CERIAS Tech Report 2017-2

A Provenance-Aware Multi-Dimensional Reputation System For Online Rating Systems

by Mohsen Rezvani, Aleksandar Ignjattovic, Elisa Bertino
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

A PROVENANCE-AWARE MULTI-DIMENSIONAL REPUTATION SYSTEM FOR ONLINE RATING SYSTEMS

MOHSEN REZVANI, ALEKSANDAR IGNJATOVIC, AND ELISA BERTINO

Abstract. Online rating systems are widely accepted as means for quality assessment on the web and users increasingly rely on these systems when deciding to purchase an item online. This makes such rating systems frequent targets of attempted manipulation by posting unfair rating scores. Therefore, providing useful, realistic rating scores as well as detecting unfair behavior are both 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 distances between options, the provenance of information and different dimensions of cast rating scores while computing aggregate rating scores and trustworthiness of users. In this paper, we propose a robust iterative algorithm which leverages information in the profile of users and provenance of information and which takes into account the distance between options to provide both more robust and informative rating scores for items and trustworthiness of users. We also prove convergence of iterative ranking algorithms under very general assumptions which are satisfied by the algorithm proposed in this paper. We have implemented and tested our rating method using both simulated data as well as four real-world datasets from various applications of reputation systems. The experimental results demonstrate that our model provides realistic rating scores even in the presence of a massive amount of unfair ratings and outperforms the well-known ranking algorithms.

reputation system, rating provenance, iterative algorithm, online rating

1. Introduction

Nowadays, the quantity of products and contents being advertised or published on the web is so tremendous that it is impossible to assess their quality or to assess trustworthiness of sellers of the advertised products or trustworthiness of sources of content based on one's personal experience. One of the widely used methods to overcome this problem is relying on the feedback received from other consumers who have had experience of buying a product from a particular source. Such a method is based on the use of *online rating systems* which collect and aggregate feedback from all members or visitors of an online community and, based on these opinions, assign both a quality level score to every product as well as a trust score to every source in the community. The movie rating system IMDb[1], Amazon[2] and eBay[3] online markets all incorporate such online rating systems.

One of the major issues with online rating systems is the credibility of the quality ranks that they produce. Such quality ranks are produced mainly based on

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

the feedback received from participants in the forms of either textual or numeric assessments. Users who have posted such feedback might have different levels of expertise and experience. For various reasons, users sometimes might also have vested interest to post unfair feedback, either individually or as an organized colluding group. Unfair feedback is 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 special personal or group interest. 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 online rating systems are widely subject to such unfair ratings [1, 2, 3]. For example, Xu et al. in [1] investigate the business model and market size of the unfair ratings and reputation manipulation on one of the most well-known online marketing systems. They detected more than 11,000 online sellers posting around 219,165 fake purchases which led to revenue of about $46,438.

Past research has developed methods for dealing with this problem which rely on clustering techniques to analyze the behavior of users and find the abnormal ones [4, 5]. The main problem with such solutions is that the clustering techniques are generally based on solutions to NP-Hard graph problems; thus, their performance degrades severely when the size of online systems is substantial. Other types of solutions to such problems are based on iterative filtering (IF) techniques [6, 7, 8]. A recently proposed algorithm [9], *Rating Through Voting (RTV)* outperforms the previous IF algorithms in terms of detection and mitigation of unfair behavior. This algorithm tries to iteratively find the community sentiment and use it to simultaneously assess quality of products and trustworthiness of users. Although RTV shows a promising robustness against unfair ratings, it still has limitations that require more investigation.

The first limitation is that in RTV the order of the choices is not taken into account and the distance between the choices is not defined. For example, when a user chooses Nominee_1 as the most popular candidate and another user selects Nominee_2, it does not make sense to talk about the distance between these two options. However, for example in a movie rating system, these choices possess a natural ordering and, if one of the users assigns a 4-star rating to a movie and another assigns a 3-star rating, then a distance between there ratings is well defined and might be an important piece of information which should be taken into account for a more reliable assessment.

Moreover, in a rating system, users 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 from four different aspects[4]. 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. The original RTV algorithm does not support such "multi-dimensional" assessments.

Finally, the provenance of a rating score is another piece of information ignored by 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 user, 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.

[4]http://www.ebay.com/gds/

In this paper we propose a novel reputation system which extends the RTV algorithm. The proposed method takes into account the distance between options to fairly propagate credibility among options. The method also considers multiple dimensions of the cast rating scores and utilizes them to achieve a more realistic and credible reputation aggregation. Our method takes advantage of the provenance of the cast feedback when calculating reputation and rating scores and consequently produces more informative and reliable scores. We prove the convergence of iterative ranking algorithms under very general assumptions which are satisfied both by the previous RTV ranking algorithms as well as by algorithms presented in this paper.

We have assessed the effectiveness of our approach using both synthetic and three real-world datasets. These 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. In Section 4, we present the proof of convergence of iterative reputation algorithms under rather general assumptions. Section 5 describes our experimental results. Section 6 discusses the related work. Finally, Section 7 outlines a few conclusions.

2. Preliminaries and Problem Statement

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 is rated by several users; moreover, each item might be rated with respect to K different aspects. An example of such a multi dimensional rating is the feedback from all students at a school, evaluating courses that each student has taken with respect to: quality of lectures, quality of the course reading material, quality of the feedback provided on student's homework, appropriateness of the assessment method, etc. 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 item j with respect to the k^{th} aspect. 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-star ratings.

2.2. **Rating through Voting.** The RTV algorithm [9] reduces the problem of rating to a voting task. In the algorithm, if a user chooses a quality level, say 4-stars, to represent the quality of a product, then one can say that the user believes that a 4-star rating 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-star options.

Thus, an item l has an associated list Λ_l of n_l quality levels to choose from. RTV assigns a credibility degree to each such quality level indicating how credible this quality level is for representing the real quality of the item, based on the quality level choices provided by users. Thereafter, it aggregates the credibility of all quality levels a users has voted for in order to build the users' trustworthiness. To briefly explain the gist of the RTV method, assume that for each item l, there is a list of quality level options $\Lambda_l = \{l_1, \ldots, l_{n_l}\}$ and each user can choose at most one option for each item. We define the credibility degree of a quality level i on a

list Λ_l, denoted by ρ_{li} as follows:

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

where $r \to li$ denotes that user r has chosen option i 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:

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

Equations (1) and (2) show that there is an interdependency between the credibility of levels ρ_{li} and trustworthiness of users T_r: once the users cast their votes about the quality of items, then the users can be judged how compliant they are with the prevailing community sentiment about these items; the level of such compliance represents the trustworthiness of a particular voter. Thus, in a sense, the prevailing community sentiment is used as a proxy for the "gold standard", i.e., for the "true" value of items. RTV leverages such interdependency and finds a solution for both all of ρ_{li} and all of T_r satisfying these two equations by an iterative procedure which is provably convergent. Given the credibility degrees obtained by such an iterative algorithm, RTV obtains the aggregate rating score of item l, denoted as R_l, as:

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

where $p \geq 1$ is a parameter. One of the contributions of this paper is a more realistic method of determining such aggregate scores.

2.3. **Problem Statement.** The following four practical issues are not addressed by existing reputation systems, which we aim to resolve in our proposed approach:

(1) *Distance dependent credibility propagation:* According to Eq. (1), if a user r chooses option li for an item l, this contributes towards the credibility of only this option. However, it would be desirable that such a choice also contributes to credibilities of nearby options lj as well, proportional to the proximity of these options to option li. The intuition behind such credibility propagation is that there is always a degree of uncertainty associated with each individual rating. So, if a user chooses a 4-star rating of a movie, while this should primarily increase the credibility of the 4-star rating for that movie, it should also somewhat increase nearby ratings of 5 and 3 stars, much more so than the ratings of 2 stars or 1 star only. This indicates that, in cases when there is a natural notion of distance between options, a reputation system should consider such partial credibility propagation in a neighborhood of each particular choice. Of course, in traditional voting systems such as a parliament voting system, there is no plausible notion of distance between candidates; however, ranking systems by their very nature allow for such distance metrics.

(2) *Multi-dimensional rating systems:* As we have mentioned, in a rating system, users may rate the items with respect to different aspects. In another such example, in eBay's detailed seller rating system, buyers are asked

about different aspects of the transaction and they can rate sellers from four aspects: *Item as described*, *Communication*, *Shipping time*, and *Shipping charges* [10]. Clearly, a reputation system must aggregate the ratings of these categories to assess the overall trustworthiness of users.

(3) *Provenance:* While the previous requirements are mostly based on similarities and confluence of ratings, a reputation system must take into account the contextual information about the origins of such ratings. We indicate such information as the *provenance* of a rating which characterizes the origin of that rating. For example, in one of our datasets, users rate short educational videos; users can be either staff or students; also, some users watch only a part of the video. Thus, the ratings have different provenance; they come from users who have different attributes, be this their competence or their level of familiarity with the videos they rate, and such attributes clearly have an impact on reliability of their rankings.

In this paper, our goal is to extend reputation systems by Allahbakhsh et al. [9] to take into account the above features of individual ratings.

