

## A topic-biased user reputation model in rating systems

Baichuan Li · Rong-Hua Li · Irwin King ·  
Michael R. Lyu · Jeffrey Xu Yu

Received: 23 May 2013 / Revised: 28 January 2014 / Accepted: 3 August 2014 /  
Published online: 19 August 2014  
© Springer-Verlag London 2014

**Abstract** In rating systems like Epinions and Amazon’s product review systems, users rate items on different topics to yield item scores. Traditionally, item scores are estimated by averaging all the ratings with equal weights. To improve the accuracy of estimated item scores, user reputation [a.k.a., user reputation (UR)] is incorporated. The existing algorithms on UR, however, have underplayed the role of topics in rating systems. In this paper, we first reveal that UR is topic-biased from our empirical investigation. However, existing algorithms cannot capture this characteristic in rating systems. To address this issue, we propose a topic-biased model (TBM) to estimate UR in terms of different topics as well as item scores. With TBM, we develop six topic-biased algorithms, which are subsequently evaluated with experiments using both real-world and synthetic data sets. Results of the experiments demonstrate that the topic-biased algorithms effectively estimate UR across different topics and produce more robust item scores than previous reputation-based algorithms, leading to potentially more robust rating systems.

---

B. Li (✉)  
Baidu, Shenzhen, China  
e-mail: tobcli@gmail.com

R.-H. Li  
Guangdong Province Key Laboratory of Popular High Performance Computers, Shenzhen University,  
Shenzhen, China

I. King · M. R. Lyu  
Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, Shenzhen Research  
Institute, The Chinese University of Hong Kong, Shenzhen, China

I. King · M. R. Lyu  
Department of Computer Science and Engineering, The Chinese University of Hong Kong,  
Shatin, N.T., Hong Kong

J. X. Yu  
Department of Systems Engineering and Engineering Management, The Chinese University  
of Hong Kong, Shatin, N.T., Hong Kong

**Keywords** User reputation · Topic-biased model · Item scores · Rating system · User study

## 1 Introduction

Since the advent of Web 2.0, there has been an increasing number of rating systems, where users rate items to produce item scores, e.g., App rating system in Apple's App Store,<sup>1</sup> product review in Epinions,<sup>2</sup> voting systems in community question answering services.<sup>3</sup> Item scores are thus utilized for making decisions, such as ranking items and recommending good quality ones. At the very beginning, the ratings of an item are averaged out, with all users having equal weights. However, this averaging activity has poor accuracy when spammers are involved. To minimize the influence of spammers' ratings, user reputation has been investigated in rating systems. In this connection, a number of studies have proposed reputation-based algorithms [8, 16, 18, 30, 31] to estimate user reputation in rating systems. Although there is no formal definition for user reputation, a running definition in rating systems is given below:

**Definition 1.1** (*User reputation (UR)*) In a rating system, user reputation evaluates a user's ability to give reliable, fair, and trustworthy ratings. It is measured by the degree to which a user ratings are close to the items' "true" scores.

This running definition assumes that an item has a "true" score which represents its objective quality. For example, in a product review system like Epinions, a computer's "true" score reflects its quality judged by its CPU, memory, hard disk, screen, price, etc. In a paper review system, a paper's "true" score comes from its significance, novelty, technical quality, clarity, etc. Therefore,

- if a user ratings always deviate from items' "true" scores, his/her UR is low;
- if a user ratings are always close to items' "true" scores, his/her UR is high.

It is worth noting that in this paper, we only consider rating systems in which items' "true" scores are objective, which means that ratings are given by users' objective evaluations, rather than personal preferences (e.g., in movie/music rating systems, ratings are usually affected by user preference). Therefore, UR only reflects the quality of one's ratings when comparing to "true" scores. However, "true" scores do not exist in almost all rating systems. Thus, the above reputation-based algorithms estimate items' "true" scores from the weighted average of users' ratings. The higher UR the user has, the more weights the user's ratings are given, and vice versa. Iteratively, UR and item scores are estimated. Due to the introduction of the reputation-based algorithms, the accuracy of estimating item scores has been improved in rating systems [16, 30].

However, the existing reputation-based algorithms have not considered one of the most essential characteristics of rating systems—*topics*. In rating systems, users usually rate items on different topics. Therefore, topic information may have influence on UR due to users' various profiles, such as educational background, knowledge, and expertise. We validate this assumption on the Epinion data set [28], which consists of 0.9 million ratings applied to 296,277 items by 22,166 users. In addition, each rating is attached with a helpfulness score (ranges from 1 to 5, the higher the better), which reflects the quality of the rating and UR.

<sup>1</sup> <https://itunes.apple.com/>.

<sup>2</sup> <http://www.epinions.com/>.

<sup>3</sup> <http://answers.yahoo.com/>.

**Table 1** An example of users' ratings

Item ID	Topic ID	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	"True" score
1	1	9	4	8	9	8	8.8
2	1	6	9	5	5	6	5.6
3	1	7	4	8	9	9	8.7
4	2	4	8	9	8	8	8.3
5	2	8	4	4	4	4	3.9
6	2	10	6	6	6	6	5.6

In this data set, each item belongs to one category (topic), such as electronics, computer hardware, sports and outdoors. To investigate the influence of topics to users' ratings, we calculate the mean variance of rater's helpfulness score and observe that:

- Among all users, the mean variance of rater's helpfulness score is 1.264, while the mean variance of rater's helpfulness score within the same category is 0.542 (decreases 57.1 %).
- Among 1,316 users who rate at least 100 products, the mean variance of rater's helpfulness score is 1.187, while the mean variance of rater's helpfulness score within the same category is 0.745 (decreases 37.2 %).

These observations reveal that in real-world rating systems, UR fluctuate across topics but is more stable within a topic. To a particular user, on some topics his UR is relatively low while in other topics his UR is higher. In other words, users have different UR on various topics. Therefore, it would be better to model UR at topic level. However, existing reputation-based algorithms only utilize a single value to model holistic UR and cannot capture this characteristic in rating systems.

We use a toy example to illustrate this problem. Consider the following case shown in Table 1, where five users give ratings to six items, together with these items' "true" scores. We estimate UR for the five users using a state-of-the-art reputation-based algorithm L1-AVG [18]. The results are 0.80, 0.84, 0.92, 0.90, and 0.92 (UR is normalized to [0,1], the higher, the better). The above results only give an overall estimation of UR, from which it is hard to distinguish UR of  $u_1$  and  $u_2$  since they are close to each other. However, when we look into users' ratings on different topics, we observe that  $u_1$  has good ratings to items in Topic 1 but poor rating to items in Topic 2. Similarly,  $u_2$ 's ratings for items in Topic 1 are differ from item scores greatly but ratings for items in Topic 2 are quite well. Therefore, if we estimate UR as a whole, we will ignore these fine-grained information and assign inaccurate weighting for these topic-biased users (e.g.,  $u_1$ 's weighting is over-estimated for Topic 2, but under-estimated for Topic 1), which furthermore affects the accuracy of estimation on item scores.

To address the above problem, we propose a topic-biased model (TBM) to estimate UR on different topics together with item scores. The model assumes that a user has a series of UR with regard to different topics, and an item has a topic distribution over topics. As such, UR on a topic is estimated from the difference between the user ratings and item scores within the topic. In addition, an item score is derived from a total of weighted average of users' ratings on those topics to which the item belongs. Iteratively, TBM computes UR and item scores until they converge to fixed values. Based on TBM, we develop six topic-biased algorithms, with four advantages highlighted as follows:

1. **Robustness.** The six algorithms are robust to unreliable ratings, especially for ratings on biased topics.

2. **Parallelization.** The estimation of UR on each topic is independent and parallel computation can be easily applicable.
3. **Adaptability.** As the setting of topics is flexible, we can manually encode topics, utilize existing ontologies [10,23], or learn from topic models [2,13].
4. **Single parameter.** Every algorithm has only one parameter, which is easy to tune in practice.

To evaluate the performance of the six topic-biased algorithms, we carry out experiments using both real-world data and synthetic data. The experimental results are evaluated using four criteria: effectiveness in estimating UR, accuracy in estimating item scores, robustness to unreliable ratings, and convergence property. To the best of our knowledge, our approach is the first attempt to explore the relationship between UR and topics, and the main contributions of the research are threefold:

- First, we provide empirical evidence to show that UR in rating systems fluctuates across topics, which is a crucial issue but did not receive much attention in previous reputation-based algorithms;
- More importantly, we propose a TBM, with six topic-biased algorithms developed, to solve the fluctuation of UR across topics;
- Finally, we conduct extensive experiments applying both a real-world data set and synthetic data sets to demonstrate the advantages of our topic-biased framework in accuracy of estimating UR and item scores, and robustness to unreliable ratings.

Findings of the research thus bring new insights into building better rating systems. For example, our proposed TBM can be employed widely in various rating systems, including product rating of Amazon, rating tasks in crowdsourcing systems, and academic paper review systems.

This paper proceeds as follows. We briefly introduce the related work in Sect. 2. Section 3 elaborates on TBM and topic-biased algorithms. Section 4 presents the experimental results using the data collected from a user study. In Sect. 5, we conduct intensive simulations on synthetic data. Finally, we conclude this work in Sect. 6.

## 2 Related work

### 2.1 Reputation system

A rating system is usually embodied in a reputation system [11,25,26], which computes item scores (or item reputation) through collecting feedbacks that other entities (users) hold about the items. Common items include services, goods, and entities. There are different approaches to collect feedbacks and compute item scores, such as analyzing the content and the collaboration process, which is referred as *content-driven* reputation systems [1,7,29]. Another approach by *user-driven* reputation systems is to provide a rating system in which users give explicit feedbacks via rating items. For instance, the mobile application rating system in Apple's App Store, Amazon and Epinion's product review systems, voting systems in community question answering services, etc. There is a large body of work for the user-driven reputation systems. A survey can be found in [15].

