

RESEARCH

Open Access



# A social trust and preference segmentation-based matrix factorization recommendation algorithm

Wei Peng and Baogui Xin\*

## Abstract

A recommendation can inspire potential demands of users and make e-commerce platforms more intelligent and is essential for e-commerce enterprises' sustainable development. The traditional social recommendation algorithm ignores the following fact: the preferences of users with trust relationships are not necessarily similar, and the consideration of user preference similarity should be limited to specific areas. To solve these problems mentioned above, we propose a social trust and preference segmentation-based matrix factorization (SPMF) recommendation algorithm. Experimental results based on the Ciao and Epinions datasets show that the accuracy of the SPMF algorithm is significantly superior to that of some state-of-the-art recommendation algorithms. The SPMF algorithm is a better recommendation algorithm based on distinguishing the difference of trust relations and preference domain, which can support commercial activities such as product marketing.

**Keywords:** Personalized recommendation, Collaborative filtering, Data sparsity, Machine learning, Trust relationship, Preference similarity

## 1 Introduction

The Internet has brought about industrial change and nurtured e-commerce. E-commerce has generated huge amount of network information, which results in information overload [1]. Information overload directly increases the difficulty of selecting products and inspires people to seek effective solutions. Today, there are mainly four types of solutions we can employ: (1) information acquisition timelines, (2) categories, (3) search engines, and (4) personalized recommendations.

The first type of solution can save information retrieval time, but it is easy to miss lots of useful information. The second type of solution is to classify the project according to the similarity feature chosen by the user; this can overcome the defects of the first type of solution, but it has the disadvantage of low efficiency and poor precision. The third type of solution allows users to retrieve and filter irrelevant information by keywords, which can solve the problem of the second type of solution but cannot consider the user's individualized

needs. Finally, the fourth type of solution can solve the problem of the third solution, and through historical data, user attributes, product attributes, and other information to mine user preferences, it can actively make personalized recommendations to users [2, 3].

A recommendation system can inspire the potential needs of users and make an e-commerce platform more intelligent and humanized. Such systems have helped Amazon, JD, Alibaba, and other companies to significantly increase sales. The recommendation algorithm is the core of the recommendation system [4], but data sparsity and other problems have always been obstacles to its further development. Most scholars [5] employ machine learning algorithms, such as those for clustering and dimensionality reduction, to fill sparse data with a small amount of original data. However, it is difficult to guarantee the quality of the filled data.

In order to solve the above problems and improve the accuracy of recommendation, many scholars [6–9] put forward recommendation algorithms based on the social network, that is, using direct or indirect trust relationships to make recommendations for target users. However, the trust relationship-based algorithm still has

\* Correspondence: [xin@tju.edu.cn](mailto:xin@tju.edu.cn)  
College of Economics and Management, Shandong University of Science and Technology, Qingdao 266590, China

three problems: (1) The preference similarity of users with the trust relationship may be small or even zero. For example, a user has a trust relationship with his parents, but their preference differences may be significant, and therefore the recommendation may not be ideal. (2) The diversity of user preference determines that the measurement of preference similarity should be limited to certain areas. For example, the preference similarity of users is low in the music field, but it may be very high in the film field, and therefore a recommendation in the film field is very satisfactory. (3) Even if the preferences of different users are very similar, the recommendation will significantly influenced because of the difference in trust relationships.

In the light of the shortcomings of traditional social recommendation, our main contributions include that we proposed a social trust and preference segmentation-based matrix factorization recommendation algorithm by setting different recommendation trust weights for different relationships and preference domains. With the social trust and preference segmentation-based matrix factorization (SPMF) algorithm, we solve the problems mentioned above and improve the recommendation accuracy.

The remainder of this paper is organized as follows. In Section 2, we review some related literature. In Section 3, we propose a social trust and preference segmentation-based matrix factorization recommendation algorithm. In Section 4, we introduce the

experiments on two public datasets. In Section 5, we present the experimental results and perform some parameter analysis. In Section 6, we conclude our work and future work.

### 2 Related work

The context of the literature in this paper is shown in Fig 1. As the level-1 classification shows, [8, 10] divided the recommendation algorithms into three categories: content-based recommendation algorithms [11], collaborative filtering algorithms [12], and hybrid recommendation algorithms [13]. Strictly speaking, the content-based recommendation algorithm is derived from the collaborative filtering recommendation algorithm, and both generate recommendations for target users based on similarity. Content-based recommendation algorithms need to filter massive amounts of information and update user profiles regularly, resulting in high time complexity and unsatisfactory recommendation results. Conversely, collaborative filtering recommendation algorithms only use neighbors with high similarity to target users to evaluate a product, predict the preferences of target users, and make recommendations. Hybrid recommendation algorithms are designed to meet a specific need and incorporate content-based recommendation algorithms and collaborative filtering recommendation algorithms.

The collaborative filtering algorithm is the most successful and widely used personalized recommendation

