# A Robust Collaborative Filtering Recommendation Algorithm Based on Multidimensional Trust Model

Dongyan Jia, Fuzhi Zhang, Sai Liu
School of Information Science and Engineering, Yanshan University, Qinhuangdao, China
Email: jdy_1983@163.com, xjzfz@ysu.edu.cn, liusai19@sina.com

***Abstract*—Collaborative filtering is one of the widely used technologies in the e-commerce recommender systems. It can predict the interests of a user based on the rating information of many other users. But the traditional collaborative filtering recommendation algorithm has the problems such as lower recommendation precision and weaker robustness. To solve these problems, in this paper we present a robust collaborative filtering recommendation algorithm based on multidimensional trust model. Firstly, according to the rating information of users, a multidimensional trust model is proposed. It measures the credibility of user's ratings from the following three aspects: the reliability of item recommendation, the rating similarity and the user's trustworthiness. Secondly, the computational model of trust and the traditional collaborative filtering approach are combined to select the reliable neighbor set and generate recommendation for the target user. Finally, the performances of the novel algorithm with others are compared from both sides of recommendation precision and robustness using MovieLens dataset. Compared with the existing algorithms, the proposed algorithm not only improves the quality of neighbor selection and the recommendation precision, but also has better robustness.**

***Index Terms*—multidimensional trust model, robustness, collaborative filtering, recommender system**

## I. INTRODUCTION

Recommender systems, as a kind of information filtering technology, have provided an effective way to solve the information overload problem [1]. Specially, the collaborative filtering [2] is one of the most successful recommendation technologies. It generates recommendation for the target user by collecting the preference information of similar users. However, due to the sparsity of ratings, the quality of neighbor selection for target user is poor based on the similarity between users. In addition, with the emergence of shilling attacks [3,4] and the lack of credibility evaluation mechanism of ratings, how to improve the recommendation precision and robustness has become the key issue to be solved.

Manuscript received January 21, 2012; revised June 1, 2012; accepted June 27, 2012.
Corresponding author: Fuzhi Zhang.

In order to measure the credibility of users' ratings and the degree of trust between users, many computational models of trust have been proposed. O'Donovan et al. [5] proposed the profile-level and item-level computational model of trust and drew a conclusion that the latter performs better than the former by conducting experiments. Similarly, Lathia et al. [6] proposed an improved computational model of trust, which computed the degree of trust of target user to the recommender user based on the error of predict rating. However, both of the models generate recommendation relying on the similarity between two users. Due to the extreme sparsity of ratings, it is very difficult to compute the similarity between two users accurately, which leads to the inaccurate computation of degree of trust, poor scalability and inapplicable in large-scale dataset. Pitsilis et al. [7,8] analyzed the trust relationship between users from the point of view of subjective logic and proposed a computational model of trust based on the theory of uncertain probabilities. But the computation of uncertainty is based on the co-rated items between users. Consequently, it can't compute the degree of trust between users accurately in the case of the extreme sparsity of ratings. Aims at the limitations of traditional collaborative filtering recommendation algorithm when selecting neighbors, Kwon et al. [9] proposed a multidimensional credibility model based on the source credibility theory [10]. They analyzed and measured the degree of trust from three aspects such as the expertise, the trustworthiness and the similarity. However, it only takes into account the heterogeneous of ratings of users and still has the vulnerability when there are attack profiles in the system.

Recently, the model-based recommendation algorithms have attracted significant attention. These algorithms use statistical methods or techniques of machine learning to construct a recommendation model of which the parameters are estimated from the rating data of users. The recommendation is generated for the target user based on the model. Jamali et al. [11] proposed a random walk model named TrustWalker, in which the predict rating for the target user on the target item would be the expected value of ratings returned by performing many random walks. But this model is greatly affected by the sparsity of ratings. Ma et al. [12] proposed a


doi:10.4304/jsw.8.1.11-18

recommendation approach based on matrix factorization named RSTE. The target user gets the recommendation by learning the latent user and item features. Moreover, they proposed the recommendation approach by incorporating social contextual information [13], and applied RSTE to the recommendation based on the implicit social relations [14]. However, this recommendation approach based on matrix factorization is greatly affected by the sparsity of direct trust information.

Aim at the problems mentioned above, on the basis of the previous work, we propose a multidimensional trust model-based robust recommendation algorithm (MTMRRA). It measures the credibility of ratings of users from different aspects. As a result, the best neighbors are selected to generate recommendation for the target user according to the computational model of trust. Our contributions are as follows.

Firstly, a multidimensional trust model is proposed, which measures the credibility of users' ratings from the reliability of item recommendation, the rating similarity and the user's trustworthiness based on the user-item rating matrix. So the degree of trust between users is regarded as the sum of product of each attribute and its importance weight.

Secondly, a robust collaborative filtering recommendation algorithm is presented. Based on the proposed model of trust, we can choose the best neighbors for the target user, and then get the recommendation by combining the traditional collaborative filtering recommendation approach.

Thirdly, we conduct the experiments on the MovieLens dataset and compare the proposed algorithm with others in terms of the MAE, RMSE and Prediction Shift metrics. Experimental results indicate that our algorithm not only improves the recommendation precision, but also has better robustness.

