Ranking Reputation and Quality in Online Rating Systems

Hao Liao[1], An Zeng[1]*, Rui Xiao[1], Zhuo-Ming Ren[1,2], Duan-Bing Chen[1,3], Yi-Cheng Zhang[1]

1 Department of Physics, University of Fribourg, Fribourg, Switzerland, **2** Research Center of Complex Systems Science, University of Shanghai for Science and Technology, Shanghai, China, **3** Web Sciences Center, University of Electronic Science and Technology of China, Chengdu, China

Abstract

How to design an accurate and robust ranking algorithm is a fundamental problem with wide applications in many real systems. It is especially significant in online rating systems due to the existence of some spammers. In the literature, many well-performed iterative ranking methods have been proposed. These methods can effectively recognize the unreliable users and reduce their weight in judging the quality of objects, and finally lead to a more accurate evaluation of the online products. In this paper, we design an iterative ranking method with high performance in both accuracy and robustness. More specifically, a reputation redistribution process is introduced to enhance the influence of highly reputed users and two penalty factors enable the algorithm resistance to malicious behaviors. Validation of our method is performed in both artificial and real user-object bipartite networks.

Citation: Liao H, Zeng A, Xiao R, Ren Z-M, Chen D-B, et al. (2014) Ranking Reputation and Quality in Online Rating Systems. PLoS ONE 9(5): e97146. doi:10.1371/journal.pone.0097146

Editor: Renaud Lambiotte, University of Namur, Belgium

Received December 18, 2013; **Accepted** April 15, 2014; **Published** May 12, 2014

Funding: This work was partially supported by the EU FP7 Grant 611272 (project GROWTHCOM) and by the Swiss National Science Foundation (grant no. 200020-143272). D.-B. Chen acknowledges support from the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and the Huawei University-Enterprise Cooperation Project YBCB2011057. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: an.zeng@unifr.ch

Introduction

With the rapid development of World Wide Web, our lives nowadays rely more and more on the Internet [1–4]. Online systems allow a large number of users to interact with each other and provide thousands of movies, millions of books, billions of web pages for them to choose [5]. Though a lot of useful online objects are out there, accurately ranking their quality is not easy. Therefore, many online websites (such as Ebay, Amazon, Netflix) introduce the so-called rating system [6,7] in which users can evaluate objects by giving discrete ratings. To approximately judge the quality of a certain object, a user can refer to the historical ratings the object received.

The most straightforward method to rank objects is to consider their average ratings (we refer it as the *mean* method). However, such methods are very sensitive to the noisy information and manipulation. In these rating systems, some users may give unreasonable ratings because they are not serious about the rating or simply not familiar with the related field [8]. In addition, the system may contain some malicious spammers who always deliberately give high ratings to some low quality objects [9,10]. To solve this problem, some ranking algorithms robust to spamming are proposed. Normally, these algorithms build a reputation system [11–14] for users. The ratings of users with higher reputation are assigned with more weight. By iteratively updating users' reputation [15,16], the quality of objects can be ranked more accurately than the average ratings method. In fact, similar iterative ranking algorithms have been used in many other fields, such as country-product [17] or author-paper [18] systems.

Under this framework, some methods have already been proposed. A representative one is called iterative refinement (*IR*) method [19]. In IR, a user's reputation is inversely proportional to the difference between his or her rating vector and objects' estimated quality vector (i.e., weighted average rating). The estimated quality of objects and reputation of users are iteratively updated until they become stable. In [20], the iterative refinement algorithm is modified by assigning trust to each individual rating. More recently, another improved iterative method is designed (we refer it as the *CR* method) [21]. A user's reputation is calculated by the Pearson correlation [22,24] between his ratings and objects' estimated quality. This method is claimed to be very robust to different spamming behaviors [25–27].

In this paper, we introduce a reputation redistribution process to the iterative ranking algorithm in [21], which can effectively enhance the weight of the highly reputed users and lower the weight of the users with low reputation in estimating the quality of objects. We test our method in both artificial and real data. The results show that the accuracy of objects' quality ranking is considerably improved. Moreover, we introduce two penalty factors to the iterative ranking algorithm which significantly improve its robustness against the malicious spamming behaviors. Interestingly, the improvement from the penalty factors is surprisingly large in real data, which indicates that there are many intentional pushing rating from spammers in real systems.

Methods

Iterative ranking algorithm with reputation redistribution

We first briefly describe the iterative algorithm with reputation redistribution (short for IARR). It is built directly on the CR method but with the reputation redistribution process for eliminating noisy information in the iterations, so as to improve the accuracy in objects' quality ranking. The rating system can be naturally described by a weighted bipartite network [28]. The users are denoted by set U and objects (e.g. books, movies or others) are denoted by set O. To better distinguish different type of nodes in the bipartite network, we use Latin letters for users and Greek letters for objects. The rating given by a user i to object α is the weight of the link, denoted by $r_{i\alpha}$. The degree of users and objects are respectively k_i and k_α. Moreover, we define the set of objects selected by user i as O_i and the set of users selecting object α as U_α.