## 2.2 Reputation-based algorithms for rating systems

In recent years, reputation-based methods [20] for rating systems have attracted increasing attention from researchers. Mizzaro [22] proposes a reputation-based ranking algorithm (Mizz) for the assessment of scholarly papers. In [22], UR is measured by the root of  $L_1$ -distance between paper quality and reader rating. Subsequently, Yu et al. [30] present an iterative refinement algorithm (YZLM) for bipartite rating systems, with UR being considered. In [30], UR is measured by the inverse of square  $L_2$ -distance between item ranking and user rating. More recently, Zhou et al. [31] propose a similar reputation-based algorithm for rating systems, where the reputation is measured by the correlation coefficient between the user rating and item ranking. However, all of these reputation-based algorithms cannot guarantee convergence, thereby they are very hard to apply in practice. For addressing the concern on convergence, Kerchove et al. [8] propose a convergence algorithm (dKVD) from an optimization perspective. However, the convergence of the algorithm is severely affected by the parameter settings. In addition, the algorithm suffers when the number of spamming users is relatively large. To address the convergence and spamming issue, Li et al. develop six reputation-based algorithms in [18], which is mostly related to this work. In this paper, both  $L_1$  and  $L_2$  distances between item ranking and user rating, and three difference aggregation approaches (i.e., average, maximum, and minimum) are employed to estimate UR. Comparing with the above algorithm, these algorithms' convergence is not sensitive to the initializations, and they are more immune to spamming users.

The above reputation-based algorithms estimate item scores and user rating reputation iteratively. Given a set of users  $U$ , a set of items  $O$ , and user ratings  $\{R_{ij} (u_i \in U, o_j \in O)\}$ , we use  $r_j^s$  to denote the  $j$ th estimated item score and  $c_i$  to denote the  $i$ th raters' UR at round  $s$ . Let the function  $\mathcal{D}(\{R_{ij}, r_j\}_{o_j \in N_i})$  measure the divergence between  $u_i$ 's ratings and item scores for those items  $u_i$  rated ( $\mathcal{D}(\cdot) \in [0, 1]$ ), the general reputation-based model works as follows:

$$r_j^{s+1} = \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \cdot c_i^s, \quad (1)$$

$$c_i^{s+1} = 1 - \mathcal{D}(\{R_{ij}, r_j^{s+1}\}_{o_j \in N_i}), \quad (2)$$

where  $M_j$  is the set of users who have rated  $o_j$  and  $N_i$  is the set of items that  $u_i$  has rated. In the above iterative system, Eq. 1 estimates  $r_j$  using weighted average of its ratings. At the same time, Eq. 2 estimates  $c_i$  with the divergence between  $u_i$ 's ratings and estimated item scores. The larger the difference, the lower the rating reputation, and vice versa.

With various divergence measurement functions, we can develop different algorithms. For instance, Kerchove et al. [8] employ mean square errors, and Li et al. [18] further explore  $L_1$  and  $L_2$  distances to measure the differences. However, all the above reputation-based algorithms do not consider UR as topic-biased and only employ a single value to model UR, without taking into account UR across different topics.

## 2.3 Topic-sensitive ranking

Topic-sensitive ranking is also relevant to our work, of which the most representative algorithm is topic-sensitive PageRank [12]. Originally, PageRank [3] computes a single PageRank value for each web page. But the single PageRank value cannot capture the importance of the page on various topics. To address this concern, Haveliwala [12] proposes the topic-sensitive PageRank algorithm, which computes a set of PageRank values biased toward different

**Table 2** Notation used in this paper

Symbol	Description
$U$	A set of users
$O$	A set of items
$T$	A set of topics
$u_i$	The $i$ th user
$o_j$	The $j$ th item
$t_k$	The $k$ th topic
$R$	Rating matrix
$R_{ij}$	$u_i$ 's rating to $o_j$
$c_i$	$u_i$ 's reputation
$c_{ik}$	$u_i$ 's reputation on $t_k$

topics. The experimental results demonstrate that topic-sensitive PageRank yields more accurate searching results. Similar topic-sensitive ranking algorithms have been applied in tag ranking [14], question retrieval [5,17], and expertise retrieval [9]. Different from topic-sensitive ranking which works on homogeneous graphs (e.g., web pages), the current study focuses on reputation-based ranking on heterogeneous graphs (e.g., users and items).

### 3 Model

This section describes the topic-biased model and algorithms for estimating UR and item scores. We first present our topic-biased model for addressing UR on various topics (Sect. 3.1). Based on this model, we propose six topic-biased algorithms. After that, we employ a toy example to demonstrate the advantages of our algorithms over traditional reputation-based algorithms (Sect. 3.2). In the end, we analyze the property of convergence for one representative algorithm (Sect. 3.3). Table 2 summarizes the notation used throughout the paper.

#### 3.1 Topic-biased model and algorithms

Traditional reputation-based model shown in Eqs. 1 and 2 leverages a single value (i.e.,  $c_i$ ) to model UR. However, it may suffer when users give unreliable ratings to items of some particular topics, while providing reliable ratings to items of other topics, as shown in the example in Sect. 1.

To better model user rating reputation over different topics, we propose TBM. Utilizing the same notations in the above, we further incorporate a set of topics  $T$  and topic distributions of items  $B$  (an  $|O| \times |T|$  matrix with the element  $b_{jk}$  representing the degree to which that  $o_j$  belongs to  $t_k$ ). Let the function  $\mathcal{D}_k(\{R_{ij}, r_j\}_{o_j \in N_i})$  measure the divergence between  $u_i$ 's ratings and item scores for those items  $u_i$  rated ( $\mathcal{D}_k(\cdot) \in [0, 1]$ ) on the  $k$ th topic, then TBM iteratively estimates item scores  $\{r_1, r_2, \dots, r_j\}$  and user reputation over various topics  $C$  (an  $|U| \times |T|$  matrix with  $c_{ik}$  representing  $u_i$ 's rating reputation on  $t_k$ . Initially,  $c_{ik}^0 = 1$ ). We have:

$$r_j^{s+1} = \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \left( \sum_{t_k \in T} c_{ik}^s b_{jk} \right), \tag{3}$$

$$c_{ik}^{s+1} = 1 - \mathcal{D}_k(\{R_{ij}, r_j^{s+1}\}_{o_j \in N_i}). \tag{4}$$

**Table 3** Different measurement functions in traditional reputation-based algorithms and corresponding topic-biased algorithms

Measurements in traditional reputation-based algorithms		Measurements in topic-biased algorithms	
Algorithm	$\mathcal{D}(\cdot)$	Algorithm	$\mathcal{D}_k(\cdot)$
$L_1$ -AVG	$\frac{\lambda \sum_{o_j \in N_i}  R_{ij} - r_j^{s+1} }{ N_i }$	TB- $L_1$ -AVG	$\frac{\lambda \sum_{o_j \in N_i} b_{jk}  R_{ij} - r_j^{s+1} }{\sum_{o_j \in N_i} b_{jk}}$
$L_1$ -MAX	$\lambda \max_{o_j \in N_i}  R_{ij} - r_j^{s+1} $	TB- $L_1$ -MAX	$\lambda \max_{o_j \in N_i} b_{jk}  R_{ij} - r_j^{s+1} $
$L_1$ -MIN	$\lambda \min_{o_j \in N_i}  R_{ij} - r_j^{s+1} $	TB- $L_1$ -MIN	$\lambda \min_{o_j \in N_i} b_{jk}  R_{ij} - r_j^{s+1} $
Square-AVG	$\frac{\frac{\lambda}{2} \sum_{o_j \in N_i} (R_{ij} - r_j^{s+1})^2}{ N_i }$	TB-Square-AVG	$\frac{\frac{\lambda}{2} \sum_{o_j \in N_i} b_{jk} (R_{ij} - r_j^{s+1})^2}{\sum_{o_j \in N_i} b_{jk}}$
Square-MAX	$\frac{\lambda}{2} \max_{o_j \in N_i} (R_{ij} - r_j^{s+1})^2$	TB-Square-MAX	$\frac{\lambda}{2} \max_{o_j \in N_i} b_{jk} (R_{ij} - r_j^{s+1})^2$
Square-MIN	$\frac{\lambda}{2} \min_{o_j \in N_i} (R_{ij} - r_j^{s+1})^2$	TB-Square-MIN	$\frac{\lambda}{2} \min_{o_j \in N_i} b_{jk} (R_{ij} - r_j^{s+1})^2$

Equation 3 utilizes the sum of productions between UR and topic distributions as the weighting of user ratings. If  $b_{jk}$  is very small,  $u_i$ 's UR on topic  $k$  will not give much influence to  $o_j$ 's item score. Through decomposing  $c_i$  to  $\sum_{t_k \in T} c_{ik} b_{jk}$ , TBM gives higher weighting to UR on relevant topics and reduces the influence of UR on irrelevant topics to an item score. It is worth noting that TBM is a generalization of traditional reputation-based algorithms. When  $b_{jk} = 1/|T|$  for all  $t_k$ , TBM degenerates to the traditional reputation-based models. From the above equations, we find that TBM only contains one parameter (i.e.,  $\lambda$ ), which makes it feasible to tune in practice. Furthermore, the computation of  $c_{ik}$  is independent and can be easily parallelized.

Using different divergence measurement functions ( $\mathcal{D}_k(\cdot)$ ), we propose the topic-biased versions of the corresponding algorithms. Since the algorithms in [18] have great advantages than other reputation-based algorithms in convergence and robustness, we adopt them as sample algorithms, as shown in Table 3. For instance, if  $\mathcal{D}_k(\cdot)$  employs the average  $L_1$  distance, the iterative system of TB- $L_1$ -AVG algorithm is:

$$r_j^{s+1} = \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \left( \sum_{t_k \in T} c_{ik}^s b_{jk} \right), \tag{5}$$

$$c_{ik}^{s+1} = 1 - \frac{\lambda \sum_{o_j \in N_i} b_{jk} |R_{ij} - r_j^{s+1}|}{\sum_{o_j \in N_i} b_{jk}}, \tag{6}$$

where  $\lambda \in (0, 1)$  is a damping factor. It is worth noting that our TBM is not limited to algorithms in [18]. We can easily extend all the reputation-based algorithms introduced in Sect. 2.2 to topic-biased algorithms using TBM.