## II. BACKGROUND

### *A. Description of User Rating Information*

In collaborative filtering recommender systems, the rating database includes a set of $m$ users, $U=\{u_1,u_2,\ldots,u_m\}$, and a set of $n$ items, $I=\{i_1,i_2,\ldots,i_n\}$. Users rate some items they know with a discrete range of possible values $\{min,\ldots,max\}$, for example, $\{1,\ldots,5\}$ or $\{1,\ldots,10\}$. Usually, the items with higher values are the user's favorite ones. So the user-item rating matrix can be described as:

$$\boldsymbol{R}=\begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,n} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ R_{m,1} & R_{m,2} & \cdots & R_{m,n} \end{bmatrix},$$

where, $R_{i,j}(1\le i\le m, 1\le j\le n)$ is the rating of user $u_i$ on item $i_j$.

Due to the large number of items, each user often only rated a certain number of items. If user $u_i$ hasn't rated the item $i_j$, we represent that as $R_{i,j}=\phi$.

### *B. Similarity Measures*

The most popular approaches of computing user similarity are cosine-based similarity and the Person correlation coefficient [15].

In cosine-based similarity approach, the ratings of each user are treated as one vector in $n$-dimensional space. Let the vector $\boldsymbol{U_i}$ and $\boldsymbol{U_j}$ denote the ratings of user $u_i$ and $u_j$ respectively, so the similarity between user $u_i$ and $u_j$ can be measured as:

$$sim(u_i,u_j)=\cos(\boldsymbol{U}_i,\boldsymbol{U}_j)=\frac{\boldsymbol{U}_i\cdot\boldsymbol{U}_j}{\|\boldsymbol{U}_i\|\|\boldsymbol{U}_j\|}. \quad (1)$$

Using Person correlation coefficient, the similarity between user $u_i$ and $u_j$ can be measured as:

$$sim(u_i,u_j)=\frac{\sum_{i_k\in I_{ij}}(R_{i,k}-\bar{R}_i)(R_{j,k}-\bar{R}_j)}{\sqrt{\sum_{i_k\in I_{ij}}(R_{i,k}-\bar{R}_i)^2}\sqrt{\sum_{i_k\in I_{ij}}(R_{j,k}-\bar{R}_j)^2}}, \quad (2)$$

where $I_{ij}$ is the item set co-rated by user $u_i$ and $u_j$, $R_{i,k}$ and $R_{j,k}$ are the ratings of user $u_i$ and $u_j$ on item $i_k$ respectively, $\bar{R}_i$ and $\bar{R}_j$ are the average ratings of user $u_i$ and $u_j$ respectively.

### *C. Prediction*

Assume the target user is $u_a$, the target item is $i_j$, so the main idea of traditional user-based collaborative filtering recommendation algorithm is as follow: based on the user-item rating matrix, the users who have rated the item $i_j$ are selected, and the rating similarity between the target user and each of these users is computed by using the similarity measures. Then the top-$k$ users who have larger user similarity are chosen as the neighbors of target user $u_a$. As a result, according to the rating information of neighbors, the predict rating $P_{a,j}$ for user $u_a$ on item $i_j$ is computed as:

$$P_{a,j}=\bar{R}_a+\frac{\sum_{u_k\in N(u_a)}(R_{k,j}-\bar{R}_k)\cdot sim(u_a,u_k)}{\sum_{u_k\in N(u_a)}|sim(u_a,u_k)|}, \quad (3)$$

where $N(u_a)$ is the neighbor set of target user $u_a$, $\bar{R}_a$ and $\bar{R}_k$ are the average ratings of target user $u_a$ and the neighbor $u_k$ respectively, $R_{k,j}$ is the rating of $u_k$ on item $i_j$, $sim(u_a,u_k)$ is the similarity between the target user $u_a$ and the neighbor $u_k$.

## III. MULTIDIMENSIONAL TRUST MODEL

Due to the sparsity of user-item rating matrix and the shilling attacks in collaborative filtering recommender systems, it is unreliable to select neighbors for the target user according to the similarity between users. As a result, the user's satisfaction for the recommendation results declines. To improve the quality of selected neighbors, we propose a multidimensional trust model which analyses and measures the credibility of users' ratings using the reliability of item recommendation, the rating similarity and the user's trustworthiness.

### A. *Reliability of Item Recommendation*

**Definition 1.** The reliability of item recommendation is defined as the degree of a user to provide an accurate prediction for every item. For user $u_k \in U$ and the item set rated by $u_k$, $I_k = \{i_j \mid R_{k,j} \neq \phi, i_j \in I\}$, the user $u_k$'s reliability of recommendation for item $i_j \in I_k$ is described as $R_k^j$.

To measure the reliability of item recommendation, we employ the item-level computational model of trust proposed by O'Donovan. Using the "leave-one-out" approach, choosing $\forall u_k \in U$ as the only recommender user, every item $i_j \in I_k$ as target item, and every user $u_a$ in the user set $U_j = \{u_a \mid R_{a,j} \neq \phi, u_a \in U, u_a \neq u_k\}$ as target user, we can compute the predict rating for the target user on the target item using (3).

Based on the deviation between the predict rating and the actual rating, we can compute the user $u_k$'s reliability of recommendation for item $i_j$ as:

$$R_k^j = \frac{\sum_{a=1}^{|U_j|} sat_{a,k}^j}{|U_j|}, \tag{4}$$

$$sat_{a,k}^j = \begin{cases} 1 & \left|P_{a,j} - R_{a,j}\right| \le \varepsilon \\ 0 & else \end{cases}, \tag{5}$$

where $P_{a,j}$ is the predict rating for the user $u_a$ on item $i_j$, $R_{a,j}$ is the actual rating of user $u_a$ on item $i_j$, $\varepsilon$ is a threshold, we set $\varepsilon$=1.2 in this paper.

