Accumulative Time Based Ranking Method to Reputation Evaluation in Information Networks

Hao Liao$^{1,2,3,4}$ (廖好), Senior Member, CCF, Qi-Xin Liu$^{1,2,3}$ (刘启鑫), Ze-Cheng Huang$^{1,2,3}$ (黄泽成)
Ke-Zhong Lu$^{1,2,3,*}$ (陆克中), Chi Ho Yeung$^{5,*}$ (杨志豪), and Yi-Cheng Zhang$^6$ (张冀成)

$^1$National Engineering Laboratory on Big Data System Computing Technology, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
$^2$Guangdong Province Key Laboratory of Popular High Performance Computers, Shenzhen University, Shenzhen 518060, China
$^3$Guangdong Province Engineering Center of China-Made High Performance Data Computing System, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
$^4$Institute of Big Data Intelligent Management and Decision, Shenzhen University, Shenzhen 518060, China
$^5$Department of Science and Environmental Studies, The Education University of Hong Kong, Hong Kong 999077, China
$^6$Department of Physics, University of Fribourg, Fribourg 1700, Switzerland

E-mail: haoliao@szu.edu.cn; {1810273008, 2140230417}@email.szu.edu.cn; kzlu@szu.edu.cn; chyeung@eduhk.hk; yi-cheng.zhang@unifr.ch

Received March 27, 2020; accepted December 3, 2021.

Abstract Due to over-abundant information on the Web, information filtering becomes a key task for online users to obtain relevant suggestions and how to extract the most related item is always a key topic for researchers in various fields. In this paper, we adopt tools used to analyze complex networks to evaluate user reputation and item quality. In our proposed Accumulative Time Based Ranking (ATR) algorithm, we take into account the growth record of the network to identify the evolution of the reputation of users and the quality of items, by incorporating two behavior weighting factors which can capture the hidden facts on reputation and quality dynamics for each user and item respectively. Our proposed ATR algorithm mainly combines the iterative approach to rank user reputation and item quality with temporal dependence compared with other reputation evaluation methods. We show that our algorithm outperforms other benchmark ranking algorithms in terms of precision and robustness on empirical datasets from various online retailers and the citation datasets among research publications. Therefore, our proposed method has the capability to effectively evaluate user reputation and item quality.

Keywords temporal network, behavior dynamics, reputation evaluation, ranking algorithm

1 Introduction

As online information networks are becoming more popular, people are getting more information from the Web. Nevertheless, the Web may not play an important role if one cannot extract the relatively small amount of useful results out of a large pool of complicated information. Therefore, how to effectively filter information has become an increasingly important task in the online science community$^{[1-5]}$. In this case, one may rely on the information acquired through interactions with strangers, which is useful for users and sometimes
even profitable for the hosts. For instance, since we cannot guarantee the products in online retailers such as flikart.com and Amazon are of high quality as they claim, one can rely on the most conventional recommendation mechanism, i.e., the word of mouth; instead of face-to-face interaction, it now takes the form of an online reputation system in the cyberspace [6]. The information used by the recommendation mechanism is the interactive information between users and items, and the most suitable item is recommended to the other users based on this information. The Reputation mechanism provides a criterion to the problem of trust-building [7]. These online reputation systems are effective virtual word-of-mouth networks where users share opinions and scores on products such as companies, hotels, video contents and more [8]. As a result, many online commercial websites such as eBay, Amazon, and Netflix have introduced the rating system [9,10] for users to evaluate items, articulating accumulative scores to reflect the quality of the products [11,12].

In view of the high commercial values, corporations pay attention to their reputation and the reputation of their products on the Web and social networks. Reputation is thus considered as one of the most important strategic resources [13-16]. A good reputation can improve corporate branding and its position in the market [17-19]. On the other hand, a bad reputation affects their business. Other than reputation, corporations also realize that online conversations and posts embed a continuous stream of valuable information [20,21]. Companies that effectively analyze this information may obtain clues for innovation and improvement. Therefore, it is beneficial for corporations to track and improve their reputations on online platforms [22-24].

The quality of an item is closely related to the reputation of users. For example, some malicious users often give low scores and the water army often gets high scores. In the news [1] in 2018, the reporters of Securities Daily interviewed some water army teams, and most of the teams claimed to publicize movies by making fake ratings. These water army teams were hired at the early stage in the release of a movie to give high ratings or low malicious ratings to other competing movies in the same period. These malicious behaviors have seriously interfered the judgment of the general audience on the movies, which may lead to a direct increase of box office of the promoted movie, or a decline of watching times of the competing movies due to malicious bad ratings [25-27]. Similarly, malicious ratings of products in online retailers also have the same influence on online consumers. Hence, an accurate evaluation of user reputation in rating systems is valuable for all online platforms where ratings of products influence consumption. Although the benefits gained from a good reputation are well known, how to accurately quantify reputation is uncertain. The most straightforward way is to consider the average rating by users. However, this method is sensitive to noisy information and malicious operations, and various alternative approaches are proposed. For instance, with BiRank [28], one iteratively assigns scores to vertices on a bipartite graph of users and products and finally gets a stationary rank among items. Nevertheless, some users may give unreasonable scores because they consider that ratings are unimportant, or they are not familiar with the field of the items.

Other than average rating, another representative method is the iterative refinement (IR) method [29]. In this method, user reputation is considered to be inversely proportional to the difference between his/her rating on an item and its estimated quality, i.e., the weighted average score of the item by all users. Item quality and user reputation are then iteratively updated until they become convergent. In addition, a new iterative optimization algorithm called Correlation-Based Ranking (CR) algorithm [30] is obtained from optimizing IR by assigning a single reputation value for each scoring event, where user reputation is calculated by Pearson correlation between his/her assigned scores and the estimated quality of items [31,32]. This method is shown to be very effective in dealing with different malicious behaviors of spammers. In order to better cope with the malicious behavior and to enhance robustness, a reputation redistribution process is used to improve the reputation of well-known users and two penalty factors are applied to make the algorithm robust against the malicious behavior, which is called the Iterative Algorithm with Reputation Redistribution (IARR) [26]. Nevertheless, reputation and quality in IARR do not consider temporal dependency and IARR does not take into account the history of users and items during evaluation. Thus, we aim to propose a novel method to perform better.

