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

Mohsen Rezvani[1](✉), Mohammad Allahbakhsh[2], Lorenzo Vigentini[1], Aleksandar Ignjatovic[1], and Sanjay Jha[1]

[1] University of New South Wales, Sydney, Australia
{mrezvani,ignjat,sanjay}@cse.unsw.edu.au, l.vigentini@unsw.edu.au
[2] University of Zabol, Zabol, Iran
allahbakhsh@uoz.ac.ir

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 rating scores for fame or profit. Therefore, both providing 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 unfair ratings. They also ignore distance between options, provenance of information and different dimensions of cast rating scores while computing aggregate rating scores and trustworthiness of raters. In this paper, we propose a robust iterative algorithm which leverages the information in the profile of raters, provenance of information and a prorating function for the distance between options to build more robust and informative rating scores for items as well as trustworthiness of raters. We have implemented and tested our rating method using both 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 unfair ratings and outperforms well-known ranking algorithms.

Keywords: Online rating · Voting · Trust · Provenance · Multi-dimensional

1 Introduction

Nowadays, millions of people generate content or advertise products online. It is very unlikely for a customer to have a personal experience with a product or to know how trustworthiness a seller might be. One of the widely used methods to overcome this problem is to rely on the feedback received from the other users who have had a direct experience with a product or have already bought it. *Online rating systems* collect feedback from users of an online community and,

H. Federrath and D. Gollmann (Eds.): SEC 2015, IFIP AICT 455, pp. 189–203, 2015.
DOI: 10.1007/978-3-319-18467-8_13

based on the feedback, assign a quality score to every product or trustworthiness of a user in the community. The Amazon[1] online market and the eBay[2] are some of the well-known outlets which incorporate an online rating systems.

One of the big issues with the online rating systems is the credibility of the quality ranks that they produce. For various reasons, users might have interest to post unfair feedback, either individually or as an organised, colluding group. If such unfair feedback is taken into account when ranks are computed, the resulting quality ranks are no longer reliable. Many pieces of evidence show that the online rating systems are widely subject to such unfair ratings [10,16]. Some studies propose methods for dealing with this problem which rely on clustering techniques to analyze the behaviour of raters and find the abnormal ones [8,11]. The main problem with such solutions is that the clustering techniques are generally based on solutions to NP-Hard graph problems; thus their performance is severely degraded when the size of an online systems is too large. The other type of solutions to such problems is based on iterative filtering (IF) techniques [4,6,20]. These techniques, while performing better than the simple aggregation techniques, are still vulnerable to sophisticated collusion attacks [13].

We have recently proposed an algorithm [1], *Rating Through Voting (RTV)*, which outperforms the previous IF algorithms in terms of detection and mitigation of unfair behaviour. Although RTV shows a promissing robustness against unfair ratings, it still has limitations that require more investigations.

The first limitation is that in RTV the order of the choices is not important and the distance between the choices is not defined. For example, when a rater chooses the Nominee_1 as the most popular candidate and another rater selects the Nominee_2, it does not make sense to talk about the distance between these two options. However, in a movie rating system, if one of the raters chooses 4 star rating of a movie and another chooses a 3 star rating then a distance between there ratings is well defined and might be important for rating methods. The distance between choices is not taken into account in the 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[3]. For a reputation to be more credible, it is necessary that the reputation system aggregates the scores received for all different aspects to build the final reputation score. This is another limitation of the RTV algorithms.

Finally, the provenance of a rating score is another piece of information that is ignored in the RTV algorithm. The contextual information around a cast rating score can give the system useful hints to adjust its weight. The profile of the rater, the time a feedback has been cast, etc., are examples of contextual meta data that can be taken into account in the computation of the ranks.

