Quality and Leniency in Online Collaborative Rating Systems

HADY W. LAUW, Nanyang Technological University
EE-PENG LIM, Singapore Management University
KE WANG, Simon Fraser University

The emerging trend of social information processing has resulted in Web users' increased reliance on user-generated content contributed by others for information searching and decision making. Rating scores, a form of user-generated content contributed by reviewers in online rating systems, allow users to leverage others' opinions in the evaluation of objects. In this article, we focus on the problem of summarizing the rating scores given to an object into an overall score that reflects the object's quality. We observe that the existing approaches for summarizing scores largely ignores the effect of reviewers exercising different standards in assigning scores. Instead of treating all reviewers as equals, our approach models the leniency of reviewers, which refers to the tendency of a reviewer to assign higher scores than other coreviewers. Our approach is underlined by two insights: (1) The leniency of a reviewer depends not only on how the reviewer rates objects, but also on how other reviewers rate those objects and (2) The leniency of a reviewer and the quality of rated objects are mutually dependent. We develop the *leniency-aware quality*, or *LQ* model, which solves leniency and quality simultaneously. We introduce both an exact and a ranked solution to the model. Experiments on real-life and synthetic datasets show that *LQ* is more effective than comparable approaches. *LQ* is also shown to perform consistently better under different parameter settings.

Categories and Subject Descriptors: H.4 [**Information Systems Applications**]; J.4 [**Social and Behavioral Sciences**]:

General Terms: Algorithms, Experimentation, Human Factors

Additional Key Words and Phrases: Quality, leniency, rating, link analysis, social network mining

ACM Reference Format:
Lauw, H. W., Lim, E.-P., and Wang, K. 2012. Quality and leniency in online collaborative rating systems. ACM Trans. Web 6, 1, Article 4 (March 2012), 27 pages.
DOI = 10.1145/2109205.2109209 http://doi.acm.org/10.1145/2109205.2109209

1. INTRODUCTION

Web 2.0 sees the emergence of a more interactive Web. Users are no longer just perusing content, but are also contributing content through their interactions on social media sites, such as blogs, wikis, content sharing (Flickr,[1] YouTube,[2]) social bookmarking

[1]http://www.flickr.com
[2]http://www.youtube.com

H. W. Lauw is currently affiliated with the Institute for Infocomm Research.
Authors' addresses: H. W. Lauw, Institute for Infocomm Research, 1 Fusionopolis Way #21-01 Connexis (South Tower), Singapore 138632; email: hwlauw@i2r.a-star.edu.sg; E.-P. Lim, School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902; email: eplim@smu.edu.sg; K. Wang, Department of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, Canada V5A 1S6; email: wangk@cs.sfu.ca.

DOI 10.1145/2109205.2109209 http://doi.acm.org/10.1145/2109205.2109209

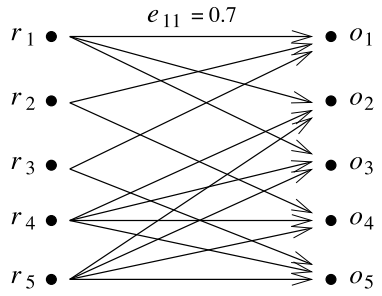


Fig. 1. Bipartite rating network.

(Del.icio.us[3]), product reviews (Epinions[4]), recommender systems (GroupLens[5]), etc. This active contribution and usage of user-generated content has motivated the study of *social information processing*, whereby one exploits the knowledge and opinions generated by users ("wisdom of the crowd") of Web 2.0 applications, for information searching and decision making.

In this article, we focus on online collaborative rating systems, whereby a user may act as a *reviewer*, by contributing opinions in the form of rating scores to objects (products, content items, etc.), as well as a *consumer*, by using the rating scores to compare objects and select the best ones. Such rating systems can be found in many social media sites as just mentioned. Thus, collaborative rating allows users (as consumers) to leverage the knowledge and opinions of others (reviewers) in the evaluation of objects.

It is also worth noting that our target is the objective rating space [Traupman and Wilensky 2004a], in which a rating score is primarily used to interpret the inherent quality of an object, and our main interest is in the quality of objects. As opposed to the subjective rating space, in which a rating score is primarily used to interpret the preference of a reviewer, such as in recommender and collaborative-filtering systems [Lemire 2005; Shen et al. 2006], in which the main interest is modeling user preferences.

Besides online collaborative rating systems, rating objects is also a vital component in many applications, including conference review [Dumais and Nielsen 1992; Geller and Scherl 1997], grant proposal selection [Hettich and Pazzani 2006], etc. In each case, it is important to ensure that the rating has been conducted as fairly and objectively as possible. Unfair ratings may result in adverse outcomes. For instance, it was reported in TIME magazine, on February 16, 2002, that a French referee at the 2002 Winter Olympics figure-skating event confessed to being pressured into voting for the Russian team for the gold medal award. Later, the Canadian team was awarded a second Gold Medal in an attempt to correct the unfair rating. The incident caused a major controversy in the Olympic community and tainted the reputation for fairness of this sporting event.

1.1. Problem

We represent a rating system as a bipartite network with reviewers and objects as the two distinct types of nodes, as shown in Figure 1. A reviewer r_i may assign a rating score $e_{ij} \in [0, 1]$ to an object o_j, which is represented as an edge from r_i to o_j, weighted by e_{ij}. In this article, we focus on the *score summarization problem*, which concerns

[3]http://del.icio.us
[4]http://www.epinions.com
[5]http://www.grouplens.org

$$\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{array} \begin{array}{c} \begin{array}{ccccc} o_1 & o_2 & o_3 & o_4 & o_5 \end{array} \\ \begin{pmatrix} 0.7 & 0.7 & 0.7 & - & - \\ 0.4 & - & - & 0.5 & - \\ 0.4 & - & - & - & 0.5 \\ - & 0.4 & 0.4 & 0.5 & 0.5 \\ - & 0.4 & 0.4 & 0.5 & 0.5 \end{pmatrix} \end{array}$$

(a) Rating data 1

$$\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{array} \begin{array}{c} \begin{array}{ccccc} o_1 & o_2 & o_3 & o_4 & o_5 \end{array} \\ \begin{pmatrix} 0.7 & 0.5 & 0.5 & - & - \\ 0.4 & - & - & 0.3 & - \\ 0.4 & - & - & - & 0.3 \\ - & 0.5 & 0.5 & 0.6 & 0.6 \\ - & 0.5 & 0.5 & 0.6 & 0.6 \end{pmatrix} \end{array}$$

(b) Rating data 2

Fig. 2. Rating data examples.

how to aggregate the e_{ij} scores assigned to an object o_j in order to derive a measure q_j that best reflects the ground-truth quality of o_j.

A straightforward approach to determining quality is to average the scores given to an object, as shown in Equation (1). This approach, which we term the *Naive* model, treats the scores by different reviewers equally, such that

$$q_j = \underset{i}{\mathcal{A}vg}\ e_{ij}. \tag{1}$$

The Naive model would be adequate if all reviewers were to rate all (or many) objects, as supported by the law of large numbers [Grimmett and Stirzaker 1982]. However, this is not a realistic and practical criterion supported by most social media applications, in which users may either voluntarily find or be assigned objects to rate. Therefore, we consider rating scenarios in which many objects are rated by a few reviewers (say, less than ten), which better represents most social media applications with long tail frequency distributions. When an object receives a small number of rating scores, one or two reviewers could adversely skew its aggregate quality. In particular, reviewers are not necessarily on an equal ground when assigning their scores, due to differences in background, perspective, standard, etc., which may affect the fairness of rating.

1.2. Approach

In this article, we model the variance among reviewers in terms of *leniency*, or the tendency of a reviewer to assign a higher score to an object than the object deserves (as determined by the quality of the object). Ours is a data-centric approach that determines leniency from the rating scores alone. Once determined, the leniency information can be used to adjust the rating scores appropriately to arrive at q_j values that better reflect the quality of objects. There could be various reasons behind leniency. For one, different reviewers may subscribe to different sub-ranges within the rating scale. However, we do not delve into the possible causes of leniency, and instead, focus on the impact of leniency on rating scores. Two insights about leniency underlie our approach.

Insight I. Networked Approach to Leniency. The leniency of a reviewer can only be determined relative to her coreviewers. A reviewer who tends to give a higher rating score than a majority of coreviewers has a tendency of being lenient. Similarly, when considering the quality of an object, we need to consider how other objects have been rated by its reviewers. The following example illustrates this point.

Example 1.1. Figure 2(a) and 2(b) show two sets of rating data under the same reviewer/object assignment. The matrix elements are the e_{ij} scores. A '—' denotes that the reviewer has not evaluated the object. In both datasets, o_1 receives the same set of scores (0.7 from r_1, 0.4 from r_2, and 0.4 from r_3). Using the averaging approach (Naive

model), we arrive at the same overall score for o_1 ($q_1 = 0.5$) in both datasets. However, a more reasonable outcome is that o_1 should receive a lower overall score in the first dataset than in the second.

Consider Figure 2(a) first. The varying scores received by o_1 suggest that either r_1's score is too high, or r_2 and r_3's scores are too low. If we consider the scores of other objects, we observe that r_1 also assigns higher scores than her coreviewers on o_2 and o_3. In contrast, r_2 and r_3 tend to agree with their coreviewers on o_4 and o_5 respectively. The record suggests that it is more likely that r_1 is lenient, and r_2 and r_3 are not. Thus, it makes sense to trust the scores by r_2 and r_3 more.

In Figure 2(b), it is r_1 who tends to agree with her coreviewers on o_2 and o_3, whereas r_2 and r_3 show a record of assigning lower scores than the majority of their coreviewers on o_4 and o_5. In this case, it makes sense to trust r_1's score more, even though r_1 is the minority.

Insight II. Mutual Dependency between Leniency and Quality. The two measures of interest, leniency and quality, are mutually dependent. One the one hand, to determine a reviewer's leniency, we need to know the quality of objects rated by the reviewer as a baseline to measure leniency. On the other hand, to determine the quality of an object, we need to know the leniency of its reviewers.

In this work, we assume that the rating scores represent a ground truth that can be trusted for the study. We believe that in general successful social media sites support a majority of reviewers who are honest, acting according to their best judgment when assigning ratings. In some cases, where rating is voluntary, reviewers may be motivated differently. For instance, some only assign ratings when they have negative experience. However, this does not present a major problem to our approach as long as this phenomenon occurs generally, in which case, the relative standing among objects will not be directly affected. The true signal really comes from the relative ratings (and not the absolute ratings) assigned by the same reviewer on two different objects.

1.3. Contributions

We make the following technical contributions in this article.