We use Q_α and R_i to note the quality of object α and the reputation of user i, respectively. The initial configuration for each user is set as $R_i = k_i/M$ (where M is the number of objects). The quality of an object depends on users' rating and can be calculated by the weighted average of rating to this object. Mathematically, it reads

$$Q_\alpha = \frac{\sum_{i\in U_\alpha} R_i r_{i\alpha}}{\sum_{i\in U_\alpha} R_i} \tag{1}$$

In the iteration, both Q_α and R_i will be updated. To calculate the reputation R_i of user i in certain step, we first calculate the Pearson correlation coefficient between the rating vector of user i and the corresponding objects α quality vector as the temporal reputation (TR_i):

$$TR_i = \frac{1}{k_i}\sum_{\alpha\in O_i}\left(\frac{r_{i\alpha}-\bar{r}_i}{\sigma_{r_i}}\right)\left(\frac{Q_\alpha-\bar{Q}_i}{\sigma_{Q_i}}\right) \tag{2}$$

where σ_{r_i} and σ_{Q_i} are, respectively, the standard deviations of the rating vector of user i and the corresponding objects' quality vector, and $\bar{r}_i$ and $\bar{Q}_i$ are their mean values. If TR_i lower than 0, the reputation of user i will be assigned to 0. Therefore, TR_i is bounded in [0,1]. As discussed in the introduction, the IR method considers a user's reputation as inversely proportional to the mean squared error between his/her rating vector and the corresponding objects' weighted average rating vector [19]. The reputation based Pearson correlation is shown to be more robust to spamming ratings than the IR method and thus lead to a more accurate estimation of object quality [21].

TR_i is then nonlinearly redistributed to all users via

$$R_i = TR_i^\theta \frac{\sum_j TR_j}{\sum_j TR_j^\theta}, \tag{3}$$

where θ is a tunable parameter. The method will reduce to the mean and CR methods when $\theta=0$ and $\theta=1$, respectively [21]. The obtained R_i will be then used as the reputation of user i to calculate the quality of objects in eq. 1. With this reputation redistribution process, the user with high TR_i will be amplified, and vice versa. By reducing the weight of the users with low TR_i, we can eliminate the noisy information in the iterative processes. This effect is accumulated in each iterative step, and will finally lead to a big improvement in the accuracy of object quality estimation. Actually, the basic idea of the reputation redistribution process is similar to the well-known *k-nearest neighbors* (KNN) algorithms which eliminate the noise by entirely drop the information of nodes outside the k-nearest neighbors [23]. The KNN algorithm is widely used in recommender systems. Here, we design a smooth way to implement the idea to object quality ranking. Though the modification of the method seems to be small, the improvement is substantial (see the following analysis).

Users' reputation and objects' quality will be updated in each step. The iteration stops when the change of the quality

$$|Q-Q'| = 1/M\sum_{l\in O}(Q_l - Q_l{}')^2 \tag{4}$$

is lower than a small value Δ (in this paper, $\Delta = 10^{-4}$).

Improving the reliability of the method

We now try to enhanced the reliability of the method. In principle, when a user only assessed a small number of objects, he cannot have very high reputation. This is natural since it is easy for a user to guess correctly the quality of one object by chance, but very difficult for a large number. Therefore, when a user rates many objects and his reputation is still high, this user is more reliable. Similar idea is applied to the object side. If an object is rated by one or two users, though the ratings are high, it is too arbitrary to claim this object has high quality. Based on above two reasons, we introduce a penalty factor to eq. 1 and eq. 2, respectively. The modified eq. 1 reads

$$Q_\alpha = \max_{i\in U_\alpha}\{R_i\}\frac{\sum_{i\in U_\alpha} R_i r_{i\alpha}}{\sum_{i\in U_\alpha} R_i}, \tag{5}$$

and the eq. 2 is modified as

$$TR_i = \frac{lg(k_i)}{\max\{lg(k_j)\}}\cdot\frac{1}{k_i}\sum_{\alpha\in O_i}\left(\frac{r_{i\alpha}-\bar{r}_i}{\sigma_{r_i}}\right)\left(\frac{Q_\alpha-\bar{Q}_i}{\sigma_{Q_i}}\right). \tag{6}$$

With these two penalty factors, the objects rated by only low reputation users can only be low and the users who only rate a small number of objects cannot have high reputation. The penalty will be amplified in the iteration and finally filter out the influence of the not yet reliable users. This enhanced iterative algorithm is referred as IARR2 in the following text.

