RIT: Enhancing Recommendation with Inferred Trust

Guo Yan(✉), Yuan Yao, Feng Xu, and Jian Lu

State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
{gyan,yyao}@smail.nju.edu.cn, {xf,lj}@nju.edu.cn

Abstract. Trust-based recommendation, which aims to incorporate trust relationships between users to improve recommendation performance, has drawn much attention recently. The focus of existing trust-based recommendation methods is on how to use the observed trust relationships. However, the observed trust relationships are usually very sparse in many real applications. In this paper, we propose to infer some unobserved trust relationships to tackle the sparseness problem. In particular, we first infer the unobserved trust relationships by propagating trust along the observed trust relationships; we then propose a novel trust-based recommendation model to combine observed trust and inferred trust where their relative weights are also learnt. Experimental evaluations on two real datasets show the superior of the proposed method in terms of recommendation accuracy.

Keywords: Recommender system · Trust-based recommendation · Observed trust · Inferred trust

1 Introduction

Recommender system has been widely used in many real applications to find the most attractive items for the users. One of the most successful methods for recommendation is collaborative filtering which is based on the existing historical feedback from users to items. In addition to the user-item feedback, the trust relationships between users have been incorporated to improve the recommendation performance [4,7–9,20,21]. These methods, which are referred to as *trust-based recommendation*, mainly employ the user homophily effect which assumes that socially connected users may have similar preferences for the same item.

To date, the existing trust-based recommendation methods mainly focus on how to make use of the observed trust relationships. However, the observed trust relationships are usually sparse in many real applications. Consider the running example in Fig. 1. As we can see in the example, user A trusts user B, and we can take the rating from user B to item X into consideration if we want to estimate the rating from user A to item X. How to make use of the rating from

T. Cao et al. (Eds.): PAKDD 2015, Part II, LNAI 9078, pp. 756–767, 2015.
DOI: 10.1007/978-3-319-18032-8_59

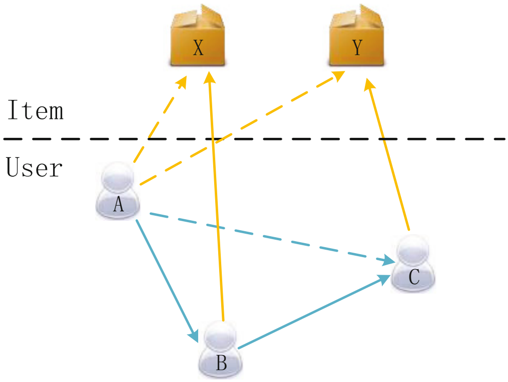


Fig. 1. The solid/dashed lines between users are the observed/inferred trust relationships. The solid lines from users to items are the observed ratings, and the dashed lines between users and items are the ratings we need to estimate (best viewed in color).

user B to item X and the connection from user A to user B is the main focus of the existing trust-based recommendation methods. However, these existing methods may become less accurate if we want to know the possible rating from user A to item Y in the example. Here, our key observation is that since user A trusts user B and user B trusts user C, user A may also trust user C to some extent; consequently, we can estimate the rating from A to item Y based on the inferred trust from user A to user C. Compared to the existing methods that use observed trust relationships only, the inferred trust may also be helpful for recommendation.

In this paper, we propose to infer some unobserved trust relationships for trust-based recommendation. By employing the transitivity property of trust, we first infer several types of unobserved trust based on the observed trust relationships. Next, we propose a novel trust-based recommendation model by incorporating both observed and inferred trust relationships, and we further propose a genetic algorithm to learn the relative weights of different trust relationships.

The main contributions of this work include:

- The proposed RIT model that infers unobserved trust and incorporates both observed and inferred trust for recommendation. The proposed method can tackle the sparseness problem of existing trust relationships, and thus achieve better recommendation performance especially for the cold-start users. To the best of our knowledge, we are the first to directly employ the inferred trust for recommendation.
- Extensive experiments on two real datasets showing the superior of the proposed method over several existing trust-based recommendation methods in terms of recommendation accuracy (e.g., up to 5.27% improvement over the best competitor in terms of MAE). Further, even greater improvement is achieved for the recommendation of cold-start users (e.g., up to 20.81% improvement over the best competitor in terms of MAE).

