

Research on Collaborative Filtering Recommendation Based on Trust Relationship and Rating Trust

Wenjun Huang^{1, a}, Junyu Chen^{1, b}, Yue Ding^{2, c}

¹Central South University, Changsha, China

²Renmin University of China, Beijing, China

^asuper20200910@163.com, ^bsuper123sb@163.com, ^cyueding0204@163.com

Abstract: In the Internet age, how to dig out useful information from massive data has become a research hotspot. The emergence of recommendation algorithms effectively solves the problem of information overload, but traditional recommendation algorithms face problems such as data sparseness, cold start, and low accuracy. Later social recommendation algorithms usually only use a single social trust information for recommendation, and the integration of multiple trust relationships lacks an efficient model, which greatly affects the accuracy and reliability of recommendation. This paper proposes a trust-based approach. Recommended algorithm. First, use social trust data to calculate user trust relationships, including user local trust and user global trust. Further based on the scoring data, an implicit trust relationship is calculated, called rating trust, which includes scoring local trust and scoring global trust. Then set the recommendation weight, build the preference relationship between users through user trust and rating trust, and form a comprehensive trust relationship. The trust relationship of social networks is integrated into the probability matrix decomposition model to form an efficient and unified trusted recommendation model TR-PMF. This algorithm is compared with related algorithms on the Ciao and FilmTrust datasets, and the results prove that our method is competitive with other recommendation algorithms.

Keywords: Trust relationship, Rating trust, Collaborative filtering, Probability matrix factorization.

1. Introduction

In traditional collaborative filtering recommendations, it learned and made suggestions simply based on users' historical ratings. However, the scoring information often has problems such as highly sparse data and scattered distribution, which lead to reducing performance of the recommendation, and this type of algorithm usually treats users as independent individuals [4]. It does not consider the relationship between users. This default method is inconsistent with interaction between people in our lives. The development of users' social platforms have made the social recommendation algorithm get more attention. This kind of algorithm adds social information to effectively alleviates the problem caused by scarcity of scoring information [16]. Friends with a trust relationship can be used as additional information to help recommend prediction [5]. A large number of experiments showed that the social recommendation algorithm can mitigate data sparseness and cold start problem, and it has strong resistance to malicious attacks by users on the network.

Recent work has demonstrated that the performance of algorithms can be improved by fusing social relations. Ma et al [7]. Used a joint decomposition of scoring matrix and trust matrix to express the user feature vectors in a shared manner. Ma et al [6]. proposed another trust-based recommendation algorithm RSTE, which have construct a trust relationship to rerepresent the user feature vector. What these two models have in common is that they combine social relationships to modify the objective function to better learn user characteristics. After that, SocialMF [11] makes up for the shortcomings of RSTE; Guo Lei, Ma Jun and others [2] proposed the StrengthMF and InfluenceMF algorithms. Although these algorithms consider the trust relationship and interest preferences of users through a shared feature space, the entire calculation process only uses rating information. At

present, trust relationship has not been really exploited and utilized.

In short, for the recommendation of trust in a big data environment, it is not fully utilized for the trust relationship, and the prediction accuracy still needs to be improved. Therefore, we proposed a new trust-based Recommended model TR-PMF to integrate scoring information and trust relationship. The main contributions are as follows:

We introduce the trust relationship of the social network in reality into the recommendation system, comprehensively consider the global trust and local trust in conjunction with the graph, and use the propagation characteristics of trust to model the relationship.

Based on the user's rating data, we model the rating trust relationship and calculate the rating trust for each user. Including local rating trust and global rating trust.

We merge user trust relationship and rating trust relationship using adaptive weights to form a comprehensive trust relationship. This also alleviates the inaccurate recommendation caused by the sparse trust data and enhances the credibility and accuracy of the recommendation.

In order to improve the accuracy of user rating prediction, we integrate the trust relationship and rating information into the probabilistic matrix factorization model, and learn the potential characteristics of users through iteration.

2. Related Work

Yuan et al [15]. It is found that the trust network is a small world network in which two random users can connect in a short distance, which fully shows the role of trust relationship in the recommendation system. Research shows that the fusion of user trust information can improve the performance of recommendation [14]. There are two main tasks in the recommender system: item recommendation and rating

prediction. Most algorithms are designed for one of the recommendation tasks, and our work focuses on the rating prediction task.

Ma et al [8] [9], developed several approaches by adding different trust regularization terms to a matrix factorization model. They first proposed a social regularization method named SoRec, by considering the constraint of social relationships. It uses an algorithm model based on matrix decomposition and shared representation of user characteristic matrix. The SoRec model has given great inspiration to scholars' research on social recommendation. Many models are based on improvements on this.

Ma et al [12], then proposed a social trust ensemble method RSTE, the model not only effectively considers the user's preference information, but also takes into account the influence of the preferences of his trusted friends when predicting and scoring the data. The authors [10] further proposed a new matrix factorization model SoReg. The SoReg model [1] first proposed the difference between friend relationship and trust relationship. The model believes that the relationship between friends should be more dependent on the social relationship of users in the real world, and the trust relationship between them should be more Depends on similar preferences among users for the project.

