A User Trust-Based Collaborative Filtering Recommendation Algorithm*

Fuzhi Zhang, Long Bai, and Feng Gao

School of Information Science and Engineering Yanshan University, Qinhuangdao , 066004, Hebei Province, P. R. China

Abstract. Due to the open nature of collaborative recommender systems, they can not effectively prevent malicious users from injecting fake profile data into the ratings database, which can significantly bias the system's output. With this problem in mind, in this paper we introduce the social trust of the users into the recommender system and build the trust relation between them. The values of trust among users are adjusted by using the reinforcement learning algorithm. On the basis of this, a user trust-based collaborative filtering recommendation algorithm is proposed. It uses the combined similarity to generate recommendation, which considers not only the similarity between user profiles but user trust as well. Experimental results show that the proposed algorithm outperforms the traditional user-based and item-based collaborative filtering algorithm in recommendation accuracy, especially in the face of malicious profile injection attacks.

Keywords: collaborative filtering; recommender system; trust model; malicious attack; Reinforcement learning.

1 Introduction

At present, personalized collaborative recommender systems have become an important part of many e-commerce Web sites. However, such recommender systems introduce security issues that must be solved if users are to perceive these systems as objective, unbiased, and accurate [1]. The open nature of collaborative recommender systems provides an opportunity for malicious users to access the systems with multiple fictitious identities and insert a number of fake user profiles in an attempt to bias the recommender systems in their favor. Traditional collaborative recommender systems can not prevent this kind of malicious attack. Thus how to ensure the quality of recommendations for personalized collaborative recommender systems in the face of profile injection attacks has become an important issue.

Recent research on the security issues of collaborative recommender systems has focused on techniques that can be used to protect the predictive integrity

* This work was supported in part by the National Basic Research Program of China (No.2005CB321902), and the Natural Science Foundation of Hebei Province, China (No.F2008000877).

S. Qing, C.J. Mitchell, and G. Wang (Eds.): ICICS 2009, LNCS 5927, pp. 411–424, 2009.

of collaborative recommenders from malicious profile injection attacks. Research work falls into two categories: techniques for detecting and discounting biased profiles [2,3]; and techniques that increase the robustness of the recommender systems [4,5]. In this paper, we will explore to combine the user trust mechanism with collaborative filtering algorithm for the purpose of improving the robustness of collaborative recommendation algorithm and ensuring the quality of recommendations.

Montaner et al. [6] introduced a trust model into the recommendation algorithm, so users could get the recommendations from the trust-building group. The tentative idea was that trust-factor was based on the customer's satisfaction with the recommended items and trust value could be dynamically adjusted. It was a fresh idea for the recommendation algorithm. The drawback of this method was lack of trust information among users at the beginning of recommendation, and what's more it was inefficient to build the trust group. So it was not an effective way to defend against the malicious noise.

Massa et al [7] proposed a method that users who accepted the recommendations would evaluate the recommended items. The active user would get a rating that stands for the trust value of target user to the active user. The trust information was propagated among users who had a trust relation with the accepter. In this way a relation network with the trust value among users in the group would be built, even if they didn't have a direct interaction with each other. This was an effective way to build the trust network among users. However, due to the lack of restriction on the propagation of trust among users in the group, this method might lead to the flooding of trust and cause it to be out of control in the end. In addition, the authors did not give an effective way to measure the relation between users, so the initial trust values could not be effectively quantified, which could not ensure the reliability of trust values.

John O'Donovan et al [8] proposed an approach to overcome the shortcoming mentioned above. The basic idea was to build a relation between users with recommended items. Based on the tentative idea, there would be a higher weight to active user who had more accurate recommendations on items than those with poor records within the recommendation process. They supposed that users with a high authentic value have less intention to deceive others. The item-trust recommendation algorithms were more effective to defend the random attacks [9], but if the malicious users changed the attack strategies, in particular, they had some collaboration with others; this method would not effectively cut down the negative effect. Due to the lack of trust between users, they couldn't clearly judge who accepted item, who can be trusted or not.