3. Reputation Aggregation System

In this section, we extend RTV by taking into account the rating provenance as well as the credibility propagation in a multi-dimensional rating system. To this end, we first define an impact function to take into account the distance between quality levels. We then leverage such distance dependent impact to extend our basic equations for computing credibility levels and users' trustworthiness. We also define the concept of rating provenance and extend our computations to consider such provenance. Finally, we propose a method to obtain the final reputation values in a multi-dimensional rating system.

3.1. **Distance between ratings.** In most of social rating systems, such as eBay's 5-star feedback system, there is a numerical distance between rating options. In order to take into account such distance in our reputation propagation method, to measure the impact of choosing an option j on credibilities of options other than j one can use any decreasing function $g(x)$ of the distance between options; such function $g(x)$ should be defined on the set of non-negative reals and should satisfy $g(0) = 1$. We then set $\mathrm{im}(i, j) = g(\mathrm{dist}(i, j))$, where $\mathrm{dist}(i, j)$ is any distance metric[5]. In our experiments reported here the impact of the distance of two options i and j is given by the quantity $\mathrm{im}(i, j) = q^{|i-j|}$, where q is a *decay factor*, $0 < q < 1$. Figure 1 shows how the impact value $\mathrm{im}(i, j)$ of a choice of an option j on the credibility of another option i decreases exponentially as the distance between the two options increases[6]. We assume that there is a limited range for the ratings in the rating system. We require that the sum of impacts of choosing level j on all options i, $i \neq j$, must be a constant value equal for all users.The impact of a choice of rating j to the very same choice j we take to be equal to 1, and we denote the sum of impacts on all other choices as b and call it the *propagation parameter*. The propagation parameter is a non-negative value which controls how much of the trustworthiness of each user propagates among options in accordance with their

[5]Since every distance function satisfies $\mathrm{dist}(i, j) = \mathrm{dist}(j, i)$, we also have $\mathrm{im}(i, j) = \mathrm{im}(j, i)$.

[6]In future work, for multidimensional rating systems we will investigate the option that, rather than being computed separately for each dimension, the value of $\mathrm{im}(\vec{i}, \vec{j})$ depends on the Euclidean distance between choices $\vec{i}$ and $\vec{j}$.

proximity to the chosen option. As we will discuss later, if the degree of uncertainty of users is small, b should be chosen small; if the users have higher degree of uncertainty then, accordingly, the impact of their choices on nearby options should be larger, but smaller than the number of options n_l, i.e., $b \leq n_l$. The extreme value $b = 0$ corresponds to perfect certainty of the user regarding his chosen level; the extreme value $b = n_l$, on the other hand, corresponds to a complete uncertainty regarding his choice, because it gives the same impact of 1 to all of n_l levels[7]. Summing the propagation on all options not equal to the chosen option j we get

$$q + q^2 + \cdots + q^{n_l - j} + q + q^2 + \cdots + q^{j-1} = b \;\Leftrightarrow$$

$$q(1 + q + \cdots + q^{n_l - j - 1}) + q(1 + q + \cdots + q^{j-2}) - b = \Leftrightarrow \tag{4}$$

$$q\left(\frac{1 - q^{n_l - j}}{1 - q}\right) + q\left(\frac{1 - q^{j-1}}{1 - q}\right) - b = 0 \;\Leftrightarrow$$

$$q^j + q^{n_l + 1 - j} - (2 + b)q + b = 0 \tag{5}$$

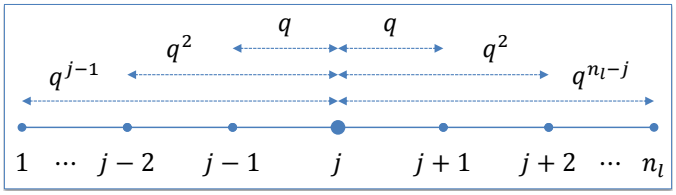


FIGURE 1. Since $0 < q < 1$, the proposed impact function $\mathrm{im}(j, i) = q^{|j-i|}$ of the choice j on other values $i = j \pm 1$, $i = j \pm 2, \ldots$ exponentially decreases as the distance $|j - i|$ increases.

The values of q are constrained to the range $0 < q < 1$; when q approaches zero, then the left side of (4) approaches $-b \leq 0$. On the other hand, if q approaches 1, then the left side of (4) approaches $n_l - b \geq 0$. Thus, since $0 \leq b \leq n_l$, there must be a value for q for which (4) is satisfied. Moreover, the left side of (4) is monotonically increasing in q, so there is precisely one solution to equation (4). Such a solution can be efficiently obtained by solving the algebraic equation (5) using standard numerical methods. In Section 5.2 we investigate the impact of various values of the propagation parameter b, $0 \leq b \leq n_l$, on the accuracy of our reputation system.

3.2. **Provenance-Aware Credibility Propagation.** Given the impact function $\mathrm{im}(i, j)$ for computing the impact of choosing level i to level j (and vice versa, see footnote 5) in accordance with the distance between options i and j, we can now refine our equations for computing the credibility degree of a level i from a list of levels l as well as for computing the trustworthiness of a user r. First, we define β_{li} as the non-normalized credibility degree of quality level i from a list l of levels. Considering the idea of credibility propagation among the options, the

[7]In the future work, we plan to investigate systems in which the value of b can be different for different users, reflecting how confident they feel about their ratings. In such a case one should make sure that the sum of such b_r of a user r and the credibility c_r conveyed by r to the option j which the user has chosen (in the present case always equal to 1) sum up to the same value $s = b_r + c_r$ equal for all users.

credibility degree of a quality level j is obtained not only from the users who have selected that particular level, but also from users who have chosen other levels, in proportion to the distance of their choices from level j. In other words, we define the credibility degree for a quality level j of an item as amount of credibility which such level has obtained from all users who rated that particular item, regardless of which particular level they chose as the most appropriate assessment of the quality of that item. Therefore, we reformulate equation Eq. (1) for computing the non-normalized credibility degree of quality level li as follows:

$$\beta_{li} = \sum_{j,r\,:\,r \to lj} (T_r)^{\alpha} \mathrm{im}(i,j) \tag{6}$$

On the other hand, some rating systems provide contextual information about the ratings, which we call *rating provenance*. Such contextual information may contain attributes such as the educational level of each user, the average grade of each student in a student feedback system, etc. Thus, a reputation system needs to take into account these contextual attributes about the quality of ratings as meta-data. Clearly, each rating system has its own list of contextual attributes. In this paper we give an example of a student feedback system, in which users rate movies which are a part of the learning material for a course; users include both staff and students. Thus, we propose that our provenance model be based on the attributes which include two contextual attributes: staff/student as well as the amount of time spent watching each movie. We will show the details and evaluation results of our algorithm over a dataset from such a system in Section 5.6. We note that this approach can easily be adapted for other contextual attributes as well. Moreover, the proposed provenance model is based on the approach proposed by Wang et al. [11], where this approach has been used in the context of participatory sensing.

The main idea of our provenance model is to define a weight function which depends on the contextual attributes provided by the rating system. To this end, we associate a weight with each attribute and then aggregate all such weights as a simple product of all weights, in order to obtain the cumulative provenance weight. Such provenance weight is then used in the computation of credibility of each level, as well as in the computation of users' trustworthiness.

As we have mentioned, in one of our examples, that is, the one of a student feedback system for evaluating educational movies, students are asked to rate movies used in an online course. For each rating, the system provides the staff/student status of each user as well as the amount of time each user has spent watching the movie.

We utilized both the staff/student information and watching time duration as two contextual attributes to create the provenance of ratings. To this end, we assign a slightly higher credibility to the staff users compared to the credibility of student users. Thus, we define the *staff weight*, denoted as $w_s(r)$, which we set $w_s(r) = 0.98$ if user r is a staff and $w_s(r) = 0.95$ for student users. Moreover, we take into account the duration of the watching time, to reflect the fact that if a user spends more time watching a movie (for example, watching the entire movie, possibly several times, versus watching only a small part of it) then such a user can provide more reliable rating of such a movie. We denote the watching time of a user r by $TW(r)$ and the duration of a movie l by $TD(l)$, respectively. Thus, we compute *the gap* between them as $|\min\{TW(r), TD(l)\} - TD(l)|$. We now define

the *watching time weight* for a user r and a movie l, denoted $w_t(r,l)$, as:

$$w_t(r,l) = e^{-|\min\{TW(r),T(l)\}-T(l)|\times\gamma} \tag{7}$$

where $0 \leq \gamma \leq 1$ is a *duration sensitivity* parameter, which controls the relative impact of the watching time weight. Note that Eq. (7) makes $w_t(r,l)$ equal to 1 when the gap between the watching time and duration is 0 and that $w_t(r,l)$ decreases when such gap increases. Given both the staff/student weight $w_s(r)$ and the watching time weight $w_t(r,l)$, we define the *provenance weight*, denoted as $w_p(r,l)$, as the aggregation of these two weights by taking their product:

$$w_p(r,l) = w_s(r) \times w_t(r,l) \tag{8}$$

Note that, in general, the provenance weight can be defined as the product of the weight values of all contextual attributes, where such weights are in the range $[0,1]$. Given the provenance weight, we use the provenance information to refine the definition of the non-normalized credibility degree β_{li}, which was given by Eq. (6), as follows:

$$\beta_{li} = \sum_{j,r\,:\,r\to lj} (T_r)^{\alpha} \times \mathrm{im}(i,j) \times w_p(r,l) \tag{9}$$

For normalizing the credibility of a level i from a list l of choices of levels, we use the same method as used in the original approach, i.e., we set

$$\rho_{li} = \frac{\beta_{li}}{\sqrt{\sum_{1\leq j\leq n_l} (\beta_{lj})^2}} \tag{10}$$

Given the credibility degree for all quality levels of items, we can now also refine the method of computation of the trustworthiness of users. Such trustworthiness of a user is the weighted sum of all credibility degrees from all quality levels of items which have been rated by such a user:

$$T_r = \sum_{l,i\,:\,r\to li\ 1\leq j\leq n_l} \rho_{li} \times \mathrm{im}(i,j) \times w_p(r,l) \tag{11}$$

Note that we formalized the uncertainty in rating systems through both credibility propagation among options as well as rating provenance; this is done both for computing the credibility degrees of quality levels and for computing the users' trustworthiness.