### B. *Rating Similarity*

**Definition 2.** Rating similarity is defined as the similarity between two users. For user $u_a \in U$ and user $u_b \in U$, the rating similarity between the two users is computed based on the item set $I_{ab} = \{i_k \mid R_{a,k} \neq \phi, R_{b,k} \neq \phi, i_k \in I\}$, and it is described as $S_{a,b}$.

The traditional similarity measures rely on the items co-rated between users. Due to the sparity of user-item rating matrix, the computation of similarity has greater occasionality. To reduce its impact, we employ a relevance weighting function $f(x)$ and set a threshold for the number of co-rated items between two users by specifying the $k$ value:

$$f(x) = \frac{1}{1 + e^{-\frac{x}{k}}}. \tag{6}$$

The similarity $S_{a,b}$ between user $u_a$ and user $u_b$ is computed as:

$$S_{a,b} = sim(u_a, u_b) \times \frac{1}{1 + e^{-\frac{|I_{ab}|}{k}}}, \tag{7}$$

where $sim(u_a, u_b)$ is calculated according to (2), $|I_{ab}|$ is the number of items co-rated by user $u_a$ and user $u_b$, $k$ is a threshold, the method of setting its value is as follows.

Let $k$=1, 2 , … ,5, the curve of $f(x)$ is shown in Fig. 1.

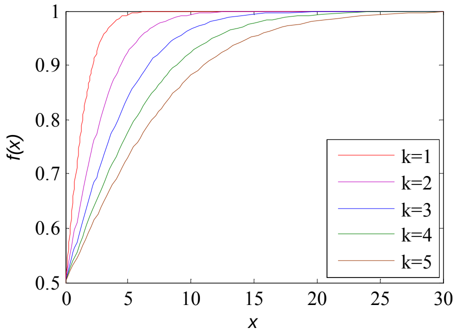



Figure 1. The curve of $f(x)$ with $k$=1,2,…,5

As shown in Figure 1, no matter what value the $k$ is, the $f(x)$ will be close to 1 infinitely when $x$ is greater than a certain value $x_0$. We can also get $S_{a,b} \approx sim(u_a, u_b)$ using (7). So we call $x_0$ the threshold of the number of co-rated items between users. Table 1 gives the comparison of value of $x_0$ when $k$ takes different values.

TABLE I.
COMPARISON OF THE VALUE OF $X_0$ WITH DIFFERENT $K$ VALUE

| $k$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $x_0$ | 6 | 11 | 16 | 22 | 27 |

Table 1 shows that with $k$ increasing gradually, the value of $x_0$ increases. Considering the sparity of user-item rating matrix, we set $x_0$ =16 and $k$ =3 to compute the rating similarity between two users.

### C. *User's Trustworthiness*

**Definition 3.** Trustworthiness of a user is defined as the degree of his ratings that reflect the user's actual opinions. For user $u_b \in U$ and the item set rated by $u_b$, $I_b = \{i_j \mid R_{b,j} \neq \phi, i_j \in I\}$, we can compute $u_b$'s trustworthiness $T_b$ using the information of the similarity of any two items in $I_b$ and the $u_b$'s ratings on the corresponding items.

Based on the user-item rating matrix, the similarity between two items is computed by using Person correlation coefficient:

$$sim(i_i, i_j) = \frac{\sum_{u_k \in U_{i,j}} (R_{k,i} - \bar{R}_i)(R_{k,j} - \bar{R}_j)}{\sqrt{\sum_{u_k \in U_{i,j}} (R_{k,i} - \bar{R}_i)^2} \sqrt{\sum_{u_k \in U_{i,j}} (R_{k,j} - \bar{R}_j)^2}}, \tag{8}$$

where $sim(i_i, i_j)$ is the similarity between item $i_i$ and item $i_j$, $U_{i,j}$ is the set of users who have both rated the item $i_i$ and item $i_j$, $U_{i,j} = \{u_k \mid R_{k,i} \neq \phi, R_{k,j} \neq \phi, u_k \in U\}$, $R_{k,i}$ and $R_{k,j}$ are the ratings of $u_k$ on item $i_i$ and item $i_j$ respectively, $\bar{R}_i$ and $\bar{R}_j$ are the average rating of item $i_i$ and $i_j$ respectively.

We can get a real value in the range [-1,+1] from (8), which is mapped to the range [0,1] by using $sim(i_i, i_j)' = \frac{1 + sim(i_i, i_j)}{2}$.

Consequently, the user $u_b$'s trustworthiness $T_b$ is computed as:

$$T_b = \frac{2}{|I_b|(|I_b|-1)}\sum_{i\in I_b, j\in I_b} t_{i,j}^b , \quad (9)$$

$$t_{i,j}^b = 1-[sim(i_i,i_j)'+\frac{|R_{b,i}-R_{b,j}|}{5}-1]^2 , \quad (10)$$

where $t_{i,j}^b$ is the trustworthiness of $u_b$ for item pair ($i_i$,$i_j$), $sim(i_i,i_j)'$ is the similarity between item $i_i$ and item $i_j$, $R_{b,i}$ and $R_{b,j}$ are the ratings of $u_b$ on item $i_i$ and item $i_j$ respectively.