To overcome the drawback, in this paper we explore to exploit trust information explicitly expressed by the users to improve the robustness of recommender systems. We give a user trust model and build a trust network for users by reinforcing learning. The trained items are chosen from the items which have been rated by the users. We also propose a user trust-based collaborative recommendation algorithm to defend malicious noise. The experimental results show that our algorithm has a significant improvement in stability compared with the standard

collaborative filtering algorithm, and the algorithms with the constraint of user trust are more robust than other model-based algorithms, especially for counteracting malicious noise.

2 Background

2.1 Traditional Collaborative Recommendation Algorithm

Traditional collaborative recommendation algorithm, for example, the user-based collaborative filtering, uses the similarity of user profiles to form a neighborhood of peer users with similar tastes, then extrapolates the user's predicted rating for a target item from the ratings of his or her peer users [1]. The core of this algorithm is to compute the similarity between users. There are several methods can be used to compute the similarity of users, such as cosine correlation coefficient, modify cosine similarity, and Pearson correlation coefficient. In this paper we use the Pearson correlation to calculate the similarity of users.

Let $D = \{U, I, R\}$ be a data source of a recommender system, where $U = \{user_1, user_2, ..., user_m\}$ is a set of users of the system, $I = \{item_1, item_2, ..., item_n\}$ is a set of items of the system, and R is a user ratings matrix, where $r_{i,j} \in R$ represents the rating of $user_i$ on $item_i$. The similarity between *user* u and *user* n is given by the following Pearson's correlation coefficient Equation [10]:

$$Sim(u,n) = \frac{\sum_{C \in I_{u,n}} (R_{u,c} - \overline{R_u})(R_{n,c} - \overline{R_n})}{\sqrt{\sum_{C \in I_{u,n}} (R_{u,c} - \overline{R_u})^2} \sqrt{\sum_{C \in I_{u,n}} (R_{u,c} - \overline{R_u})^2}} \quad (1)$$

Where $R_{u,c}$ and $R_{n,c}$ are the rating of *user* u and *user* n on *item* c, $\overline{R_u}$ and $\overline{R_n}$ are the average ratings over all rated items for u and v, respectively. The set $I_{u,n}$ stand for the rating items on which *user* u and *user* n have co-rated.It is important to underline that the coefficient can be computed only if there are items rated by both the users.

2.2 Reinforcement Learning

Reinforcement learning is a machine learning method to solve problem through trial-and-error interactions with a dynamic environment. In the standard reinforcement learning model, an agent was connected to its environment via perception and action. In this paper we use the idea of the reinforcement learning to build the direct trust between users. Formally, the reinforcement learning model consists of [11]:

(1) a set of environment states: S;
(2) a discrete set of agent actions: A;
(3) a reward function R :$S \times A \longrightarrow R$;
(4) a state transition function T:$S \times A \longrightarrow \prod(s)$,where a member of $\prod(s)$ is a probability distribution over the set S. We write $T(s, a, s')$ for the probability of making a transition from state s to s' using action a.

2.3 Item-Level Trust

In [8] the authors proposed a view that the users who had made lots of accurate recommendation predictions in the past could be viewed as trustworthy compared with those made many poor predictions. It is a good manner to evaluate the recommendation quality by the target user. The evaluation can be viewed as a trust value to the active user. In traditional collaborative filtering, participators were viewed as collectivity to predict the items for target user, which is hard for assessor to estimate trust for each user.

Accordingly, we separately calculate correctness of producer's prediction by comparing predicted rating and the actual rating of the target users.

Let $T_n(i,u)$ be a trust value of user u for user n, if user n predicts the rating for the user u, the trust value is given by Equation 2.

$$T_n(i,u) = 1 - v_n^i \tag{2}$$

Where variable v_n^i is the deviation factor which represents the deviation degree of the active user n to the target user u on item i. The value of v_n^i is given by Equation 3 [12].

$$v_n^i = \frac{|r_n^i - r_u^i|}{d} \tag{3}$$

Where variable r_n^i is the rating of user u on item i, r_u^i is the rating of the target user u on item i, d is the span value of item's rating and the default value of the variable is 5.

