Journal of Ambient Intelligence and Humanized Computing (2019) 10:3023–3034
https://doi.org/10.1007/s12652-018-0928-7

ORIGINAL RESEARCH

A trust-based collaborative filtering algorithm for E-commerce recommendation system

Liaoliang Jiang[1] · Yuting Cheng[2] · Li Yang[2] · Jing Li[1] · Hongyang Yan[3] · Xiaoqin Wang[2]

Received: 26 December 2017 / Accepted: 11 May 2018 / Published online: 29 June 2018

Abstract

The rise of e-commerce has not only given consumers more choice but has also caused information overload. In order to quickly find favorite items from vast resources, users are eager for technology by which websites can automatically deliver items in which they may be interested. Thus, recommender systems are created and developed to automate the recommendation process. In the field of collaborative filtering recommendations, the accuracy requirement of the recommendation algorithm always makes it complex and difficult to implement one algorithm. The slope one algorithm is not only easy to implement but also works efficient and effective. However, the prediction accuracy of the slope one algorithm is not very high. Moreover, the slope one algorithm does not perform so well when dealing with personalized recommendation tasks that concern the relationship among users. To solve these problems, we propose a slope one algorithm based on the fusion of trusted data and user similarity, which can be deployed in various recommender systems. This algorithm comprises three procedures. First, we should select trusted data. Second, we should calculate the similarity between users. Third, we need to add this similarity to the weight factor of the improved slope one algorithm, and then, we get the final recommendation equation. We have carried out a number of experiments with the Amazon dataset, and the results prove that our recommender algorithm performs more accurately than the traditional slope one algorithm.

Keywords Collaborative filtering · Trusted data · User similarity · Predicting accuracy

1 Introduction

Information systems have provided an unprecedented abundance of information resources, which has led to the problem of information overload at the same time. Moreover, it has become more difficult and time-consuming for users to search for information on large-scale websites. To deal with this problem, many works study at users' behavior, such as the sensor networks (Shen et al. 2018a; Bhuiyan et al. 2017). Otherwise, many personalized recommendation systems using artificial intelligence (AI) approaches have been developed. As an important information filtering tool, a recommender system can practicably provide information and push services to users based on historical behavior data, such as ratings and reviews left by the user in the past, when they do not display their own informatio n needs. Some famous electronic commerce websites, such as Amazon and CD-Now, have employed the recommender technique to recommend products to customers, and it has improved the quality and efficiency of their services (Lee et al. 2005; Ahn 2008). Collaborative filtering algorithms

✉ Jing Li
lijingbeiyou@163.com

Liaoliang Jiang
liaoliangjiang@gmail.com

Yuting Cheng
chengyt0923@163.com

Li Yang
yangli@xidian.edu.cn

Hongyang Yan
hyang.yan@foxmail.com

Xiaoqin Wang
wangxiaoqinxd@163.com

[1] School of Computer Science, Guangzhou University, Guangzhou 510000, China

[2] School of Computer Science, Xidian University, Xi'an 710071, China

[3] Nankai University, Tianjin 300000, China

(Jin et al. 2004) are classic personalized recommendation algorithms that are widely used in many commercial recommender systems (Adomavicius and Tuzhilin 2005).

The collaborative filtering algorithm is an algorithm based on the following three assumptions: people have comparable preferences and interests, their preferences and interests are stable, and we can conclude their choice by referring to their past preferences. Because of the above expectations, the collaborative algorithm is based on the connection of one user's behavior with another user's behavior to find his immediate neighbors and according to his neighbor's interests or preferences to predict his interests or inclination. Amazon, one of the most famous e-commerce sites, applied collaborative filtering to recommend products to users.

Collaborative algorithms have been developed rapidly and into a variety of improved algorithms. Many of these improved collaborative algorithms are devoted to building recommendation systems. These algorithms can be classed into the user-based and item-based approaches. Item-based CF (Tiraweerakhajohn and Pinngern 2004; Xia et al. 2010) first analyzes the user-item matrix to identify relationships between different items and then use these relationships to indirectly compute recommendations for users. However, there are some problems, such as data sparsity, cold start and poor scalability. User-based collaborative filtering (Zhang et al. 2015; Jing et al. 2016) belongs to the first generation of collaborative filtering, the basic idea of which is that we make recommendations concerning the similarity between users. Among user-based collaborative filtering, by comparing and computing the similarity between the target user and other users in terms of behavior choice, we can spot some groups that are sharing similar interests, called the ”neighborhood”. Once our system can recognize the neighbor user for the target user, we can recommend the user items liked by his or her neighbor users. Thus, we can treat these neighboring users as a standard when we are trying to recommend items. The core of collaborative filtering is to determine a group of users that share similar interests with the target user. This kind of similar user is usually referred to as the nearest neighbor (Shi et al. 2008). Nevertheless, the traditional collaborative filtering method can select insufficiently representative users as neighbors of the active user. This means that recommendations made a posteriori are not sufficiently precise. However, the rising accuracy requirement always makes recommendation algorithms complex and hard to realize. Thus, an effective but easy-to-realize algorithm is needed.

The slope one algorithm was firstly proposed by Lemire in (Lemire and Maclachlan 2005). It was not only easy to achieve but also effective. However, the prediction accuracy of the slope one algorithm is not very high. In addition, the emergence of fraudulent internet users (Chen et al. 2013) has lead to many untrusted ratings. To solve these problems, we propose a slope one algorithm based on trusted data. Otherwise, the slope one algorithm does not perform very well when dealing with personalized recommendation tasks that concern the relationship of users because the slope one scheme and most of its improved algorithms are item-based collaborative filtering algorithms.

1.1 Our contributions

To solve these problems, we propose a slope one algorithm based on the fusion of trusted data and user similarity. This algorithm involves three steps. Firstly, we should select the trusted data. Secondly, we should calculate the similarity of users. Thirdly, we add this similarity to the weight of the improved slope one algorithm and get the final recommendation expression. We have carried out a lot of experiments with the Amazon dataset, and the results prove that our algorithm performs more accurately than the traditional slope one algorithm.

1.2 Paper organization

In this paper, we will present a related definition of the improved slope one algorithm in Sect. 2. The trusted recommendation model is shown in Sect. 3. Then, three slope one algorithms will be introduced in Sect. 4. After that, we will show our improved slope one algorithm in Sect. 5. The experiment will be presented in Sect. 6. Section 7 is a discussion of the whole article and our future work. Finally, Sect. 8 contains the conclusion of the article.

