



Social Recommendation Algorithm Research based on Trust Influence

By Xue Yi, Hong Yinghan & Chen Pinghua

Guang Dong University of Technology, China

Abstract- Cold start and data sparsity greatly affect the recommendation quality of collaborative filtering. To solve these problems, social recommendation algorithms introduce the corresponding user trust information in social network, however, these algorithms typically utilize only adjacent trusted user information while ignoring the social network connectivity and the differences in the trust influence between indirect users, which leads to poor accuracy. For this deficiency, this paper proposes a social recommendation algorithm based on user influence strength. First of all, we get the user influence strength vector by iterative calculation on social network and then achieve a relatively complete user latent factor according to near-impact trusted user behavior. Depending on such a user influence vector, we integrate user-item rating matrix and the trust influence information. Experimental results show that it has a better prediction accuracy, compared to the state-of-art society recommendation algorithms.

Keywords : *collaborative filtering; cold start; data sparsity; social network; trust influence.*

GJCST-C Classification : *K.4.2 B.2.4*



Strictly as per the compliance and regulations of:



Social Recommendation Algorithm Research based on Trust Influence

Xue Yi^α, Hong Yinghan^σ & Chen Pinghua^ρ

Abstract- Cold start and data sparsity greatly affect the recommendation quality of collaborative filtering. To solve these problems, social recommendation algorithms introduce the corresponding user trust information in social network, however, these algorithms typically utilize only adjacent trusted user information while ignoring the social network connectivity and the differences in the trust influence between indirect users, which leads to poor accuracy. For this deficiency, this paper proposes a social recommendation algorithm based on user influence strength. First of all, we get the user influence strength vector by iterative calculation on social network and then achieve a relatively complete user latent factor according to near-impact trusted user behavior. Depending on such a user influence vector, we integrate user-item rating matrix and the trust influence information. Experimental results show that it has a better prediction accuracy, compared to the state-of-art society recommendation algorithms.

Keywords: collaborative filtering; cold start; data sparsity; social network; trust influence.

I. INTRODUCTION

As the third Internet revolution after the search engine, recommendation systems, without the need for exogenous information about either items or users, achieve the initiative to push personalized service by conducting analysis based on user past online behavior, such as ratings or usage. Collaborative filtering which is the classical recommendation algorithm, comprises memory and model based methods [1]. To solve the data sparsity and cold start conundrum, social recommendation has gradually become one of the important research fields in the recommender system. It usually introduces the corresponding user trust information in social network, and its underlying assumptions are: people tend to interact with the people who have similar interests and preferences, and in such process they would be more similar with each other [2].

Social network information can indeed be used to improve the accuracy of prediction score, but how to fully exploit the social network connection feature information becomes a hot topic among a number of researchers recent years. [3] proposed an approach (SoRec) based on probabilistic matrix factorization by assuming the user-specific and factor-specific latent

feature follows Gaussian distribution, combined user trust information matrix T with user-item rating matrix R to constrain user latent feature vector, and then used the result to calculate rating matrix to predict; a combination of matrix factorization model and trust neighborhood-based model was proposed in [4] (RSTE); [5] presented an method that through active neighbors feature vectors to obtain the current user latent feature, and all user-specific matrix U , and then multiply the latent item-specific matrix V to forecast user-item ratings (Social MF).

These studies are mainly concentrated on the users directly connected each other in a social network, while ignoring the differences among user influence through out the network. [6] carried on a research from the perspective of trust and trusted, then combined each other with the addition (Trust MF). However, actually, trust and trusted information are not inseparable in a real a social network, and such trust information can be propagated, i.e. the user is not only influenced by neighbors, but also the users of greater influence in the network, in addition, during the stage that the trust and trusted models are mixed, since the trust and trusted detached from each other, tuning parameters becomes more difficult, resulting in poor interpretability even for a better prediction. Therefore, these algorithms above exist poor prediction accuracy issues.