In this paper, based on the findings of related iterative algorithms, we introduce the Accumulative Time Based Ranking (ATR) algorithm, with two behavior

\[ \text{https://tech.huanqiu.com/article/9CaKrnK8l8o, May 2021.} \]
weight factors (user and item weight) and two new iterative components introduced based on the time of the scoring events and the corresponding user and item degree at that time. The behavior weight factors can significantly up-weigh the importance of a user in a certain period of time, and we will show that the proposed two-iterative-component accumulation process can effectively evaluate user reputation and item quality.

2 Baseline Methods

2.1 Average Score

In online retailers, the rating on a product is usually calculated by averaging its score evaluated by users who have experience with the product. It is continuously updated when new scoring events occur. Future potential buyers may favor products with higher average scores. Mathematically, the average score \( \overline{r}_\alpha \) of item \( \alpha \) is given as follows:

\[
\overline{r}_\alpha = \frac{\sum_{i \in \mathcal{U}_\alpha} r_{i\alpha}}{k_\alpha},
\]

where \( \mathcal{U}_\alpha \) is the set of users who rated item \( \alpha \), \( r_{i\alpha} \) denotes the rating of user \( i \) on item \( \alpha \), and \( k_\alpha \) is the number of users who have collected item \( \alpha \). The average score \( \overline{r}_\alpha \) is used as the basis for ranking items. However, the average score has a large defect. The popularity of an item may fluctuate over time, and malicious rating by spammers may greatly influence the average score.

2.2 Iterative Refinement

The iterative refinement (IR) algorithm [29] considers user reputation as inversely proportional to the average squared error between his/her degree and the vector of item rating averaged over users. Nevertheless, the correlation-based ranking (CR) algorithm [30] to be introduced below is shown to be more robust against spamming ratings than the IR method and thus leads to a more accurate estimation of item quality.

2.3 Correlation-Based Ranking

In the CR iterative algorithm proposed by Zhou et al. [30], CR evaluates the quality \( Q_\alpha \) of an item \( \alpha \) as:

\[
Q_\alpha = \frac{\sum_{i \in \mathcal{U}_\alpha} R_i \times r_{i\alpha}}{\sum_{i \in \mathcal{U}_\alpha} R_i},
\]

where \( \mathcal{U}_\alpha \) is item \( \alpha \) rated by the set of users, \( R_i \) is the reputation of user \( i \) and \( r_{i\alpha} \) is the rating of user \( i \) on item \( \alpha \). Every user’s reputation is initially assigned according to his/her degree as \( R_i = k_i / |\mathcal{O}| \), where \( k_i \) is the degree of user \( i \), i.e., the number of items collected by user \( i \), and \( |\mathcal{O}| \) represents the total number of items.

The algorithm calculates the correlation \( corr_i \) of user \( i \) between his/her assigned scores and the quality of items as follows:

\[
corr_i = \frac{1}{k_i} \sum_{\alpha \in \mathcal{N}_i} \left( \frac{r_{i\alpha} - \overline{r}_\alpha}{\sigma_i} \right) \left( \frac{Q_{i\alpha} - \overline{Q}_\alpha}{\sigma_\alpha} \right),
\]

where \( k_i \) is the degree of user \( i \), \( \mathcal{N}_i \) is the set of items collected by user \( i \), \( r_{i\alpha} \) is the score rated by user \( i \) on item \( \alpha \), and \( \overline{r}_\alpha \) and \( \sigma_i \) are the average rating by user \( i \) and the standard deviation among ratings given by user \( i \), respectively; similarly, \( Q_{i\alpha} \) is the quality of item \( \alpha \) evaluated by user \( i \), \( \overline{Q}_\alpha \) is the average quality of item \( \alpha \), and \( \sigma_\alpha \) is the standard deviation of the quality of item \( \alpha \) evaluated by all collector users. The reputation of users is then assigned according to the value of \( corr_i \); when \( corr_i < 0 \), the reputation of the user \( R_i \) is 0. Conversely, if \( corr_i \geq 0 \), the reputation of user \( i \) is \( R_i = corr_i \). In [30], the ranking algorithm based on the Pearson correlation coefficient is proved to be more powerful in dealing with spammers and able to evaluate the quality of objects more accurately.

2.4 BiRank

The BiRank algorithm [28] ranks items on user-item bipartite graphs through iterations until convergence. BiRank is analyzed by graph regularization, and a complementary Bayesian view is proposed [28]. Firstly, the ranking vector is randomly initialized. Then, the iteration process is executed until convergence. During the iteration process, the BiRank algorithm utilizes query vectors, i.e., sparse vectors with the entries for the collected items to be non-zero, thus representing vectors according to his/her degree as \( R_i = k_i / |\mathcal{O}| \), where \( k_i \) is the degree of user \( i \), i.e., the number of items collected by user \( i \), and \( |\mathcal{O}| \) represents the total number of items.

The iteration can be expressed as follows:

\[
v = \alpha S^T u + (1 - \alpha)v^0,
\]
\[
u = \beta Sv + (1 - \beta)u^0,
\]

where \( v^0 \) and \( u^0 \) represent query vectors, and \( \alpha \) and \( \beta \) are hyper-parameters which weigh the importance of the previous query vectors in the last iteration. After convergence of the query vectors, \( v \) and \( u \) become the vectors for ranking items. Besides, one can express

\[
S = D_g^{-\frac{1}{2}} W D_v^{-\frac{1}{2}},
\]

where \( D_g \) and \( D_v \) are diagonal matrices and represent the weighted degrees of all vertices.
in $U$ and $V$ respectively, i.e., the sum of weights on connected edges with $U$ and $V$ to be the set of users and items respectively. As such, $W$ is the matrix of which entries are edge weights of the graph as $|U| \times |V|$ [28].

### 2.5 KS (Kolmogorov Similarity)

The method proposed by [8] is a new reputation-based ranking algorithm, utilizing multipartite rating subnetworks, which groups users by their similarities such as Kolmogorov similarity.

