IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 1, 2019, accepted July 9, 2019, date of publication July 15, 2019, date of current version July 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928574

Personalized Recommendation via Trust-Based Diffusion

YUANZHEN LIU[1,2], **LIXIN HAN**[2], **ZHINAN GOU**[2], **AND YI YANG**[2,3]
[1]Yancheng Institute of Technology, Yancheng 224051, China
[2]College of Computer and Information, Hohai University, Nanjing 210098, China
[3]College of Computer Science and Technology, Huaibei Normal University, Huaibei 230601, China

Corresponding author: Lixin Han (lixinhan2002@aliyun.com)

This work was supported in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province of China under Grant KYCX17_0486, in part by the Fundamental Research Funds for the Central Universities under Grant 2017B708X14, and in part by the Natural Science Foundation of the Colleges and Universities in Anhui Province of China under Grant KJ2017B016.

ABSTRACT The diffusion-based algorithm is a promising member of the family of recommendation algorithms. It makes recommendations through the diffusion process on user-object bipartite graphs. However, a user's taste is often influenced by his/her trusted friends in social networks. In this paper, we propose a new trust-based diffusion on tripartite graphs, which integrates explicit trust relations and implicit trust relations into the diffusion process. Explicit trust relations are obtained from the social networks while implicit trust relations are inferred from implicit feedback. The experimental results indicate that our proposed method has a remarkable improvement in accuracy, and even only implicit trust relations employed, that is, diffusion on user-object bipartite graphs, the recommendation accuracy is still enhanced. We further present a general framework of applying implicit trust relations and explicit trust relations into basic network-based diffusion method, which is more general and flexible, and specific parameters can be selected to meet actual requirements for different datasets and real-world online platforms.

INDEX TERMS Trust-based diffusion, tripartite graphs, social networks, implicit trust relations, explicit trust relations.

I. INTRODUCTION

It becomes extremely hard to gain what people want from tons of data in modern society. Information overload has become a serious problem. To solve the problem, recommender systems have been widely employed, which provide an effective way to filter information. The task of recommender systems is to predict users' preferences based on past historical behaviors, attributes of objects, spatio-temporal context, and so on [1]. Recommender systems are applied in a variety of situations such as e-commerce platforms, news websites, social network service, research article websites, location-related service, which benefit both the service providers and the users [2]–[5].

To meet different demands, various recommendation algorithms have been proposed and one of the most successful is collaborative filtering [6]. Recently, some scholars model users and objects in recommender systems as user-object bipartite graphs, and inspired by physics dynamics on bipartite graphs, the mass diffusion [7] and heat conduction [8] methods are introduced to make recommendations for users. As these network-based physical approaches have been demonstrated to be efficient [7], [9], [10], they attract the interest of scholars in recommender systems, and many works have been done and published [11]–[14].

Nowadays, social networks have a huge impact on people's lives and provide new information to further improve the performance of recommender systems [15]–[18]. Peer influence and the effect of social relations on the adoption of products and services have been modeled by the threshold model [19], [20] and the bootstrap percolation model [21]. Social network information as additional input for recommender systems has been studied and many algorithms have been proposed, such as matrix factorization-based social recommendation [22]–[24] and probabilistic model-based social recommendation [25].

Many diffusion-based recommendation algorithms are based on user-object bipartite graphs, but only a few studies focus on heterogeneous networks, such as user-object-tag and user-user-object tripartite graphs. Zhang *et al.* [26] studied

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

diffusion on user-object-tag tripartite graphs by carrying out standard two-step turn-around mass diffusion on the user-object bipartite graph and object-tag bipartite graph, respectively, and the resource received by the object is linearly superposed as the final resource. Shang *et al.* [27] studied collaborative filtering with diffusion-based similarity on user-item-tag tripartite graphs, in which the similarity is obtained from standard mass diffusion on the user-object bipartite graph and the user-tag bipartite graph, not the object-tag graph.

For research on user-user-object tripartite graphs, that is employing social networks, there are also some published works studying trust-based diffusion or employing social networks for the diffusion-based recommendation. Chen and Gao [28] proposed CosRA+T algorithm by introducing the trust relations among users into the original CosRA method [29], and if a user is trusted by the target user, the resource of the user is scaled by a tunable parameter in the second step diffusion, which is still diffusion on bipartite graphs. Deng *et al.* [30] proposed a recommendation algorithm by combining two-step turn-around diffusion on the user-object bipartite graph and on the user-user bipartite graph, respectively. Wang *et al.* [31] studied diffusion-based recommendation with trust relations and presented a new distribution process by combining CosRA-index [29] and both implicit and explicit trust relations. However, studies on trust-based diffusion methods on tripartite graphs remain insufficiently. Trust relations are various and have weight, and trust weight based on the similar taste of objects may be more suitable for making appropriate recommendations.

In this paper, we first propose a personalized recommendation method via trust-based mass diffusion on tripartite graphs (TrustMD for short), by combining implicit trust relations into the mass diffusion process (MD) on the user-object bipartite graph and making additional diffusion process on the user-user-object tripartite graph based on the perspective that when a user trusts his/her friends, he/she tends to trust what the friends collected. Trust weight is considered to scale how much he/she trusts his/her friends and computed by the degree of interaction on collected objects. We analyze the impact of implicit trust relations and explicit trust relations, respectively, and find that even only implicit trust relations is applied into the diffusion process on user-object bipartite graphs, the accuracy of recommendation is still enhanced. Further, we propose a more general framework of trust-based diffusion (TrustDiffusion for short), in which the basic diffusion method can be standard mass diffusion (MD), heat conduction (HC), CosRA or others, determined by different datasets and different requirements.

The remainder of this paper is organized as follows. Section 2 presents a detailed description of the proposed method TrustMD. Section 3 gives a brief survey of experiments. Section 4 provides experimental results and discussion and a general framework TrustDiffusion. We conclude in Section 5.

II. PROPOSED METHOD

Most diffusion-based recommendations work on user-object bipartite graphs without considering the social relations of the target user. In real life, people often ask friends for advice or they share common interests in some domains. We propose a personalized recommendation method via trust-based mass diffusion, named Trust Mass Diffusion (TrustMD for short), by integrating trust relations into the diffusion process. When adopting social networks for recommendation, trust weight is taken into account in TrustMD.

