Robust Reputation-Based Ranking on Bipartite Rating Networks

Rong-Hua Li   Jeffery Xu Yu   Xin Huang   Hong Cheng *

Abstract

With the growth of the Internet and E-commerce, bipartite rating networks are ubiquitous. In such bipartite rating networks, there exist two types of entities: the users and the objects, where users give ratings to objects. A fundamental problem in such networks is how to rank the objects by user's ratings. Although it has been extensively studied in the past decade, the existing algorithms either cannot guarantee convergence, or are not robust to the spammers. In this paper, we propose six new reputation-based algorithms, where the users' reputation is determined by the aggregated difference between the users' ratings and the corresponding objects' rankings. We prove that all of our algorithms converge into a unique fixed point. The time and space complexity of our algorithms are linear w.r.t. the size of the graph, thus they can be scalable to large datasets. Moreover, our algorithms are robust to the spamming users. We evaluate our algorithms using three real datasets. The experimental results confirm the effectiveness, efficiency, and robustness of our algorithms.

1 Introduction

With the development of the Internet and E-commerce, bipartite rating networks, such as the product review systems of Amazon and Epinions, Community Question-Answer systems, Movie-rating systems in IMDB, Video rating system in Youtube, and paper review systems, have become increasingly popular in recent years. In such rating systems, there are two types of entities, users and objects, where users rate objects.

Given a bipartite rating network, a fundamental question is how to rank the objects based on users' ratings? A straightforward method is to average the ratings for each object, and then rank the objects in terms of their average ratings. In this case, the users' ratings are evenly trustworthy. However, in many practical systems, there exist many noisy ratings. Some users may give ratings randomly, some users always give the maximal/ minimal ratings, and some users are malicious users. As discussed in [13], building a robust ranking mechanism for bipartite rating networks is challenging. Many existing ranking algorithms such as PageRank [5], HITS [18], and trust propagation [11] suffer from such noise in the datasets, thus may result in poor ranking performance in this context [8].

To address this issue, many ranking algorithms have been proposed by incorporating users' reputation [23]. The main ideas of these algorithms are that they make use of users' reputation as the weight to eliminate unreasonable ratings influence, and the objects' ranking is determined by the average weighted rating score. The algorithms iteratively refine the users' reputation score and objects' ranking score. In these algorithms, the reputation of the user is measured by the difference between his ratings and the corresponding objects' ranking. As a result, the users whose ratings often differ from those of the other users are assigned less reputation score. However, most of these algorithms [25, 19, 32, 34] cannot guarantee convergence, thus can be hard to be used in practice. Recently, Kerchove et al. [8] propose a provably convergent reputation-based ranking algorithm, but their algorithm is not very robust to the spammers and the parameter of their algorithm is sensitive to the convergent property, thereby it is hard to be determined in practice. In bipartite rating networks, we argue that a good ranking algorithm should be (1) convergent to a unique fixed ranking vector, (2) robust to the spammers, (3) easy to be implemented in practice, and (4) scalable to large datasets.

To achieve these goals, in this paper, we propose six new reputation-based ranking algorithms (the L1-AVG, L2-AVG, L1-MAX, L2-MAX, L1-MIN, and L2-MIN algorithm). In our algorithms, user's reputation is determined by the aggregated difference between his/her ratings and the corresponding objects' rankings. Specifically, we use L1/L2-distance to measure the difference and use *average*, *maximal*, and *minimal* operators as the aggregate function. After obtaining the reputation of the users, we compute the objects' ranking by averaging the reputation-weighted ratings. Our algorithms will iteratively refine the users' reputation score and the objects' ranking score. The final ranking and reputation scores are achieved when the algorithms converge. We evaluate our algorithms using three real rating networks. We measure the performance of our algorithms

*The Chinese University of Hong Kong. Email: {rhli, yu, xhuang, hcheng}@se.cuhk.edu.hk

from two aspects: the effectiveness and the robustness. For the effectiveness, we use the average-ratings based ranking algorithm and three reputation-based ranking algorithms as the baselines, and evaluate the effectiveness by comparing the rank correlation between the proposed algorithms and the baselines using the Kendall Tau [17] metric. The experimental results show that the L1-MIN and L2-MIN algorithm outperform the the state-of-the-art reputation-based ranking algorithm on most datasets, and the performance of other proposed algorithm are also comparable to the state-of-the-art algorithm. This results indicate that our algorithms are effective for ranking. For the robustness, we consider three types of spamming users: (1) users who give random ratings, (2) uses who always give maximal ratings, and (3) users who always give minimal ratings. The experimental results show that our proposed algorithms (L1/L2-AVG and L1/L2-MAX) are significantly more robust than the state-of-the-art algorithm.

The main contributions of this paper are summarized as follows. First, we propose six new reputation-based ranking algorithms. The main advantages of our algorithms are robust to the spammers, convergent into a fixed point with an exponential rate of convergence, and easy to be implemented. Moreover, the time and space complexity of our algorithms is linear w.r.t. the size of the graphs, thus they can be scalable to large datasets. Second, we conduct extensive experiments on three real datasets (Amazon, Epinions, and Bookcrossing), and the experimental results confirm the effectiveness, efficiency and robustness of our algorithms.

The rest of the paper is organized as follows. We introduce the related work in Section 2. We propose six various reputation based ranking algorithms in Section 3, and present the analysis of our algorithms in Section 4. We show the extensive experimental results in Section 5. Section 6 concludes this work.

2 Related Work

Ranking algorithms on bipartite graphs: Since the introduction of PageRank [5] and HITS [18], developing algorithms for ranking nodes in networks have attracted much attention in both research and industry communities. In the past decade, a large number of algorithms, such as personalized PageRank [12] and Co-HITS [9], have been proposed. Most of these algorithms do not consider users' reputation in ranking. To address the users' reputation, Mizzaro [25] proposes a reputation-based ranking algorithm (Mizz) for the assessment of scholarly papers. In [25], the reputation is measured by the root of L1-distance between paper's quality and reader's rating. Subsequently, Yu et al. present an iterative refinement algorithm (YZLM) for bipartite rating networks, with reputation scores for users [32, 19]. In their papers, the reputation is measured by the inverse of square L2-distance between objects' ranking and users' rating. More recently, Zhou et al. [34] propose a similar reputation-based ranking algorithm for rating networks, where the reputation is measured by the correlation coefficient between the user's rating and objects' ranking. However, all of these reputation-based algorithms cannot guarantee convergence, thereby can be hard to be used in practice. A good survey can be found in [23]. To overcome the convergence issue, Kerchove et al. [8] propose a convergent reputation-based ranking algorithm (dKVD) from an optimization perspective. The major drawbacks of their algorithm are (1) the algorithm is not very robust to the spamming users, (2) the rate of convergence of the algorithm is q-linear, and (3) the parameter is sensitive to the convergence of the algorithm, thus it is hard to be determined in practice. More recently, Mishra, et al. [24] propose a trust-based ranking algorithm for finding biased and prestigious nodes in signed networks [21]. However, their work is tailored for the signed networks, and the simple generalization of their algorithm to the bipartite rating networks will ignore the negative bias of the nodes, thus resulting in unfair reputation measurement and poor ranking performance.

Reputation system: Our work is also closely related to the reputation system [28, 33, 27]. The reputation system computes the reputation score for a set of entities (such as the service providers and users in rating networks) through collecting and aggregating feedback about the entities' past behavior. There are two types of reputation systems: the content-driven and user-driven reputation system [7]. The content-driven systems derive entities' feedback from an analysis of all interactions [1, 7, 31]. Such systems include WikiTrust (`www.wikitrust.net`) [1] and Crowdsensus system [7]. The user-driven systems are more related to our work, which is based on user ratings. Such systems include Amazon, Epinions, and ebay systems of product reviews. There are a large body of work for user-driven reputation systems. A survey can be found in [14]. In the following, we list some important work on user-driven reputation systems. Mui et al. [26] propose a computational model of reputation management for E-businesses. Subsequently, Richardson et al. [29] propose an eigenvector based algorithm for trust (or reputation) management in semantic web. Independent to Richardson's work, Kamvar et al. [16] present a similar eigenvector based algorithm, namely Eigentrust, for reputation management in P2P networks. Guha et al. [11] study the problem of propagation of trust and distrust in the networks. Later, Theodorakopoulos et al. [30] study the trust model and trust evaluation metrics from an algebra viewpoint. They use semiring to express trust model and then model the trust evaluation problem as a path problem on a directed graph. Recently, Andersen et al. [2] propose an axiomatic approach for trust based recommendation systems [22]. However, all the above mentioned algorithms are the variants of eigenvector centrality measure [4], thus cannot be used in our problem. Another direction of research on reputation management is based on probabilistic model. Fouss et al. [10] propose a probabilistic reputation model. Their algorithm assumes the providers' reputation following a normal law. However, in some practical systems, this assumption cannot hold, thus may result in poor performance. More recently, Chen et al. [6] propose a bias-smoothed tensor model for reputation management. The major drawback of their method is that it cannot be scalable to large datasets.