Although the RSTE model effectively uses social trust as auxiliary information for recommendation. The influence of friend preference on the final prediction score is also considered. However, this model does not consider the spread of preference information among social networks. A state-of-the-art approach, SocialMF, is proposed by Jamali and Ester [3]. The author believes that the user's personal preferences are largely similar to the preferences of his trusted friends. Based on this concept, the user feature vector is represented by the weighted average of the potential vectors of his trusted friends.

More recently, Yang, et. al. [8] propose a method TrustMF. The algorithm fully considers the directionality of the trust relationship. The core idea of the algorithm is to represent each user in the system with two different feature vectors. One of the feature vectors is called the trust feature vector B_i , and the other is called the trusted feature vector W_i .

The TrustSVD model [26] was proposed by Guo et al. The model is further improved on the basis of the SVD++ model, and it adds social information among users. It not only uses social trust information to re-represent user feature vectors, but it also effectively adds implicit social feedback information to the model. Further improve the effect of the model.

In summary, the existing social recommendation algorithm only uses a single social trust information for recommendation, does not consider multiple trust information, and still faces the problem of data sparseness and cold start. This article points out that a variety of trust relationships can be mined from user trust data and user rating data, and these relationships have an important impact on the prediction results of the recommendation system. Propose and

implement a collaborative filtering recommendation algorithm TR-PMF that integrates trust relationship and rating trust.

3. Notation

Table 1 Lists Notations.

Table 1. Notations	
Symbolic	Meaning
U, V, u_i, v_j	User collection, item collection, user i , item j
R	Score matrix
R_{ij}	User i 's score on item j
R^{\wedge}	Prediction scoring matrix
ST	Social trust matrix
t_{ij}	Connection weight of user i and user j in the social network
rt_{ij}	The local trust value of user i to target user j
gt_{ij}	Global trust value of user j
ST_{ij}	User trust value of user i to user j
Sim_{ij}	Similarity values of users i and j
$UserSim(u_i)$	Similar user set of user a_i
pk_{ij}	inferred by user j , user i 's rating of item k .
$Reck_{ij}$	The correct value of item k recommended by similar user j of user i
RL_{ij}	The local rating trust value of user i to user j
RG_i	User i 's global rating trust value
RT_{ij}	User i 's rating trust value to user j
S_{ik}	Comprehensive trust value of user i to user k
$N(i)$	User i has a set of users with a comprehensive trust relationship

4. Model

In this section, we will describe the TR-PMF model in detail. TR-PMF can be divided into three parts, the first part is to calculate the user trust relationship, and the second part is to calculate the rating trust relationship. The third part is to integrate user trust and rating trust, and integrate it into PMF to make scoring prediction. As shown in Figure 1.

4.1. Problem Definition

Corresponding to the above definition, we also elaborated our model from these three perspectives.

Definition 1: (How to calculate user trust): Use trust data to calculate the user's local trust in a graph traversal method. Further calculate the global trust relationship between users, and finally combine them into user trust.

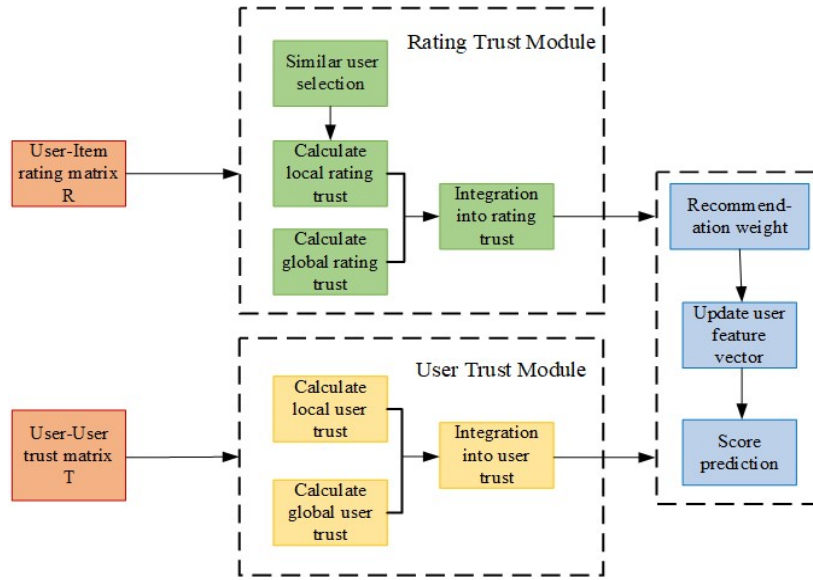


Figure 1. Framework of TR-PMF model

Definition 2: (How to calculate rating trust): Using scoring data, first use the PCC formula to calculate the similarity between users, and further calculate the partial rating trust. Secondly, the global rating trust is calculated. Finally, it is combined into a rating trust.

