

Diffusion-based recommendation with trust relations on tripartite graphs

Ximeng Wang^{1,2}, Yun Liu^{1,*}, Guangquan Zhang², Fei Xiong¹,
Jie Lu²

¹ Key Laboratory of Communication & Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, P.R. China

² Decision Systems & e-Service Intelligence Laboratory, Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia

E-mail: liuyun@bjtu.edu.cn (Y. Liu)

August 2017

Abstract. The diffusion-based recommendation approach is a vital branch in recommender systems, which successfully applies physical processes to make recommendations for users on bipartite or tripartite graphs. Trust links indicate users' social relations and can provide the benefit of reducing data sparsity. However, traditional diffusion-based algorithms only consider rating links when making recommendations. In this paper, the complementarity of users' implicit and explicit trust is exploited, and a novel resource-allocation strategy is proposed, which integrates these two kinds of trust relations on tripartite graphs. Through empirical studies on three benchmark datasets, our proposed method obtains better performance than most of the benchmark algorithms in terms of accuracy, diversity and novelty. According to the experimental results, our method is an efficient and reasonable way to integrate additional features into the diffusion-based recommendation approach.

Keywords: critical phenomena of socio-economic systems, socio-economic networks, analysis of algorithms.

1. Introduction

With the explosive development of the Internet, the amount of information available is growing far more quickly than our ability to process it, and information overload has become a serious problem in modern society [1]. To solve this problem, various information filtering technologies have been proposed and play very important roles in data processing at present [2]. Recommender systems are a typical information filtering technology, which have been widely used to offer users personalized recommendations among numerous potential choices [3, 4]. In recent years, recommender systems have started to appear in a number of applications, such as recommending academic articles

* Corresponding author.

[5], videos [6], movies [7], locations [8], music [9], and services for e-business and e-government [10].

A number of advanced recommendation algorithms have been proposed by researchers both in physics and computer science, such as global ranking methods [11], collaborative filtering [12], content-based analysis [13], matrix factorization [14] and diffusion-based methods [15, 16, 17], among which collaborative filtering and diffusion-based methods are the most representative in computer science and physics, respectively. The main idea of collaborative filtering is to assume users with past similar preferences are likely to make the same choices in the future. Historical data is used to predict users' collections [18]. User-based collaborative filtering (UCF) and item-based collaborative filtering (ICF) are crucial parts of collaborative filtering [19]. More specifically, the UCF focuses on evaluating the taste and preference of different users while the ICF calculates the similarity between items. However, collaborative filtering suffers from data sparsity, so accessorial data is introduced into recommender systems as prior information to alleviate this problem [18]. Furthermore, physical dynamics, which is employed in complex networks by diffusion-based methods and can make personalized recommendations, have attracted great attention from many researchers [20]. Mass diffusion (MD) [15] and heat conduction (HC) [16] can be regarded as the pioneers of diffusion-based recommendation approaches. They distribute the resource of each node through a two-step resource-allocation process on bipartite graphs in different ways. Subsequently, Liu et al. [21] proposed a weighted heat conduction algorithm to enhance accuracy; Jia et al. [22] found a new strategy to distribute the initial resource; Zhou et al. [23] integrated both the MD and HC methods to generate a hybrid recommendation approach, which improves accuracy and diversity at the same time; and Wang et al. [24] introduced a similarity-based resource-allocation process to make recommendations more personalized.

Social recommendation has become more and more important with the increasing popularity of social media and social networks [25, 26]. Trust information indicates users' social relations, which are widely used in social recommendation models [27]. Guo et al. [18] integrated user trust relations into a matrix factorization model; Ma [28] proposed an experimental study on exploring implicit social recommendations; Tang et al. [29] exploited signed social networks for recommendations; and Zhang et al. [30] formulated the global and local influence of user preferences and incorporated them into a matrix factorization model. However, there is a lack of diffusion-based recommendation methods integrated with social trust relations. Some previous research results demonstrate that social information can bring significant improvements to recommendations and a vital additional feature for ratings in recommender systems.

In this paper, a novel diffusion-based recommendation method with trust relations (DBRT for short) on tripartite graphs is proposed to integrate users' social trust into a recommendation process. The tripartite graph has been verified as an effective way to combine extra features into the diffusion-based recommendation approach and has already been applied in collaborative tagging systems [31, 32]. However, they only apply

the original MD algorithm on two different bipartite graphs firstly, and then combine the final resource on tripartite graphs. Compared to existing algorithms on tripartite graphs, our method provides a consistent and synergetic resource-allocation process that combines the resource from a user-object network and a user trust network in the first step and lets the resource flow back to objects in the second step. Moreover, users social trust relations, such as implicit and explicit trust, are introduced into the diffusion-based recommendation approach in our method. Extensive experiments on three real-world datasets indicate that DBRT obtains remarkable improvements over most of the benchmark approaches.

The remainder of this paper is organized as follows. Section 2 presents the proposed DBRT method that makes recommendations with social trust relations on tripartite graphs. The descriptions of the three datasets and evaluation metrics are shown in Section 3. The experimental results and comparisons are then presented in Section 4, and Section 5 concludes this paper with discussions and findings.

