weighted averaging technique similar to the algorithm proposed in [9]. The only difference between these two algorithms is in the discriminant function. The authors in [8] have leveraged discriminant function g($\vec{d}$) = $\vec{d}^{-\beta}$ and $\beta = 0.5$. We call this method as *Laureti*.

Table 4.2 shows a summary of aggregation and discriminant functions for all of the above four different IF methods. We also call our proposed method *PrRTV* and our previous method *BasicRTV*, briefly presented in Section 2.2. We use the Root Mean Square (RMS) error as the accuracy comparison metric in all experiments which is defined as follows:

$$RMS\ Error = \sqrt{\frac{\sum_{j=1}^{m}(r_j - \hat{r_j})^2}{n}} \tag{4.1}$$

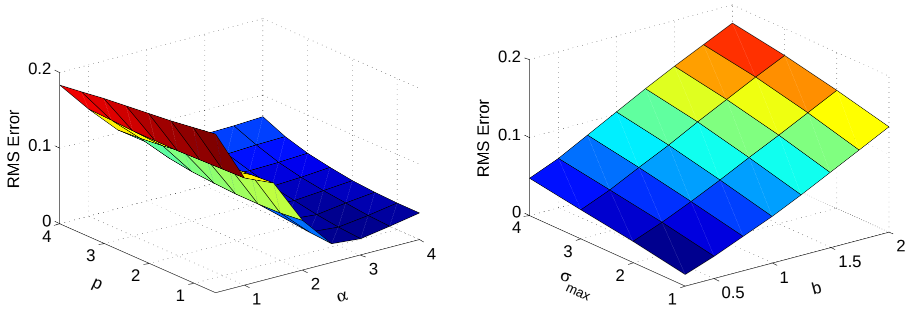
where r_j and $\hat{r_j}$ denote the true value and the estimated value of the reputation for item j, respectively.

4.2 Parameter Sensitivity Analysis

Beyond investigating the robustness of our reputation system, we also measured the sensitivity of its results with respect to the computation parameters: α, p and b. For the experiments in this section, we synthetically generated datasets with parameters similar to the MovieLens dataset. To this end, we uniformly randomly selected the users' standard deviation from the range of $[0, \sigma_{max}]$ with various values for σ_{max}.

Figure 4.1(a) shows the accuracy of our algorithm with different values for parameters α and p where we set $\sigma_{max} = 4$ and $b = 0.5$. One can see in the figure that the highest accuracy levels are obtained when $2 \leq p \leq 3$ and $2 \leq \alpha \leq 3$. Note that the larger value of α provide higher level of discrimination as well as slower convergence in our iterative algorithm. Thus, in our subsequent experiments we choose values $\alpha = 2$ and $p = 2$.

The parameter b defines the level of distance among existing options which our algorithm uses for propagating the credibility among the options. For example, if there are higher levels of uncertainty in the ratings, we consider a higher value for parameter b. Figure 4.1(b) shows the accuracy of our algorithm with various values for parameters b and σ_{max}. As shown in the figure, there is a decreasing trend in the accuracy of our approach as the value of b increases. Thus, we choose value $b = 0.5$ for our subsequent experiments.



(a) $\sigma_{max} = 4$ and $b = 0.5$

(b) $\alpha = 2$ and $p = 2$

Figure 4.1: Accuracy of *PrRTV* with various parameters' values.

4.3 Robustness Against False Ratings

In order to evaluate robustness of our algorithm against false ratings, we conduct experiments based on two types of malicious behaviour proposed in [9] over the MovieLens dataset: *Random Ratings*, and a *Promoting Attack*. For random ratings scenario, we modify the rates of 20% of the users within the original MovieLens dataset by injecting uniformly random rates in the range of [1,5] for those users.

