Social Collaborative Filtering by Trust

Bo Yang **Yu Lei** **Dayou Liu**
School of Computer Science and Technology
Jilin University, China
{ybo, liudy}@jlu.edu.cn, leiyu11@mails.jlu.edu.cn

Jiming Liu
Department of Computer Science
Hong Kong Baptist University, Hong Kong
jiming@comp.hkbu.edu.hk

Abstract

To accurately and actively provide users with their potentially interested information or services is the main task of a recommender system. Collaborative filtering is one of the most widely adopted recommender algorithms, whereas it is suffering the issues of data sparsity and cold start that will severely degrade quality of recommendations. To address such issues, this article proposes a novel method, trying to improve the performance of collaborative filtering recommendation by means of elaborately integrating twofold sparse information, the conventional rating data given by users and the social trust network among the same users. It is a model-based method adopting matrix factorization technique to map users into low-dimensional latent feature spaces in terms of their trust relationship, aiming to reflect users' reciprocal influence on their own opinions more reasonably. The validations against a real-world dataset show that the proposed method performs much better than state-of-the-art recommendation algorithms for social collaborative filtering by trust.

1 Introduction

In the literature, great efforts have been made to address the task of personalized recommendation. Among existing techniques, collaborative filtering (CF) is relatively simple and effective which has been widely used by many commercial web sites. However, the existing CF confronts two main challenges: data sparsity and cold start, which will greatly degenerate its performance. By data sparsity, it is meant the available rating data are usually very sparse. Due to cold start, CF is hardly able to give satisfactory recommendations to those users who rarely rated items.

One potential way to solve these problems is by exploring available social networks. With the rapid development of web 2.0 technologies, in addition to the ratings of items contributed by users, the social information of users get much more readily obtained than before through social networking services. It is believed that human beings usually acquire and disseminate information through their acquaintances such as friends, colleagues or partners [Resnick and Varian, 1997; Granovetter, 1973; Kautz *et al*., 1997], which implies that the underlying social networks of users might play a fundamental role in helping them filter information. Trust relationship is one of the most important types of social information in that we are more likely to accept viewpoints from whom we trust [Sinha and Swearingen, 2001; Ziegler and Lausen, 2004]. Thus, it has become a big opportunity as well as a big challenge to improve recommendation quality by sufficiently and effectively utilizing available trust information. Recently, several model-based methods have been proposed for social recommendation [Ma *et al*., 2011; Jamali and Ester, 2010; Ma *et al*., 2009; Ma *et al*., 2008]. Most of them are based on the technique of fast matrix factorization, which firstly map users and items into low-dimensional feature spaces simultaneously, and then train a prediction model by optimizing some objective functions over rating and trust data.

Distinct from above model-based methods and memory-based methods such as [Yang *et al*., 2012], this work studies a new strategy of fusing rating and trust data by sufficiently exploring how the observed ratings are generated under the influence of trust behaviors of users entangled by their trust network, rather than just simply fitting two kinds of data as most of the existing studies do. Our work is motivated by the observation that users are caught in their social network of mutuality, and whatever affects one directly also affects all indirectly. More specifically, when a user is rating, he/she will be more likely affected by the existing ratings or reviews provided by others he/she trusts, and in the same way, his/her contributions (ratings or reviews) will consequently have influence on the decisions of others who trust him/her. In modeling this, we propose a simple but effective way to map users into two low-dimensional spaces, i.e. truster space and trustee space, by factorizing trust network according to the directional properties of trust. The vectors of truster and trustee in two spaces describe "to trust by reading ratings or reviews" and "to be trusted by generating ratings or reviews" behaviors of a user, respectively. Suppose user A trusts user B with strength w, then w can be represented as the inner product of A's truster vector by B's trustee vector. Furthermore, the two spaces will be used in tandem with the user space and item space obtained by factorizing rating matrix to construct a novel fusing model, named TrustMF, to fit both rating and trust data.

We validate the performance of TrustMF and compare it with representative methods against one real dataset with big sparsity. Results show that TrustMF performs much better, especially in the case of cold start, than the state-of-the-art CF methods based on both ratings and trust.

2 TrustMF: A Model-based Method

2.1 Problem Description

Suppose we have a recommender system which involves m users and n items. Let $R = [R_{ij}]_{m\times n}$ denotes the user-item rating matrix, where each element R_{ij} represents the rating of item j given by user i, which is an integer number from 1 to 5. Let $\Omega = \{(i, j): R_{ij} \neq 0\}$ denotes the locations of observed ratings in rating matrix. One wants to map users and items to a low-dimensional feature space. Let a d-dimension vector U_i be the user-specific latent feature vector of user i and a d-dimension vector V_j be the item-specific latent feature vector of item j. Then we have the user feature matrix $U \in \mathbb{R}^{d\times m}$ and the item feature matrix $V \in \mathbb{R}^{d\times n}$. By employing low-rank approximation over the observed rating matrix, one can recover the unknown ratings by U^TV. The feature matrix U and V can be learned by minimizing a loss function as follows:

$$\mathcal{L} = \sum_{(i,j)\in\Omega}\left(U_i^T V_j - R_{ij}\right)^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) \quad (1)$$

where $\|\cdot\|_F^2$ denote the Frobenius norm, and λ is a parameter which controls the model complexity to avoid over-fitting.