Table 1. Symbols

Symbols	Definition and Description
$\mathbf{R}$	the user-item rating matrix
$\mathbf{T}$	the observed trust matrix
$\mathbf{U}, \mathbf{V}$	the low-rank matrices for $\mathbf{R}$
$\mathbf{R}'$	the transpose of matrix $\mathbf{R}$
$\mathbf{R}(:, i)$	the i^{th} column of $\mathbf{R}$
$\mathbf{R}(u, i)$	the element at the u^{th} row and i^{th} column of $\mathbf{R}$
$\mathbf{I}^R$	the indicator matrix of $\mathbf{R}$
n, m	the number of users and items
r	the low rank for $\mathbf{U}$ and $\mathbf{V}$

The rest of the paper is organized as follows. Section 2 describes the proposed RIT model. Section 3 presents the experimental results. Section 4 covers related work. Section 5 concludes the paper.

2 The Proposed Model: RIT

In this section, we present the proposed RIT model for recommendation. The main notations we use throughout the paper are listed in Table 1. For example, we use capital bold letter $\mathbf{R}$ and $\mathbf{T}$ to represent the rating matrix from users to items and the observed trust matrix between users, respectively. Based on the notations, the trust-based recommendation methods take the rating matrix $\mathbf{R}$ and the observed trust matrix $\mathbf{T}$ as input, and aim to estimate the potential preference from a user to an item.

2.1 The Proposed Formulation

Inferring Trust. As mentioned in introduction, our key observation is that the inferred trust can help to improve the recommendation performance. Here, we first describe how we obtain the inferred trust. We consider three types of inferred trust [3, 22] as shown in Fig. 2.

The first type of inferred trust is from *direct propagation*. As shown in Fig. 2(a), direct propagation means that if a certain user that is trusted by U_i (e.g., U_3 in Fig. 2(a)) trusts U_j, we can infer that U_i might also trust U_j to some extent. We can obtain this type of trust with the matrix operation $\mathbf{T}^2$.

The second type of inferred trust is called *co-citation*. The intuition behind co-citation is that if two users (e.g., U_i and U_j in Fig. 2(b)) are both trusted by some other users (e.g., U_1 in Fig. 2(b)), they may trust each other. Co-citation can be represented as $\mathbf{T}'\mathbf{T}$ in its matrix form.

The third type of inferred trust is *trust coupling* which is shown in Fig. 2(c). Similar to co-citation, trust coupling is based on the intuition that two users (e.g., U_i and U_j in Fig. 2(c)) may trust each other if both of them trust some other users (e.g., U_2 in Fig. 2(c)). We represent this kind of inferred trust as $\mathbf{T}\mathbf{T}'$.

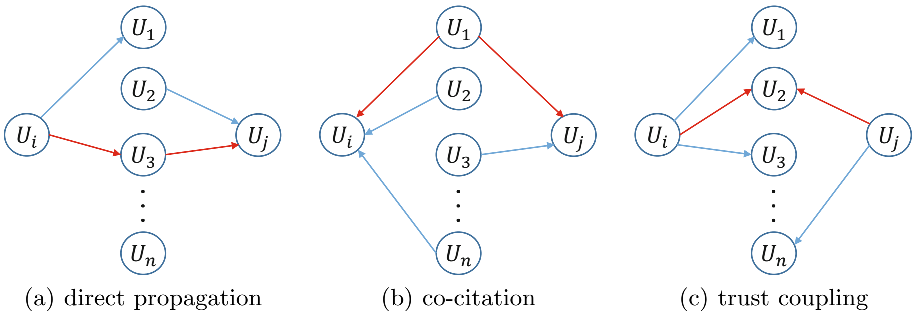


Fig. 2. Infer trust from U_i to U_j with trust propagation (best viewed in color)

Incorporating Observed and Inferred Trust. Next, we show how we incorporate both the observed trust and the three types of inferred trust. To incorporate them, the basic idea is to employ the homophily effect [10], which indicates that similar users tend to have similar preferences for items. In other words, if user u trusts another user v, they might have similar latent preferences for items. Taking observed trust as an example, we can add the following constraint

$$\min \sum_{(u,v)\in\mathbf{T}} \mathbf{T}(u,v)\|\mathbf{U}(:,u)-\mathbf{U}(:,v)\|_2^2 \tag{1}$$

where $\mathbf{U}(:,u)$ and $\mathbf{U}(:,v)$ indicate the preferences of user u and user v, respectively. In Eq. (1), we actually encourage the corresponding user latent preferences $\mathbf{U}(:,u)$ and $\mathbf{U}(:,v)$ to be closed to each other, and greater $\mathbf{T}(u,v)$ would encourage them to be closer. For the three types of inferred trust, we can substitute the $\mathbf{T}$ matrix with the inferred trust matrices such as $\mathbf{T}^2$, $\mathbf{T}'\mathbf{T}$, and $\mathbf{T}\mathbf{T}'$.