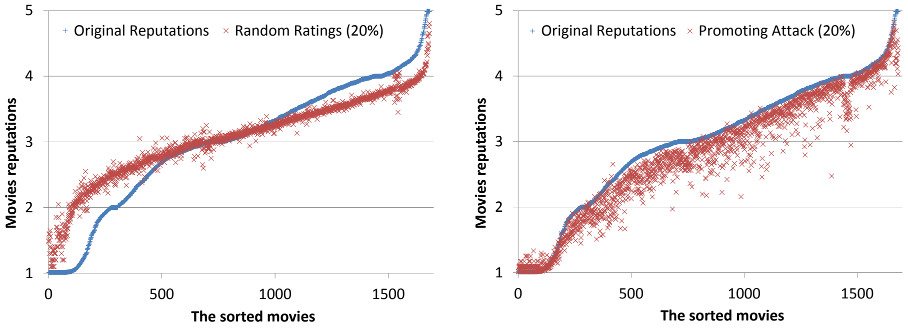
In slandering and promoting attacks, one or more users falsely produce negative and positive ratings, respectively, about one or more items [21]. The attacks can be conducted by either an individual or a coalition of attackers. The attacker may control many users, referred to as malicious users, and conduct either a slandering attack (downgrading the reputation of target items by providing negative ratings) or a promoting attack (boosting the reputation of target items by providing positive ratings) [22]. We evaluate our reputation system against a promotion attack by considering 20% of the users as the malicious users involved in the attack. In this attack scenario, malicious users always rate 1 except for their preferred movie, which they rate 5.

Let r and $\tilde{r}$ be the reputation vectors before and after injecting false ratings in each scenario (random ratings and promoting attack), respectively. In the proposed reputation system, the vectors are the results of Eq. (3.11). Table 4.3 reports the 1-norm difference between these two vectors, $||r - \tilde{r}||_1 = \sum_{j=1}^{m} |r_j - \tilde{r}_j|$ for our algorithm along with other IF algorithms. Clearly, all of the IF algorithms are more robust than *Average*. In addition, the *PrRTV* algorithm provides higher accuracy than other methods for both false rating scenarios. The results can be explained by the fact that the proposed algorithm effectively filters out the contribution of the malicious users.

Moreover, Figure 4.2(a) and 4.2(b) show the perturbations of our reputation system due to the injection of the random ratings and the promoting attack, respectively. As can be seen, the perturbations are slightly changed by using our approach.

Table 4.3: 1-norm absolute error between reputations by injecting false ratings.

	$\|r-\tilde{r}\|_1$				
	Average	dKVD-Affine	Laureti	BasicRTV	PrRTV
Random Ratings	205.32	152.40	171.55	152.75	151.54
Promoting Attack	579.65	378.29	377.72	894.25	368.81



(a) Random Ratings

(b) Promoting Attack

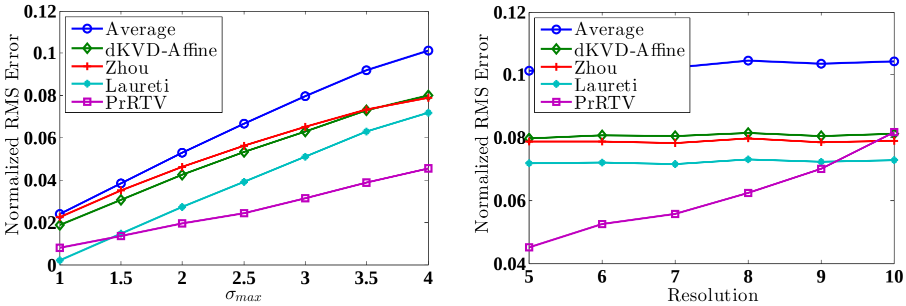
Figure 4.2: Perturbations of *PrRTV* against false ratings.

4.4 Rating Resolutions and Users Variances

Medo and Wakeling [23] reported that the accuracy of existing IF algorithms are highly sensitive to the rating resolution. Thus, we employ their evaluation methodology to investigate the accuracy of *PrRTV* over the low resolution ratings and different variance scales. For the experiments in this section, we create synthetic datasets which their number of users/items and their distribution of ratings are similar to the MovieLens dataset (see Table 4.1). The ratings scale is in the range of $[1, R]$, where R is an integer number and $R \geq 2$. Also, the standard deviation σ_i for user i is randomly selected by a uniform distribution $U[0; \sigma_{max}]$, where σ_{max} is a real value in the range of $[0, R-1]$. We also evaluate a normalized RMS error, $RMS/(R-1)$ (see Eq. (4.1) for RMS Error) for each experiment. In this section, we investigate the accuracy of our reputation system against various values for both rating resolution R and variance scale σ_{max}.