2 Related definition

Trusted data

We define the ratings in the Amazon dataset as n and the helpful ratings as m.

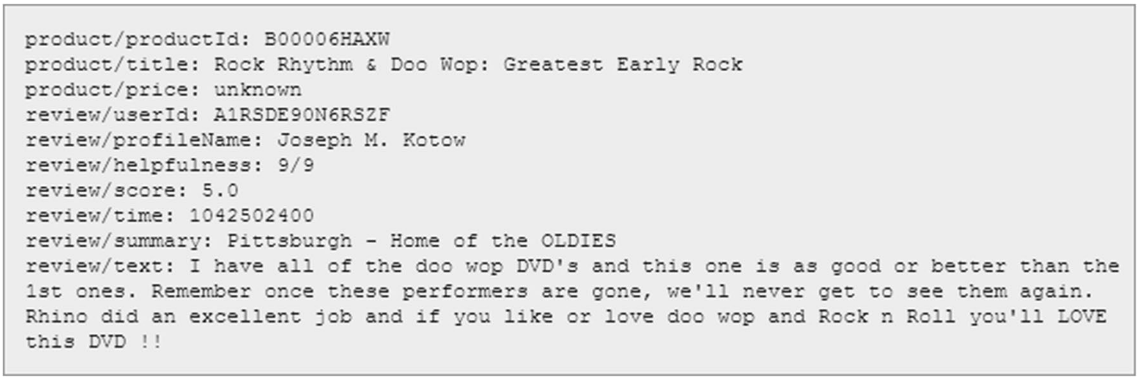
Then, we define the trusted ratio (r) as m/n, so the trusted rating is as follows:

$$r_{pi}^{T} = r_{pi} \times \frac{m}{n} \tag{1}$$

where r_{pi} is the rating of the user U_p for the item I_i, and r_{pi}^{T} denotes the trusted rating of user p for item i, as is known.

In Fig. 1, helpfulness is the trusted ratio (r). As you know, in the Amazon system, the votes are the amount of clicks of YES or NO, and the helpfulnuss is the amount of YES clicks when asked whether the rating is useful. So, we think that m / n represents the trusted ratio.

In our daily life, after we have bought a product, we may click YES or NO when asked whether the rating is useful or not to evaluated whether someone's score or review on the



```
product/productId: B00006HAXW
product/title: Rock Rhythm & Doo Wop: Greatest Early Rock
product/price: unknown
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than the
1st ones. Remember once these performers are gone, we'll never get to see them again.
Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOVE
this DVD !!
```

Fig. 1 The original data format of the Amazon data. product/productId: the id of the product commented; product/title: the title of the product; product/price: the price of the product, which is unknown; review/userId: the Id of the reviewer; review/profileName: the name of the reviewer; review/helpfulness: the fraction of users who found the review helpful; review/score: the rating of the product; review/time: the time of the product commented; review/summary: the key words in product reviews; review/text: the detailed product review

product is true and correct. So, we consider that when all the people who voted for this review click YES, the score can represent the real value of the product. Therefore, we can think of this score as the trusted score. Of course, this is the ideal situation, although there is a lot of ratings fraud in the electronic commerce network. Generally speaking, we consider that if more than half of all the people who voted for this review click YES, the score is trusted data. However, due to the existence of a lot of ratings fraud, we need a much higher ratio as the dividing line to divide the scores of trusted and untrusted in the electronic commerce network. By conducting a series of experiments, we found that when the trusted ratio is greater than 0.8, the recommended results will be fairly good.

3 The trusted recommendation model

The recommendation system has achieved great success in solving the problem of information overload, but there are still some problems, such as data sparseness, cold start and so on. How to obtain satisfactory results in the case of a sparse rating dataset has become an urgent problem in the field of recommender systems. One of the effective methods to solve the above problems is to introduce trust into the recommendation system. Cited References (Huang and Gong 2008; Ym and Nie 2007; Li et al. 2013b) et al. used the Pearson correlation coefficient to calculate the user similarity to define trust metrics. The existing trust metrics are all based on a common assumption that the data provided by the user are true, accurate, and can reflect the user's real preferences. In many cases, however, this assumption is not reasonable. So, in order to design a better credibility measure, more information about the user and the rating itself should be taken into account. Therefore, in this paper, we consider the reliability of the rating data and propose a trust-based recommendation model based on the collaborative filtering algorithm.

At present, there are many fake ratings on e-commerce websites. These fake ratings are mainly divided into the following categories. One is due to on-sale activities, where users will get back some cash if they give a high rating to the item. The other is hiring someone to rate items on purpose. Aiming at the second kind of fake ratings, the trust-based recommendation model with collaborative filtering mainly considers the following aspects: first, the model combines the trust relationship between users and the degree of trust for ratings. User similarity is regarded as the trust relationship between users. On the other hand, the degree of trust for ratings is mainly defined from two aspects: one is to spot fraudulent users and remove their ratings, the other is to provide a metric for each rating's trust-based strength based on other users' votes. Finally, the improved slope one algorithm based on the trust-based recommend model is introduced.

4 Slope one algorithms

4.1 Basic slope one algorithm

The basic idea of the algorithm is very simple, which is to use the average instead of the rating difference between two different individuals. The simplicity makes it especially easy to implement. The slope one algorithm with the form $f(x) = x + b$ assumes a linear relationship between two items, where x represents the rating of an already rated item and b denotes the average deviation. For example, the ratings for four items that user A, B and C recorded are as follows (Table 1):

If we want to know how the user C rates item 2, we must first compute the average difference value between item 2 and the other items that user C has rated, i.e., item 1 and item 4, and the calculation process is:

$$((4-3)+(2-4))/2=-0.5$$
$$((4-1)+(2-1))/2=2$$

Then we can get the prediction rating of user C for item 2 through the user C ratings of item 1 and item 4 plus the corresponding arithmetic mean:

$$((3-0.5)+(2+2))/2=3.25$$

So, we can fill the empty value table in a similar manner.

The slope one scheme takes into account information from other users who rated the same item and from the other items rated by the same user. It consists of two phases to produce the recommendation.

