Identifying online user reputation of user–object bipartite networks

Xiao-Lu Liu [a], Jian-Guo Liu [a,b,c,*], Kai Yang [a], Qiang Guo [a], Jing-Ti Han [b]

[a] Research Center of Complex Systems Science, University of Shanghai for Science and Technology, Shanghai 200093, PR China
[b] Data Science and Cloud Service Centre, Shanghai University of Finance and Economics, Shanghai 200433, PR China
[c] Department of Physics, Fribourg University, CH-1700, Fribourg, Switzerland

HIGHLIGHTS

- Based on the Bayesian analysis, we present a parameter-free algorithm for ranking online user reputation.
- The user reputation is calculated based on the beta distribution in terms of user ratings.
- The computation complexity of the presented algorithm is a linear function of the network size.

Identifying online user reputation based on the rating information of the user–object bipartite networks is important for understanding online user collective behaviors. Based on the Bayesian analysis, we present a parameter-free algorithm for ranking online user reputation, where the user reputation is calculated based on the probability that their ratings are consistent with the main part of all user opinions. The experimental results show that the AUC values of the presented algorithm could reach 0.8929 and 0.8483 for the MovieLens and Netflix data sets, respectively, which is better than the results generated by the CR and IARR methods. Furthermore, the experimental results for different user groups indicate that the presented algorithm outperforms the iterative ranking methods in both ranking accuracy and computation complexity. Moreover, the results for the synthetic networks show that the computation complexity of the presented algorithm is a linear function of the network size, which suggests that the presented algorithm is very effective and efficient for the large scale dynamic online systems.

1. Introduction

How to evaluate the user reputation in terms of their rating behaviors is important for the online rating systems [1–4]. Nowadays, online rating systems provide channels for users to show their preferences. However, not every user gives ratings subjectively since each user has his/her specific tastes and motivations [5–7]. Therefore, how to identify the online user reputation in terms of their ratings or selecting behaviors is important for building a reputation system [8–11].

Recently, the iterative ranking algorithms have been widely explored [12,13]. Zhou et al. [14] designed an iterative algorithm based on the correlation between the user rating and object quality vectors (short for the CR algorithm). The user reputation and object quality can be updated iteratively until the change between two iteration steps is smaller than a

* Corresponding author at: Research Center of Complex Systems Science, University of Shanghai for Science and Technology, Shanghai 200093, PR China.
E-mail address: liujg004@ustc.edu.cn (J.-G. Liu).

threshold. Liao et al. [15] developed an iterative algorithm with reputation redistribution (short for IARR), by redistributing the reputation to eliminate noisy information in the iterations. To filter out the influence of the unreliable users, Liao et al. [15] proposed the IARR2 method by introducing two penalty factors which assign smaller reputations to the users who rate small number of objects. The non-iterative online user reputation ranking algorithms are also discussed. Gao et al. [16] proposed a group-based ranking method (namely GR method) by grouping users according to their ratings. Then users' reputations could be determined by the corresponding group sizes.

In social networks analysis [17], by propagating ratings provided by multiple advisors, Teacy et al. [18] employed a probability density function to estimate the reputation of a selling agent. Zhang et al. [19] adopted the beta probability distribution to model the advisor's public reputation, which is estimated as the probability that he/she will provide fair ratings. Additionally, a rating will be regarded as the fair rating if it is consistent with the majority of the other ratings for one specific seller provided by other buyers. The expected value of the probability that a user will give fair ratings is calculated as his/her public reputation, which could be extended from the social networks to user–object bipartite networks to evaluate user reputation and object quality.

By introducing the Bayesian analysis, we present an parameter-free algorithm to rank online user reputation via the beta probability distribution, namely RBPD algorithm, where the user reputation is estimated as the probability that he/she will provide fair ratings to objects. Combining with users' personalities, the users' ratings are characterized to the positive or negative opinions. Finally, we use the expected value of the probability that the user will give fair ratings to calculate the reputation instead of the iteration process. Implementing our method for empirical networks and synthetic networks, the results show that the RBPD algorithm produces more accurate reputation ranking lists and the computation complexity is a linear function with the network size.

2. The RBPD algorithm

The rating system can be described by a weighted bipartite network [20–22], which consists of the users denoted by set U and the objects denoted by set O. The number of users, objects and ratings are denoted by $|U|$, $|O|$ and $|E|$, respectively. We use the Latin and Greek letters to represent the users and objects, respectively. The rating $r_{i\gamma}$ given by user i to object γ is the weight of the link in the bipartite network and all the ratings could be described as a rating matrix **A**. The user set U_γ is defined as the users who rate to object γ, and the object set O_i is defined as the objects which are rated by user i. Moreover, the degree of user i and object γ are denoted as k_i and ρ_γ, respectively.

2.1. The online user reputation evaluation

The reputation of user i is denoted by R_i. We use the Bayesian analysis to model the user reputation. Bayesian analysis [18] adopts a binary event to measure each of users' ratings: Fair rating (denoted by 1) or unfair rating (denoted by 0). The definition of fair rating for bipartite networks could be introduced in the following way. User i provides a rating $r_{i\gamma}$ to object γ, the rating will be judged to determine whether it is consistent with the majority of the other opinions to object γ given by other users. Determining consistency with the majority of opinions can be achieved by identifying if the rating's opinion accounts for more than 50% of all opinions [19]. We define a rating $r_{i\gamma}$ as the fair rating if it is consistent with the majority of all users' opinions, otherwise as the unfair rating.