To incorporate both observed trust and inferred trust, we adopt the commonly used matrix factorization framework [5] due to its flexibility. By considering all the four trust matrices (one for observed trust and three for inferred trust), we have our RIT model:

$$\begin{aligned}
\min_{\mathbf{U},\mathbf{V}} \mathcal{L}(\mathbf{R},\mathbf{T},\mathbf{U},\mathbf{V}) = \min_{\mathbf{U},\mathbf{V}} &\frac{1}{2}\|\mathbf{I}^R \odot (\mathbf{R}-\mathbf{U}'\mathbf{V})\|_F^2 + \frac{\lambda_u}{2}\|\mathbf{U}\|_F^2 + \frac{\lambda_v}{2}\|\mathbf{V}\|_F^2 \\
&+ \lambda_{g_1} \sum_{(u,v)\in\mathbf{T}} \mathbf{T}(u,v)\|\mathbf{U}(:,u)-\mathbf{U}(:,v)\|_2^2 \\
&+ \lambda_{g_2} \sum_{(u,v)\in\mathbf{T}^2} \mathbf{T}^2(u,v)\|\mathbf{U}(:,u)-\mathbf{U}(:,v)\|_2^2 \\
&+ \lambda_{g_3} \sum_{(u,v)\in\mathbf{T}'\mathbf{T}} (\mathbf{T}'\mathbf{T})(u,v)\|\mathbf{U}(:,u)-\mathbf{U}(:,v)\|_2^2 \\
&+ \lambda_{g_4} \sum_{(u,v)\in\mathbf{T}\mathbf{T}'} (\mathbf{T}\mathbf{T}')(u,v)\|\mathbf{U}(:,u)-\mathbf{U}(:,v)\|_2^2
\end{aligned} \tag{2}$$

where $\lambda_{g_1}, \lambda_{g_2}, \lambda_{g_3}, \lambda_{g_4}$ are the coefficients used to control the weights of the four trust regularization terms, and λ_u and λ_v are used to avoid over fitting. Once the matrices $\mathbf{U}$ and $\mathbf{V}$ are learnt by the above formulation, we can estimate the rating from user u to item i as

$$\hat{\mathbf{R}}(u,i) = \mathbf{U}(:,u)'\mathbf{V}(:,i). \tag{3}$$

2.2 The Proposed Algorithm

Before we present the algorithms to solve the formulation proposed in Eq. (2), we need to introduce a few notations to simplify the descriptions. There are four types of trust regularization terms in Eq. (2), which can be re-written as:

$$\lambda_g \sum_{(u,v)} (\frac{\lambda_{g_1}}{\lambda_g}\mathbf{T} + \frac{\lambda_{g_2}}{\lambda_g}\mathbf{T}^2 + \frac{\lambda_{g_3}}{\lambda_g}\mathbf{T}'\mathbf{T} + \frac{\lambda_{g_4}}{\lambda_g}\mathbf{T}\mathbf{T}')\|\mathbf{U}(:,u) - \mathbf{U}(:,v)\|_2^2 \tag{4}$$

where $\lambda_g = \lambda_{g_1} + \lambda_{g_2} + \lambda_{g_3} + \lambda_{g_4}$.

For simplicity, we define $w_i = \frac{\lambda_{g_i}}{\lambda_g}$, and define matrix $\mathbf{Q}$ as

$$\mathbf{Q} = w_1\mathbf{T} + w_2\mathbf{T}^2 + w_3\mathbf{T}'\mathbf{T} + w_4\mathbf{T}\mathbf{T}' \tag{5}$$

where $w_1 + w_2 + w_3 + w_4 = 1$. This $\mathbf{Q}$ matrix actually contains all the four types of trust (one observed trust and three inferred trust) and their relative weights.