In [8], let $I_{u,v} = I_u \cap I_v$ indicate the set of items that both users $u$ and $v$ rated. In order to cluster different users, we define the Kolmogorov similarity between user $u$ and user $v$ as: $KS(u, v) = 0$ if $I_{u,v} = \emptyset$, and otherwise

$$KS(u, v) = \frac{1}{1 + |C'(u) - C'(v)|},$$

where $C$ is a compressor, and $C(x)$ denotes the length of the string $x$. In this paper, we use the byte compressor of zlib library in Python[2]. And for user $u$, we denote by $\tilde{u}$ the string composed by the concatenation of the pairs (item, rating) of her/his rated items.

Given a specified threshold $\alpha$, the method groups users into different groups such that the Kolmogorov similarity between user $u$ and user $v$ in same group $KS(u, v) > \alpha$, which splits a multipartite graph into several subgraphs, and we set $\alpha = 0.8$ in our experiments. Lastly, it iteratively updates both the reputation of users and the rating of items in different subgraphs. More details about KS can be found in [8].

### 2.6 Iterative Algorithm with Reputation Redistribution

Based on the CR algorithm [30], Liao et al. [26] proposed the Iterative Algorithm with Reputation Redistribution (IARR). IARR uses to redistribute user reputation and obtain a new reputation $R_i$ for user $i$. IARR is based on the CR algorithm but uses reputation redistribution to filter noisy information during the iterations, which improve the accuracy in items’ quality ranking [26]. $TR_i$ in IARR is the same as $corr_i$ in CR, representing the similarity between ratings by users and the weighted average rating. Differently, the reputation $R_i$ of user $i$ in IARR is not equal to $TR_i$ but is calculated as follows:

$$TR_i = \frac{1}{k_i} \sum_{a \in C_i} \left( \frac{r_i - \bar{r}_i}{\sigma_i} \right) \left( \frac{Q_{ia} - \bar{Q}_a}{\sigma_a} \right),$$

where $\varphi$ is a parameter of the algorithm, and the reputation of users is calculated by adjusting the value of the parameter. The purpose to redistribute user reputation by the IARR algorithm is to further up-weigh users with high reputation, since evaluation by users with high-reputation is more reliable, and can reduce the interference by malicious user behaviors. During the iterations, it can filter the noise by diminishing the weight of users with low $TR_i$. With the accumulation in each iteration, the estimation of item quality would get a large improvement.

### 3 Methods and Settings

#### 3.1 Bipartite Networks

The bipartite network is a mathematical representation of real-world relations. It is named bipartite, as there are two types of nodes. For instance, if we want to represent the relationship between users and products in an online shop, one type of nodes would be users, while the other would be products. A link between a user and a product could mean that the user has bought the product. Similarly, networks can be created for users and posts in online social media, or users and music on music websites. Mathematically, the network $G = (E, V)$ contains the set of nodes $V$ and the set of edges connecting the nodes $E$. If the set $V$ of nodes can be divided into two subsets $X$ and $Y$ with no connections within the subset, the network is said to be bipartite.

Most social networks can be represented as bipartite networks. However, it is not always the case. For instance, the citation networks of scientific papers are usually studied as a relationship between papers, rather than between the scientists who write the papers. One can employ recommendation algorithms conventionally applied on bipartite networks to represent the complex citation networks between papers, though one cannot identify the subset of papers and the subset of their references as every publication can be cited in the future [33,34]. The algorithms developed for bipartite networks can be applied on monopartite networks by defining a set $A$, and then defining a set $B$, identical to set $A$. In the citation network, an edge between sets $A$ and $B$ corresponds to a citation from a paper in set $A$ to a paper in set $B$, which is a peer-to-peer network.

[2]https://pypi.org/project/python2.7(zlib)/, May 2021.
3.2 Accumulative Time Based Ranking

In this paper, we develop an approach to analyze the interactions between users in social networks, and produce a ranking of users and products. This type of rankings can be used, for instance, to make recommendations or to predict the Oscar nominations.

Compared with the existing reputation evaluation algorithms, our proposed ATR method focuses on combining the iterative approach to rank user reputation and item quality with the temporal dependence. The main idea of this algorithm is to take into account the whole history of the network to compute the reputation of users and the quality of items. Similar to previous methods, the ratings by users with higher reputation weigh more in the determination of the quality of items.

We denote the quality of item $\alpha$ by $Q_{\alpha}$, and the reputation of user $i$ by $R_{i}$. The iterative process of the ATR algorithm uses the aggregated reputation of users and the aggregated quality of items to improve their estimation of reputation and quality. Consequently, user reputation and item quality are continuously updated in the iterative process until convergence. The notations of the algorithm are summarized in Table 1.

We use the following initial configuration for better convergence in the iterative procedures. The first assumption in our algorithm is that the initial reputation of users and the initial quality of items are dependent on the number of ratings they assign or receive. The ratings of users and items with higher weights have a larger influence in the user-item evaluation.

On the other hand, reputation and quality in IARR do not consider temporal dependency and are vulnerable to malicious behaviors. Such approaches do not take into account the history of users and items during evaluation. For instance, users who are active over multiple years are less likely to be spammers. In the case of ATR, we thus define the weight of a user by (1) and the weight of an item by (2):

$$ W_{i}(t) = \frac{k_{i}(t)}{N(t)}, \quad (1) $$

$$ \tilde{W}_{\alpha}(t) = \frac{\tilde{k}_{\alpha}(t)}{N(t)}, \quad (2) $$

where $N(t)$ is the total number of user ratings in year $t$.

| Notation | Description |
|----------|-------------|
| $U$      | Set of users |
| $|U|$     | Number of all users |
| $O$      | Set of items (such as music, movies) |
| $|O|$     | Number of all items |
| $U_{\alpha}(t)$ | Set of users who evaluated item $\alpha$ in year $t$ |
| $O_{i}(t)$ | Set of items evaluated by user $i$ in year $t$ |
| $U(t)$   | Set of users in year $t$ |
| $O(t)$   | Set of items in year $t$ |