3 Robust Reputation-Based Ranking

We model a rating network as a directed and weighted bipartite graph $G = (U, O, R)$. Here, U is a set of U-typed nodes representing users, O is a set of O-typed nodes representing objects to be ranked, and R is a set of directed edges from U-typed nodes to O-typed nodes. We use $|U|$, $|O|$, and $|R|$ to represent the sizes of U, O, and R, respectively. Given a U-typed node u_i, we use O_i to denote the set of objects the user u_i rates ($O_i = \{o_j \mid (u_i, o_j) \in R\}$), and we use I_j to denote the set of users who rate the object o_j ($I_j = \{u_i \mid (u_i, o_j) \in R\}$). The specific rating score of object o_j given by a user u_i is denoted as R_{ij}, which is the weight associated with the edge $(u_i, o_j) \in R$, and the rating score is normalized in the range of [0, 1]. In addition, in this paper, we use c_i and r_j to denote the reputation of user u_i and the ranking score of object o_j, and we also use r and c to denote the ranking vector of objects and the reputation vector of users (Table 1).

Table 1: Notations.

Symbols	Descriptions
r_j	rank of object o_j
r	ranking vector of objects
c_i	reputation of user u_i
c	reputation vector of users
R_{ij}	the rating score of object o_j by user u_i
λ	decay constant, and belongs into (0,1)

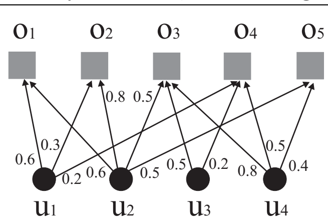

Figure 1: A rating network

An example is shown in Fig. 1. There are 4 users $U = \{u_1, u_2, u_3, u_4\}$, and 5 objects $O = \{o_1, o_2, o_3, o_4, o_5\}$. For example, u_1 rates o_1 0.6 ($R_{11} = 0.6$), o_2 0.3 ($R_{12} = 0.3$), and o_4 0.2 ($R_{14} = 0.2$); u_4 rates o_3 0.8 ($R_{43} = 0.8$), o_4 0.5 ($R_{44} = 0.5$), and o_5 0.4 ($R_{45} = 0.4$). As can be seen from the example, u_1 gives rather low rating scores to the objects, u_4 gives o_3 a high rating score among the three objects he/she gives. Given such a scenario, if we take the average as the ranking score of an object, then the ranking scores for o_1, o_2, o_3, o_4, and o_5 are: $r_1 = (0.6 + 0.6)/2 = 0.6$, $r_2 = (0.3 + 0.8)/2 = 0.55$, $r_3 = (0.5 + 0.5 + 0.8)/3 = 0.6$, $r_4 = (0.2 + 0.2 + 0.5)/3 = 0.3$, and $r_5 = (0.5 + 0.4)/2 = 0.45$. As a result, o_1 and o_3 are the top objects, and o_4 is the last in such ranking. However, such ranking does not take users' reputation into consideration. Consider u_4 who rates o_3 the highest 0.8 among the rating score. However, to the object o_3, the other two users u_2 and u_3 only rate 0.5. In other words, the rating score 0.8 given by the user u_4 to the object o_3 is not consistent with the other rating scores given by the other users who score the same object on average. This implies that the reputation of u_4 (denoted as c_4) can be possibly damaged. In this paper, we investigate how to give objects a ranking score (r_j) by taking the users reputations (c_i) into consideration over a rating network.

There are several key issues. One is the convergence, which we will discuss in detail. The other is the robustness. In many occasions, users may not necessarily spend time on the rating. Some may give a random rating score, some may always give the maximal/minimal rating score, and some may give unreasonable rating scores on purpose. This is considered as noise in this paper. Reconsider the example shown in Fig. 1. Suppose that there exist some additional users who always give all the objects ($o_1, \cdots, o_5$) the highest rating score or the lowest rating score or give random ratings. A ranking algorithm is not robust if the rankings of the objects are significantly affected by such users. In this paper, we propose new ranking algorithms that converge and are robust to such users.

3.1 The Problems of the Existing Solutions. To handle the noise, one solution is to remove such noise using some statistical methods in a preprocessing step and rank objects using an existing ranking algorithm [24]. This method has two major drawbacks: (1) the statistical methods are very hard to be used to detect the users who give random rating scores, and (2) some important rating scores given by some users can be possibly removed by the statistical methods applied. To address this issue, recently, many researchers develop reputation-based ranking algorithms that use users' reputation to weight their rating scores and then aggregate the reputation-weighted rating scores for ranking iteratively. This type of algorithms can mitigate negative influence by assigning a small reputation score to users if they give some unreasonable scores and thus reducing their contribution to the ranking.

Table 2: Ranking scores by Mizz algorithm [25].

Iter.	o_1	o_2	o_3	o_4	o_5
1	0.6000	0.5500	0.6000	0.3000	0.4500
2	0.6000	0.5521	0.5901	0.2907	0.4547
3	0.6000	0.5523	0.5880	0.2886	0.4554
4	0.6000	0.5523	0.5875	0.2881	0.4555
5	0.6000	0.5523	0.5874	0.2880	0.4556

Table 3: Ranking scores by YZLM algorithm [32].

Iter.	o_1	o_2	o_3	o_4	o_5
1	0.6000	0.5500	0.6000	0.3000	0.4500
2	0.6000	0.5658	0.5775	0.2805	0.4548
3	0.6000	0.5806	0.5628	0.2673	0.4594
4	0.6000	0.5949	0.5529	0.2581	0.4627
5	0.6000	0.6087	0.5459	0.2515	0.4654

Table 4: Ranking scores by dKVD algorithm [8].

Iter.	o_1	o_2	o_3	o_4	o_5
1	0.6000	0.5500	0.6000	0.3000	0.4500
2	0.6000	0.6621	0.5000	0.2000	0.5000
3	0.6000	0.6745	0.5000	0.2000	0.5000
4	0.6000	0.6908	0.5000	0.2000	0.5000
5	0.6000	0.7147	0.5000	0.2000	0.5000

In the existing reputation-based algorithms, the reputation is based on the same intuition: "the users whose ratings often differ from those of other users are assigned less reputation score". However, the existing reputation-based ranking algorithms [19, 32, 25] do not guarantee convergence of the algorithm as pointed out in [8, 23]. Consider the rating network in Fig. 1. Table 2 and Table 3 show the ranking scores by the Mizz algorithm [25] and the YZLM algorithm [32] in the first 5 iterations, respectively. As can be seen, the ranking scores of objects either monotonically increases or decreases as the iteration increases. We set the maximal iteration to 30, and we find that both Mizz and YZLM algorithm do not converge after 30 iterations. This results suggest that both Mizz and YZLM algorithm do not converge. As an exception, in [8], the authors proposed a provably convergent reputation-based ranking algorithm dKVD. However, the convergent rate of the dKVD algorithm is q-linear. Indeed, as shown in Table 4, the dKVD algorithm does not converge in 5 iterations on the same rating network (Fig. 1). The parameter is sensitive to the convergence of the algorithm and is hard to be determined in practice. In addition, the dKVD algorithm is not robust to the spamming users as shown in our experiments.

To summarize, as discussed in [13], developing a robust and provably convergent reputation-based ranking algorithm is challenging.