3 User Trust Model and Generation Algorithm

Traditional user-based collaborative recommendation algorithm uses the similarity of users' tastes to generate recommendations. This profile-level similarity method is subject to manipulation by malicious users. Thus the reliability of users should take into account within the recommendation process. In this paper we use trust between users to express the reliability of users and combine user trust with user-based collaborative recommendation algorithm.

3.1 Definition of User Trust Model

In this section we introduce a formal trust relationship which is the extension of the representation used in [8]. To model the degree of trust, we assume that target user can assign a certain value to the active user by using the co-rated items of the users.

Trust metrics can be imported to help the target user to compute the trust value about the active users. The trust metrics is computable on most users, even on pairs of users who have only one co-rated item. A user is also able to establish trust via trust propagation on users with whom has no co-rated item. We use two types of trust: direct trust and recommendation trust. The former can be constructed

by users with exchange experiences such as friendship, good views. The latter is credit of a user award by the other users who are reliable by public.

Definition 1. Direct trust:
LetT_n^u represent the direct trust of the target user for the active user, the direct trust value is given by Equation 4.

$$T_n^u = \frac{\sum_{i=1}^{k} t_n^i}{\sum_{i=1}^{k} |t_n^i|} \tag{4}$$

Where t_n^i represents the trust value of target user u for active user n on item i. The trust value can be divided into five parts, $t_n^i \in [-0.2, -0.1, 0, 0.1, 0.2]$, the value of each part depends on the deviation factor of v_n^i which is given by equation 3, and k is the number of the set which contains items that the active user and the target user have co-rated.

The direct trust has the following property. If the active user has a positive experience to influence the target user, then the positive experience will be strengthening. If there has a negative experience, it will be given a distrust value as a punishment. In a real recommender system, however, the number of items is huge and the number of ratings provided by each user is very small, so there are few co-rated items between users. In such cases the traditional collaborative recommendation algorithm can't effectively calculate the similarity between users, which provides few chances for users to interact with each other. As a result, the direct trust is also scarce. Thus the single direct trust is not enough to express the trust relation between users. We need to introduce the recommendation trust into the trust building process.

Definition 2. Recommendation trust:
The recommendation trust is computed by target user's trust group who has an interaction with the active user. Let m be a set of trust group of the target user, which contains the users who have a reliable intercourse with the target user u. Let w_i be a trust factor of the target user for the active user, which depends on the history of intercourse experiences. Let T_m^n represent the recommendation trust that is computed by the set of user m who has a direct trust with active user n.

$$T_m^n = \frac{\sum_{i=1}^{k} w_i T_{m_i}^n}{\sum_{i=1}^{k} w_i} \tag{5}$$

Where w_i is a trust value of the target user u for the reliable user in the set m, $T_{m_i}^n$ is the direct trust value of user m_i (in the set m) for the active user n.

Using Equation 5, we can build a trust relation between target user and active user with the expression of the trust set that has a direct interaction with the target user. In this way we can reduce the negative influence caused by lacking information about the active user. If a user has a poor credit record, the record about the user will spread via recommendation trust, so the user will have less chance to participate in the recommendation. On the contrary if a user has a perfect record, he will be viewed as a genuine user instead of a malicious.

Definition 3. User trust value:
Let $Trust_n^u$ stand for the trust value of the target user u for the active user n, it is combined with the direct trust T_n^u, which is the value of the target user u for the active user n, and the recommendation trust which is the expression of the set trusted by the target user u. The combined trust value is given by Equation 6.

$$Trust_n^u = \alpha T_n^u + \beta T_m^n \tag{6}$$

Where α,β are weighting factors to adjust the two parts, they are constrained by the equation $\alpha + \beta = 1$. The parameter α is used to balance the direct trust and β is used to balance the recommendation trust. If users have more successful interaction, there will be more similarity and direct trust between those users who have interaction with each other.