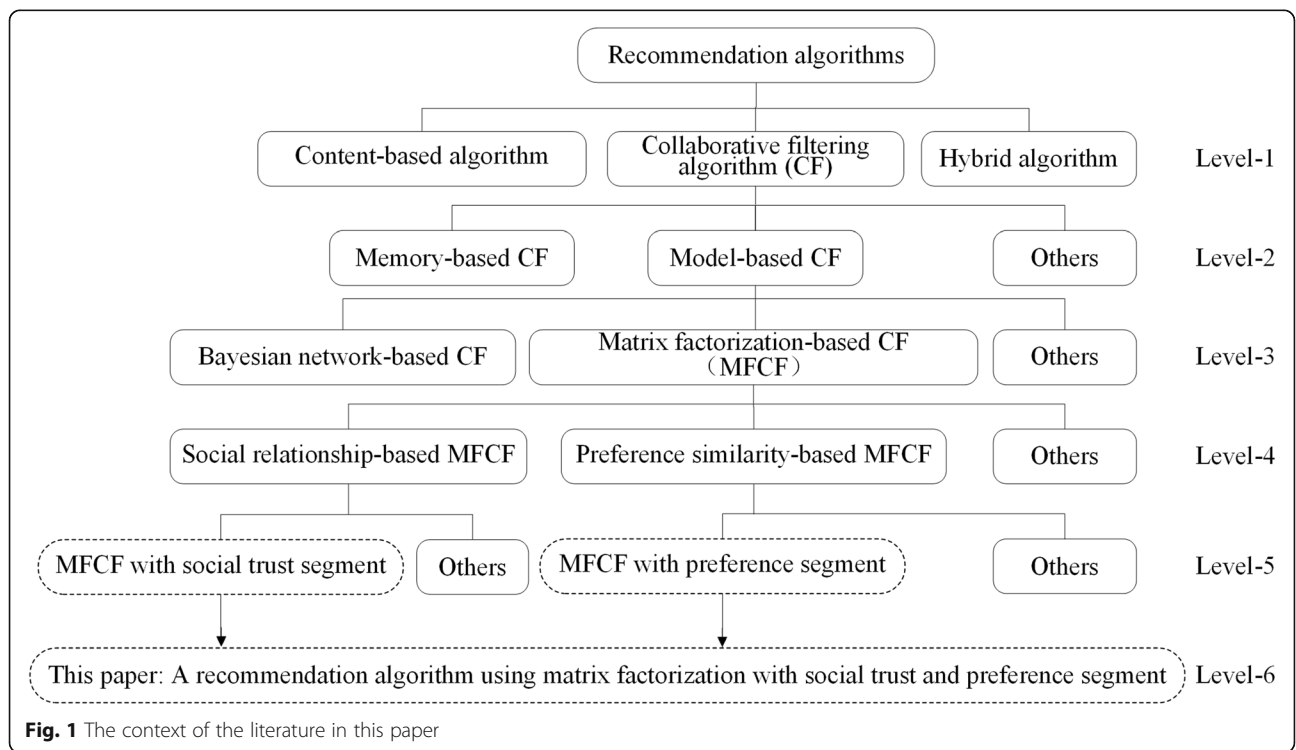


Fig. 1 The context of the literature in this paper

algorithm in business, and it is also a hotspot of academic research. As the level-2 classification shows [14]), this algorithm was divided into a memory-based collaborative filtering algorithm and a model-based collaborative filtering algorithm. The memory-based collaborative filtering algorithm does not distinguish the rated item information attributes, and therefore it directly uses the correlation matrix to make predictions, which results in a heavy workload and low efficiency. In order to improve the recommendation efficiency and accuracy, the model-based collaborative filtering algorithm employs machine learning and data mining models such as the Bayesian network [15], SVM [16], or matrix decomposition [17] for recommendation; this is shown in the level-3 classification. However, the ability to solve the sparsity problem of user-item rating data is an important indicator for evaluating the pros and cons of a recommendation algorithm. The matrix factorization-based collaborative filtering algorithm has become one of the most popular algorithms in the last decade due to its outstanding performance in the 2009 Netflix Prize competition.

As the level-4 classification shows, the matrix factorization-based collaborative filtering algorithm includes the trust relationship-based matrix factorization algorithm [18, 19] and the preference similarity-based matrix factorization algorithm [20]. Wang et al. [21] solved the problem of low-accuracy caused by sparse data by integrating the user preference similarity and matrix factorization algorithm and combining the user rating data. Lai et al. [22] constructed a social recommendation model that integrated trust relationships and product popularity and speculated on their potential interactions based on user interaction behavior. The trust relationship could lead to more accurate recommendations. Guo et al. [23] proposed TrustSVD, a trust-based matrix factorization technique for recommendations. Lee and Ma [24] constructed a recommendation algorithm that combined the KNN and matrix factorization by using a trust relationship and propagation effect, and combining user rating and the trust relationship, they achieved a higher recommendation accuracy. Apart from the classifications mentioned above, there are also some other recommendation algorithms [25–27]. Qi et al. [25] proposed SerRec<sub>two-LSH</sub>, a novel service recommendation method to provide privacy-preserving and scalable mobile service recommendations. Moreover, SerRec<sub>two-LSH</sub> achieved significant improvement both in accuracy and scalability, while privacy preservation is guaranteed. Gong et al. [26] extended the traditional LSH-based service recommendation to propose the SerRec<sub>multi-qos</sub> to protect users' privacy over multiple quality dimensions.

In this paper, we will combine the social trust segmentation and the preference domain segmentation and

construct the level-6 classification, i.e., the SPMF recommendation algorithm.

### 3 Methodology

Traditional social recommendation algorithms ignored the following fact: the preferences of users with trust relationships are not necessarily similar, and the consideration of user preference similarity should be limited to specific areas. There has no literature proposed to fulfill the literature gap mentioned above. Based on this, we proposed the SPMF recommendation algorithm.

The main ideas of the SPMF recommendation algorithm are as follows:

- (1) To show the impact of preference domain segmentation on target user recommendation.
- (2) To reveal the different influences on target users' recommendation between users with and without trust relationships.

#### 3.1 Preference domain segmentation

User preferences are diverse, but most of the existing recommendation algorithms do not take full account of it. To demonstrate the impact of preference domain segmentation on measuring user preference similarity, we provide the following example.

A website provides ten products that can be subdivided into three categories: music, movies, and books (Table 1). The numbers in the table represent rating records. The rating scale is 1–5, and 0 indicates no rating.  $u_a$  and  $u_b$  represent users, and  $I_k^P$  denotes the  $k^{th}$  item in the domain  $P$ .

Table 1 shows that the preferences of  $u_a$  and  $u_b$  in the music field are similar; the preferences are temporarily uncertain in the movie field, and the preferences are quite different in the book field.

We measure preference similarities  $\text{sim}(u_a, u_b)^{\text{COS}}$ ,  $\text{sim}(u_a, u_b)^{\text{PCC}}$ , and  $\text{sim}(u_a, u_b)^{\text{Jaccard}}$  of users  $u_a$  and  $u_b$  by using classical methods COS [28], PCC [28], and Jaccard [29], respectively. Then, we compare them with the similarity after preference domain segmentation. This comparison is shown in Table 2.

Table 2 shows that the preference similarity with preference domain segmentation can describe preference similarity among users more precisely than that without

**Table 1** Rating records of a website

Users	Items									
	Music				Movies			Books		
	$I_1^1$	$I_2^1$	$I_3^1$	$I_4^1$	$I_1^2$	$I_2^2$	$I_3^2$	$I_1^3$	$I_2^3$	
$u_a$	4	2	5	0	0	0	0	5	4	5
$u_b$	4	1	5	5	3	5	1	1	0	2

**Table 2** Comparison of preference similarity

Domains	Measures					
	Without preference domain segmentation			With preference domain segmentation		
	Values					
	$\text{sim}(u_a, u_b)^{\text{COS}}$	$\text{sim}(u_a, u_b)^{\text{Jaccard}}$	$\text{sim}(u_a, u_b)^{\text{PCC}}$	$\text{ssim}(u_a, u_b)^{\text{COS}}$	$\text{ssim}(u_a, u_b)^{\text{Jaccard}}$	$\text{ssim}(u_a, u_b)^{\text{PCC}}$
Music	0.53	0.50	0.29	0.78	0.75	0.97
Movies				--	--	--
Books				0.82	0.67	0.00

preference domain segmentation. Furthermore, PCC is the most accurate method. Moreover, from  $\text{ssim}(u_a, u_b)^{\text{PCC}}$ , we can predict that  $u_a$  may be interested in  $I_a^1$ . In this paper, we will employ  $\text{sim}(u_a, u_b)^{\text{PCC}}$  to calculate the preference similarity. The formula is as follows:

$$\text{ssim}(u_a, u_b)^{\text{PCC}} = \frac{\sum_{i \in I_{u_a}^P \cap I_{u_b}^P} (r_{u_a,i} - \bar{r}_{u_a})(r_{u_b,i} - \bar{r}_{u_b})}{\sqrt{\sum_{i \in I_{u_a}^P \cap I_{u_b}^P} (r_{u_a,i} - \bar{r}_{u_a})^2} \sqrt{\sum_{i \in I_{u_a}^P \cap I_{u_b}^P} (r_{u_b,i} - \bar{r}_{u_b})^2}}, \tag{1}$$

where  $I_{u_a}$  and  $I_{u_b}$  represent the rated item sets of  $u_a$  and  $u_b$ , respectively;  $r_{u_a,i}$  and  $r_{u_b,i}$  indicate the ratings of  $u_a$  and  $u_b$  for a specific item  $i$ , respectively; and  $\bar{r}_{u_a}$  and  $\bar{r}_{u_b}$  denote the average rating of  $u_a$  and  $u_b$ , respectively.