Definition 3: (How to Integrate User Trust and Rating Trust) The adaptive weight method is adopted to determine the proportion of user trust and rating trust based on the number of common rating items between users, thereby combining them into recommendation weights.

Corresponding to the above definition, we also elaborate our model from these three perspectives.

4.2. How to Calculate User Trust

1) User local trust. User local trust is a small-scale trust relationship between users. The user local trust calculation method is as follows:

$$rt_{ij} = \sum_{k \in N(i)} t_{ik} t_{kj} \quad (1)$$

2) User global trust. The trust value a user has in the entire social network. Each user has and only one global trust value. The calculation formula is as follows:

$$gt_j = \frac{ind(j) - \min(ind(T))}{\max(ind(T)) - \min(ind(T))} \quad (2)$$

$\max(ind(T))$ is the most trusted user in the entire social network. In order to map the value range of the global trust value to the $[0,1]$ interval, divide it by the difference between $\max(ind(T))$ and $\min(ind(T))$.

3) User trust. By combining the user's local trust with the user's global trust, the user's trust value can be expressed more accurately. The combination is as follows:

$$ST_{ij} = \beta * rt_{ij} + (1 - \beta) * gt_{ij} \quad (3)$$

β is the proportion of local trust, and the value range of β is $[0, 1]$.

4.3. How to Calculate Rating Trust

Using the user's rating data, an implicit trust relationship is calculated, which is called rating trust in this article. rating trust is divided into local rating trust and global rating trust.

Rating trust indicates the accuracy of other users' recommendation to the current user. If the recommendation result of user j to user i is very accurate, then user i has a high rating trust value to user j .

1) Partial rating trust. In the actual recommendation situation, assuming that user j has a more accurate recommendation to current user i , user i will trust user j more, and the local rating trust value is greater.

If current user i has rated item k , his similar user j predicts user i 's rating for item k , the prediction result is p_{ij}^k , the calculation formula is as follows:

$$p_{ij}^k = \bar{r}_i + (r_{jk} - \bar{r}_j) * Sim_{ij} \quad (4)$$

Sim_{ij} indicates the similar value of user i and j , calculated by the PCC formula.

On this basis, if the difference between the prediction result p_{ij}^k of user j and the actual rating r_{ik} of user i is less than the deviation threshold ε set by the system. It is considered that similar user j made a correct recommendation to target user i on item k . The recommended correct value on this item at this time is 1, otherwise it is 0. The calculation formula is as follows:

$$Rec_{ij}^k = \begin{cases} 1, & |p_{ij}^k - r_{ik}| \leq \varepsilon \\ 0 & \end{cases} \quad (5)$$

Else

Among them, Rec_{ij}^k represents the correct value recommended by similar user j to user i on item k . ε represents the error value set by the system. The setting of ε value is set by the step size of the scoring data.

For target user i and similar user j , calculate the average value of the recommended correct value on all common scoring items, which represents the local rating trust value. Calculated as follows:

$$RL_{ij} = \frac{\sum_{k=1}^{|P_{ij}|} Rec_{ij}^k}{|P_{ij}|} \quad (6)$$

The description of the rating trust value algorithm is shown in Table 2:

Table 2. Rating Trust Value Algorithm Description

rating trust value algorithm	
Input:	User rating data R Similar user set UserSim(U) Score prediction bias threshold ε
Output:	User rating trust RT
<pre>//compute local rating trust for i=1 to m do for u_j ∈ UserSim(u_i) do for v_k ∈ P_{ij} do p_{ij} = formula(1) if (p^k_{ij} - R_i(k)) ≥ ε then RL_{ij} = RL_{ij} + 1 Else RL_{ij} = RL_{ij} + 0 End for RT1(i,j) = RL_{ij}/ P_{ij} //local rating trust End for End for End for //compute global rating trust for i=1 to m do f[i] = ratingTimes(u_i) q[i] = neighborTimes(u_i) RT2(i) = 2/(1/(1 - 1/log(f[i] + 3)) + 1/(1 - 1/log(q[i] + 3))) End for //compute Rating trust for i=1 to m do for j=1 to m do RT(i,j) = 2 * RT1(i,j) * RT2(j)/(RT1(i,j) + RT2(j)) End for End for return RT End</pre>	

2) global rating trust. The calculation formula of the user's global rating trust is as follows:

$$RG_i = \frac{2}{\frac{1}{1-1/\ln(f_i+3)} + \frac{1}{1-1/\ln(q_i+3)}} \quad (7)$$

Where RG_i represents the global rating trust of user i . q_i represents the number of times user i is a similar user of other users. f_i represents the total number of scoring times of user i in the system.

4.4. How to Integrate User Trust and Rating Trust

By setting the recommendation weight S_{ik} , which adaptively balances the user trust relationship and rating trust relationship, improves the ability to identify trusted users, and avoids inaccurate rating prediction due to sparse trust data.