A trust-based recommender system can be described by a user-user-object tripartite graph, or a user-object bipartite graph combined with a user-user bipartite graph. It consists of a set of users $U = \{u_1, u_2, \ldots, u_m\}$, a set of objects $O = \{o_1, o_2, \ldots, o_n\}$, a link set between users and objects and a link set between users and users. We use i, j for the subscripts of users and α, β for objects. The user-object bipartite graph can be represented by an $m \times n$ adjacency matrix $\boldsymbol{A}$, where $a_{i\alpha} = 1$ if user u_i hasj collected object o_α and $a_{i\alpha} = 0$ otherwise, that is

$$a_{i\alpha} = \begin{cases} 1 & u_i \ has\ collected\ o_\alpha \\ 0 & otherwise \end{cases} \tag{1}$$

The degree of user u_i, denoted by $k(u_i)$, is the number of objects that user u_i has collected and $k(u_i) = \sum_{\alpha=1}^{n} a_{i\alpha}$. Similarly, the degree of object o_α, denoted by $k(o_\alpha)$, is the number of users who have collected object o_α and $k(o_\alpha) = \sum_{i=1}^{m} a_{i\alpha}$.

The user-user bipartite graph is characterized by an $m \times m$ adjacency matrix $\boldsymbol{B}$, where $b_{ij} = 1$ if user u_i trusts u_j and $b_{ij} = 0$ otherwise, that is

$$b_{ij} = \begin{cases} 1 & u_i \ trusts\ u_j \\ 0 & otherwise \end{cases} \tag{2}$$

In public published social rating datasets, they usually give out the trust relations whether u_i trusts u_j and whether the trust relations are directed or undirected, but there is no trust weight. Trust weight w_{ij} can be interpreted as how much u_i trusts u_j. We infer w_{ij} from implicit feedback. As w_{ij} is employed for recommendation, trust weight based on the similar taste of objects is a good choice. So we define trust weight w_{ij} as the degree of interaction on collected objects, also regarded as similarity of u_i and u_j, defined as

$$w_{ij} = \frac{1}{\sqrt{k(u_i)\,k(u_j)}} \sum_{\alpha=1}^{n} a_{i\alpha} a_{j\alpha} \tag{3}$$

The trust-based mass diffusion algorithm (TrustMD) contains two parts of diffusion processes, and the initial state is in Figure 1 Plot (a). One part of the resource is distributed to objects by two-step turn-around diffusion on the user-object bipartite graph, as shown in Figure 1 Plot (b), while the other part of resource spreads from the social network to the objects that the friends have collected, which is a two-step diffusion on the user-user-object tripartite graph, that is, from users

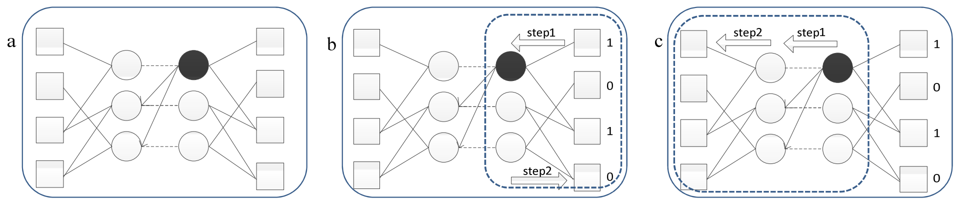


FIGURE 1. Illustration of the trust-based mass diffusion algorithm (TrustMD). Users and objects are denoted by circles and squares, respectively. The lines between users and objects represent collection relations and the arrowhead lines between users represent directed trust relations. The black circle means the target user. Plot (a) is the initial state. Plot (b) represents the diffusion on the user-object bipartite graph. Plot (c) shows the diffusion from the social network to objects.

to friends, then to objects, as shown in Figure 1 Plot (c). In other words, if some object o_β has relation to the objects that user u_i has collected and o_β also has been collected by the friend that u_i trusts, then o_β has more possibility to be recommended to u_i.

One part of resource diffuses on the user-object bipartite graph, as shown in the dotted box of Figure 1 Plot (b). We utilize the improved mass diffusion process, which works as follows.

Step 1: For the target user u_i, assign one unit initial resource to all the objects that u_i collected and assign zero to all the uncollected objects. That is, the initial resource on object o_α is set as

$$f_{i\alpha} = a_{i\alpha} \tag{4}$$

Step 2: Each object distributes its resource to all neighboring users, and then each user redistributes the received resource to all his/her collected objects. Denote the final resource vector on all objects by $\overrightarrow{f'}$, the initial resource vector by $\vec{f}$, the diffusion is described by equation $\overrightarrow{f'} = S\vec{f}$, where S is the resource transfer matrix, and $S = \{s_{\beta\alpha}\}$. For standard mass diffusion process (MD) [7], proposed by Zhou *et al.* the transfer matrix element $s_{\beta\alpha}$ is given by

$$s_{\beta\alpha}^{MD} = (\frac{1}{k(o_\alpha)}\sum\nolimits_{j=1}^{m}\frac{a_{j\beta}a_{j\alpha}}{k(u_j)}) \tag{5}$$

For an arbitrary object o_β, the final resource $f_{i\beta}^{\prime MD}$ received is

$$f'^{MD}_{i\beta} = \sum\nolimits_{\alpha=1}^{n} s_{\beta\alpha}^{MD} \times f_{i\alpha} = \sum\nolimits_{\alpha=1}^{n} (\frac{1}{k(o_\alpha)}\sum\nolimits_{j=1}^{m}\frac{a_{j\beta}a_{j\alpha}}{k(u_j)})f_{i\alpha} \tag{6}$$

In order to consider the implicit trust between users and the trust weight in the diffusion process, we integrate trust weight into Eq.(5) and get the improved $s_{\beta\alpha}^{TrustMD}$ as

$$s_{\beta\alpha}^{TrustMD} = (\frac{1}{k(o_\alpha)}\sum\nolimits_{j=1}^{m}\frac{a_{j\beta}a_{j\alpha}}{\sqrt{k(u_i)k(u_j)}}) \tag{7}$$

For an arbitrary object o_β, the final resource received from the user-object network can be written as

$$\begin{aligned} f_{i\beta}^{\prime TrustMD} &= \sum\nolimits_{\alpha=1}^{n} s_{\beta\alpha} f_{i\alpha} \\ &= \sum\nolimits_{\alpha=1}^{n} (\frac{1}{k(o_\alpha)}\sum\nolimits_{j=1}^{m}\frac{a_{j\beta}a_{j\alpha}}{\sqrt{k(u_i)k(u_j)}})f_{i\alpha} \end{aligned} \tag{8}$$

The other part of resource diffuses on the user-user-object tripartite graph, that is, from users to friends, then to objects, as shown in the dotted box of Figure 1 Plot (c). The diffusion process works as follows.

Step 1: For the target user u_i, assign initial resource to all the users that u_i trusts according to the equation

$$g_{ij}^{Trust} = b_{ij} \times w_{ij} = \frac{b_{ij}\sum_{\alpha=1}^{n} a_{i\alpha}a_{j\alpha}}{\sqrt{k\left(u_i\right)k\left(u_j\right)}} \tag{9}$$

where w_{ij} is the trust weight of user u_i and u_j computed by Eq.(3). As w_{ij} is the degree of interaction of u_i and u_j on collected objects, the higher of w_{ij}, the more common taste they share in collecting objects. So for diffusion on the user-user-object graph, more resource is distributed to the more trusted friends.

