RESEARCH ON PERSONALIZED RECOMMENDATION ALGORITHM BASED ON TRUST RELATIONSHIP

Wenhao WU[1], Rong LIU[2], Baoling JIA[3], Mingshan YIN[4], Yongkang WANG[5], Zhijun ZHANG[6]*

Personalized recommendation, as an important technology to overcome information overload, has been widely used in various fields. However, due to the deficiencies and limitations of existing recommendation algorithms, traditional collaborative filtering recommendation algorithm does not consider the impact of multi-source social information, resulting in low recommendation accuracy. To solve the above problems, this paper proposes a personalized recommendation algorithm that integrates user trust relationship. Based on matrix decomposition, this algorithm fully considers the characteristics of explicit trust relationship, implicit trust relationship and trust propagation between users. The algorithm incorporates users' explicit trust relationships and implicit trust relationships through weight factors into the matrix factorization algorithm. The experimental results show that the accuracy of the proposed algorithm is improved significantly.

Keywords: Collaborative filtering, Recommendation system, Matrix factorization, Trust relationship

1. Introduction

With the rise of social networks such as Twitter and Facebook, more and more researchers use social relationships between users to make recommendations, and the decision-making process of users is more susceptible to the influence of friend relationship or trust relationship. As we all know, the datasets of user rating on items have a quite high degree of sparsity. After years of research, scholars [1] have found that matrix decomposition algorithm can effectively alleviate the data sparsity. However, due to the high computational complexity of matrix decomposition algorithm, it is difficult to apply to large-scale scoring matrix, so it has been limited for a long time and has not been

[1] China Information and Communication Technology (CICT) Mobile, Beijing 100083, China
[2] China Information and Communication Technology (CICT) Mobile, Beijing 100083, China
[3] China Information and Communication Technology (CICT) Mobile, Beijing 100083, China
[4] School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China
[5] School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China
[6] School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China. Corresponding Author Emails: zzjsdcn@163.com

widely valued by scholars. The emergence of latent factor models has aroused academic interest in matrix decomposition, which has a core idea that users' interests and items are linked through implicit features, which improves the accuracy of prediction scoring and greatly relieves the data sparsity. Through the continuous research of scholars [1], it is found that matrix factorization can bring better recommendation results, so at present, matrix factorization algorithm has been widely used in recommendation system. Moreover, researchers have found that using trust among friends can effectively improve the accuracy of the algorithm. Wang Jianfang [2] et al. incorporated the trust relationship into the probability matrix factorization algorithm (PMF), which improved the accuracy of the algorithm to a certain extent. Guo et al. [3] proposed a TrustSVD++ algorithm which fuses explicit trust relationship based on SVD++ algorithm and found that the introduction of trust relationship further solves the cold start problem. In this paper [4], a DT-LFM personalized recommendation algorithm based on gSVD++ algorithm is presented, which combines explicit trust relationship with implicit trust relationship, and the direct trust relationship between users and the implicit trust relationship obtained by rating information between users are fully considered to further improve the accuracy of recommendation. Finally, it is found that the performance of the algorithm has been verified on Filmtrust dataset and Epinions dataset.

2. Related Work

Tags can reflect users' interests to a certain extent, and above all, to describe the characteristics of resources, which provides a new direction for the development of the recommendation system. In some fused tag systems, users have a certain degree of freedom to select tags to describe resources, at this time, the content information of users and resources has been expanded [5], which plays a crucial role in promoting the effect of personalized recommendation, and is also an important reason why tags can become a hot topic in personalized recommendation research. At present, there are many achievements in label-based recommendation algorithms. A hybrid recommendation method [6] is proposed to enhance tag-based recommendation by mining a variety of relationship information, and the results of the experiment are ideal. The relevance of tags found by analyzing user's performance based on association rules in social network can also improve the recommendation effect of resources to some extent [7].