2. Diffusion-based recommendation with trust relations

In a recommender system, user-object relations can be described on a bipartite graph $G(U, O, E)$. The user set is defined as $U = \{u_1, u_2, \dots, u_m\}$ and the object set is defined as $O = \{o_1, o_2, \dots, o_n\}$, where m and n are the numbers of users and objects. The link set between users and objects is defined as $E = \{e_1, e_2, \dots, e_z\}$ and z is the amount of links. We can use an $m \times n$ adjacent matrix A to describe the bipartite graph $G(U, O, E)$. To make it easy to understand, we use Greek and Latin letters to express object-related and user-related indices respectively. Accordingly, the element in the adjacent matrix is represented as $a_{i\alpha} = 1$ if there is a link between node o_α and node u_i , which means object α is collected by user i , and $a_{i\alpha} = 0$ otherwise. The degree of the object α and user i are defined as $k(o_\alpha)$ and $k(u_i)$, which represents the number of users who collect object α and the number of objects collected by user i , respectively. The primary purpose of a recommender system is to provide a recommendation list for a target user. That is to say, a set of objects uncollected by the target user with the highest recommendation scores should be included in the recommendation list. The length of the recommendation list is defined as L in this paper.

Before introducing our model, we classify social trust relations in recommender systems into two categories, which are explicit trust and implicit trust. The explicit trust means that trust statements are directly specified by users. For instance, users can add other users to their trust lists or establish friendships with other users on social websites, such as Ciao, Epinions and Facebook. By contrast, the implicit trust is the relationship that cannot be directly observed in social trust networks [18]. It is often inferred by other information, such as user similarities in ratings. In this section, we propose a novel diffusion-based recommendation method integrated with both categories of social trust relations. Additional details about the proposed method are discussed in the following subsections.

2.1. Implicit trust between users

Considering a situation that two users have a lot of related behaviors, such as purchasing the same things or giving similar ratings to movies. However, their trust relation cannot be observed in their trust or friend lists. According to their similar user behaviors, we assume that implicit trust exists between these two users, which means they may share and adopt each other's choices and preferences. In this paper, we use the similarity between users to calculate the implicit trust in our proposed method. The cosine index is a widely used similarity evaluation approach on bipartite graphs, and it has already proved effectual for measuring the similarity between objects [19]. As a matter of fact, the cosine index calculates an inner product space of two object vectors. For two objects α and β , the cosine index is defined as Eq. (1).

$$\cos_{\alpha\beta}^o = \frac{1}{\sqrt{k_\alpha k_\beta}} \sum_{i=1}^m a_{i\alpha} a_{i\beta}, \quad (1)$$

where k_α and k_β are the degree of objects α and β , $a_{i\alpha}$ and $a_{i\beta}$ are the elements in the adjacent matrix which indicate the links between user i and these two objects α and β . Using Eq. (1) the similarity can be measured by the cosine index if the two objects are collected by the same user at least once, otherwise the similarity will be 0.

We suggest using the analogous method to calculate the similarity between users. User rating behaviors can also be expressed as user vectors, e.g., if a user collects an object, the corresponding element in the user vector is 1; otherwise it is 0. An inner product space of two user vectors measures the similarity between these two users, which is defined as Eq. (2).

$$\cos_{ij}^u = \frac{1}{\sqrt{k_i k_j}} \sum_{\alpha=1}^n a_{i\alpha} a_{j\alpha}, \quad (2)$$

where k_i is the degree of user i and k_j is the degree of user j , respectively. We suppose that an implicit trust relation between two users exists if there is a similarity between them. The values of similarity indicate the level of implicit trust. For example, if two users collect many common objects, they will have a large inner product that is equivalent to a large value of similarity between these users, which demonstrates a high level of implicit trust between them. Although the explicit trust of these two users is not observed on trust networks, they could share each other's preferences and opinions as well. The implicit trust is a significant feature in recommender systems, and introducing it into the diffusion-based recommendation method can make resource-allocation processes be more reasonable and effective.

2.2. Resource-allocation on explicit trust networks

Explicit trust means that the trust relations between users are observed. In social networks, explicit trust can be divided into two categories: symmetric trust and asymmetric trust. Specifically, asymmetric trust is a more common situation and symmetric trust can be represented by two asymmetric trust links. Trust relations

on Twitter are a typical example of asymmetric trust that a user could be a follower for other users on Twitter, but the other users may not be followers of the user as well, which means the trust relations are asymmetric.

We propose a resource-allocation process on trust networks where explicit trust relations are observed. The main idea is a user can obtain resource from followers on asymmetric trust networks. The explicit trust network can be defined as a monopartite graph represented on an $m \times m$ adjacent matrix B , where the element $b_{ij} = 1$ if the explicit trust is observed on the trust network, otherwise $b_{ij} = 0$. We only make a one-step resource-allocation process so that users obtain resource from their trusting users, e.g., followers. This is different to the original MD method [15] because we cannot ensure that the trusted user is also a trusting user simultaneously in asymmetric trust networks, so we do not let the resource flow back. The resource-allocation process on the explicit trust network is defined as Eq. (3).

$$f_j^t = \sum_{l=1}^m \frac{b_{jl}}{k_l^t} f^t(l), \quad (3)$$

where f_j^t means the resource obtained by user j on the explicit trust network, b_{jl} is the element in the adjacent matrix and k_l^t is the degree of user l on the explicit trust network. Note that $f^t(l)$ is the initial resource of user l on the explicit trust network and it is one unit of resource for convenient calculation [31]. In this paper, the explicit trust is regarded as an auxiliary feature that helps the implicit trust improve the performance of the diffusion-based recommendation approach.

2.3. Recommendation with integrated social trust relations

This section will propose a diffusion-based recommendation method to integrate the implicit trust and explicit trust. One previous study [19] indicates CosRA-based method applies CosRA index between objects to make better recommendations, which is a two-step resource distribution process on a user-object bipartite graph. In each step, the CosRA-based method distributes resource based on the degree of both each object and its neighbouring users at the same time, shown in Eq. (4).

$$f'^{(i)} = S^{CosRA} f^{(i)}, \quad (4)$$

where $S^{CosRA} = \frac{1}{\sqrt{k_\alpha k_\beta}} \sum_{j=1}^m \frac{a_{j\alpha} a_{j\beta}}{k_j}$ is CosRA index to measure the similarity between objects α and β . $f^{(i)}$ is a n -dimensional vector indicating the initial resource of all objects given the target user i and $f'^{(i)}$ is the vector recording all the final resource of each object. However, the CosRA-based method only considers the similarity between objects without the implicit trust between users in its recommendation process. To solve this weakness, we distribute resource based on not only each object's degree but also the implicit trust between the target user and the neighbouring users of each object. In this paper, our method is described as a two-step resource distribution process.