Step 2: As all the users that user u_i trusts are also elements in the user-object network, so each user that u_i trusts redistributes the received initial resource to all his/her collected objects averagely. After the one step resource-allocation process, for an arbitrary object o_β, the final resource $f_{i\beta}^{\prime Trust}$ received from explicit trust relations is

$$\begin{aligned} f_{i\beta}^{\prime Trust} &= \sum\nolimits_{j=1}^{m} g_{ij}^{Trust} \times (\frac{a_{j\beta}f_{j\beta}}{k\left(u_j\right)}) \\ &= \sum\nolimits_{j=1}^{m} \frac{b_{ij}a_{j\beta}f_{j\beta}\sum_{\alpha=1}^{n} a_{i\alpha}a_{j\alpha}}{k(u_j)\sqrt{k\left(u_i\right)k\left(u_j\right)}} \end{aligned} \tag{10}$$

After the two parts of diffusion processes, we linearly combine the resource arrived at the object o_β and the final resource $f_{i\beta}^{\prime\prime TrustMD}$ of o_β is defined as

$$\begin{aligned} f_{i\beta}^{\prime\prime TrustMD} &= (1-\lambda) f_{i\beta}^{\prime TrustMD} + \lambda f_{i\beta}^{\prime Trust} \\ &= (1-\lambda)\sum\nolimits_{\alpha=1}^{n} (\frac{1}{k(o_\alpha)}\sum\nolimits_{j=1}^{m}\frac{a_{j\beta}a_{j\alpha}}{\sqrt{k(u_i)k(u_j)}})f_{i\alpha} \\ &\quad +\lambda\sum\nolimits_{j=1}^{m}\frac{b_{ij}a_{j\beta}f_{j\beta}\sum_{\alpha=1}^{n} a_{i\alpha}a_{j\alpha}}{k(u_j)\sqrt{k\left(u_i\right)k\left(u_j\right)}} \end{aligned} \tag{11}$$

where $\lambda \in [0, 1]$. λ is a free parameter to adjust the influence of explicit trust relations. When $\lambda = 0$, trust-based mass diffusion algorithm reduces to Eq.(8).

In the end, all the uncollected objects are sorted in descending order of their final resource $f^{\prime\prime TruustMD}$ and then a top-L objects list is recommended to the target user.

TABLE 1. Statistical properties of the two datasets.

Dataset	Users	Items	Ratings	Scale	R-Sparsity	User-links	U-Sparsity
Epinions	2600	33536	82044	[1,5]	9.41×10^{-4}	7428	1.10×10^{-3}
Ciao	4770	15103	56943	[1,5]	7.90×10^{-4}	12883	5.66×10^{-4}

In TrustMD, a linear superposition is also employed to combine the resource from two parts of diffusion processes. However, we don't simply divide the tripartite graph into two bipartite graphs and carry out respective turn-around diffusion process on the bipartite graphs. Instead, we realized diffusion on tripartite graphs by carrying out two-step turn-around diffusion on the user-object bipartite graph and two-step diffusion from users to friends and then to objects on the user-user-object tripartite graph. Implicit trust relations are integrated into the traditional two-step turn-around diffusion on the user-object bipartite graph. And we make the two-step not turn-around diffusion process on tripartite graph from a new perspective and intuitively mimic the view that if a user trusts his/her friends, he/she tends to trust more what the friends collected, and utilize trust weight to scale the degree of the trust or how much they share common interests, which is more reasonable and explicable.

We not only propose a method called TrustMD to integrated explicit trust relations and implicit trust relations into the basic mass diffusion process (MD) on tripartite graphs, but also extend the idea to a general form named TrustDiffusion in *Results and Discussion Section*, which means the basic diffusion method can be MD, HC, CosRA or others, as when we carry out experiments on different datasets of different data-sparsity, the datasets show different preferences on the basic diffusion method, and the detailed description is in *Results and Discussion Section*.

III. EXPERIMENTS

A. DATASETS

Two benchmark datasets are used to evaluate the performance of our proposed algorithm. (i) Epinions [32]: a subset of the dataset collected by Paolo Massa's team from the Epinions.com, a website where people can review products. It contains the ratings given to items and the trust statements issued by users. (ii) Ciao [33]: a subset of dataset crawled from the entire category of DVDs from the dvd.ciao.co.uk website. Ciao contains movie ratings, review-ratings, and trust statements, but we only utilize movie ratings and trust statements. Both the two datasets use a 5-star rating scale and the trust statements are directed. We delete the trust links if his/her friends have no rating records. Statistical properties of the two datasets are presented in Table1.

As some research papers [34]–[36] have proved that when the dataset is very sparse, ratings lower than the median still have positive effect on recommendation, we didn't discard ratings lower than the median (i.e. ratings <3), and all ratings (from 1 to 5) exist in our randomly selected subsets. When building the user-object matrix, $a_{i\alpha}$ is set to 1 if u_i has collected o_α regardless of the rating value. For the user-user matrix, b_{ij} is set to 1 if user u_i trusts u_j. As trust statements in Ciao and Epinions are directed, b_{ij} is not necessarily equal to b_{ij}.

A 5-fold cross-validation is used in an independent realization. Specifically, each dataset is randomly divided into five folds. Four folds are regarded as the training set (denoted by E^T), while the remaining fold as the probe set (denoted by E^P). After iterating 5 times, the 5 results are averaged to obtain the result of this independent realization. We performed 10 independent realizations to get the averaged results presented in this paper.

B. EVALUATION METRICS

To fully characterize the performance of the recommendation algorithm, six metrics are employed to evaluate the accuracy, diversity, and novelty of our proposed method.

1) ACCURACY

Accuracy is the most important index for evaluating recommendation algorithms. We employ precision, recall, and F1-measure for the quantification of recommendation accuracy, as these metrics are widely used and easy to calculate.

1.1) Precision [1]: the ratio of the number of recommended objects that were collected in the probe set to the length of a recommendation list L. The average precision is defined as

$$P(L) = \frac{1}{m}\sum\nolimits_{i=1}^{m}\frac{T_i(L)}{L} \tag{12}$$

where $T_i(L)$ is the number of corrected recommended objects for u_i in the probe set.

1.2) Recall [29]: it measures the proportion of the number of corrected recommended objects for u_i and the number of objects collected by u_i in the probe set. The average recall is defined as

$$R(L) = \frac{1}{m}\sum\nolimits_{i=1}^{m}\frac{T_i(L)}{\left|E_i^P\right|} \tag{13}$$

where $\left|E_i^P\right|$ is the number of objects collected by u_i in the probe set.