As mentioned above, a user usually rates on a very small part of the items. It is difficult to compute direct trust between arbitrary users. We can introduce recommendation trust to solve this problem. The proportion of recommendation trust must be restricted, because it will bring about a distortion recommendation. For example, a target user 'A' may have little direct trust with the active user 'D' due to the lack of co-rating items. Maybe the two users have different interest. If user 'D' has similar tastes with group 'm' trusted by user 'A', and they have a good trust with each other. Thus the recommendation trust of T_m^B will be a good value. As a result, the active user 'D' who has little common interesting with 'A' will be a candidate to participate in the prediction. We can introduce the parameters to balance the direct trust and recommendation trust.

The user trust includes two parts. The first part is the interaction of target user and active user. If the active user has many good predictions for target user, he or she will not deceive the target user. The second part is the opinions of the other users who are trusted by the active user, they are also an important reference to the target user. In this way the target user will receive a credible recommendation from the producers.

3.2 User Trust Generation Algorithm

In the open collaborative recommendation environment, ensuring the reliability of users' ratings is very important to improve the quality of recommendation. With this problem in mind, we introduce the user trust into the collaborative filtering algorithm and calculate the degree of user trust using the idea of reinforcement learning.

In this section, we will provide three algorithms. The first algorithm is to calculate the direct trust between the target user and the active user, the second

is to compute the recommendation trust between users, and the third is to calculate the combined trust of direct trust and recommendation trust.

Algorithm 1

Input: target user, active user
Output: the value of the direct trust
Step:
1: **begin**
2: **repeat**
3: $ItemU \longleftarrow getItem(userId)$
/*To store the rating items of target user into set ItemU*/
4: **end repeat**
5: **repeat**
6: $Comm_Item \longleftarrow getCommonItem(ItemU, userID)$
/*To get the co-rated items by target user and active user*/
7: **for** $i = 1$ to k **do** /*k is the number of co-rated items*/
8: compute the trust value about the common item i, by Equation 3
9: **end for**
10: compute the direct trust value T_u^n by Equation 4
11: Trust_Array1 $\longleftarrow T_u^n$ /*To store the direct trust value into the matrix*/
12: **end repeat**
13: **end**

The function of this algorithm is to build trust relationship between the target user and the active user. Line 2 to 4 is to collect the co-rated items for the preparation of computing the direct trust value, and the line 5 to 12 is the core of building the direct trust. When calculating the correctness of active user's recommendation, we separately perform the recommendation process by using active user as the target user's sole recommendation partner.

The recommendation trust is to get the expressions about the active user from the target users' partners. For this purpose, the trust derivation algorithm is used to derive all possible direct trust relationships with the active user as trusted factor from a given set of initial trust relationships. The algorithm tries for each recommendation trust expression to derive as many new trust expressions as possible. Then the considered recommendation trust is removed from the set and the new recommendation trusts are inserted into it.

Algorithm 2

Input: target user, active user, the number of the target user's partner k, and the trust matrix Trust_Array1
Output: the value of the recommendation trust T_m^n
Step:
1: **begin**
2: **repeat**
3: Trust_List$\longleftarrow$ $query(u, Trust\_Array1)$
/*To get the trust set of the target user*/

4: **end repeat**
5: $Trust\_quene(key, w_i) \longleftarrow Sort(Trust\_List, k)$
/*To obtain the k nearest of the target user's neighbor that is order by the direct trust between users. The variable key is user's id, while the w_i is the value of the trust.*/
6: **for** i =1 to k **do**
7: $T(m_i, n, T^n_{m_i}) \longleftarrow indexOf(key, n, Trust\_Array1)$
/*To index the direct trust $T^n_{m_i}$ between the user m_i and active user n*/
8: **end for**
9: Compute the recommendation trust T^n_m by Equation 5
10: **end**

This algorithm consists of two parts. The first part, from line 2 to 4, is to get the trust set of the target user and prepare for computing the recommendation trust; the second part, from line 5 to 9, is to compute the recommendation trust of active user's.