There are two kinds of opinions to the objects: Positive and negative ones. We use a coarse-graining method to distinguish them. The quantity $r'_{i\gamma}$ is defined as the extent of fanciness via the rating $r_{i\gamma}$, from which one can discover the opinion from user i to object γ. Considering the user personality that different users tend to have different rating criteria, where some users tend to give high ratings and others tend to give low ratings, we use a normalized method to transform a rating to the extent of fanciness in the following way,

$$r'_{i\gamma} = \begin{cases} 2(r_{i\gamma} - r_i^{\min})/(r_i^{\max} - r_i^{\min}) - 1 & r_i^{\max} \neq r_i^{\min} \\ 0 & r_i^{\max} = r_i^{\min}, \end{cases} \tag{1}$$

where $r_i^{\max}$ and $r_i^{\min}$ denote the maximum and minimum ratings user i gives, respectively. In this way, all the ratings given by one specific user would be transferred to $[-1, 1]$, where the maximum and minimum ratings are mapped into 1 and -1. Specifically, for the users who always give the same ratings, their ratings are normalized to 0. The normalized rating matrix is denoted by $\mathbf{A}'$, where the element is the rating's extent of fanciness. The positive and non-positive values could be interpreted as the positive and negative opinions give by users. For all the ratings, after observing whether they are fair or not, the results are denoted by matrix **B**, where the element is Y or N (an fair rating is denoted by Y, others are denoted as N).

The reputation R_i of user i is defined as the probability θ_i that user i will provide fair ratings to objects, which lies in [0,1]. Because there is only partial information about users, the best way to estimate the probability θ_i is to use its expected value,

$$R_i = E(\theta_i). \tag{2}$$

The expected value $E(\theta_i)$ of the probability θ_i is up to the probability density function, where the beta probability distribution [18] is commonly used as a prior distribution for random variables that take on continuous values in the interval [0,1]. For user i, whether the ratings are fair or not can be expressed by the following vector,

$$D_i = \left[X_i(1), X_i(2), \ldots, X_i(k_i)\right], \tag{3}$$

where the element $X_i(j)(j = 0, 1, \ldots, k_i)$ is 0 or 1, $X_i(j) = 1$ represents the jth rating given by user i is a fair rating; Otherwise, $X_i(j) = 0$ means the jth rating provided by user i is unfair.

The vector D_i can be regarded as the prerequisite of the Bayesian analysis. For the $(k+1)$th rating of user i, the probability that user i will provide a fair rating is θ_i, i.e. $P\{X_i(k_i + 1) = 1\} = \theta_i$, where the prior probability distribution of θ_i can be regarded as a beta probability distribution,

$$f(\theta_i) = \frac{\Gamma(\alpha_s + \alpha_f)}{\Gamma(\alpha_s)\Gamma(\alpha_f)}\theta_i^{\alpha_s - 1}(1 - \theta_i)^{\alpha_f - 1}, \tag{4}$$

where $0 \le \theta_i \le 1$, and the parameters α_s and α_f generate the shape of the probability density function $f(\theta_i)$. The quantities s and f denote the number of fair and unfair ratings given by user i, respectively, and $s + f = k_i$. The function $\Gamma(\cdot)$ indicates the Gamma function, where $\Gamma(x + 1) = x\Gamma(x)$ and $\Gamma(1) = 1$. The number of fair and unfair ratings given by each user are denoted as matrix **C**.

Suppose the ratings given by user i are mutually independent, the probability that each observed result (whether the ratings are fair or not) agrees with each other is $\theta_i(1 - \theta_i)$. The likelihood ratio $L(D_i|\theta_i)$ that user i gives s fair ratings and f unfair ratings can be simulated by the maximum likelihood function of the Binomial distribution $B(k_i, \theta_i)$,

$$L(D_i|\theta_i) = \theta_i^s(1 - \theta_i)^f. \tag{5}$$

Consequently, we can get the posterior probability distribution $f(\theta_i|D_i)$ based on the Bayesian formula,

$$f(\theta_i|D_i) = \frac{f(\theta_i)\theta_i^s(1 - \theta_i)^f}{f(D_i)}, \tag{6}$$

where $f(\theta_i)$ is the Beta function presented in Eq. (4). We can prove that $f(\theta_i|D_i)$ can be represented in the form of the Beta function,

$$f(\theta_i|D_i) = \mathrm{Beta}(\theta_i; \alpha_s + s, \alpha_f + f). \tag{7}$$

Regarding to the probability distribution $f(\theta_i|D_i)$, we can get the posteriori estimation of the probability that user i will provide a fair rating when he/she gives (k_i+1)th rating, which is defined as the user reputation,

$$R_i = E(\theta_i|D_i) = \frac{\alpha_s + s}{\alpha_s + \alpha_f + s + f}. \tag{8}$$

Suppose whether the first rating given by user i is fair or not is independent, therefore the beta probability distribution is uniform when $\alpha_s = \alpha_f = 1$ in Eq. (4), indicating all values of probability are considered equally. Then one can get

$$R_i = E(\theta_i|D_i) = \frac{1 + s}{(1 + s) + (1 + f)} = \frac{s + 1}{k_i + 2}, \tag{9}$$

which indicates that the more the percentage of fair ratings user i gives, the larger reputation he/she will have.