1.3) F1-measure [37]: it's a balanced evaluation of precision and recall, defined as

$$F1\,(L) = \frac{2P(L)R(L)}{P\,(L) + R(L)} \tag{14}$$

2) DIVERSITY

Diversity describes the dissimilarity of objects in recommendation lists. There are two types of diversity: Hamming distance and intra-similarity.

2.1) Hamming distance [38]: it measures how users' recommendation lists are different from each other. Given two users u_i and u_j, the Hamming distance is calculated as

$$H_{ij}(L) = 1 - \frac{C_{ij}(L)}{L} \tag{15}$$

where $C_{ij}(L)$ is the number of common objects in the top-L recommendation list of u_i and u_j. The average Hamming distance is defined as

$$H(L) = \frac{1}{m(m-1)} \sum\nolimits_{i \neq j} H_{ij}(L) \tag{16}$$

The higher the value of $H(L)$, the greater the dissimilarity of users' recommendation list and the more personalized of the recommendation.

2.2) Intra-similarity [38]: this metric measures the degree of difference between the objects in one user's recommendation list. For a target user u_i, whose top-L recommendation list is $L_i = \{o_1, o_2, \ldots, o_L, \}$, intra-similarity $I_i(L)$ is defined as

$$I_i(L) = \frac{1}{L(L-1)} \sum\nolimits_{o_\alpha \neq o_\beta}^{L_i} d_{\alpha\beta} \tag{17}$$

where $d_{\alpha\beta}$ is the cosine similarity between objects o_α and o_β and is defined as

$$d_{\alpha\beta} = \frac{1}{\sqrt{k(o_\alpha)\,k(o_\alpha)}} \sum\nolimits_{i=1}^{m} a_{i\alpha} a_{i\beta} \tag{18}$$

The intra-similarity of the whole system is defined as

$$I(L) = \frac{1}{m} \sum\nolimits_{i=1}^{m} I_i(L) \tag{19}$$

The lower $I(L)$, the more the recommended object differs from each other.

3) NOVELTY

Novelty [31] measures the capacity of recommender systems to recommend unpopular objects. Novelty is defined by the average degree of recommended objects, written as

$$= \frac{1}{m} \sum\nolimits_{i=1}^{m} \left(\frac{1}{L} \sum\nolimits_{o_\alpha}^{L_i} k(o_\alpha) \right) \tag{20}$$

where L_i is the top-L recommendation list of u_i and $k(o_\alpha)$ is the degree of object o_α. The lower value of novelty means that more unpopular objects are recommended.

C. BENCHMARK METHODS

The proposed algorithm is compared with four classic methods to verify its superiority.

1) MASS DIFFUSION (MD)

MD [7] is a classical diffusion-based recommendation algorithm on user-object bipartite graphs. For the target user, all he/she collected objects are assigned with one unit resource, and in the two-step diffusion process, the objects and users averagely distribute the resource to their neighbors based on their degree.

2) HEAT CONDUCTION (HC)

The initial step of HC [8] is the same as MD, but the resource is regarded as temperature. The user node's temperature is the average of all he/she collected objects' temperature and the object's temperature is the average of the neighbor users' temperature. Heat conduction performs well in the recommendation diversity.

3) COSRA

CosRA [29] combines both the cosine index and the resource-allocation(RA) index and finds a balance among accuracy and diversity.

4) CosRA + T

CosRA +T [28] is a recently proposed trust-based recommendation method, by introducing the trust relations among users into the resource-allocation processes of the original CosRA method, and it has a remarkable improvement in overall accuracy, diversity, and novelty.

IV. RESULTS AND DISCUSSION

A. THE IMPACT OF PARAMETER λ

Our algorithm employs the parameter λ to adjust the proportion of resource from the user-object network with implicit trust relations and from the social network with explicit trust relations. As F1 is a comprehensive metric of precision and recall, the optimal valued of λ is determined based on F1 in our algorithm. We calculate the F1 when the parameter λ varies from 0 to 0.9 at the step of 0.05 for datasets Epinions and Ciao, and the results are shown in Figure 2 with the length of recommendation list $L = 10$.

As we can see from Figure 2, when $\lambda = 0$, the resource from explicit trust relations reduces to 0. For Epinions, when $0 < \lambda \leq 0.9$, F1 is greater than when $\lambda = 0$, that is, the explicit trust relations have a positive effect, and while $\lambda = 0.75$, F1 gets the maximum value 0.02364. For Ciao, only when the parameter λ varies from 0.05 to 0.5, the explicit trust relations have a slight positive effect on F1, and while $\lambda = 0.1$, F1 gets its maximum value 0.03303. For both Epinions and Ciao, as λ gets close to 1, F1 drops sharply, which means recommendation cannot be made entirely based on social networks.

B. PERFORMANCE OF RECOMMENDATION

To verify the performance of proposed TrustMD, we compare the F1 metric with the four benchmark methods as the length of recommendation list L varies from 10 to 50 with step 10,

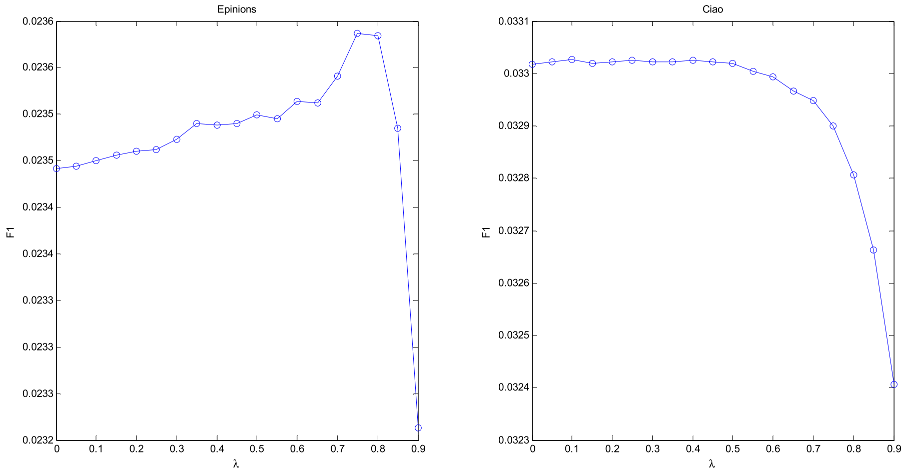


FIGURE 2. The F1 results of TrustMD as parameter λ varies from 0 to 0.9 with $L = 10$.