Set the maximum and minimum number of common scoring items in the system. The maximum value is n_{max} denotes the minimum value by n_{min} . S_{ik} is the recommended weight, which can be calculated by the following piecewise function:

$$S_{ik} = \begin{cases} 0, n \leq n_{min} \text{ and } ST_{ik} = 0 \\ ST_{ik}, n \leq n_{min} \text{ and } ST_{ik} \neq 0 \\ \frac{n}{n_{max}} RT_{ik}, n_{min} \leq n \leq n_{max} \text{ \& } ST_{ik} = 0 \\ \frac{n}{n_{max}} RT_{ik} + (1 - \frac{n}{n_{max}}) ST_{ik}, \\ n_{min} \leq n \leq n_{max} \text{ \& } ST_{ik} \neq 0 \\ RT_{ik}, n \geq n_{max} \end{cases} \quad (8)$$

4.5. How to Score Predictions

TR-PMF effectively predicts the user's scoring results for the item through the calculated recommendation weight. The recommendation weight is integrated into the PMF, the user and item feature matrix is learned, and finally the score is predicted.

Due to the influence of the trust relationship, the behavior of each user i will be affected by the directly connected user $N(i)$, where $N(i)$ is the set of users who have a recommendation weight for the user i . Then the feature vector of user i can be expressed as follows:

$$\hat{U}_i = \frac{\sum_{k \in N(i)} S_{ik} U_k}{\sum_{k \in N(i)} |S_{ik}|} \quad (9)$$

In order to facilitate the calculation, the trust matrix of each row of the user is normalized, then for each user i can be obtained $\sum_{k=1}^M |S_{ik}| = 1$.

Assuming that the rating is a Gaussian distribution with Gaussian noise [63], The formula is as follows:

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^M \prod_{j=1}^N \frac{1}{\sqrt{2\pi\sigma_R^2}} \exp\left[-\frac{(R_{ij} - U_i^T V_j)^2}{2\sigma_R^2}\right] \quad (10)$$

The item feature vector obeys the following Gaussian distribution:

$$P(V | \sigma_V^2) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi\sigma_V^2}} \exp\left[-\frac{V_j^2}{2\sigma_V^2}\right] \quad (11)$$

According to the principle of Bayesian formula, the posterior probability distribution formula of the feature vectors U and V of users and items can be obtained as follows:

Using Bayes' rule to derive the posterior probability formula of the feature vector U, V :

$$\begin{aligned}
L(R, S, U, V) = & \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N I_{ij} (R_{ij} - g(U_i^T V_j))^2 \\
& + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \\
& + \frac{\lambda_S}{2} \sum_{i=1}^M \|U_i - \sum_{k \in N(i)} S_{ik} U_k\|_F^2
\end{aligned} \quad (12)$$

Calculate the partial derivatives of U and V through the loss function. The calculation is as follows:

$$\begin{cases} \frac{\partial L}{\partial U_i} = \sum_{j=1}^N I_{ij} (g(U_i^T V_j) - R_{ij}) * g'(U_i^T V_j) * V_j \\ \quad + \lambda_U U_i + \lambda_S (U_i - \sum_{k \in N(i)} S_{ik} U_k) \\ \quad - \sum_{p \in B_i} \lambda_S (U_p - \sum_{k \in N(p)} S_{pk} U_k) U_i \\ \frac{\partial L}{\partial V_j} = \sum_{i=1}^M I_{ij} (g(U_i^T V_j) - R_{ij}) * g'(U_i^T V_j) + \lambda_V V_j \end{cases} \quad (13)$$

Using gradient descent to update the model parameters U and V, The scoring prediction formula is as follows:

$$\hat{R}_{ij} = g(U_i^T V_j) * R_{max} \quad (14)$$

R_{max} is the maximum value of the rating range in the recommendation system.

5. Evaluation

We evaluated our method on two datasets: ciao and filmtrust, which are two open real datasets [13]. The first ciao data set is the real data collected by Tang et al on the article review site ciao. The second filmtrust data set is the real data on the filmtrust website, which was captured by Guo et al.

Table 3. Structure of Dataset

Feature	Ciao	FilmTrust
users	7375	427
items	106997	11848
ratings	284086	1313
Social relations	111781	1853
Avg Ratings per Item	2.65	0.11
Avg Ratings per User	38.52	3.07

The common evaluation methods of recommendation algorithm are root mean square error (RMSE) and mean absolute error (MAE) [13]. In order to measure the accuracy of our tr-pmf algorithm.

$$MAE = \frac{1}{T} \sum_{i,j} |R_{i,j} - \hat{R}_{i,j}|, RMSE = \sqrt{\frac{1}{T} \sum_{i,j} (R_{i,j} - \hat{R}_{i,j})^2} \quad (15)$$

5.1. Experimental Setup

This experiment is based on two data sets, Ciao and FilmTrust, as shown in Table 3. Using a five-fold crossvalidation method. During the experiment, randomly selected 80% data as training data, and 20% data as test data. Both were performed five times of five-fold cross-validation. This paper reports the average results.