Results on Artificial Networks

Generating artificial networks

We start our analysis by applying IARR and IARR2 to artificial networks. To create the artificial network, we set $|U| = 6000, |O| = 4000$. We assume that each object α has an intrinsic quality denoted by Q'_α . When a user i gives a rating to the object α, he/she will inevitably have some magnitude of rating error $\delta_{i\alpha}$. Accordingly, the rating to α from user i will be

$$r_{i\alpha} = Q'_\alpha + \delta_{i\alpha}. \tag{7}$$

Without losing any generality, both users' ratings and objects' qualities are assumed to be [0,1]. In our simulation, objects' qualities Q' will be drawn from an uniform distribution (0,1). $\delta_{i\alpha}$ is draw from a normal distribution $(0,\delta_i)$ where δ_i denotes the users magnitude of rating error. For each user i, δ_i is generated from an uniform distribution ($\delta_{min}=0.1, \delta_{max}=0.5$).

To generate the user-object bipartite network, the rating (weighted links) will be added to the network one by one until the network reaches a certain sparsity ($\phi=0.2$). Under this setting, the final network will have $\phi|U||O|=4.8*10^5$ links. In most online systems, both users' and objects' degree follow quite broad degree distribution [29]. Accordingly, the preferential attachment mechanism is employed here to add links. At each step t, a random user i and a random object α will be picked and a link will be added between them with the weight from eq. 7. The probabilities for selecting a user i and object α are respectively

$$\chi_i(t)=\frac{k_i(t)+1}{\sum_{j\in U}(k_j(t)+1)}, \tag{8}$$

and

$$\chi_\alpha(t)=\frac{k_\alpha(t)+1}{\sum_{\beta\in O}(k_\beta(t)+1)}, \tag{9}$$

where $k_i(t)$ and $k_\alpha(t)$ are the degree of user i and object α at time step t [30].

Estimation of user reputation

For a good reputation estimation algorithm, the obtained user reputation R_i should be negatively correlated with δ_i. The stronger the correlation is, the better the algorithm is. Here, we compare the performance of IARR and IARR2 methods with the mean, IR [19] and CR [21] methods. The mean method is the most straight reputation estimation method in which user's reputation is calculated as one over the mean squared error between his/her rating vector and the corresponding objects' weighted average rating vector (without any iteration).

The results of each method are reported in Fig. 1. We define I equally distributed intervals between δ_{min} and δ_{max} and group the nodes whose δ fall in the same interval. Each group is denoted by its median in δ as δ_c. Since $\delta_{max}=0.5$ and $\delta_{min}=0.1$, we set $I=40$ so that the interval is 0.01. The averaged reputation $\langle R_c\rangle$ of the users in the same group is calculated. The relation between $\langle R_c\rangle$ and δ_c is reported in Fig. 1(a). Here, the parameter is set as $\theta=3$ in IARR and $\theta=5$ in IARR2. As one can see, $\langle R_c\rangle$ and δ_c in most methods are negatively correlated except the mean method. In order to quantify the correlation, we calculate the Pearson correlation ρ between R_i and δ_i. Specifically, $\rho=0.002$ in the mean method, $\rho=-0.445$ in the IR method, $\rho=-0.640$ in the CR method, $\rho=-0.791$ in IARR method and $\rho=-0.800$ in IARR2 method. The dependence of the Pearson correlation ρ on θ in IARR and IARR2 methods is studied in Fig. 1(b). Interestingly, there is an optimal θ^* in both methods ($\theta^*=3$ in IARR and $\theta^*=5$ in IARR2). In the following analysis, we will set $\theta=3$ in IARR and $\theta=5$ in IARR2.

Robustness against random and malicious ratings

A good ranking algorithm should be not only accurate in estimating users' reputation and objects' quality, but also robust against distort information, i.e. the accuracy of the algorithm shouldn't be strongly affected when the system contains some random or malicious ratings. The random ratings mainly come from the naughty users who just play around with the information and give ratings which mean nothing. The malicious ratings are from some spammers who always gives maximum/minimum allowable ratings that also try to push up some target objects. Both type of distort ratings widely exist in real systems [31,32]. Therefore, we investigate the effect of the noisy and willful distort ratings on the performance of the IARR and IARR2 methods.

We start with the system with random ratings. We first generate the artificial networks according to the rules described above. In order to add some noisy information to the systems, we randomly pick p fraction of the links and replace the rating on each of these links by a random value in range of [0,1]. Clearly, the noisy information in the system gradually increases with the parameter p. When $p=1$, there is no any true information in the rating system. In the following analysis, we set $p\in[0,0.95]$.