### D. Computation of Trust Degree

Based on the analysis above, we can compute the degree of trust of user $u_a$ to user $u_b$ as:

$$trust_{a,b} = \alpha R_b^j + \beta S_{a,b} + \gamma T_b , \quad (11)$$

where $R_b^j$ is the reliability of item recommendation of user $u_b$ for item $i_j$, $S_{a,b}$ is the rating similarity between user $u_a$ and user $u_b$, $T_b$ is the trustworthiness of user $u_b$, $\alpha$ , $\beta$ , $\gamma$ are the importance weights of each attribute above, we can set their values according to the following method.

Using the experimental dataset, we can simulate the performance of four recommendation strategies as follows:

CF - Traditional user-based collaborative filtering recommendation strategy.

RCF - Reliability of item recommendation-based collaborative filtering recommendation strategy.

SCF - Rating similarity-based collaborative filtering recommendation strategy.

TCF - User's trustworthiness-based collaborative filtering recommendation strategy.

With the number of neighbors increasing, we can get the MAE values of each recommendation strategy respectively. Compared with the recommendation precision of CF, the percentage of improvement for each recommendation strategy is calculated. Let $p_{rcf}$, $p_{scf}$ and $p_{tcf}$ be the percentage of improvement for recommendation strategy of RCF, SCF and TCF respectively, so we have:

$$\alpha = \frac{p_{rcf}}{p_{rcf}+p_{scf}+p_{tcf}} ,$$

$$\beta = \frac{p_{scf}}{p_{rcf}+p_{scf}+p_{tcf}} ,$$

$$\gamma = \frac{p_{tcf}}{p_{rcf}+p_{scf}+p_{tcf}} .$$

Obviously, the greater percentage of improvement the recommendation strategy has, the larger the corresponding importance weight is. Considering the values of $\alpha$ , $\beta$ and $\gamma$ are different using different dataset, so their values should be computed dynamically.

Let's take an example to illustrate the computational process of the values of $\alpha$ , $\beta$ and $\gamma$ . Using the MovieLens[1] dataset, we randomly select 754 users' profiles as the training set and the remaining as the test set to conduct the experiments and compare the performances of the RCF, SCF and TCF with CF. Table 2 shows the comparison of recommendation precision with different number of neighbors.

TABLE II.
COMPARISON OF RECOMMENDATION PRECISION (MAE)

| number of neighbors | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| CF | 0.7254 | 0.7331 | 0.7287 | 0.7235 | 0.7091 | 0.7061 | 0.7096 | 0.7054 | 0.7045 | 0.7010 |
| RCF | 0.6309 | 0.6478 | 0.6506 | 0.6681 | 0.6676 | 0.6527 | 0.6600 | 0.6610 | 0.6648 | 0.6585 |
| SCF | 0.7276 | 0.7324 | 0.7275 | 0.7203 | 0.7055 | 0.7066 | 0.7071 | 0.7048 | 0.7034 | 0.6998 |
| TCF | 0.7008 | 0.6911 | 0.6964 | 0.6805 | 0.6860 | 0.6825 | 0.6841 | 0.6801 | 0.6892 | 0.6857 |

As shown in Table 2, the recommendation strategy of RCF, SCF and TCF all outperform the CF in terms of recommendation precision. Compared with the CF strategy, the average percentage of improvement for the RCF, SCF and TCF is 8.06%, 0.16% and 3.76% respectively. So:

$$\alpha = \frac{8.06\%}{8.06\%+0.16\%+3.76\%} = 0.6728 ,$$

$$\beta = \frac{0.16\%}{8.06\%+0.16\%+3.76\%} = 0.0134 ,$$

$$\gamma = \frac{3.76\%}{8.06\%+0.16\%+3.76\%} = 0.3139 .$$

## IV. Multidimensional Trust Model-based Robust Recommendation Algorithm

### A. Description of Algorithm

To improve the recommendation precision, a multidimensional trust model-based robust recommendation algorithm (MTMRRA) is proposed. The main steps of MTMRRA algorithm are described as follows: according to the rating information of users, select the user set $C(u_a)$ who have rated the target item $i_j$ and use (11) to compute the degree of trust of target user $u_a$ to each user in $C(u_a)$. Based on that, select the top-$k$ users as the neighbors of target user $u_a$ and compute the predicted rating $P_{a,j}$ for the target user $u_a$ on the target item $i_j$ as:

[1] http://www.grouplens.org/node/73

$$P_{a,j} = \overline{R}_a + \frac{\sum_{u_k \in U^{T+}} (R_{k,j} - \overline{R}_k) \cdot S_{a,k}}{\sum_{u_k \in U^{T+}} \left| S_{a,k} \right|}, \qquad (12)$$

where $U^{T+}$ is the neighbor set of target user $u_a$, $U^{T+} = U^T \cap U^+$, $U^T = \{u_k \mid trust_{a,k} \geq T, \forall u_k \in U\}$, $U^+ = \{u_k \mid S_{a,k} > 0, \forall u_k \in U\}$, $T$ is the threshold of degree of trust between users, $S_{a,k}$ is the similarity between target user $u_a$ and neighbor user $u_k$, $R_{k,j}$ is the rating of $u_k$ on the target item $i_j$, $\overline{R}_a$ and $\overline{R}_k$ are the average ratings of the target user $u_a$ and the neighbor $u_k$ respectively.

According to the steps of algorithm above, MTMRRA algorithm is described as follows.