At present, with the rapid development of social networks, more and more researchers pay attention to the calculation of trust between users. In 2005, Golbeck proposed the TidalTrust algorithm [8], which performs an improved breadth-first search method in trust networks for calculating forecast scores. This method is special in that it can find the score of all relevant users through the nearest trust path to the target users, and difficult in calculating the trust between non-neighboring users. In 2007, Massa et al. proposed MoleTrust [9] algorithm, which can propagate trust relationship and estimate trust weight through trust network, replace similarity weight with trust weight, and calculate trust value between users by performing a reverse search algorithm. In 2010, Jamali proposed a personalized recommendation algorithm based on trust [10], SocialMF, which combines matrix decomposition technology with trust propagation mechanism to make recommendations, and found that the recommendation accuracy of SocialMF algorithm is high, and trust propagation mechanism is proved to be an important method in trust-based social networks.

Many domestic scholars have put forward a lot of methods. In 2013, Guo [11] proposed a socialized recommendation algorithm based on the strength of trust relationship, which can further improve the accuracy of existing recommendation algorithms. In 2014, Qin [12] proposed an algorithm, in which the use of score similarity or the use of inter user trust values were selected according to different values of the number of common scoring resources, making up for the defect that the recommendation method fails to pay attention to users' sentiment. In 2014, Zou Benyou et al. [13] proposed a personalized recommendation method under the social network environment that integrates the tension decomposition and the user trust relationship, in which the trust degree of different friends is selected according to different recommendation topics, suitable for the personalized recommendation scene with relatively fast update speed. In 2015, Zhang et al. proposed a personalized recommendation algorithm integrating trust relationship and time series [14], which comprehensively considers the trust relationship and time factors between users and ensures the timeliness of recommendation.

After in-depth analysis of the above algorithms, it is easy to find that there are still many shortcomings in the existing personalized recommendation algorithm. In the existing tag-based recommendation algorithm, the use of tag information is in the way of tag popularity, while the use of trust relationship information is relatively rough. Only through the trust relationship, the relevant users can be found, and the preference items of relevant users can be recommended to users. Other algorithms do not consider the effect of tag information and trust relationship information on recommendation results. Based on the above analysis, this paper studies the integration of trust relationship, label and time on the basis of previous work. Experiments on public datasets show that the proposed algorithm TKPMF-CF has obvious feasibility and superiority over traditional collaborative filtering algorithms and trust and tag recommendation methods.

3. Latent Factor Model for Fusing Trust Relationships

3.1 Calculation of Explicit Trust Matrix

As everyone knows, trust information on most social network sites is binary, i.e. trust and distrust, expressed in 1 and 0. However, the information expressed by binary trust values is insufficient to reflect the differences between users who trust them and other users. For example, there are five users in the trust set of user u, and it is obvious that user u has all 1 value to them in the binary trust matrix, which is obviously incompatible with the real situation.

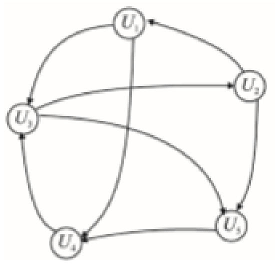

Fig. 1 Trust network

Figure 1 shows a trust network in which each user is one of the nodes. The trust values between users are represented by the edges in the diagram. Thus, a calculation method for the trust values of user u to user v is given as formula (1), where T_u refers to the set of directly trusted users of user u. The trust in-degree of user v is $Ind(v)$, and the in-degree of the user trusted by u is $Ind(d)$.

$$t(u,v) = \sqrt{\frac{Ind(v)}{\sum_{d \in T_u} Ind(d)}} \tag{1}$$

In real life, the trust degree of user u to users in the trust set cannot be the same, so a distinguishing method is needed. In this paper, based on the location of users in the trust network, from the user's point of view, the number of users trusted by other users (in-degree) and the number of users trusting others (out-degree) are considered, which reflect to some extent the value of that user in the social network. The calculation method of trust value used in this paper is shown in formula (2)

$$t(u,v) = \sqrt{\frac{Ind(v)}{Outd(u) + Ind(v)}} \tag{2}$$

The formula $T = P^T W$ is used to calculate the trust matrix by the feature vectors matrix $P^{l \times m}$ of user u and $W^{l \times m}$ of user v. T is the trust matrix between user u and user v.