[1] http://www.amazon.com/
[2] http://www.ebay.com/
[3] http://www.ebay.com/gds/

In this paper we propose a novel reputation system which is based on the RTV algorithm[4]. The proposed method takes into account the distance between options to fairly propagate credibility among options. We also, consider the different dimensions of the cast rating scores and utilize them in order to build more realistic and credible reputation aggregation. 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 reliable scores. We have assessed the effectiveness of our approach using both synthetic and three real-world datasets. The evaluation results show superiority of our method over three well-known algorithms in the area, including RTV.

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 system a set of n users cast ratings for m items. Each user rates several items (but not necessarily all) and each item might be rated from K different perspectives. We represent the set of ratings by a three dimensional array $A_{n\times m\times K}$ 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} perspective. We suppose that rating scores are selected from a discrete set of numbers each of which represent a quality level, for example 1-star to 5-stars.

2.2 Rating Through Voting

The RTV algorithm [1] reduces the problem of rating to a voting task. In the algorithm, when a rater chooses a quality level, say 4-stars, to represent quality of a product, one can say that the rater believes that 4-stars represents the quality of the product better than the other options; thus, in a sense, he has voted for it out of the list of 1-star to 5-stars options.

RTV assigns a credibility degree to each quality level in order to show how credible this quality level is for representing the real quality of the item. Thereafter, it aggregates the credibility of all quality levels a users has voted for to build the users' trustworthiness. Assume that for each item l, there is a list of options $\Lambda_l = \{I_1^l, \ldots, I_{n_l}^l\}$ 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:

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

[4] An extended version of this paper has been published as a technical report in [12].

where $r \rightarrow li$ denotes that user r has chosen option I_i^l from list Λ_l. $\alpha \geq 1$ is a parameter which can be used to tune the algorithm for a particular task. T_r is the trustworthiness of user r which is obtained as:

$$T_r = \sum_{l,i\,:\,r \rightarrow li} \rho_{li} \tag{2}$$

Equations (1) and (2) show that there is an interdependency between the credibility and trustworthiness. RTV leverages such interdependency through an iterative procedure. Given the credibility degrees obtained by such iterative algorithm, the aggregate rating score of item l, denoted as $R(\pi_l)$, is obtained as:

$$R(\pi_l) = \sum_{1 \leq i \leq n_l} \frac{i \times \rho_{li}^p}{\sum_{1 \leq j \leq n_l} \rho_{lj}^p} \tag{3}$$

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

3 Reputation Aggregation System

In this section, we extend RTV by taking into account the rating provenance as well as credibility propagation 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 distance 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 consecutive options. 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 is 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

$$\begin{aligned} q + q^2 + \cdots + q^{n_l - j} + q + q^2 + \cdots + q^{j-1} = b \;\Leftrightarrow \\ q\left(\frac{1 - q^{n_l - j}}{1 - q}\right) + q\left(\frac{1 - q^{j-1}}{1 - q}\right) = b \;\Leftrightarrow \\ 2 - q^{j-1} - q^{n_l - j} = b\frac{1 - q}{q} \end{aligned} \tag{4}$$

Note that since $0 < q < 1$, Eq. (4) has only one real solution for each positive value of b.

3.2 Provenance-Aware Credibility Propagation

Given the distance function $d(i, j)$ for computing the numerical distance between options i and j, we 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 chosen such particular level, but also from all raters who rated such item with proportion to 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. (1) for computing the non-normalized credibility degree of quality level li as follows:

$$\beta_{li} = \sum_{j,r\,:\,r \rightarrow lj} (T_r)^{\alpha} d(i, j) \tag{5}$$

As we mentioned 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. Since the rating provenance provides informative data about the quality of ratings, a reputation system needs to take into account these data in the its computations. In this paper, we propose a 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 note that the approach can be easily adapted for other contextual attributes. This provenance model is based on the approach from [17], originally 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 weights from these functions using the simple product of the weights to obtain the provenance weight. Such provenance weight is used to assess the credibility level as well as users' trustworthiness.