**Algorithm** : MTMRRA

**Input**: the user-item rating matrix $\boldsymbol{R}$

**Output**: the predicted rating $P_{a,j}$ for target user $u_a$ on the target item $i_j$

**Begin**

1: $C(u_a) \leftarrow \{u_k \mid R_{k,j} \neq \phi, u_k \in U\}$ ;
2: **for** each $u_k \in C(u_a)$ **do**
3: $I_k \leftarrow \{i_b \mid R_{k,b} \neq \phi, i_b \in I\}$ ;
4: $U_j \leftarrow \{u_t \mid R_{t,j} \neq \phi, u_t \in U, u_t \neq u_k\}$ ;
5: $sum\_satisfactory \leftarrow 0$ ;
6: **for** each $u_t \in U_j$ **do**
7: $P_{t,j} \leftarrow Predict(u_t, u_k, i_j)$ ;
8: $sat_{t,k}^j \leftarrow Satisfactory(u_t, u_k, i_j)$ ;
9: $sum\_satisfactory \leftarrow sum\_satisfactory + sat_{t,k}^j$ ;
10: **end for**
11: $R_k^j \leftarrow \dfrac{sum\_satisfactory}{|U_j|}$ ;
12: $I_{ak} \leftarrow \{i_b \mid R_{a,b} \neq \phi, R_{k,b} \neq \phi, i_b \in I\}$ ;
13: $sim(u_k, u_a) \leftarrow similarity(u_k, u_a)$ ;
14: $S_{a,k} \leftarrow sim(u_k, u_a) \times f(|I_{ak}|)$ ;
15: $sum\_trustworthy \leftarrow 0$ ;
16: **for** $\forall i_b, i_t \in I_k (i_b \neq i_t)$ **do**
17: $sim(i_b, i_t) \leftarrow similarity(i_b, i_t)$ ;
18: $t_{b,t}^k \leftarrow trustworthy(u_k, i_b, i_t)$ ;
19: $sum\_trustworthy \leftarrow sum\_trustworthy + t_{b,t}^k$ ;
20: **end for**
21: $T_k \leftarrow \dfrac{2 \times sum\_trustworthy}{|I_k|(|I_k|-1)}$ ;
22: $trust_{a,k} \leftarrow \alpha R_k^j + \beta S_{a,k} + \gamma T_k$ ;
23:**end for**
24: $N(u_a) \leftarrow \{u_k \mid S_{a,k} > 0, trust_{a,k} > T, u_k \in C(u_a)\}$ ;
25:sort the degree of trust of the target user $u_a$ to every user in the $N(u_a)$;
26: $U^{T+} \leftarrow \{u_i \mid u_i \in N(u_a), i = 1, 2, ..., k\}$ ;
27: $P_{a,j} \leftarrow Predict\_MTRRA(u_a, i_j)$ ;
28: **return** $P_{a,j}$;

**End**

This algorithm consists of three parts. The first part, the first line, is to get the user set $C(u_a)$ who have rated the target item. The second part, from line 2 to 23, is to compute the degree of trust of the target user to each user in $C(u_a)$. The third part, from line 24 to 28, is to select the neighbors for the target user and compute the predicted rating $P_{a,j}$ for target user $u_a$ on the target item $i_j$.

### *B. Complexity Analysis*

In the process of computing the degree of trust of target user $u_a$ to each user in $C(u_a)$, the computation of the reliability of item recommendation, the rating similarity and the user's trustworthiness is of complexity $O(m)$, $O(1)$ and $O(l^2)$ respectively, where $m$ denotes the number of users in the recommender system, $l$ denotes the number of ratings rated by one user. In the actual recommender system, the degree of trust between users is usually computed off-line, so its complexity is $O(1)$. The complexity of selecting neighbors and computing the predict ratings for the target user is $O(m^2)$ and $O(k)$ respectively, where $k$ denotes the number of neighbors. So the total complexity of computation online is $O(m^2+k)$. Considering the fact that we often have $k \ll m$, so the complexity of the algorithm MTMRRA is $O(m^2)$.

### *C. Proof of Correctness*

The correctness of the algorithm MTMRRA is proved from both sides of theoretical study and experimental analysis. Theoretically, the correctness of MTMRRA depends on the correct computation of the degree of trust of target user to others. The computational approach of trust proposed in this paper measures the credibility of user's ratings from three aspects, which not only considers the potential relationship between user profiles, but also evaluates the credibility of user profile itself from both sides of horizontal and vertical. The algorithm MTMRRA makes a multi-dimensional analysis for the degree of trust between users, which is feasible and correct in theory. From the experiment point of view, the algorithm is correct in case that compared to other recommendation algorithms, it not only improves the recommendation precision, but also has better robustness. We can see that the MTMRRA is correct obviously through section Ⅵ.

## VI EXPERIMENTAL RESULTS AND ANALYSIS

### *A. Dataset*

In our experiments we use the publicly available dataset provided by MovieLens site (http://movielens.umn.edu/), which 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). For the number of items which one user has rated in the dataset, the minimum is 20, the maximum is 737 and the average number is 106. In the experiment, we divide the dataset into two groups, 80% are used as the training set and the remaining 20% are used as the test set.

### *B. Evaluation Metrics*

We use MAE and RMSE to evaluate the recommendation precision of algorithm, which are computed by measuring the deviation between the prediction rating and the actual rating. Obviously, the smaller MAE (or RMSE) is, the higher the recommendation precision of algorithm is. MAE and RMSE can be computed as: [16]

$$MAE = \frac{\sum_{j=1}^{n}|p_j - r_j|}{n}, \tag{14}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_j - r_j)^2}{n}}, \tag{15}$$

where $p_j$ is the predicted rating given to the target user on target item $i_j$, $r_j$ is the actual rating of the target user on target item $i_j$, $n$ is the total number of prediction.