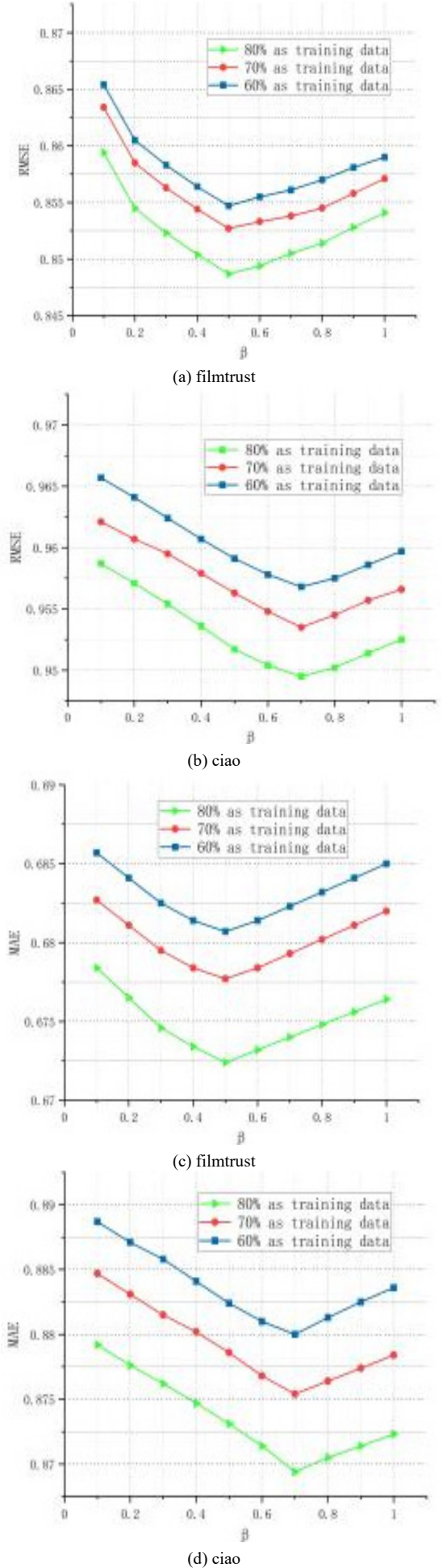


Figure 2. Parameter β analysis

5.2. Experimental Analysis

In order to observe the influence of each parameter in the TR-PMF algorithm on the prediction performance, this section mainly analyzes and tests each parameter. The parameters included in the TR-PMF algorithm flow are: the proportion of local trust in the user trust β . The maximum value n_{\max} and the minimum value n_{\min} of the number of items that users have scored together, The number of iterations d in the algorithm training, the learning rate, and the regularization parameters λ_U and λ_V . This section analyzes the impact of parameter changes on the prediction performance of TR-PMF based on the single variable principle.

In the selection of parameter value, the value range of β is set to $[0, 1]$, and the step size is 0.1. The value range of n_{\max} is 5,6,7,8,9,10,11,12. In the process of parameter adjustment, the dimension k of user features and project features is set to 10, and the regularization parameter $\lambda_U = \lambda_V = 0.001$, learning rate $\theta = 0.01$.

1) The influence of user trust relationship

The value of β in TR-PMF determines the proportion of local trust in the user trust relationship. β takes values 0, 0.1, 0.2 ... 1.0 in sequence. The impact on rating prediction is shown in Figure 2.

The experimental results show that RMSE and Mae are the least when β is 0.7 and 0.5 on ciao and filmtrust datasets, respectively. This means that in the user trust relationship, the dependence on local trust is 70%, while the dependence on global trust is 30%. Compared with ciao dataset, the impact of users' global trust on RMSE is greater on filmtrust dataset.

Analysis from the experimental results: the value of β will be affected by the data set, which is due to the different proportion of trust relationship in the data set. Compared with ciao dataset, the proportion of trust relationship in filmtrust

dataset is relatively rare. Only 609 people have direct trust relationship, and most users have less than ten trusted friends. Therefore, users can only rely more on the recommendation influence of global users, and are less affected by their own subjective consciousness. However, ciao has more trust data and relies more on the influence of its local trust relationship.

2) The influence of the threshold of the number of common items

The threshold of the number of scoring items is used to integrate user trust and rating trust according to different classification situations. The setting of this value is affected by the user's rating data. When the data is very scarce, if the minimum value n_{\min} of the number of common scoring items is set too high, then the user's comprehensive trust value is mainly affected by the user's trust. If n_{\max} is set too low, it will reduce the use of user trust. This article will analyze the values of n_{\max} and n_{\min} based on experiments.

In order to maintain the principle of single variable, the value of n_{\min} is 0 in the experiment, and the value of n_{\max} is 5, 6, 7, 8, 9, 10, 11, 12 in turn. It can be seen from Figures 3 that as the maximum common item number n_{\max} value increases, the trends of RMSE and MAE on Ciao and FilmTrust gradually increase. When the value of n_{\max} is 5 and 6, the scoring error is the smallest. And then the value of n_{\min} is 0 to 4 in turn. When the value of n_{\min} is 0, the scoring error is the smallest. Therefore, the value of n_{\min} is 0. From the analysis of the experimental results, it is concluded that since the number of user ratings for items is very scarce, the threshold for the number of common items cannot often be set too high.