Algorithm 3
Input: target user, active user
Output: the value of the combination trust $Trust^u_n$
Step:
1: **begin**
2: $T^u_n \longleftarrow T(u, n_j)$
3: $T^n_m \longleftarrow T(u, m)$
4: $Trust^u_n \longleftarrow \alpha T^u_n + \beta\ T^n_m$
5: $Trust\_Array2 \longleftarrow Trust^u_n$
6: **end**

This algorithm performs the combination of direct trust and recommendation trust, which is computed by Equation 6. As mentioned earlier, we assume that the value of each recommendation can be measured by the recommendation precision to reflect the reputation of active user, and the reputation value (trust value) will be import in the k-nearest neighbor algorithm.

4 User Trust-Based Collaborative Filtering Algorithm

4.1 Description of Algorithm

With the trust factor introduced into the traditional collaborative recommendation algorithm, there will be two factors to influence the result of prediction. One is the similarity of the users' tastes; the other is the trust value between users, which is the combination of direct trust and recommendation trust.

The steps of the trust-based collaborative recommendation algorithm are as follows.

(1) To generate the trust matrix Trust_Array2 of the users

$$Trust\_Array2 = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{n1} & T_{n2} & \cdots & T_{nn} \end{bmatrix}$$

The matrix is an n-order square that filled with trust value of the users, which has been calculated by offline. The element T_{mn} is the combination of direct and recommendation trust, which stands for the relation between the target user m and the active user n. Values symmetrically distributed on the both sides of the matrix diagonal.

(2) To generate the similarity matrix by Equation 1. Users with similar interest or behavior would be gathered (e.g., accessed the same type of information, purchased a similar set of products, liked or disliked a similar set of goods). Sim_{mn} similarity value between user u and n.

$$Sim = \begin{bmatrix} Sim_{11} & Sim_{12} & \cdots & Sim_{1n} \\ Sim_{21} & Sim_{22} & \cdots & Sim_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Sim_{n1} & Sim_{n2} & \cdots & Sim_{nn} \end{bmatrix}$$

(3) To calculate the combined similarity values, the simplest way to combine trust value with similarity to produce a compound weight that can be got by Resnick's formula. We use the CITEM algorithm introduced in [9] to generate the improved similarity value. This approach is a combination of trust-based weighting of producer's reputation with interesting similarity of producer profile ratings on the item.

$$W(u,n) = \frac{2(Sim(u,n))(Trust(u,n))}{Sim(u,n) + Trust(u,n)} \tag{7}$$

(4) The weighting approach simply adds the extra metric of trust to the standard similarity weighting in Resnick's prediction formula. The modified version of Resnick's formula is shown in Equation 8 to include the trust weighting.

$$pred(u,i) = \overline{R_u} + \frac{\sum_{Neu} W(u,n) * (R_{n,i} - \overline{R_n})}{\sum_{Neu}(|W(u,n)|)} \tag{8}$$

We use Resnick's algorithm to compute the prediction of item i for target user u. Where Neu is the set of k similar neighbors that have rated on item i; $R_{n,i}$ is the rating of i for neighbor n; $\overline{R_u}$ and $\overline{R_n}$ are the average ratings over all rated items for user u and user v, respectively.

According to the above-mentioned, we give the user trust-based collaborative filtering recommendation algorithm as follow.

Algorithm 4
Input: target user, trust matrix Trust_Array2
Output: the prediction rating of the item
Step:
1: **begin**
2: **for** i=1 to m **do** /*The 'm' is the number of the neighbor*/
3: $Item \longleftarrow getItem(u)$
/*Get the rating items of the target user*/
4: **for** j=i to m **do**
5: $Comm\_Item\_Array \longleftarrow getCommon_{I}tem()$
6: **end for**
7: $Comm\_List(u,n) \longleftarrow Comm\_Item\_Array$
8: $Sim \longleftarrow Sim(u,n)$
9: $Trust(u,n) \longleftarrow Trust\_Array2$
10: Compute the similarity $W(u,n)$ with Equation 7
11: $Sim\_List \longleftarrow W(u,n)$
12: **end for**
13: $Neu \longleftarrow Sort(Sim\_List)$
14: **for** i=1 to k **do**
15: Compute the prediction value $Pred(u,i)$
16: **end for**
17:**end**

This algorithm consists of two parts. The first part, from line 2 to 13, is the core of the algorithm. In this part, the first step is to build the trust relation of users using the algorithm 3; the second step is to get the neighbors of the target user with the improved algorithm. The second part, from line 14 to 16, is to predict the ratings of the items by Equation 8.