Considering the preference domain segmentation of users, we can measure a user's experience in a particular domain based on the number of items bought by the user. The formula is as follows:

$$\xi_u^P = \frac{|I_u|}{|I_{u_1} \cup I_{u_2} \cup \dots \cup I_{u_m}|}, \tag{2}$$

where  $I_u$  denotes the item set rated by the target user  $u$ ;  $|\cdot|$  indicates the element amount of the set; and  $u_1, u_2, \dots, u_m$  represent all users in the particular domain  $P$ .

### 3.2 Trust relationship segmentation

In Fig. 2, the user eigenmatrix  $U_u$  obeys the Gaussian distribution with the mean  $\mu = 0$  and the variance  $\sigma^2 = \sigma_U^2$ ; the item eigenmatrix  $V_i$  obeys the Gaussian distribution with the mean  $\mu = 0$  and the variance  $\sigma^2 = \sigma_V^2$ ; and the predicted rating  $\hat{R}_{ui}$  obeys the Gaussian distribution with the mean  $\mu = r_{ui}$  and the variance  $\sigma^2 = \sigma_R^2$ .  $S_{u_a, u_b}$  denotes the social link value, where  $S_{u_a, u_b} = 0$  indicates no social relationship between  $u_a$  and  $u_b$ .  $L_u$  represents the set of users who have a social link with the target user  $u$ , and  $w_1, w_2, \dots, w_n$  denote elements of  $L_u$ .  $N_u$  represents the set of users who do not have a social link with the target user  $u$ , and  $z_1, z_2, \dots, z_{n'}$  denote elements of  $N_u$ .  $t_{w,u}^P$  and  $t_{z,u}^P$  denote different recommenda-

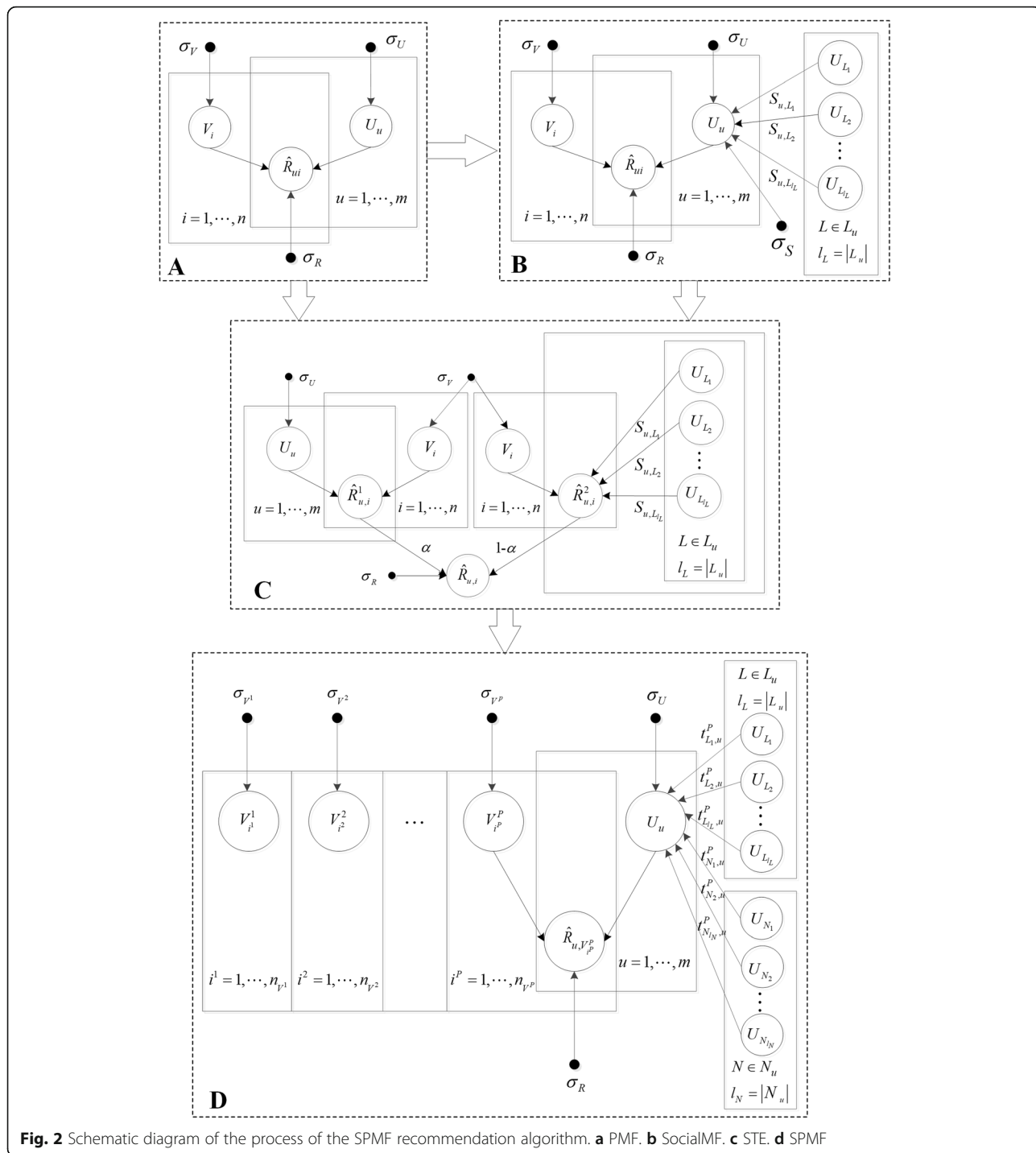
tion influences in specific domain  $P$ , which can be calculated by Eq. (3).

User relationships are complex, but in classical recommendation algorithms, we assume that user relationships are independent of each other in order to simplify the problem. Typically, we only consider the user's ratings on products, such as with the probability matrix factorization (PMF) recommendation algorithm [30], as shown in Fig. 2a. However, it is difficult for us to be completely independent when making sensible decisions, and we are often influenced by our friends and family. In view of this, some researchers introduced the trust relationship into the recommendation algorithm and only considered the user evaluation with a trust relationship to make recommendations for target users, i.e., the SocialMF recommendation algorithm [30]. This algorithm is shown in Fig. 2b. Social recommendation algorithms could improve recommendation accuracy but ignored the user's evaluation of the product. In order to overcome the shortcomings of the PMF recommendation algorithm and the trust-based recommendation algorithm and utilize their advantages, it is necessary to combine them. Doing so produces the STE recommendation algorithm [31], as shown in Fig. 2c. In order to adequately reflect the effect of the trust relationship and the preference similarity to target users, we construct a matrix factorization recommendation algorithm that combines trust relationship segmentation and preference domain segmentation, named SPME, which is shown in Fig. 2d.