$r_{i\alpha}$ Average rating of items given by user $i$

$\sigma_{i}$ Standard deviation of ratings given by user $i$

$k_{i}(t)$ Degree of user $i$ in year $t$

$\tilde{k}_{\alpha}(t)$ Degree of item $\alpha$ in year $t$

$Q_{\alpha}(t)$ Quality of item $\alpha$ in year $t$

$Q_{\alpha}(t)_{\text{avg}}$ Average quality value of the item $\alpha$ in year $t$

$R_{i}(t)$ Reputation of user $i$ in year $t$

$W_{i}(t)$ Weight of user $i$ in year $t$

$\mathcal{Y}$ Set of year in each individual dataset

$\tilde{W}_{\alpha}(t)$ Weight of item $\alpha$ in year $t$

$|R|$ Number of all ratings

$\kappa_{U}$ Average degree of all users

$\kappa_{O}$ Average degree of all items

$M$ Matching numbers of external award

$f$ Percentage of the top item quality list

$N_{\text{cum}}$ Number of samples

$AUC_{\text{real}}$ AUC on real data

$AUC_{\text{ran}}$ AUC on the malicious ratings data

$n$ Number of users assigning random ratings

$P_{\alpha}(t)$ Potential influence on item $\alpha$ by user $i$ in year $t$

The rating of user $i$ on item $\alpha$ is expressed as $r_{i\alpha}$. The reputation of users and the quality of items are initialized as:

$$ R_{i} = \frac{\sum_{\alpha \in O_{i}(t), t \in \mathcal{Y}} r_{i\alpha} \times W_{i}(t)}{\sum_{t \in \mathcal{Y}} k_{i}(t)}, \quad (3) $$

$$ Q_{\alpha} = \frac{\sum_{i \in U_{\alpha}(t), t \in \mathcal{Y}} r_{i\alpha} \times R_{i}(t) \times \tilde{W}_{\alpha}(t)}{\sum_{i \in U_{\alpha}(t), t \in \mathcal{Y}} R_{i}(t) \times \tilde{W}_{\alpha}(t)}, \quad (4) $$

where $O_{i}(t)$ and $U_{\alpha}(t)$ are the sets of objects evaluated by user $i$ in year $t$, and the set of users who evaluated item $\alpha$ in year $t$, respectively. The set of years considered for the evaluation.

The main rationale given by (1)–(4) can be described as follows: the reputation of a user depends on the total number of ratings the user assigned in a period of time, and the quality of an item depends on...
the frequency of it being rated by users in a period of
time. For example, the reputation of a user in Amaz-
on depends on the number of movies he/she watched
or rated in a period of time. The more the movies the
user watched, the higher his/her reputation. On the
item side, the quality of an item depends on the
number of ratings it received in a period of time. If many
users watched and rated the movie in a period of time,
the movie is of higher timely quality, and vice versa.

The next crucial component of the ATR method is
aggregation. The reputation of a user is not established
at a single point of time. It is defined by
the succession of his/her behavior over years.

Since the quality of items is calculated iteratively
the behavioral weight factors can reduce the impact of malicious behaviours. The user reput-
atuation is computed as the Pearson correlation coef-
ficient between the rating vector of the user and the
quality vector of the corresponding items as the tem-
poral reputation. The process runs iteratively until
$|Q_a(t) - \overline{Q}_a(t)| < r$, where $r$ is the threshold set as
$10^{-4}$ in this work. When $r = 10^{-4}$, the quality of items
does not change and the iteration becomes stable.

We note that a user’s reputation depends on the
times he/she rated items in a period of time, and the
quality of items also changed by time. Therefore, by
taking the reciprocal of the number of ratings, the in-
fluence of malicious behavior on the item quality is re-
duced. More details of our ATR method implementa-
tion and datasets are available on the Baidu Netdisk\footnote{https://pan.baidu.com/s/1J0K0UULYcjZGk7yA7Qyw0w, May 2021. Access code: ux3t}.

\section{3.3 Experiment Data}
\subsection{3.3.1 Generated Network}
In order to validate the effectiveness of our proposed
method in a more general way, we generate a generated
network with $|\mathcal{U}| = 6000$ and $|\mathcal{O}| = 4000$. The rating
network has a certain sparsity ($\beta = 0.2$) and we add
each rating to the network one by one. Therefore, the
generated network will finally have $|\mathcal{U}||\mathcal{O}| \times \beta = 4.8 \times
10^5$ links. The construction process of the generated
network is widely used in previous researches \cite{26, 30, 38}.

\subsection{3.3.2 Real Networks}
In this paper, we verify our proposed ATR algo-
rithm by four real datasets, which are obtained on Amaz-
on, APS, MovieLens and Netflix respectively. In our
experiments, we binarize different networks and con-
struct links based on the scores from 1 to 5 in the four
datasets. Fig.1 shows the distribution of records by
their release/publication years. And basic statistics of
the four datasets are shown in Table 2.

1) \textbf{Amazon Dataset.} Amazon is the largest com-
pany in the US online retailing industry, covering areas
such as books, movies and more\footnote{https://www.amazon.com/, May 2021.}. Here we use a small
data set in Amazon, consisting of 16843 users and 26285
movies. The value of ratings on the movie is from 1 to
5, and all ratings span from 1997 to 2014.

2) \textbf{APS Dataset.} The American Physical Society
(APS) is the second largest physics organization in the
Fig. 1. Number of movies/publications records in the four real datasets over years. (a) Amazon. (b) APS. (c) MovieLens. (d) Netflix.

Table 2. Basic Statistics of Datasets

| Dataset          | |U|  | |O|  | KU | KO | |R|  |
|------------------|---|---|---|---|---|---|---|---|---|---|
| Generated network| 6 000 | 4 000 | 81 | 126 | 480 000 |
| Amazon           | 16 834 | 26 258 | 30 | 19 | 4 893 777 |
| APS              | 449 935 | 449 935 | 12 | 11 | 4 672 812 |
| MovieLens        | 10 702 | 19 931 | 509 | 306 | 6 099 708 |
| Netflix          | 25 000 | 17 734 | 207 | 292 | 1 070 556 |

The APS dataset studied here is a citation network between papers, including the citation relationship between 449 935 articles from 1893 to 2009. One can consider papers as users and articles cited by other papers as items.