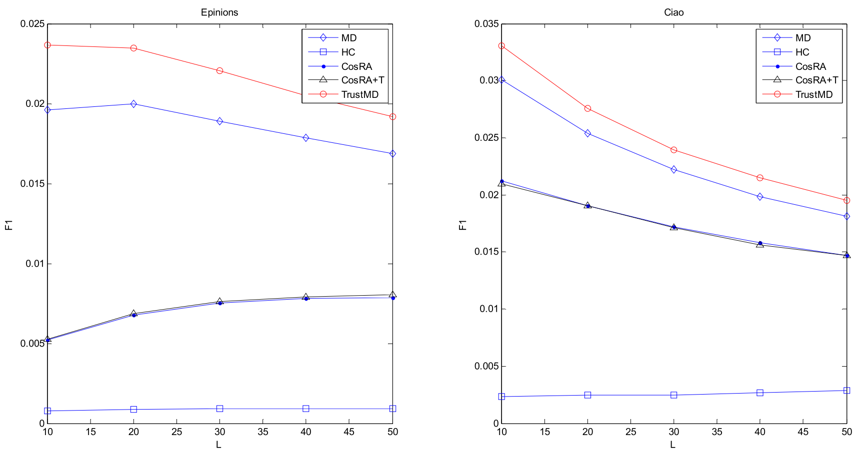


FIGURE 3. The F1 of different methods as parameter L varies from 10 to 50.

and the results are shown in Figure 3. As we can see from Figure 3, TrustMD has a stable high F1 index. In addition, as MD has better accuracy than CosRA on Epinions and Ciao, CosRA +T, which is based on CosRA, does not have remarkable performance in this experiment.

We further provide a more comprehensive evaluation of TrustMD algorithm on six metrics and compare its performance with the benchmark methods with $L = 10$ in Table 2. All the metrics are L-dependent and described in *Evaluation Metrics Section*. In the experiments, TrustMD uses the optimal value of λ, that is $\lambda = 0.75$ for Epinions and $\lambda = 0.1$ for Ciao. And we also give the result when $\lambda = 0$, denoted by TrustMD@0, to show the effect when only implicit trust relations employed.

As shown in Table 2, the proposed TrustMD algorithm outperforms all the four benchmark methods on the three accuracy metrics $\mathrm{P}(L)$, $\mathrm{R}(L)$, and $\mathrm{F1}(L)$ for both Epinions and Ciao, that is, TrustMD has the best accuracy among all the considered recommendation methods.

For the Hamming distance $\mathrm{H}(L)$, CosRA +T has the best performance for both Epinions and Ciao. For intra-similarity $\mathrm{I}(L)$, TrustMD@0 achieves the best and TrustMD achieves the second-best performance for Epinions, but MD achieves the best for Ciao. For novelty $\mathrm{N}(L)$, HC has the stable and best performance for both Epinions and Ciao. So in general, the main advantage of TrustMD is its high accuracy. As there is often a tradeoff between accuracy and diversity, TrustMD is still a very effective method.

C. THE IMPACT OF IMPLICIT TRUST RELATIONS

In our proposed TrustMD, both explicit trust relations and implicit trust relations are considered. We linearly combine the resource from social networks with explicit trust relations and resource from user-object networks with implicit trust relations. When $\lambda = 0$, only implicit trust relations employed, that is TrustMD@0. To explore the effect of implicit trust relations, we compared TrustMD@0 with MD, as the only difference between TrustMD@0 and MD is that TrustMD@0 integrates implicit trust relations into MD.

The results of TrustMD@0 and MD for Epinions and Ciao also have been presented in Table 2. As we can see

TABLE 2. Performance comparison of different recommendation methods for Epinions and Ciao with $L = 10$.

Epinions	P (L)	R(L)	F1(L)	H(L)	I(L)	Nov(L)
MD	0.01717	0.02282	0.01958	0.93704	0.07599	45.90
HC	0.00061	0.00114	0.00078	0.99099	0.09924	**1.08**
CosRA	0.00515	0.00530	0.00521	0.99758	0.13891	3.32
CosRA+T	0.00523	0.00532	0.00526	**0.99764**	0.14096	3.29
TrustMD@0	0.02021	0.02808	0.02349	0.87848	**0.06554**	62.77
TrustMD	**0.02034**	**0.02823**	**0.02364**	0.88068	0.06568	62.29
Ciao	P (L)	R(L)	F1(L)	H(L)	I(L)	Nov(L)
MD	0.01870	0.07697	0.03008	0.96163	**0.04254**	44.35
HC	0.00146	0.00623	0.00236	0.99253	0.04626	**1.86**
CosRA	0.01349	0.04982	0.02122	0.99241	0.04639	15.90
CosRA+T	0.01337	0.04895	0.02100	**0.99248**	0.04642	15.85
TrustMD@0	0.02061	0.08304	0.03302	0.94864	0.05189	50.40
TrustMD	**0.02062**	**0.08306**	**0.03303**	0.94864	0.05189	50.40

from Table 2, compared with MD, TrustMD@0 has better performance on the three accuracy metrics P(L), R(L), and F(L) for both Epinions and Ciao. This is consistent with the conclusion that TrustMD has a remarkable improvement in accuracy, which we have analyzed in previous *Performance of Recommendation Section*.

In order to further analyze the role of implicit trust relations, we integrate the implicit trust relation into another diffusion-based method CosRA [29] and use the same policy for explicit trust relations, and then get TrustCosRA. The resource transfer matrix for CosRA is written as

$$s_{\beta\alpha}^{CosRA} = \frac{1}{\sqrt{k(o_\alpha)k(o_\beta)}} \sum\nolimits_{j=1}^{m} \frac{a_{j\beta}a_{j\alpha}}{k(u_j)} \tag{21}$$

For an arbitrary object o_β, the final resource received from the user-object network with implicit trust relations can be written as

$$f_{i\beta}^{\prime TrustCosRA} = \sum\nolimits_{\alpha=1}^{n} \left(\frac{1}{\sqrt{k(o_\alpha)k(o_\beta)}} \sum\nolimits_{j=1}^{m} \frac{a_{j\beta}a_{j\alpha}}{\sqrt{k(u_i)k(u_j)}}\right) f_{i\alpha} \tag{22}$$

So after the two parts of diffusion processes, the total final resource of o_β can be defined as

$$\begin{aligned} f_{i\beta}^{\prime\prime TrustCosRA} &= (1-\lambda) \sum\nolimits_{\alpha=1}^{n} \left(\frac{1}{\sqrt{k(o_\alpha)k(o_\beta)}} \sum\nolimits_{j=1}^{m} \frac{a_{j\beta}a_{j\alpha}}{\sqrt{k(u_i)k(u_j)}}\right) f_{i\alpha} \\ &+ \lambda \sum\nolimits_{j=1}^{m} \frac{b_{ij}a_{j\beta}f_{j\beta} \sum_{\alpha=1}^{n} a_{i\alpha}a_{j\alpha}}{k(u_j)\sqrt{k\left(u_i\right)k\left(u_j\right)}} \end{aligned} \tag{23}$$

We then conducted experiments for TrustCosRA on Epinions and Ciao, the results are shown in Table 3.