(1) We identify a new approach to the score-summarization problem, which concerns how to mine the leniency behavior of reviewers and use it to derive the quality of objects more equitably.
(2) We develop the *Leniency-Aware Quality* (LQ) model that solves leniency and quality simultaneously, using the previously mentioned insights on the networked approach to leniency and the mutual dependency between leniency and quality. The model features two possible modes of compensating for leniency: *Relative* mode, which models leniency in relative terms, and *Absolute* mode, which models leniency in absolute terms.
(3) We show that two types of solution to the LQ model exists. The *exact* solution represents leniency and quality as numeric measures, and the *ranked* solution represents leniency and quality as ranked measures. We characterize the conditions for the existence of each solution.
(4) We verify the efficacy of our approach through experiments on real-life and synthetic datasets, showing that the LQ model outperforms the baseline models, both in producing more reasonable outcomes and in reconstructing the predetermined ground-truth more accurately.

This problem and the solution based on *Relative* mode (described in Section 3.1) were first explored in our earlier work [Lauw et al. 2007]. In this article, we significantly extend our treatment of this approach, by introducing the *Absolute* mode

(described in Section 3.2). In addition, we now comprehensively evaluate the proposed approaches on a much larger real-life dataset, in addition to a synthetic dataset with known ground truth, against not only the Naive baseline, but also reputation-based and distribution-based approaches.

1.4. Paper Organization

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed LQ model. Section 4 discusses the two types of solution (exact and ranked) to the LQ model. Sections 5 and 6 verify the effectiveness of our approach through experiments on real-life and synthetic datasets respectively. Section 7 concludes.

2. RELATED WORK

Our work is closely related to previous work on score summarization and score normalization, which we will discuss shortly. The key difference is that our work stands out in terms of modeling leniency (a micro-behavior of individual reviewers). Leniency is inherently different from other metrics (e.g., reputation, deviation, average) that have been previously studied. In particular, the leniency of one reviewer not only depends on her own score, but also on the score of coreviewers. So we have to extract leniency by a mutual-reinforcement model.

Score Summarization. Score summarization is concerned with aggregating rating scores into an overall quality for each object. Some previous work models the varying ability of reviewers in assessing the quality of an object (also referred to as "reputation"). Instead of the simple mean (the Naive model), reputation-based approaches use the weighted mean of rating scores, with more reputable reviewers given higher weights, as given in Equation (2). Riggs and Wilensky [2001] base the reputation w_i of a reviewer r_i on consensus, that is, how closely r_i's rating scores are to the object averages, as shown in Equation (3). Alternatively, the reputation of a reviewer may also be based on the opinions of other reviewers [Chen and Singh 2001]. However, reputation does not equal to leniency, as even a reputable reviewer may still be lenient, that is may use a higher range of scores compared to other reviewers.

$$q_j = \frac{\sum_i w_i \times e_{ij}}{\sum_i w_i}. \tag{2}$$

$$w_i = 1 - \underset{j}{\mathcal{A}vg} \; |e_{ij} - q_j|, \tag{3}$$

Score Normalization. Score normalization deals with converting rating scores of reviewers to a normalized scale. Most works assume that scores by each reviewer can fit into a particular distribution, and that reviewers rate objects with comparable distributions. Given that most reviewers tend to rate very few objects, it is unlikely this assumption will hold. In contrast, our approach does not assume any distribution, but rather, is data-centric. We use a mutual-reinforcing model to compute the converged values.

One popular choice is the normal distribution, in which each reviewer r_i is associated with a mean μ_i and standard deviation σ_i of r_i's scores on various objects. Resnick et al. [1994] employ z-score normalization [Walpole et al. 2002], in which a rating score e_{ij} by reviewer r_i is converted into its z-score z_{ij}, according to Equation (4). Given that reviewers may have different means and standard deviations, z-score normalization aims to calibrate their scores to a more equitable standard [Arkes 2003]. Other variations include simply subtracting the rating score by a reviewer's average [Sarwar et al.

2000], subtraction followed by L_p normalization [Lemire 2005] (in the case of L_2 norm, it reduces to z-score), or by factor analysis [Traupman and Wilensky 2004b].

$$z_{ij} = \frac{e_{ij} - \mu_i}{\sigma_i}. \tag{4}$$

Probability-based approaches [Fernandez et al. 2006; Jin and Si 2004; Jin et al. 2003] convert a rating score to a probability value. One of the objectives is to remove the impact of outlier ratings. However, to estimate the probability values well, this approach relies on the availability of many ratings per reviewer.

Score normalization is also used in metasearch [Arampatzis and Kamps 2009; Fernandez et al. 2006; Manmatha and Sever 2002], where the objective is to combine the outputs of several search engines in response to a query. The basic approach in such cases is to fit the relevance scores to a binary mixture model—normal distribution for relevant pages and exponential distribution for irrelevant ones. The context is very different from the problem we consider here. For one thing, a reviewer is not normally associated with irrelevant objects.

Recommender Systems. According to [Adomavicius and Tuzhilin 2005; Herlocker et al. 2000, 2002], recommender systems would consider recommending relevant items to a target user, based on some sort of similarity between the target user and other neighboring users. Content-based systems exploit certain profiles of items and users to define this similarity, whereas collaborative filtering (CF) exploits the ratings on items to define this similarity. The former assumes that profile information of items and users is available, whereas the latter assumes that a rating database is available. This is different from our problem of finding the "true" (aggregate) rating of each item given a set of raw ratings collected from a set of users. We neither assume that profile information is given, nor that the rating data is the"ground truth". Rather, our problem assumes that the raw ratings may be biased in that some users are more generous than others, and the "true" rating would correct this bias effect. Therefore, in order to find the "true" rating, our problem also finds the generosity or leniency of each user.

Other than similarity between users, several works on recommender systems also consider the notion of trust [Massa and Avesani 2005] or social relationships between users [Ma et al. 2008, 2009]. In our problem, we are not given such social or trust networks, except for the raw user-item rating data. The data is "raw" in the sense that an individual rating may be biased, therefore, is not trusted. In the review problem considered here, it is not reasonable to assume that the trust and bias information about reviewers are known. No reviewer will admit that she or he is biased. Our work detects the bias of reviewers by analyzing the collective behavior of reviewers, on the assumption that a majority of reviewers behaves normally.

Some recommender systems employ pre-processing techniques designed to "correct" some global effects such as the number of ratings or the average ratings from the rating scores [Bell and Koren 2007], or to fill up missing rating values [Shen et al. 2006]. Our problem can be seen as correcting the leniency effect of reviewers to determine the "true" quality of objects. Hence, the methods presented in this paper may potentially help recommender systems to arrive at better rating predictions. Although it is not the main focus of our work here, we will conduct a preliminary investigation of the utility of our work for rating prediction in Section 5.5.

Miscellaneous. Multi-criteria decision making (MCDM) [Figueira et al. 2005; Korhonen et al. 1992] deals with how to make an optimal decision, taking into account two or more potentially conflicting criteria. The optimal decision may vary according to the subjective preferences of the decision maker. In our problem setting, the quality

measure is objectively associated with an object and is determined from the score, data without using subjective parameters.

The rating behavior of reviewers may be influenced by certain biases. The study of cognitive biases concerns people's predisposed opinions that may come from specific heuristics or mental shortcuts [Bazerman 1990; Busenitz and Lau 1996; Simon et al. 1999]. The main difference is that cognitive science is concerned with finding the possible causes of biases, and therefore, hypothesizing on the possible causes is central to the study of cognitive biases [Blackburn and Hakel 2006]. Our approach is different, as we focus on detecting reviewer leniency and factoring it into overall scoring of object quality. Testing each possible hypothesis would require much more additional information on reviewers or objects than is available. Such studies are also more appropriately done within the cognitive sciences.

Several works have also identified types of frauds in rating systems and how to detect them in various contexts, such as product reviews [Jindal and Liu 2007], trading communities [Bhattacharjee and Goel 2005; Dellarocas 2000; Zhang and Cohen 2006], and recommender systems [Lam and Riedl 2004; Mobasher et al. 2006]. Fraudulent ratings may be different from ratings by reviewers with leniency. In our article, we assume all reviewers are doing their best when rating an object, so there is no fraud. Therefore, those methods may not apply.

Score summarization in Web-based social media can be studied as a problem in social network mining. Social network involves the study of a network of associations among entities [Wasserman and Faust 1994]. It concerns analyzing a network to address such issues as node centrality [Faust 1997], trust [Golbeck and Hendler 2006; Guha et al. 2004], privacy [Backstrom et al. 2007], and community discovery [Borgatti and Everett 1997; Tantipathananandh et al. 2007; Yang et al. 2007; Zhou et al. 2006]. Our work models a social behavior (leniency behavior) from network-structured data (rating network) that involves collaboration among reviewers. Ours is also the first work to address the issue of leniency in a network environment.

Finally, our work is also related to link analysis [Borodin et al. 2005; Haveliwala 2003], which discovers important nodes (e.g., webpages) through intensive analysis of link data (e.g., weblinks). The most well-known algorithms are PageRank [Page et al. 1998] and HITS [Kleinberg 1999]. However, these works are mainly based on the notion of popularity (e.g., link count), which is not congruent with leniency or quality. In general, the leniency of a reviewer or the quality of an object is not related to the count of scores. Rather, it is the score value that matters.

3. LENIENCY-AWARE QUALITY (LQ) MODEL

Given a score data, we seek to determine the quality q_j of each object o_j. The key principle in our approach is to model the leniency l_i of each reviewer r_i, and use it to derive q_j.

3.1. Model

Our LQ model consists of a pair of equations (Equations (5) and (6)) that determine the leniency of reviewers and the quality of rated objects respectively. To measure how lenient a reviewer r_i is, we need to know how r_i's rating scores compare to the quality of rated objects. Suppose that q_j is known, the extent to which the given score e_{ij} is inflated or deflated can be measured by $\frac{e_{ij}-q_j}{e_{ij}}$. Note that the inflation (or deflation) is measured relative to the base score e_{ij}.[6] If r_i regularly inflates her rating scores, we have even more evidence that r_i is lenient. Hence, to determine l_i, we aggregate

[6]The case of $e_{ij} = 0$ should be avoided by replacing such e_{ij} with an appropriately small value.