Formula (3) is used to get the explicit trust matrix of user u to user v.

$$\hat{t}_{u,v} = w_v^T p_u \tag{3}$$

Where,

$t_{u,v}$ =the the degree of trust of user u to user v;

p_u =the latent feature vectors of user u at the l-dimension;

w_v = the latent eigenvector of the trusted user v at the l-dimension.

3.2 Calculation of Implicit Trust Matrix

The previous mentioned is the calculation of explicit trust relationship. In real life, it has become the focus of the research on user trust relationship to measure the trust value between users by mining the similarity of the common rating items of items between users. Implicit trust relationship consists of direct trust relationship and indirect trust relationship, the former as the name implies is the direct interaction between users, while the latter is obtained through the communication mechanism of trust.

(1) Measurement of direct trust value

In this paper, JMSD [15] is used to calculate trust among users, as shown in formula (4).

$$JMSD_{u,v} = \left(1 - \frac{\sum_{i \in I_{uv}} \left(\frac{r_{u,i} - r_{v,i}}{\max - \min}\right)^2}{|I_{uv}|}\right) \times \frac{|I_{uv}|}{|I_u| + |I_v| - |I_{uv}|} \tag{4}$$

Where,

I_{uv} =the set of items with the common ratings of user u and user v;

max =the maximum rating;

min=the minimum rating;

Although JMSD measures the trust of the item rating information by the user, it ignores the user's own interests and preferences. To solve this problem, a method of calculating the similarity of user preferences is presented [16], as shown in formula (5).

$$P_{u,v} = \begin{cases} e^{(-(|\mu_u - \mu_v| + 1)\cdot(|\sigma_u - \sigma_v|)), \mu_u = \mu_v \wedge \sigma_u \neq \sigma_v} \\ e^{(-(|\mu_u - \mu_v|)\cdot(|\sigma_u - \sigma_v| + 1)), \mu_u \neq \mu_v \wedge \sigma_u = \sigma_v} \\ e^{(-(|\mu_u - \mu_v|)\cdot(|\sigma_u - \sigma_v|)), else} \end{cases} \tag{5}$$

Where, μ_u and μ_v represent the mean rating value of users u and v, σ_u and σ_v represent the rating variance of users u and v. According to the dynamicity of trust, user's trust will gradually change with the interaction between users, positive interaction between users will gradually deepen the trust between users, while negative interaction will gradually weaken the trust. This idea is introduced into the calculation of trust, expressed in formula (6):

$$Inter_{u,v} = \begin{cases} 0, (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v) < 0 \\ 1, (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v) \geq 0 \end{cases} \tag{6}$$

Where, if two users scored consistently on the same item, the trust deepening of the two users is denoted by 1, and the trust attenuation is denoted by 0. X and y were used to count the number of times trust deepened and trust attenuated, x + 1 if trust deepened once and y + 1 if trust attenuated once. In summary, the direct trust value is calculated as formula (7).

$$DTru_{u,v} = JMSD_{u,v} \times P_{u,v} \times \frac{y}{x+y} \tag{7}$$

(2) Measurement of indirect trust value

In order to fully explore the trust relationship between users, the middle user is introduced to transfer the trust relationship among specific users by utilizing the characteristics of trust transitivity. Weighted average aggregation is used to calculate indirect trust values [17], as shown in formula (8).

$$PDTru_{u,v} = \frac{\sum_{w \in N(u)} DTru_{u,w} \times (DTru_{w,v} \times \beta_d)}{\sum_{w \in N(u)} DTru_{u,w}} \tag{8}$$

β_d is calculated by using the trust attenuation function in Mole Trust, as shown in formula (9).

$$\beta_d = \frac{(MPDT - d + 1)}{MPDT} \tag{9}$$

The local trust value as shwon in formula (10) between users is obtained by integrating direct trust and indirect trust as follows:

$$InDTru_{u,v} = \begin{cases} DTru_{u,v}, d = 1 \\ PTru_{u,v}, d \in [2, MPDT] \end{cases} \tag{10}$$