Using trust metrics in the traditional collaborative recommendation model can enhance the robustness of recommendation algorithm and improve the prediction accuracy of collaborative filtering in the face of malicious noise.

4.2 Algorithm Computational Complexity

The computational complexity of our user trust-based collaborative recommendation algorithm depends on the amount of time required to build the model and the amount of time to compute the recommendation using this model.

To build the model, we need to compute the similarity between each user u and all the other users, and select the most similar users. Let n be the number of users in a recommender system, m the number of items. We need to compute $n(n-1)$ similarities, so the time complexity for similarity computation is $O(n^2)$. Each potentially requires m operations, the time complexity is $O(m)$. According to Pearson's correlation coefficient Equation 1 , the time complexity is $O(n^2+m)$. In Algorithm 4, the user trust value used in Equation 7 can be computed via offline, it does not bring about additional computation. Thus our user trust-based collaborative recommendation algorithm does not increase the computational complexity.

5 Experimental Results

To evaluate our algorithm that combined user trust with the collaborative filtering approach. We have carried out our experiments and given a comparison with other collaborative filtering algorithm. The experimental condition: the hardware environment is PC of Intel Pentium IV 2.4GHz CPU and 1G RAM, Operating system is Windows XP Professional. Data stored by Mysql database, and algorithm programming with Java.

In our experiment we used the publicly available dataset provided by MovieLens Site (http://movielens.umn.edu/). The site is a Web-based recommendation system provided by GroupLens (http://www.grouplen.org) team. The dataset contains 100,000 ratings on 1682 movies by 943 users. All ratings are integer values between 1 and 5, where 1 is the lowest (disliked) and 5 is the highest (most liked). Our data includes all the users who have rated at least 20 movies.

5.1 Evaluation Metrics

The most used technique for evaluating Recommender Systems is based on leave-one-out. It is an offline technique that can be run on a previously acquired dataset and involves hiding one rating and then trying to predict it with a certain algorithm. The predicted rating is then compared with the real rating and the difference in absolute value is the prediction error. The procedure is repeated for all the ratings and an average of all the errors is computed, the Mean Absolute Error (MAE).

MAE is usually applied to measure the accuracy, which measures the average absolute difference between the predictions and the ratings over all items. The formula is as follows:

$$MAE = \frac{\sum_1^N |P_i - Q_i|}{N} \tag{9}$$

Where $N$is number of the rated items, and P_i is predicted rating on the item i, Q_i is rating of target user on item i.

5.2 Accuracy Analysis

To evaluate the recommendation precision of our algorithm, we have carried out the experiments with our user trust-based collaborative filtering recommendation algorithm, traditional collaborative filtering recommendation algorithm [10] and item-trust collaborative filtering recommendation algorithm [8].

Figure 1 shows the comparison of recommendation quality using different weighting factor α. The performance of our user trust-based collaborative recommendation algorithm is better than that of the user-based collaborative recommendation algorithm, and the different weight value of α performs different accuracy. Appropriate value of the weight can effectively balance direct trust and recommendation trust and avoid the similar users being hijacked by the user trust value, which may lead to a distortion recommendation.