5.3. Experimental Comparison

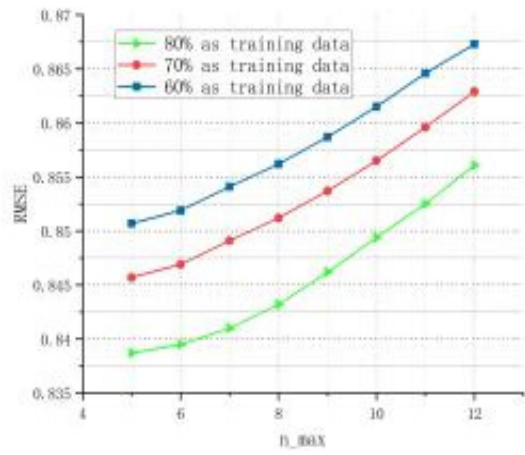
Comparison methods. In order to verify the effectiveness of the proposed TR-PMF algorithm, it is compared with the following algorithms:

Table 4. Performance comparison in the testing view of 'all', where * indicates the best performance among all the other methods, and column 'improve' indicates the percentage of improvements that tr-pmf achieves relative to the *result

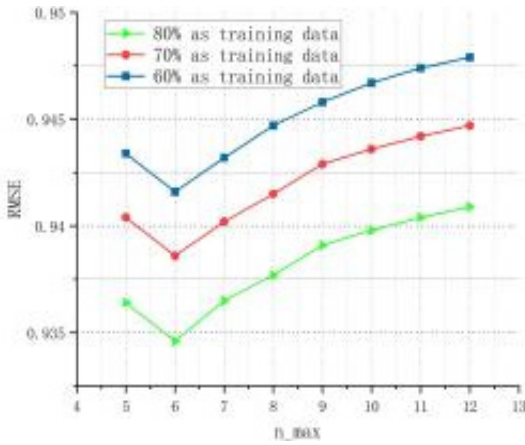
All	Metrics	UAvg	IAvg	PMF	RSTE	SoReg	SocialMF	TrustMF	SVD++	TrustSVD	TR-PMF	Improve
FilmTrust	MAE	0.636	0.725	0.714	0.628	0.661	0.638	0.631	0.613*	0.615	0.603	1.63%
K=10	RMSE	0.823	0.927	0.949	0.81	0.866	0.837	0.81	0.804	0.798*	0.790	1%
Ciao	MAE	0.632	0.728	0.735	0.64	0.644	0.642	0.631	0.611*	0.613	0.604	1.14%
K=10	RMSE	0.831	0.943	0.967	0.827	0.857	0.849	0.823	0.808	0.795*	0.786	1.13%

Table 5. Performance comparison in the testing view of 'cold start'

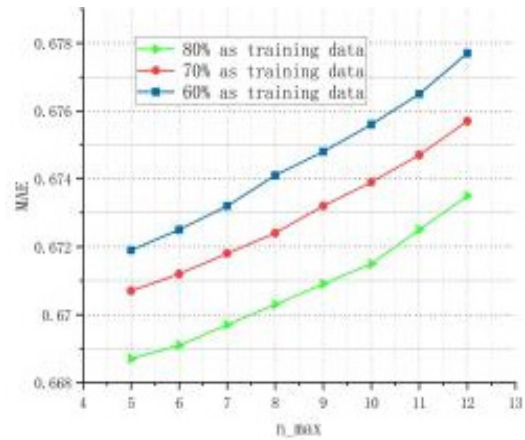
All	Metrics	UAvg	IAvg	PMF	RSTE	SoReg	SocialMF	TrustMF	SVD++	TrustSVD	TR-PMF	Improve
FilmTrust	MAE	0.756	0.841	0.824	0.739	0.781	0.759	0.742	0.733	0.723*	0.718	0.69%
K=10	RMSE	1.071	1.121	1.135	1.021	1.047	1.029	1.018	1.004	0.928*	0.907	2.26%
Ciao	MAE	0.778	0.859	0.839	0.747	0.787	0.774	0.753	0.747	0.737*	0.728	1.22%
K=10	RMSE	1.093	1.144	1.152	1.043	1.065	1.047	1.035	1.021	0.946*	0.925	2.21%