The first step is to calculate the average deviation of two items. Given a training set and any two items I_j and I_k, the algorithm considers the average deviation of item I_j with respect to item I_k as:

$$\overline{dev_{jk}}=\sum_{U_i\in II_{jk}}\frac{r_{ij}-r_{ik}}{|UI_{jk}|} \tag{2}$$

where $\overline{dev_{jk}}$ is the average deviation, and the rating of user i for item j and k are denoted as r_{ij} and r_{ik} respectively. $|UI_{jk}|$ is the number of the user set who rate both item j and k. U_i is the user i who rates both item j and k.

The second step is to produce the prediction.

$$P(r_{uj})=\frac{\sum_{k\in II_j}(\overline{dev_{jk}}+r_{uk})}{|II_j|} \tag{3}$$

where r_{uk} denotes the rating of user u for item k, and $P(r_{uj})$ denotes the prediction rating of user u for item j, as is known. $\overline{dev_{jk}}$ is the average deviation of item I_j with respect to item I_k. $|II_j|$ is the number of the items that are rated together with item j.

4.2 The weighted slope one algorithm

One of the disadvantages of slope one is that the number of ratings observed is not taken into consideration. Generally, to predict user A's rating of item L given user A's rating of items J and K, if 2000 users rated the pair of items J and L, whereas only 20 users rated the pair of items K and L, then user A's rating of item J is likely to be a far better predictor for item L than user A's rating of item K is.

4.3 The bi-polar slope one algorithm

While weighting served to favor frequently occurring rating patterns over infrequent rating patterns, we will now consider favoring another kind of especially relevant rating pattern. We accomplish this by splitting the prediction into two parts. Using the Weighted slope one algorithm (Guo et al. 2014), we derive one prediction from items users liked and another prediction using items that users disliked.

Given a rating range, say from 0 to 20, it might seem reasonable to use the middle of the range, 10, as the threshold, and to say that items rated above 10 are liked and those rated below 10 are not. This may work well if one's ratings are distributed evenly. Because we need to consider all types of users, including balanced, optimistic, pessimistic, and bimodal users, we treated the user's average as a threshold between the user's liked and disliked items. For example, optimistic users who tend to like every item they rate are assumed to hate those items which are rated below their average rating. This threshold ensures that our algorithm has a reasonable number of liked and disliked items for each user.

As usual, we base our prediction for item J by user B on the deviation from item I of users (like user A) who rated both items I and J. The bi-polar slope one algorithm restricts the set of ratings that are predictive further than this. First, in terms of items, only deviations between two liked items or deviations between two disliked items are taken into account. Second, in terms of users, only deviations from pairs of users who rated both item I and J and who share a like or dislike of item I are used to predict ratings for item J.

The splitting of each user into user likes and user dislikes effectively doubles the number of users. Observe, however, that the bi-polar restrictions just outlined necessarily reduce the overall number of ratings in the calculation of the predictions. Although any improvement in accuracy in light of such a reduction may seem counterintuitive where data sparseness is a problem, failing to filter out ratings that are irrelevant may prove even more problematic. Crucially, the bi-polar slope one algorithm predicts nothing from the fact that user A likes item K and user B dislikes this same item K.

To solve the problem that the prediction accuracy of the slope one algorithm is not very high, we propose a slope one algorithm based on trusted data. Furthermore, to solve the problem that the slope one algorithm does not perform so well when dealing with personalized recommendation tasks that concern the relationship of users, we propose an improved slope one algorithm based on the fusion of trusted data and user similarity.

5 The improved slope one algorithms

Computing the user similarity (Xie et al. 2011):

(1) Recording the rating matrix of the users-items
(2) Computing the similarity

In the rating matrix of the users-items, we define all the items rating of users as the user vector, so that each user can be represented as a m dimension rating vector, that is $U_p = (r_{p1}, r_{p2}, \dots, r_{pm})$, r_{pm} is the rating of user U_p for item I_m.

Then, we can compute the user similarity based on the users-items rating matrix.

5.1 Similarity measures

In order to analyze the effect of user similarity on the slope one algorithm, we need to find reliable similarity measures. Similarity measures play an important role because they are used both for selecting the neighborhood members and for weighting, so the way in which to calculate the similarity between two users is a key issue of collaborative filtering algorithms. Usually there are two models to measure the similarity of users. They are the Pearson correlation coefficient (PCC) (Breese et al. 1998) and Cosine-based similarity (CS)(Resnick et al. 1994). Equation (4) is PCC. Equation (5) is CS.

$$sim(p,v) = \frac{\sum_{i\in I}(r_{pi} - \overline{r_p}) \cdot (r_{vi} - \overline{r_v})}{\sqrt{\sum_{i\in I}(r_{pi} - \overline{r_p})^2 \cdot \sum_{i\in I}(r_{vi} - \overline{r_v})^2}} \tag{4}$$

where I denotes the item set rated both by user p and user v, $\overline{r_p}$ and $\overline{r_v}$ represent the mean of user p's rating and the mean of user v's rating, respectively. The rating of user p for item i and user v for item i is denoted as r_{pi} and r_{vi} respectively.

$$sim(p,v) = \frac{\sum_{i\in I} r_{pi} \cdot r_{vi}}{\sqrt{\sum_{i\in I} r_{pi}^2 \cdot \sum_{i\in I} r_{vi}^2}} \tag{5}$$

The CS does not take into account the difference between the average user ratings, so the reliability of calculating the similarity is very different. That is, the CS is more differentiated from the direction, but it is not sensitive to the absolute value. Therefore, there is no way to measure the difference in each dimension. For example, there are two users who are X and Y. Their ratings are (1,2) and (4,5), respectively. The result obtained by the CS is 0.98, which means that they are very similar, but X does not seem to love the second item; instead, Y loves the second item very much from the rating view. The CS is not sensitive to the absolute value, which leads to the above wrong results. We can avoid that error given the fact that the original rating can be replaced by the deviation between the rating and the average rating. Therefore, we compute the user similarity using the PCC in this paper.

5.2 The definition of trusted data

We define the votes in the Amazon dataset as n, the helpful as m, so the trusted ratio(r) is m / n. Then, we define the trusted rating as r^T_{pi}, as it is in (1).