3.3. **Iterative Aggregation.** Equations (9), (10) and (11) provide interdependent definitions of credibility degrees of levels and of trustworthiness of users. Clearly, the credibility degree ρ_{li} of a quality level i of an item l depends on the trustworthiness of users who rated such an item with level i or with a level close to i; on the other hand, the trustworthiness of a user depends on the credibility degree of the level i he has chosen as well as the near by levels for items l which have been rated by such a user.

We now propose an iterative algorithm to compute both the credibility degrees of levels as well as the trustworthiness scores of users in a manner similar to that used in the original RTV algorithm. Thus, we denote the non-normalized credibility of a level i, normalized credibility of a level i and the trustworthiness of a user r at an

iteration stage t as $\beta_{li}^{(t)}$, $\rho_{li}^{(t)}$ and $T_r^{(t)}$, respectively. Therefore, equations (9), (10) and (11) now take the following form dependent on the round of iteration t:

$$\beta_{li}^{(t+1)} = \sum_{j,r\,:\,r\to lj} (T_r^{(t)})^{\alpha} \times \mathrm{im}(i,j) \times w_p \tag{12}$$

$$\rho_{li}^{(t+1)} = \frac{\beta_{li}^{(t+1)}}{\sqrt{\sum_{1\le j\le n_l} \left(\beta_{lj}^{(t+1)}\right)^2}} \tag{13}$$

$$T_r^{(t+1)} = \sum_{l,i\,:\,r\to li\ 1\le j\le n_l} \rho_{li}^{(t+1)} \times \mathrm{im}(i,j) \times w_p \tag{14}$$

Algorithm 1 is an iterative process for computing simultaneously the values of $\beta_{li}^{(t)}$, $\rho_{li}^{(t)}$ and $T_r^{(t)}$. The algorithm starts with identical trustworthiness scores for all users, $T_r^{(0)} = 1$. The iteration stops when there is no significant change of the credibility degrees of all options, i.e., when

$$\left(\sum_{l,i} (\rho_{li}^{(t+1)} - \rho_{li}^{(t)})^2 \right)^{1/2} < \varepsilon$$

where ε is the precision target desired, which, in our experiments, was set to 10^{-6}. When this happens we say that the computation of credibilities of the levels has converged (see Algorithm 1).

Algorithm 1 Iterative algorithm to compute the credibility and trustworthiness.

procedure CredTrustComputation($A, b, \alpha, \gamma, n_l$)
  Compute q using (5)
  for each $1 \le i, j \le n_l$ **do**
    $\mathrm{im}(i,j) \leftarrow q^{|i-j|}$
  end for
  $T_r^{(0)} \leftarrow 1$
  $t \leftarrow 0$
  repeat
    for each level i and item l **do**
      Compute $\beta_{li}^{(t+1)}$ using (12)
    end for
    for each level i and item l **do**
      Compute $\rho_{li}^{(t+1)}$ using (13)
    end for
    for each user r **do**
      Compute $T_r^{(t+1)}$ using (14)
    end for
    $t \leftarrow t + 1$
  until credibilities have converged
  Return $\vec{\rho}$ and $\vec{T}$
end procedure

3.4. **Multi-dimensional Reputation.** Examples of the eBay multi-categories feedback system and of student course evaluation in educational systems suggest that a reputation system needs to consider the correlation among users' feedback across multiple categories. A traditional approach is to apply the trust computation method 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; see, for example, [12].

In Eq. (3) we proposed an aggregation method for a single category rating system. In order to extend this method to multi-dimensional rating systems, we first aggregate the ratings of items and the trustworthiness of users by applying Algorithm 1 over each category separately. In this way, if there are K categories in the rating system, we obtain K trustworthiness ranks $T_r^p,\ 1 \leq p \leq K$, for each user. We aggregate these trustworthiness ranks using a simple average to obtain the final users' trustworthiness, denoted as $\hat{T}_r$, i.e., we set $\hat{T}_r = (\sum_{p=1}^{K} T_r^p)/K$. After that, we employ a weighted average to compute the final rating of each item l in category k, as follows

$$R_{lk} = \frac{\sum_{i,r\,:\,r \to lik} i \times (\hat{T}_r)^p}{\sum_{i,r\,:\,r \to lik} (\hat{T}_r)^p} \tag{15}$$

where $r \to lik$ denotes the fact that user r has chosen option i from the l^{th} list Λ_l for category k. An overall rating of an item l can now be obtained as an average of rating of all K aspects, i.e., $R_l = (\sum_{k=1}^{K} R_{lk})/K$. Constant $p \geq 1$ is a design parameter that can be tuned to obtain optimal performance in each particular context.

3.5. **Algorithm Complexity.** Since in all practical applications the rating matrix is very sparse, we evaluate the time complexity of our reputation system based on the number of ratings, denoted as L (see Table 1 for a similar observation in the MovieLens dataset). Let n be the number of users and m the total number of items rated; then $L \ll n \times m$. The initial part of Algorithm 1, i.e., lines 2-7, take 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 have complexities $O(L \times n_l)$, $O(m \times n_l)$, 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 $O(k \times L)$, where k is the number of iterations; such k depends on the threshold of precision ε of vector $\vec{\rho}^{(k)}$. As we will discuss later, in our examples for accuracy $\varepsilon = 10^{-12}$ the number of iterations k was smaller than 20 and for $\varepsilon = 10^{-3}$ it was around 10. Thus, overall, our algorithm is very efficient.

4. General Proof of Convergence

In this section we prove that our algorithm converges. To do that, we actually prove the convergence of a very general class of algorithms of similar kind. The proof generalizes the proofs from [9,13] and it not only covers algorithms introduced in [9, 13] and the algorithm we presented in this paper, but, in all likelihood, it is also sufficiently general to guarantee the convergence of possible future refinements and extensions of iterative algorithms similar to the present algorithm. To present it in such generality we first introduce a few definitions and specify our notation.

4.1. **Notation and Definitions.** Assume that N *sources* $S_1, \ldots, S_N$ provide answers a_{li} to L queries $Q_1, \ldots, Q_L$. We will regard each piece of information a_{li} provided by a source S_i as an answer to a query Q_l. For the same query Q_l two different sources S_{i_1}, S_{i_2} might provide either equal or unequal answers a_{li_1}, a_{li_2}. Answers can be either numerical or non-numerical values. Each source can be seen as providing a vote of confidence for each piece of information it provides. The objective of our algorithm is to aggregate these answers in a most robust way, minimizing the impact of both stochastic errors as well as malicious reporting by the sources. We consider queries whose answers are either numerical within a predetermined range, or belong to a predetermined set of finitely many choices. We allow the possibility that for a query sources can provide multiple answers; in this case the answer is represented by a vector $\langle a_{li_1}, \ldots, a_{li_m} \rangle$, and either the sources of information or the aggregation authority provide positive weights $\langle w_{li}^1, \ldots, w_{li}^m \rangle$ for these multiple answers, with weights summing up to a constant, which, without loss of generality, can be assumed to be equal to one, $\sum_{1 \le k \le m} w_{li}^k = 1$. We denote by Λ_l a list which includes all answers to the query Q_l.

Not necessarily all the sources provide answers to all queries. Since some of the sources might provide the same answer to a particular query, the total number of answers provided can be smaller than the number of sources providing answers to that query. However, we also allow the possibility that the list Λ_i can contain more answers than those actually provided by the sources, to allow for scenarios where for a particular query the sources choose the best answer from a preselected list of choices.

The degree of agreement of the sources regarding the most appropriate answer on each particular query is reflected in the calculated *credibility degree* for each answer. The credibility degree of an answer $a_i \in \Lambda_l$ is denoted by ρ_{li}.

Based on the credibility of each answer, we calculate a measure of *trustworthiness* T_r of each source S_r. We use notation $r \to li$ to denote the fact that a source S_r has provided an answer $\vec{a}_{li}$ for a query Q_l. Also, let n_l denote the number of answers on a list of answers Λ_l to query Q_l, which are either provided by the sources or are a predetermined list of choices from which sources S_r can choose an answer; thus, $n_l = |\Lambda_l|$.