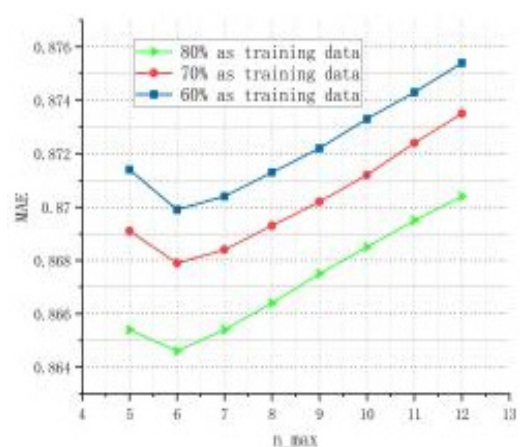
(a) FilmTrust



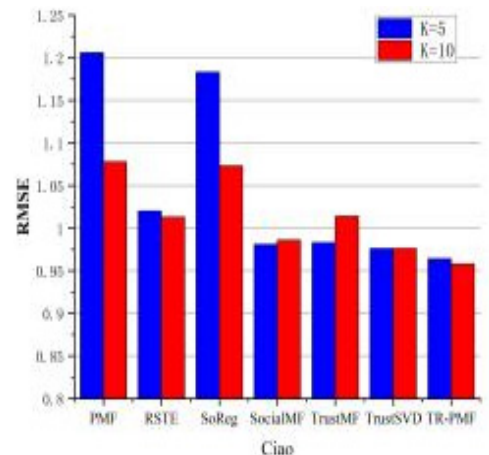
(b) Ciao



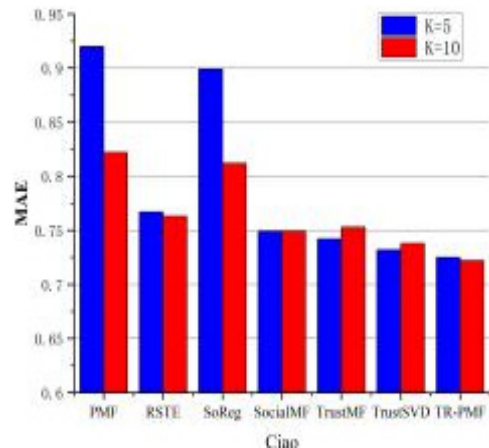
(c) FilmTrust



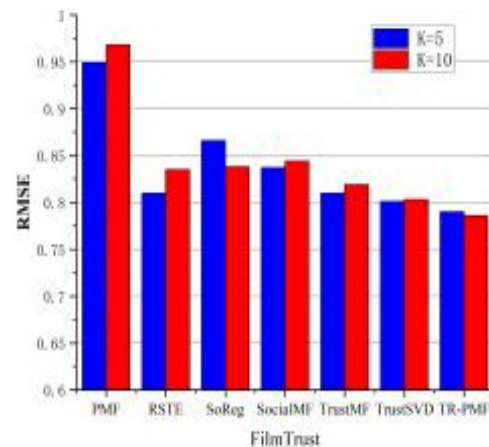
(d) Ciao



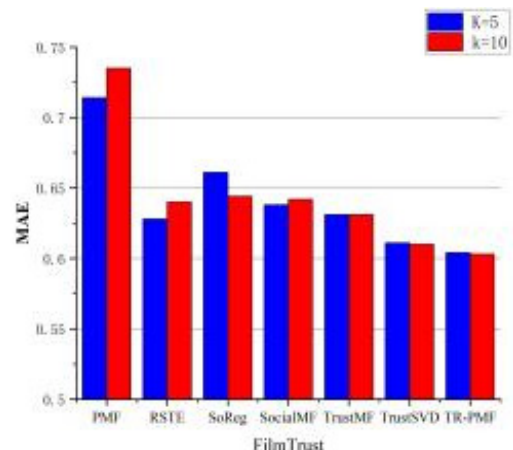
(a) Ciao



(b) Ciao



(c) FilmTrust



(d) FilmTrust

Figure 3. The influence of the threshold of the number of common items

Figure 4. Model comparison results

(1) UAvg, IAvG are baselines that predict a user's rating by the average of her historical ratings, and of ratings received by the target item, respectively;

(2) PMF: Introduced a probabilistic model to optimize the matrix factorization model;

(3) RSTE [23], SoReg [6] are earlier trust-based recommendation models by Ma et al.;

(4) SocialMF [7], TrustMF [8], Fang's [9] are the latest and state-of-the-art trust-based models that are reported to achieve better performance than simple baselines and other counterparts [8], [9];

(5) SVD++ [12] is a state-of-the-art recommendation method merely based on ratings, and also adopted as a key comparison method in Fang et al. [9]. TrustSVD, a trust-based matrix factorization technique for recommendations;

(6) TrustSVD: was proposed by Guo et al. It not only uses social trust information to re-represent user feature vectors, but it also effectively adds implicit social feedback information to the model. Further enhance the effect of the model;

The comparison result is visually shown in Figure 3. And The experimental results are shown in Table 4 and table 5, which correspond to the "all" and cold start test views. In all test views, the performance of trustsvd in filmtrust and epinions is better than other comparison methods. In ciao, the trust based approach (SocialMF) provides the best solution performance. Our method tr-pmf is always superior to the best of other methods in all data sets. Although the percentage of relative improvement is very small (RMSE is about 1.34%), Koren [11] pointed out that even small improvements in Mae and RMSE may lead to significant differences in practical recommendations. SoReg is the only one that makes significant improvement when adjusting the number of potential factors from 5 to 10.