For the first experiment, we set $R = 5$ and vary the value of σ_{max} in the range of $[1, 4]$. By choosing such a range at the worst case, a highest noisy user with $\sigma_i = \sigma_{max} = 4$ could potentially report a very low reputation for an item with a real reputation of 5, and vice versa. Figure 4.3(a) shows the accuracy of the *PrRTV* algorithm along with the accuracy of the other IF algorithms for this experiment. We observe that *PrRTV* is the least sensitive to the increasing error level, maintaining the lowest normalized RMS error.

In order to investigate the effect of changing the ratings' resolution, we set $\sigma_{max} = R - 1$ and vary the value of R in the range of $[5, 10]$, so that the maximum possible users' errors cover the ratings' scale. Figure 4.3(b) shows the accuracy of the algorithms for this experiment. As we can see, although the accuracy of the *PrRTV* algorithm is higher than the accuracy of other IF algorithms, the algorithm provides more sensitivity for the high resolution values. In other words, the accuracy of our reputation system significantly



(a) Variance Changes

(b) Resolution Changes

Figure 4.3: Accuracy with different variances and resolutions.

drops as the ratings resolution increases. The reason of this behaviour is that Eq. (3.11) for computing the final rating scores gives more credibility to the options with higher numerical values, particularly when there is a large distance between lowest and highest options in the ratings scales. We plan to investigate other possible functions for computing the final ratings which provide more robustness for higher resolution rating systems.

4.5 Accuracy Over HetRec 2011 MovieLens Dataset

In this section, we evaluate the performance of our reputation system based on the accuracy of the ranked movies in the *HetRec 2011 MovieLens* dataset [19]. This dataset is an extension of *MovieLens 10M* dataset, published by GroupLeans research group. It links the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb)[2] and Rotten Tomatoes movie critics systems[3]. Thus, we use the top critics ratings from Rotten Tomatoes as the domain experts for evaluating the accuracy of our approach.

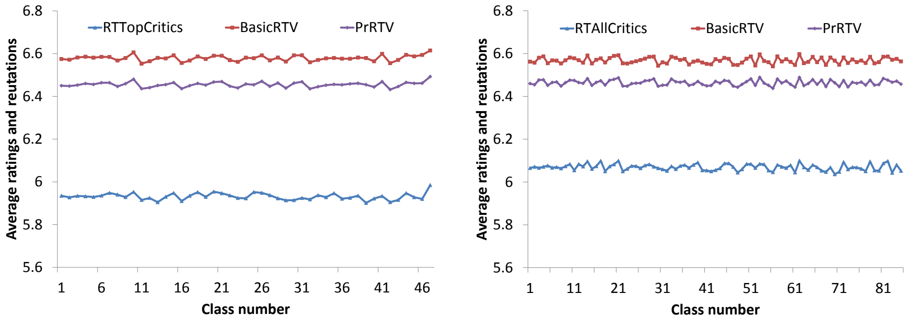
There are 10,109 movies in the HetRec 2011 MovieLens dataset rated by users. The dataset also includes the average ratings of the top and all critics of Rotten Tomatoes for 4645 and 8404 movies, respectively. We consider such average ratings as two ground truth data to evaluate the accuracy of our approach and we call them *RTTopCritics* and *RTAllCritics*, respectively. In order to clearly compare the results of our reputation system with those provided by RTTopCritics and RTAllCritics, we first classify the movies by randomly assigning every 100 movies in a class. We then compute two average values for each class: the average of reputation values given by our algorithm and the average of rating given by RTTopCritics and RTAllCritics. Now, we use such average values to compare the reputations given by our algorithm with the ratings of RTTopCritics and RTAllCritics. Note that this method is employed only for clarifying this comparison over such large number of movies.