In the student feedback system, users are asked to rate the movies in an online course. In this system, each user has an status which indicates whether such user is staff or non-staff. Moreover, the system provides for each rating the time spent for watching the movie. We utilize both the staff status and watching duration as two contextual attributes to model the rating provenance. To this end, we consider a somewhat 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 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:

$$w_t = e^{-|\min\{T_r, T_v\} - T_v| \times \beta} \tag{6}$$

where $0 \leq \beta \leq 1$ is the *duration sensitivity* parameter which controls the watching time weight. Note that Eq. (6) 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:

$$w_p = w_s \times w_t \tag{7}$$

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. (5) as follows:

$$\beta_{li} = \sum_{j, r\,:\, r \to lj} (T_r)^{\alpha} \times d(i,j) \times w_p \tag{8}$$

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

$$\rho_{li} = \frac{\beta_{li}}{\sqrt{\sum_{1 \leq j \leq n_l} (\beta_{lj})^2}} \tag{9}$$

The trustworthiness of 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

$$T_r = \sum_{l, i\,:\, r \to li} \sum_{1 \leq j \leq n_l} \rho_{li} \times d(i,j) \times w_p \tag{10}$$

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 (8), (9) and (10), we have interdependent definitions for credibility degree and trustworthiness. Clearly, the credibility degree of a quality level in for item depends on the trustworthiness of users who rated such item. on the other hand, the trustworthiness of a user depends on the credibility of the quality levels of the items which have been rated by such user. Thus, we propose an iterative algorithm to compute both the credibility degrees and trust scores simultaneously in a single recursive procedure. 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 which are computed from the values obtained in the previous iteration of the algorithm.

Algorithm 1 shows our iterative process for computing the credibility and trustworthiness values. One can see that the algorithm starts with identical trust scores for all users, $T_r^{(0)} = 1$. In each iteration, it 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.

Algorithm 1. Iterative algorithm to compute the credibility and trustworthiness.

procedure CredTrustComputation(A, b, α, n_l)
  Compute q using (4)
  $d(i,j) \leftarrow q^{|i-j|}$ for each $1 \leq i, j \leq n_l$
  $T_r^{(0)} \leftarrow 1$
  $l \leftarrow 0$
  repeat
    Compute β_{li} using (5) for each level i and item l
    Compute ρ_{li} using (9) for each level i and item l
    Compute T_r using (10) for each each use r
    $l \leftarrow l + 1$
  until credibilities have converged
  Return $\boldsymbol{\rho}$ and $\boldsymbol{T}$
end procedure

3.4 Multi-dimensional Reputation

As we discussed, a reputation system needs to consider the correlation among raters' perceptions among multiple categories. The eBay's feedback system and student course evaluation in educational systems are two examples of rating systems with multiple categories. A traditional approach is to apply the computations over the ratings of each category, separately. However, the correlation among ratings in various categories can help a reputation system to accurately assess the quality of ratings [15].

In Eq. (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 perform Algorithm 1 over each category to obtain K weights for each user (Note that we have K dimensions in the ratings). Then, we aggregate the weights using simple averaging to obtain the final users' trustworthiness. Thereafter, we employ a weighted averaging method to compute the final reputation of item l in category k, as follows

$$R(\pi_{lk}) = \frac{\sum_{i,r\,:\,r \to lik} i \times (\hat{T}_r)^p}{\sum_{i,r\,:\,r \to lik} (\hat{T}_r)^p} \tag{11}$$

where $r \rightarrow lik$ denotes that user r chose 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.

4 Experiments

In this section, we detail the steps taken to evaluate the robustness and effectiveness of our approach in the presence of faults and unfair rating attacks.

4.1 Experimental Environment

Although there are a number of real world datasets for evaluating reputation systems such as MovieLens[5] and HetRec 2011[6], none of them provides a clear ground truth. Thus, we conduct our experiments by both real-world datasets and generating synthetic datasets.