$\frac{e_{ij}-q_j}{e_{ij}}$ over the set of objects that r_i has rated, as shown in Equation (5). Here, we use *average* as the aggregation function. Consequently, $l_i > 0$ denotes a lenient reviewer, $l_i < 0$ denotes a strict reviewer; and $l_i = 0$ denotes a neutral reviewer.

$$l_i = \underset{j}{\mathcal{A}vg} \left(\frac{e_{ij} - q_j}{e_{ij}} \right) \tag{5}$$

Note that q_j in Equation (5) is not known beforehand. It is to be determined as an aggregation (here, we assume average) of rating scores assigned to o_j. However, suppose that we know l_i of each r_i who has rated o_j, we can then compensate for each r_i's tendency to inflate or deflate rating scores. This compensation approach of deriving q_j is shown in Equation (6). If $l_i < 0$, we revise the rating score e_{ij} upwards. If $l_i > 0$, we revise it downwards. The adjustment is proportional to the base score e_{ij}. $\alpha \in [0, 1]$ is a user-determined compensation factor, which controls the extent to which the scores may be adjusted to compensate for leniency. Larger α would lead to larger compensation.

$$q_j = \underset{i}{\mathcal{A}vg} \left[e_{ij} \cdot (1 - \alpha \cdot l_i) \right]. \tag{6}$$

By compensating, Equation (6) estimates the score that would have been assigned by a lenient reviewer had she been neutral ($l \approx 0$). That way, the quality scores of two objects rated by different sets of reviewers, who may use different ranges within the rating scale, would be more comparable. Equation (6) can even return $q_j < min_i(e_{ij})$, if all or most of o_j's reviewers have $l_i > 0$, or $q_j > max_i(e_{ij})$ if all or most reviewers have $l_i < 0$. This is possible as the LQ model does not look at each object in isolation, but instead considers the broader context (how its reviewers have rated other objects, how those other objects are rated by other reviewers, and so on). In contrast, the Naive model (Equation (1)) and the Weighted model (Equation (2)) confine q_j to $[min_i(e_{ij}),\ max_i(e_{ij})]$. Thus, the LQ model is better positioned than these two models in salvaging an object from a very skewed assignment of reviewers (such as an object whose reviewers are all strict).

The two variables l_i and q_j are mutually dependent and must be determined simultaneously. This dependency extends to all reviewers and objects connected to one another within the rating network. This is because to know a given l_i requires us to know the q_j of all objects rated by r_i. However, for each q_j, we need to know the leniency of r_i, as well as those of r_i's coreviewers on o_j. This dependency could only be resolved by considering the leniency of every reviewer and the quality of every object simultaneously.

Note that the Naive approach is a special case of this model. When $\alpha = 0$, no adjustment for leniency is done, and Equation (6) is reduced into Naive's Equation (1).

Example 3.1. Table I(a) and I(b) display the quality and leniency computed using Naive and *LQ* models for Figure 2(a) and 2(b), respectively. For this example, *LQ* uses the *exact* solution (to be introduced in Section 4) at $\alpha = 0.5$. In both scenarios, Naive gives all objects the same quality of 0.50. We claim, however, that the different rankings of objects by *LQ* are more intuitive.

Consider Table I(a) first. *LQ* considers objects rated by r_1 (o_1, o_2, and o_3) to be of lower quality than the other objects (o_4 and o_5). Note that r_1 is considered lenient ($l_1 = 0.31$ by *LQ*) due to r_1's tendency to give higher scores than her coreviewers on o_1, o_2, and o_3. Adjusting for r_1's leniency, *LQ* arrives at the net lower quality of o_1, o_2, and o_3 (0.48 by *LQ*), as compared to that of o_4 and o_5 (0.53 by *LQ*).

In Table I(b), *LQ* considers objects rated by r_2 or r_3 (o_1, o_4, and o_5) to be of higher quality than the other objects (o_2 and o_3). Adjusting for the strict scoring by r_2 and r_3

Table I. Quality and Leniency

(a) Rating Data 1

Naive	*LQ*	
$q_1 = 0.50$	$q_1 = 0.48$	$l_1 = 0.31$
$q_2 = 0.50$	$q_2 = 0.48$	$l_2 = -0.14$
$q_3 = 0.50$	$q_3 = 0.48$	$l_3 = -0.14$
$q_4 = 0.50$	$q_4 = 0.53$	$l_4 = -0.14$
$q_5 = 0.50$	$q_5 = 0.53$	$l_5 = -0.14$

(b) Rating Data 2

Naive	*LQ*	
$q_1 = 0.50$	$q_1 = 0.56$	$l_1 = 0.10$
$q_2 = 0.50$	$q_2 = 0.47$	$l_2 = -0.55$
$q_3 = 0.50$	$q_3 = 0.47$	$l_3 = -0.55$
$q_4 = 0.50$	$q_4 = 0.51$	$l_4 = 0.10$
$q_5 = 0.50$	$q_5 = 0.51$	$l_5 = 0.10$

($l_2 = -0.55$, $l_3 = -0.55$ by LQ), LQ results in the higher quality of objects rated by r_2 or r_3 ($q_1 = 0.56$, $q_4 = 0.51$, $q_5 = 0.51$ by LQ) than those of o_2 and o_3 ($q_2 = 0.47$, $q_3 = 0.47$ by LQ).

3.2. Relative and Absolute Compensation Modes

Equation (5) (and correspondingly Equation (6)) models leniency in relative terms. A reviewer's leniency is assumed to induce her to inflate (or deflate) her scores by a certain fraction or percentage. For instance, a reviewer with $l_i = 0.1$ tends to inflate her scores by 10%. In Equation (5), the difference between e_{ij} and q_j is taken relative to e_{ij}. In turn, in Equation (6), the compensation component $(1 - \alpha \cdot l_i)$ adjusts the score e_{ij} in relative terms, as well. We term this approach the *Relative* compensation mode.

Another approach is to model leniency in absolute terms, which we term the *absolute* compensation mode. In this approach, l_i is an absolute value by which r_i inflates (or deflates) her scores. For instance, a reviewer with $l_i = 0.1$ tends to inflate her scores by 0.1. This compensation mode gives rise to a different pair of leniency and quality equations (Equation (7) and Equation (8)). In determining leniency, the difference between e_{ij} and q_j is taken as an absolute value $(e_{ij} - q_j)$ in Equation (7). In determining quality, the adjustment to e_{ij} is absolute $(e_{ij} - \alpha \cdot l_i)$ in Equation (8).

$$l_i = \underset{j}{\mathcal{A}vg}\,(e_{ij} - q_j) \tag{7}$$

$$q_j = \underset{i}{\mathcal{A}vg}\,(e_{ij} - \alpha \cdot l_i). \tag{8}$$

The main difference between the two compensation modes is the underlying assumption on the mechanism by which a lenient reviewer would inflate (or deflate) her scores. However, since the equation for quality adjusts for leniency accordingly, the outcomes of the two modes may not be very different. Our experiments in Section 6 show that there are only minor differences between the two modes in terms of the quality of objects.

4. SOLUTION TYPES

A solution to the LQ model tells us the relative comparison among objects in terms of quality, and among reviewers in terms of leniency. We identify two approaches for reaching a solution. The first approach, which we call the *exact* solution, treats the model as a linear system of equations to be solved for exact values of quality and leniency. The second approach, which we call the Ranked solution, treats the model as a ranking problem and solves it for a ranking of objects by quality and a ranking of reviewers by leniency. The two solutions may not be identical. Ranked solution is valuable, as sometimes no Exact solution exists, but a unique ranking still exists, and knowing the ranking suffices for the application. For instance, a conference program chair may only be interested in ranking all submitted papers by quality, so as

to accept the best papers. Below, we characterize these two solutions for the Relative compensation mode. Similar discussions apply to Absolute.

Before describing the solutions, we first rewrite Relative's Equations (5) and (6) into Equations (9) and (10), respectively, where $c_{ij} \in \{0, 1\}$ is the connectivity flag. $c_{ij} = 1$ when r_i has evaluated o_j, and 0 otherwise. We assume that every object is rated by some reviewers, and that every reviewer evaluates some objects. Therefore, there will be no division by zero.

$$q_j = \frac{\sum_i [c_{ij} \cdot e_{ij} \cdot (1 - \alpha \cdot l_i)]}{\sum_i c_{ij}}, \tag{9}$$

$$l_i = \frac{\sum_j [(c_{ij}/e_{ij}) \cdot (e_{ij} - q_j)]}{\sum_j c_{ij}}. \tag{10}$$

The set of equations comprising Equation (9) for every o_j and Equation (10) for every r_i can be more compactly expressed as a pair of matrix equations, as in Equation (11) and Equation (12). For m reviewers and n objects, $\boldsymbol{Q}$ is $n \times 1$ vector of q_j's, L is $m \times 1$ vector of l_i's, and $\mathbf{1}$ is a vector of appropriate length containing all 1's. U is $m \times n$ matrix whose element $u_{ij} = [(c_{ij} \cdot e_{ij})/\sum_i c_{ij}]$. V is $m \times n$ matrix whose element $v_{ij} = (c_{ij}/\sum_j c_{ij})$. W is $m \times n$ matrix whose element $w_{ij} = [(c_{ij}/e_{ij})/\sum_j c_{ij}]$. $\boldsymbol{Q}$ and L are variables, and the rest are inputs.

$$\boldsymbol{Q} = U^T\mathbf{1} - \alpha U^T L, \tag{11}$$

$$L = V\mathbf{1} - W\boldsymbol{Q}. \tag{12}$$

Substituting Equation (12) into Equation (11), we get a recursive equation in terms of $\boldsymbol{Q}$, given in Equation (13). A simpler form is given in Equation (14), where $X = (U^T\mathbf{1} - \alpha U^T V\mathbf{1})$ and $Y = (\alpha U^T W)$. Intuitively, any q_j (in left-hand side $\boldsymbol{Q}$) could be expressed in terms of the quality of other objects (in right-hand side $\boldsymbol{Q}$), as determined by X and Y that govern how these objects are connected in the network. Thus, we need to solve for $\boldsymbol{Q}$ (which can then be used to solve for L) yielding.

$$\boldsymbol{Q} = U^T\mathbf{1} - \alpha U^T V\mathbf{1} + \alpha U^T W\boldsymbol{Q}, \tag{13}$$

$$\boldsymbol{Q} = X + Y\boldsymbol{Q}. \tag{14}$$

Subsequently, we distinguish between the Exact solution, which solves Equation (14) as a linear system of equations, and the Ranked solution, which derives a unique ranking from an eigenvector equation modified from Equation (14).

4.1. Exact Solution

The Exact solution is the unique value of $\boldsymbol{Q}$ (and the corresponding (L)) satisfying Equation (14). The matrix Equation (14) stands for a system of n linear equations in terms of various q_j's. From linear algebra [Anton and Rorres 1987], we know that such a system of linear equations may be in one of three situations.

Case 1. *Consistent and Uniquely Determined.* There is one unique solution, which is the intersection point of the n linear equations.