Step 1: we assume the objects collected by the target user i are assigned with one unit of resource which will be distributed to all the neighbouring users. The resource of

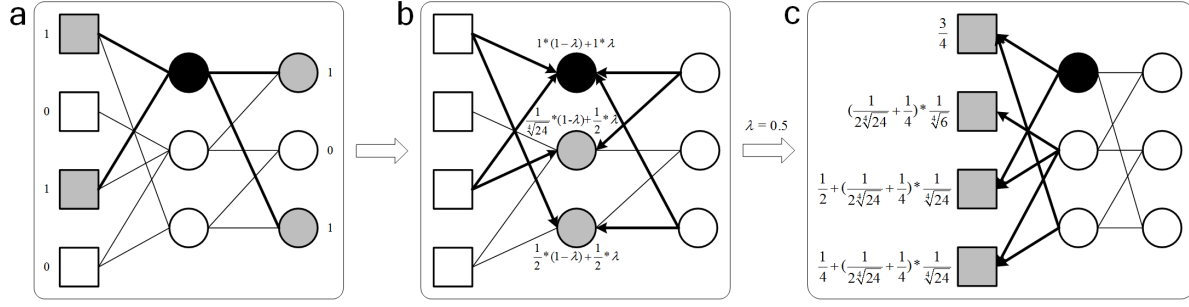


Figure 1. (Color online) An illustration of the diffusion-based recommendation method with trust relations (DBRT). Users and objects are represented by circles and squares, respectively. In each plot, the links between circles and squares on the left side are user-object relations, and the links between circles on the right side are explicit trust relations. The black circle means the target user. The circles and squares with grey color indicate the resource is distributed on these nodes. Plot (a) is the initial configuration, objects and followers linked with the target user obtain one unit of resource. Plot (b) indicates the step 1, in which the resource of objects and followers linked with the target user flows to users on tripartite graphs, the resource of each user can be calculated by Eq. (6). Plot (c) is the step 2, in which the users distribute resource to all the objects linked with them, the final resource of each object can be calculated by Eq. (7). $\lambda = 0.5$ is used as an example.

user j obtained from objects can be written as Eq. (5).

$$f'_{ij} = \sum_{\alpha=1}^n \frac{a_{i\alpha}a_{j\alpha}}{\sqrt{k_{\alpha}}\sqrt{k_i k_j}} f(\alpha), \quad (5)$$

where k_{α} is the degree of object α , k_i is the degree of the target user i and k_j is the degree of user j who obtains resource from the neighbouring objects. $f(\alpha)$ is the initial resource of object α , which is one unit of resource in this paper. It can be seen from Eq. (5), the cosine index between is used to evaluate the implicit trust between users. Furthermore, the explicit trust also needs to be utilized in our method to improve the recommendation performance. As mentioned in Section 2.2, we consider to employ a one-step resource-allocation process to represent the effect of explicit trust. So, the user j not only gets resource from the user-object network but also obtains resource from the explicit trust network simultaneously. We adopt a simply way to linearly combine the resource from the explicit trust network with the resource from objects, defined as Eq. (6).

$$\begin{aligned} f'_{ij} &= \lambda * f_{ij}^t + (1 - \lambda) * \sum_{\alpha=1}^n \frac{a_{i\alpha}a_{j\alpha}}{\sqrt{k_{\alpha}}\sqrt{k_i k_j}} f(\alpha) \\ &= \lambda * \sum_{l=1}^m \frac{b_{il}b_{jl}}{k_l^t} f^t(l) + (1 - \lambda) * \sum_{\alpha=1}^n \frac{a_{i\alpha}a_{j\alpha}}{\sqrt{k_{\alpha}}\sqrt{k_i k_j}} f(\alpha), \end{aligned} \quad (6)$$

where $f_{ij}^t = \sum_{l=1}^m \frac{b_{il}b_{jl}}{k_l^t} f^t(l)$ is extended from f_j^t in Eq. (3), because the target user i is considered in the recommendation process. The parameter $\lambda \in [0, 1]$ is a tunable

parameter to control the proportion of resource from objects and followers.

Step 2: the resource of users should flow back to objects. Assuming object β will obtain the resource from users, the final resource of object β can be calculated as Eq. (7).

$$f'_{i\beta} = \sum_{j=1}^m \frac{a_{j\beta}}{\sqrt{k_\beta} \sqrt{k_i k_j}} f'_{ij}, \quad (7)$$

where k_β is the degree of object β and f'_{ij} is the resource of user j after the step 1. We substitute Eq. (6) into Eq. (7) to generate the final model of our proposed method, as presented in Eq. (8). An example of our method is shown in Figure 1.

$$f'_{i\beta} = \sum_{j=1}^m \frac{a_{j\beta}}{\sqrt{k_\beta} \sqrt{k_i k_j}} \left(\lambda * \sum_{l=1}^m \frac{b_{il} b_{jl}}{k_l^t} f^t(l) + (1 - \lambda) * \sum_{\alpha=1}^n \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha} \sqrt{k_i k_j}} f(\alpha) \right). \quad (8)$$

Finally, all objects are sorted by their final resource and then a top- L recommendation list of uncollected objects is generated for the target user i . A simpler model is shown in Eq. (9), when $\lambda = 0$.

$$\begin{aligned} f'_{i\beta} &= \sum_{j=1}^m \frac{a_{j\beta}}{\sqrt{k_\beta} \sqrt{k_i k_j}} \sum_{\alpha=1}^n \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha} \sqrt{k_i k_j}} f(\alpha) \\ &= \sum_{j=1}^m \frac{a_{j\beta}}{\sqrt{k_i k_j}} \sum_{\alpha=1}^n \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha} k_\beta} f(\alpha). \end{aligned} \quad (9)$$

Under this circumstance, our model only takes advantage of the implicit trust to improves the recommendation performance.