As we can see from Table 3, TrustCosRA has the best performance on the three accuracy metrics P(L), R(L), and F(L) for both Epinions and Ciao. Besides, TrustCosRA@0 outperforms CosRA on P(L), R(L), and F1(L) for both two datasets, which helps us to confirm the finding that implicit trust relations can effectively enhance the accuracy of the basic diffusion-based method.

D. THE IMPACT OF EXPLICIT TRUST RELATIONS

To explore the effect of explicit trust relations, we compare TrustMD with TrustMD@0, and also TrustCosRA with TrustCosRA@0, as different from TrustMD@0 and TrustCosRA@0, TrustMD and TrustCosRA use extra explicit trust relations obtained from social networks. As we can see from Table 2, when employing explicit trust relations, TrustMD performs better than TrustMD@0 on P(L), R(L), F1(L), H(L), and Nov(L) for Epinions, though the optimal value of parameter λ is set based on F1-measure. However, for Ciao, TrustMD only has a slight advantage over TrustMD@0 on P(L), R(L), and F1(L).

From Table 3, TrustCosRA outperforms TrustCosRA@0 on P(L), R(L), F1(L) and I(L) for Epinions. However, the result for Ciao in Table 3 is the same with TrustMD in Table 2, that is, TrustCosRA only has a slight advantage over TrustCosRA@0 on P(L), R(L), F1(L), and H(L).

So for different datasets, social networks don't always work well. One possible reason is that the social data is very sparse, so it has little impact on recommendation results, and the other reason may lie on the structure of the social networks, the public published social datasets are limited and the trust relations are different, some may have no significant

TABLE 3. Performance comparison of CosRA, trustCosRA@0, and trustCosRA for Epinions and Ciao with $L = 10$.

Epinions	P (L)	R(L)	F1(L)	H(L)	I(L)	Nov(L)
CosRA	0.00515	0.00530	0.00521	**0.99758**	**0.13891**	**3.32**
TrustCosRA@0	0.01291	0.01279	0.01284	0.99595	0.18981	10.37
TrustCosRA	**0.01326**	**0.01335**	**0.01329**	0.99573	0.18773	10.97
Ciao	P (L)	R(L)	F1(L)	H(L)	I(L)	Nov(L)
CosRA	0.01349	0.04982	0.02122	**0.99241**	**0.04639**	**15.90**
TrustCosRA@0	0.01725	0.05965	0.02676	0.99081	0.07959	18.91
TrustCosRA	**0.01728**	**0.05967**	**0.02679**	0.99082	0.07950	18.93

TABLE 4. Performance comparison of different recommendation methods for FilmTrust with $L = 10$.

FilmTrust	P (L)	R(L)	F1(L)	H(L)	I(L)	Nov(L)
MD	0.35241	0.63744	0.45388	0.62733	0.25910	570.21
HC	0.00697	0.01322	0.00911	**0.84899**	**0.01962**	**3.74**
CosRA	0.35814	0.64725	0.46111	0.66931	0.25626	541.05
CosRA+T	0.35807	0.64690	0.46097	0.68177	0.25196	530.85
TrustCosRA@0	0.35935	0.64681	0.46200	0.68719	0.25851	530.20
TrustCosRA	**0.35990**	**0.64812**	**0.46279**	0.68963	0.25826	530.01

impact on recommendations. Therefore, explicit trust relations should be used as an auxiliary input for recommender systems.

E. A GENERAL FRAMEWORK

As we can see from Table 2 and Table 3, though TrustMD and TrustCosRA all improve the basic method MD and CosRA on F1 metric, respectively, but TrustMD has better performance on F1 than TrustCosRA for both Epinions and Ciao. The reason is that MD achieves the best performance on F1 in the three basic methods MD, HC, and CosRA, and TrustMD stands on the shoulders of the giant MD. However, the giant changes when we carried out experiments on relatively dense dataset FilmTrust [39], which has 1508 users, 2071 items, 35497 ratings, and 1632 user links, with rating-sparsity 1.1366e-2 and user-sparsity 7.1765e-4. The experimental results of Filmtrust are shown in Table 4. As we can see from Table 4, CosRA achieves the best performance on F1 in the three benchmark methods MD, HC, and CosRA, and TrustCosRA achieves overall the best performance on F1. So in order to find the best trust-based diffusion method, we must first find the best underlying basic diffusion method and apply our implicit and explicit trust relations. Moreover, MD achieves the best performance on F1 for very sparse datasets Epinions and Ciao among the basic methods MD, HC, and CosRA, while CosRA achieved the best performance on F1 for relatively dense datasets FilmTrust. The results indicate that data sparsity may be an important factor affecting the performance of algorithms on datasets. When only implicit trust relations or both implicit and explicit trust relations are integrated into MD or CosRA, better accuracy of recommendation is achieved compared with their basic method MD or CosRA.

Motivated by the results of experiments, we further extend our idea to a general form by introducing the implicit trust relations into the more general diffusion recommendation method Hybrid HeatS +Probs (HHP for short), which is proposed by Zhou et al. [40], and employ the same strategy for explicit trust relations. We call this general form Trust-Diffusion for convenience. The transfer matrix of HHP is written as

$$s_{\beta\alpha}^{HHP} = \frac{1}{k(o_\beta)^{1-\theta} k(o_\alpha)^{\theta}} \sum_{j=1}^{m} \frac{a_{j\beta} a_{j\alpha}}{k(u_j)} \tag{24}$$

where θ is a tunable parameter. When $\theta = 1$, HHP method reduces to MD, when $\theta = 0.5$, reduces to CosRA, and when $\theta = 0$, reduces to HC. So the final resource received from the user-object network can be written as

$$f_{i\beta}^{\prime TrustHHP} = \sum_{\alpha=1}^{n} \left(\frac{1}{k(o_\beta)^{1-\theta} k(o_\alpha)^{\theta}} \sum_{j=1}^{m} \frac{a_{j\beta} a_{j\alpha}}{\sqrt{k(u_i) k(u_j)}}\right) f_{i\alpha} \tag{25}$$

Furthermore, the integration of implicit trust relations can be written in a more general form by introducing another tunable parameter φ, and we denote this form as TrustDiffusion, that is

$$f_{i\beta}^{\prime TrustDiffusion} = \sum\nolimits_{\alpha=1}^{n} \left(\frac{1}{k(o_\beta)^{1-\theta} k(o_\alpha)^{\theta}} \times \sum\nolimits_{j=1}^{m} \frac{a_{j\beta} a_{j\alpha}}{k(u_i)^{1-\varphi} k(u_j)^{\varphi}}\right) f_{i\alpha} \quad (26)$$

When $\varphi = 0.5$, Eq.(26) reduces to Eq.(25).