For each $p \ge 1$ we denote by $\|\vec{x}\|_p$ the usual p-norm of the vector $\vec{x} = \langle x_1, \ldots, x_n \rangle$, i.e.,

$$\|\vec{x}\|_p = \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}}.$$

We denote by $\mathbb{R}^+$ the set of non-negative reals. Let

$$\vec{\rho} = \langle \rho_{li} \ : \ 1 \le l \le L; \ 1 \le i \le n_l \rangle$$

be any vector in $(\mathbb{R}^+)^M$, and let $\mathbf{T}(\vec{\rho}, r)$ be any function from $(\mathbb{R}^+)^M \times \mathbb{N}$ which for every $1 \le r \le N$ satisfies:

(a) $\mathbf{T}(\vec{\rho}, r) \ge 0$ for every $\vec{\rho} \in (\mathbb{R}^+)^M \times \mathbb{N}$;

(b) $\mathbf{T}(\vec{\rho}, r)$ has continuous second order partial derivatives with respect to variables ρ_{li};

(c) for all l such that $1 \leq l \leq L$,

$$\frac{\partial \mathbf{T}(\vec{\rho}, r)}{\partial \rho_{li}} \geq 0; \tag{16}$$

$$\max_{1 \leq i \leq n_l} \frac{\partial \mathbf{T}(\vec{\rho}, r)}{\partial \rho_{li}} \leq 1. \tag{17}$$

For each vector $\rho \in (\mathbb{R}^+)^M$ we denote the vector $\langle \mathbf{T}(\vec{\rho}, r) \ : \ 1 \leq r \leq N \rangle \in \mathbb{R}^N$ by $\overrightarrow{\mathbf{T}}(\vec{\rho})$. Let us consider the set of equations

$$\mathcal{C} = \left\{ \sum_{1 \leq i \leq n_l} \rho_{li}^2 = 1 \quad : \quad 1 \leq l \leq L \right\}.$$

Since the function $\| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_p = \left(\sum_{1 \leq r \leq N} \mathbf{T}(\vec{\rho}, r)^p \right)^{\frac{1}{p}}$ is continuous, it achieves its maximum on the set of all $\vec{\rho} \in (\mathbb{R}^+)^M$ which satisfy constraints $\mathcal{C}$, because such a set is a Cartesian product of L finitely dimensional hyper-spheres and is thus compact. This fact makes the following definition correct.

Definition 1. *An assignment of ranks $\vec{\rho}$ is $(p+1)$–fair if it maximizes the value of the norm $\| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_{p+1}$ of the trustworthiness vector $\overrightarrow{\mathbf{T}}(\vec{\rho})$.*

Thus, a $(p+1)$–fair assignment of ranks $\vec{\rho}$ gives "the benefit of the doubt" to the sources, giving them the largest possible "joint trustworthiness", i.e., the largest possible $(p+1)$–norm of the vector comprising of their trustworthiness scores, for the given trust function $\mathbf{T}(\vec{\rho}, r)$.

Theorem 4.1. *If an assignment of ranks $\vec{\rho}$ is $(p+1)$–fair then for all $1 \leq l \leq L$ and all $1 \leq i \leq n_l$ it satisfies equation*

$$\rho_{li} = \frac{\frac{\partial \left(\| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_{p+1} \right)^{p+1}}{\partial \rho_{li}}}{\left(\sum_{1 \leq j \leq n_l} \left(\frac{\partial \left(\| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_{p+1} \right)^{p+1}}{\partial \rho_{lj}} \right)^2 \right)^{1/2}} \tag{18}$$

The above equation has a simple intuitive explanation: the rank of an answer i on the list Λ_l should be proportional to its impact on the norm of the trust vector $\| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|$, relative to the impacts of all items on that list.

Let $\vec{\rho}$ be any vector in $\mathbb{R}^M$; we define $(\vec{\rho})_l$ to be the projection $\langle \rho_{li} \ : \ 1 \leq i \leq n_l \rangle$ of $\vec{\rho}$ onto the subspace corresponding to a single list of answers Λ_l to a query Q_l. Then for any fixed l, the above equations for all $1 \leq i \leq n_l$ can be put in a compact vector form:

$$(\vec{\rho})_l = \frac{\nabla_l \ \| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_{p+1}^{p+1}}{\left\| \nabla_l \ \| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_{p+1}^{p+1} \right\|_2}. \tag{19}$$

Note that a $p+1$–fair vector $\vec{\rho}$ is a fixed point of the vector function defined by the right hand side of the above equation.

Proof. Let

$$F(\vec{\rho}) = \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}; \tag{20}$$

then to prove the theorem it is enough to show that if $\vec{\sigma}$ is a maximum of $F(\vec{\rho})$, subject to the constraints $\mathcal{C}$, then σ must satisfy (19). For this purpose we introduce for each list Λ_l a Lagrangian multiplier λ_l and define define $\vec{\lambda} = \langle \lambda_l, : 1 \leq l \leq L \rangle$. We now look for the stationary points of the Lagrangian function

$$\Phi(\vec{\rho}, \vec{\lambda}) = F(\vec{\rho}) - \sum_{q=1}^{L} \lambda_q \left(-1 + \sum_{m=1}^{n_q} \rho_{qm}^2 \right).$$

Let l, m be list indices and i, j item indices; we now have

$$\frac{\partial F(\vec{\rho})}{\partial \rho_{li}} = \frac{\partial \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}}{\partial \rho_{li}}; \tag{21}$$

$$\frac{\partial \Phi(\vec{\rho}, \vec{\lambda})}{\partial \rho_{li}} = \frac{\partial \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}}{\partial \rho_{li}} - 2\lambda_l \rho_{li}; \tag{22}$$

$$\frac{\partial \Phi(\vec{\rho}, \vec{\lambda})}{\partial \lambda_l} = -1 + \sum_{j=1}^{n_l} \rho_{lj}^2. \tag{23}$$

If $(\vec{\rho}, \vec{\lambda})$ is a stationary point of Φ then by (22) $\frac{\partial \Phi(\vec{\rho}, \vec{\lambda})}{\partial \rho_{li}} = 0$ for all l, i and thus

$$\rho_{li}\lambda_l = \frac{1}{2} \frac{\partial \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}}{\partial \rho_{li}}. \tag{24}$$

This yields

$$\rho_{li}^2 \lambda_l^2 = \frac{1}{4} \left(\frac{\partial \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}}{\partial \rho_{li}} \right)^2,$$

and by summing the above equations for all indices j of objects on the list l we get

$$\lambda_l{}^2 \sum_{j=1}^{n_l} \rho_{lj}^2 = \frac{1}{4} \sum_{1 \leq j \leq n_l} \left(\frac{\partial \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}}{\partial \rho_{lj}} \right)^2.$$

Since $(\vec{\rho}, \vec{\lambda})$ is a stationary point of Φ also $\frac{\partial \Phi(\vec{\rho}, \vec{\lambda})}{\partial \lambda_l} = 0$; this by (23) implies $\sum_{i=1}^{n_l} \rho_{li}^2 = 1$, and since by (24) λ_l must be positive, we obtain

$$\lambda_l = \frac{1}{2} \left(\sum_{1 \leq j \leq n_l} \left(\frac{\partial \|\overrightarrow{\mathbf{T}}(\vec{\rho})\|_{p+1}^{p+1}}{\partial \rho_{lj}} \right)^2 \right)^{1/2}.$$

This, together with (24) implies (18). □

We now show that a $p+1$–fair vector can be approximated by an iterative procedure given by Algorithm 4.2.

4.2. **Algorithm.** Let us define *the weight function* $\mathbf{W}(\vec{\rho}, r, l, i)$ as

$$\mathbf{W}(\vec{\rho}, r, l, i) = \frac{\partial \mathbf{T}(\vec{\rho}, r)}{\partial \rho_{li}}, \tag{25}$$

and *the benefit function* $\mathbf{B}(\vec{\rho}, l, i)$ as

$$\mathbf{B}(\vec{\rho}, l, i) = \sum_{r \to li} \mathbf{W}(\vec{\rho}, r, l, i)\, \mathbf{T}(\vec{\rho}, r)^p. \tag{26}$$

Note that

$$\mathbf{B}(\vec{\rho}, l, i) = \frac{1}{p+1} \frac{\partial \ \| \overrightarrow{\mathbf{T}}(\vec{\rho}) \|_{p+1}^{\ p+1}}{\partial \rho_{li}}, \tag{27}$$

and that condition (18) for $p+1$–fairness of a rank $\vec{\rho}$ can be written as

$$\rho_{li} = \frac{\mathbf{B}(\vec{\rho}, l, i)}{\sum_{1 \leq j \leq n_l} \mathbf{B}(\vec{\rho}, l, j)^2 \ ^{1/2}}. \tag{28}$$

In our iterative algorithm, for each answer a_{li} on the list Λ_l we will keep track of its credibility degree at the step of iteration k, denoted by $\rho_{li}^{(k)}$. These individual credibility degrees $\rho_{li}^{(k)}$ will be collected into a single vector $\vec{\rho}^{\,(k)}$:

$$\vec{\rho}^{\,(k)} = \langle \rho_{li}^{(k)} \ : \ 1 \leq l \leq L, \ 1 \leq i \leq n_l \rangle.$$

We will also keep track of the benefit which a_{li} gets from all the sources, denoted by $\beta_{li}^{(k)}$. For each source S_r we will keep track of its *trustworthiness* $T_r^{(k)}$ at the stage of iteration k, as well as the weight w_{rli} with which the p-power of the trustworthiness of a source r is conferred to answer a_{li}.

Let $\varepsilon > 0$ be the precision threshold for our iterative algorithm; the value of p, $p \geq 1$ in the norm used has a role of a "discrimination" setting parameter, as it will be explained later.