In order to compare the performance of different ranking algorithms, we here adopt two metrics: Kendall's tau [33] and AUC (the area under the receiver operating characteristic curve) [34]. The Kendall's tau here measures the rank correlation between the estimated quality of objects Q and the "true"quality of them Q'. Mathematically, it reads

$$\tau=\frac{\sum_{\alpha\in O}\sum_{\beta\in O}sgn[(Q_\alpha-Q_\beta)(Q'_\alpha-Q'_\beta)]}{|O|(|O|-1)} \tag{10}$$

where $sgn(x)$ is the sign function, which returns 1 if $x>0$; -1 if $x<0$; and 0 for $x=0$. Here $(Q_\alpha-Q_\beta)(Q'_\alpha-Q'_\beta)>0$ means concordant and negative means discordant. According to the definition, $\tau\in[-1,1]$. A higher τ indicates a more accurate estimation of objects' true quality.

In real cases, the true quality of objects is unknown, which makes it impossible to evaluate the algorithm by τ. Therefore, we consider another accuracy measure called AUC. To calculate AUC, one should select a group of benchmark objects which are considered to be generally with high quality. We selected 5% objects with highest Q' as the benchmark objects. The AUC requires n times of independent comparison of the benchmark objects and non-benchmark objects. After the comparison, we record $n1$ as the number of times in which the benchmark object has higher Q than non-benchmark object, and $n2$ as the number of times in which the benchmark object and the non-benchmark object are having the same Q. The final AUC is calculated as $AUC=(n1+0.5*n2)/n$. If all the objects are ranked randomly by some algorithm, $AUC=0.5$. When $AUC=1$, all the benchmark objects are ranked higher than the non-benchmark objects.

Here, we compare the Kendall's tau and AUC in five algorithms: Mean, IR, CR, IARR and IARR2. In Fig. 2(a) and (b), we respectively report the dependence of τ and AUC on p in different algorithms. As one can see, IARR and IARR2 methods outperform the other three methods, especially when p is large. However, the difference between IARR and IARR2 algorithms is almost indistinguishable. This is due to the reason that the random rating attack cannot fully model the spamming behavior in real systems.

We further consider the malicious rating attack in the artificial networks. In practice, we randomly pick p fraction of the links in the generated artificial network and set half of them to be the maximum rating (i.e. 1) and the other half of them to be the minimum rating (i.e. 0). This scenario models the so-called push rating in which spammers try to promote the target low quality objects. The results of τ and AUC of different ranking algorithms in this case are shown in Fig. 2(c) and (d). One can observe that IARR and IARR2 still have advantage over other methods.

The parameters are respectively set as $\theta=3$ and $\theta=5$ in IARR and IARR2 in the robustness analysis above. In Fig. 3, we analyze the effect of θ on the resultant AUC and τ in these two methods. We set $p=0.9$ in both random rating and malicious rating attacks. The results show that the parameter θ can indeed improve the

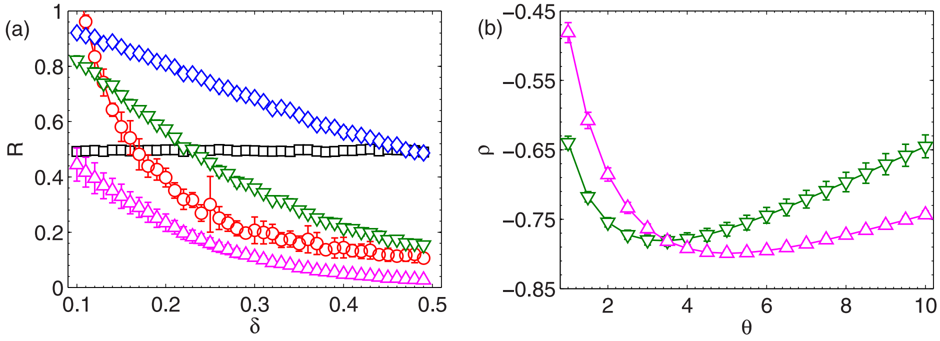

Figure 1. (a) The relation between $\langle R_c \rangle$ and δ_c in different methods. The parameters are set as $\theta = 3$ in IARR and $\theta = 5$ in IARR2. (b) the dependence of the Pearson correlation ρ on θ in IARR and IARR2 methods. The results in this figures are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.
doi:10.1371/journal.pone.0097146.g001

performance of the ranking algorithms (Note that when $\theta = 1$, IARR degenerates to the CR algorithm). Moreover, we can observe that the optimal θ^* in IARR and IARR2 are more or less the same. Specifically, $\theta^* = 4$ in the random rating attack case, and $\theta^* = 3$ in the malicious rating attack case. Finally, it shows that IARR2 enjoys a higher AUC and τ than IARR in the malicious attack case, which implies that IARR2 may have high performance in real systems (since the malicious ratings are more common in real case).