We generate the synthetic datasets by using statistical parameters of the MovieLens 100k dataset, as shown in Table 1. 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]. We conducted parameter analysis experiments to find the values of parameters α, p and b. The results of these experiments are reported in [12] and consequently we choose $\alpha = 2$, $p = 2$ and $b = 0.5$ for our subsequent experiments.

Table 1. MovieLens 100k dataset statistics

Parameter	MovieLens 100k
Ratings	100,000
Users	943
Movies	1682
# 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. Table 2 shows a summary of discriminant functions for these IF methods. We also call our new method *PrRTV* and the previous one *BasicRTV*, briefly presented in Section 2.2.

4.2 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 [4] over

[5] http://grouplens.org/datasets/movielens/
[6] http://grouplens.org/datasets/hetrec-2011/

Table 2. Summary of different IF algorithms

Name	Discriminant Function
dKVD-Affine [4]	$w_i^{l+1} = 1 - k\frac{1}{T}\left\|\mathbf{x}_i - \mathbf{r}^{l+1}\right\|_2^2$
Zhou [20]	$w_i^{l+1} = \frac{1}{T}\sum_{i=1}^{T}\left(\frac{x_i^t - \bar{\mathbf{x}}^t}{\sigma_{\mathbf{x}_i}}\right)\left(\frac{r^t - \bar{\mathbf{r}}}{\sigma_r}\right)$
Laureti [6]	$w_i^{l+1} = (\frac{1}{T}\left\|\mathbf{x}_i - \mathbf{r}^{l+1}\right\|_2^2)^{-\frac{1}{2}}$

the MovieLens dataset: *Random Ratings*, and a *Promoting Attack*. For random ratings, 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 [2]. The attacks can be conducted by either an individual or a coalition of attackers. We evaluate our approach against a promotion attack by considering 20% of the users as the malicious users involved in the attack. In this attack, malicious users always rate 1 except for their preferred movie, which they rate 5.

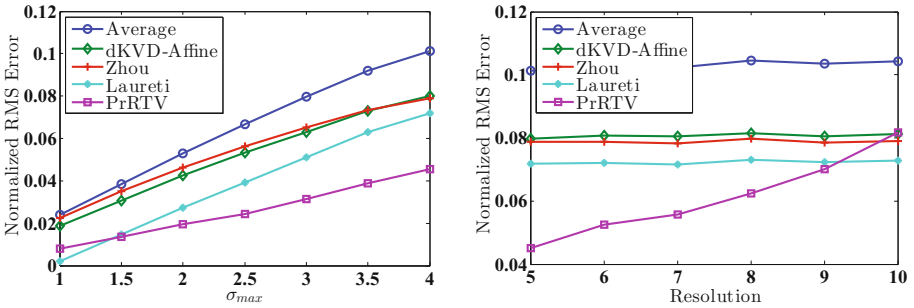
Let r and $\tilde{r}$ be the reputation vectors before and after injecting false ratings in each scenario (random ratings and promoting attack), respectively. In the proposed reputation system, the vectors are the results of Eq. (11). Table 3 reports the 1-norm difference between these two vectors, $||r - \tilde{r}||_1 = \sum_{j=1}^{m} |r_j - \tilde{r}_j|$ 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.

Table 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

4.3 Rating Resolutions and Users Variances

In this section, we investigate the accuracy of *PrRTV* over the low resolution ratings and different variance scales using synthetic datasets. 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, $RMSE/(R-1)$ (see [12] 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}.



(a) Variance Changes **(b)** Resolution Changes

Fig. 1. Accuracy with different variances and resolutions

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. Fig. 1a 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.