In order to distinguish the different influences of  $L_u$  and  $N_u$  on  $u$ , we define the specific domain recommendation influence  $t_{u',u}^P$  as follows:

$$t_{u',u}^P = \begin{cases} \alpha \times \text{ssim}(u', u)^{\text{PCC}} \times \xi_{u'}^P, & u' \in L_u, \\ (1-\alpha) \times \text{ssim}(u', u)^{\text{PCC}} \times \xi_{u'}^P, & u' \in N_u, \end{cases} \tag{3}$$

where  $\alpha$  is an adjustment factor that is used to weigh the recommendation impact of  $L_u$  and  $N_u$  on the target user  $u$ ; and  $u'$  represents another user who belongs to either  $L_u$  or  $N_u$ .



**Fig. 2** Schematic diagram of the process of the SPMF recommendation algorithm. **a** PMF. **b** SocialMF. **c** STE. **d** SPMF

In the specific domain  $P$ , we can define the predicted target user eigenvector  $\hat{U}_u$  according to the eigenvectors of  $L_u$  and  $N_u$ :

$$\hat{U}_u = \sum_{w \in L_u} t_{w,u}^P U_w + \sum_{z \in N_u} t_{z,u}^P U_z \quad (4)$$

If we incorporate the specific domain recommendation influence into the PMF algorithm, the posterior probability distribution of the user eigenmatrix  $U$  and the item eigenmatrix  $V$  is as follows:

$$\begin{aligned} p(U, V | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) &\propto p(R | U, V, \sigma_R^2) p(U | T, \sigma_U^2, \sigma_T^2) p(V | \sigma_V^2) \\ &= \prod_{u=1}^m \prod_{i=1}^n [N(r_{ui} | U_u^T V_i, \sigma_R^2)]^{\delta_{ui}} \\ &\times \prod_{u=1}^m N\left(U_u \mid \sum_{w \in L_u} t_{w,u}^P U_w + \sum_{z \in N_u} t_{z,u}^P U_z, \sigma_U^2 I\right) \\ &\times \prod_{u=1}^m N(U_u | 0, \sigma_U^2 I) \times \prod_{i=1}^n N(V_i | 0, \sigma_V^2 I), \end{aligned} \quad (5)$$

where  $N(x | \mu, \sigma^2)$  denotes a Gaussian distribution with a mean  $\mu$  and a variance  $\sigma^2$ ;  $\delta_{ui}$  is a coefficient in which  $\delta_{ui} = 1$  means that  $u$  has rated  $i$ , while  $\delta_{ui} = 0$  represents no rating record; and  $I$  is a unit matrix of  $K$  dimensions.

The objective function of this algorithm is obtained by taking a negative logarithm of Eq. (5) as follows:

$$\begin{aligned} L(R, T, U, V) &= \frac{1}{2} \sum_{u=1}^m \sum_{i=1}^n I_{ui} (r_{ui} - U_u^T V_i)^2 \\ &+ \frac{\lambda_u}{2} \sum_{u=1}^m U_u^T U_u + \frac{\lambda_v}{2} \sum_{i=1}^n V_i^T V_i \\ &+ \frac{\lambda_t}{2} \sum_{u=1}^m \left( \left( U_u - \left( \sum_{w \in L_u} t_{w,u}^P U_w + \sum_{z \in N_u} t_{z,u}^P U_z \right) \right)^T \left( U_u - \left( \sum_{w \in L_u} t_{w,u}^P U_w + \sum_{z \in N_u} t_{z,u}^P U_z \right) \right) \right) + C, \end{aligned} \quad (6)$$

where  $\lambda_u = \frac{\sigma_R^2}{\sigma_U^2}$ ,  $\lambda_v = \frac{\sigma_R^2}{\sigma_V^2}$ ,  $\lambda_t = \frac{\sigma_R^2}{\sigma_T^2}$ , and  $C$  is constant.

In order to obtain the optimal value of the objective function (6), we employ the gradient descent algorithm to get the partial derivatives of the eigenvectors  $U_u$  and  $V_i$ :

**Table 3** Statistics of the experimental datasets

	Ciao	Epinions
Number of users ( $m$ )	7375	40,163
Number of items ( $n$ )	99,746	139,738
Number of rating records	280,391	664,824
Number of trust relationships	111,781	487,183
Sparsity of rating records	99.9619%	99.9882%
Sparsity of trust relationships	99.7945%	99.9698%

$$\begin{aligned} \frac{\partial L}{\partial U_u} &= \sum_{i=1}^n I_{ui} V_i (U_u^T V_i - r_{ui}) + \lambda_u U_u \\ &+ \lambda_t \left[ U_u - \left( \sum_{w \in L_u} t_{w,u}^P U_w + \sum_{z \in N_u} t_{z,u}^P U_z \right) \right] \\ &- \lambda_t \sum_{u \in L_w} t_{u,w}^P \left[ U_w - \left( \sum_{a \in L_w} t_{a,w}^P U_a + \sum_{b \in N_w} t_{b,w}^P U_b \right) \right] \\ &- \lambda_t \sum_{u \in N_z} t_{z,u}^P \left[ U_z - \left( \sum_{c \in L_z} t_{c,z}^P U_c + \sum_{d \in N_z} t_{d,z}^P U_d \right) \right], \end{aligned} \quad (7)$$

$$\frac{\partial L}{\partial V_i} = \sum_{u=1}^m I_{ui} U_u (U_u^T V_i - r_{ui}) + \lambda_v V_i \quad (8)$$

In this way, we can get the following eigenvectors by iteratively updating the gradient descent algorithm:

$$U_u^{(\tau+1)} = U_u^{(\tau)} + \gamma \left[ \begin{aligned} &\sum_{i=1}^n I_{ui} V_i (r_{ui} - U_u^{(\tau)T} V_i) - \lambda_u U_u - \lambda_t \\ &\times \left[ U_u - \left( \sum_{w \in L_u} t_{w,u}^P U_w + \sum_{z \in N_u} t_{z,u}^P U_z \right) \right] \\ &- \lambda_t \sum_{u \in L_w} t_{u,w}^P \left[ U_w - \left( \sum_{a \in L_w} t_{a,w}^P U_a + \sum_{b \in N_w} t_{b,w}^P U_b \right) \right] \\ &- \lambda_t \sum_{u \in N_z} t_{z,u}^P \left[ U_z - \left( \sum_{c \in L_z} t_{c,z}^P U_c + \sum_{d \in N_z} t_{d,z}^P U_d \right) \right] \end{aligned} \right], \quad (9)$$

$$V_i^{(\tau+1)} = V_i^{(\tau)} + \gamma \left[ \sum_{u=1}^m I_{ui} U_u (r_{ui} - U_u^T V_i) - \lambda_v V_i \right], \quad (10)$$

where  $\tau$  is the sequence number of the iterative updating, and  $\gamma$  is the learning rate.