To further avoid over-fitting when learning parameters, a so-called weighted-λ-regularization was introduced to above model [Zhou *et al*, 2008]. It increases the penalization to the feature vectors of users or items involving more ratings. Correspondingly, new objective is obtained as follows:

$$\mathcal{L} = \sum_{(i,j)\in\Omega}\left(U_i^T V_j - R_{ij}\right)^2 + \lambda\left(\sum_i n_{u_i}\|U_i\|_F^2 + \sum_j n_{v_j}\left\|V_j\right\|_F^2\right) \quad (2)$$

where n_{u_i} and n_{v_j} denote the numbers of ratings given by user i and given to item j, respectively.

2.2 Matrix Factorization of the Trust Network

Suppose we have a trust network N with m nodes, where nodes represent users and arcs represent directional trust relation among users. One can use adjacent matrix $T = [T_{ik}]_{m\times m}$ to describe the structure of N. T_{ik} is a real number within the interval [0, 1], where "0" indicates user i distrusts k while "1" indicates user i trusts k heavily. Since user i trusts user k do not always means k trusts i in the same way, the matrix T is not symmetric. Let $\Psi = \{(i, k): T_{ik} \neq 0\}$ denote the locations of observed trust relations in trust network. Due to dissymmetrical property of trust, we map each user i of trust network as two distinct latent feature vectors, depicted by truster-specific feature vector B_i and trustee-specific feature vector W_i, respectively. B_i and W_i characterize the behaviors of "to trust others" and "to be trusted by others", respectively. Give such vectors, one can model trust value T_{ik} as the inner product of B_i and W_k. It is important to notify that, this way of modeling trust value is totally distinct from existing methods, most of which model trust value from purely mathematical perspectives like [Ma *et al*., 2008] without explicitly exploring the real-world implications of feature vectors. We will see later this new way of modeling trust value not only characterizes the directional property of trust relationship in a more proper way, but also gives a better interpretation on how the mutual trust among users are generated and how they affect users' respective opinions.

Provided we have only trust data, the feature matrices $B \in \mathbb{R}^{d\times m}$ and $W \in \mathbb{R}^{d\times m}$ can be learned by minimizing the following objective function:

$$\mathcal{L} = \sum_{(i,k)\in\Psi}\left(B_i^T W_k - T_{ik}\right)^2 + \lambda(\|B\|_F^2 + \|W\|_F^2) \quad (3)$$

2.3 The TrustMF Algorithm

We have performed matrix factorization just based on either rating data or trust data. In this section, we will present our matrix factorization model to fuse both of them.

Truster Model

Users of social networking such as Epinions enable to browse and generate opinions (ratings or reviews) on items they are interested and then build their respective turst nets based on such opinions. Through tangled trust ties, the opinions of an individual will be affected by others and vise verse. Here we first propose a truster model to characterize first aspect, that is, how others will affect a specific user's opinions.

Note that the m users getting involved in rating matrix R and trust matrix T are the same. So, one can associate R and T into one matrix factorization process by sharing a common user-specific latent space. Fig. 1 shows the proposed truster model that is able to characterize how a user A's ratings are affected by other users he trusts by means of $B_A{}^TV_j$.

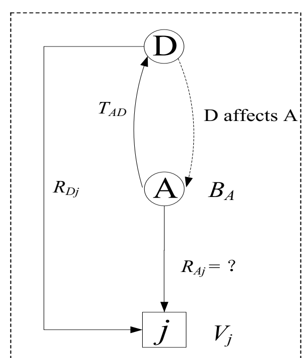

Figure 1: Truster model B^TV: How others affect user A's rating on item j.

In this model, we choose the truster-specific feature matrix B as the latent space commonly shared by R and T. Vector B_A simultaneously characterizes twofold meanings: How a user A trusts (or is affected by) others and how the same user rates items. The item-specific latent feature vector V_j depicts how an item j is rated by users. Putting together, the $B_A{}^TV_j$ indicates how other users affect user A to rate item j, which is the approximation of real score R_{Aj}. Therefore, one can learn the feature matrics B, V and W simultaneously by minimizing the following objective function:

$$\mathcal{L} = \sum_{(i,j)\in\Omega}\left(B_i^T V_j - R_{ij}\right)^2 + \lambda_T \sum_{(i,k)\in\Psi}\left(B_i^T W_k - T_{ik}\right)^2 + \lambda(\|B\|_F^2 + \|V\|_F^2 + \|W\|_F^2) \quad (4)$$

where the parameter λ_T controls the proportion of effect between rating preference and trust relation when training model. In this way, one can integrate two kinds of data sources and hence obtain latent spaces B and V that can work together to produce a more accurate prediction. In our validations, we set λ_T =1.