Theorem 4.2. *The iterative procedure given by Algorithm 1 always converges to a* $(p+1)$*–fair vector.*

Proof. Our proof generalizes our convergence proofs from [9,13]. Let $F(\rho)$, $\Phi(\vec{\rho}, \vec{\lambda})$ be as in the proof of Theorem 4.1. We will show that our iterative algorithm converges to a $(p+1)$–fair vector, which, as shown in the proof of Theorem 4.1 is a stationary point of $\Phi(\vec{\rho}, \vec{\lambda})$. If we define $\vec{\rho} \mapsto (\vec{\rho})^*$ to be the mapping such that for an arbitrary $\vec{\rho}$,

$$((\vec{\rho})^*)_l = \frac{(\nabla F(\vec{\rho}))_l}{\|(\nabla F(\vec{\rho}))_l\|_2}, \tag{29}$$

then (19) shows that $\vec{p}$ is a $p+1$–fair vector just in case $(\vec{\sigma})^* = \vec{\sigma}$, i.e., for all $1 \leq l \leq L$,

$$(\vec{\sigma})_l = \frac{(\nabla F(\vec{\sigma}))_l}{\|(\nabla F(\vec{\sigma}))_l\|_2}. \tag{30}$$

Note also that in our algorithm the approximation $\vec{\rho}^{\,(n+1)}$ of the vector $\vec{\rho}$ obtained at the stage of iteration $(n+1)$ can be written as

$$\vec{\rho}^{\,(n+1)} = (\vec{\rho}^{\,(n)})^*,$$

Algorithm 2 Algorithm

Initialization:

For all $1 \leq r \leq N$, for all $1 \leq l \leq L$ and all $1 \leq i \leq n_l$,

$T_r^{(0)} = 1$; (trustworthiness of source r)

$\rho_{li}^{(0)} = \dfrac{1}{\sqrt{n_l}}$; (rank of answer a_{li})

$w_{rli}^{(0)} = 0$; (weight for source r conferring credit to answer a_{li})

$\beta_{li}^{(0)} = 0$; (benefit conferred to answer a_{li})

$k = 0$; (round of iteration)

Repeat:

$$w_{rli}^{(k+1)} = \mathbf{W}(\vec{\rho}^{(k)}, r, l, i);$$

$$\beta_{li}^{(k+1)} = \sum_r w_{rli}^{(k+1)} \left(T_r^{(k)}\right)^p;$$

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

$$T_r^{(k+1)} = \mathbf{T}(\vec{\rho}^{(k+1)}, r);$$

until: $\|\vec{\rho}^{(k+1)} - \vec{\rho}^{(k)}\|_2 < \varepsilon$.

and that our algorithm will halt when $\vec{\rho}^{(n)}$ get sufficiently close to a fixed point[8] $\vec{\sigma} = (\vec{\sigma})^*$ of the mapping $\vec{\rho} \to (\vec{\rho})^*$.

Let $\vec{\rho}$ be an arbitrary vector such that $\|(\vec{\rho})_l\|_2 = 1$ for all $1 \leq l \leq L$; we abbreviate $(\vec{\rho})^*$ with $\vec{\rho}^*$ and let $\vec{h} = \vec{\rho}^* - \vec{\rho}$. By applying the Taylor formula with a remainder in the Lagrange form, we get that for some $0 < c < 1$ and $\vec{\mu}_c = c\vec{\rho} + (1-c)\vec{\rho}^*$ we have

$$\begin{aligned} F(\vec{\rho}^*) &= F(\vec{\rho} + \vec{h}) \\ &= F(\vec{\rho}) + \nabla F(\vec{\rho}) \cdot \vec{h} + \frac{1}{2} \sum_{l,m,i,j} \frac{\partial^2 F(\vec{\mu}_c)}{\partial \rho_{li} \partial \rho_{mj}} h_{li} h_{mj}. \end{aligned} \tag{31}$$

Since

$$(\vec{h})_l = (\vec{\rho}^*)_l - (\vec{\rho})_l = \frac{(\nabla F(\vec{\rho}))_l}{\|(\nabla F(\vec{\rho}))_l\|_2} - (\vec{\rho})_l, \tag{32}$$

[8]Note that we do not need to prove the uniqueness of such a fixed point because our final ranks are *defined* as the output of our algorithm, and we only need to prove that our algorithm eventually terminates.

using also (29), we get

$$\begin{aligned}(\nabla F(\vec{\rho}))_l \cdot (\vec{h})_l &= (\nabla F(\vec{\rho}))_l \cdot \quad \frac{(\nabla F(\vec{\rho}))_l}{\|(\nabla F(\vec{\rho}))_l\|_2} - (\vec{\rho})_l \\ &= \|(\nabla F(\vec{\rho}))_l\|_2 - (\nabla F(\vec{\rho}))_l \cdot (\vec{\rho})_l \\ &= \|(\nabla F(\vec{\rho}))_l\|_2 - \|(\nabla F(\vec{\rho}))_l\|_2 (\vec{\rho}^{\,*})_l \cdot (\vec{\rho})_l \\ &= \|(\nabla F(\vec{\rho}))_l\|_2 (1 - (\vec{\rho}^{\,*})_l \cdot (\vec{\rho})_l)\end{aligned}$$

Let θ_l be the angle between the unit vectors $(\vec{\rho})_l$ and $(\vec{\rho}^{\,*})_l$, i.e., such that $\cos\theta_l = (\vec{\rho})_l \cdot (\vec{\rho^*})_l$. Then, (see Figure 2, left)

$$\frac{(\vec{h})_l}{2}^{\,2}_{\,2} = \sin\frac{\theta_l}{2}^{\,2} = \frac{1-\cos\theta_l}{2} = \frac{1-(\vec{\rho})_l \cdot (\vec{\rho}^{\,*})_l}{2}.$$

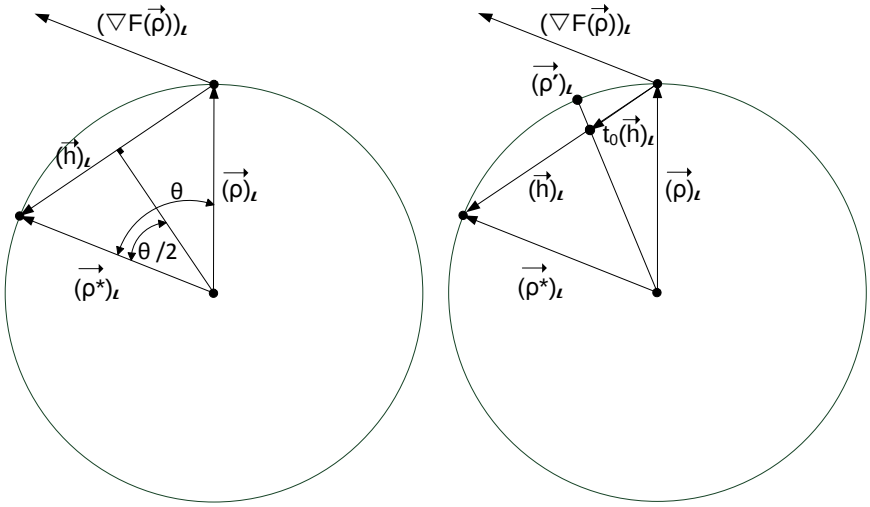


FIGURE 2. A geometric representation of the iterative procedure.

Combining the last two formulas we get

$$(\nabla F(\vec{\rho}))_l \cdot (\vec{h})_l = \frac{\|(\nabla F(\vec{\rho}))_l\|_2 \, \|(\vec{h})_l\|_2^2}{2}. \tag{33}$$

Assume first that $\|\vec{h}\|_2$ is sufficiently small, so that the contribution of the second order terms in (31) is small compared to the first order term, and, consequently

$$F(\vec{\rho}^{\,*}) \approx F(\vec{\rho}) + \nabla F(\vec{\rho}) \cdot \vec{h}. \tag{34}$$

Since $\|\nabla F(\vec{\rho})\|_2$ is a continuous function, it achieves its minimum on the compact set defined by our constraints, i.e., on the set $\mathcal{C} = \{\vec{\rho} \,:\, \|(\vec{\rho})_l\|_2 = 1,\ 1 \leq l \leq L\}$. It is easy to see that the condition given by equation (16) ensures that the directional derivative of $F(\vec{\rho})$ in the (radial) direction of vector $\vec{\rho}$ is always strictly positive; thus, on the compact set defined by our constraints its minimum must also be

strictly positive. Consequently, there exists $\kappa > 0$ such that $\|\nabla F(\vec{\rho})\|_2 > \kappa$ for all $\vec{\rho} \in \mathcal{C}$; using this and by summing equations (33) for all $1 \leq l \leq L$, we get

$$\nabla F(\vec{\rho}) \cdot \vec{h} > \frac{\kappa \, \|\vec{h}\|_2^2}{2}. \tag{35}$$

Together with (34) this implies that, for $\vec{\rho}^{\,(n)}$ and $\vec{h}^{\,(n)} = \vec{\rho}^{\,(n+1)} - \vec{\rho}^{\,(n)}$ obtained in our iterations,

$$F(\vec{\rho}^{\,(n+1)}) - F(\vec{\rho}^{\,(n)}) = F((\vec{\rho}^{\,(n)})^*) - F(\vec{\rho}^{\,(n)}) > \frac{\kappa \, \|\vec{h}^{\,(n)}\|_2^2}{2}$$

Thus, since $F(\vec{\rho})$ must be bounded on a compact set defined by the constraints $\mathcal{C}$, we get that $\|\vec{h}^{\,(n)}\|_2^2$ must converge to zero, i.e., $\|\vec{\rho}^{\,(n)} - (\vec{\rho}^{\,(n)})^*\|_2$ will eventually be smaller than the prescribed threshold and the algorithm will terminate.