### 3.2 A toy example

As a toy example, we estimate the UR of five users on two topics in Table 1 using TB- $L_1$ -AVG. Table 4 reports the estimated UR of these users, ranging from 0 to 1. As can be seen from Table 4,  $L_1$ -AVG only produces an overall estimation but fails to identify  $u_1$  and  $u_2$ ' varying rating reputation on topic 1 and topic 2. When examining the results of TB- $L_1$ -AVG,

**Table 4** UR of toy examples

	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$
$L_1$ -AVG	0.80	0.84	0.92	0.90	0.92
TB- $L_1$ -AVG (Topic 1)	0.90	0.76	0.92	0.86	0.88
TB- $L_1$ -AVG (Topic 2)	0.72	0.96	0.94	0.96	0.96

we can find that, in TB- $L_1$ -AVG (topic 1)/TB- $L_1$ -AVG (topic 2),  $u_2/u_1$  has the lowest UR. Therefore, TB- $L_1$ -AVG is able to distinguish  $u_1$  and  $u_2$  from other users on these two topics, respectively.

### 3.3 Convergence property of topic-biased algorithms

This section analyzes the convergence property of topic-biased algorithms. As shown in Eqs. 5 and 6, each topic-biased algorithm is a iteration system. To be specific, at round  $s + 1$ ,  $r_j^{s+1}$  is computed based on  $c_{ik}^s$  estimated at round  $s$ . Next, new  $c_{ik}^{s+1}$  is derived from  $r_j^{s+1}$ . Iteratively,  $r_j$  and  $c_{ik}$  are updated. In rating systems, we usually need to deal with large-scale data. Thus, it is essential to consider the model’s time complexity. From Eqs. 3 and 4, we can easily find that the time complexity of TBM is linear to the number of ratings in each round (i.e., the time complexity in each round is  $O(|T| \cdot |\{R_{ij}\}|)$ ). Thus, the time cost depends on the number of iterations. By using similar approach to [18], we show that the above topic-biased algorithms converge and the rate of convergence is exponential. In other words,  $r_j$  and  $c_{ik}$  in topic-biased algorithms only change little after a few iterations. In the following, we take TB- $L_1$ -AVG as an example (the proofs of TB- $L_1$ -AVG and TB- $L_1$ -AVG are given in the “Appendix”) and it is easy to obtain similar results for other algorithms.

**Lemma 1** *Let  $|r_\alpha^1 - r_\alpha^0| = \max_j |r_j^1 - r_j^0|$ , then we have  $r_j^{s+1} - r_j^s \leq \lambda^s \cdot |r_\alpha^1 - r_\alpha^0|$ .*

*Proof* We prove it by induction. If  $s = 1$ , then,

$$\begin{aligned}
 |r_j^2 - r_j^1| &= \left| \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} c_{ik}^1 b_{jk} - \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} c_{ik}^0 b_{jk} \right| \\
 &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^1 - c_{ik}^0| \\
 &= \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \left| \frac{\lambda \sum_{o_j \in N_i} b_{jk} (|R_{ij} - r_j^0| - |R_{ij} - r_j^1|)}{\sum_{o_j \in N_i} b_{jk}} \right| \\
 &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot \lambda \cdot \sum_{o_j \in N_i} \frac{b_{jk}}{\sum_{o_j \in N_i} b_{jk}} |r_j^1 - r_j^0| \\
 &= \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot \lambda \cdot |r_j^1 - r_j^0| \\
 &= \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \cdot \lambda \cdot |r_j^1 - r_j^0| \\
 &\leq \lambda \cdot |r_\alpha^1 - r_\alpha^0|.
 \end{aligned}$$



Assume when  $s = t$  the lemma holds, then we show  $s = t + 1$  the lemma still holds.

$$\begin{aligned}
 |r_j^{t+1} - r_j^t| &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^t - c_{ik}^{t-1}| \\
 &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot \lambda \cdot \sum_{o_j \in N_i} \frac{b_{jk}}{\sum_{o_j \in N_i} b_{jk}} |r_j^t - r_j^{t-1}| \\
 &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot \lambda \cdot \sum_{o_j \in N_i} \frac{b_{jk}}{\sum_{o_j \in N_i} b_{jk}} \lambda^{t-1} |r_j^1 - r_j^0| \\
 &\leq \lambda^t |r_j^1 - r_j^0| \\
 &\leq \lambda^t |r_\alpha^1 - r_\alpha^0|.
 \end{aligned}$$

This completes the proof. □

**Theorem 1**  $r_j$  converges to a unique fixed point in TB-L<sub>1</sub>-AVG.

*Proof* First, we show  $r_j$  converges to a fixed point. Let  $|r_\alpha^1 - r_\alpha^0| = \max_j |r_j^1 - r_j^0|$ , then there exists  $N$  satisfying

$$\lambda^N \leq \frac{(1 - \lambda)\epsilon}{|r_\alpha^1 - r_\alpha^0|}.$$

For any  $s > t \geq N$ , we have

$$\begin{aligned}
 |r_j^s - r_j^t| &\leq |r_j^s - r_j^{s-1}| + |r_j^{s-1} - r_j^{s-2}| + \dots + |r_j^{t+1} - r_j^t| \\
 &\leq \lambda^{s-1} |r_\alpha^1 - r_\alpha^0| + \lambda^{s-2} |r_\alpha^1 - r_\alpha^0| + \dots + \lambda^t |r_\alpha^1 - r_\alpha^0| \\
 &\leq \lambda^t \sum_{k=0}^{s-t-1} \lambda^k |r_\alpha^1 - r_\alpha^0| \\
 &\leq \lambda^t \cdot \frac{1}{1 - \lambda} |r_\alpha^1 - r_\alpha^0| \\
 &\leq \epsilon
 \end{aligned}$$

Clearly,  $r_j^s$  is a Cauchy sequence, thus it converges to a fixed point.

For uniqueness, we prove it by contradiction. Suppose it has two fixed points  $r^{(1)}$  and  $r^{(2)}$ . Let  $X = |r_\alpha^{(1)} - r_\alpha^{(2)}| = \max_j |r_j^{(1)} - r_j^{(2)}|$ , then

$$\begin{aligned}
 X &= |r_\alpha^{(1)} - r_\alpha^{(2)}| \\
 &= \frac{1}{|M_\alpha|} \left| \sum_{u_i \in M_\alpha} R_{i\alpha} \left[ \sum_{t_k \in T} b_{\alpha k} (c_{ik}^{(1)} - c_{ik}^{(2)}) \right] \right| \\
 &= \frac{1}{|M_\alpha|} \left| \sum_{u_i \in M_\alpha} R_{i\alpha} \left[ \sum_{t_k \in T} b_{\alpha k} \frac{\lambda \sum_{o_\beta \in N_i} b_{\beta k} |R_{i\beta} - r_\beta^{(1)}|}{\sum_{o_\beta \in N_i} b_{\beta k}} \right. \right. \\
 &\quad \left. \left. - \sum_{t_k \in T} b_{\alpha k} \frac{\lambda \sum_{o_\beta \in N_i} b_{\beta k} |R_{i\beta} - r_\beta^{(2)}|}{\sum_{o_\beta \in N_i} b_{\beta k}} \right] \right| \\
 &\leq \frac{1}{|M_\alpha|} \left| \sum_{u_i \in M_\alpha} R_{i\alpha} \left[ \sum_{t_k \in T} b_{\alpha k} \frac{\lambda \sum_{o_\beta \in N_i} b_{\beta k} |r_\beta^{(1)} - r_\beta^{(2)}|}{\sum_{o_\beta \in N_i} b_{\beta k}} \right] \right|
 \end{aligned}$$

$$\begin{aligned}
 &\leq \frac{\lambda}{|M_\alpha|} \left| \sum_{u_i \in M_\alpha} R_{i\alpha} \left[ \sum_{t_k \in T} b_{\alpha k} |r_\alpha^{(1)} - r_\alpha^{(2)}| \right] \right| \\
 &= \frac{\lambda}{|M_\alpha|} \sum_{u_i \in M_\alpha} R_{i\alpha} |r_\alpha^{(1)} - r_\alpha^{(2)}| \\
 &\leq \lambda \cdot |r_\alpha^{(1)} - r_\alpha^{(2)}| \\
 &= \lambda X.
 \end{aligned}$$

Since  $\lambda \in [0, 1)$ , we get  $X < X$ , which is impossible. Thus,  $r_j$  converges to a unique fixed point. □

**Theorem 2** *Let  $|r_\alpha^\infty - r_\alpha^s| = \max_j |r_j^\infty - r_j^s|$ , then we have  $|r_\alpha^\infty - r_\alpha^s| \leq \lambda^s$ . In other words, the rate of convergence for TB-L<sub>1</sub>-AVG is exponential.*

*Proof* We prove it by induction. For  $t = 1$ ,

$$\begin{aligned}
 |r_\alpha^\infty - r_\alpha^1| &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^\infty - c_{ik}^0| \\
 &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \lambda |r_\alpha^\infty - r_\alpha^0| \\
 &\leq \lambda |r_\alpha^\infty - r_\alpha^0| \\
 &\leq \lambda.
 \end{aligned}$$

Assume when  $t = s$  the theorem holds, we prove  $t = s + 1$  it still holds.

$$\begin{aligned}
 |r_\alpha^\infty - r_\alpha^{s+1}| &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^\infty - c_{ik}^s| \\
 &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \lambda |r_\alpha^\infty - r_\alpha^s| \\
 &\leq \lambda |r_\alpha^\infty - r_\alpha^s| \\
 &\leq \lambda \cdot \lambda^s \\
 &= \lambda^{s+1}.
 \end{aligned}$$

This completes the proof. □

### 4 Experiment 1: Estimating topic-biased UR and item scores in user study

The above section introduces TBM and applies one toy example to illustrate the capability of our model in estimating UR on different topics. Now our research focuses on the following three questions:

1. Do the topic-biased algorithms produce accurate estimations of UR in the real data set?
2. Through modeling topic-biased UR, can TBM give more accurate estimations to item scores?
3. Are topic-biased algorithms more robust to unreliable ratings, compared with traditional reputation-based algorithms?