For all the comparison methods in the cold start test view, trustsvd and SVD++ perform best in filmtrust and Ciao. Since all trust based models aim to optimize the square error between the predicted value and the actual value, RMSE is more indicative than MAE, so the overall performance of TR-PMF is the best.

In order to verify that adding trust information and item information can effectively alleviate the cold start problem in the recommendation system. First assume cold start users and cold start projects. In this paper, users in Ciao and FilmTrust who have scored less than 5 on the project are regarded as cold-start users. Ciao and FilmTrust regard projects with less than 3 and 5 scores as cold-start projects.

Analysis of Ciao's experimental results: In Table 4 and Table 5, because PMF is only the most basic algorithm model, it is a probability matrix decomposition model that only uses user-item rating information, so the recommendation accuracy is the lowest. Other models also use social trust, which has better recommendation accuracy. Among them, TR-PMF has the best effect, which fully shows that the addition of trust information can effectively alleviate the impact of sparse data and improve the recommendation effect. No matter the feature dimension K is 5 or 10, TR-PMF recommendation accuracy is the best. It fully mines a variety of trust relationships, not only mining the user's local trust and global trust relationship, but also calculates the rating trust based on scoring information, and integrating these trust relationships can more accurately model the user's behavior preferences.

Analysis of the experimental results on the FilmTrust

dataset: As shown in Table 4 and Table 5, when the feature dimension K is 10, the effect is slightly better than when it is 5.

Comparing the PMF algorithm with the trust recommendation algorithm, TR-PMF has the best recommendation results under the RMSE and MAE evaluation indicators. It can be concluded that TR-PMF integrates user trust relationship and rating trust relationship, and can more accurately model user behavior preferences, thereby improving recommendation accuracy.

6. Conclusions

Aiming at improving the performance for rating prediction, This paper proposes a collaborative filtering recommendation algorithm TR-PMF that integrates trust relationship and rating trust. TR-PMF first mines the trust relationship through user trust data, and combines the calculated local trust of the user with the global trust to form a user trust relationship. And further calculate the rating trust relationship based on the user's historical rating information. A comprehensive trust relationship is obtained by fusing the user trust relationship and the rating trust relationship. It can more accurately indicate the degree of trust between users. Compared with the shortcomings of traditional social recommendation algorithms that only use a single trust data, TR-PMF also effectively utilizes the calculated relationship, alleviating the problem of sparse trust data. Finally, the integrated trust relationship is integrated into the probability matrix decomposition model, the user and item feature matrix is learned, and the rating prediction is finally made. It provides new ideas for trust-based recommendation research. Of course, there is still room for further research on trust-based recommendation algorithms, such as the dynamic model of trust relationships that considers time factors, the integration of item-based collaborative filtering ideas, and the impact of untrusted relationships on recommendation results. In our future work, we plan to further consider the impact of project relationship on scoring prediction.

References

- [1] Iman Barjasteh, Rana Forsati, Dennis Ross, Abdol-Hossein Esfahanian, and Hayder Radha. Cold-start recommendation with provable guarantees: A decoupled approach. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1462–1474, 2016.
- [2] Guang-Neng Hu, Xin-Yu Dai, Feng-Yu Qiu, Rui Xia, Tao Li, Shu-Jian Huang, and Jia-Jun Chen. Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):1–30, 2018.
- [3] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142, 2010.
- [4] Liaoliang Jiang, Yuting Cheng, Li Yang, Jing Li, Hongyang Yan, and Xiaoqin Wang. A trust-based collaborative filtering algorithm for ecommerce recommendation system. *Journal of Ambient Intelligence and Humanized Computing*, 2019.
- [5] Mingyang Jiang, Zhifeng Zhang, Jingqing Jiang, Qinghu Wang, and Zhili Pei. A collaborative filtering recommendation algorithm based on information theory and bi-clustering. *Neural Computing and Applications*, 2019.

- [6] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [7] Hanjun Lee, Keunho Choi, Donghee Yoo, Yongmoo Suh, Soowon Lee, and Guijia He. Recommending valuable ideas in an open innovation community: A text mining approach to information overload problem. *Industrial Management & Data Systems*, 2018.
- [8] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210, 2009.
- [9] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940, 2008.
- [10] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296, 2011.
- [11] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.
- [12] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112, 2015.
- [13] Jiliang Tang, Huiji Gao, and Huan Liu. mtrust: Discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 93–102, 2012.
- [14] Weilong Yao, Jing He, Guangyan Huang, and Yanchun Zhang. Modeling dual role preferences for trust-aware recommendation. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 975–978, 2014.
- [15] Weiwei Yuan, Donghai Guan, Young-Koo Lee, Sungyoung Lee, and Sung Jin Hur. Improved trust-aware recommender system using smallworldness of trust networks. *Knowledge-Based Systems*, 23(3):232–238, 2010.
- [16] Yuan Zhang, Ke Meng, Weicong Kong, and Zhao Yang Dong. Collaborative filtering-based electricity plan recommender system. *IEEE Transactions on Industrial Informatics*, 15(3):1393–1404, 2019.