## 4 Experimental section

### 4.1 Experimental datasets

In this section, two public datasets<sup>1</sup>, Ciao and Epinions, are employed. The performance of the proposed SPMF is evaluated by the mean absolute error (MAE) and the root mean square error (RMSE) [30, 31]. The statistics of datasets Ciao and Epinions are shown in Table 3. Obviously, both datasets are highly sparse.

<sup>1</sup><https://www.librec.net/datasets.html>



## 4.2 Algorithm implementations and complexity analysis

### 4.2.1 Generate a recommendation influence matrix of specific domains

For the SPMF recommendation algorithm, the recommendation influence matrix element of specific domains can be calculated by Eq. (3), and the algorithm generated by the recommendation influence matrix  $T^{m \times m}$  of specific domains is shown by Algorithm 1.

**Algorithm 1.** Generate a recommendation matrix  $T^{m \times m}$  of specific domains

<b>Input:</b> the rating matrix $R^{m \times n}$ of the user-item, the original trust relationship matrix $S^{m \times m}$ among users
<b>Output:</b> the recommendation influence matrix $T^{m \times m}$ of specific domains
<pre> 1  Procedure Rec_Influence(R, S, alpha); 2  Count = []; 3  for i in sum_users do 4  count_1 = 0; R_sum = 0; R_avg = []; 5  for j in sum_items do 6  if R[i][j]&gt;0 then 7  count_1 = count_1 + 1; Count.append(count_1) 8  R_sum = R_sum + R[i][j]; 9  end if 10 R_avg.append(count_1 / R[i]_sum); 11 end for 12 end for 13 c = [] 14 for u in sum_users do 15 count_2 = 0; 16 for v in sum_users do 17 S_c = 0; S_u = 0; S_v = 0; 18 if (S[u][v]=1); 19 alpha[u][v] = 0.4; 20 else: 21 alpha[u][v] = 0.6; 22 end if 23 for j in sum_items do 24 if ((R[u][j]&gt;0) or (R[v][j]&gt;0)) then 25 count_2 = count_2 + 1; 26 S_c = S_c + (R[u][j] - R_avg[u]) * (R[v][j] - R_avg[v]); 27 S_u = S_u + (R[u][j] - R_avg[u]) * (R[u][j] - R_avg[u]); 28 S_v = S_v + (R[v][j] - R_avg[v]) * (R[v][j] - R_avg[v]); 29 end if 30 end for 31 s[u][v] = S_c / ((sqrt(S_u)) * (sqrt(S_v))); 32 end for 33 e[u] = (Count[u] / float(count_2)); 34 t[u][v] = alpha[u][v] * s[u][v] * e[u]; 35 end for 36 end Procedure;</pre>

In Algorithm 1,  $\alpha$  is set to 0.4. We explain this in the section titled “Parameter analysis on  $\alpha$ .”

From the above process, the complexity of Algorithm 1 is  $O(n^3)$ .

### 4.2.2 The user-item rating matrix for prediction

For the SPMF recommendation algorithm, the algorithm of the user-item rating matrix for prediction is shown by Algorithm 2.

**Algorithm 2.** The user-item rating matrix for prediction

<b>Input:</b> the rating matrix $R^{m \times n}$ , the recommendation influence matrix $T^{m \times m}$ of specific domains
<b>Output:</b> the predicted rating matrix $\hat{R}^{m \times n}$
<pre> 1  for epo in epochs do 2  for i in sum_users do 3  for j in sum_items do 4  if R[i][j]&gt;0 then 5  eij=R[i][j]-np.dot(P[i,:],Q[:j]); 6  end if 7  for k in K do 8  for m in sum_users do 9  for n in sum_users do 10 P[i][k]=P[i][k]+lr*(eij*Q[k][j]-lambda_u_v*P[i][k]-lambda_l*(P[i][k]- 11 t[i][m]*P[m][k])+lambda_l*(T[i][m]*P[m][k]-[n][n]*P[n][k])); 12 Q[k][j]=Q[k][j]+lr*(eij*P[i][k]-lambda_u_v*Q[k][j]); 13 end for 14 end for 15 end for 16 end for 17 R_MF=np.dot(P,Q); 18 end for</pre>

In Algorithm 2,  $K$  represents matrix factorization dimensions.

From the above process, the complexity of Algorithm 2 is  $O(n^6)$ .

## 5 Results and discussion

### 5.1 Results

In this section, we compare the proposed SPMF with some state-of-the-art algorithms.

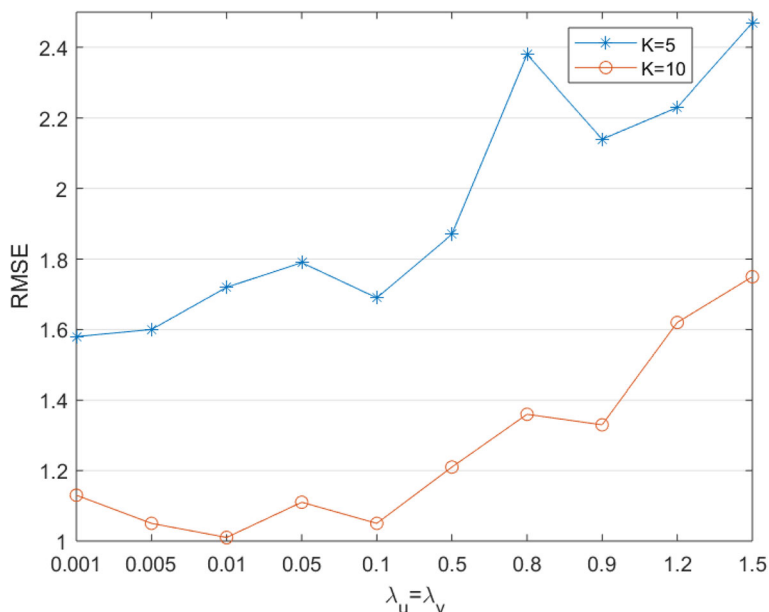
- PMF [32] is the probabilistic matrix factorization algorithm, which decomposes the original rating matrix into lower-rank latent feature matrices of user and item, then takes the matrices product as the predicted rating matrix.
- RSTE [31] is a probabilistic matrix factorization framework with social trust ensemble.
- SocialMF [30] is a model-based recommendation algorithm, which incorporates the crucial phenomenon of trust propagation into matrix factorization.
- TrustSVD [23] is a trust-based matrix factorization technique for recommendation.