Results on Real Networks

In this section, we will study the IARR and IARR2 methods in real systems. Here, we select two commonly used real data sets containing ratings on movies: Netflix and MovieLens. MovieLens is provided by GroupLens project at University of Minnesota (www.grouplens.org). We use a subset of the complete data. In our subset, there are 1 million ratings given on the integer rating scale from 1 to 5. Each user in subset has at least 20 ratings. Netflix is a huge data set released by the DVD rental company Netflix for its Netflix Prize (www.netflixprize.com). We again extracted a smaller data set by choosing 5000 users who have rated at least 20 movies (the same as MovieLens) and took all movies they had rated. The Netflix ratings are also given on the integer rating scale from 1 to 5. Some basic characteristics of these data sets are summarized in table 1.

We run different ranking algorithms in these two data sets and study the distribution of the obtained Q. As shown in Fig. 4, Q of both CR and IARR algorithms roughly follow a normal distribution. One can also see that there is an abrupt peak in each integer rating, especially in the Netflix data. This is because

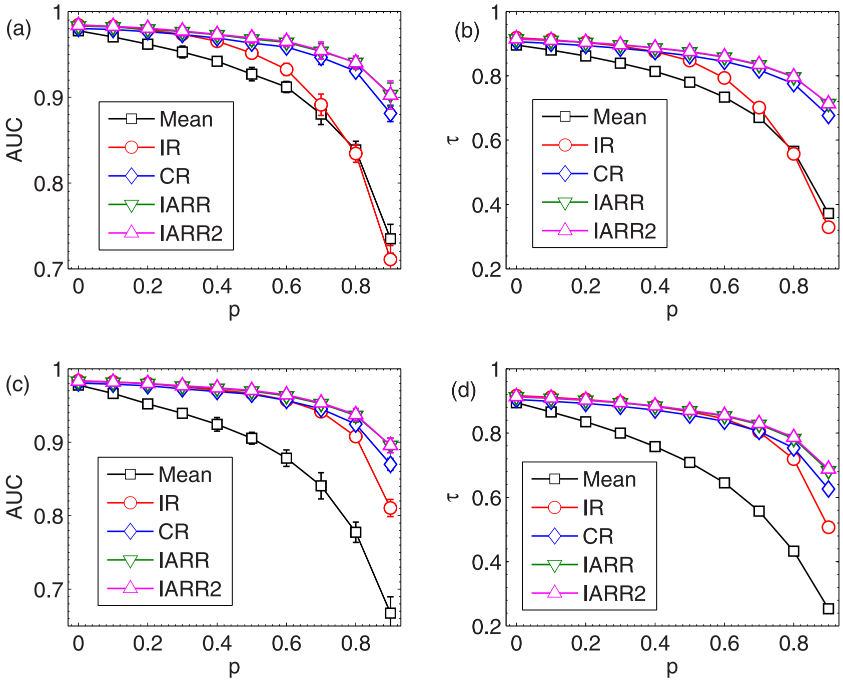

Figure 2. (a) and (b) the AUC and τ of different algorithms to random rating spamming. (c) and (d) different algorithms to malicious push rating spamming. The results in this figure are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.
doi:10.1371/journal.pone.0097146.g002

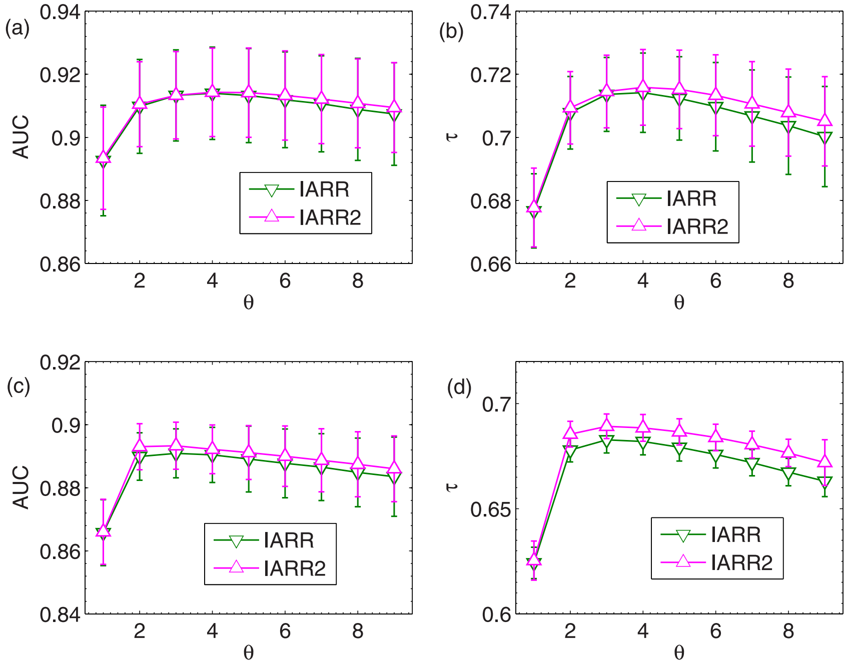

Figure 3. (a) and (b) the dependence of AUC and τ on θ in IARR and IARR2 methods in the random rating attack case. (c) and (d) the dependence of AUC and τ on θ in IARR and IARR2 methods in the malicious rating attack case. The results in this figure are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.
doi:10.1371/journal.pone.0097146.g003