2.2. *The online object quality evaluation*

We denote Q_γ as the quality of object γ. Regarding to the IARR2 method [15], the quality of an object is not only determined by the received weighted average rating, but also relied on the maximum reputation of the users who rate it, which could be expressed as

$$Q_\gamma = \max_{i \in U_\gamma}\{R_i\}\frac{\Sigma_{i \in U_\gamma} R_i r_{i\gamma}}{\Sigma_{i \in U_\gamma} R_i}, \tag{10}$$

where $\max_{i \in U_\gamma}\{R_i\}$ is defined as a penalty factor based on the hypothesis: If an object is rated only by low reputation users, regarding to the high ratings, this object could not be assigned with high quality. A schematic illustration of the RBPD algorithm is shown in Fig. 1.

3. Results for empirical networks

We investigate the performance of the RBPD algorithm for two empirical data sets containing ratings for movies: MovieLens and Netflix. The MovieLens data is downloaded from the GroupLens.[1] We sample a subset in which there are 1 million ratings and each user has at least 20 ratings. The Netflix data is provided by the Netflix Prize.[2] We extract a smaller data set by choosing 5000 users who have rated at least 20 movies. The MovieLens and Netflix ratings are both given by the integer ratings scaling from 1 to 5. Some basic statistical properties of two data sets are summarized in Table 1.

1 http://www.grouplens.org.
2 http://www.netflixprize.com.

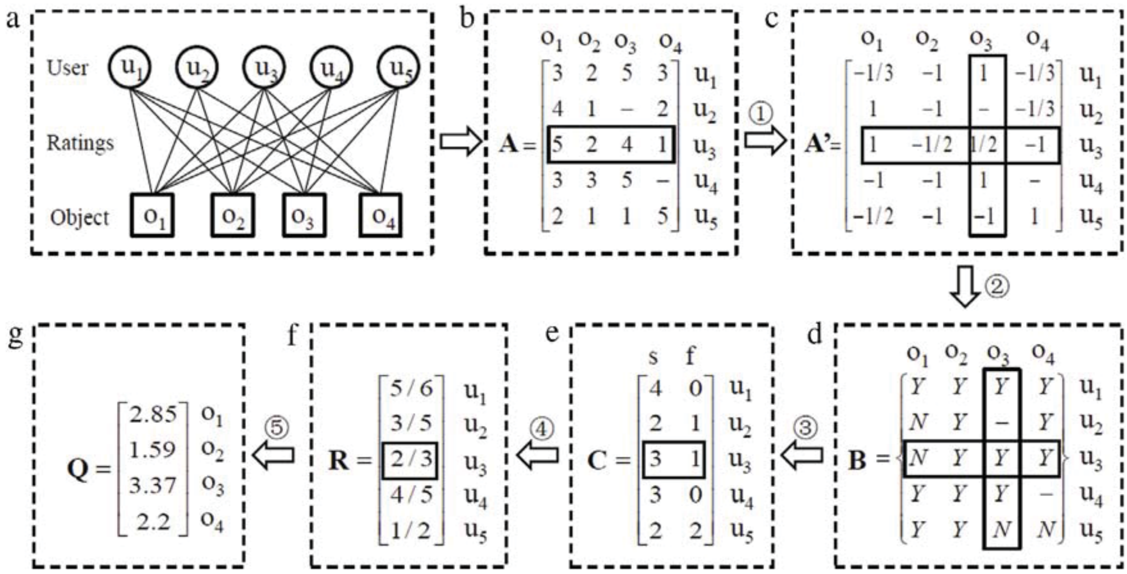

Fig. 1. A schematic illustration of the presented algorithm. The black arrow shows the steps of the procedure. (a) The weighted bipartite network describes the rating system. (b) The corresponding rating matrix, **A**. The row and column correspond to users and objects, respectively. The symbol "-" indicates there is no rating behavior. (c) The normalized rating matrix, **A′**. Take U_3 as an example, $r_{32} = 2$, $r'_{32} = 2 * (2 - 1)/(5 - 1) - 1 = -1/2$. (d) Whether each rating is fair or not could be represented as the matrix **B**. Take O_3 as an example, since $r'_{13} > 0$, $r'_{33} > 0$, $r'_{43} > 0$, $r'_{53} < 0$, the ratings given by U_1, U_3 and U_4 to O_3 are regarded as fair ratings (denoted by Y) and the rating given by U_5 is defined as an unfair rating (denoted by N). (e) The number of fair and unfair ratings, say s and f, given by each of users, could be denoted as matrix **C**. Take U_3 as an example, U_3 gives 3 fair ratings and 1 unfair rating, so $s_3 = 3, f_3 = 1$. (f) The reputation matrix, **R**. $R_3 = (1 + s_3)/(2 + s_3 + f_3) = 2/3$. (g) The quality matrix, **Q**. Take O_3 as an example, $Q_3 = 5/6 * (5 * 5/6 + 4 * 2/3 + 5 * 4/5 + 1 * 1/2)/(5/6 + 2/3 + 4/5 + 1/2) = 3.37$.

Table 1
Basic statistical properties of the empirical data sets used in this paper, where $\langle k_U \rangle$ and $\langle \rho_O \rangle$ are the average degrees of users and objects, and η denotes the network sparsity, which is the proportion of the number of links to the maximum possible number of links.