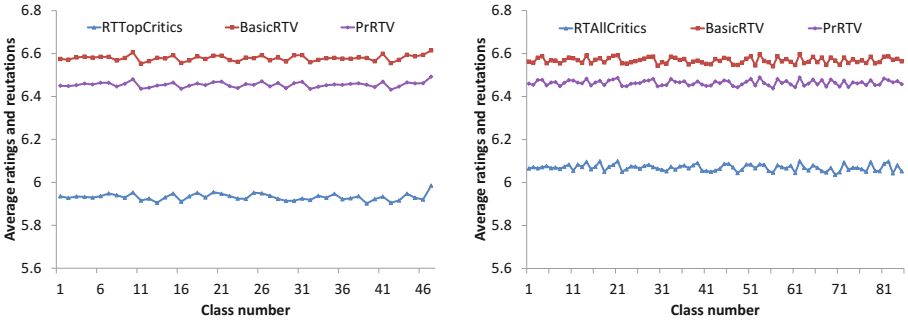
In order to investigate the effect of changing the ratings' resolution, we set $\sigma_{max} = R-1$ and vary the value of R in the range of $[5, 10]$, so that the maximum possible users' errors cover the ratings' scale. Fig. 1b shows the accuracy of the algorithms for this experiment. As we can see, 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 drops as the ratings resolution increases. The reason of this behaviour is that Eq. (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 extend our reputation aggregation method to provide more robustness for high resolution rating systems.

4.4 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. This dataset links the movies in the MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb)[7] and Rotten Tomatoes movie critics systems[8]. 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

[7] http://www.imdb.com/
[8] http://www.rottentomatoes.com/critics/



(a) RTTopCritics **(b)** RTAllCritics

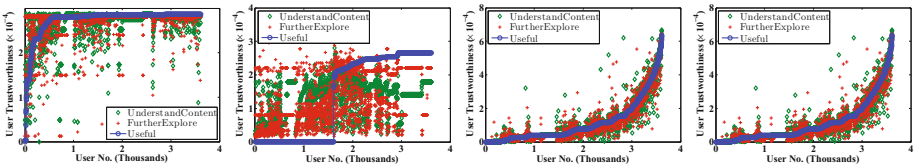
Fig. 2. Average reputations obtained by our algorithms and RTCritics

Rotten Tomatoes for 4645 and 8404 movies, respectively. We consider such average ratings as two ground truth 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.

Fig. 2a and 2b illustrate the comparison between the results of our algorithm with the ratings provided by RTTopCritics and RTAllCritics, respectively. The results confirm 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* algorithm 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.

4.5 Accuracy Over Student Feedback Dataset

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, called *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



(a) Affine **(b)** Laureti **(c)** Zhou **(d)** PrRTV

Fig. 3. Users' weights obtained by the IF algorithms over three categories

Table 4. Correlation among users' weights over three categories

	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

rate the movies in the range of [1-5] and for three different categories: *Useful* (U), *UnderstandContent* (UC), *FurtherExplore* (FE). 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. Fig. 3 compares the users' weights among three categories obtained by each IF algorithm. Moreover, Table 4 reports the Pearson correlation coefficient 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.

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 evaluate the correlation among the reputation values for various categories over the CATEI dataset for these two methods. To this end, we first perform the IF algorithms over each category and compute the correlation among the obtained reputation vectors for each category. After that, we perform the proposed method in Section 3.4, and compute the correlation among the new reputation vectors. Table 5 reports the percentage of increasing such correla-

tion among categories by performing 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 using the traditional reputation computation method.

Table 5. Percentage of increasing correlation among reputations by aggregating the weights obtained through each category

	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**

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 [16]. To solve this problem, Mukherjee et al., [11] 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. In a more general setup, detection of unfair ratings has been studied in P2P and reputation management systems; good surveys can be found in [14]. EigenTrust [3] is a well known algorithm as a robust trust computation system. However, Lian et al. [7] demonstrate that it is not robust against collusion. Another series of works [9,18,19] use a set of signals and alarms to point to a suspicious behavior. The most famous ranking algorithm of all, the PageRank algorithm [5] was also devised to prevent collusive groups from obtaining undeserved ranks for webpages.

Several papers have proposed IF algorithms for reputation systems [4,6,20]. While such IF algorithms provide promising performance for filtering faults and simple cheating attacks, we recently showed that they are vulnerable against sophisticated attacks [13]. 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. Finally, the provenance of rating scores are considered while giving credit and weight to them. To the best of our knowledge, no existing work considers all of these issues in reputation systems.