3) MovieLens Dataset. MovieLens is one of the traditional movies recommendation systems. It is a non-commercial, research-oriented experimental online platform. We use a subset of the complete data. There are 5 000 000 rating records in this subset, each with a score from 1 to 5.

4) Netflix Dataset. Netflix is an American company originally mainly engaged in the online rental business of customized DVDs and high-quality compact discs. By extracting a small dataset from the data provided by it, we select 25 000 users and each user has rated at least 20 movies on average. Each movie that has been rated by any users has a score from 1 to 5.

---

①https://journals.aps.org, May 2021.
②https://www.grouplens.org, May 2021.
③https://www.netflixprize.com, May 2021.
In order to analyse the capability of the algorithm, a wide range of accuracy evaluation metrics are used including AUC, precision, recall and the $F$ value\textsuperscript{[39, 40]}.  

1) **AUC.** AUC is a standard metric used to measure the accuracy of classification or prediction tools. To compute AUC, we set $N$ independent comparisons between the score of the correct items identified by the algorithm with that of the incorrect items. Among the $N$ comparisons, if the correct items have a higher score than the incorrect items in $N_1$ comparisons, while in $N_2$ comparisons the correct and the incorrect items have the same score, then the value of AUC is given by $AUC = (N_1 + 0.5N_2)/N$. If all items are ranked randomly, $AUC = 0.5$; if the correct items always have a higher score than the incorrect items, $AUC = 1$.

2) **Precision.** Precision measures the accuracy of correct retrieval. Precision is the ratio of the number of correct items to the total number of retrieved items. Let $X$ be the set of correct items, and $Y$ be the set of items identified by the algorithm, and then the value of precision is given by $\frac{|X \cap Y|}{|Y|}$.

3) **Recall.** Recall also measures the accuracy of correct retrieval, and is the ratio of the number of correct items and the total number of correct items. In other words, $\text{recall} = \frac{|X \cap Y|}{|X|}$.

4) **$F$ Value.** The $F$ value based on precision and recall is used for a comprehensive measurement: $F = 2 \times \text{precision} \times \text{recall}/(\text{precision} + \text{recall})$.

### 4. Results

In this experiment, the average score, KS\textsuperscript{[8]}, CR\textsuperscript{[30]}, IR\textsuperscript{[29]}, IARR\textsuperscript{[26]}, BiRank\textsuperscript{[28]} and ATR are verified on the five datasets. Since item quality and user reputation are accumulated over a long period of time, there is always some inaccuracy. For example, the quality of some commodities is initially good, but when the time passes some of the stocks have lower qualities and received lower ratings, which makes the ratings of valuable items lower. The same applies to user reputation.

The evaluation of item quality is not formed instantaneously, but resulted from a long-term accumulation process, which changes with users’ subjective and emotional choices and experiences. For example, when a valuable product hits the shelves, users are influenced by favorable comments towards the product and may end up choosing the product. As time goes on, the rating of the product gradually decreases. Therefore, the quality of an item and the reputation of a user are determined by the accumulation of the reputation of multiple users in a time period. Fig.2 shows the AUC of different algorithms on the generated network.

**3.4 Experimental Evaluation Metrics**

**In order to analyse the capability of the algorithm, a wide range of accuracy evaluation metrics are used including AUC, precision, recall and the $F$ value\textsuperscript{[39, 40]}.**  

1) **AUC.** AUC is a standard metric used to measure the accuracy of classification or prediction tools. To compute AUC, we set $N$ independent comparisons between the score of the correct items identified by the algorithm with that of the incorrect items. Among the $N$ comparisons, if the correct items have a higher score than the incorrect items in $N_1$ comparisons, while in $N_2$ comparisons the correct and the incorrect items have the same score, then the value of AUC is given by $AUC = (N_1 + 0.5N_2)/N$. If all items are ranked randomly, $AUC = 0.5$; if the correct items always have a higher score than the incorrect items, $AUC = 1$.

2) **Precision.** Precision measures the accuracy of correct retrieval. Precision is the ratio of the number of correct items to the total number of retrieved items. Let $X$ be the set of correct items, and $Y$ be the set of items identified by the algorithm, and then the value of precision is given by $\frac{|X \cap Y|}{|Y|}$.

3) **Recall.** Recall also measures the accuracy of correct retrieval, and is the ratio of the number of correct items and the total number of correct items. In other words, $\text{recall} = \frac{|X \cap Y|}{|X|}$.

4) **$F$ Value.** The $F$ value based on precision and recall is used for a comprehensive measurement: $F = 2 \times \text{precision} \times \text{recall}/(\text{precision} + \text{recall})$.

**4. Results**

In this experiment, the average score, KS\textsuperscript{[8]}, CR\textsuperscript{[30]}, IR\textsuperscript{[29]}, IARR\textsuperscript{[26]}, BiRank\textsuperscript{[28]} and ATR are verified on the five datasets. Since item quality and user reputation are accumulated over a long period of time, there is always some inaccuracy. For example, the quality of some commodities is initially good, but when the time passes some of the stocks have lower qualities and received lower ratings, which makes the ratings of valuable items lower. The same applies to user reputation.

The evaluation of item quality is not formed instantaneously, but resulted from a long-term accumulation process, which changes with users’ subjective and emotional choices and experiences. For example, when a valuable product hits the shelves, users are influenced by favorable comments towards the product and may end up choosing the product. As time goes on, the rating of the product gradually decreases. Therefore, the quality of an item and the reputation of a user are determined by the accumulation of the reputation of multiple users in a time period. Fig.2 shows the AUC of different algorithms on the generated network.

**4.1 Identification of High-Quality Movies and Publications**

We use our proposed ATR algorithm and other benchmark algorithms to identify Oscar-winning movies by the ratings in Amazon, MovieLens and Netflix datasets, and the Nobel Prize winning publications in the APS datasets.

Since the quality of items in the four empirical datasets is unknown, we use awarded movies or publications as the target items to be identified by the algorithms. In the movie datasets, including the Amazon, MovieLens and Netflix datasets, we select 521 Oscar-winning films from 1928 to 2014 as the target items.