3.2 Our Solutions. In this paper, we propose six provably convergent reputation-based ranking algorithms, based on the basic idea "the users whose ratings often differ from those of other users are assigned less reputation score". Moreover, most of our algorithms (L1/L2-AVG and L1/L2-MAX) are quite robust as shown in the experiments. The two main components in our algorithms are, namely, the distance function and the aggregate operator. Consider a user u_i and an object o_j. The distance function is used to measure the difference between the user's rating score R_{ij} and the corresponding object's ranking score r_j. We focus on two distance functions: L1-distance $|R_{ij} - r_j|$, and the square of L2-distance $(R_{ij} - r_j)^2$. In the following, a distance function is either L1-distance or square of L2-distance. As also indicated in our experiments, L1-distance based algorithms are more robust than the square of L2-distance based algorithms, because L1-distance is more robust than L2-distance for the noisy data. The aggregate operators, AVG, MAX, and MIN, are used to determine the reputation c_i of user u_i by averaging the differences, taking the maximum difference, and taking the minimum difference, respectively. Intuitively, the aggregator MAX heavily penalizes a user who gives even one unreasonable rating score. Here the unreasonable rating score means that the rating score is significantly different from the ranking score computed. On the other hand, the aggregator MIN gives a user minimum penalty even though the user may give many unreasonable rating scores. As an extreme case, as long as a user gives only one reasonable rating, then his/her reputation will be 1 (highest reputation). The aggregator AVG averages over all the differences, thus it would be a good tradeoff between the aggregator MAX and MIN. We summarize the proposed algorithms in Table 5, and discuss them in details below.

The L1-AVG algorithm: The reputation c_i of a user u_i is defined by using the L1-distance to measure the rating difference and employing AVG to aggregate user's rating difference. Formally, the reputation c_i of a user u_i is defined as

$$c_i = 1 - \frac{\lambda}{|O_i|} \sum_{o_j \in O_i} |R_{ij} - r_j|, \tag{3.1}$$

where O_i is the set of objects rated by a user u_i, R_{ij} is the scaled rating score of the user u_i to an object o_j, r_j is the ranking score of the object o_j, and λ is a decay constant in $(0, 1)$. In L1-AVG, the reputation score of user u_i is inversely proportional to the L1-distance between his/her rating vector and the corresponding

Table 5: Summary of the proposed algorithms.

Algorithm	Distance	Aggregate	Convergence	Convergence Rate	Time complexity	Space complexity
L1-AVG	L1	Average	Yes	Exponential	$O(k\|R\|)$	$O(\|R\| + \|U\| + \|O\|)$
L2-AVG	L2	Average	Yes	Exponential	$O(k\|R\|)$	$O(\|R\| + \|U\| + \|O\|)$
L1-MAX	L1	Max	Yes	Exponential	$O(k\|R\|)$	$O(\|R\| + \|U\| + \|O\|)$
L2-MAX	L2	Max	Yes	Exponential	$O(k\|R\|)$	$O(\|R\| + \|U\| + \|O\|)$
L1-MIN	L1	Min	Yes	Exponential	$O(k\|R\|)$	$O(\|R\| + \|U\| + \|O\|)$
L2-MIN	L2	Min	Yes	Exponential	$O(k\|R\|)$	$O(\|R\| + \|U\| + \|O\|)$

objects' ranking scores. In other words, the larger L1 distance, the smaller the reputation score.

Our L1-AVG algorithm takes an iterative approach by applying the reputation defined in Eq. (3.1) to weight the user's rating score and then refining the objects' ranking and users' reputation as follows.

$$\begin{cases} r_j^{k+1} = \frac{1}{|I_j|} \sum\limits_{u_i \in I_j} R_{ij} c_i^k \\ c_i^{k+1} = 1 - \frac{\lambda}{|O_i|} \sum\limits_{o_j \in O_i} |R_{ij} - r_j^{k+1}| \end{cases} \tag{3.2}$$

Here, k indicates the k-th iteration, and c_i^k is the user's reputation c_i in the k-th iteration, and r_j^k is the rank of object o_j in the k-th iteration. Below, all the equations and algorithms are defined in a similar fashion. We omit explanation unless necessary.

Table 6 shows the ranking scores of the objects in Fig. 1. The results show that the L1-AVG algorithm converges in 4 iterative steps, as the rate of convergence of this algorithm is exponential.

Table 6: Ranking scores by L1-AVG algorithm.

Iter.	o_1	o_2	o_3	o_4	o_5
1	0.6000	0.5500	0.6000	0.3000	0.4500
2	0.5935	0.5443	0.5927	0.2961	0.4445
3	0.5935	0.5442	0.5927	0.2961	0.4444
4	0.5935	0.5442	0.5927	0.2961	0.4444

The L2-AVG algorithm: Unlike the L1-AVG algorithm, the L2-AVG algorithm uses the square of L2-distance.

$$c_i = 1 - \frac{\lambda}{2|O_i|} \sum_{o_j \in O_i} (R_{ij} - r_j)^2. \tag{3.3}$$

Note that the additional constant coefficient (1/2) is used to ensure the convergence of the algorithm. The corresponding iterative system of the L2-AVG algorithm is formulated as follows.

$$\begin{cases} r_j^{k+1} = \frac{1}{|I_j|} \sum\limits_{u_i \in I_j} R_{ij} c_i^k \\ c_i^{k+1} = 1 - \frac{\lambda}{2|O_i|} \sum\limits_{o_j \in O_i} (R_{ij} - r_j^{k+1})^2 \end{cases} \tag{3.4}$$

The L1-MAX algorithm: The reputation c_i of a user u_i is defined by using L1-distance to measure the rating difference and employing MAX to aggregate user's rating difference.

$$c_i = 1 - \lambda \max_{o_j \in O_i} |R_{ij} - r_j| \tag{3.5}$$

The L1-MAX algorithm takes an iterative approach by applying the reputation defined in Eq. (3.5) to weight the user's rating score and then refining the objects' ranking and users' reputation as follows.

$$\begin{cases} r_j^{k+1} = \frac{1}{|I_j|} \sum\limits_{u_i \in I_j} R_{ij} c_i^k \\ c_i^{k+1} = 1 - \lambda \max\limits_{o_j \in O_i} |R_{ij} - r_j^{k+1}|. \end{cases} \tag{3.6}$$

The L2-MAX algorithm: Instead of L1-distance, the L2-MAX algorithm uses the square of L2-distance.

$$c_i = 1 - \frac{\lambda}{2} \max_{o_j \in O_i} (R_{ij} - r_j)^2. \tag{3.7}$$

Also, the additional coefficient (1/2) is used to guarantee the convergence. The corresponding iterative system of the L2-MAX algorithm is given as follows.

$$\begin{cases} r_j^{k+1} = \frac{1}{|I_j|} \sum\limits_{u_i \in I_j} R_{ij} c_i^k \\ c_i^{k+1} = 1 - \frac{\lambda}{2} \max\limits_{o_j \in O_i} (R_{ij} - r_j^{k+1})^2. \end{cases} \tag{3.8}$$

The L1-MIN algorithm: The reputation c_i of a user u_i is defined by using L1-distance to measure the rating difference and employing MIN to aggregate user's rating difference.

$$c_i = 1 - \lambda \min_{o_j \in O_i} |R_{ij} - r_j| \tag{3.9}$$

The L1-MIN algorithm takes an iterative approach by applying the reputation defined in Eq. (3.9) to weight the user's rating score and then refining the objects' ranking and users' reputation as follows.

$$\begin{cases} r_j^{k+1} = \frac{1}{|I_j|} \sum\limits_{u_i \in I_j} R_{ij} c_i^k \\ c_i^{k+1} = 1 - \lambda \min\limits_{o_j \in O_i} |R_{ij} - r_j^{k+1}|. \end{cases} \tag{3.10}$$

The L2-MIN algorithm: We develop the L2-distance based minimal reputation algorithm as follows.

$$c_i = 1 - \frac{\lambda}{2} \min_{o_j \in O_i} (R_{ij} - r_j)^2. \tag{3.11}$$

Like other L2-distance based algorithms, the additional coefficient (1/2) is used to ensure the convergence. The corresponding iterative system is given as follows.

$$(3.12)\quad \begin{cases} r_j^{k+1} = \frac{1}{|I_j|}\sum_{u_i \in I_j} R_{ij} c_i^k \\ c_i^{k+1} = 1 - \frac{\lambda}{2}\min_{o_j \in O_i} (R_{ij} - r_j^{k+1})^2. \end{cases}$$

4 Analysis of the proposed algorithms

In this section, we analyze the convergent properties, the rate of convergence, and the complexity of the proposed algorithms. Since the proofs of convergent properties of the L2-distance based algorithms (L2-AVG, L2-MAX, and L2-MIN) are similar to the L1-distance based algorithms (L1-AVG, L1-MAX, L1-MIN), we focus on analyzing the properties of L1-distance based algorithms, and omit the details for L2-distance based algorithms.