Figure 4.4(a) and 4.4(b) illustrate the comparison between the results of our algorithm with the ratings provided by RTTopCritics and RTAllCritics, respectively. Clearly, the figure confirms that the reputation values given by our algorithm is very close to the experts opinions given by RTCritics. Moreover, comparing the results of *PrRTV* with *BasicRTV* shows that the *PrRTV* algo-

[2] http://www.imdb.com/

[3] http://www.rottentomatoes.com/critics/

rithm provide a better accuracy than the *BasicRTV* algorithm as its aggregate ratings are more closer to the ratings provided by Rotten Tomatoes critics. As one can see, our algorithm ranks the movies slightly higher than RTCritics ratings for all classes. This can be explained by the fact that the ratings of our algorithm are based on the scores provided by public users through the MovieLens web site, however, both RTTopCritics and RTAllCritics ratings provided by Rotten Tomatoes critics who tend to rank the movies more critically. This results can confirm the acceptable accuracy of the proposed reputation system over this real-world dataset.



(a) RTTopCritics

(b) RTAllCritics

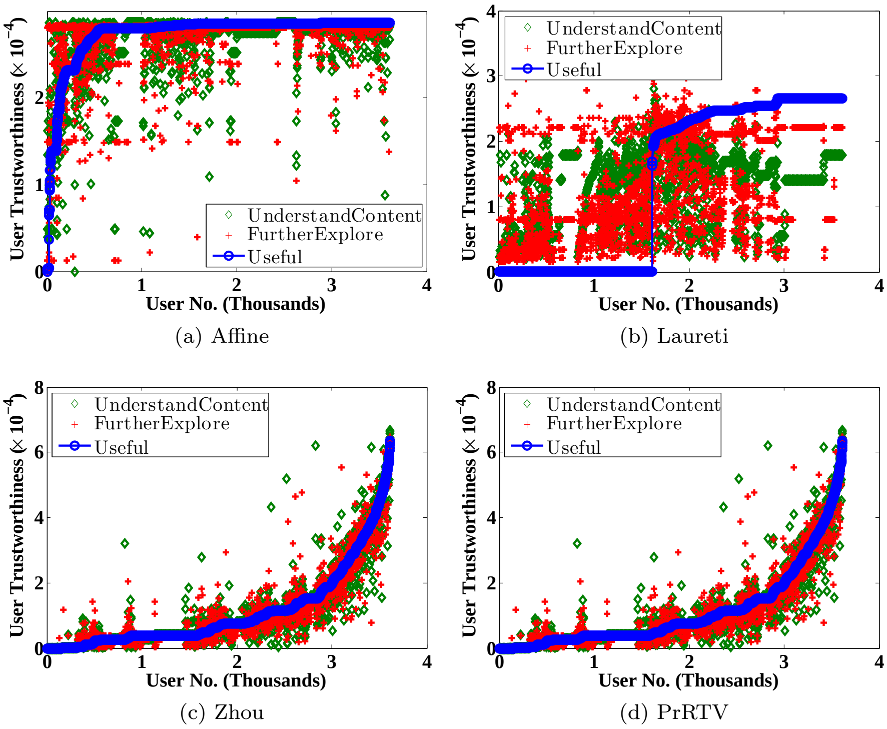
Figure 4.4: Average reputations for movies computed by our algorithms and Rotten Tomatoes movie critics.

4.6 Accuracy Over Student Feedback Dataset

While student evaluations and feedback have significant roles to improve the quality of an education system, they have been criticised for being biased by students' perceptions [24]. Moreover, students are usually asked to rate the courses on multiple categories. Thus, obtaining an overall teaching effectiveness needs to take into account an aggregation of all existing rating dimensions.