In this paper we proposes a social recommendation algorithm based on the strength of trust influence: First of all, we conduct iterative calculation by using the connectivity of the trust network to obtain user-specific impact factors in the whole social trust network. Secondly, according to the influence differences among users, we constrain the user-specific latent factor matrix during the process of matrix decomposition, and by multiplying the user-specific and item-specific latent factor matrix to seek more accurate prediction scores.

II. PRELIMINARIES

Conveniently, we describe the basic algorithm by the following example, while denote user User1 abbreviated as U1, as well Item1 as I1. Figure 1 is an abstraction for the real social network, in which each circle represents a user and the arrow points to the ones that can be trusted, e.g. user U3 trusted user U1, while

Author α σ ρ : School of Computer, Guang Dong University of Technology, Guangdong Guangzhou, 510006, China.
e-mail: andy.xueyi@gmail.com

the user U1 does not trust U3. We also use matrix to represent the trust information in the network, where 1 represents the trust, 0 mistrust, thus the matrix is non-symmetric matrix, denoted by T_n , n is the number of nodes, as shown in Table 1. User-item ratings information is shown as Table 2. We integrate the two

datasets as input to get user-specific latent factor U and item-specific latent factor V by matrix decomposition, and then use the product of the two factors to get predicted ratings \hat{R} , as shown in table 3. We distinguish predicted ratings from known ones, by using the notation \hat{R} for the predicted value of R .