Case 2. *Consistent and Underdetermined.* There are infinitely many solutions, which lie on the line or plane where the linear equations meet.

Case 3. *Inconsistent.* There is no solution, as the linear equations do not meet.

Hence, Exact solution exists only under Case 1, which produces a unique Q. This solution is given in Equation (15). For the solution to be unique, $(I - Y)$ must be invertible, which is true if and only if $det(I - Y) \neq 0$. Once Q is determined, L can be derived using Equation (12). Elements of Q and L are the exact values of quality and leniency that we are interested in. Equation (15) is

$$Q = (I - Y)^{-1}X. \tag{15}$$

Failing the test $det(I - Y) \neq 0$, Equation (14) falls under Case 2 or Case 3, for which an Exact solution does not exist. However, a Ranked solution may still exist for Case 2. We describe this solution which preserves the ordering among quality and leniency values in Section 4.2. In the ill-conditioned or rank-degenerate instances (e.g., Case 3), one option is to fall back on the Naive model, which will always produce a solution. This is reasonable in that the ill-conditioned problem structure in this case does not allow a feasible solution. However, such cases are rare, thus, our approaches often provide better solutions than the Naive model.

4.2. Ranked Solution

For the Ranked solution, we are only interested in the ranking by quality (and by leniency). We could derive such a ranking from Equation (16), which is modified from Equation (14) by adding a nonzero, real-valued scalar variable λ. Intuitively, Equation (16) says that any q_j (in left-hand side Q) could be expressed in terms of the quality of other objects (in right-hand side Q), after rescaling by λ. In other words, the Q that satisfies Equation (16) would preserve the relative ratio among q_j elements (and the ranking by quality),

$$\lambda\, Q = X + YQ. \tag{16}$$

Due to the λ variable, Ranked's Equation (16) is fundamentally different from Exact's Equation (14). Thus, the two solutions may not produce identical rankings. For Ranked, we are only interested that such a λ exists. The value or sign of λ is not important, as once λ is known, we could always rescale $\lambda\, Q$ back to Q (normalization). Moreover, as we are solving Equation (16) as an eigenvector equation, the existence of a Ranked solution is dependent on conditions different from the three cases mentioned in Section 4.1.

Since we are only interested in the direction of vector Q (Q or any scaling of Q is acceptable), we can reformulate Equation (16) as an eigenvector equation (Equation (17)). The $n \times n$ matrix X^n is formed by replicating the $n \times 1$ vector X across n columns. β is the inverse of the sum of elements of Q, that is, $\beta = (\sum_j q_j)^{-1}$. We see that Q is in fact an eigenvector of $(\beta X^n + Y)$. In fact, what we want is the dominant eigenvector.

$$\lambda\, Q = (\beta X^n + Y)\, Q. \tag{17}$$

As β and Q are mutually dependent, we could break this dependency by fixing the value of β in order to derive a unique dominant eigenvector Q. An intuitive choice for value β is the inverse of the sum of quality by the Naive model. This has the advantage of preserving the sum of quality before and after compensation, which would prevent a general inflation or deflation of quality (for the quality of some objects to go up, those

of others must come down). For a fixed value of β, the eigenvector equation can be more simply expressed as Equation (18), where $Z = \beta X^n + Y$,

$$\lambda\ Q = Z\ Q. \tag{18}$$

Iterative methods [Anton and Rorres 1987] can be used to solve Equation (18) to get the dominant eigenvector Q. The iterative form is $Q_{k+1} = Z\ Q_k$. The only variable is Q, as λ is removed by normalizing Q after each iteration. Normalization in this case returns Q to the state of $\sum_j q_j = \beta^{-1}$, where β^{-1} is the sum of quality by Naive. Subject to the assumption that Z is diagonalizable (it has linearly-independent eigenvectors) and has a uniquely largest eigenvalue [Golub and Van Loan 1996], as k increases, Q_k will converge to the dominant eigenvector of Z, almost independently of the initial Q_0.

The notion of convergence to fixed points is defined by the relative ratio of quality (i.e., the ranking), instead of absolute quality. The ratios of 2:3, 4:6, 6:9, and so on, are all considered the same fixed point. The normalization or rescaling does not affect the existence of a fixed point, and neither does it imply that there is a lack of a fixed point in the iterations. The fixed point is reached when the ratio converges. Therefore, the fixed point, in this sense, refers to the relative ratio and is independent of scaling absolute values. A similar formulation and convergence have been previously attempted in works on Web search ranking (HITS [Kleinberg 1999] and PageRank [Page et al. 1998]). Once converged, the elements of Q (and the corresponding L) are used to rank objects (and reviewers).

Ranked Solution vs. Exact Solution. In summary, we have introduced two independent solution types derived from similar, but slightly different matrix equations. Exact solution produces exact values of quality/leniency, which can also be used for ranking. Ranked solution produces only the rankings by quality/leniency. Although Ranked solution is the weaker solution, since it only produces the ranking of objects, it is still necessary because in some cases the ranking of objects by quality may still be determined even if no Exact solution exists. We provide one such example scenario (this is not the only such scenario, but it is chosen for expository purpose).

Consider the following scenario with three reviewers and three objects. r_1 rates objects $\{o_1, o_2\}$; r_2 rates $\{o_1, o_2, o_3\}$; r_3 rates $\{o_2, o_3\}$. All the rating scores are uniformly 0.5.

$$\begin{array}{c} \quad o_1 \quad o_2 \quad o_3 \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \end{array} \begin{pmatrix} 0.5 & 0.5 & - \\ 0.5 & 0.5 & 0.5 \\ - & 0.5 & 0.5 \end{pmatrix} \end{array}$$

To obtain the Exact solution for the object quality, we need to solve the system of Equations (7) and (8) for each reviewer and object, respectively (for *Absolute* compensation mode and $\alpha = 1$). However, this scenario falls under Case 2 (consistent and underdetermined). Because of the symmetry of connectivity (the adjacency matrix would be identical if we swap the objects and the reviewers), as well as the same rating averages for all reviewers and objects, the equations for leniency mirror the equations for quality. As a result, we effectively have only three equations to solve six variables $\{q_1, q_2, q_3, l_1, l_2, l_3\}$, resulting in an underdetermined case. The system of linear equations reduce to the following three equalities: $q_1 = q_2$, $q_2 = q_3$, and $q_1 = q_3$. No exact solution can be found.

However, the relative ranking of quality can be deduced from the above equalities, as well as from an inspection of the rating scenario. That is, *all three objects should*

Table II. LQ Solutions

		Solution Type	
		Exact	*Ranked*
Compensation Mode	*Relative*	R-Exact	R-Ranked
	Absolute	A-Exact	A-Ranked

Table III. Data Size

	Original	*After removing reviewers/objects with* < 3 *ratings*	*Keeping only the first 7 ratings per object*
Reviewers	6,040	6,040	1,071
Objects	3,706	3,503	3,431
Ratings	1,000,209	999,917	22,268
Ratings per object	1–3428 (median: 124)	3–3,428 (median: 140)	3–7 (median: 7)
ratings per reviewer	20–2314 (median: 96)	19–2,290 (median: 96)	3–886 (median: 7)

be ranked similarly. In this case, the ranked solution exists and will produce an equal ordering of the objects by quality.

5. EXPERIMENTS ON REAL-LIFE DATASETS

The objective of experiments on real-life datasets is to verify the efficacy of the proposed LQ model, primarily by comparing it against the Naive model (Equation (1)). First, we investigate whether and how the number of ratings that an object has affects the results. Next, we conduct overall comparison of ranked lists generated by different models to reveal whether LQ results in a significant differentiation in rankings. Using several case examples, we investigate whether extreme disagreements in quality rankings can be explained in favor of LQ. Finally, we conduct a preliminary investigation into the potential application of the proposed model for the rating prediction task.

Table II shows LQ's four possible solutions, owing to two compensation modes (Relative and Absolute) and two solution types (Exact and Ranked). In these experiments, we set $\alpha = 0.5$. For this α value, Exact and Ranked solutions exist for our dataset. Experiments on different α values will be covered by our experiments on synthetic datasets (see Section 6). The four solutions (R-Exact, R-Ranked, A-Exact, and A-Ranked) are compared against one another as well as against Naive, Zscore, and Riggs. Zscore is a standard distribution-based score normalization method, while Riggs is based on reviewers' reputation. We have described both in Section 2.

5.1. Dataset

The dataset used in these experiments was collected from GroupLens.[7] The "One Million MovieLens Dataset" contains ratings by users of the movie recommendation site MovieLens.[8] We chose this dataset as it was a large, public, and well-cited dataset. As shown in Table III, there were 6,040 reviewers, 3,706 objects (movies) and 1,000,209 scores. Each reviewer evaluated at least 20 objects. Each object may be evaluated by as few as 1 reviewer. Although this dataset better fits the notion of subjective rating, we find that we still get good results in this dataset. Morever, there is a lack of other large-scale objective rating datasets.

[7] www.grouplens.org
[8] www.movielens.org

The rating information was extracted and further processed as follows. We rescaled the rating scores, originally on the scale of 1 to 5 stars, to a new range of 0.2 to 1.0 by a simple division by 5. We also ensured that each object had at least three reviewers and each reviewer had at least three objects, by iteratively removing objects with less than three reviewers and reviewers with less than three objects, until there were no more such objects/reviewers. This removed the occasional reviewers/objects and lent greater support when inferring the behavior of reviewers/objects. As shown in Table III, the data size after filtering was still large, with 6,040 reviewers and 3,503 objects. The reviewers and objects actively participated in the evaluation, as shown by the high number of ratings per object (with median of 140) and ratings per reviewer (with median of 140). This filtered data will be used in Section 5.2.

5.2. Varying Rating Count

As mentioned in Section 1, we hypothesize that the proposed LQ solution will perform better than Naive where there are relatively few ratings per object. According to the law of large numbers [Grimmett and Stirzaker 1982], when the number of ratings per object is very high, the mean will be a sufficient approximation of the true quality.

To see if this hypothesis bears out in real-life data, we conduct an experiment by varying the number of ratings per object, while measuring its impact on the similarity between LQ and Naive solutions. We extract a subset of the dataset for each rating count n, by retaining only the first n ratings (chronologically) of each object with more than n ratings. We compare the resulting rankings of objects by quality of LQ and Naive, using the *Kendall* similarity measure [Dwork et al. 2001; Fagin et al. 2003].

Given two ranked lists X and Y, Kendall counts the number of pairs for which X and Y agree on their relative ranks, as shown in Equation (19), where k is the size of X and Y. For an item t, its rank in X is $rank_X(t)$ and $rank_Y(t)$ in Y. Kendall penalizes each pair of items (t_1, t_2) where $rank_X(t_1) > rank_X(t_2)$ but $rank_Y(t_1) < rank_Y(t_2)$. The similarity value is in the range of [0%, 100%], with 100% indicating a complete agreement between LQ and Naive.