We answer the three questions by conducting intensive experiments employing both real-world and synthetic data. In this section, we present our experiments with a user study. In the next section (Sect. 5), we show our simulations on a real-world data set and several synthetic data sets. In Sect. 4.1, we present the setup for our user study. We make some observations in Sect. 4.2. Sections 4.3 and 4.4 report the UR estimation and item score estimation results from our user study.

### 4.1 Setup

The user study involves a rating task and a survey. In the task, the raters are invited to rate the quality of answers crawled from a community question answering (CQA) portal called Sina iAsk.<sup>4</sup> In Sina iAsk, users can post their own questions and answer other users' questions. In most circumstances, a question receives more than one answer and the asker picks one as the best answer. Besides, the asker can let the community vote for the best answer. We randomly select 200 solved questions from four categories of Sina iAsk (namely, *Computer & Internet, Sports & Hobbies, Education, and Transportation*, 50 questions each) with each question having 3–9 answers. A total of 883 QA pairs are used in the rating task. In addition, we resort the answers of each question randomly to eliminate position bias. We invite 30 raters in the task and ask each one to rate all QA pairs. These participants are all well educated with a Bachelor's or Master's degree or currently studying in their PhD programs. More importantly, their backgrounds cover a wide range of disciplines in natural sciences, social sciences, and engineering. All raters are first briefly informed of the research design. Afterward, they rate all these QA pairs using a 3-point Likert scale (A, B, and C, with A indicating "the answer completely satisfies the information need," B indicating "the answer partially satisfies the information need," and C indicating "the answer is irrelevant" [27]). We give them sufficient and flexible time but do not allow them to visit Sina iAsk to avoid plagiarism. By using our self-designed user interface, all the 30 raters complete the rating task, with a mean duration of 2 h and 38 min. After the ratings, raters are also invited to provide feedback or comments in the survey.

The ground truth of UR is defined as follows. Recall that UR measures the degree to which a user ratings are consistent with other users' on the same items. We follow Mishra et al.'s approach [21] to set the variance of users' ratings as the ground truth of UR. Let  $u_i$  denote the  $i$ th rater,  $o_j$  denote the  $j$ th QA pair,  $R_{ij}$ <sup>5</sup> represent  $u_i$ 's rating to  $o_j$ . We define the variance as follows:

$$UR(u_i) \equiv \frac{1}{\text{Variance}(u_i)} = \frac{|N_i|}{\sum_{o_j \in N_i} |R_{ij} - \hat{r}_j|^2}, \tag{7}$$

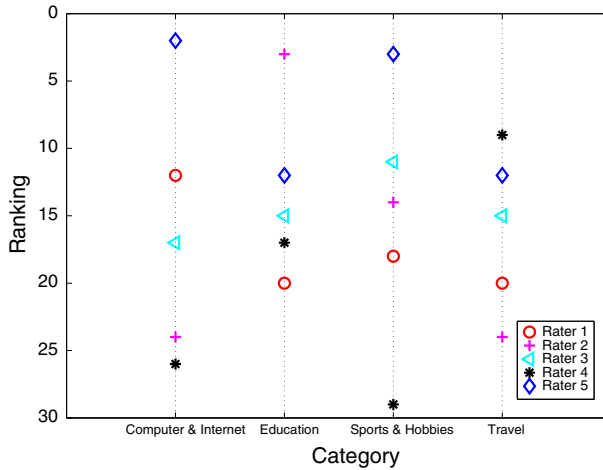
$$\hat{r}_j = \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij}, \tag{8}$$

where  $N_i$  is the set of items that  $u_i$  has rated and  $M_j$  is the set of users who rate  $o_j$ . Let  $b_{jk}$  represent the degree to which that  $o_j$  belongs to topic  $k$ . We set the ground truth of UR on topic  $k$ :

$$UR(u_i, k) \equiv \frac{1}{\text{Variance}(u_i, k)} = \frac{|N_i|}{\sum_{o_j \in N_i} b_{jk} |R_{ij} - \hat{r}_j|^2}. \tag{9}$$

<sup>4</sup> <http://iask.sina.com.cn>.

<sup>5</sup> Without loss of generality, we normalize  $R_{ij}$  to [0,1].



**Fig. 1** Rater ranking on four leaf categories for five selected raters

We compare our topic-biased algorithms with the corresponding reputation algorithms in [18] and the averaging rating (AR) approach. The evaluations are made from two aspects: UR and item scores. We evaluate the performance of these algorithms in estimating UR using accuracy and measure their performance in estimating item scores from both accuracy and robustness. Kendall  $\tau$  coefficient, which represents the similarity between two rankings, is leveraged in our evaluation.

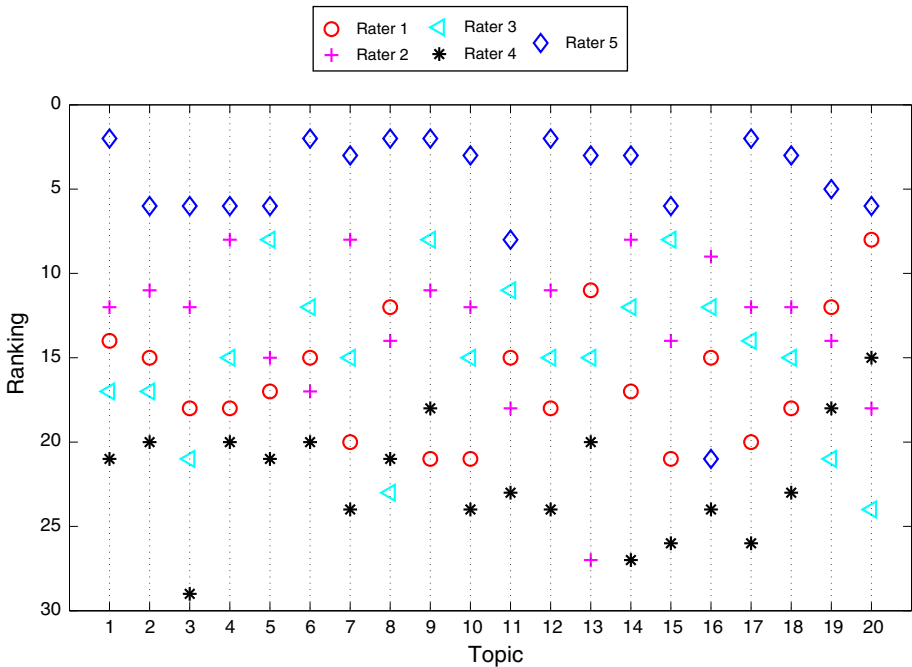
#### 4.2 Observations

We first study the relationships between UR and leaf categories. Since each QA pair only belongs to one particular leaf category,  $b_{jk}$  becomes a binary variable. We apply Eq. 3 to calculate each UR on each “topic” and rank raters in a descending order. If UR is stable across “topics,” the rankings should be stable. Figure 1 reports the rankings of five raters’ UR across four categories, where the x axis represents categories and the y axis stands for rankings. We observe that user rankings fluctuates on different categories. For instance, among these five raters, Rater 1 ranks second on *Computer & Internet*, but ranks fifth on *Education*; Rater 4 ranks first on *Travel*, but last on *Computer & Internet*. One possible explanation for the divergence is that raters lack expertise in some questions/answers of certain topics, as one of the raters commented below: “I am not sure about my ratings of answers to some topics, as I have little knowledge of those topics. So, I just rated using my common sense...”

As the category is manually designed, and may cover various topics, we further study the connections between UR and different topics learned by topic models. We employ the latent Dirichlet allocation (LDA) [2] to automatically learn topics and topic distributions from the content of QA pairs. In our experiments, we use the tool GibbsLDA++ [24] with 20 topics and 1,000 iterations. Table 5 presents the top 5 words of some topics generated from LDA. We can see that, compared with categories, these learned topics are more fine-grained. For instance, topic 2 represents NBA games, and topic 8 is related to computer problems. Given refined topics, we examine user reputation across different topics. Figure 2 reports the rankings of the same five raters’ UR across topics. As shown in Fig. 2, many raters’ rankings fluctuate greatly, like Rater 1, Rater 2, and Rater 3. Taking Rater 3 as an example: among

**Table 5** Selected topic ids and top five words (original Chinese words are translated into English words)

ID	Top five words in the topic
2	Ball, Adversary, Player, The Lakers, Defender
3	Sports, Climbing, Body, Shadowboxing, Function
8	System, Computer, Question, Game, Software
9	Winter, Travel, Compare, Temperature, Climate
10	Japan, University, School, Ability, Japanese
12	Cup, League, Team, Champion, Game
16	Liver, Fat, Blood-enrichment, Food, Eat



**Fig. 2** Rater ranking on 20 topics for five selected raters

these five raters, Rater 3 ranks second on topics 5, 9, and 15, but takes the last position on topics 8, 19, and 20. The above observations once again indicate that UR is topic-biased and varies across topics. Thus, it is reasonable to model UR taking account of topics.

However, previous reputation-based algorithms overgeneralize UR and use one single value to model it, resulting in an inaccurate estimation of UR and item scores. In the following two sections (Sects. 4.3, 4.4), we report the results of our topic-biased models on UR estimation and item scores estimation, respectively.

### 4.3 UR estimation results

In this section, we report the performance of various methods in estimating UR. The ground truth of UR over different topics is provided in Eq. 9. We first compare the accuracy of various algorithms, and then investigate the influence of the parameter  $\lambda$  to topic-biased algorithms.