some objects are only rated by one user, or all users give the object with the same rating. We further study the occurring frequency of this case in real online systems. We first study the degree distribution of objects in the real systems. Fig. 5(a) and (b) show the frequency distribution of object degree in Movielens and Netflix, respectively. One can clearly see that both distributions follow power-law form. Another message one can get from these two figures is that there are many objects are only rated by one user, around 100 objects in Movielens and 1000 in Netflix. Once these objects are rated with 5, they will be considered as the highest quality objects by the mean and CR method. Furthermore, we check the frequency of these low degree objects with high ratings. Here, we select the object with the same degree k and calculate the frequency C that all raters give them high ratings (in our case, we consider rating 4 and 5 as high ratings). In Fig. 5 (c) and (d), we show the relation between frequency C and k in movielens and Netflix, respectively. As one can see, the value of C is rather big, especially when k is small. These objects, though with low degree, will be considered as highest quality objects by the mean and CR method.

The above analysis implies that the ranking provided by CR and IARR algorithms are not very reliable since many small degree objects will appear in the top of quality ranking list. This problem is well solved in the IARR2 method. With the penalty factors, IARR2 will give low score to those suspicious objects (i.e. objects with high rating but small degree). In Fig. 4, we can see that the abrupt peak disappear in the Q distribution from the IARR2 algorithm. The penalty factors will decrease the maximum value of Q. For better illustration, the distribution of Q in the IARR2 is rescaled to [1,5] in Fig. 4. We remark that the object ranking from the IARR2 algorithm can well reflect objects' true quality. We will use some awarded movies to support this statement in the following.

Since we don't know the true quality of the movies in these two data sets, we adopt the AUC metric to study the IARR and IARR2 here. To calculate the AUC, we select those movies which nominated at Annual Academy Award (source:www.filmsite.org) as benchmark good movies. In movieLens and Netflix data contains 203 benchmark movies and 293 benchmark movies.

Table 2 shows the AUC resulted from four different algorithms applying to the real data sets. One can immediately see that the AUC is generally lower in the Netflix data, which indicates that there are more spammers (or more harmful spammers) in Netflix data. Moreover, it shows that the CR method doesn't actually have significant advantage towards the Mean and IR methods, though it largely outperforms the Mean and IR methods in the artificial networks. This result indicates that the CR method is very sensitive to the "real"spammers. The IARR can slightly improve the performance of CR method by introducing the reputation redistribution process (the parameter is set as $\theta = 3$ here).

Table 1. Some basic characteristics of the real data sets considered in this paper.

Methods	$\lvert U\rvert$	$\lvert O\rvert$	$\langle k_u\rangle$	$\langle k_o\rangle$	**Sparsity**
MovieLens	6040	3706	166	270	0.0447
Netflix	5000	16195	214	66	0.0132