The other part of resource diffuses on the user-user-object tripartite graph, and the strategy is the same as Eq.(10) in the previous *Proposed Method Section*. So after the two parts of diffusion processes, the total final resource of o_β can be defined in a general framework as

$$\begin{aligned} f_{i\beta}^{\prime\prime TrustDiffusion} &= (1-\lambda) f_{i\beta}^{\prime TrustDiffusion} + \lambda f_{i\beta}^{\prime Trust} \\ &= (1-\lambda) \sum\nolimits_{\alpha=1}^{n} \left(\frac{1}{k(o_\beta)^{1-\theta} k(o_\alpha)^{\theta}} \sum\nolimits_{j=1}^{m} \frac{a_{j\beta} a_{j\alpha}}{k(u_i)^{1-\varphi} k(u_j)^{\varphi}}\right) f_{i\alpha} \\ &\quad + \lambda \sum\nolimits_{j=1}^{m} \frac{b_{ij} a_{j\beta} f_{j\beta} \sum_{\alpha=1}^{n} a_{i\alpha} a_{j\alpha}}{k(u_j) \sqrt{k(u_i)\, k(u_j)}} \end{aligned} \quad (27)$$

where $\lambda \in [0, 1]$, $\theta \in [0, 1]$, $\varphi \in [0, 1]$. When $\theta = 1$ and $\varphi = 0.5$, Eq.(27) reduces to Eq.(11), that is what we called TrustMD in *Proposed Method Section*.

F. COMPLEXITY ANALYSIS

The time complexity of MD is $O(n \times m \times n) = O(mn^2)$ (see Eq.(6)). The time complexity of TrustMD is $O(n \times m + m \times n) \times n = O(mn^2 + mn^2)$ (see Eq.(11)), with two terms accounting for the calculation of resource from the user-object bipartite graph and from the user-user-object tripartite graph. As $O(mn^2 + mn^2) = O(mn^2)$, TrustMD has the same order of magnitude of time complexity as MD. In terms of space complexity, TrustMD needs additional m^2 memory to store the trust matrix $\boldsymbol{B}$. For TrustDiffusion, it is the same with TrustMD. Moreover, if we set $\lambda = 0$ in TrustMD (or TrustDiffusion), then no additional computation and memory from the user-user-object tripartite graph is needed, but it still can get an improvement in accuracy. Though TrustMD needs extra computation and memory, it still has the same order of magnitude of time complexity, and social network as additional input can improve the accuracy of recommender systems, which is a good choice especially for very sparse data. Nevertheless, our work provides a promising way to enhance the performance of recommendation by employing both implicit and explicit trust relations.

V. CONCLUSION

In this paper, we explored how to integrate trust relations into network-based diffusion process to improve the accuracy of recommendation and proposed TrustMD algorithm. User trust relations are classified into two categories: implicit trust relations and explicit trust relations, and both of them are integrated into the standard mass diffusion process (MD). In TrustMD, one part of resource spreads on the user-object bipartite graph, while the other resource spreads on the user-user-object tripartite graph, which is a new respective to employ social networks for recommendation and design diffusion process on heterogeneous networks. We then analyzed the impact of parameter λ, the performance of recommendation, the impact of implicit trust relations and explicit trust relations, respectively. Interestingly, we find that even only implicit trust relations integrated into the diffusion process on user-object bipartite graphs, the accuracy of recommendation is still enhanced.

We further extended our idea to a general form called TrustDiffusion by introducing trust relations into the more general diffusion method. TrustDiffusion broadens the applicable range of our idea, and proper basic network-based diffusion process can be chosen according to different datasets and real-world online platforms, and also specific parameters can be selected to get the desired performance.

We consider the widely used trust relations and estimate the trust weight based on rating similarity among users in this paper. However, in addition to trust relations, some other auxiliary information, such as the reputation of users [41], [42], can also be used to improve the performance of recommendation. In the future, we will continue our study on the diffusion-based recommendation with additional features.

REFERENCES

[1] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Phys. Rep.*, vol. 519, no. 1, pp. 1–49, 2012.
[2] G. Zheng, "DRN: A deep reinforcement learning framework for news recommendation," in *Proc. World Wide Web Conf.*, Apr. 2018, pp. 167–176.
[3] M. Mao, J. Lu, J. Han, and G. Zhang, "Multiobjective e-commerce recommendations based on hypergraph ranking," *Inf. Sci.*, vol. 471, pp. 269–287, Jan. 2019.
[4] T. Bogers and A. Bosch, "Recommending scientific articles using CiteULike," in *Proc. ACM Conf. Recommender Syst.*, Oct. 2008, pp. 287–290.
[5] D. Lian, X. Xie, F. Zhang, N. J. Yuan, T. Zhou, Y. Rui, and B. Data, "Mining location-based social networks: A predictive perspective," *IEEE Data Eng. Bull.*, vol. 38, no. 2, pp. 35–46, Jun. 2015.
[6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[7] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, Oct. 2007, Art. no. 046115.
[8] Y.-C. Zhang, M. Blattner, and Y.-K. Yu, "Heat conduction process on community networks as a recommendation model," *Phys. Rev. Lett.*, vol. 99, Oct. 2007, Art. no. 154301.
[9] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, and F. Yang, "Recommendation model based on opinion diffusion," *Europhys. Lett.*, vol. 80, no. 6, Nov. 2007, Art. no. 68003.
[10] T. Zhou, L.-L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *Europhys. Lett.*, vol. 81, no. 5, Feb. 2008, Art. no. 58004.
[11] F. Yu, A. Zeng, S. Gillard, and M. Medo, "Network-based recommendation algorithms: A review," *Physica A, Stat. Mech. Its Appl.*, vol. 452, pp. 192–208, Jun. 2016.
[12] X. Wang, Y. Liu, G. Zhang, Y. Zhang, H. Chen, and J. Lu, "Mixed similarity diffusion for recommendation on bipartite networks," *IEEE Access*, vol. 5, pp. 21029–21038, 2017.
[13] Q.-X. Wang, J. Li, X. Luo, J.-J. Xu, and M.-S. Shang, "Effects of the bipartite structure of a network on performance of recommenders," *Phys. A, Stat. Mech. Its Appl.*, vol. 492, pp. 1257–1266, Feb. 2018.

[14] X. Wang, Y. Liu, and F. Xiong, "Improved personalized recommendation based on a similarity network," *Phys. A, Stat. Mech. Its Appl.*, vol. 456, pp. 271–280, Aug. 2016.

[15] C.-C. Hsu, M.-Y. Yeh, and S.-D. Lin, "A general framework for implicit and explicit social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2228–2241, Dec. 2018.

[16] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Comput. Commun.*, vol. 41, pp. 1–10, Mar. 2014.

[17] F. Xiong, X. Wang, S. Pan, H. Yang, H. Wang, and C. Zhang, "Social recommendation with evolutionary opinion dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[18] Z. Li, F. Xiong, X. Wang, H. Chen, and X. Xiong, "Topological influence-aware recommendation on social networks," *Complexity*, vol. 2019, Dec. 2019, Art. no. 6325654.