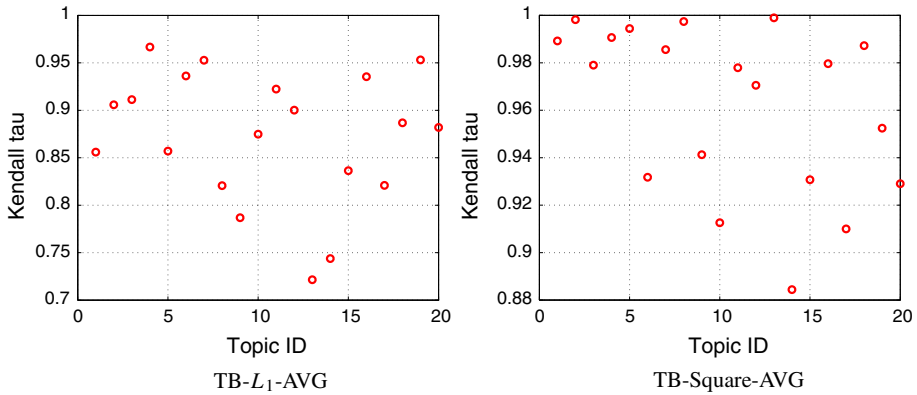


Fig. 3 Kendall  $\tau$  on various topics for TB- $L_1$ -AVG and TB-Square-AVG TB- $L_1$ -AVG. TB-Square-AVG

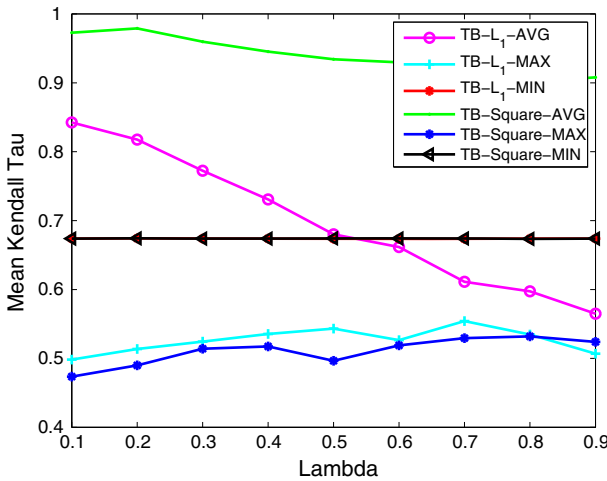


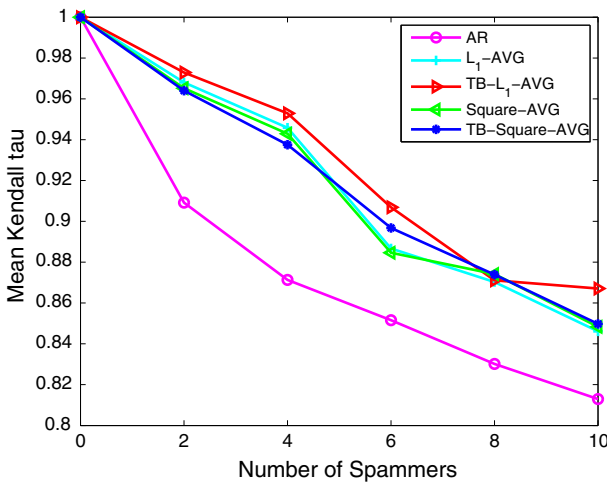
Fig. 4 Mean Kendall  $\tau$  with the ground truth across different  $\lambda$ s

**Accuracy.** Figure 3 reports the Kendall  $\tau$ s of 30 raters’ rankings on various topics generated from TB- $L_1$ -AVG (TB-Square-AVG) and from Eq. 9. In most topics, the rankings of TB- $L_1$ -AVG and TB-Square-AVG are in line with the ground truth ( $\tau > 0.75$ ). For other topic-biased algorithms, the Kendall  $\tau$ s are all above 0.5 on each topic. These results demonstrate the effectiveness of TBM in estimating UR across various topics.

**Effect of  $\lambda$ .** Figure 4 shows the mean Kendall  $\tau$  of 30 raters’ rankings generated from six topic-biased algorithms and from Eq. 9. Across all six algorithms, TB- $L_1$ -AVG and TB-Square-AVG generate the closest rankings to the ground truth. When  $\lambda = 0.1$ , the rankings of TB- $L_1$ -AVG and TB-Square-AVG are mostly consistent with the ground truth. With an increase of  $\lambda$ , Kendall  $\tau$  decreases. The setting of  $\lambda$  has no much impact on the performance TB- $L_1$ -MAX and TB-Square-MAX. The best rankings of TB- $L_1$ -MAX and TB-Square-MAX are achieved when  $\lambda = 0.7$  and  $\lambda = 0.9$ , respectively. TB- $L_1$ -MIN and TB-Square-MIN are not sensitive to the setting of  $\lambda$  since they give the most conservative estimations on UR: only the minimum differences are taken into account.

**Table 6** Pairwise mean Kendall  $\tau$  among AR,  $L_1$ -AVG, and TB- $L_1$ -AVG

$\lambda$	AR versus $L_1$ -AVG	AR versus TB- $L_1$ -AVG	$L_1$ -AVG versus TB- $L_1$ -AVG
0.1	0.9240	0.9418	0.9808
0.2	0.9282	0.9363	0.9841
0.3	0.9309	0.9446	0.9850
0.4	0.9220	0.9383	0.9832
0.5	0.9159	0.9397	0.9839
0.6	0.9154	0.9280	0.9788
0.7	0.9192	0.9260	0.9755
0.8	0.9113	0.9155	0.9779
0.9	0.9036	0.9156	0.9747



**Fig. 5** Robustness to spammers

#### 4.4 Item score estimation results

In this section, we report the performance of various methods in estimating item scores. We rank answers of each question according to their scores generated from AR, traditional reputation-based algorithms, and topic-biased algorithms. For each method, 200 rankings are obtained—one ranking for each question. The best values of  $\lambda$  are set to each algorithm so that the generated rankings of UR are closest to the ground truth derived from Eqs. 8 and 9, respectively.

**Accuracy.** Due to the lack of ground truth, we employ pairwise Kendall  $\tau$  between two rankings to measure the accuracy of item scores estimation. Table 6 shows the pairwise mean Kendall  $\tau$  between average rating (AR),  $L_1$ -AVG, and TB- $L_1$ -AVG. We find that the rankings of TB- $L_1$ -AVG and  $L_1$ -AVG are most consistent. In most cases, the mean Kendall  $\tau$  between AR and TB- $L_1$ -AVG is also higher than the mean Kendall  $\tau$  between  $L_1$ -AVG and AR. Other topic-biased algorithms report similar results. From these results, we know that our proposed algorithms generate mediate rankings between AR and traditional reputation-based algorithms although we cannot say which one produces the most accurate rankings.

**Robustness.** To evaluate the robustness of various methods, we generate spammers who rate randomly from A to C for all QA pairs and calculate the mean Kendall  $\tau$  between the rankings of original data and the rankings of data with spammers. Figure 5 shows the results. We can find that the Kendall  $\tau$  of TB- $L_1$ -AVG and TB-Square-AVG is always much higher than that of AR with the number of spammers varying from 1 to 5. In addition, TB- $L_1$ -AVG and TB-Square-AVG perform slightly better than  $L_1$ -AVG and Square-AVG, respectively. As the topic-biased algorithms consider user reputation across topics, they produce more comprehensive estimations of spammers' UR across topics, with spammers' influences on item scores being reduced.

## 5 Experiment 2: Estimating topic-biased UR and item scores in simulations

We undertake intensive simulations in this section. As we have no ground truth for item scores (i.e., answer quality) in our user study, it is hard to measure TBM's accuracy in estimating item scores. Furthermore, the performance of TBM on large-scale data is unknown. Our simulations in this section address these concerns. In Sect. 5.1, we detail the process of generating simulation data. In Sect. 5.2, we report the performance of different methods in estimating UR and item scores. Finally, we discuss the algorithm selection and parameter setting in Sect. 5.3.

### 5.1 Setup

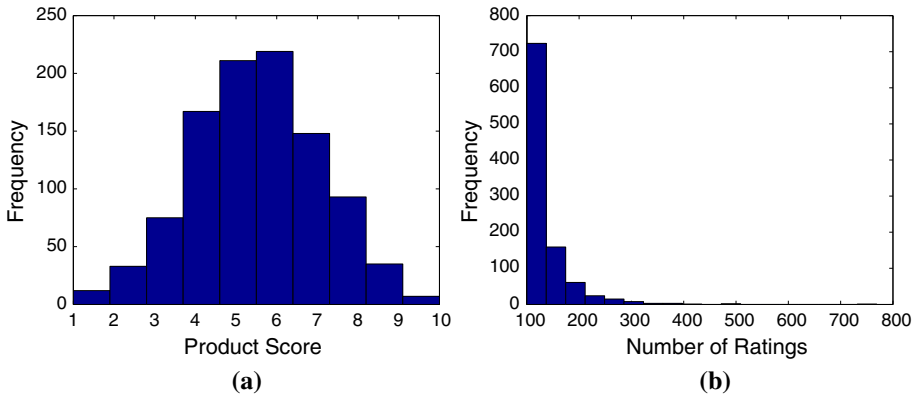
The synthetic data sets generated are ratings for products, each one encompassing 1,000 items, 10,000 users, and 20 categories (topics). The procedures are as follows:

1. We generate true scores of products from a normal distribution with a mean of 5.5 and a variance of 3. This setting ensures more than 99% of random numbers ranging from 1 to 10. Each product's score is then converted into the nearest integer numbers [1–10]. The distribution of product scores is shown in Fig. 6a.
2. We generate UR vector (**URV**) for each user. **URV** has a length of 20, with the value of each element reflecting UR on the corresponding topic of products. We simulate three common types of users with each element allocated from one of the three values:
  - 0: Normal users (NU). It means the user's ratings are usually close to true scores.
  - 1: Category-biased users (BU). It indicates the user usually gives poor-quality ratings on some categories' products but provides good ratings on other products.
  - 2: Spammers (SP). It signals the user is a spammer and rates randomly.

Initially, all **URV**'s elements are set as 0. Consequently, we randomly select  $S$  spammers and fill all the elements of their **URV**s with 2. As for the remaining users, we pick  $B$  biased users randomly for each topic and fill the values of corresponding elements with 1. We repeat the operation 20 times and the biased users on each topic are determined.