5.3 Fusion of trusted data and similarity

Based on the trusted rating and user similarity as the weight, we can obtain the following weighted average deviation:

$$\overline{dev_{jk}}^{U\_sim} = \frac{\sum_{U_p \in UI_{jk} \cap sim(U_p,U_v)>0} sim(p,v)(r^T_{pj} - r^T_{pk})}{\sum_{U_p \in UI_{jk} \cap sim(U_p,U_v)>0} sim(p,v)} \tag{6}$$

where $\overline{dev_{jk}}^{U\_sim}$ is the average deviation of the improved slope one algorithms, $sim(p, v)$ is the similarity between user p and user v, the trusted rating of user i for items j and k are denoted as r^T_{pj} and r^T_{pk}, respectively. $|UI_{jk}|$ is the number of the user set who rate both item j and k.

Bringing the average deviation obtained above into (3), we get (7).

$$P(r_{uj}) = \frac{\sum_{k\in II_j}(\overline{dev}^{U\_sim} + r^T_{uk})}{|II_j|} \tag{7}$$

where $P(r_{uj})$ is the user u's prediction rating for item j from the improved slope one algorithms, r^T_{uk} denotes the trusted rating of user u for item k, as is known. $|II_j|$ is the number of items that are rated together with item j.

6 Experiment

6.1 Dataset

In this paper, we use a part of Amazon's items rating dataset (http://snap.stanford.edu/data/web-Amazon.html), and the dataset uses offline experiments to compare the prediction accuracy of various algorithms. First, we generate a standard dataset according to a certain format, then the dataset is divided into a training set and a test set (Li et al. 2013a) according to the ratio of 4:1.

The accuracy of a recommendation is the most basic index by which to evaluate the recommendation algorithm. Accuracy measures the extent to which the recommendation algorithm is able to accurately predict the user's liking for the recommended product. At present, most of the research on the evaluation index of the recommender system is based on the recommendation accuracy. There are many kinds of accuracy indices: some measure the proximity between the prediction rating and the actual rating of the items, some measure the correlation between the prediction rating and the actual rating, some consider the specific scoring, and some consider only the recommendation ranking. This paper mainly considers the accuracy of the prediction.

6.2 Evaluation metrics

Several metrics have been proposed to assess the accuracy of collaborative filtering methods. They are divided into two main categories: statistical accuracy metrics and decision-support accuracy metrics. In this paper, we use the statistical accuracy metrics.

Statistical accuracy metrics evaluate the accuracy of a prediction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. There are many accuracy indicators of the prediction rating. The idea of accuracy metrics is very simple, that is, calculating the difference between the predicting rating and the actual rating. The most classical metric is the mean absolute error (MAE) (Gong and Ye 2009). The MAE mainly calculates the average absolute error between the prediction rating and the actual rating in the test dataset. The smaller the MAE is, the more accurate the predictions would be, allowing for better recommendations to be formulated. Assuming the actual rating set is $r_1, r_2, \dots, r_N$, the prediction rating set is $p_1, p_2, \dots, p_N$, and the MAE is defined as:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{8}$$

More stringent than the MAE is the root mean square error (RMSE), which increases the punishment (punishment of square) of the prediction rating that is not accurate; therefore, the evaluation of system is more demanding. The smaller the RMSE, the more accurate the predictions would be, allowing for better recommendations to be formulated. The RMSE is defined as:

$$RMSE = \frac{\sqrt{\sum_{i=1}^{N} (p_i - r_i)^2}}{N} \tag{9}$$

6.3 Experimental results

1. The comparison between the slope one algorithm based on trusted data and the traditional algorithm:

   We define r to represent the trusted ratio of user ratings. The following table considers the trusted data that is greater than the trusted ratio in the table (the last column $r = Null$, which is the traditional algorithm, not taking into account the trusted ratio). The results of the rating prediction accuracy are as follows:

   First, we illustrate the selection of the trusted ratio. When the trusted ratio is close to 0, the prediction is not accurate, which shows there are many low-trusted ratio data in our data. So, we consider the data in which the trusted ratio is greater than 0.5, and the result is still not very high. Therefore, we chose the data in which the trusted ratio is relatively high, namely, the trusted ratio is greater than 0.8. The result shows that the prediction accuracy is greater than the data without taking into account the trusted ratio. Of course, when the trusted ratio is 1, the prediction accuracy is the best.

   Table 2 shows that the greater the trusted ratio is, the smaller the MAE is, and it is proved considering that trusted data are correct. At the same time, without considering the trusted data, the MAE of the traditional algorithm is 0.967, and it can be known that only when the trusted ratio is more than 0.8 is the prediction accuracy higher than the traditional algorithm. According to the survey analysis, the main reason is that people don’t care for this kind of behavior of clicking helpful and there is the emergence of the internet fraudulent users, so completely trusted data is very rare. Most importantly of all, when the trusted ratio is 1, the rating predicting accuracy could enhance the precision by approximately 31.9% more than the traditional algorithm, which is a very large increase. Therefore, the prediction based on the trusted data deserves considering. If we could improve people’s subjective behavior and discriminate the fraudulent internet users from normal users, the prediction accuracy will have great room for improvement.

2. The comparison between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on trusted data using MAE:

   The slope one algorithm based on trusted data has greatly improved the prediction accuracy, but it doesn’t consider the relation of users. Additionally, in real life, user similarity plays an important role in a user’s preferences, so we consider adding the similarity of users to the slope one algorithm based on the trusted data.

   Table 3 shows the comparison results between the slope one algorithm based on the fusion of trusted data and similarity and the traditional slope one algorithm. It is very obvious that the MAE of the improved slope one algorithm is smaller when the trusted ratio is greater than 0.8 compared with the traditional slope one algorithm, as seen in Table 3.