Convergence of the L1-AVG algorithm: We first prove the L1-AVG algorithm converges to a fixed point, and then we prove the uniqueness of the fixed point. To prove the convergence, we give Lemma 4.1.

LEMMA 4.1. *In L1-AVG algorithm, let* $|r_\beta^1 - r_\beta^0| = \max_j |r_j^1 - r_j^0|$, *then for any object* o_j, *we have* $|r_j^{k+1} - r_j^k| \le \lambda^k |r_\beta^1 - r_\beta^0|$.

Proof. We prove it by induction. First, we prove the lemma holds when $k = 1$.

$$\begin{aligned}
|r_j^2 - r_j^1| &= |\tfrac{1}{|I_j|}\sum_{u_i \in I_j} c_i^1 R_{ij} - \tfrac{1}{|I_j|}\sum_{u_i \in I_j} c_i^0 R_{ij}| \\
&= \tfrac{1}{|I_j|}|\sum_{u_i \in I_j}(\tfrac{\lambda}{|O_i|}\sum_{o_\gamma \in O_i}|R_{i\gamma} - r_\gamma^0| - \tfrac{\lambda}{|O_i|}\sum_{o_\gamma \in O_i}|R_{i\gamma} - r_\gamma^1|)R_{ij}| \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|(|R_{i\gamma} - r_\gamma^0| - |R_{i\gamma} - r_\gamma^1|)|R_{ij} \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|r_\gamma^1 - r_\gamma^0|R_{ij}) \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|r_\beta^1 - r_\beta^0|R_{ij}) \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}|r_\beta^1 - r_\beta^0|R_{ij}) \\
&\le \lambda|r_\beta^1 - r_\beta^0|
\end{aligned}$$

Second, we assume the lemma holds when $k = t$, and show the lemma holds when $k = t+1$.

$$\begin{aligned}
&|r_j^{t+2} - r_j^{t+1}| \\
&= \tfrac{1}{|I_j|}|\sum_{u_i \in I_j}(\tfrac{\lambda}{|O_i|}\sum_{o_\gamma \in O_i}|R_{i\gamma} - r_\gamma^t| \\
&\qquad - \tfrac{\lambda}{|O_i|}\sum_{o_\gamma \in O_i}|R_{i\gamma} - r_\gamma^{t+1}|)R_{ij}| \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|(|R_{i\gamma} - r_\gamma^t| - |R_{i\gamma} - r_\gamma^{t+1}|)|R_{ij}) \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|r_\gamma^{t+1} - r_\gamma^t|R_{ij}) \\
&\le \tfrac{\lambda^{t+1}}{|I_j|}\sum_{u_i \in I_j}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|r_\beta^1 - r_\beta^0|R_{ij}) \\
&\le \lambda^{t+1}|r_\beta^1 - r_\beta^0|
\end{aligned}$$

where the third inequality holds by the induction assumption, and the last inequality holds due to $R_{ij} \in [0, 1]$. This completes the proof. ☐

With Lemma 4.1, we prove the following convergence theorem.

THEOREM 4.1. *The L1-AVG algorithm converges to a unique fixed point.*

Proof. We first prove the L1-AVG algorithm converges into a fixed point, and then prove the uniqueness of the fixed point. Let $|r_\beta^1 - r_\beta^0| = \max_j |r_j^1 - r_j^0|$, and $|r_\beta^1 - r_\beta^0| \neq 0$. For $\varepsilon > 0$, there exists N such that

$$\lambda^N \le \frac{(1-\lambda)\varepsilon}{|r_\beta^1 - r_\beta^0|}$$

Then, for any $s > t \ge N$, we have

$$\begin{aligned}
|r_j^s - r_j^t| &\le |r_j^s - r_j^{s-1}| + |r_j^{s-1} - r_j^{s-2}| + \cdots + |r_j^{t+1} - r_j^t| \\
&\le \lambda^{s-1}|r_\beta^1 - r_\beta^0| + \lambda^{s-2}|r_\beta^1 - r_\beta^0| + \cdots + \lambda^t|r_\beta^1 - r_\beta^0| \\
&\le |r_\beta^1 - r_\beta^0|\lambda^t \sum_{k=0}^{s-t-1}\lambda^k \\
&< |r_\beta^1 - r_\beta^0|\lambda^t \sum_{k=0}^{\infty}\lambda^k \\
&= |r_\beta^1 - r_\beta^0|\lambda^t \tfrac{1}{1-\lambda} \\
&\le |r_\beta^1 - r_\beta^0|\lambda^N \tfrac{1}{1-\lambda} \\
&\le \varepsilon,
\end{aligned}$$

where the first inequality holds due to the triangle inequality, the second inequality is due to Lemma 4.1. By Cauchy convergence theorem, we conclude that the sequence r_j^k converges to a fixed point.

For the uniqueness, we prove it by a contradiction. Suppose the iterative system (Eq. (3.2)) has at least two fixed points. Let $r^{(1)}$ and $r^{(2)}$ be the two fixed points, and $M = |r_\beta^{(1)} - r_\beta^{(2)}| = \max_j |r_j^{(1)} - r_j^{(2)}|$. Then, we have

$$\begin{aligned}
M &= |r_\beta^{(1)} - r_\beta^{(2)}| = \tfrac{1}{|I_\beta|}|\sum_{u_i \in I_\beta}(c_i^{(1)} - c_i^{(2)})R_{i\beta}| \\
&= \tfrac{1}{|I_\beta|}|\sum_{u_i \in I_\beta}(\tfrac{\lambda}{|O_i|}\sum_{o_\gamma \in O_i}|R_{i\gamma} - r_\gamma^{(1)}| - \tfrac{\lambda}{|O_i|}\sum_{o_\gamma \in O_i}|R_{i\gamma} - r_\gamma^{(2)}|)R_{i\beta}| \\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i \in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|r_\gamma^{(1)} - r_\gamma^{(2)}|R_{i\beta}) \\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i \in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma \in O_i}|r_\beta^{(1)} - r_\beta^{(2)}|R_{i\beta}) \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j}|r_\beta^{(1)} - r_\beta^{(2)}|R_{ij}) \\
&\le \lambda|r_\beta^{(1)} - r_\beta^{(2)}| \\
&= \lambda M.
\end{aligned}$$

Since $\lambda \in (0, 1)$, we have $M < M$, which is a contradiction. This completes the proof. ☐

The rate of convergence of L1-AVG algorithm: We show the rate of convergence of the L1-AVG algorithm is exponential. By definition, we have the following lemma.

LEMMA 4.2. *For any iterative steps a and b,*

$$|r^a_\beta - r^b_\beta| = \max_j |r^a_j - r^b_j| \le 1.$$

Then, we can prove that the L1-AVG algorithm will converge into the fixed point in exponential rate.

LEMMA 4.3. *In the L1-AVG algorithm, let r^∞ and c^∞ denote the final ranking vector of the objects and the final reputation vector of the users, respectively. And let $|r^\infty_\beta - r^k_\beta| = \max_j |r^\infty_j - r^k_j|$. Then, we have $|r^\infty_\beta - r^k_\beta| \le \lambda^k$.*