Data sets	$\|U\|$	$\|O\|$	$\langle k_U \rangle$	$\langle \rho_O \rangle$	η
MovieLens	6040	3706	166	270	0.0447
Netflix	4998	13272	179	67	0.0135

In this paper, the AUC curve is introduced to measure the ranking accuracy of different methods [23]. To calculate the AUC values, one should select a subset of objects as benchmark objects which are considered to be generally with high qualities, and the others as non-benchmark objects. The AUC value requires n times independent comparisons of one pair of benchmark and non-benchmark objects, which are randomly selected from two kinds of object sets, respectively. After the comparisons, we record n_1 as the number of times that the benchmark object has higher quality than the non-benchmark object, and n_2 as the number of times that the benchmark and the non-benchmark objects have the same qualities, then the AUC value is calculated as $\text{AUC} = (n_1 + n_2 \times 0.5)/n$. When $\text{AUC} = 1$, all the benchmark objects are ranked higher than the non-benchmark objects, while $\text{AUC} = 0.5$ means all the objects are randomly ranked.

The benchmark objects are regarded as the ones with higher qualities. In this paper, the Annual Academy Awards, popularly known as Oscars,[3] is the most influential film award among the world. Consequently, the movies nominated for the Oscars can be regarded as the high-quality movies. To calculate the AUC values, we select movies nominated for the best picture category at the Oscars, as benchmark objects. There are 162 and 286 benchmark movies in MovieLens and Netflix data sets, respectively.

Table 2 shows the AUC values and the running time T of different methods for MovieLens and Netflix data sets, respectively, in which the threshold that the change of the quality between two iteration steps is set to be 10^{-5} for the CR, IARR and IARR2 methods and the running time T is measured in seconds, from which one can find that the AUC value of the RBPD algorithm is larger than the ones obtained by the CR and IARR methods. Besides, the running time T of the RBPD algorithm is the shortest. Moreover, one can also find that the AUC value of the RBPD algorithm is smaller than that of the IARR2 method by 1.30%, while the running time T of the RBPD algorithm is 12.12% of that of the IARR2 method. For the Netflix data set, one can find that the AUC value of the RBPD algorithm is larger than the CR, IARR and IARR2 methods. In addition, the running time T of the CR and IARR methods is infinite, i.e. these two methods cannot converge to a stable value for the Netflix data set. The results for MovieLens and Netflix data sets indicate that the RBPD algorithm can generate more

[3] http://www.filmsite.org.

Table 2
The AUC values and the running time T of different methods for MovieLens and Netflix data sets, respectively. The running time T is measured in seconds and the symbol "$+\infty$" means infinite. One can find that the AUC value of the RBPD algorithm is larger than the ones obtained by the CR and IARR methods for the MovieLens data set. Although the AUC value of the IARR2 method is a little bit larger than the one obtained by the RBPD algorithm, the corresponding running time T is much longer than that of the RBPD algorithm. Meanwhile, for the Netflix data set, the RBPD algorithm obtains the best performance in quality measurement. More importantly, the computation complexity of the RBPD algorithm is a linear function of the number of links in the systems. The results are averaged over 10 independent realizations.

Data sets		CR	IARR	IARR2	RBPD
MovieLens	AUC	0.8667	0.8638	**0.9047**	0.8929
	T	43	57	306	**1.5**
Netflix	AUC	0.7850	0.7961	0.7848	**0.8483**
	T	$+\infty$	$+\infty$	1176	**1.2**

Table 3
The group information for MovieLens and Netflix data sets. The upper and lower boundaries are set as the corresponding user degrees in each group.

Data sets	k	$G1$	$G2$	$G3$	$G4$	$G5$	$G6$	$G7$	$G8$
MovieLens	U_b	150	300	450	600	750	900	1050	2400
	L_b	0	150	300	450	600	750	900	1050
Netflix	U_b	200	400	600	800	1000	1200	1400	3000
	L_b	0	200	400	600	800	1000	1200	1400

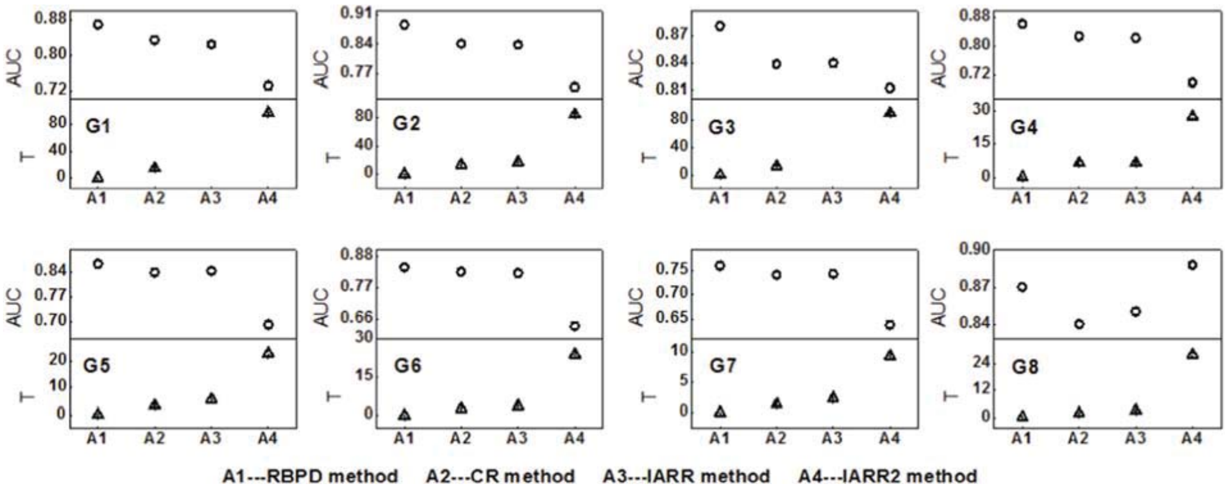

Fig. 2. The AUC values and the running time T of different methods in each of groups for the MovieLens data set. A1, A2, A3 and A4 represent RBPD, CR, IARR and IARR2 methods, respectively. One can find that in 7 groups (from $G1$ to $G7$) the AUC values of the RBPD algorithm are larger than the ones obtained by other methods, while in the group $G8$ the AUC value of the RBPD algorithm is smaller than that of the IARR2 method. Moreover, the running time T of the RBPD algorithm in every group is the shortest. The results are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.

accurate quality ranking lists. More importantly, the computation complexity of the parameter-free algorithm is a linear function of the network size, and could be easily implemented to the large scale dynamic data sets.