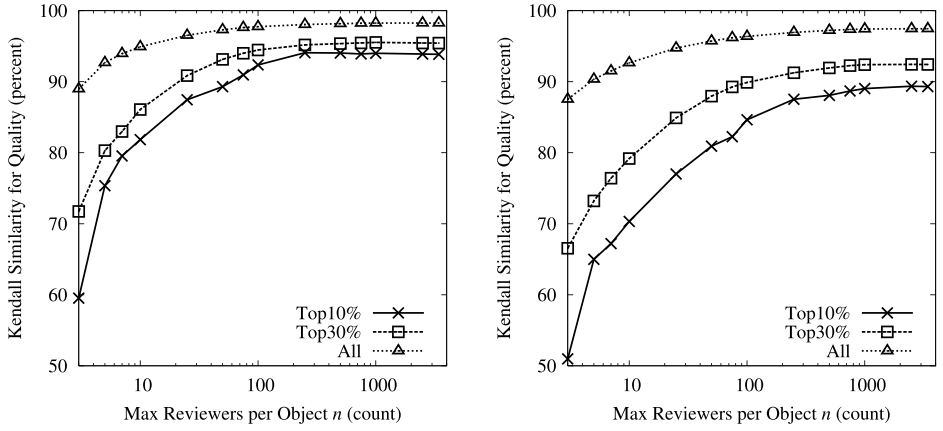
$$Kendall(X, Y) = \frac{|\{(t_1, t_2)|X \text{ and } Y \text{ agree on order of } (t_1, t_2)\}|}{\frac{1}{2}k(k-1)}. \tag{19}$$

Figure 3 shows the plot of Kendall similarity between A-Ranked and Naive's quality rankings for different values of n. Figure 3 shows the same for R-Ranked versus Naive. In both figures, two things are apparent. First, as n increases, LQ's solutions are increasingly similar to Naive. Second, there are more differences at the top ranks (e.g., Top 10%) than there are in lower ranks (e.g., Top 30% or All).

The first observation confirms the earlier hypothesis that when there are many ratings per object, the average rating (i.e., Naive) is sufficient, and LQ derives the same outcome. Commonly, the purpose of quality ranking is to evaluate and identify the top-ranked objects. Hence, the second observation further bolsters the value of LQ since it generates a more different outcome from Naive for the top-ranked objects. We will also take a deeper look at certain case examples in Section 5.4 to see if these rank differences imply a better ranking by LQ.

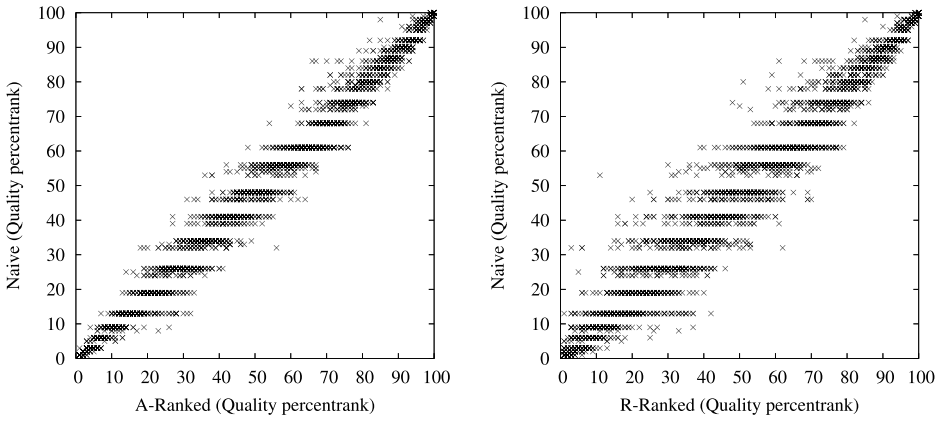
5.3. Comparison of Ranked Lists

Having studied the effects of rating counts in Section 5.2, we now focus on one rating-count setting. For this, we select the filtered dataset from the previous section in which each object has at most seven ratings (i.e., $n = 7$). The third column of Table III shows that more than 90% of objects were still represented after this filtering step.



(a) A-Ranked vs. Naive. (b) R-Ranked vs. Naive.

Fig. 3. Varying rating count.



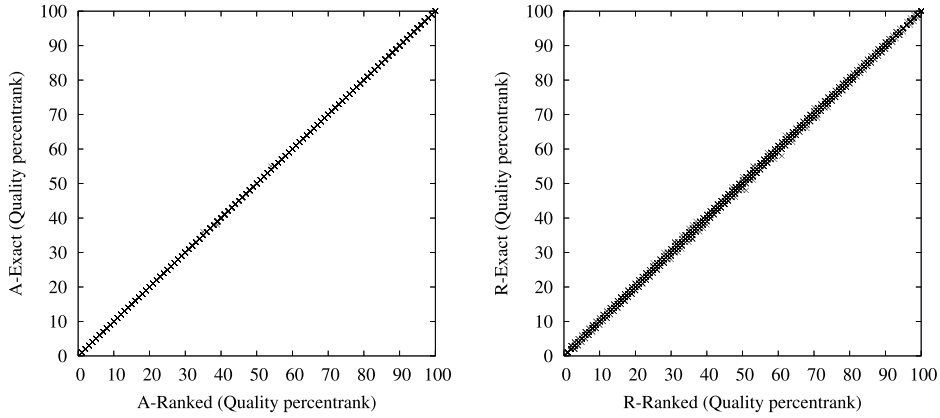
(a) Absolute compensation. (b) Relative compensation.

Fig. 4. Quality rank scatterplots: Naive vs. Ranked.

Here we conduct an overall comparison of the quality ranked lists generated by different solutions to gain a sense of how different LQ solutions are, from one another, as well as from Naive. For each solution, objects (or reviewers) were ranked in descending order of quality (or leniency). The highest value was given rank 1. Same values shared the same rank. For example, if the next three highest values were the same, they would share rank 2. Since there are 3,431 objects in the dataset, rank values goes from 1 to 3,431.

For ease of analysis and comparison, we further normalize the rank values into percentrank, which go from 1 to 100. Quality percentrank of an object reviewer o_j is derived from its rank as follows; $percentrank(o_j) = \lceil rank(o_j) \times 100 \div n \rceil$, where n is the total number of objects. For instance, a percentrank of 1 means an object's rank places it in the top 1% in terms of quality. Leniency percentrank of a reviewer is derived in a similar manner.

Naive vs. Ranked. First, we compare the ranked lists produced by A-Ranked and R-Ranked to that by Naive. Figure 4(a) shows a scatterplot of quality percentranks



(a) Absolute compensation. (b) Relative compensation.

Fig. 5. Quality rank scatterplots: Exact vs. Ranked.

for A-Ranked vs. Naive. Each point represents an object. Values on the x- and y-axes represent quality percentranks computed by A-Ranked and Naive, respectively. Figure 4 is the corresponding scatterplot for R-Ranked vs. Naive.

As shown in Figure 4(a) and (b), there are significant variances around the diagonal, revealing that both A-Ranked and R-Ranked rank objects differently from Naive. In particular, there are 139 objects sharing percentrank 9 by Naive. The same objects are given percentranks ranging from 4 to 26 by A-Ranked (Figure 4(a)) and from 3 to 37 by R-Ranked (Figure 4). Thus, the LQ solutions are more successful at differentiating even very competitive objects. This is useful in such situations as selecting the very best papers at conferences or proposals for funding.

Exact vs. Ranked. Figure 5(a) and (b) are the scatterplots for A-Ranked vs. A-Exact and R-Exact vs. R-Ranked, respectively. The points line up along the diagonal, implying that for this experiment, the Exact and Ranked solutions are practically identical. As leniency is mutually dependent on quality, the scatterplots for leniency are similar to Figures 5(a) and (b). Due to the Exact/Ranked similarity, we use only Ranked solutions to represent LQ, moving forward.

5.4. Case Examples

Here we showcase how the LQ model is more intuitively correct, by providing two examples of objects upon which LQ solutions disagree with Naive on their quality percentranks and showing how the disagreement can be explained in favor of LQ. These are followed by two examples of reviewers with very different leniency percentranks, showing how the difference comes about due to their rating behaviors.

Object Examples. Table IV describes the profile of *object-1880*, showing its quality values (and percentranks) computed by different solutions, its rating scores, and the leniency values (and percentranks) of its reviewers. Results for A-Exact and R-Exact are not shown, as they were practically identical to A-Ranked and R-Ranked, respectively. For object-1880, the quality percentranks assigned by A-Ranked (23) and R-Ranked (33) are much lower than those assigned by Naive (8), because object-1880's reviewers are generally lenient (with $l_i > 0$). A-Ranked and R-Ranked recognize and compensate for their tendency to inflate the rating scores, resulting in a lower-quality percentrank for object-1880.

Table IV. Profiles of Object *object-1880*

Object		**Quality (Rank)**		
		Naive	A-Ranked	R-Ranked
object-1880		0.85 (8)	0.77 (23)	0.72 (33)
Reviewers	$\mathbf{e_{ij}}$	**Leniency (Rank)**		
			A-Ranked	R-Ranked
user-3067	1.0		0.20 (7)	0.23 (8)
user-4682	0.8		0.21 (6)	0.25 (7)
user-4277	0.8		0.16 (14)	0.19 (13)
user-4937	0.8		0.10 (26)	0.12 (23)

Table V. Profiles of Object *object-1236*

Object		**Quality (Rank)**		
		Naive	A-Ranked	R-Ranked
object-1236		0.83 (9)	0.87 (6)	0.92 (3)
Reviewers	$\mathbf{e_{ij}}$	**Leniency (Rank)**		
			A-Ranked	R-Ranked
user-5987	1.0		−0.08 (81)	−0.42 (85)
user-5530	1.0		−0.07 (80)	−0.31 (79)
user-5693	0.8		−0.16 (93)	−0.56 (90)
user-5754	0.8		−0.08 (81)	−0.36 (82)
user-6036	0.8		−0.05 (75)	−0.17 (67)
user-5755	0.8		−0.01 (62)	−0.11 (60)
user-5493	0.6		−0.13 (90)	−0.42 (85)

The second object example *object-1236*, whose profile is shown in Table V, receives higher quality percentranks from A-Ranked (6) and R-Ranked (3) than from Naive (9), object-1236's reviewers are mostly strict (with $l_i < 0$). These reviewers' tendency to deflate their rating scores is taken into account by A-Ranked and R-Ranked, which then lift their quality percentranks correspondingly.