With these new notations, the optimization problem in Eq. (2) can be re-written as

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} \mathcal{L}(\mathbf{R},\mathbf{T},\mathbf{U},\mathbf{V}) = \min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\|\mathbf{I}^R \odot (\mathbf{R} - \mathbf{U}'\mathbf{V})\|_F^2 + \frac{\lambda_u}{2}\|\mathbf{U}\|_F^2 + \frac{\lambda_v}{2}\|\mathbf{V}\|_F^2 \\ \lambda_g \sum_{(u,v)\in\mathbf{Q}} \mathbf{Q}(u,v)\|\mathbf{U}(:,u) - \mathbf{U}(:,v)\|_2^2\end{aligned} \tag{6}$$

Then, the task of the algorithm is to learn the weights ($W = [w_1, w_2, w_3, w_4]$) that are contained in the $\mathbf{Q}$ matrix and the two low-rank matrices ($\mathbf{U}$ and $\mathbf{V}$).

Learning Coefficients W. We adopt the so-called real-valued Genetic Algorithm [11] to learn the weights W. The basic idea is to treat the coefficients $W = [w_1, w_2, w_3, w_4]$ as a chromosome. After randomly generate a group of chromosomes, we can generate its offsprings by three basic operators, i.e., selection, crossover, and mutation.

For the selection operator, we can simply select several best chromosomes. Here, we need a fitness function to determine how good the chromosome is. In this work, we choose prediction accuracy and measure the fitness through cross validation. For the crossover operator, we select two parent chromosomes and generate the child chromosome as $W^{new} = \sigma W_1^{old} + (1-\sigma)W_2^{old}$, where σ is a random number between 0 and 1 to control the proportion of each parent chromosome. For the mutation operator, we select two coefficients in

Algorithm 1. The *computeCoefficient* Algorithm

Input: The $n \times m$ rating matrix $\mathbf{R}$, the $r \times n$ user latent matrix $\mathbf{U}$, the $r \times m$ item latent matrix $\mathbf{V}$, the $n \times n$ trust matrices $\mathbf{T}$, $\mathbf{T^2}$, $\mathbf{T}'\mathbf{T}$, and $\mathbf{T}\mathbf{T}'$, the number of coefficient groups κ, and the maximum iteration number l_c

Output: The coefficients $W = [w_1, w_2, w_3, w_4]$

```
1:  randomly initialize κ groups of coefficients W;
2:  while current iteration number ≤ l_c do
3:     select √κ groups of best coefficients;
4:     for i = 1 to √κ do
5:        for j = 1 to √κ do
6:           crossover a new chromosome W^new from W_i^old and W_j^old;
7:        end for
8:     end for
9:     for i = 1 to κ do
10:       mutate chromosome W_i^new from W_i^old;
11:    end for
12: end while
13: return the best coefficients W;
```

W (e.g., w_i and w_j) and set $w_i^{new} = w_i^{old} - \tau$ and $w_j^{new} = w_j^{old} + \tau$. Notice that, both crossover and mutation will keep the sum of W as 1.

The algorithm is summarized in Alg. 1. We first randomly initialize a group of W's (step 1). For each iteration, we select several best W's (step 3), and generate their offsprings via crossover (step 4-8). Then, we adopt the mutation operator for each offspring (step 9-11). Finally, after several iterations, we return the best W.

Updating Matrices U and V. Next, we show how we update matrices $\mathbf{U}$ and $\mathbf{V}$. First of all, the fourth term of Eq. (6) can be written as

$$
\begin{aligned}
&\sum_{(u,v)\in\mathbf{Q}} \mathbf{Q}(u,v)\|\mathbf{U}(:,u) - \mathbf{U}(:,v)\|_2^2 \\
&= \sum_{(u,v)\in\mathbf{Q}} \sum_{k=1}^{r} \mathbf{Q}(u,v)\|\mathbf{U}(k,u) - \mathbf{U}(k,v)\|_2^2 \\
&= \sum_{(u,v)\in\mathbf{Q}} \sum_{k=1}^{r} \mathbf{Q}(u,v)\mathbf{U}^2(k,u) + \sum_{(u,v)\in\mathbf{Q}} \sum_{k=1}^{r} \mathbf{Q}(u,v)\mathbf{U}^2(k,v) \\
&\quad - \sum_{(u,v)\in\mathbf{Q}} \sum_{k=1}^{r} 2\mathbf{Q}(u,v)\mathbf{U}(k,u)\mathbf{U}(k,v) \\
&= \sum_{k=1}^{r} \mathbf{U}(k,:)(\mathbf{D}_1 + \mathbf{D}_2 - 2\mathbf{Q})\mathbf{U}'(:,k) \\
&= Tr(\mathbf{U}(\mathbf{D}_1 + \mathbf{D}_2 - 2\mathbf{Q})U')
\end{aligned}
$$