doi:10.1371/journal.pone.0097146.t001

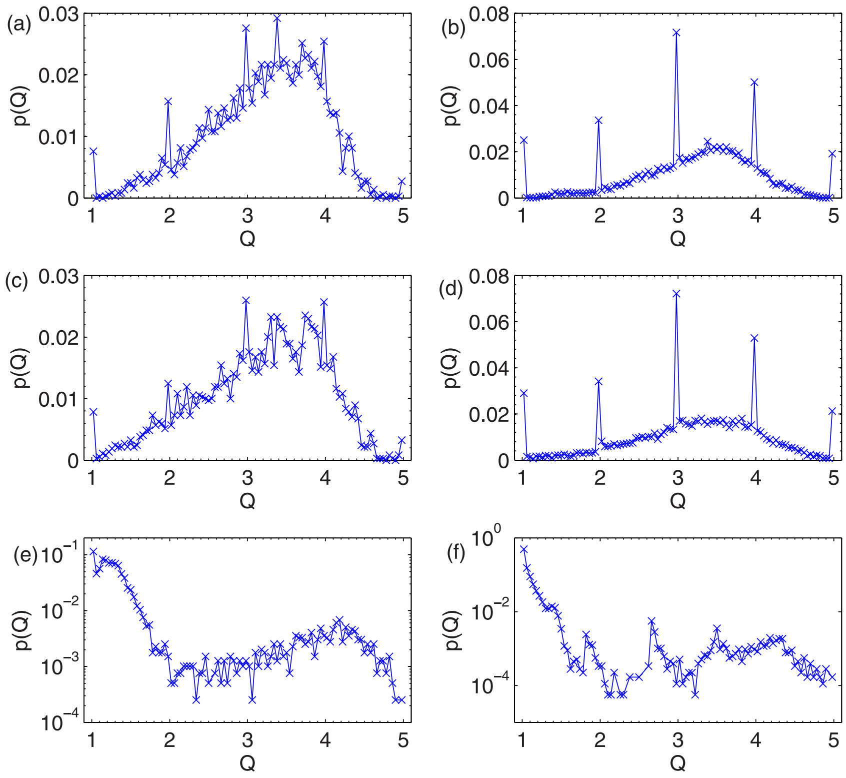

Figure 4. (a), (c) and (e) are the distribution of Q of the CR, IARR and IARR2 algorithms in Movielens data, respectively. (b), (d), (f) are the distribution of Q of the CR, IARR and IARR2 algorithms in Netflix data, respectively. θ is set as 3 in both IARR and IARR2 algorithms.
doi:10.1371/journal.pone.0097146.g004

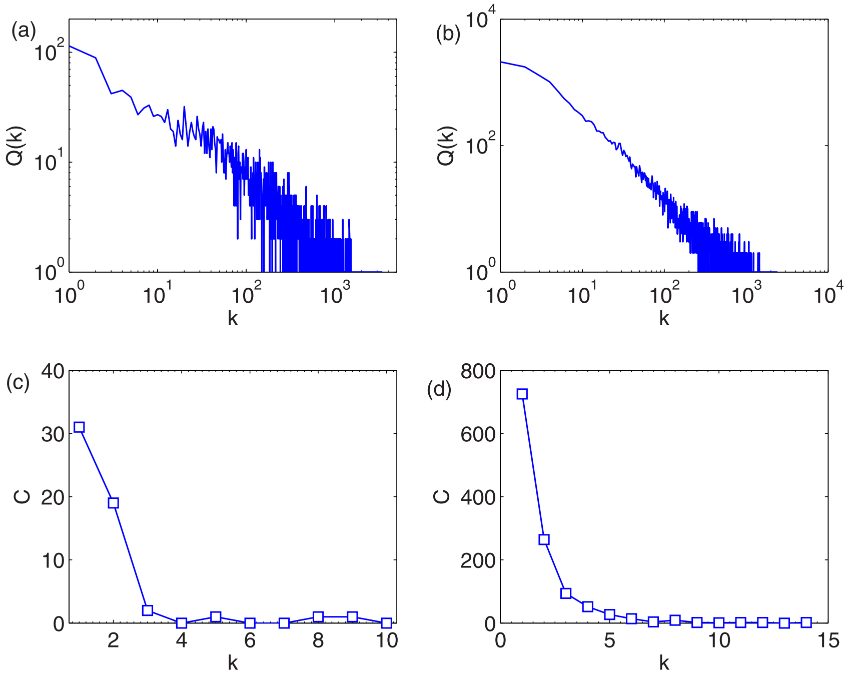

Figure 5. (a) and (b) are the frequency distribution of object degree in Movielens and Netflix, respectively. (c) and (d) are the relation between frequency C and k in movielens and Netflix, respectively.
doi:10.1371/journal.pone.0097146.g005

Table 2. AUC values of different algorithms for the real-data sets.

Methods	Mean	IR	CR	IARR	IARR2
MovieLens	0.873	0.876	0.872	0.876	**0.902**
Netflix	0.729	0.746	0.746	0.758	**0.886**

doi:10.1371/journal.pone.0097146.t002

Interestingly, the IARR2 method remarkably outperform all the other methods. This implies that the IARR2 method indeed captures the harmful features of the real spammers. More specifically, the IARR2 method is very robust against the cases where low quality objects are highly rated by several unreliable users. Moreover, it also punishes some spamming users who want to increase their reputation by giving several movies the mean ratings. The results in table 2 indicates that these spamming behaviors happen frequently in real online rating systems.

Conclusions

In this paper, we propose a robust iterative ranking algorithm with reputation redistribution process. The reputation redistribution process can effectively enhance the weight of the highly reputed users and reduce the weight of the users with low reputation in estimating the quality of objects. Two penalty terms to the iterative ranking algorithm which significantly improve its robustness against some malicious spamming behavior. We test our method in both artificial and real data. The results show that the accuracy of ranking the quality of objects is considerably improved. Interestingly, the improvement from the penalty terms is surprisingly large in real data, which implys that there are many intentional pushing rating from spammers in real cases.

Finally, we remark that our work is of great significance from practical point of view. Nowadays, the internet plays a significantly important role in our daily lives. Online users usually select products by referring to peers' ratings. Without a reputation system, there is a risk that users' choices might be misled by some spamming ratings. Our method in this paper is not only effective in estimating the true quality of the objects but also very robust to spamming ratings. Therefore, we believe that our method can be very useful when applied to real online websites.

Author Contributions

Conceived and designed the experiments: HL AZ YCZ. Performed the experiments: HL AZ. Analyzed the data: HL AZ RX ZMR DBC. Wrote the paper: HL AZ DBC YCZ.