If $\|\vec{h}\|_2$ is not sufficiently small so that the impact of the second order term in (31) makes the inequality (35) false, we supplement our algorithm with an initial phase which involves a line search. While this ensures a provable convergence of our algorithm, in all of our (very numerous) experiments such a line search was never activated; however, we were unable to prove without any additional assumptions that indeed such line search is superfluous, so we present a slight modification of our algorithm. Let

$$f(\vec{\rho}, t) = F(\vec{\rho} + t(\vec{\rho}^{\,*} - \vec{\rho}));$$

then, by the previous considerations, for sufficiently small t function $f(\vec{\rho}^{\,(n)}, t)$ is increasing in t. We now modify our iteration step as follows. If there exists $t_0 \in (0, 1)$ such that $\frac{\partial f}{\partial t}(\vec{\rho}, t_0) = 0$ (testing this amounts to solving a low degree algebraic equation), then we let $\vec{\rho}^{\,(n+1)} = \vec{\rho}^{\,\prime}$, where $\vec{\rho}^{\,\prime}$ is defined so that for all $1 \leq l \leq L$, and for the smallest root t_0 of the above equation,

$$(\vec{\rho}^{\,\prime})_l = \frac{(\vec{\rho}^{\,(n)})_l + t_0((\vec{\rho}^{\,(n)\,*})_l - (\vec{\rho}^{\,(n)})_l)}{\|(\vec{\rho}^{\,(n)})_l + t_0((\vec{\rho}^{\,(n)\,*})_l - (\vec{\rho}^{\,(n)})_l)\|_2};$$

see Figure 2, right; if no such t_0 exists, we let

$$\vec{\rho}^{\,(n+1)} = (\vec{\rho}^{\,(n)})^*.$$

The convergence now follows from an argument similar to the one in the previous case. □

Note that in the proof of convergence we did not use condition given by equation (17). However, this condition is necessary for the values of our algorithm to be practically meaningful. Note that by (25) and (26) we have

$$\mathbf{B}(\vec{\rho}, l, i) = \sum_{r \to li} \frac{\partial \mathbf{T}(\vec{\rho}, r)}{\partial \rho_{li}} \, \mathbf{T}(\vec{\rho}, r)^p,$$

i.e., the benefit that a source r can confer to an answer a_{li} is a positive fraction of $\mathbf{T}(\vec{\rho}, r)^p$. Thus, each source can confer to all answers on a single list Λ_l in total at most a benefit of $\sum_{1 \leq i \leq n_l} \frac{\partial \mathbf{T}(\vec{\rho}, r)}{\partial \rho_{li}} \mathbf{T}(\vec{\rho}, r)^p \leq \mathbf{T}(\vec{\rho}, r)^p$. Thus, trust functions satisfying the condition given by equation (17) limit the total benefit a single source can distribute to all choices on a single list.

5. 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.

5.1. **Experimental Environment.** Although there are a number of real world datasets for evaluating reputation systems such as MovieLens[9] and HetRec 2011 [14], none of them provides a clear ground truth. Thus, we conduct our experiments using both real-world datasets and synthetically generated datasets.

In order to generate our synthetic datasets, we used the statistical parameters of the MovieLens 100k dataset. These parameters are listed in Table 1. In this table, a statistical distribution for the number of votes per movie and a statistical distribution for the number of votes per user for the dataset were obtained from the available data by using MATLAB distribution fitting tools. Our synthetic datasets were obtained by using these probability distributions for the number of ratings. Moreover, we set both the minimum number of ratings for each user and the minimum number of ratings for each movie to 20. The quality of each movie has been randomly selected using the uniform distribution in the range $[1, 5]$. The ratings of each user were obtained by adding to the true values a zero mean Gaussian noise with different variance value for each user. All ratings are also rounded to be discrete values in the range $[1, 5]$. For each experiment on synthetic datasets, we run the algorithms over 100 different synthetically generated datasets, and then average the results. All the experiments have been conducted on an HP PC with 3.30GHz Intel Core i5-2500 processor with 8 GB RAM, running 64-bit Windows 7 Enterprise. The program code has been written in MATLAB R2012b.

Table 1. MovieLens 100k dataset statistics.

Parameter	MovieLens 100k
Ratings	100,000
Users	943
Movies	1,682
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 [7]. Two proposed discriminant functions are used, $g(\vec{d}) = \vec{d}^{-1}$ and $g(\vec{d}) = 1 - k_l\vec{d}$, see [7]; we call them *dKVD-Reciprocal* and *dKVD-Affine*, respectively. We recently introduced [15,16] a collusion attack against the *dKVD-Reciprocal* function

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

TABLE 2. Summary of different IF algorithms.

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

and showed that an attacker can compromise such function using its pole at the point $d = 0$. Thus, we consider the *dKVD-Affine* function for our comparative experiments as such a function is more robust against the attack [15].

The second IF method we consider is a correlation based ranking algorithm proposed by Zhou et al. [8]. 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 [17] between users ratings and the current estimated reputation values. We call this method *Zhou*.

The third algorithm is the pioneer IF algorithm proposed by Laureti et al. [6] and is an IF algorithm based on a weighted averaging technique similar to the algorithm proposed in [7]. The only difference between these two algorithms is in the discriminant function. The authors in [6] have leveraged discriminant function $\mathrm{g}(\vec{d}) = \vec{d}^{-\beta}$ and $\beta = 0.5$. We call this method *Laureti*.

Table 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:

$$RMS\ Error = \sqrt{\frac{\sum_{j=1}^{m}(r_j - \hat{r_j})^2}{n}} \tag{36}$$

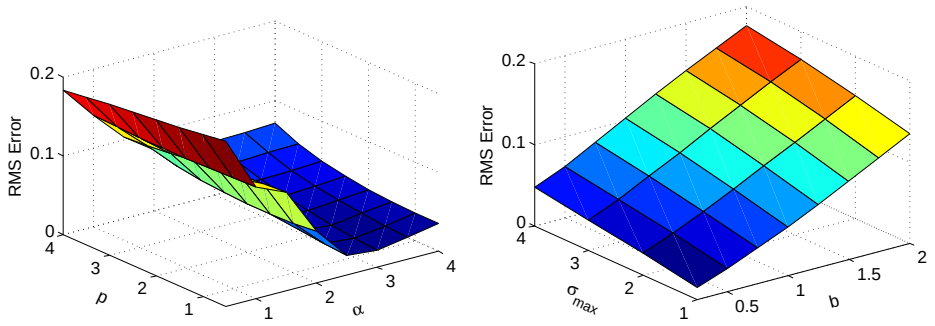
where r_j and $\hat{r_j}$ denote the true value and the estimated value of the reputation for item j, respectively.

5.2. **Parameter Sensitivity Analysis.** Beyond investigating the robustness of our reputation system, we also measured the sensitivity of its results with respect to 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 3(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 3(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 3. Accuracy of *PrRTV* with respect to various parameters' values.

5.3. **Robustness Against False Ratings.** In order to evaluate robustness of our algorithm against false ratings, we conduct experiments based on two types of malicious behavior proposed in [7] over the MovieLens dataset: *Random Ratings*, and a *Promoting Attack*. For the 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 [1,5] for those users.

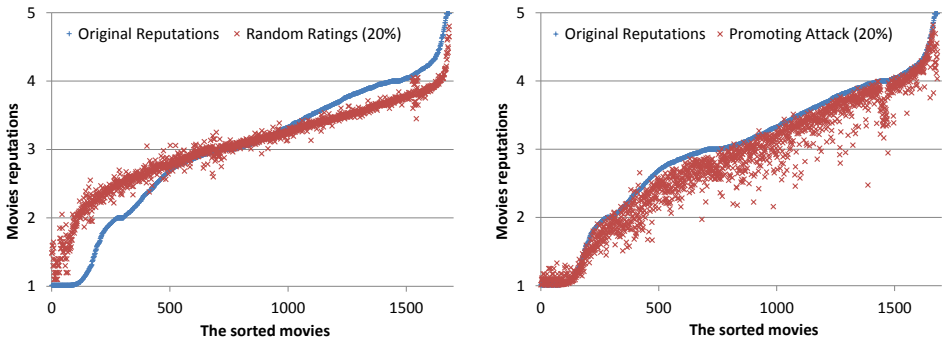
In slandering and promoting attacks, one or more users falsely produce negative and positive ratings, respectively, about one or more items [18]. The attacks can be conducted by either an individual or a coalition of attackers. The attacker may control many users, referred to as malicious users, and 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) [19]. 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. (15). 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.