We then apply various algorithms to rank the quality of movies, and denote the matching number of Oscar-winning movies in the list of the top 5% ($f = 5$) of estimated quality by a specific algorithm to be $M$. Similarly, we identify 87 Nobel Prize-winning publications in the APS dataset and denote the matching number of Nobel Prize-winning publications in the top 5% ($f = 5$) of the quality list to be $M$.

In Fig.3, we examine items in the top $f$% of the item list ranked by their quality estimated by the different algorithms. As we can see in Fig.3, when $f$ increases, the values of $M$, i.e., the total number of Oscar-winning movies or Nobel Prize-winning publications identified by all the algorithms, tend to rise. For the datasets of MovieLens and Netflix, both ATR and KS obtain the highest value of $M$, as shown in Fig.3, and their results among items in the top $f$% of the estimated quality list almost overlap. However, for the datasets of Amazon
and APS, ATR identifies more high-quality items than KS. This also shows that the ATR algorithm can well utilize the interactive procedures to update the evaluation on both user reputation and item quality, such that it can achieve better results.

4.2 Matching Numbers by Years

In Fig. 4, we show the results of $M$ obtained by various algorithms over time. As we can see, the ATR algorithm outperforms the other algorithms in the four real network datasets. First of all, the ATR algorithm obtains much larger values of $M$ in the Amazon dataset compared with the other algorithms. We also see that from our results, some of the award-winning movies identified by the other algorithms are overlapped with one another, suggesting that these algorithms can only identify obviously good movies. In addition, there are malicious behaviors or operations such as the water army in the Amazon dataset [41], which leads to a reduction in the accuracy obtained by the other benchmark algorithms. On the other hand, we see that other benchmark algorithms also show a poor accuracy in the APS dataset. Finally, in the Netflix and the MovieLens dataset, it is clear that the ATR algorithm obtains larger values of $M$ and hence a higher accuracy compared with the other algorithms. Other than $M$, we also examine the standard metrics, including AUC, precision, recall and the $F$ values. As shown by the metrics in Fig. 5 and Fig. 6, even though the other algorithms can identify most of correct high-quality items, ATR and KS significantly outperform the other base-
Fig. 4. Dependence of $M$, i.e., the number of Oscar-winning movies or Nobel Prize-winning publications, on years, among items in the top 5% of the list ranked according to their quality estimated by the seven algorithms studied. (a) Amazon. (b) APS. (c) MovieLens. (d) Netflix.

line algorithms in the Amazon dataset.

In contrast, the ATR algorithm outperforms all the other algorithms in the APS dataset. Nevertheless, the KS algorithm slightly outperforms the ATR algorithm in the MovieLens and the Netflix dataset. Based on the observations, we can conclude that ATR and KS can accurately identify most of the high-quality items.

Using other evaluation metrics such as recall and the values of $M$, we can also observe the high accuracy of ATR and KS algorithms, which indicates that they can accurately identify valuable high-quality items from Amazon with malicious rating behaviors. The detailed comparison of the results in ATR and KS is shown in Table 3. Specially, in the Amazon and the APS dataset, the values of precision obtained by the method of the average score and IR are almost zero, suggesting that the reputation is not well evaluated. However, the ATR algorithm always obtains the largest value of precision in the four datasets. We also remark that the ATR algorithm has a less computational cost on top of its better performance.

In order to separately examine the effectiveness of user and item behavior weighting factors in (5) in detail, we compare the performance of ATR with that of the algorithms with these two individual factors respectively, i.e., ATR$_{NU}$ and ATR$_{NI}$ in Fig. 7. From Fig. 7, we can see that the ATR algorithm has better performance than ATR$_{NU}$ (removing $acc_{ua}(t)$ from (5)) and ATR$_{NI}$ (removing $acc_{ui}(t)$ from (5)). It proves that our idea with two behavioral weighting factors enables our proposed algorithm (ATR) to obtain more potential good items.
Fig. 5. Results of AUC, recall, the $F$ and the $M$ value obtained by different algorithms.

Fig. 6. Results of precision obtained by different algorithms. (a) Amazon. (b) APS. (c) MovieLens. (d) Netflix.
### 4.3 Robustness

Robustness refers to the ability of the algorithm to maintain a good result under various parameters, e.g., structure and size, and scenarios such as malicious attacks [8, 42]. Robustness could be used to evaluate the ability of the method to reduce the influence of random ratings. We examine the robustness of the ATR algorithm and the other methods through intentionally generating random ratings.

Initially, we assume that there is no malicious behaviors in all the datasets we studied. In other words, the ratings by users on specific items are independently assigned according to their preferences. We then apply 1000, 2000, 3000 to 10000 users to assign random ratings in the Amazon dataset, and from 10000, 20000, 30000 to 100000 publications to have random citations in the APS dataset. We apply the same way in the MovieLens dataset and the Netflix dataset as in the Amazon dataset. In the APS dataset, the rate of the references of papers in the original APS dataset is 5 (we assume the reference papers cited are usually highly relevant). A score from 1 to 5 represents the rating to a paper. The higher the ratings, the higher the quality of research as judged by the authors citing the papers.

We then compute RMSE, as shown in (6):

$$
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N_{\text{sam}}} (AUC_{\text{ran}}^i - AUC_{\text{real}}^i)^2},
$$

where $AUC_{\text{ran}}^i$ is the value of AUC obtained by the datasets with random ratings, $AUC_{\text{real}}^i$ is the value of AUC obtained by the original datasets without malicious behaviors, the label $i$ corresponds to the index of sample, and $N_{\text{sam}}$ corresponds to the number of samples. As we can see in Fig.8, the values of RMSE obtained by the IARR and ATR algorithms only increase slowly with the number of users or publications with random ratings or citations, suggesting that these two algorithms are more robust against malicious behaviors. The overall robustness of the ATR algorithm is better than that of the other baseline algorithms. Fig.8(b) shows that the ATR algorithm does not fluctuate much and tends to be stable as the number of random citations increases, indicating that the ATR algorithm has better robustness. The IARR and the IR algorithm show a declining trend, and the values of AUC gradually approaches their values in the original datasets.