6 Conclusions

In this paper, we proposed a novel reputation system which leverages the distance between the quality levels, provenance of cast rating scores and multi-dimensional reputation scores to address the problem of robust reputation aggregation. 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.

References

1. Allahbakhsh, M., Ignjatovic, A.: An iterative method for calculating robust rating scores. IEEE Transactions on Parallel and Distributed Systems **26**(2), 340–350 (2015)
2. Hoffman, K., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. **42**(1), 1:1–1:31 (2009)
3. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web, pp. 640–651 (2003)
4. de Kerchove, C., Van Dooren, P.: Iterative filtering in reputation systems. SIAM J. Matrix Anal. Appl. **31**(4), 1812–1834 (2010)
5. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, February 2012
6. Laureti, P., Moret, L., Zhang, Y.C., Yu, Y.K.: Information filtering via Iterative Refinement. EPL (Europhysics Letters) **75**, 1006–1012 (2006)
7. Lian, Q., Zhang, Z., Yang, M., Zhao, B.Y., Dai, Y., Li, X.: An empirical study of collusion behavior in the maze P2P file-sharing system. In: Proceedings of the 27th IEEE International Conference on Distributed Computing Systems. ICDCS 2007, pp. 56–56 (2007)
8. Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., Lauw, H.W.: Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 939–948. ACM (2010)
9. Liu, Y., Yang, Y., Sun, Y.: Detection of collusion behaviors in online reputation systems. In: 2008 42nd Asilomar Conference on Signals, Systems and Computers, pp. 1368–1372. IEEE (2008)
10. Morgan, J., Brown, J.: Reputation in online auctions: The market for trust. California Management Review **49**(1), 61–81 (2006)
11. Mukherjee, A., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st International Conference on World Wide Web. WWW 2012, pp. 191–200 (2012)
12. Rezvani, M., Allahbakhsh, M., Ignjatovic, A., Jha, S.: An iterative algorithm for reputation aggregation in multi-dimensional and multinomial rating systems. Tech. Rep. UNSW-CSE-TR-201502, January 2015
13. Rezvani, M., Ignjatovic, A., Bertino, E., Jha, S.: Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. IEEE Transactions on Dependable and Secure Computing **12**(1), 98–110 (2015)
14. Sun, Y.L., Liu, Y.: Security of online reputation systems: The evolution of attacks and defenses. IEEE Signal Process. Mag. **29**(2), 87–97 (2012)

15. Tang, J., Gao, H., Liu, H.: mTrust: Discerning multi-faceted trust in a connected world. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining. WSDM 2012, pp. 93–102 (2012)
16. Wang, G., Wilson, C., Zhao, X., Zhu, Y., Mohanlal, M., Zheng, H., Zhao, B.Y.: Serf and turf: crowdturfing for fun and profit. In: Proceedings of the 21st International Conference on World Wide Web. WWW 2012, pp. 679–688 (2012)
17. Wang, X.O., Cheng, W., Mohapatra, P., Abdelzaher, T.F.: ARTSense: anonymous reputation and trust in participatory sensing. In: INFOCOM, pp. 2517–2525. IEEE (2013)
18. Yang, Y.F., Feng, Q.Y., Sun, Y., Dai, Y.F.: Dishonest behaviors in online rating systems: cyber competition, attack models, and attack generator. J. Comput. Sci. Technol. **24**(5), 855–867 (2009)
19. Yang, Y., Feng, Q., Sun, Y.L., Dai, Y.: RepTrap: a novel attack on feedback-based reputation systems. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks. SecureComm 2008, pp. 8:1–8:11 (2008)
20. Zhou, Y.B., Lei, T., Zhou, T.: A robust ranking algorithm to spamming. EPL (Europhysics Letters) **94**(4), 48002–48007 (2011)