3. For each item, we draw a random number of raters from a power law distribution [6] with parameter  $\alpha = 2.5$  and the minimum number is set as 100 (As reported in [6], most networks fit power law distributions with  $\alpha$  between 2 and 3). Figure 6b shows the distribution of raters' numbers (ratings).
4. We generate the topic distribution vector **TDV** for every item, with each element representing the degree of which that the item belongs to that topic. For simplicity, each item is grouped into one of the 20 topics, with each topic comprising 50 items (i.e., Topic 1:





**Fig. 6** Distributions of generated product scores (a) and number of ratings (b)

No.1 to No. 50 and so forth). We set the values of corresponding elements in **TDVs** as 1 and keep the remaining as 0.

5. For each item, we randomly select the number of raters generated from step 3 and simulate their ratings as follows:
  - If a user is a spammer (all elements in the **URV** are 2), we follow the work in [4, 19] and generate a random integer score [1–10] from a uniform distribution, which simulates *random attacks*.
  - Otherwise, we compute the inner product of the user’s **URV** and **TDV**. If the result is higher than 0, we set the rating as 10. Otherwise, we set the rating as a random number drawn from the normal distribution which has a mean of the item’s true score and a square deviation of 1.

We generate 9 data sets with various  $S$ s and  $B$ s ranging from 0 to 5,000. When  $S = 0$  and  $B = 0$ , all users are normal users; when  $S = 0$  and  $B \neq 0$ , the data sets contain normal users and topic-biased users; when  $S \neq 0$  and  $B = 0$ , the data sets contain normal users and spammers; when  $S \neq 0$  and  $B \neq 0$ , the data sets contain all the above three types of users. We compare the performance of algorithms listed in Table 3. Each algorithm runs 20 iterations with different settings of  $\lambda$  ranging from 0.1 to 0.9. In addition, we employ the averaged rating (AR) as our baseline method for estimating item scores. It is worth noting that AR lacks an estimation of UR.

The evaluations are also made from two aspects: UR and item scores. We employ Z test to verify the effectiveness of the estimated UR. Specifically, we first obtain three groups of UR in terms of normal raters, biased raters, and spammers. Then, we conduct Z tests to examine whether the estimated UR values of these groups are significantly different or not. For item scores, we employ mean average error (MAE) and root mean square error (RMSE) to measure each algorithm’s accuracy and robustness.

## 5.2 Results

### 5.2.1 UR estimation results

We present the performance of different methods in estimating UR in this section. Recall that the proposed TBM has the advantage of estimating UR across topics. For verification, we collect the estimated UR for normal users, biased users, and spammers from each topic, and

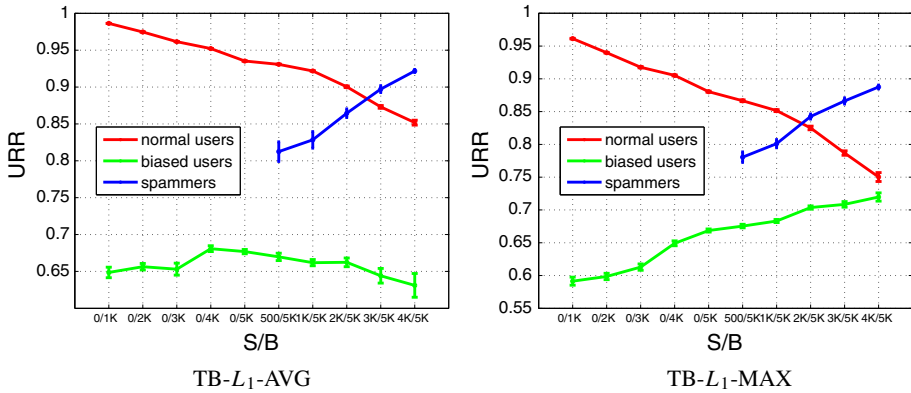


Fig. 7 Error bars of UR estimated from TB-L<sub>1</sub>-AVG and TB-L<sub>1</sub>-MAX TB-L<sub>1</sub>-AVG. TB-L<sub>1</sub>-MAX

calculate the means and standard deviations. Figure 7 shows error bars of estimated UR of these users using TB-L<sub>1</sub>-AVG and TB-L<sub>1</sub>-MAX. We can see that the estimated UR values of these three types of users are distinct from each other. When there is no spammers, the UR values of normal users are much higher than those of biased users. With the increase of number of spammers, the UR of normal users decreases, while the UR of spammers and biased users increases. When the number of spammer users is large enough, their UR is even higher than that of normal users. Since spammers become the majority, normal users' ratings are not treated as reliable ones any more. It is worth noting that the estimated UR of spammers is higher than that of biased users. The reason is that in our setting, the biased users constantly give higher ratings. Although the spammers give random ratings, the estimated UR of biased users are still lower than that of spammers. We further conduct Z test for each pair (i.e., NU vs. SP, NU vs. BU, NU vs. (SP+BU), BU vs. SP), with all 150 (25×6) tests producing significant differences ( $p_s < 0.005$ ). These results demonstrate that our topic-biased algorithms are able to effectively estimate UR across topics and distinguish unreliable ratings from biased users or spammers, and reliable ratings from normal users.

### 5.2.2 Item score estimation results

Tables 7 and 8 report the MAE and RMSE of various algorithms across different number of biased users and spammers (the best results are in bold). For reputation-based algorithms and topic-biased algorithms, the best settings of  $\lambda$  are adopted.

**Accuracy.** The MAE and RMSE of topic-biased algorithms are lower than those of AR and traditional reputation-based algorithms in almost all cases. Among all topic-biased algorithms, TB-L<sub>1</sub>-MAX performs the best. This is due to the simulation of biased users: Here, all biased users give the highest ratings. Thus, the  $max()$  captures UR best. Among all settings, TBM algorithms perform the best except when  $S = 0$  and  $B = 1,000$ . As there are only a few biased users in this setting, some traditional algorithms like L<sub>1</sub>-MAX and Square-MAX are able to give accurate estimations of UR. With the increasing number of unreliable ratings, however, our TBM algorithms outperform traditional methods. For instance, when  $B = 5,000$ , the MAE of TB-L<sub>1</sub>-AVG is 118 % lower than that of AR and 27 % lower than that of L<sub>1</sub>-AVG; the MAE of TB-L<sub>1</sub>-MAX is 134 % lower than that of AR and 45 % lower than that of L<sub>1</sub>-MAX. Similar results are observed for other reputation-based and topic-biased algorithms. The RMSEs of TBM algorithms in Table 8 show similar results. The above

**Table 7** Different algorithms' MAEs (the best results are shown in bold)

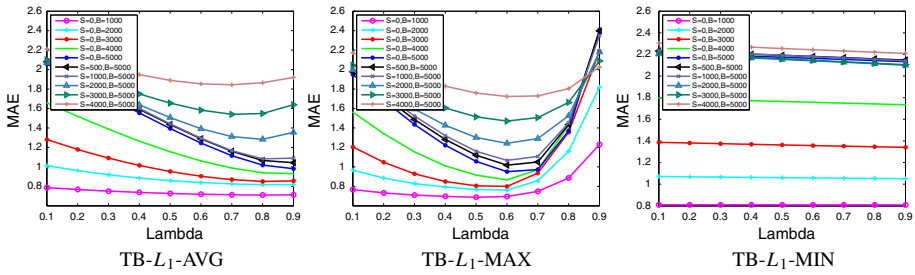
# of Spammers	0	0	0	0	0	500	1,000	2,000	3,000	4,000
# of Topic-biased users	1,000	2,000	3,000	4,000	5,000	5,000	5,000	5,000	5,000	5,000
AR	0.809	1.075	1.393	1.804	2.222	2.241	2.25	2.217	2.227	2.317
$L_1$ -AVG	0.717	0.833	0.917	1.099	1.296	1.334	1.342	1.408	1.746	1.919
TB- $L_1$ -AVG	0.711	0.818	0.85	0.939	1.018	1.066	1.082	1.283	1.549	1.865
$L_1$ -MAX	<b>0.685</b>	0.772	0.863	1.079	1.383	1.439	1.484	1.548	1.659	1.827
TB- $L_1$ -MAX	0.696	<b>0.762</b>	<b>0.800</b>	<b>0.866</b>	<b>0.951</b>	<b>1.018</b>	<b>1.067</b>	<b>1.243</b>	<b>1.471</b>	<b>1.723</b>
$L_1$ -MIN	0.808	1.067	1.374	1.776	2.186	2.203	2.207	2.169	2.176	2.262
TB- $L_1$ -MIN	0.809	1.051	1.341	1.735	2.133	2.148	2.141	2.1	2.105	2.21
Sq-AVG	0.797	1.038	1.326	1.718	2.127	2.149	2.158	2.13	2.149	2.251
TB-Sq-AVG	0.787	0.999	1.246	1.617	2.018	2.044	2.055	2.036	2.071	2.197
Sq-MAX	0.742	0.926	1.155	1.531	1.952	1.981	1.999	1.991	2.029	2.152
TB-Sq-MAX	0.745	0.899	1.053	1.358	1.734	1.779	1.813	1.853	1.946	2.111
Sq-MIN	0.809	1.074	1.392	1.802	2.22	2.239	2.247	2.213	2.223	2.311
TB-Sq-MIN	0.81	1.073	1.389	1.799	2.217	2.236	2.243	2.208	2.215	2.305