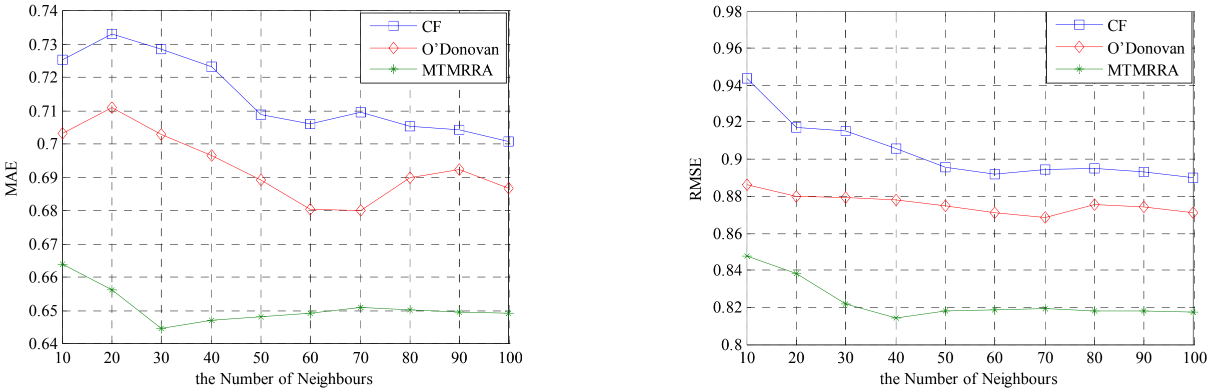
Moreover, we use MAE, RMSE and *Prediction Shift* to evaluate the robustness of algorithm. The *Prediction Shift* measures the deviation of prediction on the attacked item (before and after attack) of the recommendation algorithm. The smaller the *Prediction Shift* is, the better robustness the algorithm has. It is computed as: [17]

$$\Pr edShift(u,i) = \frac{1}{n}\sum p^{'}(u_k,i_j) - p(u_k,i_j), \tag{16}$$

where $p(u_k,i_j)$ and $p^{'}(u_k,i_j)$ are the predicted ratings for user $u_k$ on item $i_j$ before and after the item $i_j$ is attacked, $n$ is the total number of prediction.

### *C. Recommendation Precision Analysis*

To evaluate the recommendation precision of algorithms, we have carried out the experiments with our multidimensional trust model-based robust recommendation algorithm (MTMRRA), the traditional collaborative filtering recommendation algorithm (CF) and the recommendation algorithm proposed by O'Donovan. With the number of neighbors increasing gradually, Fig. 2 gives the comparisons of recommendation precision.




(a) MAE (b) RMSE

Figure 2. Comparison of recommendation precision

As shown in Fig. 2, the MTMRRA algorithm outperforms the CF algorithm and the O'Donovan's in term of recommendation precision. Compared with CF algorithm, MTMRRA algorithm improves by 9.18%(MAE) and 8.95% (RMSE); compared with O'Donovan's algorithm, MTMRRA algorithm improves by 6.39% (MAE) and 6.01% (RMSE). That is to say that it is helpful to improve the quality of chosen neighbors and the recommendation precision by measuring the credibility of users' ratings from several aspects. Therefore, our MTMRRA algorithm has better recommendation precision.

### *D. Robustness Analysis*

To evaluate the robustness of the algorithms, we inject some hybrid attack profiles (the same number of profiles of random attack, average attack and bandwagon attack) into the training set. Let the filler size be 1%, 3%, 5%, 10%, 25%, the attack size be 1%, 2%, 5%, 10%, and the number of neighbors be 40, with the filler size and attack size increasing gradually, the comparisons of recommendation precision of algorithm MTMRRA, CF and O'Donovan's are shown in Table 3 and Table 4.