Note that the resource from the explicit trust network is regarded as additional resource for the resource from the user-object network. Each user obtains the additional resource in step 1 and then transfers it to linked objects in step 2. A user with large degree on the explicit trust network is an influential user who can transfer more additional resource to objects. Therefore, the objects collected by the influential user get more final resource and these objects are more likely to be arranged at the top of a recommendation list. As a result, the recommendation performance will be improved, because the influential user's choices have a higher probability to be accepted by others.

3. Dataset and Evaluation

In this section, details of the three benchmark datasets are described, and then the evaluation methods used in this paper are shown.

3.1. Data descriptions

The three benchmark datasets used in our experiments are Ciao, Epinions and Flixster. These three datasets all contain social information that can be used as trust relations in recommender systems [18]. Ciao, Epinions and Flixster are public, real-world datasets,

Table 1. Statistics of the Ciao, Epinions and Flixster datasets.

Datasets	Users	Objects	Rating links	Sparsity	Trust links
Ciao	2960	4394	77861	5.99×10^{-3}	56998
Epinions	5000	3000	70438	4.70×10^{-3}	139982
Flixster	6072	5366	115840	3.56×10^{-3}	167552

and widely used in the evaluation of previous trust-combined recommender systems. When building bipartite graphs, we convert ratings to binary links by assigning 1 as ‘relevant’ for the ratings above 3 and 0 as ‘non-relevant’ for the remaining ratings. The Ciao dataset consists of 2960 users, 4394 objects and 77,861 observed rating links, while the Epinions dataset has 70,438 rating links from 5000 users and 3000 objects. In addition, the Flixster dataset contains 6072 users, 5366 objects and 115,840 rating links. There are also 56,998, 139,982 and 167,552 trust links in the Ciao, Epinions and Flixster, respectively. Statistics of the datasets are illustrated in Table 1. A five-fold cross-validation is used for evaluations in our experiments. Specifically, we randomly divide each dataset into five folds. Four are regarded as the training set, with the remaining fold treated as the testing set. Five iterations are arranged to make sure that all folds are tested.

3.2. Metrics

To present a comprehensive evaluation of recommendation performance, some widely investigated evaluation metrics are employed to measure the accuracy and diversity of our proposed method.

Precision [19] is an important metric in recommender systems that measures the proportion of the number of recommended objects appearing in the test set to the length of a recommendation list. Mathematically, the average value of precision for all users is defined as

$$P(L) = \frac{1}{m} \left(\sum_{i=1}^m \frac{D_i(L)}{L} \right), \quad (10)$$

where $D_i(L)$ is the number of objects appearing in both the recommendation list L and in the test set aimed at user i .

Recall [23] is also a crucial metric in recommender systems, it measures the proportion of correct recommended objects and the number of total objects in the test set, as

$$R(L) = \frac{1}{m} \left(\sum_{i=1}^m \frac{D_i(L)}{T_i(L)} \right), \quad (11)$$

where $T(i)$ is the number of objects collected by user i in the test set.

F1 [33] is used to provide a comprehensive assessment of our method. It is a two-dimensional vector, which considers both precision and recall simultaneously and

provides a balanced evaluation. The F1 metric is defined as

$$F1(L) = \frac{2P(L) \times R(L)}{P(L) + R(L)}, \quad (12)$$

where $P(L)$ and $R(L)$ are precision and recall when the length of the recommendation list is L .

Rank-biased precision (RBP) [34] assumes each user scans recommended objects from the first place in a recommendation list and browses the next object with a fixed probability p , defined as

$$RBP(L) = \frac{1}{m} \left(\sum_{i=1}^m (1-p) \sum_{\alpha=1}^L c_{i\alpha} p^{\alpha-1} \right), \quad (13)$$

where $c_{i\alpha} = 1$ means user i has already collected α th object in the recommendation list and $c_{i\alpha} = 0$ is the opposite. Users always accept the recommended objects at the top of a recommendation list so that evaluating the performance of recommendation algorithms based on a recommendation sequence is very necessary. RBP is very close to an individuals' actual habits of collecting objects, which is a more reasonable way to evaluate the accuracy of recommendation methods.

Average reciprocal hit rank (ARHR) [35] is different from RBP, as it directly uses the reciprocal of the object's position in a recommendation list without a hypothetical collection probability. The rank of the objects in a recommendation list impacts the possibility that they can be collected by users in the top- n recommender systems. ARHR is defined as

$$ARHR = \frac{1}{m} \left(\sum_{i=1}^m \sum_{\alpha \in hits} \frac{1}{pos_{i\alpha}} \right), \quad (14)$$

where hits means the recommendations are correctly verified in the test sets, accordingly, $pos_{i\alpha}$ is the position of object α in the recommendation list for user i . Note that the object at the bottom of the list should get a low score in this metric, so a higher ARHR denotes better accuracy.