$z_{u,v}$ is used to represent the degree of trust from user u to user v. p_u and f_v are used to represent the latent eigenvectors of the truster u and the trusted v at the l-dimension. The trust matrix is obtained by the formula $T = P^T F$ through the eigenmatrix $P^{l \times m}$ of user u and $F^{l \times m}$ of v. The implicit trust matrix from user u to user v can be obtained by formula (11).

$$\hat{z}_{u,v} = f_v^T p_u \tag{11}$$

3.3 Latent Factor Model for Fusing Trust Relationships

Matrix decomposition provides a solution to alleviate data sparsity. Since the advent of the original SVD algorithm, much attention has been paid to it. The emergence of latent factor models has brought the research of matrix decomposition algorithm to a climax. In 2010, Koren proposed an SVD++ algorithm [18], based on the latent factor model, which takes into account user/item bias and the impact of rating items (rather than user/item specific vectors) on rating forecast. The model of SVD++ algorithm is as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^{\mathrm{T}} \cdot (p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i) \tag{12}$$

SVD++ algorithm greatly improves the accuracy of the recommendation algorithm and has a profound impact on the subsequent research of scholars. Manzato proposed the gSVD++ model [19], which takes into account not only the potential space for describing users and items, but also the available metadata related to content. The model of gSVD++ algorithm is as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + (q_i + |G(i)|^{-\frac{1}{2}} \sum_{g \in G(i)} x_g)^{\mathrm{T}} \cdot (p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i) \tag{13}$$

In this paper, explicit and implicit trust relationships of users are added to the gSVD++ model, and DT-LMF model is obtained:

$$\begin{aligned} \hat{r}_{ui} = \mu + b_u + b_i + (q_i + |G(i)|^{-\frac{1}{2}} \sum_{g \in G(i)} x_g)^{\mathrm{T}} \cdot (p_u + \\ |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j + \alpha |\mathrm{T}_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v) + (1-\alpha) |\mathrm{Z}_u|^{-\frac{1}{2}} \sum_{v \in Z_u} f_v) \end{aligned} \tag{14}$$

where α is the weight factor for explicit trust relationships, obtained from several experiments.

It is necessary to define the loss function and use the gradient descent method to obtain each parameter in the above model. The penalty factor is added in the way [20] to prevent over-fitting. The loss function with regular terms is shown in formula (15), and then the gradient descent method is used to get each parameter.

$$\begin{aligned} C = &\frac{1}{2} \sum_u \sum_{i \in I_u} (\hat{r}_{u,i} - r_{u,i})^2 + \frac{\lambda_t}{2} \sum_u \sum_{v \in T_u} (\hat{t}_{u,v} - t_{u,v})^2 + \frac{\lambda_z}{2} \sum_u \sum_{v \in Z_u} (\hat{z}_{u,v} - z_{u,v})^2 \\ &+ \frac{\lambda}{2} \sum_u |I_u|^{-\frac{1}{2}} b_u^2 + \frac{\lambda}{2} \sum_{i \in I_u} |U_i|^{-\frac{1}{2}} b_i^2 + \sum_u (\frac{\lambda}{2} |I_u|^{-\frac{1}{2}} + \frac{\lambda_t}{2} |T_u|^{-\frac{1}{2}} + \frac{\lambda_z}{2} |Z_u|^{-\frac{1}{2}}) \| p_u \|_F^2 + \\ &\sum_{i \in I_u} (\frac{\lambda}{2} |U_i|^{-\frac{1}{2}} + \frac{\lambda}{2} |G_i|^{-\frac{1}{2}}) \| q_i \|_F^2 + \frac{\lambda}{2} \sum_{j \in I_u} |U_j|^{-\frac{1}{2}} \| y_j \|_F^2 + \frac{\lambda}{2} \sum_{g \in G_i} |U_i|^{-\frac{1}{2}} \| x_g \|_F^2 + \\ &\frac{\lambda}{2} |T_v^+|^{-\frac{1}{2}} \| w_v \|_F^2 + \frac{\lambda}{2} |Z_v^+|^{-\frac{1}{2}} \| f_v \|_F^2 \end{aligned} \tag{15}$$

The notations we will use throughout the article are summarized in Table 1.