Moreover, Figure 4(a) and 4(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 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

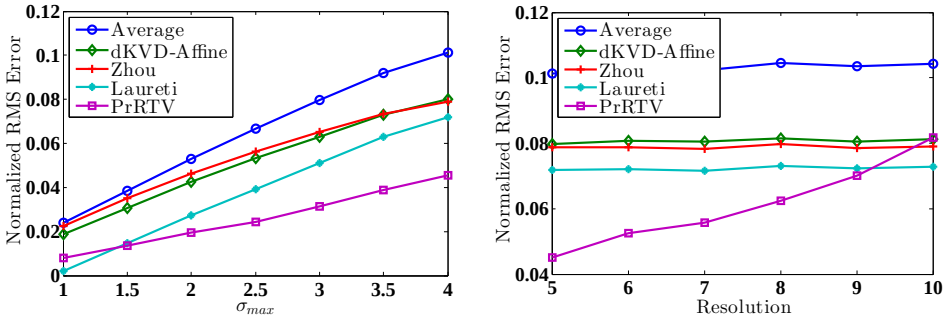


(A) Random Ratings (B) Promoting Attack

FIGURE 4. Perturbations of *PrRTV* against false ratings.

5.4. **Rating Resolutions and Users Variances.** Medo and Wakeling [20] reported that the accuracy of existing IF algorithms is 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 in which their number of users/items and their distribution of ratings are similar to the MovieLens dataset (see Table 1). The ratings scale is in the range $[1, R]$, where R is an integer number and $R \geq 2$. Also, the standard deviation σ_i for user i is randomly selected with a uniform distribution $U[0; \sigma_{max}]$, where σ_{max} is a real value in the range $[0, R-1]$. We also evaluate a normalized RMS error, $RMS/(R-1)$ (see Eq. (36) 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 $[1, 4]$. 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 5(a) shows the accuracy of the *PrRTV* algorithm along with the accuracy of the other IF algorithms for this



(A) Variance Changes (B) Resolution Changes

FIGURE 5. Accuracy with different variances and resolutions.

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 resolution of ratings, we set $\sigma_{max} = R - 1$ and vary the value of R in the range $[5, 10]$. Figure 5(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 is more sensitive to high resolution values. In other words, the accuracy of our reputation system significantly drops as the ratings resolution increases. The reason of this behavior is that Eq. (15) 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.

5.5. **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 [14]. 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 the Internet Movie Database (IMDb)[10] and Rotten Tomatoes movie critics systems[11]. Thus, we use the top critics ratings from Rotten Tomatoes as the domain experts for evaluating the accuracy of our approach.

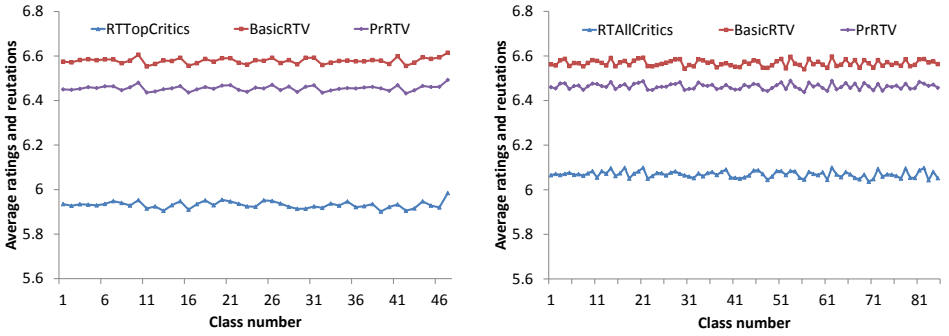
There are 10,109 movies in the HetRec 2011 MovieLens dataset rated by users. The dataset also includes the average ratings of the top and all critics of Rotten Tomatoes for 4645 and 8404 movies, respectively. We consider such average ratings as two ground truth data to evaluate the accuracy of our approach 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

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

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

this method is employed only for clarifying this comparison over such large number of movies.

Figure 6(a) and 6(b) report the comparison between the results of our algorithm with the ratings provided by RTTopCritics and RTAllCritics, respectively. Clearly, the results confirm that the reputation values given by our algorithm is very close to the experts opinions given by RTCritics. Moreover, the comparison of the results of *PrRTV* with the results of *BasicRTV* shows that the *PrRTV* algorithm provide a better accuracy than the *BasicRTV* algorithm as its aggregate ratings are 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 are provided by Rotten Tomatoes critics who tend to rank the movies more critically. These results confirm the acceptable accuracy of the proposed reputation system over this real-world dataset.



(A) RTTopCritics (B) RTAllCritics

FIGURE 6. Average reputations for movies computed by our algorithms and Rotten Tomatoes movie critics.

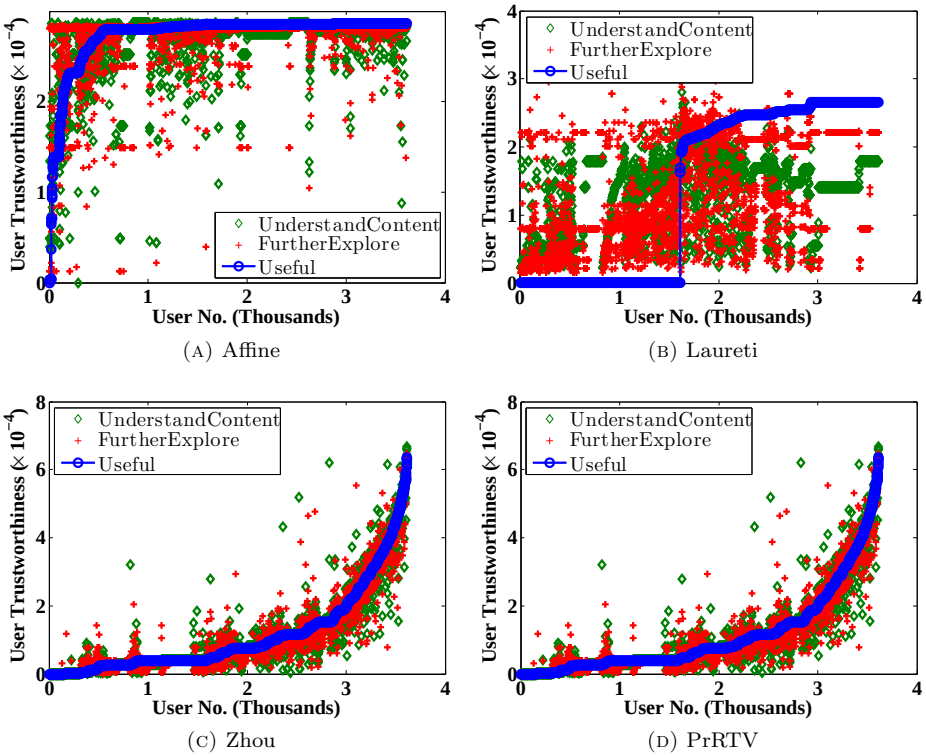
5.6. **Accuracy Over Student Feedback Dataset.** While student evaluations and feedback have significant roles to improve the quality of an education system, they have been criticized for being biased by students' perceptions [21]. Moreover, students are usually asked to rate the courses with respect to 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 two privately accessed student feedback datasets: 1) the first is provided by the Learning and Teaching Unit at UNSW, we call it *CATEI*; and 2) the second is provided by the School of Computer Science and Engineering at UNSW, we call it *CESCSE*.

The CATEI dataset consists of 17,854 ratings provided by 3,910 students (221 staff and 3,690 non-staff) for 20 movies in an online course presented in UNSW. In the CATEI dataset, students were asked to rate the movies in the range $[1-5]$ and with respect to 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 weight 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 7 compares the users' weights among three categories obtained by each IF algorithm. Moreover, Table 4 reports the Pearson correlation coefficient [17] among such weight values. One can see in the results that our reputation system provides the highest correlation among the weights for various categories. Those results validate the effectiveness of our approach over the CATEI dataset.



(A) Affine (B) Laureti

(C) Zhou (D) PrRTV

FIGURE 7. Users' weights obtained by the IF algorithms over three categories.

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

TABLE 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

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 5 reports the percentage of such correlation increase 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 using the traditional reputation computation method.

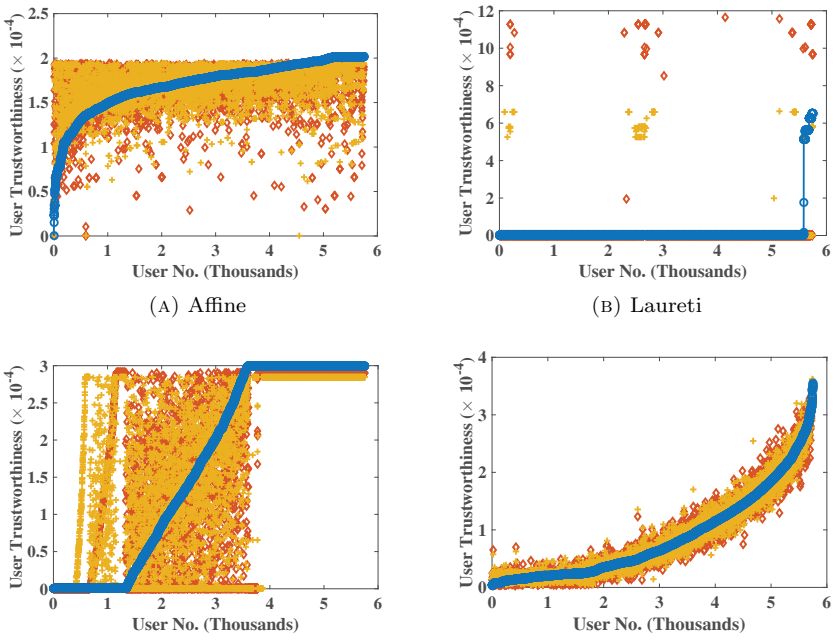
TABLE 5. Percentage of correlation increase among reputations by aggregating the weights obtained through each category (U: *Useful*, UC:*UnderstandContent*, FE:*FurtherExplore*).