   We can know that the slope one algorithm based on the fusion of trusted data and similarity is better than the slope one algorithm based on the trusted data to some extent in Fig. 2. That is to say, when the trusted ratio is

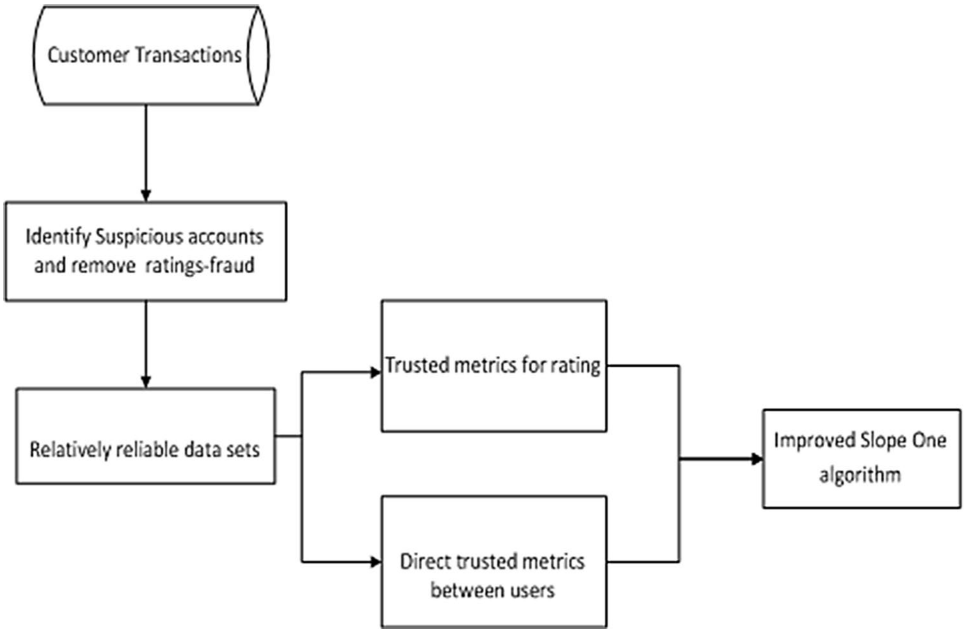

Customer Transactions
Identify Suspicious accounts and remove ratings-fraud
Relatively reliable data sets
Trusted metrics for rating
Direct trusted metrics between users
Improved Slope One algorithm

Fig. 2 The trusted recommendation model

Table 1 Three user's rating for items

User/items	1	2	3	4
A	3	4	2	1
B	4	2	3	1
C	3	?	?	2

Table 2 The result of MAE comparison between the algorithm based on trusted data and the traditional algorithm

r	>0	>0.5	>0.8	=1	Null
MAE	1.389	1.214	0.776	0.659	0.967

Table 3 The result of the MAE of our algorithm based on the fusion of trusted data and similarity

r	>0	>0.5	>0.8	=1	Null
MAE	1.189	1.064	0.758	0.598	0.799

the same, the MAE of the slope one algorithm based on the fusion of trusted data and similarity is smaller than the MAE of the slope one algorithm based on the trusted data.

Table 3 shows the MAE of our slope one algorithm based on the fusion of trusted data and similarity. It is very clear that the MAE of the improved slope one algorithm is smaller when the trusted ratio is greater than 0.8 compared with the traditional slope one algorithm, as seen in Table 2. We can see that the slope one algorithm based on the fusion of trusted data and similarity is better than the slope one algorithm based on the trusted data from Fig. 3. That is to say, when the trusted ratio is the same, the MAE of the slope one algorithm based on the fusion of trusted data and similarity is smaller than the MAE of the slope one algorithm based on the trusted data.

3. The Comparison between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on trusted data using RMSE:

   In Table 4, the RMSE is based on the trusted data, and the srrRMSE is based on the fusion of trusted data and similarity.

   Table 4 shows the prediction accuracy of the three kinds of algorithms when using RMSE as an indicator. When the trusted ratio (*r*) is different, the dataset is different. As *r* increases, the size of the dataset becomes smaller, especially when *r* is close to 1. Thus, as *r* increases, the RMSE does not completely show the decreasing trend. However, with the same dataset size, under the same trusted ratio, the slope one algorithm based on the fusion of trusted data and similarity is clearly better than the slope one algorithm based on the trusted data, as is shown in Fig. 3. That is to say, when the trusted ratio is the same, the RMSE of the slope one algorithm based on the fusion of trusted data and similarity is smaller than the slope one algorithm based on the trusted data. Briefly, these experiments show that the slope one algorithm based on the fusion of trusted data and similarity is the best (Fig. 4).

4. The comparison of our slope one algorithm based on user similarity under different sizes of datasets:

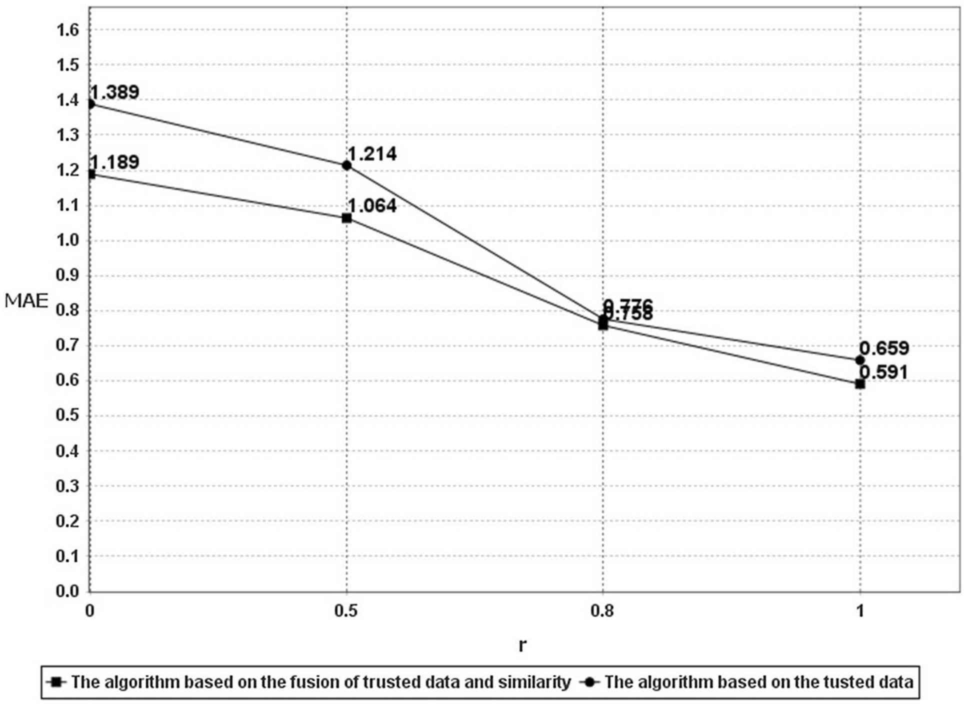


Fig. 3 The result of MAE comparison between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on the trusted data.