Table 1

Notations and definitions

Notation	Description
R	User-item rating matrix of N users on M items
$U \in R^{D \times N}$	User's latent matrix of feature vectors
$V \in R^{D \times M}$	Item's latent matrix of feature vectors
U_i	Specific user's latent feature vectors (column vector)
V_j	Item-specific latent feature vectors (column vector)
r_{ui}	Real rating by user u on item i
$\hat{r}_{ui}$	Prediction rating by user u on item i
$\bar{r}_i$	Average rating by user u on item i
b_{ui}	The bias involved in rating r_{ui}
μ	The overall average rating
b_u	The observed deviations of user u
b_i	The observed deviations of item i
p_u	User u is associated with a vector p_u, $p_u \in R^f$
q_i	Item i is associated with a vector q_i, $q_i \in R^f$
λ	λ is a regularization parameter
$t_{u,v}$	$t_{u,v}$ indicates the time interval

4. Experimental Results and Analysis

4.1 Introduction to dataset

Filmtrust mainly includes ratings.txt and trust.txt, covering 35,497 rating records and 1,853 trust data of 2,071 items from 1,508 users. The Epinions dataset mainly includes 664,824 rating records and 487,183 trust records of 40,163 users for 139,738 items. The 2 datasets were randomly divided into 5 parts for cross validation.

4.2 Impact of α on the algorithm

The main parameters and values involved in the experiments in this paper (obtained from repeated experiments) are shown in Table 2.

Table 2

Parameters

PARAMETERS	VALUES
Learning efficiency μ	$\mu = 0.01$
$MPDT$	$MPDT = 4$
Latent number L	$L = 40$
ITERATIONS $Epochs$	$Epochs = 40$

Regularization coefficient of *FilmTrust* dataset	$\lambda = 1.0, \lambda_t = 0.9, \lambda_z = 0.6$
Regularization coefficient of *Epinions* dataset	$\lambda = 0.9, \lambda_t = 0.6, \lambda_z = 0.5$

The impact of parameter α on the algorithm under the two different datasets is shown in Figs. 2 and 3. It is thus clear that with the increase of α in the two datasets, the RMSE gradually increases and the recommendation accuracy continues to improve, indicating that the greater the weight of the implicit trust relationship, the higher the accuracy of the recommendation system. Therefore, it can be inferred that the introduction of the implicit trust relationship in the recommendation algorithm can greatly improve the performance of the recommendation system. However, compared with explicit trust relationship, implicit trust is based on the score information between users, which is less robust and vulnerable to attack. In order to ensure the accuracy and anti-attack ability of the algorithm, α is set as 0.4 in the subsequent algorithm analysis.

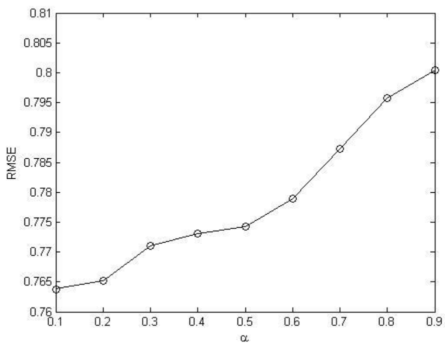

Fig. 2 Impact of α in Filmtrust

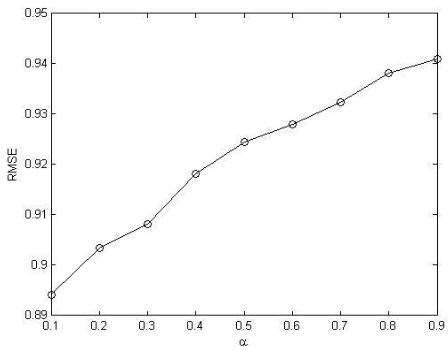

Fig. 3 Impact of α in Epinions

4.3 Impact of dimension on the algorithm

As is known to all, the latent number has a crucial influence on the algorithm in matrix decomposition. The comparison of recommended accuracy of each algorithm in different dimensions is analyzed in Filmtrust dataset and Epinions dataset. The SVD++ algorithm and TrustSVD++ algorithm are selected to compare with the DT-LFM proposed in this paper. The RMSE effects of different latent classes on each algorithm are shown in Figs. 4 and 5. According to the figures, the RMSE of all three algorithms kept decreasing with the increase of latent classes in two different datasets, especially in the initial stage; when L > 35, the RMSE gradually leveled off, while DT-LFM algorithm and TrustSVD++ are better than SVD++ algorithm, indicating that the trust relationship between users can improve the recommendation accuracy, and the RMSE indicator of DT-LFM algorithm is better than TrustSVD++ algorithm, which further proves that the implicit trust relationship can further improve the recommendation effect. L=40 is set in this chapter.