In this section, we evaluate the effectiveness of our reputation system using a privately accessed student feedback dataset provided by the Learning and Teaching Unit at UNSW, we call it *CATEI*. The dataset consists of 17,854 ratings provided by 3,910 students (221 staffs and 3,690 non-staffs) for 20 movies in an online course presented in UNSW. In the CATEI dataset, students were asked to rate the movies in the range of [1-5] and for three different categories: *Useful*, *UnderstandContent*, *FurtherExplore*. Moreover, the dataset includes the starting and ending times of the watching of the movie for each rating which allow us to compute the watching duration for each rating. We also set the duration sensitivity, $\beta = 0.2$ for computing the watching time weight of each rating. As we mentioned in Section 3.2, the rating provenance is obtained as the product of staff weight and watching weigh for each rating.

In the first part of the experiments over the CATEI dataset, we apply the IF algorithms over each rating category separately and then investigate the correlation between the obtained users' weights. We expected to observe high correlation among the weights on different categories. We first obtained all the users' weights, then sorted them in an increasing order based on the *Useful* category. Figure 4.5 compares the users' weights among three categories obtained by each IF algorithm. Moreover, Table 4.4 reports the Pearson correlation coefficient [20] among such weight values. One can see in the results that our reputation system provides the highest correlation among the weights for various categories. This can validate the effectiveness of our approach over the CATEI dataset.



(a) Affine

(b) Laureti

(c) Zhou

(d) PrRTV

Figure 4.5: Users' weights obtained by the IF algorithms over three categories.

Table 4.4: Correlation among users' weights obtained by the IF algorithms over three categories (*U*: *Useful*, *UC*:*UnderstandContent*, *FE*:*FurtherExplore*).

	dKVD-Affine	*Laureti*	*Zhou*	*PrRTV*
U and *UC*	0.52	0.42	0.58	0.96
U and *FE*	0.61	0.40	0.61	0.97
UC and *FE*	0.45	0.50	0.63	0.97

In Section 3.4, we proposed the idea of aggregation of users' weights obtained for each category to obtain the final reputation values over multi-dimensional rating datasets. A traditional approach is to separately apply the reputation system over each dimension. In order to investigate the effectiveness of the proposed approach, we evaluated the correlation among the reputation values for various categories over the CATEI dataset for these two methods. To this end, we first applied the IF algorithms over each category and computed the correlation among the obtained reputation vectors for each category. After that, we applied the proposed method in Section 3.4, and computed the correlation among the new reputation vectors. Table 4.5 reports the percentage of increasing such correlation among categories by applying our multi-dimensional reputation method. One can see that our approach improved the average correlation value for all four algorithms. The results also show a significant improvement in the *Zhou* algorithm. This can be explained by some negative correlations obtained by the algorithm when the traditional reputation computation method applied.

Table 4.5: Percentage of increasing correlation among reputations by aggregating the weights obtained through each category (U: *Useful*, UC:*UnderstandContent*, FE:*FurtherExplore*).

	dKVD-Affine	*Laureti*	*Zhou*	*PrRTV*
U and UC	0.70	2.79	2.80	13.90
U and FE	0.03	8.54	72.12	-0.65
UC and FE	-0.26	0.12	0.09	-0.73
Average	**0.16**	**3.81**	**25.00**	**4.17**

4.7 Analysis of Sparsity Pattern

The datasets provided by rating system are usually very sparse. For example, one can see in Table 4.1 that the MovieLens dataset provides an average around 6% rating density. In this section, we evaluate the performance of our approach along with other IF algorithms over the sparse rating datasets. To this end, we define a density factor $0 < \eta \leq 1$, which is the proportion of number of ratings for each user. Clearly, a value of $\eta = 1$ indicates no sparsity pattern.

To conduct experiments in this section, we synthetically generated datasets with various values for density factor, η, in the range of $[0.1, 0.5]$. Accordingly, we first generated a dense rating dataset as the base dataset. Then, we uniformly randomly removed $m \times (1 - \eta)$ ratings for each user to inject the appropriate sparsity pattern.