Hamming distance [36] is a common way to measure the diversity of recommendation algorithms, which can be defined as

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

where $C_{ij}(L)$ means the common objects in the recommendation lists for user i and user j . If two users have the same recommendation list, $H_{ij}(L)$ will be 0 and two completely different recommendation lists lead to $H_{ij} = 1$. Finally, the average of H_{ij} over all the user pairs is denoted by the mean distance in Eq. (16).

$$H(L) = \frac{1}{m(m-1)} \sum_{i \neq j} H_{ij}(L). \quad (16)$$

Novelty [3] is a metric to evaluate the algorithm's ability to generate unpopular results. In general, the average popularity of objects in recommendation lists is used to

represent the novelty, which is defined as

$$N(L) = \frac{1}{mL} \left(\sum_{i=1}^m \sum_{o_\alpha \in o_i^L} k_\alpha \right), \quad (17)$$

where k_α is the degree of object α in the recommendation list o_i^L of user i . To some extent, small-degree objects are regarded as unpopular objects. Hence, a small novelty value means the algorithm is good at pushing unpopular objects out.

4. Experimental Results

An introduction of benchmark methods is proposed in Section 4.1. The impact of the parameter in our method is then analyzed in Section 4.2. Comparisons on the performance of accuracy, diversity and novelty between DBRT and eight benchmark approaches are shown in Section 4.3.

4.1. Benchmark methods

Global ranking method (GRM) [11]: All the objects in the dataset are sorted in the descending order based on their degree firstly. Then, the objects with the largest degree that are not collected by the target user are recommended. The number of proposed objects depends on the length of recommendation list L .

User-based collaborative filtering (UCF) [37]: For a target user, collaborative filtering mainly focuses on recommending objects from users who have the similar taste with the target user. Generally, the target user prefers to accept the shared opinions of the most similar users. UCF takes advantage of the cosine similarity to evaluate the preference and taste of each user and then quantify which object the target user will collect.

Mass diffusion (MD) [15]: MD is an alternative name of network-based inference (NBI), and is a classical resource-allocation process on user-object bipartite graphs. In MD, users and objects distribute their resources based on their own degree.

Similarity-preferential mass diffusion (SPMD) [38]: SPMD improves mass diffusion by introducing a parameter that is used to enhance or suppress the weight of users who are most similar to the target user.

Heat conduction (HC) [16]: HC is another resource-allocation process on bipartite graphs. Actually, HC is a physical phenomenon applied in recommender systems in which each node allocates its resource in accordance with the degree of its adjacent nodes. Some previous studies has demonstrated that HC has an outstanding performance in pushing small-degree objects out.

Biased heat conduction (BHC) [21]: BHC is an improvement of the HC method. It considers the degree effects in the last step of the local heat conduction process, which can greatly enhance the accuracy of the standard HC algorithm.

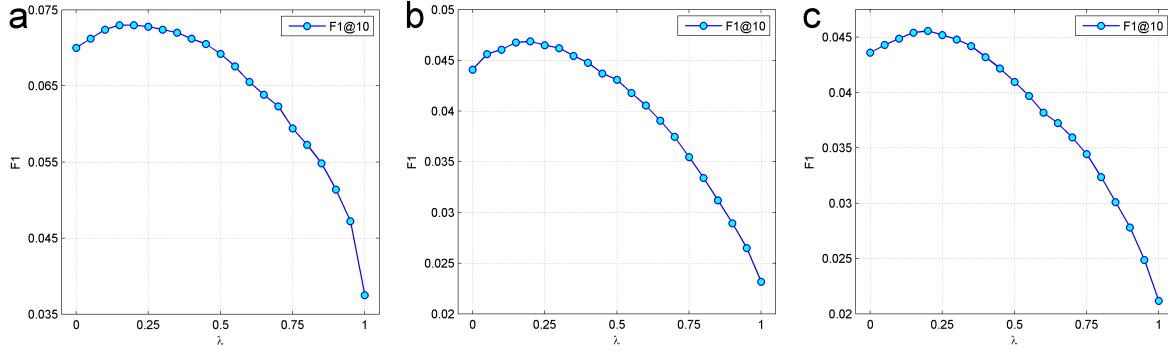


Figure 2. (Color online) The F1 results of DBRT changing the parameter of λ between 0 and 1 at calculation step 0.05 in three datasets when the recommendation list is 10. (a) The optimal value is $\lambda = 0.15$ in the Ciao dataset. (b) The optimal value is $\lambda = 0.2$ in the Epinions dataset. (c) The optimal value is $\lambda = 0.2$ in the Flixster dataset.

CosRA-based method (CosRA) [19]: Both cosine index and resource-allocation index are integrated into the CosRA-based method, which avoids a strong bias on the degree of objects.

Diffusion-based similarity (DBS) [31]: DBS combines a user-object network and a user-tag network into tripartite graphs, which brings tag information of users to the recommendation process. In DBS, the original NBI process is applied independently in the user-object network and the user-tag network firstly, and then the final resource of users on these two networks are combined linearly.

4.2. The impact of parameter λ

Our method uses the parameter λ to integrate the resource from objects and the explicit trust network linearly. To determine the optimal value of λ in our method, we adjust the parameter in experiments on different datasets. The F1 metric provides a comprehensive evaluation of precision and recall, which indicates the accuracy of a recommendation algorithm. Hence, using the F1 metric is a fair way to determine the optimal λ for DBRT.

Figure 2 reports the results of the F1 metric for our method when the parameter λ is changed from 0 to 1 at calculation step of 0.05. The results of the Ciao, Epinions and Flixster datasets are presented in Figure 2(a), (b) and (c), respectively. Because users often pay more attention to the top objects in a recommendation list, the most effective length of a recommendation list is $L = 10$ [39]. Therefore, the optimal value of λ for the top-10 recommendation can be employed in DBRT. In Figure 2(b) and (c), the optimal λ is 0.2 in the Epinions and Flixster datasets, and $\lambda = 0.15$ is the optimal value in the Ciao dataset in Figure 2(a). Although the sparsity of rating links and explicit trust links is totally different, the parameter has an optimal value around 0.2. If $\lambda = 0$, this means that our method only uses the implicit trust calculated by rating data to make recommendations on bipartite graphs. As the value of λ increases, the

Table 2. Results of the seven evaluation metrics after applying our method and the eight benchmark methods on the three datasets. The length of the recommendation list is $L = 50$. All the results are based on a five-fold cross-validation. The best result of each metric for each dataset is in bold.