[19] F. Riquelme, P. Gonzalez-Cantergiani, X. Molinero, and M. Serna, "Centrality measure in social networks based on linear threshold model," *Knowl.-Based Syst.*, vol. 140, pp. 92–102, Jan. 2018.

[20] S. Peng, Y. Zhou, L. Cao, S. Yu, J. Niu, and W. Jia, "Influence analysis in social networks: A survey," *J. Netw. Comput. Appl.*, vol. 106, pp. 17–32, Mar. 2018.

[21] J. Gao, T. Zhou, and Y. Hu, "Bootstrap percolation on spatial networks," *Sci. Rep.*, vol. 5, Oct. 2015, Art. no. 14662.

[22] Y. Meng, G. Chen, J. Li, and S. Zhang, "PSREC: Social recommendation with pseudo ratings," in *Proc. 12th ACM Conf. Recommender Syst.*, Oct. 2018, pp. 397–401.

[23] D. F. Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti, "Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization," *Future Gener. Comput. Syst.*, vol. 78, pp. 430–439, Jan. 2018.

[24] S. M. Taheri, H. Mahyar, M. Firouzi, K. E. Ghalebi, R. Grosu, and A. Movaghar, "Extracting implicit social relation for social recommendation techniques in user rating prediction," in *Proc. 26th Int. Conf. World Wide Web Companion, Int. World Wide Web Conf. Steering Committee*, Apr. 2017, pp. 1343–1351.

[25] A. J. B. Chaney, D. M. Blei, and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," in *Proc. 9th ACM Conf. Recommender Syst.*, Sep. 2015, pp. 43–50.

[26] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Personalized recommendation via integrated diffusion on user—Item—Tag tripartite graphs," *Phys. A, Stat. Mech. Its Appl.*, vol. 389, pp. 179–186, Jan. 2010.

[27] M.-S. Shang, Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Collaborative filtering with diffusion-based similarity on tripartite graphs," *Phys. A, Stat. Mech. Its Appl.*, vol. 389, no. 6, pp. 1259–1264, Mar. 2010.

[28] L.-J. Chen and J. Gao, "A trust-based recommendation method using network diffusion processes," *Phys. A, Stat. Mech. Its Appl.*, vol. 506, pp. 679–691, Sep. 2018.

[29] L.-J. Chen, Z.-K. Zhang, J.-H. Liu, J. Gao, and T. Zhou, "A vertex similarity index for better personalized recommendation," *Phys. A, Stat. Mech. Its Appl.*, vol. 466, pp. 607–615, Jan. 2017.

[30] X. Deng, Y. Zhong, L. Lü, N. Xiong, and C. Yeung, "A general and effective diffusion-based recommendation scheme on coupled social networks," *Inf. Sci.*, vol. 417, pp. 420–434, Nov. 2017.

[31] X. Wang, Y. Liu, G. Zhang, F. Xiong, and J. Lu, "Diffusion-based recommendation with trust relations on tripartite graphs," *J. Stat. Mech., Theory Exp.*, vol. 8, Aug. 2017, Art. no. 083405.

[32] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. ACM Conf. Recommender Syst.*, Oct. 2007, pp. 17–24.

[33] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "ETAF: An extended trust antecedents framework for trust prediction," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2014, pp. 540–547.

[34] M.-S. Shang, L. Lü, W. Zeng, Y.-C. Zhang, and T. Zhou, "Relevance is more significant than correlation: Information filtering on sparse data," *Europhys. Lett.*, vol. 88, no. 6, Jan. 2009, Art. no. 68008.

[35] W. Zeng, Y.-X. Zhu, L. Lü, and T. Zhou, "Negative ratings play a positive role in information filtering," *Phys. A, Stat. Mech. Its Appl.*, vol. 390, pp. 4486–4493, Nov. 2011.

[36] L. Hu, L. Ren, and W. Lin, "A reconsideration of negative ratings for network-based recommendation," *Phys. A, Stat. Mech. Its Appl.*, vol. 490, pp. 690–701, Jan. 2018.

[37] L. Luo, H. Xie, Y. Rao, F. L. Wang, "Personalized recommendation by matrix co-factorization with tags and time information," *Expert Syst. Appl.*, vol. 119, pp. 311–321, Apr. 2019.

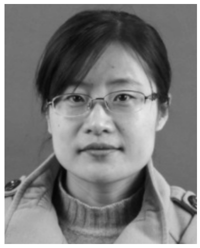
[38] G. Chen, T. Gao, X. Zhu, H. Tian, and Z. Yang, "Personalized recommendation based on preferential bidirectional mass diffusion," *Phys. A, Stat. Mech. Its Appl.*, vol. 469, pp. 397–404, Mar. 2017.

[39] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel Bayesian similarity measure for recommender systems," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Aug. 2013, pp. 2619–2625.

[40] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proc. Nat. Acad. Sci.*, vol. 107, no. 10, pp. 4511–4515, Mar. 2010.

[41] J. Gao, Y.-W. Dong, M.-S. Shang, S.-M. Cai, and T. Zhou, "Group-based ranking method for online rating systems with spamming attacks," *Europhys. Lett.*, vol. 110, no. 2, May 2015, Art. no. 28003.

[42] J. Gao and T. Zhou, "Evaluating user reputation in online rating systems via an iterative group-based ranking method," *Phys. A, Stat. Mech. Its Appl.*, vol. 473, pp. 546–560, May 2017.



YUANZHEN LIU received the M.S. degree in technology of computer application from Jiangnan University, Wuxi, China, in 2008. She is currently pursuing the Ph.D. degree with Hohai University. She is currently a Lecturer with the Yancheng Institute of Technology. Her research interests include mainly in the area of diffusion process, recommender systems, social networks, and machine learning.

LIXIN HAN received the Ph.D. degree in computer science from Nanjing University, Nanjing, China. He has been a Postdoctoral Fellow with the Department of Mathematics, Nanjing University, and a Research Fellow with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He is currently a Professor with the Institute of Intelligence Science and Technology, Hohai University, Nanjing, China. He has published over 60 research papers. He is also an Invited Reviewer for several renowned journals and has been a Program Committee Member of many international conferences. He is listed in Marquis' Who's Who in the World and Marquis' Who's Who in Science and Engineering.

ZHINAN GOU received the B.S. degree in computer science and technology from Hebei Normal University, Shijiazhuang, China, in 2008. He is currently pursuing the Ph.D. degree in software engineering, Hohai University, Nanjing, China. His research interests include information retrieval, recommendation systems, social networks, and pattern recognition.

YI YANG received the M.S. degree in technology of computer application from Nanjing Normal University, Nanjing, China, in 2009. He is currently pursuing the Ph.D. degree with Hohai University. He is currently a Lecturer with Huaibei Normal University. His research interests include mainly in the area of tensor analysis, machine learning, and recommender systems.