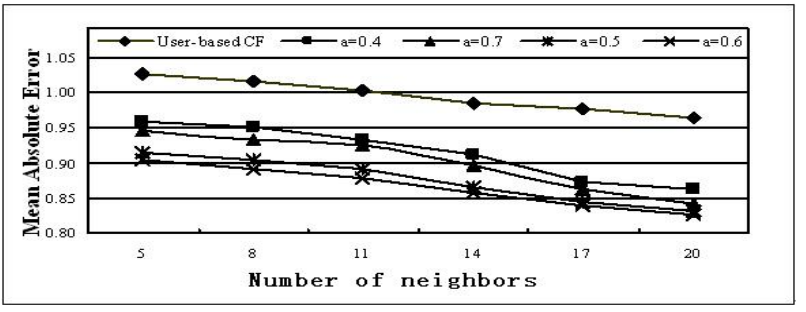

Fig. 1. Comparison of recommendation quality using different weighting factor

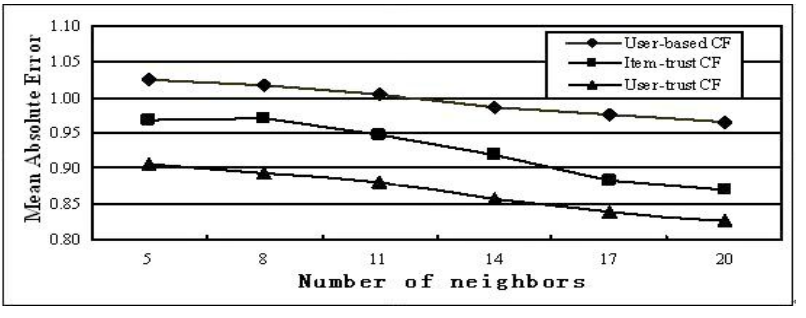

Fig. 2. Comparison of recommendation precision

Figure 2 shows the comparison of recommendation precision of three algorithms. The precision of item trust-based collaborative filtering algorithm and our user trust-based collaborative filtering algorithm is better than that of traditional user-based collaborative filtering algorithm, because they have considered the trust factor. The precision of our user trust-based collaborative filtering algorithm outperforms the item trust-based collaborative filtering algorithm. The performance is more remarkable with the number of neighbors increasing.

5.3 Robustness Analysis

In order to evaluate the robustness of the algorithm, we inserted some malicious ratings into the original data set. The rate of the malicious attack reached to 15%, namely the number of ratings achieved 252. But there were only 9.65% of the original users in the system reached the standard, under this condition there was only 91 ordinary user, a lower number compared with the malicious number. We selected 42 normal users as the test users whose ratings were over 320 so as to fully reflect the user's hobby with the rating items and avoid the negative effects caused by lack of user hobby information. Figure 3 shows the comparison of attack resist capability of the three algorithms under different attack size. And the size of neighbor users takes 11 when calculating MAE.

In Figure 3, although the malicious users only occupy the recommendation user 5%, the influence on recommendation algorithms is significantly obvious.

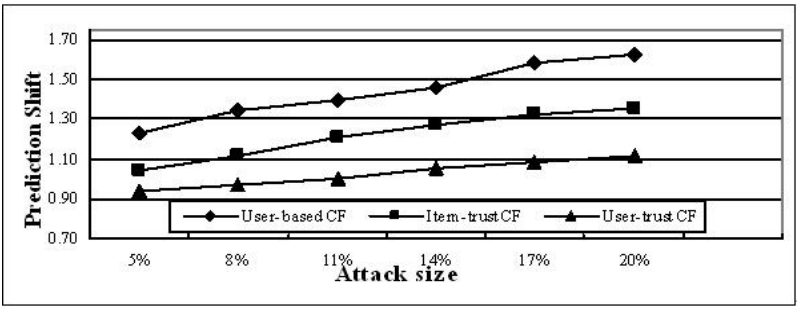

Fig. 3. Comparison of attack resist capability

Referencing figure 2 is not difficult to discover, the precision deviation separately expands from original (1.004, 0.947, 0.879) to (1.23, 1.04, 0.94). Because the traditional collaborative filtering algorithm generates recommendation only depends on the user ratings, the malicious users can provide a number of ratings which may become the similarity neighbors of the active user so as to have a great influence on its recommendation precision. Although item trust-based collaborative filtering algorithm and our user trust-based collaborative filtering algorithm also have some extent deviation, they introduce the trust pattern when they produce the recommendation neighbor, so they can effectively resist the influence of malicious noise. Thus the robustness of item trust-based and user trust-based collaborative filtering algorithm surpasses the traditional collaborative filtering algorithm in the face of malicious noise.