Proof. We prove the lemma by induction. First, for $k = 1$, we have

$$\begin{aligned}
&|r^\infty_\beta - r^1_\beta| = |\tfrac{1}{|I_\beta|}\sum_{u_i\in I_\beta} c^\infty_i R_{i\beta} - \tfrac{1}{|I_\beta|}\sum_{u_i\in I_\beta} c^0_i R_{i\beta}|\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma\in O_i}|(|R_{i\gamma}-r^0_\gamma| - |R_{i\gamma}-r^\infty_\gamma|)|R_{i\beta})\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma\in O_i}|r^\infty_\gamma - r^0_\gamma|R_{i\beta})\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma\in O_i}|r^\infty_\beta - r^0_\beta|R_{i\beta})\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}|r^\infty_\beta - r^0_\beta|R_{ij})\\
&\le \lambda|r^\infty_\beta - r^0_\beta|\\
&\le \lambda,
\end{aligned}$$

where the last inequality is due to Lemma 4.2. Second, suppose the lemma holds when $k = t$. Then, for $k = t + 1$, we have

$$\begin{aligned}
&|r^\infty_\beta - r^{t+1}_\beta| = |\tfrac{1}{|I_\beta|}\sum_{u_i\in I_\beta} c^\infty_i R_{i\beta} - \tfrac{1}{|I_\beta|}\sum_{u_i\in I_\beta} c^t_i R_{i\beta}|\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma\in O_i}|(|R_{i\gamma}-r^t_\gamma| - |R_{i\gamma}-r^\infty_\gamma|)|R_{i\beta})\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma\in O_i}|r^\infty_\gamma - r^t_\gamma|R_{i\beta})\\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i\in I_\beta}(\tfrac{1}{|O_i|}\sum_{o_\gamma\in O_i}|r^\infty_\beta - r^t_\beta|R_{i\beta})\\
&\le \tfrac{\lambda^{t+1}}{|I_\beta|}\sum_{u_i\in I_\beta}|r^\infty_\beta - r^0_\beta|R_{ij})\\
&\le \lambda^{t+1}|r^\infty_\beta - r^0_\beta|\\
&\le \lambda^{t+1}.
\end{aligned}$$

Thus, we have $|r^\infty_\beta - r^k_\beta| \le \lambda^k$. This completes the proof. □

Assume r_j is the true ranking score of the object o_j, we can determine the number of iterations k for the algorithm to converge to the true r_j. Formally, for $\varepsilon \to 0$, $|r_j - r^k_j| \le \varepsilon$. In terms of Lemma 4.3, we can set

$$k = \log_\lambda \varepsilon. \tag{4.13}$$

Thus, the number of iterations of the L1-AVG algorithm (k) is a small constant. In our experiments, $k = 10$ is enough to guarantee the convergence of the algorithm.

Complexity of L1-AVG algorithm: The time complexity of computing the ranking of an object in a single iteration is $O(|\bar{I}||\bar{O}|)$, where $|\bar{I}|$ and $|\bar{O}|$ denote the average in-degree of the objects and the average out-degree of the users, respectively. By analysis, the amortized cost in a single iteration is $O(|R|)$, where $|R|$ is the total number of edges (ratings) in the bipartite graph. Consequently, for k iterations, the total running time of the L1-AVG algorithm is $O(k|R|)$, where k is a small constant. The space complexity is $O(|R| + |U| + |O|)$, because we only need to store the ranking vector r, the reputation vector c, and the bipartite graph G.

Convergence of L1-MAX algorithm: Similar to the proof of L1-AVG algorithm, we first prove the error bound for the two consecutive iterations and then use Cauchy convergence theorem to prove the convergence. First, we show the following two lemmas.

LEMMA 4.4. *Let $r^{(1)}$ and $r^{(2)}$ be two ranking vectors, and $|r^{(1)}_\beta - r^{(2)}_\beta| = \max_j |r^{(1)}_j - r^{(2)}_j|$. Then, for any user u_i with the corresponding object set O_i, we have $|\max_{o_j\in O_i}|R_{ij} - r^{(1)}_j| - \max_{o_j\in O_i}|R_{ij} - r^{(2)}_j|| \le |r^{(1)}_\beta - r^{(2)}_\beta|$.*

Proof. Let $|R_{i\gamma} - r^{(1)}_\gamma| = \max_{o_j\in O_i}|R_{ij} - r^{(1)}_j|$, and $|R_{i\theta} - r^{(2)}_\theta| = \max_{o_j\in O_i}|R_{ij} - r^{(2)}_j|$. Obviously, $|\max_{o_j\in O_i}|R_{ij} - r^{(1)}_j| - \max_{o_j\in O_i}|R_{ij} - r^{(2)}_j||$ either equals to $\max_{o_j\in O_i}|R_{ij} - r^{(1)}_j| - \max_{o_j\in O_i}|R_{ij} - r^{(2)}_j|$ or equals to $\max_{j\in O_i}|R_{ij} - r^{(2)}_j| - \max_{j\in O_i}|R_{ij} - r^{(1)}_j|$. Since

$$\begin{aligned}
&\max_{o_j\in O_i}|R_{ij} - r^{(1)}_j| - \max_{o_j\in O_i}|R_{ij} - r^{(2)}_j|\\
&\le |R_{i\gamma} - r^{(1)}_\gamma| - |R_{i\gamma} - r^{(2)}_\gamma|\\
&\le |r^{(1)}_\gamma - r^{(2)}_\gamma|\\
&\le |r^{(1)}_\beta - r^{(2)}_\beta|
\end{aligned}$$

and

$$\begin{aligned}
&\max_{o_j\in O_i}|R_{ij} - r^{(2)}_j| - \max_{o_j\in O_i}|R_{ij} - r^{(1)}_j|\\
&\le |R_{i\theta} - r^{(2)}_\theta| - |R_{i\theta} - r^{(1)}_\theta|\\
&\le |r^{(1)}_\theta - r^{(2)}_\theta|\\
&\le |r^{(1)}_\beta - r^{(2)}_\beta|,
\end{aligned}$$

thus the lemma holds. □

LEMMA 4.5. *In the L1-MAX algorithm, let $|r^1_\beta - r^0_\beta| = \max_j |r^1_j - r^0_j|$, then for any object o_j, we have $|r^{k+1}_j - r^k_j| \le \lambda^k|r^1_\beta - r^0_\beta|$.*

Proof. We prove it by induction. For $k = 1$, we have

$$\begin{aligned}
&|r^2_j - r^1_j| = |\tfrac{1}{|I_j|}\sum_{u_i\in I_j} c^1_i R_{ij} - \tfrac{1}{|I_j|}\sum_{u_i\in I_j} c^0_i R_{ij}|\\
&= \tfrac{1}{|I_j|}|\sum_{u_i\in I_j}\lambda(\max_{o_\gamma\in O_i}|R_{i\gamma} - r^0_\gamma| - \max_{o_\gamma\in O_i}|R_{i\gamma} - r^1_\gamma|)R_{ij}|\\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i\in I_j}|\max_{o_\gamma\in O_i}|R_{i\gamma} - r^0_\gamma| - \max_{o_\gamma\in O_i}|R_{i\gamma} - r^1_\gamma||R_{ij}\\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i\in I_j}|r^1_\beta - r^0_\beta|R_{ij}\\
&\le \lambda|r^1_\beta - r^0_\beta|,
\end{aligned}$$

where the second inequality is due to Lemma 4.4. Suppose the lemma holds when $k = t$, we prove the lemma holds, for $k = t+1$. Let $|r_\eta^{t+1} - r_\eta^t| = \max_j |r_j^{t+1} - r_j^t|$. Then, we have

$$
\begin{aligned}
&|r_j^{t+2} - r_j^{t+1}| = |\tfrac{1}{|I_j|}\sum_{u_i \in I_j} c_i^{t+1} R_{ij} - \tfrac{1}{|I_j|}\sum_{u_i \in I_j} c_i^t R_{ij}| \\
&= \tfrac{1}{|I_j|}|\sum_{u_i \in I_j} \lambda(\max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^t| - \max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^{t+1}|)R_{ij}| \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j} |\max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^t| - \max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^{t+1}||R_{ij} \\
&\le \tfrac{\lambda}{|I_j|}\sum_{u_i \in I_j} |r_\eta^{t+1} - r_\eta^t|R_{ij} \\
&\le \tfrac{\lambda^{t+1}}{|I_j|}\sum_{u_i \in I_j} |r_\beta^1 - r_\beta^0|R_{ij} \\
&\le \lambda^{t+1}|r_\beta^1 - r_\beta^0|,
\end{aligned}
$$

where the second inequality holds due to Lemma 4.4, and the third inequality is due to the induction assumption. This completes the proof. □

With Lemma 4.5, we prove the convergence of the L1-MAX algorithm.