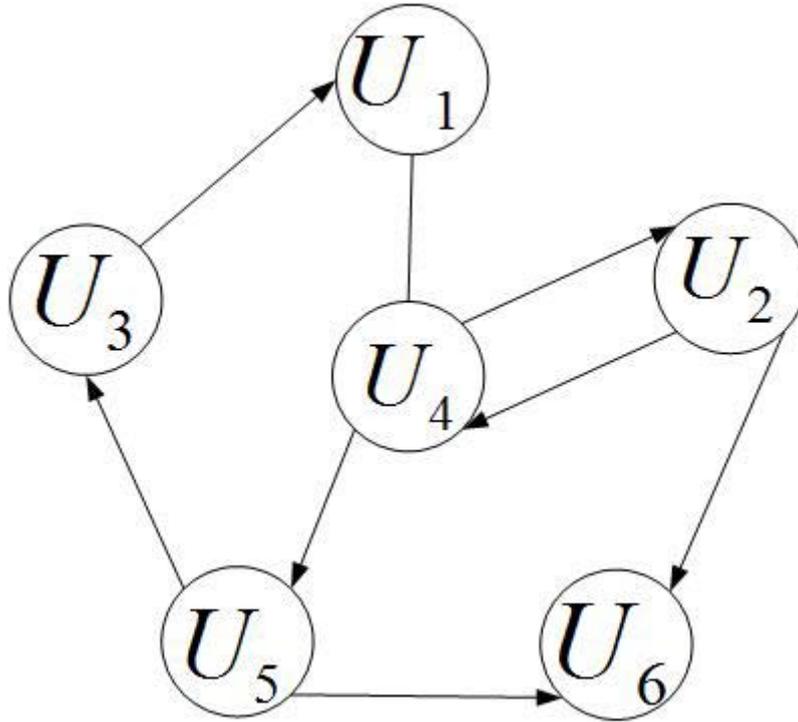


Fig. 1: Abstraction of Social Trust Network

Table 1: User trust information Matrix U

	U1	U2	U3	U4	U5	U6
U1	0	0	0	0	0	0
U2	0	0	0	1	0	1
U3	1	0	0	0	0	0
U4	1	1	0	0	1	0
U5	0	0	1	0	0	1
U6	0	0	0	0	0	0

Table 1 : user-item rating matrix V

	I1	I2	I3	I4	I5	I6
U1	5	2		3		4
U2	4	3			5	
U3	4		2			
U4						
U5	5	1	2		4	3
U6	4	3		2	4	

Table 2 : predicted user-item rating matrix \hat{R}_{ij}

	I1	I2	I3	I4	I5	I6
U1	5	2	2.5	3	4.8	4
U2	4	3	2.4	2.9	5	4.1

U3	4	1.7	2	3.2	3.9	3.0
U4	4.8	2.1	2.7	2.6	4.7	3.8
U5	5	1	2	3.4	4	3
U6	4	3	2.9	2	4	3.4

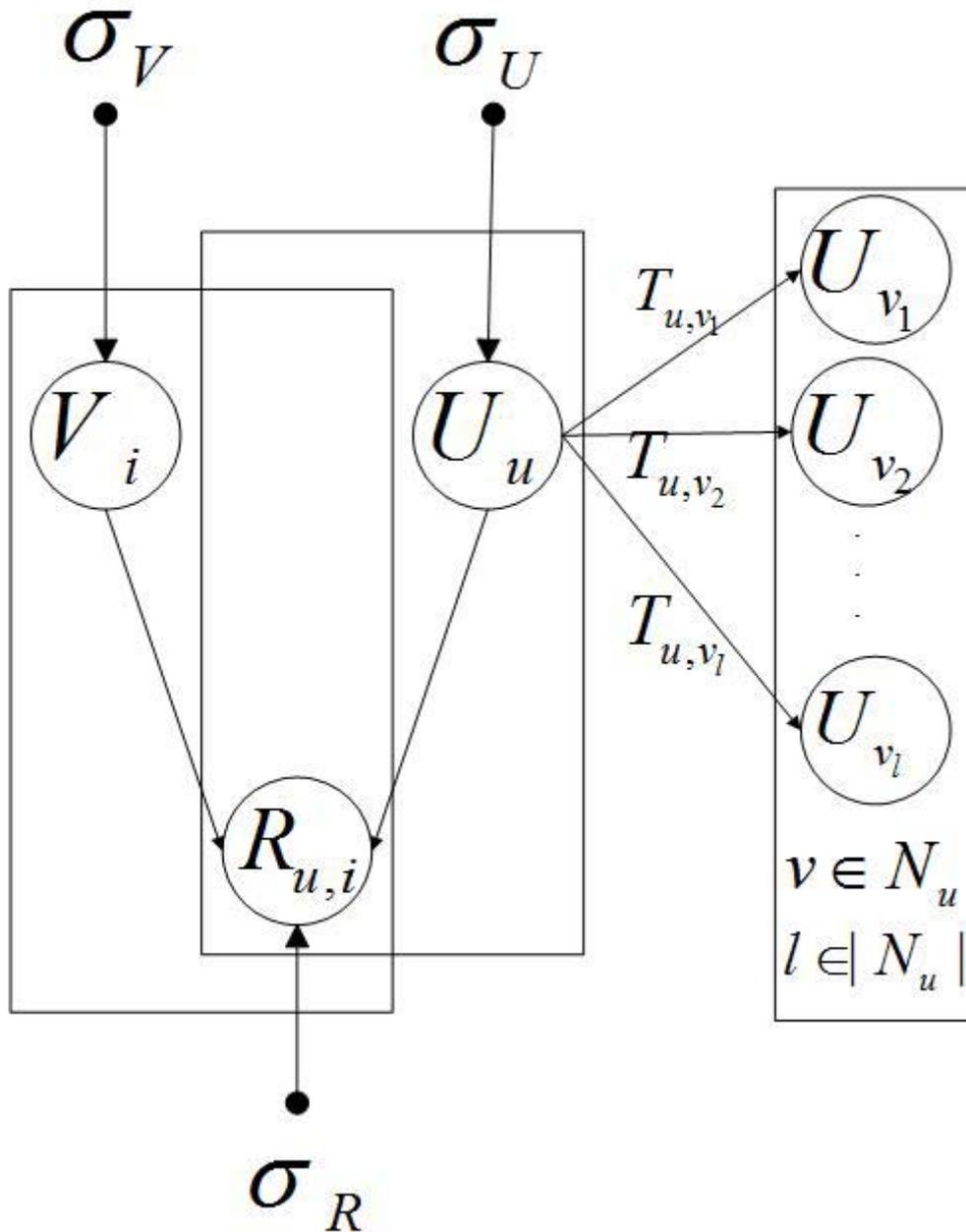


Fig. 2 : Factorization Diagram based on Rating Matrix and Social Network

According to the literature [7], introducing the social network into the user-rating ratings matrix can be treated as the rating matrix decomposition with a joint trust network matrix decomposition, as figure 2, they both constrain the user-specific latent factor U . We first define user-item rating matrix decomposition method, specifically address how to introduce a social network approach in the next section. We assume that the

recommendation system involving m users and n items, the user-items rating matrix denoted as $R = [R_{ij}]_{mn}$, in which each rating R_{ij} indicates the preference by user i of item j , where high values mean stronger preference, e.g. values can be integers ranging from 1 to 5. We denote an labeled matrix to show whether the user rates the item in user-item rating matrix