Algorithm 2. The *RIT* Algorithm

Input: The $n \times m$ rating matrix $\mathbf{R}$, the $n \times n$ observed trust matrix $\mathbf{T}$, and the latent factor size r

Output: The low-rank matrices $\mathbf{U}$ and $\mathbf{V}$

1: compute inferred trust: $[\mathbf{T}^2, \mathbf{T}'\mathbf{T}, \mathbf{T}\mathbf{T}'] = \mathit{computeInferredTrust}(\mathbf{T})$;
2: initialize $\mathbf{U}$ and $\mathbf{V}$ randomly;
3: learn coefficients: $W = \mathit{computeCoefficient}(\mathbf{U}, \mathbf{V}, \mathbf{R}, \mathbf{T}, \mathbf{T}^2, \mathbf{T}'\mathbf{T}, \mathbf{T}\mathbf{T}')$;
4: compute $\mathbf{Q}$ as defined in Eq. (5);
5: **while** not convergent **do**
6: $\quad \mathbf{U} \Leftarrow \mathbf{U} - \eta\frac{\partial \mathcal{L}}{\partial \mathbf{U}}$ where $\frac{\partial \mathcal{L}}{\partial \mathbf{U}}$ is defined in Eq. (8) and η is the learning step;
7: $\quad \mathbf{V} \Leftarrow \mathbf{V} - \eta\frac{\partial \mathcal{L}}{\partial \mathbf{V}}$ where $\frac{\partial \mathcal{L}}{\partial \mathbf{V}}$ is defined in Eq. (8) and η is the learning step;
8: **end while**
9: **return** $\mathbf{U}$ and $\mathbf{V}$;

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are two diagonal matrices with $\mathbf{D}_1(u,u) = \sum_{v=1}^{n} \mathbf{Q}(u,v)$ and $\mathbf{D}_2(v,v) = \sum_{u=1}^{n} \mathbf{Q}(u,v)$. Since matrix $\mathbf{Q}$ may be asymmetrical, $\mathbf{D}_1$ and $\mathbf{D}_2$ may not equal to each other.

Eq. (6) can be re-written as

$$\begin{aligned}\mathcal{L} = &\frac{1}{2}Tr[-2(\mathbf{I}^R \odot \mathbf{R})\mathbf{V}'\mathbf{U} + (\mathbf{I}^R \odot (\mathbf{U}'\mathbf{V}))\mathbf{V}'\mathbf{U}] \\ &+ \frac{\lambda_u}{2}Tr(\mathbf{U}\mathbf{U}') + \frac{\lambda_v}{2}Tr(\mathbf{V}\mathbf{V}') + \lambda_g Tr(\mathbf{U}(\mathbf{D}_1 + \mathbf{D}_2 - 2\mathbf{Q})\mathbf{U}')\end{aligned} \tag{7}$$

Then, we have the following derivatives

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{U}} &= -\mathbf{V}(\mathbf{I}^R \odot \mathbf{R})' + \mathbf{V}(\mathbf{I}^R \odot (\mathbf{U}'\mathbf{V}))' + \lambda_u \mathbf{U} \\ &\qquad + \lambda_g(\mathbf{U}(\mathbf{D}_1 + \mathbf{D}_2 - 2\mathbf{Q})' + \mathbf{U}(\mathbf{D}_1 + \mathbf{D}_2 - 2\mathbf{Q})) \\ \frac{\partial \mathcal{L}}{\partial \mathbf{V}} &= -\mathbf{U}(\mathbf{I}^R \odot \mathbf{R}) + \mathbf{U}(\mathbf{I}^R \odot (\mathbf{U}'\mathbf{V})) + \lambda_v \mathbf{V}\end{aligned} \tag{8}$$

Finally, we summarize the overall algorithm to solve the problem defined in Eq. (2) as in Alg. 2. As shown in the algorithm, we first compute the inferred trust from observed trust (step 1) and then initialize the $\mathbf{U}$ and $\mathbf{V}$ matrices (step 2). Next, we learn the coefficients to control the weights of each trust regularization terms (step 3), and define the $\mathbf{Q}$ matrix (step 4). After that, the algorithm begins the iteration procedure. In each iteration, we alternatively update the user latent matrix $\mathbf{U}$ and the item latent matrix $\mathbf{V}$ with gradient descent method (step 6 and 7). The iteration procedure will continue until at least one of the following conditions is satisfied: either the Frobenius norm between successive estimates of both matrices $\mathbf{U}$ and $\mathbf{V}$ is below a certain threshold or the maximum iteration step is reached.