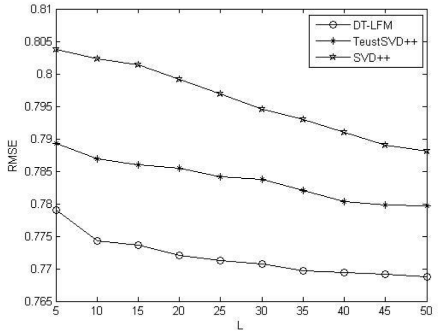

Fig. 4 Impact of L in Filmtrust dataset

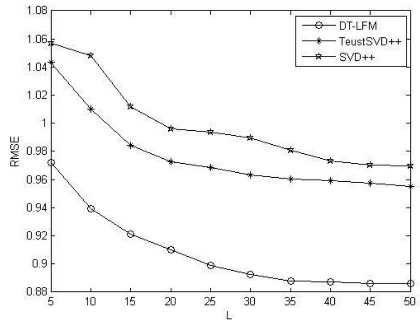

Fig. 5 Impact of L in Epinions dataset

4.4 Performance analysis

The performance of three algorithms are compared given L=40, α =0.4. Figs. 6 and 7 show that the RMSE of the three algorithms decreases rapidly when Epochs < 30 in the Filmtrust dataset, stabilizes when Epochs > 30, and stabilizes when the number of iterations reaches 25 in the Epinions dataset.

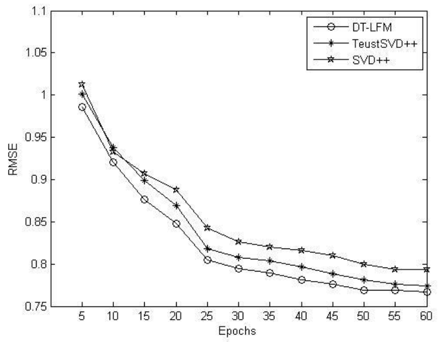

Fig. 6 Performance in Filmtrust

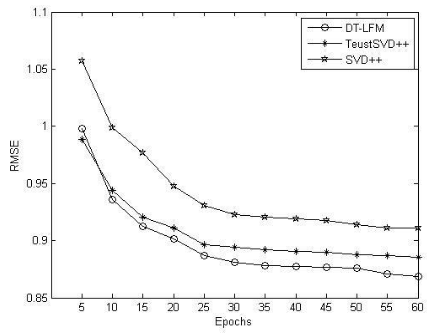

Fig. 7 Performance in Epinions

The two figures (Fig. 6-7) show that the algorithm proposed in this paper is significantly better than the other two algorithms in both Filmtrust and Epinions datasets. The specific experimental results are shown in Table 3 and Table 4.

Table 3

RMSE performance comparisons in Filmtrust dataset

	Epochs	5	10	15	20	25	30	35	40	45	50	55	60
RMSE	SVD++	1.02	0.93	0.92	0.89	0.84	0.83	0.825	0.82	0.81	0.809	0.805	0.795
	TrustSVD++	1.01	0.935	0.91	0.875	0.825	0.81	0.809	0.8	0.79	0.786	0.782	0.78
	DT-LFM	0.99	0.925	0.88	0.85	0.82	0.79	0.78	0.775	0.774	0.773	0.772	0.771