as $I = [I_{ij}]_{mm}$, if the user i has rated item j , set I_{ij} is 1, otherwise 0. By matrix factorization we can map the higher dimension matrix into a lower d dimension matrix. We denote user i latent factor as U_i , while item j latent factor as V_j , thus can get user-specific latent factor $U \in R^{d \times m}$, and item-specific latent factor $V \in R^{d \times n}$

$$L = \sum_{i=1}^m \sum_{j=1}^n I_{ij} (U_i^T V_j - R_{ij})^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_F^2 \tag{1}$$

In the formula, we denote $\|\cdot\|_F^2$ as Frobenius norm, and in case of over-fitting, set parameters $\lambda_1, \lambda_2 > 0$ to control model complexity. In addition, a regularization method for weighting λ was proposed in [8] to avoid the parameters above in the model learning process, λ_1, λ_2 respectively account for user-specific and item-specific latent factor in the whole training

$$L = \sum_{i=1}^m \sum_{j=1}^n I_{ij} (U_i^T V_j - R_{ij})^2 + \lambda_1 \sum_i n_{u_i} \|U_i\|_F^2 + \lambda_2 \sum_j n_{v_j} \|V_j\|_F^2 \tag{2}$$

and then use the product of user-specific and item-specific latent factor by $U^T V$ to fit the available user-item ratings matrix R , whereby item not rated by users can be filled with a predicted score. The main learning function is obtained by minimizing the loss function as follows:

model, at the same time, the introduction both of number of user U_i ratings and that of items rated is used to prevent the trained mode tilt to users who rated items too many or items rated by too many users, resulting in over-fitting problem, therefore, for the appropriate model items and users can be regularized as follows:

III. ALGORITHM DESCRIPTION

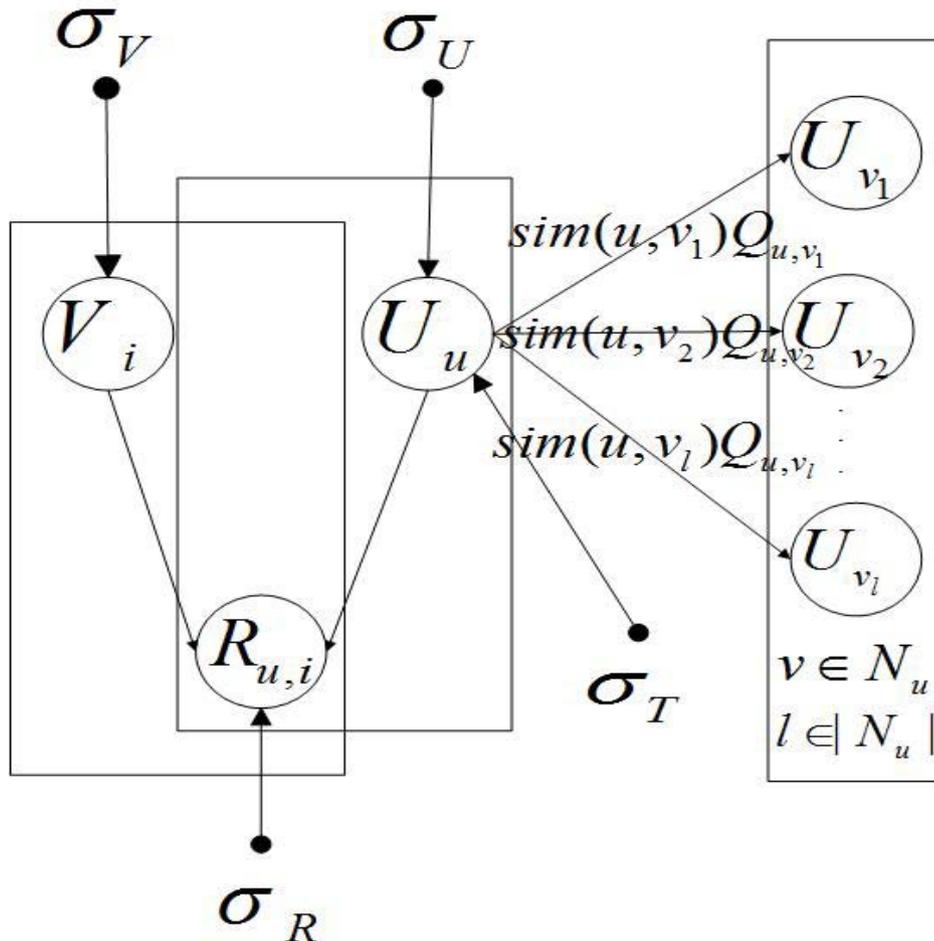


Fig. 3 : Our improved Algorithm T

The primary objective of this algorithm is to obtain the impact factors of user trust influence in the entire social network, which is utilized to measure the degree of users affected by their neighbors. We refer to the idea of Page Rank algorithm: the larger the number of user trusted throughout the trust network is, the bigger the trust influence such user has. We denote that there are m users involved in the ratings matrix R and trust matrix T , therefore, we can simultaneously both

$$PR(i) = \frac{1-d}{M} + d \cdot \sum_{j \in T_i} \frac{PR(j)}{O_j} \tag{3}$$

M represents the number of whole nodes in the network, and $\sum_{i=0}^N \|PR(i)\| = 1$, T_i is denoted as user set that user i trusts, O_i is the number trusted by user i , and $d(0 \leq d \leq 1)$ is the damping factor, usually set to

$$E_{n+1} = dT_0E_n + \frac{1-d}{N}I^+ \tag{4}$$

E_i represents the whole users trust influence in the i time iterative calculation, T_0 represents the with T_0 . When $n \rightarrow +\infty$, i.e. the matrix steps into the steady state:

$$E = dT_0E + \frac{1-d}{N}I^+ \tag{5}$$

Whereby the equation (5) can be simplified as equation (6):

$$E = (I - dT_0)^{-1} \frac{1-d}{N}I^+ \tag{6}$$

When $0 < d < 1$, the solution exists. Using the iterative calculation described as equation (5), with the number increases, the model will gradually converge [14], then we can get entire user trust influence vector $E = [e_i]_{m \times 1}$, in the section above, we proposed a regularized method on social networking information. As for social network research, the majority of studies are

$$L = \sum_{i=1}^m \sum_{j=1}^n I_{ij} (U_i^T V_j - R_{ij})^2 + \lambda_1 \sum_i n_{u_i} \|U_i\|_F^2 + \lambda_2 \sum_j n_{v_j} \|V_j\|_F^2 + \lambda_3 \sum_{i=1}^m \sum_{j \in \Psi} Sim(i, j) \|U_i - U_j\|^2 \tag{7}$$