Ciao	Precision	Recall	F1	RBP	ARHR	Hamming	Novelty
GRM	0.0157	0.1525	0.0285	0.0526	0.1530	0.0926	167
UCF	0.0220	0.2139	0.0399	0.0753	0.2154	0.5396	138
MD	0.0228	0.2161	0.0412	0.0778	0.2227	0.6827	120
SPMD	0.0228	0.2171	0.0413	0.0782	0.2231	0.6926	119
HC	0.0067	0.0556	0.0119	0.0078	0.0318	0.9467	8
BHC	0.0233	0.2200	0.0421	0.0825	0.2343	0.7792	84
CosRA	0.0243	0.2219	0.0437	0.0817	0.2358	0.8786	74
DBS	0.0229	0.2187	0.0415	0.0778	0.2233	0.6523	125
DBRT	0.0252	0.2346	0.0455	0.0874	0.2501	0.8419	82
Epinions	Precision	Recall	F1	RBP	ARHR	Hamming	Novelty
GRM	0.0081	0.1162	0.0151	0.0226	0.0670	0.0505	152
UCF	0.0136	0.1863	0.0254	0.0358	0.1113	0.6868	112
MD	0.0144	0.1897	0.0268	0.0361	0.1141	0.8070	83
SPMD	0.0145	0.1901	0.0269	0.0365	0.1145	0.8170	80
HC	0.0051	0.0649	0.0095	0.0058	0.0251	0.9235	10
BHC	0.0149	0.1921	0.0277	0.0408	0.1256	0.8431	73
CosRA	0.0151	0.1839	0.0279	0.0388	0.1211	0.9039	51
DBS	0.0144	0.1923	0.0269	0.0366	0.1153	0.7739	100
DBRT	0.0163	0.2088	0.0302	0.0503	0.1471	0.8703	64
Flixster	Precision	Recall	F1	RBP	ARHR	Hamming	Novelty
GRM	0.0076	0.0837	0.0139	0.0190	0.0588	0.0477	174
UCF	0.0138	0.1544	0.0253	0.0353	0.1098	0.6912	129
MD	0.0148	0.1581	0.0271	0.0353	0.1134	0.8302	95
SPMD	0.0150	0.1587	0.0274	0.0355	0.1138	0.8240	96
HC	0.0048	0.0501	0.0087	0.0058	0.0242	0.9511	8
BHC	0.0153	0.1587	0.0279	0.0408	0.1264	0.8901	75
CosRA	0.0154	0.1537	0.0280	0.0339	0.1128	0.9408	50
DBS	0.0148	0.1600	0.0271	0.0353	0.1137	0.8019	109
DBRT	0.0169	0.1800	0.0309	0.0474	0.1441	0.9001	71

impact of explicit trust becomes larger. The optimal λ in the three datasets indicates the explicit trust actually improves the performance of DBRT. If λ is very large, it leads to a biased evaluation, because the user-object network is the main domain in assessing user preferences, and the explicit trust network is regarded as auxiliary data used to improve recommendations.

4.3. Performance of recommendations

We use three real-world online rating datasets with trust relations to evaluate our method. In the Ciao dataset, there are fewer trust links than rating links. By contrast,

the trust links are more plentiful in the Epinions and Flixster. These different kinds of datasets reflect various social conditions of users that can verify our method in different situations. We compare our method with eight benchmark methods using the optimal value of parameter for SPMD, BHC and DBS on all three datasets.

The typical results of all methods are presented in Table 2 using a recommendation list $L = 50$. Accuracy is indicated by precision, recall, F1, RBP and ARHR and the higher value in these five metrics, the better the performance. DBRT achieves the highest value for all accuracy metrics across all datasets. Specifically, compared to the DBS method, which is similar to our method and also takes advantage of the additional user’s related feature to improve recommendations on tripartite graphs. Our method enhances precision by 10%, 13.2%, and 14.2% for Ciao, Epinions and Flixster, respectively, when $L = 50$. This means our method provides a more reasonable and effective way to integrate the additional feature into the diffusion-based recommendation approach that increases the accuracy of recommendations. Moreover, compared to CosRA and BHC, our method also achieves much better results on all the accuracy metrics. Figure 3 reports the F1 of all nine algorithms with a recommendation list ranging from 1 to 100. DBRT outperforms the comparison methods. ARHR and RBP are two metrics used to assess algorithms based on an object’s position in a recommendation list. The higher value in these two metrics, the more likely the recommendation is to be collected by users. In this paper, RBP assumes users browse the next object from the first place in a recommendation list with a fixed probability $p = 0.5$. As shown in Table 2, the results demonstrate some significant improvements for RBP provided by our method. Similar to RBP, ARHR is also a sequence-based metric, which means the higher the value in this metric, the more rational and efficient the rank of recommendations. Figure 4 reports that DBRT has some remarkable advantages in ARHR when the length of the recommendation list ranged from 1 to 100. Note that although CosRA has a good performance in precision, its results are worse than BHC in RBP and ARHR, which means CosRA lacks the ability to put the user preferred objects at the top of a recommendation list. Conversely, our method simultaneously improves the performance of precision, RBP and ARHR.