3 Experiments

In this section, we present the experimental results. The experiments are designed to answer the following questions:

- How accurate is the proposed method for recommendation?
- How does the proposed method perform for cold-start users?
- How robust is the proposed method?

3.1 Experimental Setup

In our experiment, we use two datasets: Epinions and Ciao [18]. These two datasets contain not only the user-item ratings but also the trust relationships between users. For the trust relationships, we assign $\mathbf{T}(u, v) = 1$ if user u trusts user v and $\mathbf{T}(u, v) = 0$ otherwise. Table 2 shows the statistics of the datasets.

Table 2. The statistics of Epinions and Ciao datasets

Data	Epinions	Ciao
users	33,725	6,102
items	43,542	12,082
ratings	684,371	149,530
trust relationships	405,047	98,856
sparsity of ratings	0.0466%	0.2028%
sparsity of trust relationships	0.0356%	0.2655%

For both datasets, we randomly select a set of ratings as the training set and use the rest as test set. We conduct two groups of experiments with different percentage of training set (50% and 90%). We fix the rank $r = 10$, set $\lambda_g = 0.1$, and $\lambda_u = \lambda_v = 0.1$ unless otherwise stated. We use cross-validation to learn the coefficients W on the training set.

For the evaluation metrics, we adopt the Root Mean Square Error ($RMSE$) and the Mean Absolute Error (MAE) to compare with the existing trust-based recommendation methods:

$$RMSE = \sqrt{\frac{1}{|\mathbf{R}_T|} \sum_{(u,i) \in \mathbf{R}_T} (\mathbf{R}_T(u, i) - \hat{\mathbf{R}}_T(u, i))^2}$$

$$MAE = \frac{1}{|\mathbf{R}_T|} \sum_{(u,i) \in \mathbf{R}_T} |\mathbf{R}_T(u, i) - \hat{\mathbf{R}}_T(u, i)|$$

where $\mathbf{R}_T$ is the test set, $\mathbf{R}_T(u, i)$ denotes the actual rating that user u gives to item i, $\hat{\mathbf{R}}_T(u, i)$ denotes the predicted rating from user u to item i, and $|\mathbf{R}_T|$ is the total number of ratings in the test set. Lower values of $RMSE$ and MAE mean better prediction performance.

The compared methods include RSTE [7], SoRec [8], SocialMF [4], FIP [20], and SR [9]. Notice that all these compared methods use the observed trust only.

Table 3. Effectiveness comparisons. Our RIT method outperforms all the compared methods.

Data	Training	Metrics	RSTE	SoRec	SocialMF	FIP	SR	Basic	RIT
Epinions	50%	*RMSE*	1.1975	1.2082	1.1905	1.1880	1.1897	1.1732	**1.1467**
		MAE	0.9548	0.9655	0.9484	0.9452	0.9469	0.9325	**0.9134**
	90%	*RMSE*	1.1154	1.1202	1.1140	1.1111	1.1137	1.1023	**1.0872**
		MAE	0.8768	0.8824	0.8748	0.8713	0.8739	0.8648	**0.8528**
Ciao	50%	*RMSE*	1.2022	1.2392	1.2051	1.2067	1.2054	1.1827	**1.1469**
		MAE	0.9180	0.9580	0.9215	0.9216	0.9213	0.9017	**0.8696**
	90%	*RMSE*	1.0899	1.1108	1.0889	1.0947	1.0908	1.0769	**1.0561**
		MAE	0.8157	0.8415	0.8159	0.8203	0.8166	0.8056	**0.7852**

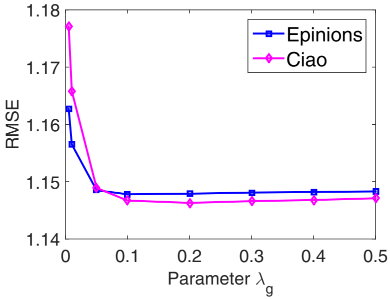
Table 4. Effectiveness comparisons for cold-start users. Our RIT method can achieve even greater improvement for cold-start users.