In empirical rating systems, users with different degrees are evolved by different mechanisms [24–28]. We investigate the performance of different methods for users with different degrees. The users are divided into 8 groups ($G1, G2, \ldots, G8$) in terms of their degrees for MovieLens and Netflix data sets, respectively, the group information is shown in Table 3.

Fig. 2 shows the AUC values and the running time T of different methods in each of groups for the MovieLens data set, from which one can find that in groups from $G1$ to $G7$ the AUC values of the RBPD algorithm are larger than those of the CR, IARR and IARR2 methods. While in group $G8$, the AUC value of the RBPD algorithm is a little smaller than that of the IARR2 method. What is more, the running time T of the RBPD algorithm in each group is the shortest.

The results for the Netflix data set are shown in Fig. 3, from which one can find that in each group the AUC values of the RBPD algorithm are larger than the ones obtained by the CR, IARR and IARR2 methods and the running time T of the RBPD algorithm is the shortest.

In the RBPD algorithm, we define the rating $r_{i\gamma}$ given from user i to object γ as a fair rating if it is consistent with the majority of opinions of all users, which is defined as 50% of all ratings. In this paper, the parameter q is denoted as the majority of opinions. Fig. 4 shows the AUC values of the RBPD algorithm with different parameters q for MovieLens and Netflix data sets. We investigate the cases where the parameter q lies in [50%,70%] and the interval is 1%. One can find that the AUC values of the RBPD algorithm increase with the parameter q and reach a maximum value, then have a decreasing

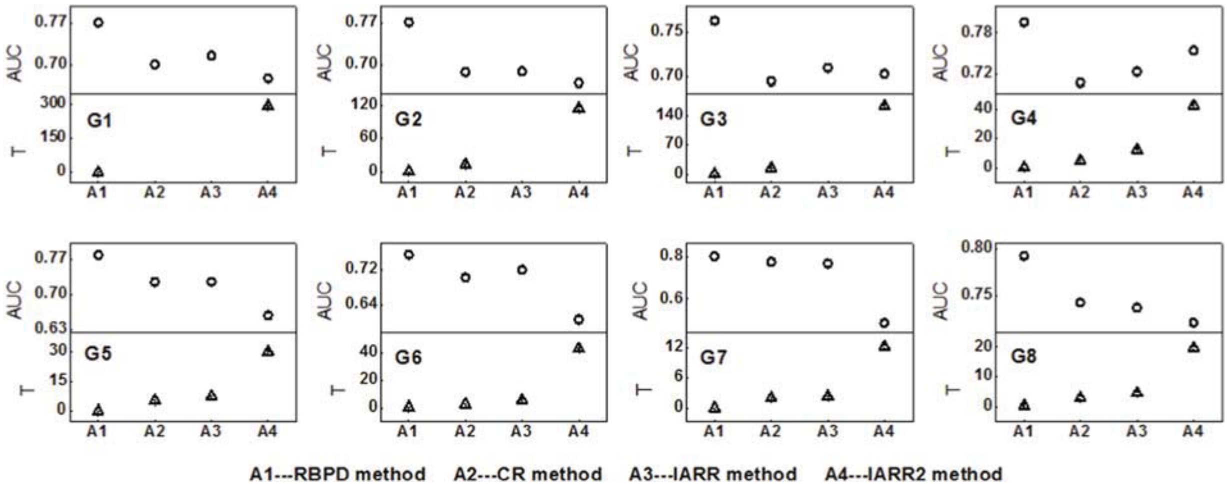

Fig. 3. The AUC values and the running time T of different methods in each of groups for the Netflix data set. The RBPD, CR, IARR and IARR2 methods are denoted as A1, A2, A3 and A4, respectively. One can find that in each group the AUC values of the RBPD algorithm are larger than the ones obtained by other methods and the computation time T of the RBPD algorithm is the shortest. The error bars are the corresponding standard deviations.

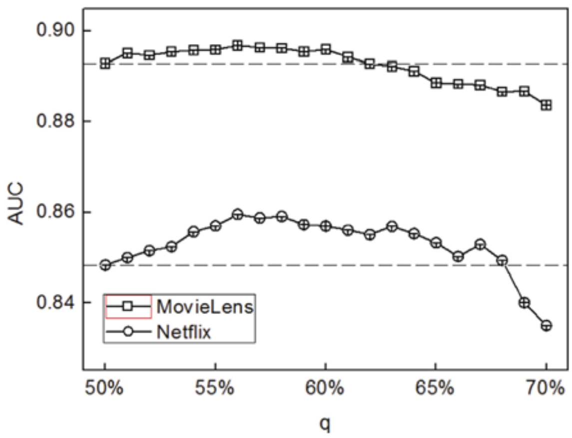

Fig. 4. The AUC values of the RBPD algorithm with different parameters q for MovieLens and Netflix data sets. One can find, when the parameter q lies in [51%,61%], [51%,68%], the AUC values of the RBPD algorithm would be larger than the results when $q = 50\%$. The results are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.

trend with parameter q for two empirical data sets. In addition, the AUC values of the RBPD algorithm when the parameter q lies in [51%,61%] are larger than the result when $q = 50\%$ for the MovieLens data set. For the Netflix data set, the case when the parameter q lies in [51%,68%] the presented algorithm could get better performance than the case where $q = 50\%$. Therefore, appropriately adjusting the definition of fair rating can improve the ranking accuracy of the RBPD algorithm.