$Sim(i, j)$ is on behalf of cosine similarity between the user i and adjacent user j , Ψ is a labeled matrix indicating whether the user is directly connected to all the others in the matrix. Taking into account the diversity of user preferences, we thus make regularization to balance user similarity and user-specific latent factor, due to the fact that when two users

$$Q_{ij} = \frac{|e_j|}{\sum_{k \in \Psi} |e_k|} \tag{8}$$

Q_{ij} indicates the trust proportion of user j in all user i directly connected users, we hope that user-specific latent factor tends to the overall average users, while taking into account the influence of adjacent users throughout the trust network, so we proposed a method which integrates both neighborhood similar user

matrix decomposition operation by the same user-specific latent factor U . As mentioned in the section 1, for a given user trust network, denoted as $G := (M, N)$, which M represents user nodes, $T = [t_{m,n}]_{M, M}$ is denoted as adjacency labeled matrix, if user m trust user n , then $t_{m,n} = 1$, $t_{m,n} = 0$ and vice versa. Thus the user i trust influence can be defined as:

0.85 [9]. Equation (3) only describes solution to achieve the influence of a particular user, so as to calculate the influence of the overall user-specific latent vector, it can be expressed as the formula (4):

original large-scale sparse trust matrix, I^+ is on behalf of a matrix filled with 1, and has the same dimension

based on the similarity of adjacent users, while ignore the fact that user trust someone does not mean they share a reliable presence of the same preference. If user trusted also has the same presence of target user behavior history information, the two are more likely to have same similar preferences, the objective function can be adjusted to the formula (7):

are similar, $Sim(i, j)$ will become very large, and the user latent factor $\|U_i - U_j\|^2$ will become very small, and vice versa. In addition, users are more inclined to follow the similar users who are more influential and keep consistent in behavior with them, and that is the reason why we introduce the strength of trust influence, equation (8) is defined as below.

characteristics and their trust influence factors to associate a user-item ratings and trust network information, resulting in a more accurate model. In order to facilitate learning and weight adjustment, at first, we handle the data with normalization, as the trust value Q_{ij} is between 0 and 1, we normalize the original data by

using mapping function $f(x) = x / R_{max}$, so that its final value is between $[0,1]$. According to the literature [5], a logistic function $g(x) = 1 / (1 + \exp(-x))$ was proposed to make the product of latent vectors to $[0,1]$, so after the training, we can get the final predicted result by

$$L = \sum_{i=1}^m \sum_{j=1}^n I_{ij} (g(U_i^T V_j) - R_{ij})^2 + \lambda_1 \sum_i n_i \|U_i\|_F^2 + \lambda_2 \sum_j n_j \|V_j\|_F^2 + \lambda_3 \sum_{i=1}^m \sum_{j \in \Psi} Sim(i, j) \|U_i - Q_j U_j\|^2 \quad (9)$$

During the training phase, we adopt the gradient descent method to minimize the objective

$$\frac{1}{2} \cdot \frac{\partial L}{\partial U_i} = \sum_{j=1}^n I_{ij} g'(U_i^T V_j) (U_i^T V_j - R_{ij}) V_j + \lambda_1 \sum_j n_j U_i + \lambda_3 \sum_{j \in \Psi} Sim(i, j) (U_i - Q_j U_j) \quad (10)$$

$$\frac{1}{2} \cdot \frac{\partial L}{\partial V_j} = \sum_{i=1}^m I_{ij} g'(U_i^T V_j) (U_i^T V_j - R_{ij}) U_i + \lambda_2 \sum_i n_i V_j \quad (11)$$

When using Pearson correlation coefficients to conduct calculation of cosine similarity, since the scope of its value between $[-1,1]$, for taking a positive number, we use the mapping function $f(x) = (1 + x) / 2$ instead. $Sim(i, f)$ is the similarity value for user i and user f ,

$$Sim(i, f) = \frac{\sum_{j \in (I(i) \cap I(f))} (R_{ij} - \bar{R}_i) \cdot (R_{if} - \bar{R}_f)}{\sqrt{\sum_{j \in (I(i) \cap I(f))} (R_{ij} - \bar{R}_i)^2} \sqrt{\sum_{j \in (I(i) \cap I(f))} (R_{if} - \bar{R}_f)^2}} \quad (12)$$