TABLE III.
COMPARISON OF MAE VALUES

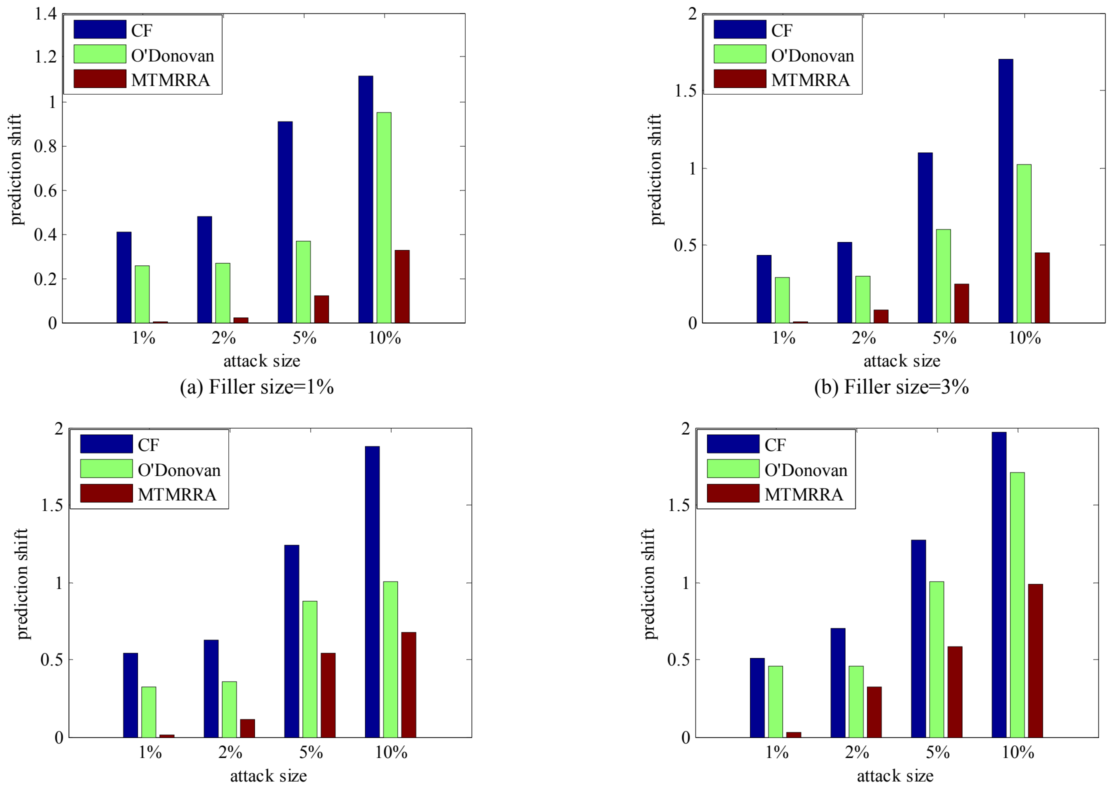
| attack size | 1% | | | 2% | | | 5% | | | 10% | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| algorithm | CF | O'Donovan | MTMRRA | CF | O'Donovan | MTMRRA | CF | O'Donovan | MTMRRA | CF | O'Donovan | MTMRRA |
| filler size=1% | 0.7374 | 0.7028 | 0.6705 | 0.7359 | 0.7041 | 0.6770 | 0.7478 | 0.7057 | 0.6446 | 0.7642 | 0.7122 | 0.6783 |
| filler size=3% | 0.7337 | 0.6984 | 0.6755 | 0.7411 | 0.7046 | 0.6730 | 0.7441 | 0.7061 | 0.6717 | 0.7534 | 0.7104 | 0.6964 |
| filler size=5% | 0.7371 | 0.7015 | 0.6760 | 0.7409 | 0.7056 | 0.6707 | 0.7541 | 0.7303 | 0.6694 | 0.7651 | 0.7405 | 0.6918 |
| filler size=10% | 0.7340 | 0.7078 | 0.6596 | 0.7454 | 0.7143 | 0.6619 | 0.7550 | 0.7093 | 0.6780 | 0.7458 | 0.7092 | 0.6672 |
| filler size=25% | 0.7219 | 0.6963 | 0.6510 | 0.7488 | 0.7106 | 0.6563 | 0.7309 | 0.6976 | 0.6686 | 0.7381 | 0.7122 | 0.6869 |

TABLE IV.
COMPARISON OF RMSE VALUES

| attack size | 1% | | | 2% | | | 5% | | | 10% | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| algorithm | CF | O'Donovan | MTMRRA | CF | O'Donovan | MTMRRA | CF | O'Donovan | MTMRRA | CF | O'Donovan | MTMRRA |
| filler size=1% | 0.9130 | 0.8823 | 0.8417 | 0.9143 | 0.8805 | 0.8494 | 0.9324 | 0.8814 | 0.8113 | 0.9621 | 0.8941 | 0.8422 |
| filler size=3% | 0.9135 | 0.8790 | 0.8480 | 0.9205 | 0.8806 | 0.8482 | 0.9248 | 0.8790 | 0.8400 | 0.9405 | 0.8816 | 0.8691 |
| filler size=5% | 0.9139 | 0.8821 | 0.8497 | 0.9137 | 0.8812 | 0.8494 | 0.9378 | 0.9053 | 0.8384 | 0.9460 | 0.9193 | 0.8723 |
| filler size=10% | 0.9070 | 0.8775 | 0.8224 | 0.9107 | 0.8802 | 0.8213 | 0.9157 | 0.8700 | 0.8475 | 0.9248 | 0.8682 | 0.8294 |
| filler size=25% | 0.8959 | 0.8685 | 0.8221 | 0.9183 | 0.8820 | 0.8261 | 0.8877 | 0.8595 | 0.8245 | 0.8942 | 0.8688 | 0.8560 |

As shown in Table 3 and Table 4, the MTMRRA algorithm outperforms the CF algorithm and the O'Donovan's algorithm in term of recommendation precision whatever the attack size is. Also, with the attack size increasing gradually, the recommendation precision of three algorithms follows to descend. It is now clear that the more attack users in the system, the lower quality of recommendation. Compared with the recommendation precision of CF, MTMRRA algorithm improves by 9.74%(MAE) and 8.56% (RMSE); compared with the recommendation precision of O'Donovan's, MTMRRA algorithm improves by 5.32% (MAE) and 4.6% (RMSE). It can be proved that our MTMRRA algorithm has better robustness.

Under the hybrid attack, let the filler size be 1%, 3%, 5%, 10% and the attack size be 1%, 2%, 5%, 10%, with the attack size increasing gradually, the comparisons of prediction shift are shown in Figure 3.




(a) Filler size=1% (b) Filler size=3%

(c) Filler size=5% (d) Filler size=10%

Figure 3. Comparison of prediction shift with different filler size

As shown in Fig. 3, under the same filler size, with the attack size increasing gradually, the prediction shift of all algorithms increases. So the more attack users in the system, the lower quality of recommendation. In addition, under the same attack size and filler size, the MTMRRA algorithm outperforms the CF and O'Donovan's in terms of prediction shift.