References

1. Watts DJ, Strogatz SH(19998) Collective dynamics of ‘small-world’ networks. Nature 393: 440–442.
2. Albert R, Barabasi AL (2002) Statistical mechanics of complex networks. Rev. Mod.Phys 74: 47–97.
3. Guttman RH, Moukas AG, Maes P (2001) Agent-mediated electronic commerce: A survey. The Knowledge Engineering Review 13: 147–159.
4. Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. J. ACM 46: 604–632.
5. Lü L, Medo M, Yeung CH, Zhang YC, Zhang ZK, et al. (2012) Recommender systems. Physics Report 519: 1–49.
6. Dellarocas C (2000) Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In:Proceedings of IEEE 2nd ACM conference on Electronic commerce. ACM Press, pp. 150–157
7. Goldberg D, Nichols D, Oki BM, Terry D (1992) Using collaborative filtering to weave an information tapestry. Communications of the ACM 35: 61–70.
8. Pan WK, Xiang EW, Yang Q (2013) Transfer learning in collaborative filtering with uncertain ratings. In: Proceedings of the Twenty-sixth AAAI Conference on Artificial Intelligence (AAAI). The AAAI Press, pp.39–55.
9. Wang G, Xie S, Liu B, Yu PS (2011) Review graph based online store review spammer detection. In: Proceedings of IEEE 11th International Conference on data mining. IEEE Press., pp. 1242–1247
10. Benevenuto F, Rodrigues T, Almeida V, Almeida J, Gonalves M (2009) Detecting spammers and content promoters in online video social networks.In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM press, pp.620–627.
11. Masum H, Zhang YC (2004) Manifesto for the reputation society. First Monday 9: 7.
12. Herlocker JK, Konstan JA, Terverrn LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans. Inf.Syst 22: 5–53.
13. Resnick P, Kuwabara K, Zeckhauser R, Friedman E (2000) Reputation systems. Communications of the ACM 43: 45–48.
14. Jsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decision support systems, 43: 618–644.
15. Liao H, Cimini G, Medo M (2012) Measuring quality, reputation and trust in online communities. In: Proceedings of the 20th International Symposium on methodologies for intelligent systems (ISMIS). Springer Lecture Notes in Artificial Intelligence, pp. 405–414
16. Liao H, Xiao R, Cimini G, Medo M (2013) Ranking users, papers and authors in online scientific communities. arXiv: 1311.3064.
17. Tacchella A, Cristelli M, Caldarelli G, Gabrielli A, Pietronero L (2012) A New Metrics for Countries' Fitness and Products' Complexity. Scientific Report 2: 723.
18. Zhou YB, Lü L, Li MH (2012) Quantifying the influence of scientists and their publications: distinguishing between prestige and popularity. New Journal of Physics 14: 033033.
19. Laureti P, Moret L, Zhang YC, Yu YK (2006) Information filtering via Iterative Refinement. Europhysics Letter 75: 1006.
20. Yu YK, Zhang YC, Laureti P, Moret L (2006) Decoding information from noisy, redundant, and intentionally distorted sources. Physica A 371: 732–744.
21. Zhou YB, Lei T, Zhou T (2011) A robust ranking algorithm to spamming. Europhysics Letter 94: 48002.
22. Lin LI (1989) A concordance correlation coefficient to evaluate reproducibility. Biometrika, 45: 255–268.
23. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46: 175.
24. Nagelkerke NJD (1991) A note on a general definition of the coefficient of determination. Biometrika, 78: 691–692.
25. Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In:Proceedings of the 19th ACM international conference on Information and knowledge management. ACM Press, pp.939–948.
26. Lee K, Caverlee J, Webb S (2010) Uncovering social spammers: social honeypots + machine learning. In:Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM Press , pp.435–442.
27. Grier C, Thomas K, Paxson V, Zhang M (2010) spam: the underground on 140 characters or less. In:Proceedings of the 17th ACM conference on computer and communications security. ACM Press, pp.27–37.
28. Zeng A, Yeung CH, Shang MS, Zhang YC (2012) The reinforcing influence of recommendations on global diversification. Europhysics Letter 97: 18005.
29. Shang MS, Lü L, Zhang YC, Zhou T (2010) Empirical analysis of web-based user-object bipartite networks. Europhysics Letter 90: 48006.
30. Barabasi AL, Albert R (1999) Emergence of Scaling in Random Networks. Science 286: 509–512.
31. Zeng A, Cimini G (2012) Removing spurious interactions in complex networks. Phys. Rev.E 85: 036101.
32. Zhang QM, Zeng A, Shang MS (2013) Extracting the information backbone in online system. PloS one 8(5): e62624.
33. Kendall M (1938) A new measure of rank correlation. Biometrika, 30: 81–93.
34. Hanley JA, Mcneil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143: 29–36.