These observations demonstrate that IR and IARR do not consider temporal dependency in user reputation and have a poorer anti-interference capability, which causes fluctuant results.

### 5 Conclusions

The proposed ATR algorithm considers the temporal factors of ratings to improve the evaluation of reputation significantly, and indeed, two behavioral weighting factors in the ATR method can capture the reputation and quality evolving features for each user and item respectively. The results on the temporal datasets from Amazon, MovieLens, Netflix and the American Physical Society citation data (APS) showed the significant improvement in identifying award-winning movies.
or milestone publications. In the meantime, the robustness enhancement of the ATR method against random ratings is also proved. Remarkably, our purposed ATR algorithm largely improves the accuracy of reputation evaluation and the robustness against random ratings when compared with the state-of-the-art algorithms.

However, malicious behaviors may also have great impact on reputation or quality aggregation. In addition, the attributes of items such as movie genre and cast are not considered in our algorithm. For instance, a celebrity may significantly impact how users select a movie. Potential future work is to study in-depth reputation and quality evaluation from the above perspectives.

References

[1] Guo L, Zhang C, Fang Y G, Lin P. A privacy-preserving attribute-based reputation system in online social networks. *Journal of Computer Science and Technology*, 2015, 30(3): 578-597. DOI: 10.1007/s11390-015-1547-9.

[2] Albert R, Barabási A L. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 2002, 74(1): Article No. 47. DOI: 10.1103/RevModPhys.74.47.

[3] Wu X, Fan W, Gao J, Feng Z M, Yu Y. Detecting marionette microblog users for improved information credibility. *Journal of Computer Science and Technology*, 2015, 30(5): 1082-1096. DOI: 10.1007/s11390-015-1584-4.

[4] Lian D, Ge Y, Zhang F, Yuan J N, Xie X, Zhou T, Rui Y. Scalable content-aware collaborative filtering for location recommendation. *IEEE Transactions Knowledge and Data Engineering*, 2018, 30(6): 1122-1135. DOI: 10.1109/TKDE.2018.2789445.

[5] Nie D C, An Y H, Dong Q, Fu Y, Zhou T. Information filtering via balanced diffusion on bipartite networks. *Physica A: Statistical Mechanics and Its Applications*, 2015, 421: 44-53. DOI: 10.1016/j.physa.2014.11.018.

[6] Lian D, Chen J, Zheng K, Chen E H, Zhou X F. Ranking-based implicit regularization for one-class collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*. DOI: 10.1109/TKDE.2021.3069057.

[7] Dellarocas C. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 2003, 49(10):1407-1424. DOI: 10.1287/mnsc.49.10.1407.17308.
[8] Dai L, Guo Q, Liu X L, Liu J G, Zhang Y C. Identifying online user reputation in terms of user preference. *Physica A: Statistical Mechanics and Its Applications*, 2018, 494: 403-409. DOI: 10.1016/j.physa.2017.12.020.

[9] Dellorocas C. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proc. the 2nd ACM Conference on Electronic Commerce*, Oct. 2000, pp.150-157. DOI: 10.1145/352871.352889.

[10] Goldberg D, Nichols D, Oki B M, Terry D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992, 35(12): 61-71. DOI: 10.1145/138850.138867.

[11] Massucci F A, Docampo D. Measuring the academic reputation through citation networks via PageRank. *Journal of Informetrics*, 2019, 13(1): 185-201. DOI: 10.1016/j.joi.2018.12.001.

[12] Nunes I, Taylor P, Barakat L, Griffiths N, Miles S. Explaining reputation assessments. *International Journal of Human-Computer Studies*, 2019, 123: 1-17. DOI: 10.1016/j.ijhcs.2018.12.001.

[13] Syed R. Enterprise reputation threats on social media: A case of data breach framing. *The Journal of Strategic Information Systems*, 2018, 28(3): 257-274. DOI: 10.1016/j.jsis.2018.12.001.

[14] Jia T, Wang D, Szymanski B K. Quantifying patterns of research-interest evolution. *Nature Human Behaviour*, 2017, 1(4): Article No. 0078. DOI: 10.1038/s41562-017-0078.

[15] Yang W P, Wang J, Xia C. Evolution of cooperation in Ponzi L J, Fombrun C J, Gardberg N A. RepTrak Pulse: Conceptualizing and validating a short-form measure of corporate reputation. *Corporate Reputation Review*, 2011, 14(1): 15-35. DOI: 10.1057/crr.2011.5.

[16] Sun J, Feng L, Xie J, Ma X, Wang D S, Hu Y Q. Revealing the predictability of intrinsic structure in complex networks. *Nature Communications*, 2020, 11(1): Article No. 574. DOI: 10.1038/s41467-020-14416-6.

[17] Centeno R, Fresno V, Chaquet J. From textual reviews to the spatial public goods game with the third-order reputation. *Scientific Reports*, 2014, 4(1): Article No. 5379. DOI: 10.1038/srep05379.

[18] Liao H, Mariani M S, Medo M, Zhang Y C, Zhou M Y. Ranking in evolving complex networks. *Physics Reports*, 2017, 689: 1-54. DOI: 10.1016/j.physrep.2017.05.001.

[19] Roberts N, Grover L. Leveraging information technology infrastructure to facilitate a firm’s customer agility and competitive activity: An empirical investigation. *Journal of Management Information Systems*, 2012, 28(4): 231-270. DOI: 10.2753/MIS0742-1222084009.

[20] Sarstedt M, Wilczynski P, Melewar T. Measuring reputation in global markets—A comparison of reputation measures’ convergent and criterion validities. *Journal of World Business*, 2013, 48(3): 329-339. DOI: 10.1016/j.jwb.2012.07.017.

[21] Gao J, Zhou T. Evaluating user reputation in online rating systems via an iterative group-based ranking method. *Physica A: Statistical Mechanics and Its Applications*, 2017, 473: 546-560. DOI: 10.1016/j.physa.2017.01.055.

[22] Saúde J, Ramos G, Caleiro C, Kar S. Reputation-based ranking systems and their resistance to bribery. In *Proc. IEEE International Conference on Data Mining*, November 2017, pp. 1063-1068. DOI: 10.1109/ICDM.2017.139.