For comparison, we use the same strategy as most of the literature: the matrix factorization dimensions are

**Table 4** RMSE comparison with baselines

Datasets	Dimension	Indicator	PMF	RSTE	SocialMF	TrustSVD	SPMF	Accuracy improvement
Epinions	$K = 5$	MAE	0.979	0.950	0.825	0.804	<i>0.794</i>	1.24%
		RMSE	1.290	1.196	1.070	1.043	<i>0.989</i>	5.18%
	$K = 10$	MAE	0.909	0.958	0.826	0.805	<i>0.762</i>	5.34%
		RMSE	1.197	1.278	1.082	1.044	<i>0.974</i>	6.70%
Ciao	$K = 5$	MAE	0.920	0.767	0.749	0.723	<i>0.571</i>	21.02%
		RMSE	1.260	1.020	0.981	0.955	<i>0.759</i>	20.52%
	$K = 10$	MAE	0.822	0.763	0.749	0.723	<i>0.565</i>	21.85%
		RMSE	1.078	1.013	0.976	0.956	<i>0.756</i>	20.92%

The italic values reflected inside Table 4 is the accuracy of the SPMF algorithm



**Fig. 3** RMSE vs.  $\lambda_u$  and  $\lambda_v$  on Ciao. The blue line with  $K = 5$ , and the green line with  $K = 10$

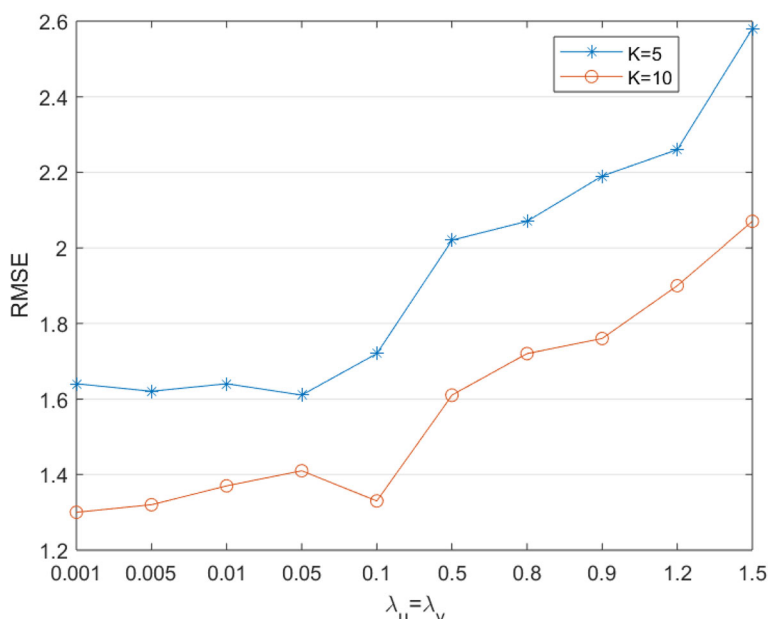
set to  $K = 5$  and 10. The MAE and RMSE of the proposed SPMF algorithm and other recommendation algorithms on Epinions and Ciao datasets are shown in Table 4.

Table 4 shows that the MAE and RMSE of the SPMF algorithm are less than those of algorithms PMF, RSTE, SocialMF, and TrustSVD in the cases of  $K = 5$  and 10, i.e., the accuracy of the SPMF algorithm is the highest among all compared algorithms. Moreover, from Table 4,

we can find that the SPMF has a significant accuracy improvement compared with some existing algorithms.

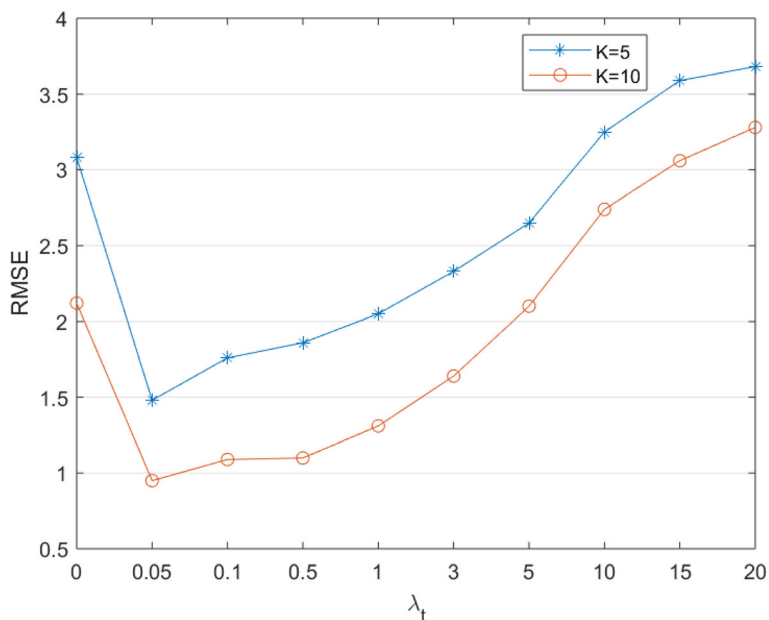
### 6 Discussion

In exploring the influence of various parameters on the performance of the algorithm, this paper adopts the idea of control variables. The more the epoch increases, the longer the training time becomes. Therefore, the epochs in this section are all set at 10.



**Fig. 4** RMSE vs.  $\lambda_u$  and  $\lambda_v$  on Epinions. The blue line with  $K = 5$ , and the green line with  $K = 10$





**Fig. 5** RMSE vs.  $\lambda_t$  on Ciao. The blue line with  $K = 5$ , and the green line with  $K = 10$

**6.1 Parameter analysis on  $\lambda_u$  and  $\lambda_v$**

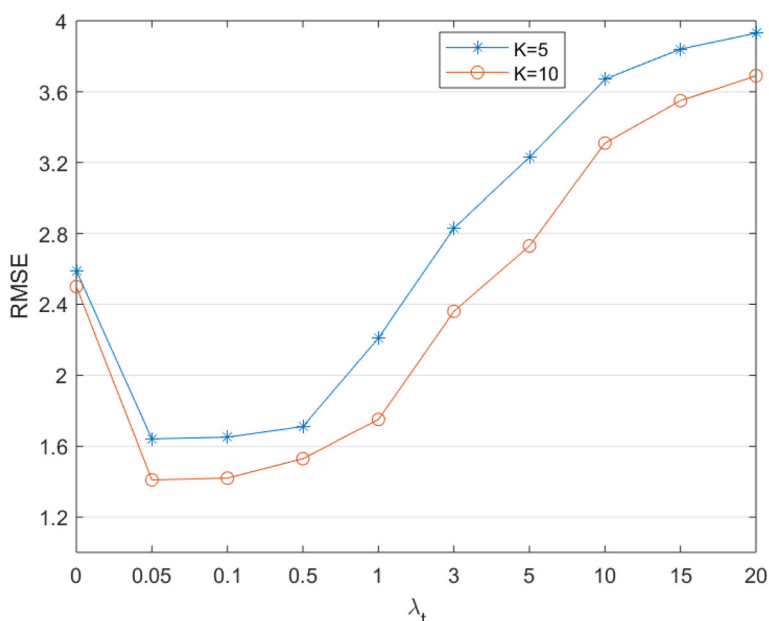
In this section, we explore the effect of  $\lambda_u$  and  $\lambda_v$  values on algorithm performance under different decomposition dimensions. The results are shown in Figs 3 and 4.