In Figure 3, when the attack size is smaller than 11%, the capability of attack resist for our user trust-based collaborative filtering algorithm outperforms the traditional collaborative filtering algorithm and item trust-based collaborative filtering algorithm. We notice that in general the user-trust approaches perform better than the item-trust approaches, especially the malicious attack size oversteps 11%. This is to be expected as the user-trust values provide a far more reliability of profile during recommendation and prediction. The reason of the phenomenon showed in Figure 3 is that the malicious users have a higher filled size with the rated items, and there would be more co-rating items with the target user, so that malicious users have more impact on the target users. But our user trust-based collaborative filtering algorithm is able to maintain a high precision in the face of malicious noise.

6 Conclusions

Traditional collaborative filtering recommendation algorithm is quite vulnerable in the face of malicious noise, which is a serious threat to the recommender systems that use collaborative filtering algorithm as an essential recommendation component. Recent research has showed that traditional collaborative recommendation algorithms can not ensure the precision of recommendations in the face of malicious noise.

In this paper, we have proposed an improved recommendation algorithm. It is an effective method to reduce the negative impact caused by malicious user ratings through the combination of user trust with the traditional user similarity. The user trust computation model and the corresponding user trust production algorithm are described. Combining user trust mechanism with traditional collaborative recommendation model can provide an effective way to defend against malicious noise. Experimental results have shown that the proposed algorithm is better in recommendation precision and resist-attack capability than the mentioned algorithm in this paper.

In the future we will improve our user trust-based collaborative recommendation algorithm and eliminate the vibration problem of the recommendation user trust value by introducing the detection mechanism of malicious user feature. This might be accomplished by filtering out bias recommendations or malicious neighbors in the recommendation algorithm.

References

1. Mobasher, B., Burke, R., Bhaumik, R., Sandvig, J.J.: Attacks and Remedies in Collaborative Recommendation. IEEE Intelligent Systems. J. 22(3), 56–63 (2007)
2. Williams, C.A., Mobasher, B., Burke, R.: Defending Recommender Systems: Detection of Profile Injection Attacks. J. Service Oriented Computing and Applications 1(3), 157–170 (2007)
3. Mehta, B., Nejdl, W.: Unsupervised strategies for shilling detection and robust collaborative filtering. J. User Modeling and User Adapted Interaction 19(1-2), 65–97 (2009)
4. Mehta, B., Hofmann, T.: A Survey of Attack-Resistant Collaborative Filtering Algorithms. J. Data Engineering Bulletin 31(2), 14–22 (2008)
5. Li, Y.-M., Kao, C.-P.: TREPPS:A Trust-based Recommender System for Peer Production Services. J. Expert Systems with Applications 36(2), 3263–3277 (2009)
6. Montaner, M., Lopez, B., de la Rosa, J.L.: Developing trust in recommender agents. In: 1st International Conference on Autonomous Agents, pp. 304–305. ACM Press, Bologna (2002)
7. Paolo, M., Paolo, A.: Trust-aware collaborative filtering for recommender systems. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 492–508. Springer, Heidelberg (2004)
8. John, O., Barry, S.: Trust in Recommender Systems. In: 10th International Conference on Intelligent User Interfaces, pp. 167–174. ACM Press, New York (2005)
9. John, O., Barry, S.: Is Trust Robust? An Analysis of Trust-Based Recommendation. In: 11th International Conference on Intelligent User Interfaces, pp. 101–108. ACM Press, New York (2006)
10. Deng, A., Zhu, Y., Shi, B.: A collaborative filtering recommendation algorithm based on item rating prediction. J. Journal of Software 14(9), 1621–1628 (2003) (in Chinese)
11. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
12. O'Mahony, M.P., Hurley, N.J., Silvestre, G.C.M.: Detecting noise in recommender system databases. In: 11th International Conference on Intelligent User Interfaces, pp. 109–115. ACM Press, New York (2006)