4. Results for synthetic networks

Besides empirical networks, we investigate the performance of the RBPD algorithm for synthetic networks. When generating the synthetic networks, we set $|U| = 6000$, $|O| = 4000$, the weighted links (ratings) will be added one by one until the network sparsity η reaches 0.02. In this way, the number of the ratings is $|E| = \eta|U|\,|O| = 4.8 * 10^5$. In the synthetic networks, the users rate objects in terms of the object degree preferentially [29]. At each time step t, we choose a pair of user and object according the following possibilities,

$$p_i(t) = \frac{k_i(t) + 1}{\Sigma_{j\in U}(k_j(t) + 1)}, \tag{11}$$

$$p_\gamma(t) = \frac{k_\gamma(t) + 1}{\Sigma_{\theta\in O}(k_\theta(t) + 1)}, \tag{12}$$

where $k_i(t)$ and $k_\gamma(t)$ are the degree of user i and object γ at time step t. The rating $r_{i\gamma}$ given from user i to object γ consists of two components: the object intrinsic quality Q'_γ and the rating error $\Delta\delta_{i\gamma}$, where Q'_γ obeys a uniform distribution $U(1, 5)$

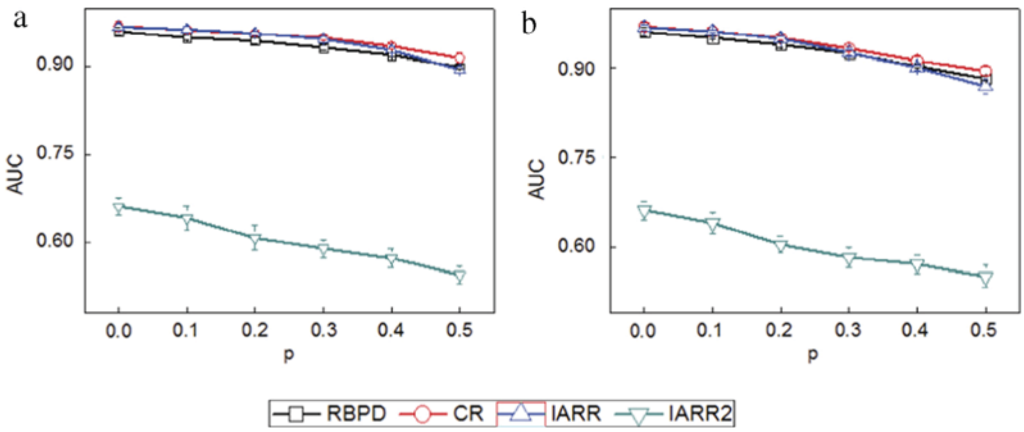

Fig. 5. The AUC values for different fractions p of distorted ratings: (a) random ratings, (b) malicious ratings. One can find that the AUC values of the RBPD, CR and IARR methods are larger than the ones obtained by the IARR2 methods. In addition, the AUC values of the RBPD, CR and IARR methods are very close. The results are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.

Table 4
The number of users $|U|$, objects $|O|$ and ratings $|E|$ of the synthetic networks, where the sparsity η is set as 0.02.

Properties	Size 1	Size 2	Size 3	Size 4	Size 5	Size 6	Size 7	Size 8
$\|U\|$	3000	6000	9000	12,000	15,000	18,000	21,000	24,000
$\|O\|$	2000	4000	6000	8,000	10,000	12,000	14,000	16,000

Table 5
The polynomial fitting results of the time T for the network size $|E|$. One can find that the time complexity of the RBPD algorithm is linear w.r.t. the number of ratings $|E|$ of the bipartite networks, which indicates that the RBPD algorithm can save a lot of time comparing with the iterative ranking methods in the reputation and quality measurement for online rating systems.

Methods	Fitting functions	Adj. R^2	Complexity
RBPD	$T = 6.4 \times 10^{-7} \times \|E\|$	0.999	$O(\|E\|)$
CR	$T = 1.2 \times 10^{-12} \times \|E\|^2 + 4.2 \times 10^{-5} \times \|E\|$	0.997	$O(\|E\|^2)$
IARR	$T = 9.1 \times 10^{-13} \times \|E\|^2 + 6.2 \times 10^{-5} \times \|E\|$	0.996	$O(\|E\|^2)$
IARR2	$T = 1.6 \times 10^{-25} \times \|E\|^4 - 7.4 \times 10^{-18} \times \|E\|^3 + 1.2 \times 10^{-10} \times \|E\|^2 + 2.2 \times 10^{-17} \times \|E\|$	0.997	$O(\|E\|^4)$

and $\Delta\delta_{i\gamma}$ obeys the normal distribution $N(0, \Delta\delta_i)$. The parameter $\Delta\delta_i$ is the rating error of user i and it is drawn from a uniform distribution $U(\Delta\delta_{\min}, \Delta\delta_{\max})$, in which $\Delta\delta_{\min} = 1$, $\Delta\delta_{\max} = 5$. Accordingly, the rating $r_{i\gamma}$ is defined as,

$$r_{i\gamma} = [Q'_\gamma + \Delta\delta_{i\gamma}], \tag{13}$$

where $r_{i\gamma}$ lies in the set {1,2,3,4,5} and it set as the close integer to the value $Q'_\gamma + \Delta\delta_{i\gamma}$.