Table 4 The result of RMSE comparison between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on the trusted data

r	>0	>0.5	>0.8	=1	Null
srrRMSE	0.143	0.146	0.144	0.158	0.049
RMSE	0.164	0.164	0.145	0.187	0.057

Given Fig. 5, along with increasing of the size of the dataset, the MAE is decreasing, which means that the prediction precision is improved. Based on this result, the improved slope one algorithm will have better prediction precision when the size of the dataset increases. Moreover, when the trusted ratio is increasing, the size of the dataset is smaller; hence, the MAE should

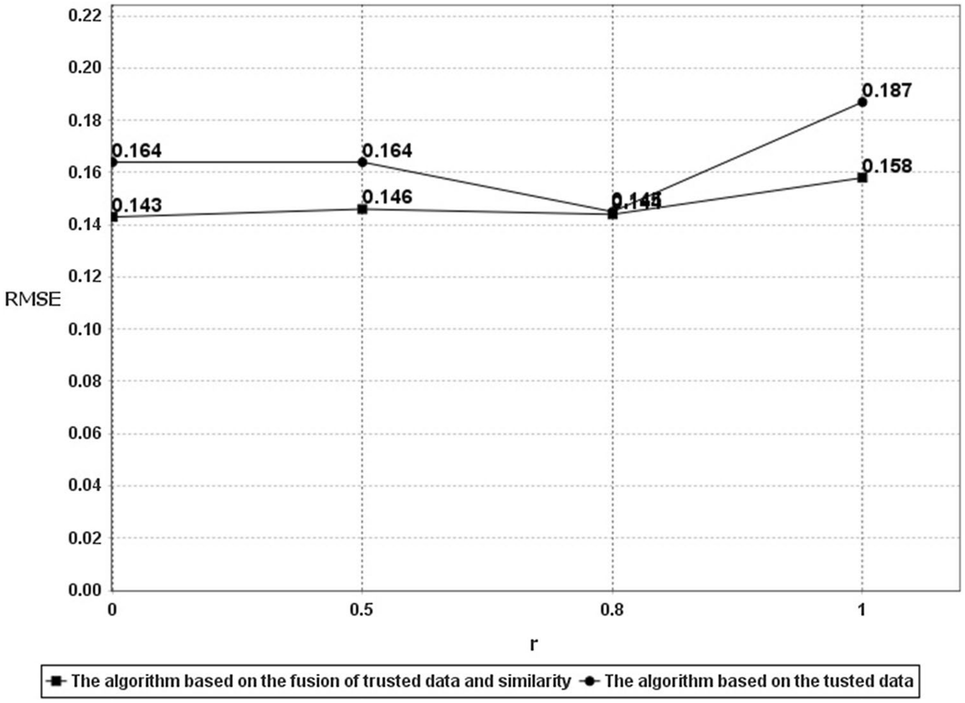


Fig. 4 The result of the RMSE comparison between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on the trusted data

increase according to Fig. 5. However, the MAE is actually decreasing with the increase in r. It can be seen that introducing the trust of ratings helps a lot in improving the prediction precision (Table 5).

6.4 Algorithm complexity analysis

To present a further explanation of the slope one algorithm based on the fusion of trusted data and similarity, we list the pseudocode of the partial calculation, which is the core of the algorithm.

Algorithm 1 The slope one algorithm based on the fusion of trusted data and similarity

```
Begin
Initialization:
$i$, $j$: the item $i$, the item $j$;
$dev_{ij}$: the average deviation between item $i$ and the item $j$;
Pseudocode:
for every item $j$ that user $u$ expresses no preference for
  for every item $i$ that user $u$ expresses a preference for
    for every other user $m$ expressing a preference for both $i$ and $j$
      find $sim(m, u)$
      add $dev_{ij}$, multiply $sim(m, u)$ to an total
      add above total to $u$'s preference value for $i$
End
```

where user u is the target user and item j is the target item for which we want to calculate the predicted rating. Assume *M* and *N* are the maximum number of users and items, respectively. From the description we have presented above, the complexity of the difference (i, j) is $O(n)$, and the complexity of the similarity is $O(m)$. Thus, the complexity of the slope one algorithm based on the fusion of trusted data and similarity is $O(m^2n^2)$ for all users and all items. Thus, the complexity of the slope one algorithm based on trusted data is $O(mn^2)$ (the maximum number of items is always much larger than the maximum number of users) (Song and Wu 2012). This analysis also proves that the complexity does not become a negative factor that affects the realization of the algorithm.

Table 5 The result of MAE under different sizes of datasets

The result of MAE under different sizes of datasets			
63,670	119,982	178,651	219,011
0.983	0.921	0.857	0.805

7 Some discussion

For further work, we mainly consider the following aspects. Firstly, finding a better way to calculate the similarity of users is very important, such as a new closeness evaluation algorithm (Yang et al. 2016). The closeness is introduced to map the relationship between the nodes according to the different interaction types in an online social network. In order to measure the impact of the information transmission between non-adjacent nodes in online social networks, an algorithm evaluating the closeness of the adjacent nodes and the nonadjacent nodes is given based on the relational features between users. By adopting the algorithm, the closeness between the adjacent nodes and the non-adjacent nodes can be obtained depending on the interaction time of nodes and the delay of their hops. Secondly, we want to compare the prediction accuracy of the several common recommendation algorithms based on the trusted data. Thirdly, based on the improved the accuracy of the recommendation algorithm, we want to join the privacy protection (Agrawal and Srikant 2000) of user rating data. This will make a very important progress. Privacy can be preserved by simply suppressing all sensitive data before any disclosure or computation occurs. Given a database, we can suppress specific attributes in particular records as dictated by our privacy policy. Rather than protecting the sensitive values of individual records, we may be interested in suppressing the identity (of a person) linked to a specific record. With the personalized recommendation service appearing, the user could quickly pick up the products in which they are interested, as well as expose the privacy information (Huang et al. 2016) that