**Table 8** Different algorithms' RMSEs (the best results are shown in bold)

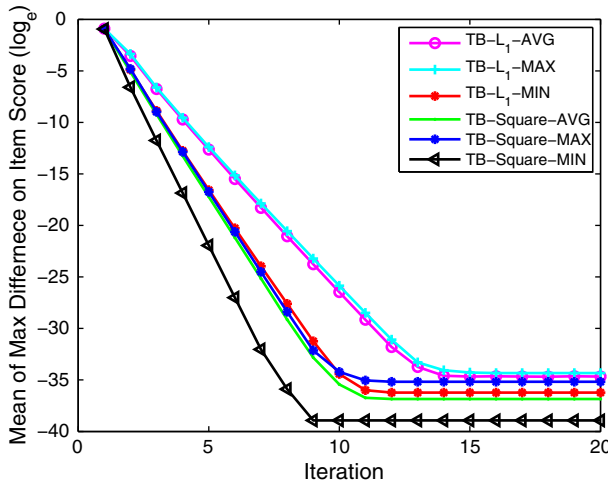
# of Spammers	0	0	0	0	0	500	1,000	2,000	3,000	4,000
# of Topic-biased users	1,000	2,000	3,000	4,000	5,000	5,000	5,000	5,000	5,000	5,000
AR	1.002	1.294	1.617	2.025	2.447	2.494	2.529	2.567	2.644	2.786
$L_1$ -AVG	0.929	1.025	1.146	1.352	1.577	1.63	1.65	1.755	1.927	2.206
TB- $L_1$ -AVG	0.943	1.006	1.071	1.171	1.272	1.333	1.356	1.594	1.909	2.297
$L_1$ -MAX	<b>0.887</b>	0.954	1.08	1.329	1.666	1.739	1.801	1.907	2.049	2.259
TB- $L_1$ -MAX	0.907	<b>0.944</b>	<b>0.999</b>	<b>1.081</b>	<b>1.182</b>	<b>1.269</b>	<b>1.333</b>	<b>1.546</b>	<b>1.818</b>	<b>2.142</b>
$L_1$ -MIN	1.001	1.287	1.601	2.002	2.418	2.461	2.491	2.523	2.596	2.733
TB- $L_1$ -MIN	1.003	1.272	1.574	1.969	2.375	2.413	2.433	2.462	2.532	2.683
Sq-AVG	0.992	1.256	1.556	1.95	2.364	2.412	2.447	2.488	2.571	2.723
TB-Sq-AVG	0.987	1.216	1.486	1.861	2.269	2.32	2.354	2.405	2.5	2.675
Sq-MAX	0.946	1.134	1.393	1.777	2.206	2.259	2.301	2.354	2.449	2.617
TB-Sq-MAX	0.958	1.103	1.294	1.616	2.007	2.075	2.128	2.228	2.372	2.584
Sq-MIN	1.002	1.294	1.617	2.023	2.445	2.492	2.527	2.564	2.64	2.781
TB-Sq-MIN	1.002	1.293	1.614	2.021	2.443	2.489	2.523	2.559	2.634	2.776

results demonstrate the effectiveness of TBM: Estimating topic-biased UR and allocating lower weighting to unreliable ratings provide a more accurate estimation of item scores relative to previous algorithms.

**Robustness.** An increase in  $B$  and  $S$  leads to a growing number of biased users and spammers, leading to an increase in MAE and RMSE of the three algorithms. In other words, all algorithm's robustness decreases with the increase in proportions of unreliable



**Fig. 8**  $\lambda$ 's influence on MAE of TB- $L_1$ -AVG, TB- $L_1$ -MAX, and TB- $L_1$ -MIN TB- $L_1$ -AVG. TB- $L_1$ -MAX. TB- $L_1$ -MIN

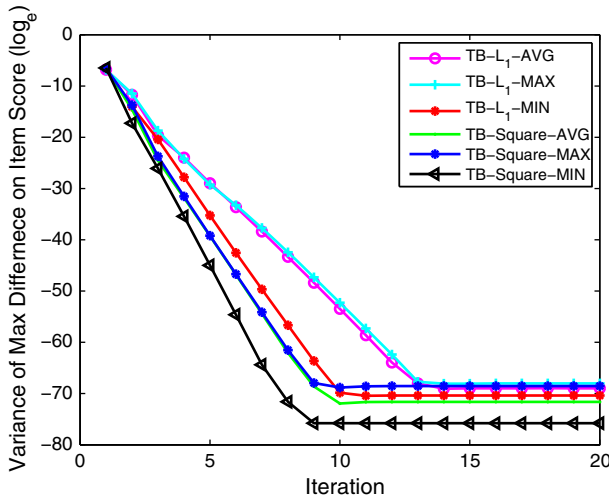


**Fig. 9** Convergence rate of TS algorithms ( $S = 4,000$  and  $B = 5,000$ ): mean of max difference on item scores versus number of iteration

ratings. However, compared to AR and traditional reputation-based algorithms, topic-biased algorithms are more robust to unreliable ratings due to the slowest growth in MAE and RMSE. For instance, when  $S = 0$ , from  $B = 1,000$  to  $B = 5,000$ , the MAEs of AR and  $L_1$ -AVG increase by 175% and by 81%, respectively, while the MAE of TB- $L_1$ -AVG only increases by 43%.

**Effect of  $\lambda$ .** We attempt different settings of  $\lambda$  (namely 0.1, 0.2, . . . , 0.9) to investigate its influence on the performance of topic-biased algorithms. Figure 8 shows the MAE of TB- $L_1$ -AVG, TB- $L_1$ -MAX, and TB- $L_1$ -MIN under different settings of  $\lambda$ . The best  $\lambda$  is 0.8 for TB- $L_1$ -AVG, and the best  $\lambda$  is around 0.6 for TB- $L_1$ -MAX. Like the results of CQA data, TB- $L_1$ -MIN is less sensitive to the parameter setting.

**Rate of convergence.** Figures 9 and 10 show the mean and variance of maximum difference of item scores between two consecutive iterations for different topic-biased algorithms when  $B = 5,000$  and  $S = 4,000$ . We find that both the difference and the variance decreases rapidly and TBM converges within 15 iterations. These results tally with their good properties of convergence, which enables these topic-biased algorithms to deal with large-scale data.



**Fig. 10** Convergence rate of TS algorithms ( $S = 4,000$  and  $B = 5,000$ ): variance of max difference on item scores versus number of iteration

### 5.3 Discussion

#### 5.3.1 Algorithm selection

From two experiments, we find that topic-biased algorithms usually perform better than corresponding reputation-based algorithms. However, among six topic-biased algorithms, there is no one algorithm which consistently outperforms others. The possible reason is that in different data sets rating behavior and degree of topic bias varies, which leads to inconsistent performance of each algorithm. AVG-based algorithms capture the mean deviation as the measurement of UR, while MIN-based and MAX-based algorithms use the minimum and maximum deviations. In some cases where the differences between user ratings and true scores within the same topic do not change too much, AVG-based algorithms obtain better performance, such as our user study. In some other cases where the differences between user ratings and true scores within the same topic vary greatly, MAX-based algorithms may perform better, like our simulations. These suggest that, in order to get the best UR and item score estimation, we need to analyze data characteristics and choose the most suitable algorithm.

#### 5.3.2 Parameter setting

In our models,  $\lambda$  is a damping factor which controls the speed of convergence. According to Theorem 2, smaller value of  $\lambda$  leads to faster convergence. In addition, we investigate the influence of  $\lambda$  to different models' performance in our experiments. Experimental results show that AVG-based and MAX-based algorithms are more sensitive to the setting of  $\lambda$ , while MIN-based algorithms are insensitive. The reason is that MIN-based algorithms are the most lenient in UR estimation; thus, different parameter settings do not change results too much. For AVG-based and MAX-based algorithms, there is no optimal  $\lambda$  for all cases. In other words, the best setting of the damping factor is depended on the data rather than the

algorithm. Therefore, in practice a validation set is required to get a good parameter setting for each particular task.

## 6 Conclusion

In rating systems, UR plays an essential role in estimating item scores. For this purpose, a number of studies have therefore employed UR in estimating item scores. However, traditional reputation-based approaches neglect the role of topics, resulting in biased ratings. To this end, we first show that UR fluctuates across different topics from the Epinions data set. Based on this, we thereby develop a topic-biased model to estimate UR and item scores simultaneously. Accordingly, we develop six topic-biased algorithms. After analyzing the convergence property mathematically, we subsequently conduct a two-step validation of the topic-biased algorithms. First, we present the effectiveness and robustness of topic-biased algorithms with the data collected from our user study. Second, we conduct large-scale simulations to further examine their performance. Results of the experiments indicate that these topic-biased algorithms are effective in estimating UR on various topics and item scores, and are more robust to unreliable ratings (including biased ratings and spam ratings), in relation to the existing reputation-based algorithms.

Our proposed algorithms, with good convergence property, can contribute to better user reputation estimation and item score estimation. In addition, the proposed algorithms are applicable to most objective rating systems, like product review system and academic paper review system. What's more, it is feasible to incorporate topic modeling methods in the topic-biased model.

Currently, we apply our topic-biased algorithms to objective ratings like product reviews. For subjective ratings like movie ratings, in which user preference and taste are incorporated in ratings, these algorithms may not be applicable for making recommendations. However, they can still be employed to identify similar user profiles, which is helpful for content-based filtering or collaborative filtering. In future studies, we aim to investigate better modeling of UR since the influence of the damping factor (i.e.,  $\lambda$ ) to current models is data dependent. In addition, we plan to collect more data using crowdsourcing systems (e.g., Amazon's Mechanical Turk) to test the performance of our algorithms with different data sets.

**Acknowledgments** The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (No. 2014CB340405), the Shenzhen Basic Research Program (No. JCYJ20120619152636275), the Research Grants Council of the Hong Kong Special Administrative Region, China (Nos. CUHK 413212 and CUHK 415212), and the Microsoft Research Asia Grant in Big Data Research (No. FY13-RES-SPONSOR-036).

## 7 Appendix

In this appendix, we give the proof that Lemma 1 holds for TB- $L_1$ -MAX and TB- $L_1$ -MIN. Theorem 1 and 2 for them and other topic-biased algorithms are similar to the proof of TB- $L_1$ -AVG.