Reviewer Examples. A reviewer's leniency is determined by her rating behavior—whether she consistently rates higher or lower than the derived quality. Table VI shows the profile of *user-4556*, a strict reviewer ($l_i = -0.24$ by A-Ranked, $l_i = -0.94$ by R-Ranked) with very low leniency ranks (98 by A-Ranked, 97 by R-Ranked. Note that the leniency values are in absolute and relative terms for A-Ranked and R-Ranked, respectively. Comparing user-4556's rating score e_{ij} and the quality q_j of each rated object, we observe that the rating scores are consistently lower across the five objects ($0.6 < 0.67$, $0.4 < 0.64$, $0.4 < 0.59$, $0.2 < 0.67$, $0.2 < 0.43$ for e_{ij} vs. q_j by A-Ranked).

Table VII shows the profile of a lenient reviewer *user-2635*, with positive leniency values and very high leniency percentranks (0.57 and 1 by A-Ranked, 0.58 and 1 by R-Ranked). User-2635's rating scores on her four rated objects are consistently higher than the respective quality values ($1.0 > 0.50$, $1.0 > 0.42$, $1.0 > 0.41$, $1.0 > 0.41$ for e_{ij} vs. q_j by A-Ranked).

Table VI. Profile of Reviewer *user-4556*

Reviewer	**Leniency (Rank)**	
	A-Ranked	R-Ranked
user-4556	−0.24 (98)	−0.94 (97)

Objects	$\mathbf{e_{ij}}$	**Quality (Rank)**	
		A-Ranked	R-Ranked
object-3794	0.6	0.67 (46)	0.68 (43)
object-3718	0.4	0.64 (55)	0.66 (49)
object-3652	0.4	0.59 (67)	0.58 (69)
object-3747	0.2	0.67 (46)	0.66 (48)
object-2452	0.2	0.43 (92)	0.43 (92)

Table VII. Profile of Reviewer *user-2635*

Reviewer	**Leniency (Rank)**	
	A-Ranked	R-Ranked
user-2635	0.57 (1)	0.58 (1)

Objects	$\mathbf{e_{ij}}$	**Quality (Rank)**	
		A-Ranked	R-Ranked
object-3939	1.0	0.50 (83)	0.49 (85)
object-3942	1.0	0.42 (92)	0.40 (93)
object-3940	1.0	0.41 (93)	0.39 (94)
object-3941	1.0	0.41 (93)	0.38 (95)

5.5. Rating Prediction

The previous sections seek to evaluate the comparative solutions through in-depth analyses of the different outcomes. Another means of evaluation is whether the outcome could help improve the utility of an application. The objective of this experiment is to compare different score-summarization methods in terms of improving the task of rating prediction. For this experiment, we compare the LQ solutions against Naive (Equation (1)), Riggs (Equations (2) and (3)), and Zscore. The last solution involves first normalizing e_{ij} scores into z-scores, according to Equation (4), before deriving $q_j = \mathcal{A}vg_i\ z_{ij}$ in a similar way to Naive.

We employ a simple means of rating prediction as follows. At any one time, we remove one rating score e_{ij} from the dataset and attempt to produce a prediction e'_{ij} for this score, based on the remaining data. For each solution, the predicted score is the value that would best fit the q_j or l_i values computed from the remaining data. Specifically, for A-Ranked, we have $e'_{ij} = q_j + \alpha \cdot l_i$ based on Equation (8). For R-Ranked, we have $e'_{ij} = q_j \div (1 - \alpha \cdot l_i)$ based on Equation (6). For Naive, the predicted score e'_{ij} is the q_j value computed by Equation (1). For Riggs, while Equation (3) suggests that there could be two predicted values (i.e., $e'_{ij} = q_j \pm (1 - w_i)$), we take the average of the two, resulting in the predicted score $e'_{ij} = q_j$. For Zscore, the predicted score is the denormalized rating score, that is, $e'_{ij} = q_j \times \sigma_i + \mu_i$, based on Equation (4).

This rating-prediction exercise is repeated over a sample of 1,000 "missing" scores randomly selected from the dataset. To measure the performance, we take the mean absolute error (MAE), or the average of the absolute difference between the predicted

Table VIII. MAE Comparison

Solution	*First3*	*First5*	*First7*
A-Ranked	0.181	0.174	0.163
R-Ranked	0.181	0.176	0.165
Naive	0.187	0.186	0.175
Riggs	0.186	0.185	0.175
Zscore	0.187	0.181	0.167

and the true scores. Equation (20) shows how MAE is computed, where k denotes a particular missing score sample, such that

$$MAE = \frac{\sum_{k=1}^{1000} |e_{ij}^{'k} - e_{ij}^{k}|}{1000}. \tag{20}$$

We ran the rating-prediction experiments on three subsets of the data. In addition to the subset used in the previous sections, in which we keep only the first seven ratings per object (*First7*), we also created two other subsets in which we retain only the first 3 (*First3*) and 5 (*First5*) ratings per object, respectively.

The MAE values obtained by the different comparative solutions are shown in Table VIII. Note that a lower MAE value indicates better performance. In general, the fewer the number of ratings per object, the higher the MAE values. In addition, A-Ranked and R-Ranked have the best performance with the lowest MAE errors, followed by Zscore, and then Naive and Riggs. This result speaks in favor of our proposed LQ solutions. It shows that LQ is better at modeling the rating behaviors of reviewers. It also shows that LQ's outcome is more consistent than the other comparative solutions.

Table VIII also implies that the proposed methods are able to provide a meaningful answer with less data. It shows that A-Ranked and R-Ranked's errors, when considering only the first 3 ratings (First3) are similar to Naive's, when considering first 5 ratings (First5). Similarly, A-Ranked and R-Ranked's First5 errors are similar to Naive's First7 errors.

6. EXPERIMENTS ON SYNTHETIC DATASET

Most real-life datasets do not have ground-truth information on quality or leniency. Our experiments with synthetically generated datasets address the need to verify LQ's effectiveness against a known ground truth and to study the effects of various parameters on LQ's ability to reconstruct the ground truth.

6.1. Dataset Generation

Synthetic data generation is a rather complex process, as the propagation effect within a network (such as a rating network) and the interaction between data-generation parameters cannot be very precisely controlled. Hence, we choose to keep the data-generation scheme simple, with a few well-chosen parameters that would still allow us to draw meaningful insights.

The synthetic data simulates the scenario in which there are three classes of reviewers: strict ($l_i < 0$), neutral ($l_i = 0$), and lenient ($l_i > 0$) associated with different rating behaviors. The quality of objects follows a uniform distribution in the range of [0.2, 1.0]. There are four parameters (k, m, and n), and α, as described in Table IX.

The data-generation scheme involves the following steps.

(1) *Assign quality values.* We assign to each object a quality q_j, which is a random value in the range of [0.2, 1.0].

Table IX. Data Generation Parameters

Parameter	*Description*	*Default Value*
k	percentage of non-*neutral* reviewers	60%
m	percentage of *lenient* among non-*neutral* reviewers	50%
n	number of reviewers assigned to each object	10
α	compensation factor	0.7

(2) *Assign leniency values.* Select k% of reviewers, label m% of them as lenient ($l_i > 0$), and label the other $(100 - m)$% as strict ($l_i < 0$). Label the remaining $(100 - k)$% of reviewers as neutral ($l_i = 0$).
(3) *Assign reviewers to objects.* Randomly assign n reviewers to every object. On average, there would be n objects per reviewer, but the actual number may vary among reviewers.
(4) *Generate rating scores.* The rating score e_{ij} is generated as follows. If the reviewer is lenient (or strict), we assign a random value higher (or lower) than q_j to e_{ij}, while still in the range of [0.2, 1.0]. Otherwise, $e_{ij} = q_j$. Note that there is no presupposition of the value of α.

Metric. Given the generated rating scores (without the predetermined leniency and quality values), each comparative solution computes the quality values and attempts to reconstruct the predetermined quality ranking. To measure the solution's performance, as a metric, we measure the Kendall similarity between the solution's ranked list with the predetermined "true" ranked list. A similarity value of 100% indicates a complete agreement with the ground truth (best performance).

In each of the subsequent experiments, we vary one parameter (n, k, m, α), and keep the others fixed at the default values shown in Table IX. For each parameter setting, we average the Kendall similarity values over 25 independently generated synthetic datasets. Each dataset has 1,000 reviewers and 1,000 objects. We have conducted separate experiments with larger number of reviewers/objects with similar results. We only use datasets where each object has at least three reviewers and each reviewer has at least three objects.

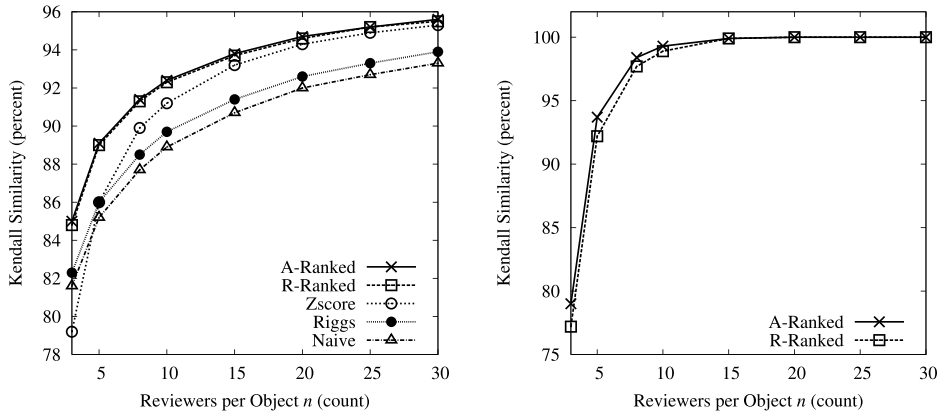
6.2. Varying Number of Reviewers per Object *n*

Here we study how the number of reviewers assigned to each object n affects the Kendall similarity values.

Kendall Similarity. Figure 6(a) plots the Kendall similarity for quality at different values of n. It shows that performance generally increases with n, and that A-Ranked and R-Ranked generally outperform Zscore, Riggs, and Naive. These observations can be explained as follows.

The random assignment of reviewers to objects means that objects may have different compositions of reviewers, in terms of leniency.

— For small n, there is a higher probability for an object to be assigned mainly lenient (or strict) reviewers, which would highly distort its rating scores. However, LQ solutions take the leniency of each reviewer into account, better compensating for the distortion, resulting in higher Kendall similarity. Zscore especially suffers at very low values of n, as Zscore's normalization may be incorrect for objects having all or mostly lenient (or strict) reviewers.
— As n increases, statistically the assignment gets more even and more objects will share a similar composition of reviewers, which is the underlying distribution of reviewers in terms of leniency. As a result, the variance due to reviewers' leniency



(a) Quality. (b) Leniency.