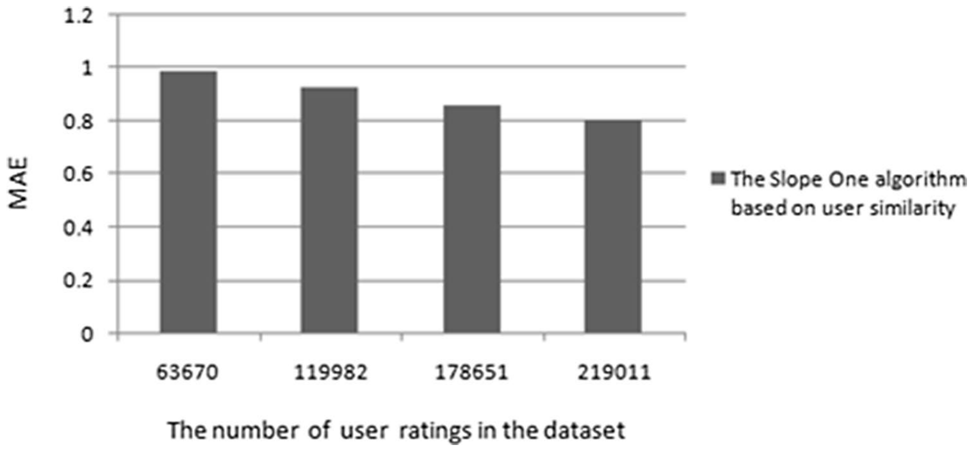


Fig. 5 The result of MAE comparison of the slope one algorithm based on user similarity under different sizes of datasets

many users do not want to be happened. This is a very serious problem, which brings a lot of trouble to people's daily life. On the other hand, we can sign the sensitive values to protect its integrality via some effective signature algorithm (Chen et al. 2016). Finally, we may consider to combine our algorithm with deep learning (Liu et al. 2017) and machine learning (Wu et al. 2017).

In addition, when we discuss the recommendation system, we are destined to face huge amount of data (Wu et al. 2016). If we don't have enough data as input, it is impossible to make accurate recommendations, at least not accurate enough. That is to say, with more data, there is a better recommendation effect. When we are able to fetch tons of user-related information, we have to run the same recommendation algorithm on a larger dataset. Such a huge dataset will definitely slow down the speed of computing recommendation results. If we spend too much time, users will be too impatient to wait for our recommendations, which is a disaster for recommendation applications. When dealing with such huge amount of data, a common solution is cloud computing (Voorsluys et al. 2011; Li et al. 2018; Gao et al. 2018; Tian et al. 2018), which employs lots of computers to do the actual computing procedure in parallel. Using this method, the whole computing job is divided into many tasks that can be executed on thousands of computers simultaneously. As you may guess, this kind of computing will dramatically decrease the overall time spent to produce reasonable recommendation results. Therefore, if we can combine cloud computing with the recommendation algorithm, we may have a jump on the computing speed. One big problem for collaborative filtering is scalability. When the volume of the dataset is very large, the cost of computation for CF will be very high. Recently, cloud computing has been the focus in order to solve the problem of large scale computation tasks. Cloud computing provides dynamically scalable and often virtualized resources as a service over the Internet (Xia et al. 2016; Shen et al. 2015; Guo et al. 2014; Ibtihal and Hassan 2017; Shen et al. 2018b; Xu et al. 2018). Users need not have knowledge of, expertise in, or control over the technology infrastructure in the ”cloud” that supports them. So, cloud computing is very powerful and easy to use.

Moreover, a limitation of our approach as well as the common problem for a recommender system is the cold-start problem (Schein et al. 2002), where recommendations are required for items that no one (in our dataset) has yet rated. Pure collaborative filtering cannot help in a cold start setting, since no user preference information is available to form any basis for recommendations. However, content information can help bridge the gap from existing items to new items by inferring similarities among them. Thus, we can make recommendations for new items that appear similar to other recommended items. This is valuable for our further research.

8 Conclusion

This paper is aimed at the problem of low accuracy of the traditional slope one algorithm and the untrusted ratings in recommender systems. Moreover, we propose a slope one algorithm based on the fusion of trusted data and user similarity. The algorithm we proposed can applyed in many applications, such as the recommendation system for social networks (Peng et al. 2017a; Cai et al. 2017; Jiang et al. 2016), or loaction-based services (Peng et al. 2017b).

We implement our experiment on parts of Amazon's items rating dataset, we do evaluation in four aspects. Firstly, we compared slope one algorithm based on trusted data and the traditional algorithm. Secondly, we researched the difference between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on trusted data using MAE. Thirdly, the comparison between the slope one algorithm based on the fusion of trusted data and similarity and the algorithm based on trusted data using RMSE. Finally, we had a comparison between our slope one algorithm based on user similarity under different sizes of datasets. The experimental results show that the slope one algorithm based on the fusion of trusted data and user similarity has greatly improved the prediction accuracy than traditional slope one algorithm. If we could improve people's subjective behavior of clicking on votes and identify fraudulent internet users, the prediction accuracy will dramatically improve and we will be able to provide more accurate recommendation services for users. Moreover, we can provide more extensive recommendation services based on different data types (McAuley et al. 2015). On the other hand, we may consider other method applyed to recommendation system, such as semisupervised feature analysis (Chang and Yang 2017).

Acknowledgements This work was supported by the Natural Science Foundation of Guangdong Province for Distinguished Young Scholars (2014A030306020), the Guangzhou Scholars Project for the Universities of Guangzhou (No. 1201561613), the Science and Technology Planning Project of Guangdong Province, China (2015B010129015), the National Natural Science Foundation of China (No. 61472091) and the National Natural Science Foundation for Outstanding Youth Foundation (No. 61722203).