[23] Liu X L, Guo Q, Hou L, Cheng C Y, Liu J G. Ranking online quality and reputation via the user activity. *Physica A: Statistical Mechanics and Its Applications*, 2015, 436: 629-636. DOI: 10.1016/j.physa.2015.05.043.

[24] Liao H, Zeng A, Xiao R et al. Ranking reputation and quality in online rating systems. *PLoS One*, 2014, 9(5): Article No. e97146. DOI: 10.1371/journal.pone.0097146.

[25] Gao J, Dong Y W, Shang M S, Cai S M, Zhou T. Group-based ranking method for online rating systems with spamming attacks. *Europhysics Letters*, 2015, 110(2): Article No. 28003. DOI: 10.1209/0295-5075/110/28003.

[26] He X, Gao M, Kan M Y, Wang D X. BiRank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(1): 57-71. DOI: 10.1109/TKDE.2016.2611584.

[27] Laureti P, Moret L, Zhang Y C, Yu Y K. Information filtering via iterative refinement. *Europhysics Letters*, 2006, 75(6): 1006-1012. DOI: 10.1209/epl/i2006-10204-8.

[28] Zhou Y B, Lei T, Zhou T. A robust ranking algorithm to spamming. *Europhysics Letters*, 2011, 94(4): Article No. 48002. DOI: 10.1209/0295-5075/94/48002.

[29] Lin K. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 1989, 45(1): 255-268. DOI: 10.2307/2532051.

[30] Naglerkerke N J. A note on a general definition of the coefficient of determination. *Biometrika*, 1991, 78(3): 691-692. DOI: 10.1093/biomet/78.3.691.

[31] Bao J, Zheng Y, Wilkie D, Mokbel M. Recommendations in location-based social networks: A survey. *GeoInformatica*, 2015, 19(3): 525-565. DOI: 10.1007/s10707-014-0220-8.

[32] Guárdal S, Yeung C H, Zhang Y C. Tracing the evolution of physics on the backbone of citation networks. *Physical Review E*, 2011, 84(4): Article No. 046104. DOI: 10.1103/PhysRevE.84.046104.

[33] Lü L, Medo M, Yeung C H, Zhang Y C, Zhang Z K, Zhou T. Recommender systems. *Physics Reports*, 2012, 519(1): 1-49. DOI: 10.1016/j.physrep.2012.02.006.

[34] Pardoe I, Simonott D K. Applying discrete choice models to predict academy award winners. *Journal of the Royal Statistical Society: Series A*, 2008, 171(2): 375-394. DOI: 10.1111/j.1467-985X.2007.00518.x.

[35] Jiang W, Wu J, Wang G, Zheng H Y. FluidRating: A time-evolving rating scheme in trust-based recommendation systems using fluid dynamics. In *Proc. the 2014 IEEE Conference on Computer Communications*, April 27-May 2, 2014, pp.1707-1715. DOI: 10.1109/INFOCOM.2014.6848108.

[36] Vidmer A, Medo M, Zhang Y C. Unbiased metrics of friends’ influence in multi-level networks. *EPJ Data Science*, 2015, 4(1): Article No. 20. DOI: 10.1140/epjds/s13688-015-0057-x.
[39] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In Proc. the 23rd International Conference on Machine Learning, June 2006, pp.233-240. DOI: 10.1145/1143844.1143874.

[40] Hanczar B, Hua J, Sima C, Weinstein J, Bittner M, Dougherty E R. Small-sample precision of ROC-related estimates. Bioinformatics, 2010, 26(6): 822-830. DOI: 10.1093/bioinformatics/btp037.

[41] Xu C, Zhang J, Chang K, Long C. Uncovering collusive spammers in Chinese review websites. In Proc. the 22nd ACM International Conference on Conference on Information & Knowledge Management, October 27-November 1, 2013, pp.979-988. DOI: 10.1145/2505515.2505700.

[42] Li R H, Yu J X, Huang X, Cheng H. Robust reputation-based ranking on bipartite rating networks. In Proc. the 2012 SIAM International Conference on Data Mining, April 2012, pp.612-623. DOI: 10.1137/1.9781611972825.53.

Hao Liao is currently a tenured associate professor of the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen. He received his Ph.D. degree in science from University of Fribourg, Fribourg, in 2015. His research interests include complex networks and big data mining, particularly, information mining, fake news detection, and social network analysis. Dr. Liao has served as TPC chair, SPC/TPC member for several International/national workshops and conferences. He has published over 50 papers on international conferences and journals.

Qi-Xin Liu received his B.S. degree in computer science from School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, in 2014. He is studying for his M.S. degree in software engineering in College of Computer Science and Software Engineering, Shenzhen University, Shenzhen.

Ze-Cheng Huang received his B.S. degree in computer science from the Central South University of Forestry and Technology of China, Changsha, in 2012, and his M.S. degree in computer science from College of Computer Science and Software Engineering, Shenzhen University, Shenzhen. He is currently an engineer in Pingan group, Shenzhen.

Ke-Zhong Lu received his Bachelor’s and Ph.D. degrees in computer science from the University of Science and Technology of China, Hefei, in 2001 and 2006, respectively. He is a professor of the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen. His research concerns big data, parallel and distributed computing, and wireless sensor network. He is a vice dean of the College of Computer Science and Software Engineering at Shenzhen University, Shenzhen.

Chi Ho Yeung received his Ph.D. degree in physics from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2009. He then worked as a postdoctoral research fellow in University of Fribourg in Switzerland and Aston University in the United Kingdom for four years. He is currently an associate professor in the Department of Science and Environmental Studies, The Education University of Hong Kong, Hong Kong. His major research interests include statistical physics, transportation networks, complex and social networks, artificial intelligence, and STEM education.

Yi-Cheng Zhang received his Ph.D. degrees in physics from the University of Rome Sissa Trieste, Roma, and La Sapienza University, Roma, in 1984. He has successively worked in the National Laboratory of Bookhaven (Brookhaven) in the United States, the Italian Institute of Nuclear Physics (INFN), and the University of Denmark Nordita in frontier research of physics. He is now a chair professor for the theoretical physics, University of Fribourg, Fribourg.