Let $\mathbf{r}$ and $\tilde{\mathbf{r}}$ be the reputation vectors before and after injecting the sparsity patterns. Table 4.6 shows the 1-norm difference between these two vectors, $\left|\left|\vec{r} - \tilde{\vec{r}}\right|\right|_1 = \sum_{t=1}^{m} |r_t - \tilde{r}_t|$ for the *PrRTV* algorithm along with other IF algorithms. One can see in the table that our algorithm provides more robustness against sparse ratings. Moreover, the experiment results show that, increasing the density factor improves the accuracy of all the IF algorithms. This can be explained by the fact that all of these algorithms are using a kind of collaborative technique among users to estimate the reputation values as well as users trustworthiness; and the density of the ratings has a significant effect in the performance of every collaborative method [25].

4.8 Analysis of Error and Convergence

In this section, we conduct a set of experiments to analyze behaviours of our iterative algorithm in terms of error and convergence. Thus, we investigate two

Table 4.6: 1-norm absolute error between reputation vectors with various density factors in the ratings matrix.

	$\left\Vert \vec{r} - \tilde{\vec{r}} \right\Vert_1$				
	Average	*dKVD-Affine*	*Laureti*	*Zhou*	*PrRTV*
$\eta = 0.1$	169.57	149.23	143.27	130.24	123.73
$\eta = 0.2$	113.42	98.28	95.02	86.65	80.10
$\eta = 0.3$	86.06	73.51	71.95	65.73	60.76
$\eta = 0.4$	68.87	57.82	57.38	52.47	48.25
$\eta = 0.5$	56.47	46.94	47.04	42.96	39.42

types of errors for both users trustworthiness and credibility values computed in each iteration of the proposed algorithm over the MovieLens dataset. For each of trustworthiness and credibility values, we define the maximum error by choosing the worst-case error for all users and items, respectively. Therefore, the maximum errors at iteration l is computed as follows:

$$error_{\rho}^{(l)} = \max_{l_i} \left| \rho_{l_i}^{(\infty)} - \rho_{l_i}^{(l)} \right|$$

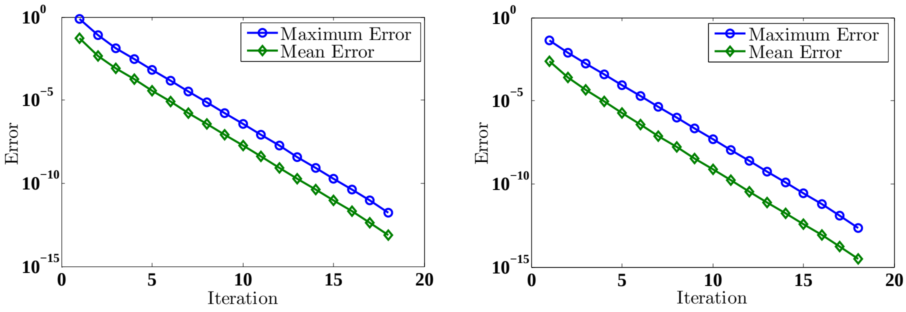
$$error_{T}^{(l)} = \max_{r} \left| {T_r}^{(\infty)} - {T_r}^{(l)} \right|$$

We also define the mean error of credibility and trustworthiness values as follows:

$$error_{\rho}^{(l)} = \frac{1}{m \times n_l} \sum_{l=1}^{m} \sum_{i=1}^{n_l} \left| \rho_{l_i}^{(\infty)} - \rho_{l_i}^{(l)} \right|$$

$$error_{T}^{(l)} = \frac{1}{n} \sum_{r=1}^{n} \left| {T_r}^{(\infty)} - {T_r}^{(l)} \right|$$

Figure 4.6 illustrates how the aforementioned errors decline for both credibility and trustworthiness values. For all experiments, we set convergence threshold with an error $\left\Vert \vec{\rho}^{(l+1)} - \vec{\rho}^{(l)} \right\Vert_2$ less than 10^{-12}. Figure 4.6 shows that the error decreases exponentially in the *PrRTV* algorithm.



(a) Credibility

(b) Trustworthiness

Figure 4.6: Convergence and error of credibility and trust scores over the MovieLens dataset.