Fig. 6. Vary n: Kendall similarity.

will become less important, as most objects are affected similarly. It becomes easier to separate the two classes of objects, and all solutions move toward higher Kendall similarity.

Figure 6 plots Kendall similarity for leniency for A-Ranked and R-Ranked, which shows the same trend of increasing similarity with n. In general, this is expected, as leniency and quality are mutually dependent. Better leniency performance leads to better quality performance (and vice versa). Hence, in most cases and as in the subsequent experiments, showing only the Kendall similarity for quality is sufficient.

Distribution of Leniency Classes among an Object's Reviewers. To show that as n increases, more objects will get a similar composition of reviewers, we look at how the distribution of leniency classes among an object's reviewers changes with n.

Each object o_j has a distribution vector $\mathbf{d}_j = [d_-, d_0, \text{and } d_+]$, where $\mathbf{d}_j.d_-$, $\mathbf{d}_j.d_0$, $\mathbf{d}_j.d_+$ are the percentages of o_j's reviewers who are strict, neutral, and lenient (according to ground truth), respectively. Based on the input parameters in Table IX, the expected distribution vector is $\mathbf{d}_J = [30\%, 40\%, 30\%]$. However, due to the random assignment of reviewers to objects, $\mathbf{d}_j$ may deviate from $\mathbf{d}_J$. The *distribution error* of an object o_j is defined as the Euclidean distance from the actual distribution $\mathbf{d}_j$ to the expected distribution $\mathbf{d}_J$, (as shown in Equation (21)).

$$dist(\mathbf{d}_j, \mathbf{d}_J) = \sqrt{(\mathbf{d}_j.d_- - \mathbf{d}_J.d_-)^2 + (\mathbf{d}_j.d_0 - \mathbf{d}_J.d_0)^2 + (\mathbf{d}_j.d_+ - \mathbf{d}_J.d_+)^2}. \quad (21)$$

Figure 7 plots the median and the standard deviation of the distribution errors ($dist(\mathbf{d}_j, \mathbf{d}_J)$ values) across all o_j's. As n increases, both median and standard deviation decrease. All objects uniformly approach the expected distribution $\mathbf{d}_J$.

6.3. Varying Proportion of Nonneutral Classes k

Here we study how the proportion of nonneutral reviewers k affects the Kendall similarity for quality. We expect that with more lenient or strict reviewers in the system, there will be greater distortion in rating scores, which results in more difficulties in separating the two classes of objects.

Figure 8(a) plots Kendall similarity for quality for different values of k. It shows that similarity decreases with k, and that A-Ranked and R-Ranked again generally

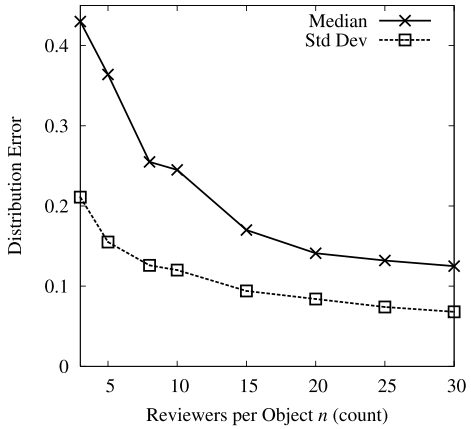
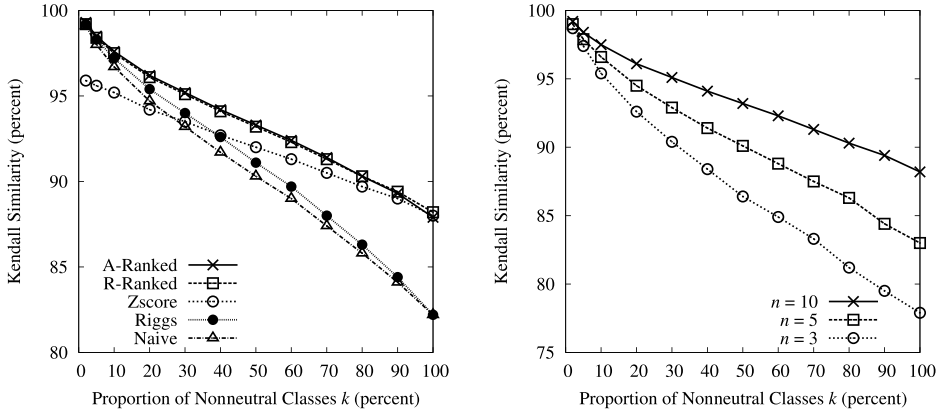


Fig. 7. Vary n: Leniency-class distribution error.



(a) Vary k. (b) Vary k, n for R-Ranked.

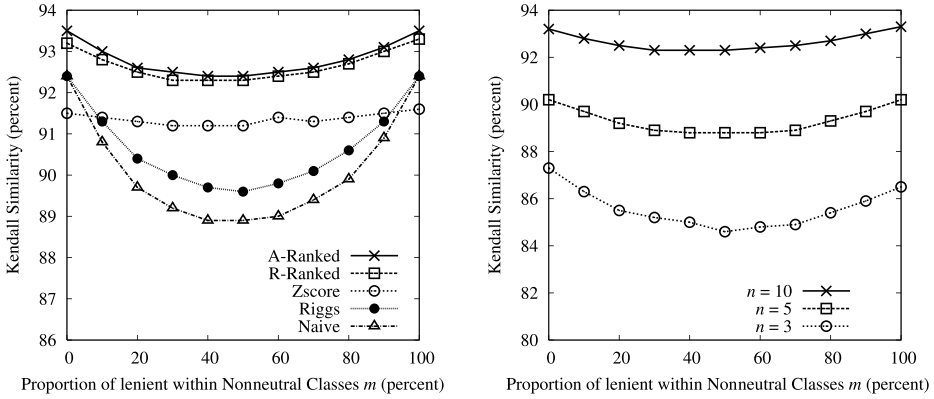
Fig. 8. Vary k: Kendall similarity for quality.

outperform Zscore, Riggs, and Naive. For R-Ranked, Figure 8(b) shows quality similarity curves at different values of n. The same trend of decreasing similarity applies to other values of n, only with even lower similarity values for lower n.

The reason for decreasing similarity with increasing k is the greater likelihood for an object to get an imbalanced distribution of reviewers. Table X shows, for varying k, the percentage of objects assigned mainly strict reviewers ($\mathbf{d}_j.d_- \geq 60\%$, 70%, or 80%) and the percentage of objects assigned mainly lenient reviewers ($\mathbf{d}_j.d_+ \geq 60\%$, 70%, or 80%). It shows that the percentages of both types of objects are higher when k is higher. At $k = 10\%$, there is no object with $\mathbf{d}_j.d_- \geq 60\%$ or $\mathbf{d}_j.d_+ \geq 60\%$. At $k = 90\%$, 27% of objects have $\mathbf{d}_j.d_- \geq 60\%$ and 26% of objects have $\mathbf{d}_j.d_+ \geq 60\%$. Since a ranking mistake occurs when a lower-quality object with mainly lenient reviewers is confused with a higher-quality object with mainly strict reviewers, it follows that as the number of objects with imbalanced distribution of reviewers rises, the rate of making ranking mistakes also increases (Kendall similarity decreases).

Table X. Vary k: Percentages of Objects with High $\mathbf{d}_j.d_-$ or $\mathbf{d}_j.d_+$

k	$\mathbf{d}_j.d_-$			$\mathbf{d}_j.d_+$		
	$\geq 60\%$	$\geq 70\%$	$\geq 80\%$	$\geq 60\%$	$\geq 70\%$	$\geq 80\%$
10%	0	0	0	0	0	0
30%	0	0	0	0	0	0
50%	2	0	0	2	0	0
70%	9	3	0	9	3	1
90%	27	10	3	26	10	3



(a) Vary m. (b) Vary m, n for R-Ranked.

Fig. 9. Vary m: Kendall similarity for quality.

6.4. Varying Proportion of Lenient within Nonneutral Classes

Experiments in previous sections involve a 50:50 balance between lenient and strict reviewers. In this section, we study how varying m, or the proportion of lenient within nonneutral reviewers, affects Kendall similarity for quality.

Figure 9(a) plots Kendall similarity for quality for different values of m. Again, A-Ranked and R-Ranked outperform Zscore, Riggs, and Naive. As m increases, Kendall similarity initially decreases, reaches a trough in the $40\% \leq m \leq 60\%$ range, and then increases. Figure 9(b) shows quality similarity curves for R-Ranked at different values of n. It shows a similar trend, except at lower similarity values for lower n.

To see why similarity is lowest in the $40\% \leq m \leq 60\%$ range, we again compare the percentage of objects assigned mainly strict reviewers with the percentage of objects assigned mainly lenient reviewers. Table XI shows that the percentage of objects with mainly strict reviewers and the percentage of objects with mainly lenient reviewers are inversely related. When m is low, the former is high, but the latter is low. Since misclassification requires the confusion of a low-quality object with mainly lenient reviewers with a high-quality object with mainly strict reviewers, it follows that misclassification rate is highest (similarity is lowest) when there is a sizeable number of both types of objects, which is when m is within the 40% to 60% range.

6.5. Varying Compensation Factor α

Previous experiments in this section use the setting $\alpha = 0.7$. Here, we study the effect of the compensation factor α on Kendall similarity for quality. Figure 10 plots Kendall similarity at different values of α. For A-Ranked and R-Ranked, similarity

Table XI. Vary m: Percentages of Objects with High $\mathbf{d}_j.d_-$ or $\mathbf{d}_j.d_+$

m	$\mathbf{d}_j.d_-$			$\mathbf{d}_j.d_+$		
	$\geq 60\%$	$\geq 70\%$	$\geq 80\%$	$\geq 60\%$	$\geq 70\%$	$\geq 80\%$
10%	48	25	9	0	0	0
30%	20	7	2	0	0	0
50%	4	1	0	5	1	0
70%	0	0	0	21	7	2
90%	0	0	0	48	24	9

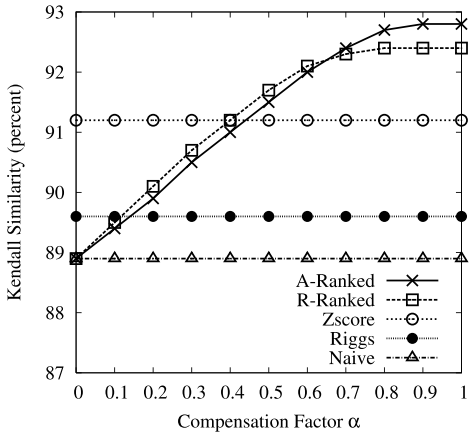


Fig. 10. Vary α.

increases as α approaches 1. For $\alpha \geq 0.4$, A-Ranked and R-Ranked outperform all other comparative solutions.