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749

Agrawal R, Srikant R (2000) Privacy-preserving data mining. ACM Sigmod Record, ACM 29:439–450

Ahn HJ (2008) A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf Sci 178(1):37–51

Bhuiyan MZA, Wang G, Wu J, Cao J, Liu X, Wang T (2017) Dependable structural health monitoring using wireless sensor networks. IEEE Trans Dependable Secure Comput 14(4):363–376

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., pp 43–52

Cai J, Wang Y, Liu Y, Luo JZ (2017) Enhancing network capacity by weakening community structure in scale-free network. Future Generation Computer Systems. https://doi.org/10.1016/j.future.2017.08.014

Chang X, Yang Y (2017) Semisupervised feature analysis by mining correlations among multiple tasks. IEEE Trans Neural NetwLearn Syst 28(10):2294–2305

Chen C, Wu K, Srinivasan V, Zhang X (2013) Battling the internet water army: Detection of hidden paid posters. In: Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, IEEE, pp 116–120

Chen W, Lei H, Qi K (2016) Lattice-based linearly homomorphic signatures in the standard model. Theor Comput Sci 634:47–54

Gao Cz, Cheng Q, Li X, Xia Sb (2018) Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network. Cluster Computing, pp 1–9

Gong S, Ye H (2009) Joining user clustering and item based collaborative filtering in personalized recommendation services. In: Industrial and Information Systems, 2009. IIS'09. International Conference on, IEEE, pp 149–151

Guo P, Wang J, Geng XH, Kim CS, Kim JU (2014) A variable threshold-value authentication architecture for wireless mesh networks. J Internet Technol 15(6):929–935

Huang CB, Gong SJ (2008) Employing rough set theory to alleviate the sparsity issue in recommender system. In: Machine Learning and Cybernetics, 2008 International Conference on, IEEE, vol 3, pp 1610–1614

Huang Y, Li W, Liang Z, Xue Y, Wang X (2016) Efficient business process consolidation: combining topic features with structure matching. Soft Computing, pp 1–13

Ibtihal M, Hassan N et al (2017) Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. Int J Cloud Appl Comput (IJCAC) 7(2):27–40

Jiang W, Wang G, Bhuiyan MZA, Wu J (2016) Understanding graph-based trust evaluation in online social networks: methodologies and challenges. ACM Comput Surv (CSUR) 49(1):10

Jin R, Chai JY, Si L (2004) An automatic weighting scheme for collaborative filtering. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 337–344

Jing Y, Zhang L, Phelan C (2016) A novel recommendation strategy for user-based collaborative filtering in intelligent marketing. J Dig Inf Manag 14(2)

Lee JS, Jun CH, Lee J, Kim S (2005) Classification-based collaborative filtering using market basket data. Expert Syst Appl 29(3):700–704

Lemire D, Maclachlan A (2005) Slope one predictors for online rating-based collaborative filtering. In: Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, pp 471–475

Li B, Huang Y, Liu Z, Li J, Tian Z, Yiu SM (2018) Hybridoram: practical oblivious cloud storage with constant bandwidth. Information Sciences. https://doi.org/10.1016/j.ins.2018.02.019

Li J, Feng P, Lv J (2013a) An improved slope one algorithm for collaborative filtering. In: Natural Computation (ICNC), 2013 Ninth International Conference on, IEEE, pp 1118–1123

Li J, Wang Y, Wu J, Yang F (2013b) Application of user-based collaborative filtering recommendation technology on logistics platform. In: Business Intelligence and Financial Engineering (BIFE), 2013 Sixth International Conference on, IEEE, pp 135–138

Liu Y, Ling J, Liu Z, Shen J, Gao C (2017) Finger vein secure biometric template generation based on deep learning. Soft Computing, pp 1–9

Ym Luo, Nie Gh (2007) Research of recommendation algorithm on integration of semantic similarity and the item-based cf [j]. J Wuhan Univ Technol 1:023

McAuley J, Targett C, Shi Q, Van Den Hengel A (2015) Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp 43–52

Peng S, Yang A, Cao L, Yu S, Xie D (2017a) Social influence modeling using information theory in mobile social networks. Inf Sci 379:146–159

Peng T, Liu Q, Meng D, Wang G (2017b) Collaborative trajectory privacy preserving scheme in location-based services. Inf Sci 387:165–179

Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work, ACM, pp 175–186

Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 253–260

Shen C, Chen Y, Guan X (2018a) Performance evaluation of implicit smartphones authentication via sensor-behavior analysis. Inf Sci 430:538–553

Shen J, Tan HW, Wang J, Wang JW, Lee SY (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. J Internet Technol 16(1):171–178

Shen J, Gui Z, Ji S, Shen J, Tan H, Tang Y (2018b) Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. J Netw Comput Appl 106:117–123

Shi X, Ye H, Gong S (2008) A personalized recommender integrating item-based and user-based collaborative filtering. In: Business and Information Management, 2008. ISBIM'08. International Seminar on, IEEE, vol 1, pp 264–267

Song S, Wu K (2012) A creative personalized recommendation algorithm user-based slope one algorithm. In: Systems and Informatics (ICSAI), 2012 International Conference on, IEEE, pp 2203–2207

Tian H, Chen Z, Chang CC, Huang Y, Wang T, Huang Za, Cai Y, Chen Y (2018) Public audit for operation behavior logs with error locating in cloud storage. Soft Computing, pp 1–14

Tiraweerakhajohn C, Pinngern O (2004) Finding item neighbors in item-based collaborative filtering by adding item content. In: Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th, IEEE, vol 3, pp 1674–1678

Voorsluys W, Broberg J, Buyya R (2011) Introduction to cloud computing. Principles and paradigms, Cloud computing, pp 1–41

Wu J, Guo S, Li J, Zeng D (2016) Big data meet green challenges: big data toward green applications. IEEE Syst J 10(3):888–900

Wu Z, Lin W, Zhang Z, Wen A, Lin L (2017) An ensemble random forest algorithm for insurance big data analysis. In: IEEE International Conference on Computational Science and Engineering, pp 531–536

Xia S, Zhao Y, Zhang Y, Xin C, Roepnack S, Huang S (2010) Optimizations for item-based collaborative filtering algorithm. In: Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on, IEEE, pp 1–5

Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352

Xie H, Zhu Q, Qu H, Yang R (2011) User-based collaborative recommendation filtering algorithm using extremely valued ratings. Int J Digital Content Technol Appl 5(9)

Xu J, Wei L, Zhang Y, Wang A, Zhou F, Cz Gao (2018) Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures. J Netw Comput Appl 107:113–124

Yang L, Qiao Y, Liu Z, Ma J, Li X (2016) Identifying opinion leader nodes in online social networks with a new closeness evaluation algorithm. Soft Computing, pp 1–12

Zhang Z, Kudo Y, Murai T (2015) Applying covering-based rough set theory to user-based collaborative filtering to enhance the quality of recommendations. In: International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making, Springer, pp 279–289

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.