5 Related Work

According to several research evidences, as the reliance of the users of online stores on the rating systems to decide on purchasing a product constantly increases, more efforts are put in building up fake rating or reputation scores in order to gain more unfair income [1]. To solve this problem, Mukherjee et.al., [5] proposed a model for spotting fake review groups in online rating systems. The model analyzes feedbacks cast on products in Amazon online market to find collusion groups. They employ FIM [26] algorithm to identify candidate collusion groups and then use 8 indicators to identify colluders. Allahbakhsh et. al., proposed another collusion detection technique based on FIM algorithm in which the clustering techniques are used to detect collusion groups [27].

In a more general setup, collusion detection has been studied in P2P and reputation management systems; good surveys can be found in [28] and [4]. EigenTrust [29] is a well known algorithm proposed to produce collusion free reputation scores; however, authors in [30] demonstrate that it is not robust against collusion. Another series of works [31, 32, 33] use a set of signals and alarms to point to a suspicious behavior. The most famous ranking algorithm of all, the PageRank algorithm [34] was also devised to prevent collusive groups from obtaining undeserved ranks for webpages.

Several papers have proposed IF algorithms for reputation systems [9, 8, 10, 35, 36]. While such IF algorithms provide promising performance for filtering faults and simple cheating attacks, we recently showed that they are vulnerable against sophisticated attacks [11, 12]. Medo and Wakeling [23] investigate the sensitivity of the IF algorithms to rating' resolutions as well as discrete/continuous ratings. Galletti et al. [37] proposed a mathematical framework for modeling the convergence of the IF algorithms. In this paper, we compared the robustness of our approach with some of the existing IF methods.

The method we propose in this paper is different from the existing related work, mainly from its ancestor RTV, from three various aspects. First, the distance between the options is taken into account in this work. Second, reputation scores are in fact multi dimensional, and finally, the provenance of rating scores are considered while giving credit and weight to them.

6 Conclusions

In this paper, we proposed a novel reputation system which utilizes several novel parameters to compute a more dependable and realistic reputation and rating scores. Taking distance between the quality levels into account, considering the provenance of cast rating scores and computing multi-dimensional reputation scores are three main novelties of our proposed reputation calculation algorithm. The experiments conducted on both synthetic and real-world data show the superiority of our model over three well-known iterative filtering algorithms. Since the proposed framework has shown a promising behaviour, we plan to extend the algorithm to propose a distributed reputation system.

Bibliography

[1] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and turf: crowdturfing for fun and profit. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 679–688, 2012.

[2] John Morgan and Jennifer Brown. Reputation in online auctions: The market for trust. *California Management Review*, 49(1):61–81, 2006.

[3] AMY HARMON. Amazon glitch unmasks war of reviewers. In *NY Times (2004, Feb. 14)*.

[4] Yan Lindsay Sun and Yuhong Liu. Security of online reputation systems: The evolution of attacks and defenses. *IEEE Signal Process. Mag.*, 29(2):87–97, 2012.

[5] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 191–200, 2012.

[6] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 939–948. ACM, 2010.

[7] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Seyed-Mehdi-Reza Beheshti, Elisa Bertino, and Norman Foo. Collusion detection in online rating systems. In *Proceedings of the 15th Asia Pacific Web Conference (APWeb 2013)*, pages 196–207, 2013.

[8] P. Laureti, L. Moret, Y.-C. Zhang, and Y.-K. Yu. Information filtering via Iterative Refinement. *EPL (Europhysics Letters)*, 75:1006–1012, September 2006.

[9] Cristobald de Kerchove and Paul Van Dooren. Iterative filtering in reputation systems. *SIAM J. Matrix Anal. Appl.*, 31(4):1812–1834, 2010.

[10] Yan-Bo Zhou, Ting Lei, and Tao Zhou. A robust ranking algorithm to spamming. *EPL (Europhysics Letters)*, 94(4):48002–48007, 2011.

[11] Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. A robust iterative filtering technique for wireless sensor networks in the presence of malicious attacks. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 30. ACM, 2013.