The diversity of recommendations is evaluated by the hamming distance, and novelty represents the ability to push out small-degree objects. HC achieves the best performance in both diversity and novelty, which indicates it focuses on recommending small-degree objects. The diversity of DBRT is close to the CosRA method and shows overwhelming advantages over the other benchmark methods. Particularly for the DBS method, DBRT improves the hamming distance by 29.1%, 12.5% and 12.2% in Ciao, Epinions and Flixster, respectively, when $L = 50$. This indicates our method improves the way auxiliary features are combined into the diffusion-based recommendation. When resource is diffused on tripartite graphs, the node that has a larger degree plays a more important role. In our method, large-degree users can transfer more resource from the explicit trust network to objects, which means the objects linked with the large-degree users can obtain more final resource and will be placed at the top of a recommendation

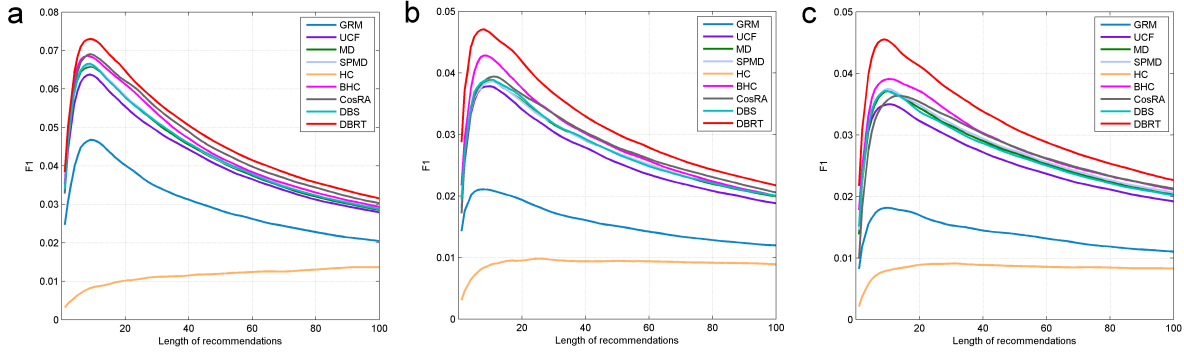


Figure 3. (Color online) The F1 of our method and the eight benchmark methods for the Ciao, Epinions and Flixster are represented in the diagrams (a), (b) and (c), respectively. The length of recommendation list is from 1 to 100.

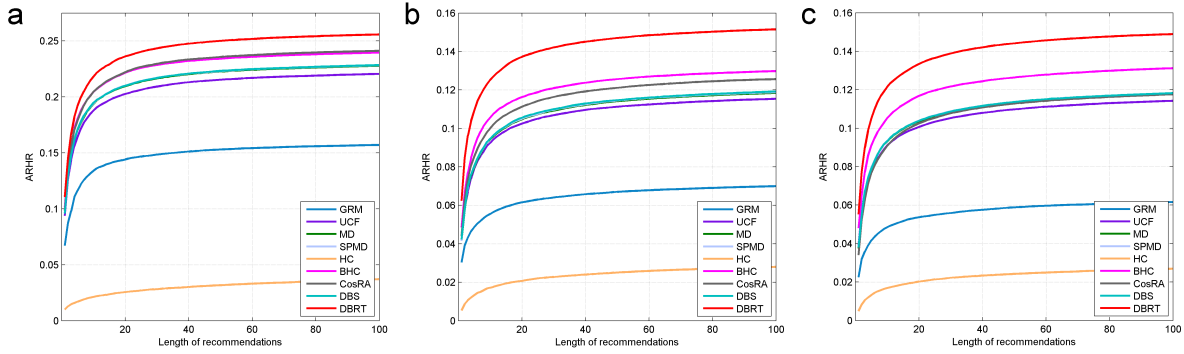


Figure 4. (Color online) The ARHR of our method and the eight benchmark methods for the Ciao, Epinions and Flixster represented in the diagrams (a), (b) and (c), respectively. The length of recommendation list is from 1 to 100.

list. Large-degree users' choices have a greater influence and are more likely to be accepted by other users [21], so it is easy to understand why DBRT achieves better performance in accuracy. Paying more attention to large-degree users and large-degree objects will decrease the diversity and novelty, which explains why DBRT does not show better results over HC or CosRA in diversity and novelty.

5. Conclusion and Further Study

In this paper, users' social trust relations are divided into two parts: implicit trust relations and explicit trust relations, which have different effects on improving recommendations. A novel combination approach is proposed, called diffusion-based recommendation method with trust relations (DBRT), that integrates trust information into the diffusion-based recommendation on tripartite graphs. Experiments on three real-world datasets show our method performs better than most benchmark methods. Specifically, DBRT provides many remarkable improvements in terms of accuracy, diversity and novelty over the DBS, which also uses tripartite graphs to combine

additional features to make better recommendations. The main contributions of our work are: 1) the introduction of implicit trust into the diffusion-based recommendation; 2) the handling of explicit trust relations via a one-step diffusion strategy; and 3) the proposal of a synergetic resource-allocation process to integrate two kinds of trust relations on tripartite graphs. Large-degree objects and users play very important roles in recommender systems. Large-degree objects are preferred by many users, and users with larger degrees may have a greater influence on user-object networks. Therefore, our method pays more attention to large-degree users, which leads to outstanding improvements in accuracy.

Our proposed algorithm provides a new perspective for combining auxiliary data into the diffusion-based recommendation approach, which means our method can be extended to collaborative tagging systems and other recommender systems based on additional features. Moreover, we could further improve our method by introducing two or three parameters, such as in previous studies [21, 23], that use a parameter instead of the cosine similarity in our proposed algorithm. However, it is hard to retain universality in different application scenarios when introducing too many parameters. In future, we will continue our research on improving the performance and effectiveness of the diffusion-based recommendation via additional features, such as social trust relations and time information [40, 41]. We hope this paper will inspire readers in this significant direction.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China under Grant No. 61401015 and the Talent Fund of Beijing Jiaotong University under Grant No. 2015RC013. This work has also been supported by the Fundamental Research Funds for the Central Universities under Grant Nos. 2017JBM013 and W17JB00060.