THEOREM 4.2. *The L1-MAX algorithm converges to a unique fixed point.*

Proof. For the convergence of the L1-MAX algorithm, it can be proved in a similar way to prove the convergence of L1-AVG algorithm, thus we omit it. For the uniqueness, we prove it by contradiction. Assume L1-MAX converges to at least two fixed points. Let $r^{(1)}$ and $r^{(2)}$ be the two fixed points, and let $M = |r_\beta^{(1)} - r_\beta^{(2)}| = \max_j |r_j^{(1)} - r_j^{(2)}|$. Then, we have

$$
\begin{aligned}
&M = |r_\beta^{(1)} - r_\beta^{(2)}| \\
&= \tfrac{1}{|I_\beta|}|\sum_{u_i \in I_\beta} (c_i^{(1)} - c_i^{(2)})R_{i\beta}| \\
&= \tfrac{1}{|I_\beta|}|\sum_{u_i \in I_\beta} \lambda(\max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^{(1)}| - \max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^{(2)}|)R_{i\beta}| \\
&\le \tfrac{\lambda}{|I_\beta|}\sum_{u_i \in I_\beta} |\max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^{(1)}| - \max_{o_\gamma \in O_i} |R_{i\gamma} - r_\gamma^{(2)}||R_{i\beta} \\
&\le \tfrac{\lambda}{|I_j|}\sum_{i \in I_j} |r_\beta^{(1)} - r_\beta^{(2)}|R_{ij} \\
&\le \lambda|r_\beta^{(1)} - r_\beta^{(2)}| \\
&= \lambda M.
\end{aligned}
$$

Since $\lambda \in (0, 1)$, we get a contradiction. Therefore, the L1-MAX algorithm converges into a unique fixed point. □

The rate of convergence of L1-MAX algorithm: The rate of convergence of the L1-MAX algorithm is exponential, as given in Lemma 4.6, which can be proved in a similar way to prove Lemma 4.3. We omit the proof.

LEMMA 4.6. *In the L1-MAX algorithm, let* $|r_\beta^\infty - r_\beta^k| = \max_j |r_j^\infty - r_j^k|$. *Then, we have* $|r_\beta^\infty - r_\beta^k| \le \lambda^k$.

Complexity of L1-MAX algorithm: Similar to the L1-AVG algorithm, the amortized time complexity of L1-MAX algorithm is $O(k|R|)$. With the exponential rate of convergence of the L1-MAX algorithm, k typically is very small in practice. Hence, the time complexity of the L1-MAX algorithm is linear w.r.t. the size of the graph. In addition, the space complexity of the L1-MAX algorithm is $O(|R| + |U| + |O|)$.

Convergence of L1-MIN algorithm: First, we prove the following lemma.

LEMMA 4.7. *Let* $r^{(1)}$ *and* $r^{(2)}$ *be two ranking vectors, and* $|r_\beta^{(1)} - r_\beta^{(2)}| = \max_j |r_j^{(1)} - r_j^{(2)}|$. *Then, for any user* u_i *with the corresponding* O_i, *we have* $|\min_{o_j \in O_i} |R_{ij} - r_j^{(1)}| - \min_{o_j \in O_i} |R_{ij} - r_j^{(2)}|| \le |r_\beta^{(1)} - r_\beta^{(2)}|$.

Proof. Let $|R_{i\gamma} - r_\gamma^{(1)}| = \min_{o_j \in O_i} |R_{ij} - r_j^{(1)}|$, and $|R_{i\theta} - r_\theta^{(2)}| = \min_{o_j \in O_i} |R_{ij} - r_j^{(2)}|$. Apparently, $|\min_{o_j \in O_i} |R_{ij} - r_j^{(1)}| - \min_{o_j \in O_i} |R_{ij} - r_j^{(2)}||$ either equals to $\min_{o_j \in O_i} |R_{ij} - r_j^{(1)}| - \min_{o_j \in O_i} |R_{ij} - r_j^{(2)}|$ or equals to $\min_{o_j \in O_i} |R_{ij} - r_j^{(2)}| - \min_{o_j \in O_i} |R_{ij} - r_j^{(1)}|$. Since

$$
\begin{aligned}
&\min_{o_j \in O_i} |R_{ij} - r_j^{(1)}| - \min_{o_j \in O_i} |R_{ij} - r_j^{(2)}| \\
&\le |R_{i\theta} - r_\theta^{(1)}| - |R_{i\theta} - r_\theta^{(2)}| \\
&\le |r_\theta^{(1)} - r_\theta^{(2)}| \\
&\le |r_\beta^{(1)} - r_\beta^{(2)}|,
\end{aligned}
$$

and

$$
\begin{aligned}
&\min_{j \in O_i} |R_{ij} - r_j^{(2)}| - \min_{j \in O_i} |R_{ij} - r_j^{(1)}| \\
&\le |R_{i\gamma} - r_\gamma^{(2)}| - |R_{i\gamma} - r_\gamma^{(1)}| \\
&\le |r_\gamma^{(1)} - r_\gamma^{(2)}| \\
&\le |r_\beta^{(1)} - r_\beta^{(2)}|
\end{aligned}
$$

thus the lemma holds. □

LEMMA 4.8. *In the L1-MIN algorithm, let* $|r_\beta^1 - r_\beta^0| = \min_j |r_j^1 - r_j^0|$, *then for any object* o_j, *we have* $|r_j^{k+1} - r_j^k| \le \lambda^k |r_\beta^1 - r_\beta^0|$.

THEOREM 4.3. *The L1-MIN algorithm converges to a unique fixed point.*

Similarly, the proof of Lemma 4.8 and Theorem 4.3 are similar to the proof of Lemma 4.5 and Theorem 4.2 respectively, thus we omit the proofs.

The rate of convergence of L1-MIN algorithm: We can prove that the rate of convergence of L1-MIN algorithm is exponential. We present the main result in the following lemma.

LEMMA 4.9. *In the L1-MIN algorithm, let* $|r_\beta^\infty - r_\beta^k| = \max_j |r_j^\infty - r_j^k|$. *Then, we have* $|r_\beta^\infty - r_\beta^k| \le \lambda^k$.

Complexity of L1-MIN algorithm: Similar to the L1-AVG and L1-MAX algorithms, the amortized time complexity of the L1-MIN algorithm is $O(k|R|)$. Due to the exponential rate of convergence of the L1-MIN algorithm, k is very small in practice. As a result, the time complexity of the L1-MIN algorithm is linear w.r.t. the size of the graph. Additionally, the space complexity of the L1-MIN algorithm is $O(|R| + |U| + |O|)$.

5 Experiments

In this section, we evaluate the effectiveness, robustness, and efficiency of the proposed algorithms, and report our findings.

Datasets: We conduct our experiments on three real datasets. (1) Amazon dataset: we collect the product review information of Amazon dataset from Stanford network analysis data collections [20]. This dataset contains 1,555,170 users, 402,724 objects, and 6,359,182 ratings. We scale the original rating score ([0, 5]) into [0, 1]. (2) Bookcrossing dataset: Bookcrossing (`www.bookcrossing.com`) is an online book club website. Users in Bookcrossing can give ratings to some books. In our experiments, we use the dataset collected by Ziegler et al. [35]. This dataset includes 125,274 users, 340,545 books, and 1,071,158 ratings. The original ratings ([0, 10]) are scaled into [0, 1]. (3) Epinions dataset: Epinions (`www.epinions.com`) is a product review website. Users in Epinions can give ratings to some products. In our experiments, we use the dataset crawled by Massa et al. [22]. The dataset contains 40,163 users, 139,738 products, and 1,149,766 ratings, where the original ratings ([0, 5]) are scaled into [0, 1].

Baselines: We compare our proposed algorithms with five baselines: (1) Arithmetic average algorithm (AA) which ranks the objects by the average ratings of the objects, (2) HITS algorithm [18], (3) Mizz algorithm [25], (4) YZLM algorithm [32], and (5) dKVD algorithm [8]. The original HITS algorithm works on unweighed graph. In our experiments, we use the HITS algorithm proposed in [9] that work on weighted graphs. Mizz [25] is an iterative reputation-based ranking algorithm. The user's reputation in Mizz is determined by the root of L1-distance between user's ratings and the corresponding objects' ranking. Mizz cannot guarantee convergence. YZLM is also an iterative reputation-based ranking algorithm, and is the state-of-the-art algorithm as indicated in [23]. However, YZLM cannot guarantee convergence. The dKVD algorithm is also an iterative reputation-based ranking algorithm, which converges but in q-linear rate. Additionally, dKVD is not very robust to the spamming users and its parameters are hard to be determined in practice.