In the synthetic networks, we suppose that there are two kinds of distorted ratings: random and malicious ratings, where the random case indicates that users will rate objects totally random, while the malicious case indicates that users will give maximum or minimum ratings. In the synthetic networks, we replace p fraction of the links with the distorted ratings. Therefore, the larger the parameter p, the more noisy information in the networks. When $p = 0$, there is all true information. While $p = 1$ represents there is not any true information.

To measure the AUC value of the synthetic networks, we select top 10% high-quality objects as benchmark objects based on their intrinsic qualities. The experimental results (see in Fig. 5) show that the AUC values of the RBPD, CR and IARR methods are larger than the ones obtained by the IARR2 methods with different fractions of distorted ratings. Meanwhile, the differences of the AUC values of the RBPD, CR and IARR methods are very close, which indicates that, comparing with the iterative algorithms, the presented parameter-free algorithm could generate closely accurate reputation results.

Furthermore, we investigate the computation time for network sizes which can be seen in Table 4. The experimental results (see in Fig. 6 and Table 5) show that the computation time T for different synthetic networks which is measured by seconds. For different synthetic networks, the computation times T of the RBPD algorithm are the shortest. It should be emphasized that the computation complexity of the RBPD algorithm is a linear function of the network size.

5. Conclusion and discussions

By taking into account the Bayesian analysis, we evaluate the online user reputation based on the beta distribution. Firstly, we use a normalized method to transform a rating to the binary event (positive or negative opinion), which could be

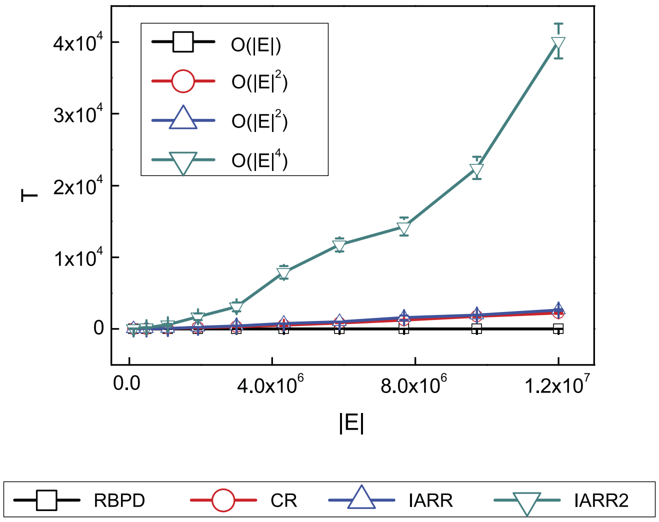

Fig. 6. The computation time T for different network sizes. One can find that the time T of the RBPD algorithm is the shortest with each number of ratings $|E|$ for the synthetic networks. The results are averaged over 10 independent realizations. The error bars are the corresponding standard deviations.

defined as the fair rating if it is consistent with the majority of opinions to the object. Then based on the beta distribution, the expected value of the probability that the user will give is calculated as his/her reputation.

The experimental results for empirical networks show that the AUC values of the RBPD algorithm can reach 0.8929 and 0.8483 in MovieLens and Netflix data sets, which outperforms the CR and IARR methods. Although the IARR2 method identifies the qualities more accurately than the RBPD algorithm for the MovieLens data set, the corresponding computation time T is much longer than that of the RBPD algorithm. Furthermore, we investigate the performances of different methods for users with different degrees. The results for two data sets indicate that in most groups the RBPD algorithm outperforms other methods in both ranking accuracy and running time. The results for synthetic networks show that the RBPD algorithm could accurately evaluate the user reputation and object quality. Meanwhile, the RBPD algorithm has great advantages for the incremental data, which is very suitable for reputation measurement for large scale dynamic online systems.

We denote $|U|$, $|O|$ and $|E|$ as the number of users, objects and ratings in the bipartite network, the computation complexity of the RBPD algorithm is $O(|U|\langle k_U\rangle+|O|\langle k_O\rangle+|U|\langle k_U\rangle+|O|\langle k_O\rangle)$, where the first term accounts for the calculation of the normalized method (see Eq. (1)), the second term accounts for the calculation of the results whether it is fair rating for each of ratings, the third term accounts for the calculation of the user reputation (see Eq. (9)), and the last term accounts for the calculation of the object quality (see Eq. (10)). Substituting the equation $|U|\langle k_U\rangle = |O|\langle k_O\rangle = |E|$, one has the fact that the computation complexity of the presented algorithm is $O(|E|)$, which corresponds with the results in Fig. 6 and Table 5.

Different users have different behaviors, Liu et al. [24] found that small-degree users prefer to rate popular objects, while the large-degree users like to select unpopular ones. Taking into account the specific behavior patterns for different user groups, the RBPD algorithm could identify the online user reputation more accurately. Specifically, the presented algorithm is parameter-free, which is helpful to be implemented for different online rating systems. Subsequently, we could identify the online user reputation in view of truth-finding algorithms [30–32] in our future work.