IV. COMPLEXITY ANALYSIS

The model training time consists two parts, the first part is the algorithm calculate the user trust influential feature vector, the second parts is for solving the latent factors by using the gradient descent method. We conduct iterative calculation to get the trust influential feature vector, if the entire number of nodes in the trust network is m , the average number of users trusted by per user is n , usually n is a relatively small number. Given t_1 times of iterations to achieve convergence, time complexity is $O(mnt_1)$, when the data is in large scale, the literature [10] proposed a distributed solution that can significantly reduce the time.

The time complexity of solving L is

$O(t_2 d (\Omega + N))$. t_2 is a specified number of iterations, d is the dimension of latent factors, $|\Omega|$ is the scale of the observed ratings, N represents time complexity of similarity calculation, since the matrix is so sparse that the time complexity $O(N) \ll O(n^2)$. After the combination of these two steps, the total time complexity of the algorithm model is $O(t_2 d (\Omega + N) + mnt_1)$.

$g(U_i^T V_j) \cdot R_{max}$. In view of keeping the same benchmarks among user-item ratings, social network data and the same constraint variable U , the new objective function is updated to equation (9):

function for solving the above-mentioned latent factors matrix, respectively U , V .

R_{ij} represents the rating of user i for item j , \bar{R}_i indicates user i average rating score, $I(i)$ is a labeled matrix indicating whether user i rate the item or not, as well as user f .

V. EXPERIMENTS AND RESULTS

a) Dataset Selection

The classical data set were Epinions (665KB) [11], which contains not only the user-item ratings data, but also trust relationship between users, and thus it is usually recommended as baseline data set to test social recommendation algorithms. We use it for testing and validating the currently mainstream recommendation algorithms as well as our improved algorithm. First of all, we made the basic statistics of the data set as shown in Table 4. Trust (degree) and trusted (out-degree) statistics information shows that when the trust or be trusted number of persons increases, the corresponding statistics number gradually reduces, which follows the power law distribution.

b) Cross-Validation

We use 5-fold cross-validation methods for training and testing the models. For each test we randomly selected 80% of the whole data as the training data set and the remaining 20% is used for testing. The next experiment results discussed in the final comparison is obtained by averaging the results of tests from five repeated times.

c) Evaluation

Evaluation criteria used in the Experiments are based on the average absolute error MAE and root mean square error RMSE:

$$MAE = \frac{\sum_{i,j} |R_{ij} - \hat{R}_{ij}|}{N} \quad (13)$$

$$RMSE = \sqrt{\frac{\sum_{i,j} |R_{ij} - \hat{R}_{ij}|^2}{N}} \tag{14}$$

Table 4 : Basic statistics of the data set Epinions

Basic Data				Scenes			Others	
Users	Items	Rating Scale	Density	Users	Link Types	Items	tags	
40,163	139,738	664,824 [1,5]	0.000118458	66,807	487,183 Trust	General	None	

Table 6 : Results in Dimension 5 and 10

	Evaluation	SoRec	RSTE	SocialMF	TrustMF	T
d=5	MAE	0.9197	0.8635	0.8826	0.8212	0.8010
	RMSE	1.151	1.1071	1.1107	1.0585	1.0320
d=10	MAE	0.9152	0.8572	0.8567	0.8148	0.8059
	RMSE	1.1773	1.1483	1.1113	1.0771	1.0408

From the table, we can conclude that, regardless of index value of MAE or RMSE, the smaller its value is, the more accurately the algorithm predicts the score.

d) Comparison

In consideration of the similar algorithms, our main work is integrating users trust influence in the social network into the user-item ratings. Thus while making the lateral comparisons, in addition to the

original matrix decomposition algorithms, we also made comparisons among social recommendation algorithms which are focus on trust.

e) Contrast Algorithms and Reasons

When training the models, we set parameters of each algorithms by referring to the literature [8] [9], which introduced the optimal parameters of each algorithms. Our algorithm is denoted as T and more details described as the following table 5:

Table 5 : Contrast Algorithm and Params Settings

Algorithm Name	Parameters
SoRec	$\lambda_u = \lambda_v = \lambda_z = 0.001, \lambda_c = 1$
RSTE	$\lambda_u = \lambda_v = 0.001, \alpha = 0.4$
SocialMF	$\lambda_u = \lambda_v = 0.001, \lambda_r = 1$
TrustMF	$\lambda = 0.001, \lambda_r = 1$
T	$\lambda_1 = \lambda_2 = 0.001, \lambda_3 = 1$

For all the algorithms above based on matrix decomposition, the dimension of latent factors are set to 5 and 10 respectively, and during model training stage, the same initialized strategy is adopted: the original matrixes involved are filled with a random number uniformly distributed values between 0 and 1.

f) Results

Experiment results show that with the increase of the number of iterations, the index value of RMSE and MAE keeps declining, which indicates the introduction of the trust influence factors has a positive impact on enhancing the overall effectiveness of recommendation and improves the accuracy of predicted results. On the dataset, we have achieved results as showed in Table 6, while our algorithm is denoted as T. Cross-validation results show the performance of the our proposed algorithm is slightly better than all the other compared algorithms. In addition, we conducted statistical analysis of RMSE and MAE index value during the iterative calculation and found that with increasing number of iterations, RMSE and MAE index value gradually reduce and eventually keep stabilized, indicating that the algorithm utilizes influence characteristics among users

in the trust network to predict user rating propensity is relatively effective and eventually enhances the accuracy of the prediction score of recommendation.

VI. CONCLUSION

As the common recommendation system algorithm, collaborative filtering encountered sparse data and cold start problems, which leads to poor accuracy. Given the fact that the impact of opinion leaders for individuals in the social network, this paper proposed one social recommendation algorithm based on trust influence to alleviate such problems above. Compared to the previous research merely on adjacent user preferences, we consider the trust propagation mechanism and try out iterative calculations based the connectivity of social trust network to obtain user influence values, and then integrate both similar neighbor characteristics and their trust influence factors. Experiments show that compared the most state-of-art social algorithms, our approach optimizes the recommended results and improves the prediction accuracy. However, obviously, this algorithm complexity positively correlates with the number of iterations, when

faced with large-scale data, the computation time is too long, especially for single node. Therefore, the next research focus on the implementation of distributed algorithm for the iterative calculations involving large-scale data.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(12).
2. Kautz, H., Selman, B., & Shah, M. (1997). Referral web: combining social networks and collaborative filtering. *Communications of the Acm*, 40(3), 63-65.
3. Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). So Rec: Social Recommendation Using Probabilistic Matrix Factorization. *International Conference on Information & Knowledge Management* (pp.931-940). ACM.
4. Ma, H., King, I., & Lyu, M. R. (2009). Learning to recommend with social trust ensemble. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp.203-210). ACM.
5. Jamali M, Ester M.(2010). A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the fourth ACM conference on Recommender systems* (pp.135-142). ACM.
6. Yang, B., Lei, Y., Liu, D., & Liu, J. (2013). Social collaborative filtering by trust. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence* (pp.2747-2753). AAAI Press.
7. Salakhutdinov, R., & Mnih, A. (2012). Probabilistic matrix factorization. *Nips*, 1257-1264.
8. Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). Recommender systems with social regularization. *Fourth Acm International Conference on Web Search & Data Mining* (pp.287-296). ACM.
9. Amy N. Langville, & Carl D. Meyer. (2003). Deeper inside pagerank. *Internet Mathematics*, 1(3), 335-380.
10. Sarma, A. D., Molla, A. R., Pandurangan, G., & Upfal, E. (2013). *Fast Distributed Page Rank Computation*. *Distributed Computing and Networking*. Springer Berlin Heidelberg.
11. Epinions data set: <http://www.trust.org/wiki/Epinions>