## VII CONCLUSIONS

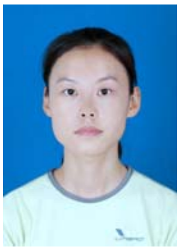
With the wide application of the collaborative filtering algorithm in e-commerce, how to improve the recommendation precision and the robustness becomes more and more important. We have made some beneficial explorations in this area. In this paper we propose a multidimensional trust model which measures the credibility of users' ratings from three attributes. Based on the model of trust, we present a robust collaborative filtering recommendation algorithm. Compared with other algorithms, the proposed algorithm not only improves the recommendation precision, but also has better robustness. The experiments conducted on MovieLens prove the effectiveness of the proposed algorithm. With the new ratings adding into the system gradually, how to design an incremental recommendation algorithm and generate recommendation for the target user accurately will be our future work.

## ACKNOWLEDGMENT

The work described in this paper was supported by the Natural Science Foundation of Hebei Province, China (No. F2011203219).

## REFERENCES

[1] H.L. Xu, X. Wu, X.D. Li, B.P. Yan, “Comparison Study of Internet Recommendation System”, Journal of Software, 20(2), pp. 350-362, 2009 (in Chinese).
[2] Y. Koren, R. Bell, *Recommender Systems Handbook*, Springer, 2011, pp. 145-186.
[3] B. Mobasher, R. Burke, R.Bhaumik, C. Williams, “Toward Trustworthy Recommender Systems: an Analysis of Attack Models and Algorithm Robustness”, ACM Transactions on Internet Technology, 7(4), pp. 1-38, 2007.
[4] B. Mobasher, R. Burke, R. Bhaumik, and J.J. Sandvig, “Attacks and Remedies in Collaborative Recommendation”, IEEE Intelligent Systems, 22(3), pp. 56-63, 2007.
[5] J. O’Donovan, B. Smyth, “Trust in Recommender Systems”, In: Proc. of the 10th Int’l Conf. on Intelligent User Interfaces. San Diego, California, USA: ACM Press, 2005, pp. 167-174.
[6] N. Lathia, S. Hailes, L. Capra, “Trust-based Collaborative Filtering”, In Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM). Trondheim, Norway: Springer, 2008, pp. 119-134.
[7] G. Pitsilis. L.F. Marshall, “A Model of Trust Derivation from Evidence for Use in Recommendation Systems”, Technical Report Series, CS-TR-874 [R], Newcastle Upon Tyne, NEI 7RU, UK: University of Newcastle Upon Tyne, 2004.
[8] G. Pitsilis, L.F. Marshall, “Modeling Trust for Recommender Systems using Similarity Metrics”, IFIP International Federation for Information Processing, Trust Management II, 2008, volume 263, pp.103-118.
[9] K. Kwon, J. Cho, Y. Park, “Multidimensional Credibility Model for Neighbor Selection in Collaborative Recommendation”, Expert Systems with Applications, 36(3), pp. 7114-7122, 2009.
[10] M. Eisend, “Source Credibility Dimensions in Marketing Communication-a Generalized Solution”, Journal of Empirical Generalizations in Marketing, volume 10, pp. 1-33, 2006.
[11] M. Jamali, M. Ester, “TrustWalker: a Random Walk Model for Combining Trust-based and Item-based Recommendation”, In: Proc. of the 15th ACM SIGKDD Int’l Conf. on Knowledge discovery and data mining. New York, NY, USA: ACM Press, 2009, pp. 397-406.
[12] H. Ma, I. King, R.L. Michael, “Learning to Recommend with Social Trust Ensemble”, In: Proc. of the 32nd Annual ACM SIGIR Conf. on Research and Development in Information Retrieval. New York, NY, USA: ACM Press, 2009, pp. 203-210.
[13] H. Ma, T.C. Zhou, M.R. Lyu, I. King, “Improving Recommender Systems by Incorporating Social Contextual Information”, ACM Transactions on Infromation Systems, 29(2), article 9, 2011.
[14] H. Ma, I. King, M.R. Lyu, “Learning to Recommend with Explicit and Implicit Social Relations”, ACM Transactions on Intelligent System and Technology, 2(3),article 29, 2011.
[15] H.F. Sun, P. Yong, J.L. Chen, C.C. Liu, Y.Z. Sun, “A New Similarity Measure based on Adjusted Euclidean Distance for Memory-based Collaborative Filtering”, Journal of software, 6(6), pp. 993-1000, 2011.
[16] F. Cacheda, V. Carneiro, D. Fernandez, V. Formoso, “Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-performance Recommender System”, ACM Transactions on the Web, 5(1),article 2, 2011.
[17] B. Mehta, W. Nejdl, “Attack Resistant Collaborative Filtering”, In Proc. of the 31st annual Int’l ACM SIGIR Conf. on Research and Development in information retrieval. New York, NY, USA: ACM, 2008, pp. 75-82.



**Dongyan Jia** was born in Hebei Province, China in 1983. She received her M.S. in Computer Application Technology from Yanshan University, China, in 2009.
She is currently a Ph.D. Candidate in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. Her research interests include collaborative filtering, trusted computing and information security.

**Fuzhi Zhang** was born in Henan Province, China in 1964. He received his Ph.D. in Computer Application Technology from Beijing Institute of Technology, China, in 2003
He is currently a professor in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research domains cover intelligent network information processing, network and information security, service-oriented computing.

**Sai Liu** was born in Henan Province, China in 1987. He received his B.S. in Computer Application Technology from Zhengzhou University, China, in 2010.
He is currently a M.S. Candidate in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research interests include collaborative filtering and contextual recommendation.