Figures 3 and 4 show that the optimal parameter values should be  $\lambda_u = \lambda_v = 0.005$  on both Ciao and Epinions since the average RMSE corresponding to different

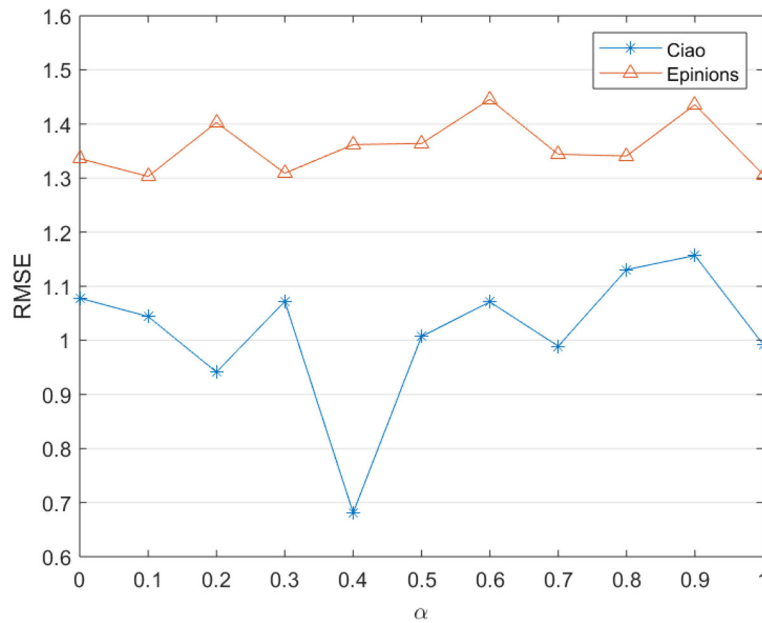
decomposition dimensions is the smallest only when  $\lambda_u = \lambda_v = 0.005$ .

**6.2 Parameter analysis on  $\lambda_t$**

In this section, we explore the effect of  $\lambda_t$  values on algorithm performance under different decomposition dimensions. The results on datasets Ciao and Epinions are



**Fig. 6** RMSE vs.  $\lambda_t$  on Epinions. The blue line with  $K = 5$ , and the green line with  $K = 10$



**Fig. 7** RMSE vs.  $\alpha$  on Ciao and Epinions. The blue line for Ciao, and the green line for Epinions

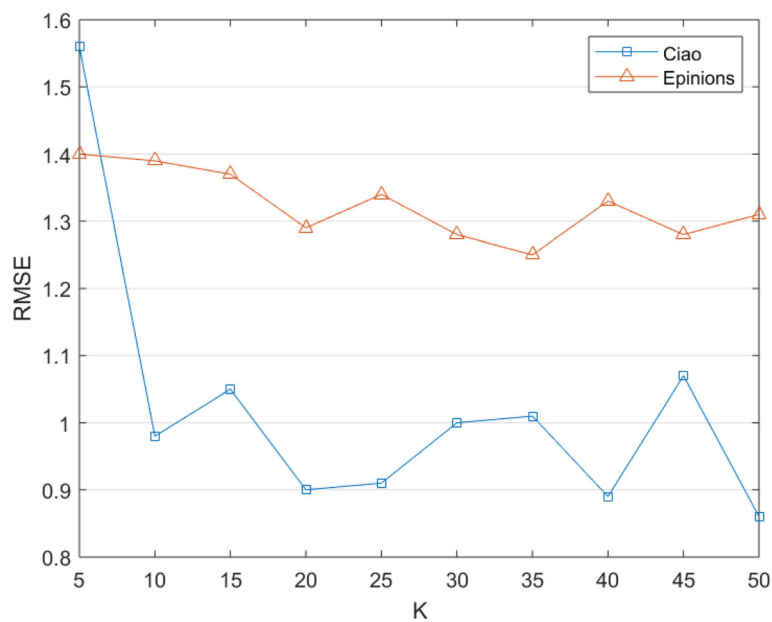
shown in Figs. 5 and 6. The different  $\lambda_t$  values are set as in the previous literature.

From Figs 5 and 6, we can conclude that in different decomposition dimensions of different datasets, the R.MSE value of the algorithm is the lowest when  $\lambda_t = 0.05$  on both datasets.

### 6.3 Parameter analysis on $\alpha$

In this section, we explore the optimal  $\alpha$  of the algorithm. The result is shown in Fig 7.

Figure 7 shows that the optimal parameter value should be  $\alpha = 0.4$  on datasets Ciao and Epinions. Because the sum of RMSEs on Ciao and Epinions is smallest



**Fig. 8** RMSE vs.  $K$  on Ciao and Epinions. The blue line for Ciao, and the green line for Epinions

when  $\alpha = 0.4$ , that is to say, the SPMF algorithm can fit different datasets better when  $\alpha = 0.4$  compared with other values of  $\alpha$ .

#### 6.4 Parameter analysis on $K$

In this section, we will explore the optimal decomposition dimension  $K$ . The result is shown in Fig. 8.

Figure 8 shows that the optimal factorization dimension of both datasets is  $K = 20$ . On Ciao, the RMSE of  $K = 20$  is not the smallest in this experiment, but too large of a decomposition dimension may cause overfitting, and the difference of RMSE between  $K = 20$  and  $K = 40$ ,  $K = 50$  is within a reasonable range, and. Similarly, we can conclude that  $K = 20$  is the optimal decomposition dimension of Epinions.

#### 7 Conclusions

The traditional social recommendation algorithm ignores the following fact: the preferences of users with trust relationships are not necessarily similar, and the consideration of user preference similarity should be limited to specific areas. To solve these problems, this paper proposed a SPMF recommendation algorithm by setting different recommendation trust weights for different relationships and preference domains. The experimental results on Ciao and Epinions datasets show that the accuracy of the SPMF algorithm is much higher than that of some state-of-the-art recommendation algorithms.

Because we take into consideration different recommendation trust weights for different relationships and preference domains, the limitation of SPMF is its high complexity of the algorithm. In the future, researchers can explore simpler algorithms by incorporating our ideas.

#### Abbreviations

SPMF: Social trust and preference segmentation-based matrix factorization; MAE: Mean absolute error; RMSE: Root mean square error

#### Acknowledgements

The authors would like to express sincere gratitude to the referees for their valuable suggestions and comments.

#### Authors' contributions

The authors have made the same contribution. All authors read and approved the final manuscript.

#### Funding

The work is supported by the National Social Science Foundation of China [No. 16FJY008], the National Natural Science Foundation of China [No. 11801060], and the Natural Science Foundation of Shandong Province [No. ZR2016FM26], and the innovation program of Shandong University of Science and Technology (No. SDKDYC190114).

#### Availability of data and materials

[1] Ciao dataset

<https://www.librec.net/datasets/CiaoDVD.zip>

[2] Epinions dataset

[http://www.trustlet.org/downloaded\\_epinions.html](http://www.trustlet.org/downloaded_epinions.html).