7.1 TB- $L_1$ -MAX

*Proof* If  $s = 1$ , let

$$o_h = \begin{cases} \arg \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^1|, & \text{if } c_{ik}^1 > c_{ik}^0 \\ \arg \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^0|, & \text{otherwise,} \end{cases}$$

then

$$\begin{aligned} |r_j^2 - r_j^1| &= \left| \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} c_{ik}^1 b_{jk} - \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} c_{ik}^0 b_{jk} \right| \\ &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^1 - c_{ik}^0| \\ &= \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |\lambda \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^1| - \lambda \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^0| | \\ &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |\lambda b_{hk} |r_h^1 - r_h^0| \\ &\leq \frac{\lambda}{|M_j|} \sum_{u_i \in M_j} R_{ij} |r_\alpha^1 - r_\alpha^0| \\ &\leq \lambda \cdot |r_\alpha^1 - r_\alpha^0|. \end{aligned}$$

Assume when  $s = t$  the lemma holds, then we show  $s = t + 1$  the lemma still holds. Let

$$o_x = \begin{cases} \arg \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^t|, & \text{if } c_{ik}^t > c_{ik}^{t-1} \\ \arg \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^{t-1}|, & \text{otherwise,} \end{cases}$$

then

$$\begin{aligned} |r_j^{t+1} - r_j^t| &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^t - c_{ik}^{t-1}| \\ &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot |\lambda \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^t| - \lambda \max_{o_j \in N_i} b_{jk} |R_{ij} - r_j^{t-1}| | \\ &\leq \frac{\lambda}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot b_{xk} |r_x^t - r_x^{t-1}| \\ &\leq \frac{\lambda}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot \lambda^t |r_\alpha^1 - r_\alpha^0| \\ &\leq \lambda^t |r_\alpha^1 - r_\alpha^0|. \end{aligned}$$

This completes the proof. □

7.2 TB- $L_1$ -MIN

*Proof* If  $s = 1$ , let

$$o_h = \begin{cases} \arg \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^0|, & \text{if } c_{ik}^1 > c_{ik}^0 \\ \arg \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^1|, & \text{otherwise,} \end{cases}$$

then

$$\begin{aligned} |r_j^2 - r_j^1| &= \left| \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} c_{ik}^1 b_{jk} - \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} c_{ik}^0 b_{jk} \right| \\ &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^1 - c_{ik}^0| \\ &= \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |\lambda \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^1| - \lambda \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^0| | \\ &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |\lambda b_{hk} |r_h^1 - r_h^0| \\ &\leq \frac{\lambda}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |r_\alpha^1 - r_\alpha^0| \\ &\leq \lambda \cdot |r_\alpha^1 - r_\alpha^0|. \end{aligned}$$

Assume when  $s = t$  the lemma holds, then we show  $s = t + 1$  the lemma still holds. Let

$$o_x = \begin{cases} \arg \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^{t-1}|, & \text{if } c_{ik}^t > c_{ik}^{t-1} \\ \arg \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^t|, & \text{otherwise,} \end{cases}$$

then

$$\begin{aligned} |r_j^{t+1} - r_j^t| &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} |c_{ik}^t - c_{ik}^{t-1}| \\ &\leq \frac{1}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot |\lambda \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^t| - \lambda \min_{o_j \in N_i} b_{jk} |R_{ij} - r_j^{t-1}| | \\ &\leq \frac{\lambda}{|M_j|} \sum_{u_i \in M_j} R_{ij} \sum_{t_k \in T} b_{jk} \cdot b_{xk} |r_x^t - r_x^{t-1}| \\ &\leq \frac{\lambda}{|M_j|} \sum_{u_i \in M_j} R_{ij} \lambda^{t-1} |r_\alpha^1 - r_\alpha^0| \\ &\leq \lambda^t |r_\alpha^1 - r_\alpha^0|. \end{aligned}$$

This completes the proof. □



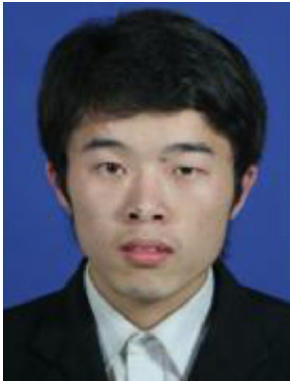
## References

1. Adler BT, de Alfaro L (2007) A content-driven reputation system for the wikipedia. In: Proceedings of the 16th international conference on World Wide Web, WWW '07. ACM, New York, NY, USA, pp 261–270
2. Blei DM et al (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022
3. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the seventh international conference on World Wide Web, pp 107–117
4. Burke R et al (2006) Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06, pp 542–547
5. Cao X et al (2010) A generalized framework of exploring category information for question retrieval in community question archives. In: Proceedings of the 19th international conference on World Wide Web, WWW '10, pp 201–210
6. Clauset A et al (2009) Power-law distributions in empirical data. *SIAM Rev* 51:661–703
7. De Alfaro L et al (2011) Reputation systems for open collaboration. *Commun ACM* 54(8):81–87
8. de Kerchove C, Van Dooren P (2010) Iterative filtering in reputation systems. *SIAM J Matrix Anal Appl* 31:1812–1834
9. Deng H et al (2009) Enhancing expertise retrieval using community-aware strategies. In: Proceedings of the 18th ACM conference on information and knowledge management, CIKM '09, pp 1733–1736
10. Du TC et al (2009) Managing knowledge on the Web—extracting ontology from HTML Web. *Decis support Syst* 47:319–331
11. Gupta M et al (2003) A reputation system for peer-to-peer networks. In: Proceedings of the 13th international workshop on network and operating systems support for digital audio and video, pp 144–152
12. Haveliwala TH (2002) Topic-sensitive PageRank. In: Proceedings of the 11th international conference on World Wide Web, WWW '02, pp 517–526
13. Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '99, pp 50–57
14. Jin Y et al (2010) Topic-sensitive tag ranking. In: The 20th international conference on pattern recognition, pp 629–632
15. Jøsang A et al (2007) A survey of trust and reputation systems for online service provision. *Decis Support Syst* 43(2):618–644
16. Laureti P et al (2006) Information filtering via iterative refinement. *Europhys Lett* 75:1006
17. Li B et al (2011) Question routing in community question answering: putting category in its place. In: Proceedings of the 20th ACM international conference on information and knowledge management, CIKM '11, pp 2041–2044
18. Li R-H et al (2012) Robust reputation-based ranking on bipartite rating networks. In: Proceedings of the 2012 SIAM international conference on data mining, pp 612–623
19. Ling G et al (2013) A unified framework for reputation estimation in online rating systems. In: Proceedings of the twenty-third international joint conference on artificial intelligence, IJCAI'13, pp 2670–2676
20. Medo M, Wakeling JR (2010) The effect of discrete vs. continuous-valued ratings on reputation and ranking systems. *Europhys Lett* 91(4):6
21. Mishra A, Bhattacharya A (2011) Finding the bias and prestige of nodes in networks based on trust scores. In: Proceedings of the 20th international conference on World Wide Web, WWW '11, pp 567–576
22. Mizzaro S (2003) Quality control in scholarly publishing: a new proposal. *J Am Soc Inf Sci Technol* 54(11):989–1005
23. Noy NF, McGuinness DL (2001) Ontology development 101: a guide to creating your first ontology. Tech. rep
24. Phan X-H, Nguyen C-T (2007) GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA). Tech. rep
25. Resnick P et al (2000) Reputation systems. *Commun ACM* 43(12):45–48
26. Resnick P, Zeckhauser R (2002) Trust among strangers in Internet transactions: empirical analysis of eBay's reputation system. *Adv Appl Microecon* 11:127–157
27. Sakai T et al (2011) Using graded-relevance metrics for evaluating community QA answer selection. In: Proceedings of the fourth ACM international conference on Web Search and Data Mining (WSDM '11), pp 187–196
28. Tang J et al (2012) eTrust: understanding trust evolution in an online world. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '12, pp 253–261
29. Vydiswaran VV et al (2011) Content-driven trust propagation framework. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '11, pp 974–982

30. Yu Y-K et al (2006) Decoding information from noisy, redundant, and intentionally distorted sources. *Physica A Stat Mech Its Appl* 371(2):732–744
31. Zhou Y-B et al (2011) A robust ranking algorithm to spamming. *Europhys Lett* 94(4):48002



**Baichuan Li** received a B.E. degree from Shanghai Jiao Tong University, China, in 2009 and a Ph.D. degree from The Chinese University of Hong Kong in 2014. He is currently a senior research engineer at Baidu. His main research interests are in data mining and natural language processing (NLP). Incorporating NLP and machine learning techniques into data mining, Dr. Li constructed a computational framework for analyzing, routing, and structuring user-generated content (UGC) in social networks between 2009 and 2013.



**Rong-Hua Li** received the Ph.D. degree from the Chinese University of Hong Kong in 2013. He is currently an Assistant Professor at Shenzhen University, China. His research interests include algorithmic aspects of social network analysis, big graph data management, and mining.



**Irwin King** research interests include machine learning, social computing, web intelligence, data mining, and multimedia information processing. In these research areas, he has over 200 technical publications in journals and conferences. In addition, he has contributed over 30 book chapters and edited volumes. Prof. King is the Book Series Editor for Social Media and Social Computing with Taylor and Francis (CRC Press). He is also an Associate Editor of the *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)* and *Journal of Neural Networks*. He is a member of the Board of Governors and Vice President for both INNS and APNNA. Prof. King is Associate Dean (Education), Faculty of Engineering and Professor at the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He received his B.Sc. from California Institute of Technology, Pasadena, and his M.Sc. and Ph.D. degree in Computer Science from the University of Southern California, Los Angeles.



**Michael R. Lyu** received the B.S. (1981) in electrical engineering from National Taiwan University, the M.S. (1985) in computer engineering from University of California, Santa Barbara, and the Ph.D. (1988) in computer science from University of California, Los Angeles. He is a Professor in the Computer Science and Engineering Department of the Chinese University of Hong Kong. His research interests include software reliability engineering, software fault tolerance, distributed systems, data mining, social networks, machine learning, multimedia information retrieval, and mobile networks, where he has published over 450 papers. He is an Associate Editor of *IEEE Transactions on Reliability*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Services Computing*, *IEEE Access*, and *Wiley Software Testing, Verification & Reliability Journal*. Dr. Lyu is an IEEE Fellow, an AAAS Fellow, and a Croucher Senior Fellow. He received IEEE Reliability Society 2010 Engineer of the Year Award.



**Jeffrey Xu Yu** received the B.E., M.E., and the Ph.D. degrees in Computer Science, from the University of Tsukuba, Japan, in 1985, 1987, and 1990, respectively. He is currently a professor at the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong. His major research interests include graph mining, graph database, keyword search, social network, and query processing and optimization. He is a senior member of the IEEE, a member of the IEEE Computer Society, and a member of ACM.