References

- [1] Ma R T, Lui J and Misra V 2015 *IEEE/ACM Trans. Netw.* **23** 85–98
- [2] Adomavicius G and Tuzhilin A 2005 *IEEE Trans. Knowl. Data Eng.* **17** 734–749
- [3] Lü L, Medo M, Yeung C H, Zhang Y C, Zhang Z K and Zhou T 2012 *Phys. Rep.* **519** 1–49
- [4] Lu J, Wu D, Mao M, Wang W and Zhang G 2015 *Decis. Support Syst.* **74** 12–32
- [5] Bogers T and Van den Bosch A 2008 Recommending scientific articles using citeulike *Proceedings of the 2nd ACM conference on Recommender systems* (ACM) pp 287–290
- [6] Yan M, Sang J and Xu C 2015 Unified youtube video recommendation via cross-network collaboration *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (ACM) pp 19–26
- [7] Bu J, Shen X, Xu B, Chen C, He X and Cai D 2016 *IEEE Trans. Knowl. Data Eng.* **28** 2363–2375
- [8] Jiang S, Qian X, Shen J, Fu Y and Mei T 2015 *IEEE Trans. Multimedia* **17** 907–918
- [9] Deng S, Wang D, Li X and Xu G 2015 *Expert Syst. Appl.* **42** 9284–9293
- [10] Wu D, Zhang G and Lu J 2015 *IEEE Trans. Fuzzy Syst.* **23** 29–43
- [11] Ricci F, Rokach L and Shapira B 2011 *Introduction to recommender systems handbook* (Springer)

- [12] Sarwar B, Karypis G, Konstan J and Riedl J 2001 Item-based collaborative filtering recommendation algorithms *Proceedings of the 10th international conference on World Wide Web* (ACM) pp 285–295
- [13] Balabanović M and Shoham Y 1997 *Comm. ACM* **40** 66–72
- [14] Koren Y, Bell R and Volinsky C 2009 *Computer* **42** 42–49
- [15] Zhou T, Ren J, Medo M and Zhang Y C 2007 *Phys. Rev. E* **76** 046115
- [16] Zhang Y C, Blattner M and Yu Y K 2007 *Phys. Rev. Lett.* **99** 154301
- [17] Zhu X, Tian H and Cai S 2014 *J. Stat. Mech.-Theory Exp.* **2014** P07004
- [18] Guo G, Zhang J and Yorke-Smith N 2016 *IEEE Trans. Knowl. Data Eng.* **28** 1607–1620
- [19] Chen L J, Zhang Z K, Liu J H, Gao J and Zhou T 2017 *Physica A* **466** 607–615
- [20] Yu F, Zeng A, Gillard S and Medo M 2016 *Physica A* **452** 192–208
- [21] Liu J G, Zhou T and Guo Q 2011 *Phys. Rev. E* **84** 037101
- [22] Jia C X, Liu R R, Sun D and Wang B H 2008 *Physica A* **387** 5887–5891
- [23] Zhou T, Kuscsik Z, Liu J G, Medo M, Wakeling J R and Zhang Y C 2010 *Proc. Natl. Acad. Sci. USA* **107** 4511–4515
- [24] Wang X, Liu Y and Xiong F 2016 *Physica A* **456** 271–280
- [25] Zhao Z, Lu H, Cai D, He X and Zhuang Y 2016 *IEEE Trans. Knowl. Data Eng.* **28** 2522–2534
- [26] Xiong F, Liu Y, Zhang Z j, Zhu J and Zhang Y 2012 *Phys. Lett. A* **376** 2103–2108
- [27] Tang J, Gao H, Sarma A D, Bi Y and Liu H 2015 *IEEE Trans. Knowl. Data Eng.* **27** 1724–1738
- [28] Ma H 2013 An experimental study on implicit social recommendation *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (ACM) pp 73–82
- [29] Tang J, Aggarwal C and Liu H 2016 Recommendations in signed social networks *Proceedings of the 25th International Conference on World Wide Web* (ACM) pp 31–40
- [30] Zhang Q, Wu J, Yang H, Lu W, Long G and Zhang C 2016 Global and local influence-based social recommendation *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (ACM) pp 1917–1920
- [31] Shang M S, Zhang Z K, Zhou T and Zhang Y C 2010 *Physica A* **389** 1259–1264
- [32] Zhang Z K, Zhou T and Zhang Y C 2010 *Physica A* **389** 179–186
- [33] Wang W, Zhang G and Lu J 2016 *Decis. Support Syst.* **87** 80–93
- [34] Moffat A and Zobel J 2008 *ACM Trans. Inform. Syst.* **27** 2
- [35] Kabbur S, Ning X and Karypis G 2013 Fism: factored item similarity models for top-n recommender systems *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM) pp 659–667
- [36] Zhou T, Jiang L L, Su R Q and Zhang Y C 2008 *Europhys. Lett.* **81** 58004
- [37] Liu R R, Jia C X, Zhou T, Sun D and Wang B H 2009 *Physica A* **388** 462–468
- [38] Zeng A, Vidmer A, Medo M and Zhang Y C 2014 *Europhys. Lett.* **105** 58002
- [39] Pan W, Liu M and Ming Z 2016 *IEEE Intell. Syst.* **31** 43–49
- [40] Vidmer A and Medo M 2016 *Europhys. Lett.* **116** 30007
- [41] Zhang F, Liu Q and Zeng A 2017 *Expert Syst. Appl.* **85** 270–287