Acknowledgments

We acknowledge GroupLens Research Group for providing us MovieLens data and the Netflix Inc. for Netflix data. This work is supported by the National Natural Science Foundation of China (Grant Nos. 71271126, 61374177, and 71371125), the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the Shuguang Program Project of Shanghai Educational Committee (Grant No. 14SG42).

References

[1] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. USA 107 (10) (2010) 4511–4515.
[2] L. Muchnik, S. Aral, S.J. Taylor, Social influence bias: a randomized experiment, Science 341 (6146) (2013) 647–651.
[3] M. Medo, J.R. Wakeling, The effect of discrete vs. continuous-valued ratings on reputation and ranking systems, Europhys. Lett. 91 (4) (2010) 48004.
[4] Z. Yang, Z.K. Zhang, T. Zhou, Anchoring bias in online voting, Europhys. Lett. 100 (2012) 68002.
[5] A. Zeng, G. Cimini, Removing spurious interactions in complex networks, Phys. Rev. E 85 (2012) 036101.
[6] Q.M. Zhang, A. Zeng, M.S. Shang, Extracting the information backbone in online system, PLoS One 8 (5) (2013) e62624.
[7] M. Allahbakhsh, A. Ignjatovic, An iterative method for calculating robust rating scores, IEEE Trans. Parallel Distrib. 26 (2015) 340–350.
[8] B. Li, R.H. Li, I. King, M.R. Lyu, J.X. Yu, A topic-biased user reputation model in rating systems, Knowl. Inf. Syst. (2014) 1–27.

[9] Z. Noorian, J. Zhang, Y. Liu, S. Marsh, M. Fleming, Trust-oriented buyer strategies for seller reporting and selection in competitive electronic marketplaces, Auton. Agents Multi-Agent 28 (6) (2014) 896–933.
[10] A. Galletti, G. Giunta, G. Schmid, A mathematical model of collaborative reputation systems, Int. J. Comput. Math. 89 (17) (2012) 2315–2332.
[11] F. Fouss, Y. Achbany, M. Saerens, A probabilistic reputation model based on transaction ratings, Inform. Sci. 180 (11) (2010) 2095–2123.
[12] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (5) (1999) 604–632.
[13] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, Comput. Netw. ISDN Syst. 30 (1–7) (1998) 107–117.
[14] Y.B. Zhou, T. Lei, T. Zhou, A robust ranking algorithm to spamming, Europhys. Lett. 94 (4) (2011) 48002.
[15] H. Liao, A. Zeng, R. Xiao, Z.M. Ren, D.B. Chen, Y.C. Zhang, Ranking reputation and quality in online rating systems, PLoS One 9 (5) (2014) e97146.
[16] J. Gao, Y.W. Dong, M.S. Shang, S.M. Cai, T. Zhou, Group-based ranking method for online rating systems with spamming attacks, Europhys. Lett. 110 (2015) 28003.
[17] G.N. Wang, H. Gao, L. Chen, D.N.A. Mensah, Y. Fu, Predicting positive and negative relationships in large social networks, PLoS One 10 (2015) e0129530.
[18] W.T.L. Teacy, J. Patel, N.R. Jennings, M. Luck, Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model, in: Proceedings of 4th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, 2006, pp. 183–198.
[19] J. Zhang, R. Cohen, A framework for trust modeling in multiagent electronic marketplaces with buying advisors to consider varying seller behavior and the limiting of seller bids, ACM Trans. Intell. Syst. Technol. 4 (2013) 24.
[20] J.G. Liu, L. Hou, X. Pan, Q. Guo, T. Zhou, Stability of similarity measurements for bipartite networks, Sci. Rep. 6 (2016) 18653.
[21] H. Liao, A. Zeng, Reconstructing propagation networks with temporal similarity, Sci. Rep. 5 (2015) 11404.
[22] G.Y. Shi, Y.X. Kong, H. Liao, Y.C. Zhang, Analysis of ground state in random bipartite matching, Physica A 444 (2016) 397–402.
[23] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.
[24] J.G. Liu, T. Zhou, Q. Guo, Information filtering via biased heat conduction, Phys. Rev. E 84 (3) (2011) 037101.
[25] X.L. Liu, Q. Guo, L. Hou, C. Cheng, J.G. Liu, Ranking online quality and reputation via the user activity, Physica A 436 (2015) 629–636.
[26] J. Ni, Y.L. Zhang, Z.L. Hu, W.J. Song, L. Hou, Q. Guo, J.G. Liu, Ceiling effect of online user interests for the movies, Physica A 402 (2014) 134–140.
[27] D.D. Zhao, A. Zeng, M.S. Shang, J. Gao, Long-term effects of recommendation on the evolution of online systems, Chin. Phys. Lett. 30 (11) (2013) 118901.
[28] H. Liao, A. Zeng, Y.C. Zhang, Predicting missing links via correlation between nodes, Physica A 436 (2015) 216–223.
[29] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[30] G.J. Qi, C.C. Aggarwal, J. Han, T. Huang, Mining collective intelligence in diverse groups, in: Proceedings of the 22nd International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2013, pp. 1041–1052.
[31] D. Wang, L. Kaplan, H. Le, T. Abdelzaher, On truth discovery in social sensing: A maximum likelihood estimation approach, in: Proceedings of the 11th International Conference on Information Processing in Sensor Networks, 2012, pp. 233–244.
[32] J. Pasternack, D. Roth, Making better informed trust decisions with generalized fact-finding, in: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 22(3), 2011, p. 2324.