Data	Metrics	RSTE	SoRec	SocialMF	FIP	SR	Basic	RIT
Epinions	*RMSE*	1.4315	1.4085	1.4073	1.4007	1.4012	1.3905	**1.2857**
	MAE	1.2124	1.1789	1.1876	1.1778	1.1791	1.1681	**1.0523**
Ciao	*RMSE*	1.4781	1.4771	1.4784	1.4904	1.4892	1.4420	**1.2398**
	MAE	1.2079	1.1983	1.2058	1.2141	1.2114	1.1635	**0.9489**

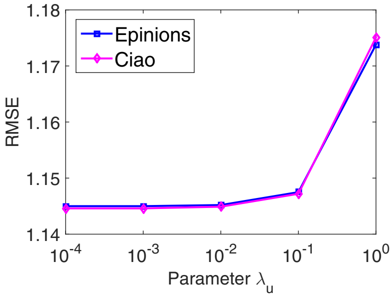
3.2 Experimental Results

Effectiveness Comparisons. We first compare the effectiveness of different methods in Table 3. In the table, we also report the results of the Basic case of our model where only the observed trust matrix is used. The Basic method is similar to the SR method except that it uses the trust as side information while SR uses the rating similarity. As we can see from the table, the proposed RIT method outperforms all the other trust-based recommendation methods wrt both *RMSE* and *MAE* on both datasets. For example, on the Epinions data with 50% training data, our method improves the best competitor (FIP) by 3.48% wrt *RMSE* and by 3.36% wrt *MAE*; for the Ciao data with 50% training data, our method improves the best competitor (RSTE) by 4.60% wrt *RMSE* and by 5.27% wrt *MAE*. Overall, the results indicate that the inferred trust is indeed helpful to improve the performance of recommender systems.

Effectiveness for Cold-Start Users. The cold-start problem, which aims to recommend items to cold-start users, is still a remaining challenge in recommender systems. Intuitively, the proposed RIT model can better handle the cold-start problem as we infer many unobserved trust relationships. We conduct experiments to validate this intuition, and define users who have expressed less than 5 ratings as cold-start users. The results with 50% training data are shown in Table 4. As we can see, our RIT method achieves larger improvement in the cold-start scenario. For example, on the Epinions data, our method improves the best competitor (FIP) by 8.21% wrt *RMSE* and by 10.66% wrt *MAE*; on



(a) Influence of Parameter λ_g



(b) Influence of Parameter λ_u

Fig. 3. Influence of parameters. Our method is robust wrt the parameters.

the Ciao data, our method improves the best competitor (SoRec) by 16.07% wrt *RMSE* and by 20.81% wrt *MAE*. This result indicates that our RIT model can better handle the cold-start problem by inferring unobserved trust.

Influence of Parameters. Finally, we study the influence of the parameters in our RIT model: λ_g which controls the importance of trust relationships, and λ_u and λ_v which control regularization. The results are shown in Fig. 3 where we use 50% training data and only report *RMSE* on y-axis for simplicity. Similar results can be observed for 90% training data and the *MAE* metric. For convenience, we set $\lambda_u = \lambda_v$. As we can see from Fig. 3(a), on both datasets, *RMSE* decreases quickly and then stays stable as λ_g increases. For λ_u and λ_v, as we can see from Fig. 3(b), *RMSE* is stable when they are less than a certain threshold (e.g., 0.1) on both datasets. Overall, the proposed RIT model is robust wrt the parameters.

4 Related Work

In this section, we review some existing approaches for recommender systems, including traditional recommendation, trust-based recommendation, and the cold-start recommendation.

Collaborative filtering is widely used in many traditional recommender systems. These traditional recommender systems mainly focus on the mining of user preferences via the user-item rating matrix. Collaborative filtering can be divided into two categories: memory-based approaches and model-based approaches. Memory-based approaches aim to find similar users or items for recommendation, known as user-based methods [1] and item-based methods [15]. Some researchers also propose to combine user-based methods and item-based methods [19]. Model-based approaches focus on learning models from data with statistical and machine learning techniques, in which matrix factorization has been widely adopted. These matrix factorization methods [6,12,14] make predictions by factorizing the user-item rating matrix into low-rank latent matrices.