Methodology and evaluation metrics: We evaluate our algorithms in two aspects: the effectiveness and the robustness. How to get the ground truth is an important problem for evaluating the effectiveness of the ranking algorithms. In practice, no ground truth rankings are known in advance. To evaluate the effectiveness of our algorithms, we use the rank yielded by AA as the ground truth. The reason is of twofold: (1) AA uses the meta-information (i.e. average ratings) for ranking, which is a popular method for generating ground truth for ranking in IR community [15], and (2) although AA ignores user's reputation in ranking, some practical systems use AA to rank the objects as it is easy to be implemented [8]. After obtaining the ground truth, we apply the widely used Kendall Tau metric [17] to measure the rank correlation between the rank yielded by a ranking algorithm and the ground truth. In addition, we compare our proposed methods with other baselines in Kendall Tau metric.

To evaluate the robustness of our algorithms, we first add random spamming users into the original datasets to generate noisy datasets. Here the spamming users include (1) the users who give completely random ratings, (2) the users who only give maximal ratings, and (3) the users who only give minimal ratings. Second, we run our algorithms and the baselines on both the original datasets and the noisy datasets. Third, we compute the Kendall Tau for each algorithm. Here the Kendall Tau is computed between two ranks which are yielded by the ranking algorithm on the original datasets and the noisy datasets, respectively. Finally, we compare the Kendall Tau among all algorithms. Intuitively, the larger Kendall Tau of the algorithm implies the algorithm is more robust.

Parameter settings and experimental environment: In our proposed algorithms, there is only one parameter λ, which is the decay constant. In all our experiments without otherwise specified, we set $\lambda = 0.1$. For all parameters of the baselines, we use the same as given in their original papers respectively. We set the maximal iterative steps to 30. All the experiments are conducted on a Window Server 2008 with 4x6-core Intel Xeon 2.66 Ghz CPU, and 128G memory. All algorithms are implemented by MATLAB 2010a and VC 6.0.

5.1 Experimental results

We report our experimental results as follows.

Effectiveness: Fig. 2, Fig. 3, and Fig. 4 show the rank correlation between two ranks by a ranking algorithm and AA on Amazon, Bookcrossing, and Epinions datasets, respectively.

As shown in Fig. 2, Fig. 3, and Fig. 4, we find that the L1-MIN and L2-MIN achieve the best rank correlation with AA on most datasets. Moreover, the ranking performance of the L1-AVG, L1-MAX, L2-AVG, and L2-MAX are also comparable with the reputation based ranking algorithms (Mizz, YZLM, and dKVD). The results indicate that our proposed algorithms are effective for ranking objects on rating networks. It is worth noting that HITS exhibits quite poor performance (even exhibits negative correlation to AA on Amazon datasets), as it does not consider the users' reputation in ranking. Consequently, in the following experiments, we do not report the results of HITS because it is not effective for ranking on rating networks.

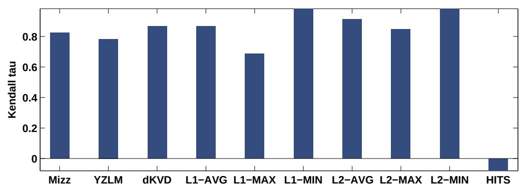

Figure 2: Comparison of effectiveness on Amazon dataset.

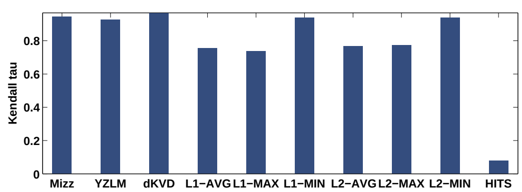

Figure 3: Comparison of effectiveness on Bookcrossing dataset.

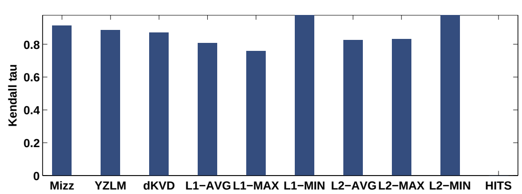

Figure 4: Comparison of effectiveness on Epinions dataset.

We also compare the Kendall Tau between two ranks by our algorithms and other baselines. As shown in Table 7, Table 8, and Table 9, the proposed algorithms are strong correlated to the baselines on all three real datasets. This results confirm that our proposed algorithms are effective as other reputation-based algorithms. In particular, L1-MIN and L2-MIN achieve the best correlation with the baselines, followed by L2-AVG, L1-AVG, L2-MAX, and L1-MAX. In general, on all three datasets, our algorithms exhibit the best correlation with AA, and then with Mizz, YZLM, and dKVD. It is interesting to note that L1-MIN and L2-MIN achieve the same performance on all three datasets.

Table 7: Kendall Tau on Amazon dataset.

Algorithms	AA	Mizz	YZLM	dKVD
L1-AVG	0.866	0.819	0.783	0.746
L1-MAX	0.686	0.662	0.639	0.627
L1-MIN	**0.982**	**0.867**	**0.822**	**0.782**
L2-AVG	0.914	0.864	0.822	0.782
L2-MAX	0.847	0.816	0.783	0.756
L2-MIN	**0.982**	**0.867**	**0.822**	**0.782**

Table 8: Kendall Tau on Bookcrossing datasets

Algorithms	AA	Mizz	YZLM	dKVD
L1-AVG	0.756	0.757	0.747	0.742
L1-MAX	0.738	0.735	0.727	0.725
L1-MIN	**0.938**	**0.939**	**0.921**	**0.902**
L2-AVG	0.768	0.764	0.751	0.745
L2-MAX	0.773	0.770	0.758	0.753
L2-MIN	**0.938**	**0.939**	**0.921**	**0.902**

Table 9: Kendall Tau on Epinions datasets

Algorithms	AA	Mizz	YZLM	dKVD
L1-AVG	0.807	0.811	0.808	0.795
L1-MAX	0.759	0.758	0.754	0.749
L1-MIN	**0.978**	**0.869**	**0.909**	**0.882**
L2-AVG	0.824	0.826	0.823	0.808
L2-MAX	0.832	0.829	0.826	0.817
L2-MIN	**0.978**	**0.869**	**0.909**	**0.882**

Robustness: To evaluate the robustness of an algorithm, we add to the original datasets three types of spamming users, who give ratings to a random set of objects. We test the ranking algorithms on both original datasets and the noisy datasets with 10% to 50% spamming users, where each type of spamming users has the same proportion. Fig. 5, Fig. 6, and Fig. 7 show the robustness of the ranking algorithms by Kendall Tau vs. spamming ratio.

As depicted in Fig. 5, Fig. 6, and Fig. 7, we can clearly see that L1-MAX and L2-MAX achieve the best robustness on Amazon and Epinions datasets, while in Bookcrossing dataset L1-AVG and L2-AVG achieve the best performance. In Aamzon and Bookcrossing datasets, we find that L1/L2-MAX, and L1/L2-AVG are significantly more robust than the other algorithms. However, in Epinions datasets, L1-MAX and L2-MAX perform better than the other algorithms. The robustness of L1-MIN and L2-MIN are not good, but they are still comparable with the other algorithms. It is important to note that the state-of-the-art reputation-based algorithm (YZLM) is not very robust in Amazon and Bookcrossing datasets, and its robustness is even worse than AA. Similarly, the robustness of dKVD is not desirable, which is worse than AA on Amazon and Epionions datasets. In general, for all algorithms, the robustness decreases as the spamming ratio increases.

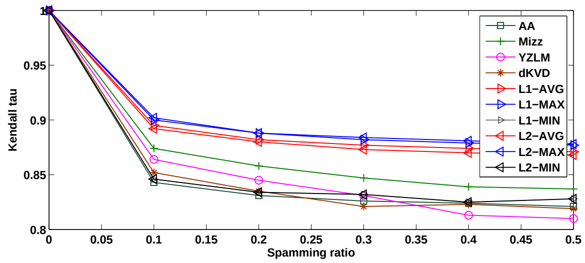

Figure 5: Robustness testing on Amazon dataset.

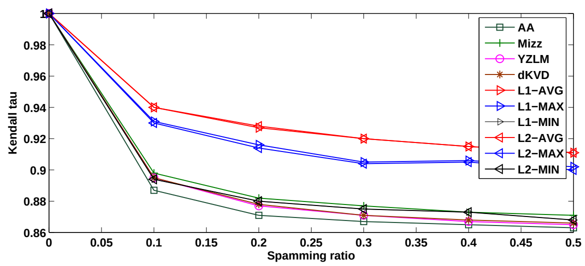

Figure 6: Robustness testing on Bookcrossing dataset.