Table 4

RMSE performance comparisons in Epinions dataset

	Epochs	5	10	15	20	25	30	35	40	45	50	55	60
RMSE	SVD++	1.065	1.01	0.975	0.951	0.942	0.938	0.935	0.932	0.93	0.928	0.92	0.918
	TrustSVD++	1.01	0.94	0.92	0.92	0.918	0.91	0.90	0.895	0.891	0.885	0.88	0.875
	DT-LFM	0.99	0.938	0.916	0.91	0.893	0.89	0.883	0.881	0.879	0.87	0.879	0.868

5. Conclusions and Prospects

It is well known that matrix decomposition algorithm can alleviate data sparsity of rating matrix, and trust relationship between users can further improve the accuracy of recommendation. Based on this, in this paper, the explicit trust relationship and implicit trust relationship between users are fused on the basis of gSVD++, and the trust relationship between users is effectively combined with the matrix decomposition algorithm, that is, alleviating the data sparsity and further improving the recommendation performance. Finally, good experimental results have been obtained on both the Filmtrust dataset and the Epinions dataset.

The limitation of this study is its sole use of trust information among users. Moreover, minimal usable information is gathered because of privacy problems in social networks. Distrust information is provided in many social networks. Therefore, future research should explore ways to integrate trust and distrust information into the recommendation model.

R E F E R E N C E S

1. Gu Q, Zhou J, Ding C H. Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs//SDM. 2010: 199-210.
2. Wang J, Miao Y, Han P, et al. Probabilistic Matrix Factorization Algorithm of Collaborative Filtering Based on Trust Mechanism. Journal of Chinese Computer Systems, 2019, 40(1): 31-35.
3. Guo G, Zhang J, Yorke-Smith N. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings//Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.
4. Manzato M G. gSVD++: supporting implicit feedback on recommender systems with metadata awareness//Proceedings of the 28th Annual ACM Symposium on Applied

Computing. ACM, 2013: 908-913.
5. Gu Y, Chen M. One Tag Time-weighted Recommend Approach on Tripartite Graphs Networks. Computer Science, 2012, 39(8):96-98.
6. Shi M, Liu J, Zhou D. A Hybrid Approach for Au- TomaticMashup Tag Recommendation. Journal of Web Engineering, 2017, 16(7-8):676- 692.
7. Beldjoudi S, Seridi H, Zucker C F. Personalizing and Imp- roving Resource Recommendation by Analyzing Users Pre- ferences in Social Tagging Activities. Computing & Info- rmatics, 2017, 36(1):223-256.
8. Golbeck J A. Computing and applying trust in web-based social networks. PhD thesis, University of Maryland College Park, 2005.
9. Massa P, Avesani P. Trust-aware recommender systems. Proceedings of the 2007 ACM conference on Recommender systems. ACM, 2007: 17-24.
10. Jamali M, Ester M.A matrix factorization technique with trust propagation for recommendation in social networks. Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 135-142.
11. Guo Lei, Ma Jun, and Chen Zhumin. Trust Strength Aware Social Recommendation Method. Journal of Computer Research and Development, 2013, 50(9): 1805-1813.
12. Qin JW, Zheng QH, Tian F, Wang K. Collaborative filtering algorithm integrating trust and preference of user’s emotion. Journal of Software, 2013, 24 (2):61−72.
13. Zou BY, Li CP, Tan LW, Chen H, Wang SQ. Social recommendations based on user trust and tensor factorization. Journal of Software, 2014, 25(12):2852−2864.
14. Zhang Z, Liu H. Social recommendation model combining trust propagation and sequential behaviors. Applied Intelligence, 2015, 43(3): 695-706.
15. Bobadilla J, Serradilla F, Bernal J. A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 2010, 23(6): 520-528.
16. Zheng P, Wang Y, Liang W. Collaborative filtering recommendation algorithm based on trust and matrix factorization. CEA, 2018, 54(13): 34-40.
17. Malinowski J, Weitzel T, KeimT. Decision support for team staffing: An automated relational recommendation approach. Decision Support Systems, 2008, 45 (3): 429-447.
18. Koren Y. Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Transactions on Knowledge Discovery from Data (TKDD), 2010, 4(1): 1-24.
19. Manzato M G. gSVD++: supporting implicit feedback on recommender systems with metadata awareness//Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, 2013: 908-913.
20. Guo G, Zhang J, Yorke-Smith N. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings//Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.