Later, researchers begin to take the trust relationships between users into consideration. The basic intuition is that users may prefer to accept the recommendations by their trusted people. With this intuition, many trust-based recommender systems have been proposed [4,7–9,20,21]. For example. Ma et al. [7] develop a model which combines the tastes of users and their trusted people together for the rating prediction; Jamali and Ester [4] propose that users' preferences are dependent on the preferences of their trusted people ; Ma et al. [8] propose a model in which the trust matrix and the user-item rating matrix are connected with shared user latent preferences; Yang et al. [20] aim to use the inner product between two users' latent preferences to recover their social link; Ma et al. [9] introduce the social regularization term which serves as a constraint to regularize the user latent preferences; Yao et al. [21] consider both the relationships between users and the relationships from items. All the above trust-based methods employ the observed trust for better recommendation. Different from them, we also consider the inferred trust which might be useful for recommendation. In the recent work by Fazeli et al [2], they try to infer implicit trust into trust-based recommendation methods when explicit trust relationships are not available. In contrast, we infer the implicit trust from explicit trust and combine them together to improve the prediction accuracy of recommender systems.

One of the remaining challenges of recommender systems is the cold-start problem, i.e., how to recommend items for users that have expressed few ratings. Existing solutions can be roughly classified into two classes: interview based and side-information based. The interview based methods add an interview process to collect the preferences of cold-start users in their sign-up phase [17,24]; the main problem of this type of methods is the additional burdens. The second type of methods incorporate the side information to enhance prediction accuracy for cold-start users, such as the content information of items [16], the attributes of users/items [13,23], or the social relationships between users (e.g., trust-based recommendation). In this work, we enhance the trust-based recommendation by inferring unobserved trust from observed trust.

5 Conclusion

In this paper, we have proposed a trust-based recommendation model RIT based on matrix factorization and social regularization. In RIT, we first infer several types of unobserved trust based on the observed trust relationships by employing the transitivity property of trust. Next, we propose to incorporate both observed and inferred trust relationships for better recommendation. Experimental evaluations on two real datasets show that our method outperforms the existing trust-based recommendation models in terms of prediction accuracy, and greater improvement is observed for the recommendation of cold-start users.

Acknowledgments. This work is supported by the National 973 Program of China (No. 2015CB352202) and the National Natural Science Foundation of China (No. 91318301, 61321491).

References

1. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI, pp. 43–52 (1998)
2. Fazeli, S., Drachsler, H., Loni, B., Sloep, P., Bellogin, A.: Implicit vs. explicit trust in social matrix factorization. In: RecSys, pp. 317–320 (2014)
3. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: WWW, pp. 403–412 (2004)
4. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys, pp. 135–142 (2010)
5. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
7. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: SIGIR, pp. 203–210 (2009)
8. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: CIKM, pp. 931–940 (2008)
9. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296 (2011)
10. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual Review of Sociology, 415–444 (2001)
11. Mitchell, M.: An introduction to genetic algorithms. MIT Press (1998)
12. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264 (2007)
13. Park, S.T., Chu, W.: Pairwise preference regression for cold-start recommendation. In: RecSys, pp. 21–28 (2009)
14. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: ICML, pp. 880–887 (2008)
15. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295 (2001)
16. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: SIGIR, pp. 253–260 (2002)
17. Sun, M., Li, F., Lee, J., Zhou, K., Lebanon, G., Zha, H.: Learning multiple-question decision trees for cold-start recommendation. In: WSDM, pp. 445–454 (2013)
18. Tang, J., Gao, H., Liu, H.: mtrust: discerning multi-faceted trust in a connected world. In: WSDM, pp. 93–102 (2012)
19. Wang, J., De Vries, A.P., Reinders, M.J.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: SIGIR, pp. 501–508 (2006)
20. Yang, S.H., Long, B., Smola, A., Sadagopan, N., Zheng, Z., Zha, H.: Like like alike: joint friendship and interest propagation in social networks. In: WWW, pp. 537–546 (2011)
21. Yao, Y., Tong, H., Yan, G., Xu, F., Zhang, X., Szymanski, B.K., Lu, J.: Dual-regularized one-class collaborative filtering. In: CIKM, pp. 759–768 (2014)
22. Yao, Y., Tong, H., Yan, X., Xu, F., Lu, J.: Matri: a multi-aspect and transitive trust inference model. In: WWW, pp. 1467–1476 (2013)
23. Zhang, M., Tang, J., Zhang, X., Xue, X.: Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In: SIGIR (2014)
24. Zhou, K., Yang, S.H., Zha, H.: Functional matrix factorizations for cold-start recommendation. In: SIGIR, pp. 315–324 (2011)