In summary, our experiments on synthetic datasets show that the proposed LQ model is more effective at reconstructing the ground truth and consistently outperforms the comparative Zscore, Riggs, and Naive models. LQ is especially effective when each object is assigned a small number of reviewers. One factor that contributes to its efficacy is its robustness in handling objects assigned predominantly lenient (or strict) reviewers.

7. CONCLUSION

In this article, we address the score-summarization problem of how to aggregate the rating scores given to an object to arrive at an overall score that reflects the object's quality. Our approach is premised on mining the leniency behavior of reviewers from the rating scores and using the leniency information to adjust the quality scores correspondingly. We propose the *Leniency-aware Quality* (*LQ*) model, which determines leniency and quality simultaneously.

We further show that the LQ model is better than Naive which relies on simple averaging; Riggs, which weighs reviewer's scores by reputation; and Zscore, which seeks to normalize the reviewers' rating scales. Experiments on real-life datasets shows that LQ results are different from the comparative methods and have less error in the rating-prediction task than the comparative methods. Experiments on synthetic datasets show that LQ consistently achieves a higher performance than the comparative methods.

Several avenues exist for future work. The effectiveness of the LQ model can be further verified by integrating it into various social media applications and allowing users to evaluate the new quality ranking. The problem addressed here also touches upon aspects beyond computer science. It would also be interesting to verify which of the two proposed compensation modes (Relative and Absolute) is more consistent with the psychology of reviewers, as studied in behavioral sciences.

REFERENCES

ADOMAVICIUS, G. AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17,* 6, 734–749.

ANTON, H. AND RORRES, C. 1987. *Elementary Linear Algebra with Applications*. John Wiley & Sons, Hoboken, NJ.

ARAMPATZIS, A. AND KAMPS, J. 2009. A signal-to-noise approach to score normalization. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*. ACM, New York, NY, 797–806.

ARKES, H. R. 2003. The nonuse of psychological research at two federal agencies. *Psychol. Science 14,* 1, 1–6.

BACKSTROM, L., DWORK, C., AND KLEINBERG, J. 2007. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International World Wide Web Conference*. ACM, New York, 181–190.

BAZERMAN, M. H. 1990. *Judgment in Managerial Decision-Making* 2nd Ed. Wiley, Hoboken, NJ.

BELL, R. M. AND KOREN, Y. 2007. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of the 7th IEEE International Conference on Data Mining*. 43–52.

BHATTACHARJEE, R. AND GOEL, A. 2005. Avoiding ballot stuffing in Ebay-like reputation systems. In *Proceeding of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*. ACM, New York, 133–137.

BLACKBURN, J. L. AND HAKEL, M. D. 2006. An examination of sources of peer-review bias. *Psychol. Science 17,* 5, 378–382.

BORGATTI, S. P. AND EVERETT, M. G. 1997. Network analysis of 2-mode data. *Social Netw. 19,* 3, 243–269.

BORODIN, A., ROBERTS, G. O., ROSENTHAL, J. S., AND TSAPARAS, P. 2005. Link analysis ranking: Algorithms, theory, and experiments. *ACM Trans. Internet Technol. 5,* 1, 231–297.

BUSENITZ, L. AND LAU, C. 1996. A cross-cultural cognitive model of new venture creation. *Entrepreneurship: Theory Pract. 20,* 4, 25–39.

CHEN, M. AND SINGH, J. P. 2001. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*. ACM, New York, 154–162.

DELLAROCAS, C. 2000. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*. ACM, New York, 150–157.

DUMAIS, S. AND NIELSEN, J. 1992. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, 233–244.

DWORK, C., KUMAR, R., NAOR, M., AND SIVAKUMAR, D. 2001. Rank aggregation methods for the Web. In *Proceedings of the 10th International World Wide Web Conference*. 613–622.

FAGIN, R., KUMAR, R., AND SIVAKUMAR, D. 2003. Comparing top k lists. *SIAM J. Discrete Math. 17,* 1, 134–160.

FAUST, K. 1997. Centrality in affiliation networks. *Social Netw. 19,* 2, 157–191.

FERNANDEZ, M., VALLET, D., AND CASTELLS, P. 2006. Probabilistic score normalization for rank aggregation. In *Proceedings of the European Conference on Information Retrieval*. 553–556.

FIGUEIRA, J., GRECO, S., AND EHRGOTT, M., Eds. 2005. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer Science and Business Media, Inc.

GELLER, J. AND SCHERL, R. 1997. Challenge: Technology for automated reviewer selection. In *Proceedings of the International Joint Conferences on Artificial Intelligence*. 55–61.

GOLBECK, J. AND HENDLER, J. 2006. Inferring binary trust relationships in Web-based social networks. *ACM Trans. Internet Technol. 6,* 4, 497–529.

GOLUB, G. H. AND VAN LOAN, C. F. 1996. *Matrix Computations* 3rd Ed. Johns Hopkins University Press.

GRIMMETT, G. R. AND STIRZAKER, D. R. 1982. *Probability and Random Processes*. Oxford University Press, Oxford, UK.

GUHA, R., KUMAR, R., RAGHAVAN, P., AND TOMKINS, A. 2004. Propagation of trust and distrust. In *Proceedings of the International World Wide Web Conference*. ACM, New York.

HAVELIWALA, T. H. 2003. Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search. *IEEE Trans. Knowl. Data Engin. 15,* 4, 784–796.

HERLOCKER, J. L., KONSTAN, J. A., AND RIEDL, J. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. ACM, New York, NY, 241–250.

HERLOCKER, J., KONSTAN, J. A., AND RIEDL, J. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval 5*, 287–310.

HETTICH, S. AND PAZZANI, M. J. 2006. Mining for proposal reviewers: Lessons learned at the national science foundation. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 862–871.

JIN, R. AND SI, L. 2004. A study of methods for normalizing user ratings in collaborative filtering. In *Proceedings of the 27th ACM SIGIR Conference on Research and Development in Information Retrieval*. 568–569.

JIN, R., SI, L., ZHAI, C.-X., AND CALLAN, J. 2003. Collaborative filtering with decoupled models for preferences and ratings. In *Proceedings of the International Conference on Information and Knowledge Management*. ACM, New York, NY, 309–316.

JINDAL, N. AND LIU, B. 2007. Analyzing and detecting review spam. In *Proceedings of the IEEE International Conference on Data Mining*. IEEE Computer Society, Los, Alamitos, CA.

KLEINBERG, J. M. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46,* 5, 604–632.

KORHONEN, P., MOSKOWITZ, H., AND WALLENIUS, J. 1992. Multiple criteria decision support—a review. *Euro. J. Operational Res. 63,* 3, 361–375.

LAM, S. K. AND RIEDL, J. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th International World Wide Web Conference*. ACM, New York, 393–402.

LAUW, H. W., LIM, E.-P., AND WANG, K. 2007. Summarizing review scores of "unequal" reviewers. In *Proceedings of the 2007 SIAM International Conference on Data Mining*.

LEMIRE, D. 2005. Scale and translation invariant collaborative filtering systems. *Info. Retrieval 8,* 1, 129–150.

MA, H., YANG, H., LYU, M., AND KING, I. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*. ACM, New York, NY, 931–940.

MA, H., KING, I., AND LYU, M. R. 2009. Learning to recommend with social trust ensemble. In *SIGIR*. ACM, New York, NY, 203–210.

MANMATHA, R. AND SEVER, H. 2002. A formal approach to score normalization for meta-search. In *Proceedings of the 2nd International Conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 98–103.

MASSA, P. AND AVESANI, P. 2005. Controversial users demand local trust metrics: An experimental study on epinions.com community. In *AAAI*. AAAI Press, 121–126.

MOBASHER, B., BURKE, R., AND SANDVIG, J. 2006. Model-based collaborative filtering as a defense against profile injection attacks. In *Proceedings of the National Conference on Artificial Intelligence*.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The PageRank citation ranking: Bringing order to the Web. Stanford Digital Library Technologies Project.

RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., AND RIEDL, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 175–186.

RIGGS, T. AND WILENSKY, R. 2001. An algorithm for automated rating of reviewers. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, New York, 381–387.

SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. 2000. Application of dimensionality reduction in recommender system – a case study. In *Proceedings of the ACM WebKDD Web Mining for E-Commerce Workshop*.

SHEN, J., LIN, Y., XUE, G.-R., ZHU, F.-D., AND YAO, A.-G. 2006. IRFCF: Iterative rating filling collaborative filtering algorithm. In *Proceedings of the 8th Asia-Pacific Web Conference*. 862–867.

SIMON, M., HOUGHTON, S. M., AND AQUINO, K. 1999. Cognitive biases, risk perception, and venture formation: How individuals decide to start companies. *J. Bus. Venturing 15*, 113–134.

TANTIPATHANANANDH, C., BERGER-WOLF, T., AND KEMPE, D. 2007. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 717–726.

TRAUPMAN, J. AND WILENSKY, R. 2004a. Collaborative quality filtering: Establishing consensus or recovering ground truth? In *Proceedings of the KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

TRAUPMAN, J. AND WILENSKY, R. 2004b. Collaborative quality filtering: Establishing consensus or recovering ground truth? In *Proceedings of the KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 73–86.

WALPOLE, R. E., MYERS, R. H., MYERS, S. L., AND YE, K. 2002. *Probability & Statistics for Engineers & Scientists* 7th Ed. Prentice Hall, NJ.

WASSERMAN, S. AND FAUST, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK.

YANG, B., CHEUNG, W., AND LIU, J. 2007. Community mining from signed social networks. *IEEE Trans. Knowl. Data Engin. 19,* 10, 1333–1348.

ZHANG, J. AND COHEN, R. 2006. Trusting advice from other buyers in e-marketplaces: The problem of unfair ratings. In *Proceedings of the 8th International Conference on Electronic Commerce: The new e-Commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet*. ACM, New York, 225–234.

ZHOU, D., MANAVOGLU, E., LI, J., GILES, C. L., AND ZHA, H. 2006. Probabilistic models for discovering e-communities. In *Proceedings of the 15th International World Wide Web Conference*. ACM, New York, 173–182.

Received March 2009; revised July 2009, July 2010, March 2011; accepted October 2011