Efficiency: We compare the running time of our algorithms and the baselines on Amazon datset, and similar results can be observed on other two datasets. For all algorithms, we set the termination condition by either the iterations exceed 30 or the average ranking difference between two consecutive iterations is less than 1e-8. As shown in Fig. 8, we can observe that our algorithms are more efficient than the other iterative algorithms. The running time of our algorithms is around 2-5 seconds. Mizz and YZLM are not efficient, as these two algorithms do not converge. dKVD is also not efficient, because the rate of convergence is q-linear. This results confirm our time complexity analysis.

Analysis of convergence: We study the convergence of our algorithms. In particular, we study the maximal ranking difference between two consecutive iterations. Fig. 9 shows how this difference decreases as the iterations increases on Amazon dataset, and similar results can be observed in other two datasets. As shown in Fig. 9, we can observe that all of our algorithms converge in only 8 iterations. This results confirm the fact that the algorithms converge in exponential rate as analyzed in Section 4.

Effect of parameter λ: We study the effect of parameter λ in our algorithms on Amazon dataset and the corresponding noisy dataset with 20% spamming ratio. Similar results can be observed on the other datasets. Specifically, we study the effect of λ w.r.t. the effectiveness and robustness of the algorithms. Fig. 10(a) depicts the effectiveness of our algorithms under various λ, where the effectiveness is measured by the rank correlation between our algorithms and AA. From Fig. 10(a), we can observe that L1/L2-MIN, and L2-AVG are quite robust w.r.t. λ, and the effectiveness of L1-AVG, L1-MAX, and L2-MAX decrease as λ increases. Fig. 10(b)

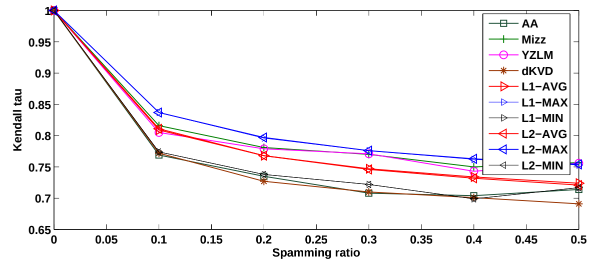

Figure 7: Robustness testing on Epinions dataset.

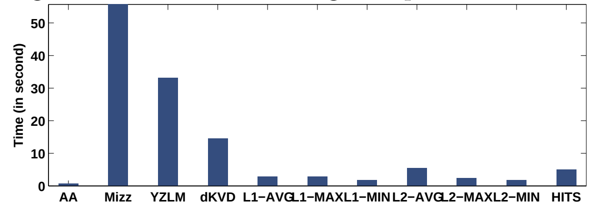

Figure 8: Comparison of efficiency on Amazon dataset.

shows the robustness of our algorithms under different λ. As depicted in Fig. 10(b), we can find that the robustness of L1-MIN, L2-MIN, L2-AVG, and L2-MAX are not very sensitive w.r.t. λ. The robustness of L1-AVG decreases as λ increases. The robustness of L1-MAX decreases as λ increases when $\lambda \leq 0.65$, and otherwise the robustness of L1-MAX algorithm increases as λ increases. In general, the difference between the maximal and minimal robustness of all our algorithms do not exceed 0.08 under various λ, which indicates that the robustness of our algorithms is not very sensitive w.r.t. the parameter λ. In practice, we suggest to set $\lambda = 0.1$ as all our algorithms achieve both effective and robust rank when $\lambda = 0.1$.

6 Conclusion

In this paper, over a bipartite rating network, we propose six new reputation-based ranking algorithms, where user's reputation is measured by the aggregated difference between the user's rating and the corresponding object's ranking. We prove the convergence properties of the proposed algorithms. We evaluate the proposed algorithms in three real datasets, and the results confirm that our algorithms are effective, efficient, and robust. Future work includes generalizing the proposed algorithms to time-evolving bipartite networks and exploring our algorithms for other data mining applications, such as finding reliable users and content in Community Question Answering systems [3], and recognizing high quality papers in paper review systems [25].

Acknowledgments

The work was supported by grants of the Research Grants Council of the Hong Kong SAR, China No. 411211, 419008 and 419109.

References

[1] B. T. Adler and L. de Alfaro. A content-driven reputation system for the wikipedia. In *WWW*, 2007.

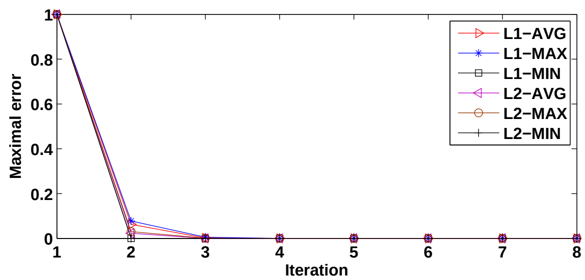

Figure 9: Convergence of our algorithms.

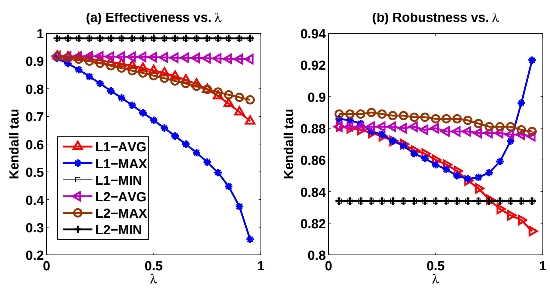

Figure 10: The effect of λ.

[2] R. Andersen, C. Borgs, J. T. Chayes, U. Feige, A. D. Flaxman, A. Kalai, V. S. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *WWW*, 2008.

[3] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *WWW*, 2009.

[4] P. Bonacich. Power and centrality: A family of measures. *The American Journal of Sociology*, 92(5):1170–1182, 1987.

[5] S. Brin and L. Page. Pagerank: Bringing order to the web. Technical report, Stanford Digital Library Project, 1997.

[6] B.-C. Chen, J. Guo, B. Tseng, and J. Yang. User reputation in a comment rating environment. In *KDD*, 2011.

[7] L. de Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler. Reputation systems for open collaboration. *Commun. ACM*, 54(8):81–87, 2011.

[8] C. de Kerchove and P. V. Dooren. Iterative filtering in reputation systems. *SIAM J. Matrix Analysis Applications*, 31(4):1812–1834, 2010.

[9] H. Deng, M. R. Lyu, and I. King. A generalized co-hits algorithm and its application to bipartite graphs. In *KDD*, 2009.

[10] F. Fouss, Y. Achbany, and M. Saerens. A probabilistic reputation model based on transaction ratings. *Inf. Sci.*, 180(11):2095–2123, 2010.

[11] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

[12] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW '02*.

[13] A. Jøsang and J. Golbeck. Challenges for robust trust and reputation systems. In *STM*, 2009.

[14] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[15] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. In *CIKM*, 2007.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW*, 2003.

[17] M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–89, 1938.

[18] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[19] P. Laureti, L. Moret, Y.-C. Zhang, and Y.-K. Yu. Information filtering via iterative refinement. *EPL*, 75(6):1006–1012), 2006.

[20] J. Leskovec. Standford network analysis project. 2010.

[21] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Signed networks in social media. In *CHI*, 2010.

[22] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys*, pages 17–24, 2007.

[23] M. Medo and J. R. Wakeling. The effect of discrete vs. continuous-valued ratings on reputation and ranking systems. *EPL*, 91(48004), 2010.

[24] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *WWW*, 2011.

[25] S. Mizzaro. Quality control in scholarly publishing: A new proposal. *JASIST*, 54(11):989–1005, 2003.

[26] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *HICSS*, 2002.

[27] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *AAMAS*, 2002.

[28] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[29] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *International Semantic Web Conference*, pages 351–368, 2003.

[30] G. Theodorakopoulos and J. S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, 2006.

[31] V. G. V. Vydiswaran, C. Zhai, and D. Roth. Content-driven trust propagation framework. In *KDD*, 2011.

[32] Y.-K. Yu, Y.-C. Zhang, P. Laureti, and L. Moret. Decoding information from noisy, redundant, and intentionally distorted sources. *Physica A*, 371:732–744), 2006.

[33] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, 29(4):371–388, 2000.

[34] Y.-B. Zhou, T. Lei, and T. Zhou. A robust ranking algorithm to spamming. *EPL*, 94(48002), 2011.

[35] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05*.