	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**

The CESCSE dataset consists of 29,812 ratings provided by 5,895 students for 137 courses. The dataset contains anonymized data from the Course Experience Survey performed in 2001-2006 for the courses presented at the School of Computer Science and Engineering, UNSW Sydney, Australia. In the CESCSE dataset, students were asked to rate the courses in the range [1-5] with respect to 12 questions. The last question in this survey is about the overall satisfaction of the students in the course which is considered as the base question (category) for this experiment as it was suggested by experts who provided the dataset. Moreover, the dataset includes the student mark in the course. In the UNSW grading system, there are several grades including: High Distinction (HD), Distinction (DN), Credit (CR), Pass (PS), and Fail (FL). In this experiment, we utilize the students marks as the only contextual attribute to create the rating provenance. To this end, we give a slightly higher provenance weight to the ratings by students with higher marks. Thus, we set the values of rating provenance weights as follows: $w_p = 0.98$ for users with an HD mark, $w_s = 0.95$ for a DN mark, $w_s = 0.92$ for a CR mark, $w_s = 0.90$ for a PS mark, and $w_s = 0.85$ for a FL mark.

In this experiment, we investigate the correlation of users' weights obtained by each IF algorithm over the various questions (rating categories) in the CESCSE

dataset. To this end, we first apply the IF algorithms over each rating category and then sort the obtained users' weights in an increasing order based on the last category. Figure 8 compares the scatter plots of users' weights obtained by each IF algorithm for three categories. One can see from 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 CESCSE dataset.



(A) Affine (B) Laureti

(C) Zhou (D) PrRTV

FIGURE 8. Users' weights obtained by the IF algorithms over different questions in the CESCSE dataset. Blue marker: last question; Red marker: first question; Orange marker: second question.

5.7. **Analysis of Sparsity Pattern.** The datasets provided by rating systems are usually very sparse. For example, one can see in Table 1 that the MovieLens dataset provides an average around 6% rating density (proportion of number of ratings for each user). In this section, we evaluate the performance of our approach along with other IF algorithms over 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 such experiments, we synthetically generated datasets with various values for the density factor, η, in the range $[0.1, 0.5]$. 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 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 from the table that our algorithm is more robust 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 these algorithms use 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 [22].

TABLE 6. 1-norm absolute error between reputation vectors with various density factors in the ratings matrix.

	$\vec{r} - \tilde{\vec{r}}_1$					
	Average	*dKVD-Affine*	*Laureti*	*Zhou*	*BasicRTV*	*PrRTV*
$\eta = 0.1$	169.57	149.23	143.27	130.24	160.71	123.73
$\eta = 0.2$	113.42	98.28	95.02	86.65	106.00	80.10
$\eta = 0.3$	86.06	73.51	71.95	65.73	80.45	60.76
$\eta = 0.4$	68.87	57.82	57.38	52.47	64.02	48.25
$\eta = 0.5$	56.47	46.94	47.04	42.96	52.50	39.42

5.8. **Analysis of Error and Convergence.** In this section, we conduct a set of experiments to analyze behaviors of our iterative algorithm in terms of error and convergence. Thus, we investigate two 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 the 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:

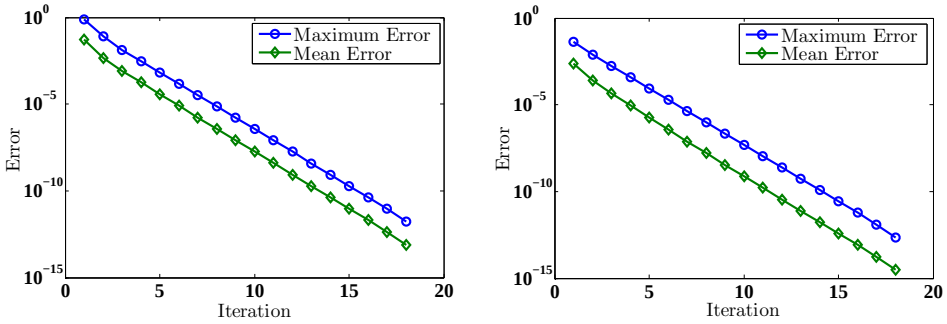
$$error_{\rho}^{(l)} = \frac{1}{m \times n_l} \sum_{l=1}^{m} \sum_{i=1}^{n_l} \left| \rho_{l_i}^{(\infty)} - \rho_{l_i}^{(l)} \right|$$

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

The results reported in Figure 9 show that the aforementioned errors decline for both credibility and trustworthiness values. For all experiments, we set convergence threshold with an error $\left\| \vec{\rho}^{(l+1)} - \vec{\rho}^{(l)} \right\|_2$ less than 10^{-12}. The results show that the error decreases exponentially in the *PrRTV* algorithm.

6. RELATED WORK

According to to previous work, as users of online retail stores rely more an more on rating systems to decide which product to purchase, more and more efforts devoted to manipulating reputation scores provided by the system in order to gain unfair advantage over competitors [2]. To solve this problem, Mukherjee et al., [5]



(A) Credibility (B) Trustworthiness

FIGURE 9. Convergence and error of credibility and trust scores over the MovieLens dataset.

proposed a model for spotting fake review groups in online rating systems. Their model analyzes feedback cast on products in Amazon online market to find collusion groups.

In a more general setup, collusion detection has been investigated in P2P and reputation management systems; we refer the readers to surveys [23, 24]. EigenTrust [25] is a well known algorithm proposed to produce collusion free reputation scores; however, Lian et al. [26] have shown that it is not robust against collusion. Other approaches have been proposed [27, 28, 29] that use a set of signals and alarms to detect suspicious behavior. The most well-known ranking algorithm e.g. the PageRank algorithm [30], was also designed to prevent collusive groups from obtaining undeserved ranks for webpages.

Several IF algorithms have been proposed for reputation systems [6, 7, 8, 31, 32]. While such IF algorithms provide adequate performance for filtering faults and simple cheating attacks, we recently showed that they are all vulnerable against sophisticated attacks [15, 16]. Medo and Wakeling [20] investigated the sensitivity of the IF algorithms to rating resolution as well as discrete/continuous ratings. Galletti et al. [33] proposed a mathematical framework for modelling 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 methods, and in particular from its ancestor RTV in three aspects. First, the distance between the rating options is taken into account in our method. Second, reputation scores are in fact multi dimensional, and finally, the provenance of rating scores is taken into account.

7. 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. Moreover, we provided a mathematical framework for proving the convergence of iterative filtering algorithms which yields a proof of convergence of our algorithm.

The experiments conducted on both synthetic and three real-world datasets show the superiority of our model over three well-known iterative filtering algorithms. Since the proposed framework has shown considerable promise, we plan to extend the algorithm to distributed and privacy-preserving reputation system.

References

[1] Haitao Xu, Daiping Liu, Haining Wang, and Angelos Stavrou. E-commerce reputation manipulation: The emergence of reputation-escalation-as-a-service. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1296–1306, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[2] 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.

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

[4] 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.

[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] Paolo Laureti, L. Moret, Yi-Cheng Zhang, and Yi-Kuo Yu. Information filtering via Iterative Refinement. *EPL (Europhysics Letters)*, 75:1006–1012, September 2006.

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

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

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] Matus Medo and Joseph R. Wakeling. The effect of discrete vs. continuous-valued ratings on reputation and ranking systems. *CoRR*, abs/1001.3745, 2010.
[21] 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.
[22] 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.
[23] Gianluca Ciccarelli and Renato Lo Cigno. Collusion in peer-to-peer systems. *Computer Networks*, 55(15):3517 – 3532, 2011.
[24] 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.
[25] 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.
[26] 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.
[27] Yuhong Liu, Yafei Yang, and Yan Lindsay Sun. Detection of collusion behaviors in online reputation systems. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 1368–1372. IEEE, 2008.
[28] 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.
[29] 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.
[30] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond:The Science of Search Engine Rankings*. Princeton University Press, February 2012.
[31] 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.
[32] Erman Ayday, Hanseung Lee, and Faramarz Fekri. An iterative algorithm for trust and reputation management. In *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 3*, ISIT'09, pages 2051–2055, Piscataway, NJ, USA, 2009. IEEE Press.
[33] Ardelio Galletti, Giulio Giunta, and G. Schmid. A mathematical model of collaborative reputation systems. *Int. J. Comput. Math.*, 89(17):2315–2332, November 2012.

(Mohsen Rezvani) Shahrood University of Technology
E-mail address, Mohsen Rezvani: `mrezvani@shahroodut.ac.ir`

(Aleksandar Ignjatovic) UNSW Sydney
E-mail address, Aleksandar Ignjatovic: `ignjat@cse.unsw.edu.au`

(Elisa Bertino) Purdue University
E-mail address, Elisa Bertino: `bertino@cs.purdue.edu`