[12] Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. *IEEE Transactions on Dependable and Secure Computing*, 2014. PrePrints.

[13] Mohammad Allahbakhsh and Aleksandar Ignjatovic. An iterative method for calculating robust rating scores. *IEEE Transactions on Parallel and Distributed Systems*, 2014. PrePrints.

[14] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, and Boualem Benatallah. Robust evaluation of products and reviewers in social rating systems. *World Wide Web*, pages 1–37, 2013.

[15] Understanding eBay's new feedback system. `http://www.ebay.com/gds/`, March 2011. [Online; accessed 1-January-2014].

[16] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 22(140):55, 1932.

[17] Xinlei Oscar Wang, Wei Cheng, Prasant Mohapatra, and Tarek F. Abdelzaher. ARTSense: Anonymous reputation and trust in participatory sensing. In *INFOCOM*, pages 2517–2525. IEEE, 2013.

[18] Jiliang Tang, Huiji Gao, and Huan Liu. mTrust: Discerning multi-faceted trust in a connected world. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 93–102, 2012.

[19] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.

[20] Larry Wasserman. *All of statistics : a concise course in statistical inference.* Springer, New York, 2010.

[21] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42(1):1:1–1:31, December 2009.

[22] Yan Sun and Yuhong Liu. Security of online reputation systems: The evolution of attacks and defenses. *Signal Processing Magazine, IEEE*, 29(2):87 –97, March 2012.

[23] Matus Medo and Joseph R. Wakeling. The effect of discrete vs. continuous-valued ratings on reputation and ranking systems. *CoRR*, abs/1001.3745, 2010.

[24] Benjamin Fauth, Jasmin Decristan, Svenja Rieser, Eckhard Klieme, and Gerhard Bttner. Student ratings of teaching quality in primary school: Dimensions and prediction of student outcomes. *Learning and Instruction*, 29(0):1 – 9, 2014.

[25] Zan Huang, Daniel Zeng, and Hsinchun Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22(5):68–78, September 2007.

[26] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, 1994.

[27] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Seyed-Mehdi-Reza Beheshti, Norman Foo, and Elisa Bertino. Representation and querying of unfair evaluations in social rating systems. *Computers & Security*, 41(0):68 – 88, 2014.

[28] Gianluca Ciccarelli and Renato Lo Cigno. Collusion in peer-to-peer systems. *Computer Networks*, 55(15):3517 – 3532, 2011.

[29] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651, 2003.

[30] Qiao Lian, Zheng Zhang, Mao Yang, Ben Y Zhao, Yafei Dai, and Xiaoming Li. An empirical study of collusion behavior in the maze P2P file-sharing system. In *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on*, pages 56–56. IEEE, 2007.

[31] Y. Liu, Y. Yang, and Y.L. Sun. Detection of collusion behaviors in online reputation systems. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 1368–1372. IEEE, 2008.

[32] Yafei Yang, Qinyuan Feng, Yan Lindsay Sun, and Yafei Dai. RepTrap: a novel attack on feedback-based reputation systems. In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, SecureComm '08, pages 8:1–8:11, 2008.

[33] Ya-Fei Yang, Qin-Yuan Feng, Yan Sun, and Ya-Fei Dai. Dishonest behaviors in online rating systems: cyber competition, attack models, and attack generator. *J. Comput. Sci. Technol.*, 24(5):855–867, September 2009.

[34] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond:The Science of Search Engine Rankings.* Princeton University Press, February 2012.

[35] Rong-Hua Li, Jeffrey Xu Yu, Xin Huang, and Hong Cheng. Robust reputation-based ranking on bipartite rating networks. In *SDM*, pages 612–623, 2012.

[36] E. Ayday, Hanseung Lee, and F. Fekri. An iterative algorithm for trust and reputation management. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2051–2055, 28 2009-July 3.

[37] A. Galletti, G. Giunta, and G. Schmid. A mathematical model of collaborative reputation systems. *Int. J. Comput. Math.*, 89(17):2315–2332, November 2012.