#### Competing interests

The authors declare that they have no competing interests.

Received: 4 April 2019 Accepted: 11 November 2019

Published online: 16 December 2019

#### References

- Z.P. Fan, Y. Xi, Y. Li, Supporting the purchase decisions of consumers: a comprehensive method for selecting desirable online products. *Kybernetes* **47**(4), 689–715 (2018). <https://doi.org/10.1108/K-03-2017-0116>
- L. Kuang et al., A Personalized QoS Prediction Approach for CPS Service Recommendation Based on Reputation and Location-Aware Collaborative Filtering. *Sensors* **18**(5), 1556 (2018). <https://doi.org/10.3390/s18051556>
- C.Wu, et al., Recommendation algorithm based on user score probability and project type. *EURASIP J. Wirel. Commun. Netw.*, 2019(1),80(2019). (doi: <https://doi.org/10.1186/s13638-019-1385-5>)
- P.C. Chang, C.H. Lin, M.H. Chen, A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms* **9**(3), 47 (2016). <https://doi.org/10.3390/a9030047>
- E. Khazaei, A. Alimohammadi, An automatic user grouping model for a group recommender system in location-based social networks. *ISPRS Int. J. Geo Inf.* **7**(2), 67 (2018). <https://doi.org/10.3390/ijgi7020067>
- K. Haruna et al., Context-aware recommender system: A review of recent developmental process and future research direction. *Appl. Sci.* **7**(12), 1211 (2017). <https://doi.org/10.3390/app7121211>
- B. Albanna et al., Interest aware location-based recommender system using geo-tagged social media. *ISPRS Int. J. Geo Inf.* **5**(12), 245 (2016). <https://doi.org/10.3390/ijgi5120245>
- L. Cui et al., A novel context-aware recommendation algorithm with two-level SVD in social networks. *Futur. Gener. Comput. Syst.* **86**, 1459–1470 (2018). <https://doi.org/10.1016/j.future.2017.07.017>
- N. Iltaf, A. Ghafoor, M. Hussain, Modeling interaction using trust and recommendation in ubiquitous computing environment. *EURASIP J. Wirel. Commun. Netw.*, 2012(1),119(2012). (doi: <https://doi.org/10.1186/1687-1499-2012-119>)
- X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques. *Advances in artificial intelligence* **2009**, 421425 (2009). <https://doi.org/10.1155/2009/421425>
- M.J. Pazzani, D. Billsus, Content-based recommendation systems, in *The adaptive web*. Springer, 325–341 (2007). [https://doi.org/10.1007/978-3-540-72079-9\\_10](https://doi.org/10.1007/978-3-540-72079-9_10)
- J.L. Herlocker, J.A. Konstan, J. Riedl, Explaining collaborative filtering recommendations. in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 2000. ACM. <https://doi.org/10.1145/358916.358995>
- R. Burke, Hybrid recommender systems: Survey and experiments. *User Model. User-Adap. Inter.* **12**(4), 331–370 (2002). <https://doi.org/10.1023/A:1021240730564>
- J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering. in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. 1998. Morgan Kaufmann Publishers Inc.. <https://doi.org/10.1109/ICCV.2009.5459155>
- S. Kang, Outgoing call recommendation using neural network. *Soft. Comput.* **22**(5), 1569–1576 (2018). <https://doi.org/10.1007/s00500-017-2946-3>
- L. Ren, W. Wang, An SVM-based collaborative filtering approach for Top-N web services recommendation. *Futur. Gener. Comput. Syst.* **78**, 531–543 (2018). <https://doi.org/10.1016/j.future.2017.07.027>
- J. Zhao, G. Sun, Detect user's rating characteristics by separate scores for matrix factorization technique. *Symmetry* **10**(11), 616 (2018). <https://doi.org/10.3390/sym10110616>
- H. Ma et al., Sorec: Social recommendation using probabilistic matrix factorization. in *Proceedings of the 17th ACM conference on Information and knowledge management*. 2008. ACM. <https://doi.org/10.1145/1458082.1458205>
- C. Mi, P. Peng, R. Mierzwiak, A recommendation algorithm considering user trust and interest. in *Lecture Notes in Computer Science, 17th International Conference on Artificial Intelligence and Soft Computing*, 2018, 408–422. IEEE. (doi: [https://doi.org/10.1007/978-3-319-91262-2\\_37](https://doi.org/10.1007/978-3-319-91262-2_37))
- H. Han et al., An extended-tag-induced matrix factorization technique for recommender systems. *Information* **9**(6), 143 (2018). <https://doi.org/10.3390/info9060143>

21. J. Wang, A.P. De Vries, M.J. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. 2006. ACM. (doi: <https://doi.org/10.1145/1148170.1148257>)
22. C.H. Lai, S.J. Lee, H.L. Huang, A social recommendation method based on the integration of social relationship and product popularity. *International Journal of Human-Computer Studies* **121**, 42–57 (2019). <https://doi.org/10.1016/j.ijhcs.2018.04.002>
23. G. Guo, J. Zhang, N. Yorke-Smith, A novel recommendation model regularized with user trust and item ratings. *IEEE Trans. Knowl. Data Eng.* **28**(7), 1607–1620 (2016). <https://doi.org/10.1109/TKDE.2016.2528249>
24. W.P. Lee, C.Y. Ma, Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks. *Knowl.-Based Syst.* **106**, 125–134 (2016). <https://doi.org/10.1016/j.knosys.2016.05.037>
25. L. Qi et al., *A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment*. *Futur. Gener. Comput. Syst.* **88**, 636–643 (2018). <https://doi.org/10.1016/j.future.2018.02.050>
26. Gong, W., L. Qi, and Y. Xu, Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. *Wireless Communications and Mobile Computing*, 2018, 3075849(2018). (doi: <https://doi.org/10.1155/2018/3075849>)
27. H. Liu, H. Kou, C. Yan, L. Qi, Link prediction in Paper Citation Network to Construct Paper Correlated Graph. *EURASIP Journal on Wireless Communications and Networking*, 2019. (In press).
28. A. Salah, N. Rogovschi, M. Nadif, *A dynamic collaborative filtering system via a weighted clustering approach*. *Neurocomputing*, 175, 206–215(2016). <https://doi.org/10.1016/j.neucom.2015.10.050>
29. H. Liu et al., A new user similarity model to improve the accuracy of collaborative filtering. *Knowl.-Based Syst.* **56**, 156–166 (2014). <https://doi.org/10.1016/j.knosys.2013.11.006>
30. M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks. in Proceedings of the fourth ACM conference on Recommender systems. 2010. ACM. (doi: <https://doi.org/10.1145/1864708.1864736>)
31. H. Ma, I. King, M.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. 2009. ACM. (doi: <https://doi.org/10.1145/1571941.1571978>)
32. R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization. in *Advances in Neural Information Processing Systems 20 (NIPS 2007)* . 2008: 1257-1264. (<http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.pdf>)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---