Since the trust value T_{ik} is between 0 and 1, in order to learn parameters in a more convenient way, we map raw rating R_{ij} into an interval [0, 1] by employing the function $f(x) = x/R_{max}$ .Where R_{max} (equal to 5 in this work) is the maximum of ratings in a recommender system. In order to fit data more conveniently, we adopt a logistic function $g(x) = 1/(1+\exp(-x))$ suggested by [Salakhutdinov and Mnih, 2007] to bound the inner product of latent feature vectors into the interval [0, 1]. One can get the prediction by $g(B_i^T V_j) \cdot R_{max}$ after model training. Moreover, we incorporate a similar regularization approach derived from [Zhou *et al*., 2008] into our model. In summary, the objective to be optimized in the proposed truster model is as follows:

$$\mathcal{L} = \sum_{(i,j)\in\Omega}\left(g(B_i^T V_j) - R_{ij}\right)^2 + \lambda_T \sum_{(i,k)\in\Psi}\left(g(B_i^T W_k) - T_{ik}\right)^2 + \lambda\left(\sum_i (n_{b_i} + m_{b_i})\|B_i\|_F^2 + \sum_j n_{v_j}\|V_j\|_F^2 + \sum_k m_{w_k}\|W_k\|_F^2\right) \quad (5)$$

where n_{b_i} and n_{v_j} denote the numbers of ratings given by user i and given to item j, respectively, m_{b_i} denotes the number of users who are trusted by user i and m_{w_k} denotes the number of users who trust user k. One can minimize the above objective function by performing the following gradient descents on B_i, V_j and W_k for all users and items.

$$\frac{1}{2}\cdot\frac{\partial\mathcal{L}}{\partial B_i} = \sum_{j\in\mathcal{R}(i)} g'(B_i^T V_j)(g(B_i^T V_j) - R_{ij})V_j + \lambda_T \sum_{k\in\mathcal{F}(i)} g'(B_i^T W_k)(g(B_i^T W_k) - T_{ik})\,W_k + \lambda(n_{b_i} + m_{b_i})B_i\,,$$

$$\frac{1}{2}\cdot\frac{\partial\mathcal{L}}{\partial V_j} = \sum_{i\in\mathcal{R}^+(j)} g'(B_i^T V_j)(g(B_i^T V_j) - R_{ij})B_i + \lambda n_{v_j} V_j\,,$$

$$\frac{1}{2}\cdot\frac{\partial\mathcal{L}}{\partial W_k} = \lambda_T \sum_{i\in\mathcal{F}^+(k)} g'(B_i^T W_k)(g(B_i^T W_k) - T_{ik})B_i + \lambda m_{w_k} W_k\,,$$

where $\mathcal{R}(i)$ denotes the set of items which user i has rated, $\mathcal{R}^+(j)$ denotes the set of users who have rated item j, $\mathcal{F}(i)$ denotes the set of users who are trusted by user i and $\mathcal{F}^+(k)$ denotes the set of users who trust user k. $g'(x) = \exp(-x)/(1+\exp(-x))^2$ is the derivative of logistic function $g(x)$.

As the model in Equation (5) uses truster-specific feature matrix B as the commonly shared user latent space, we refer to the learning algorithm as Truster-MF, and its pseudo-code is given in Table 1.

Algorithm 1 Truster-MF

Input: R, T, d, λ, λ_T, α (learning rate)

Initialize B, V, W with small random numbers

while $\mathcal{L}$ has not converged:

$\quad \boldsymbol{V_j \leftarrow V_j - \alpha \cdot \frac{\partial\mathcal{L}}{\partial V_j}, j = 1, \dots, n}$

$\quad \boldsymbol{B_i \leftarrow B_i - \alpha \cdot \frac{\partial\mathcal{L}}{\partial B_i}, i = 1, \dots, m}$

$\quad \boldsymbol{W_k \leftarrow W_k - \alpha \cdot \frac{\partial\mathcal{L}}{\partial W_k}, k = 1, \dots, m}$

end while

Output the predicted rating: $\boldsymbol{\hat{R}_{ij} \leftarrow g(B_i^T V_j) \cdot R_{max}}$

Table 1: The learning algorithm of truster model

Time Complexity Analysis

The time of learning a truster model is mainly taken by computing object function $\mathcal{L}$ and its gradients against feature vectors. The time of computing $\mathcal{L}$ is $O(td(|\Omega|+|\Psi|))$, where t is the required iterations, d is the dimensionality of feature vectors, $|\Omega|$ and $|\Psi|$ are the numbers of observed ratings and observed trust links, respectively. The costs of computing the gradients of $\mathcal{L}$ against B, V, W are $O(td(|\Omega|+|\Psi|))$, $O(td|\Omega|)$ and $O(td|\Psi|)$, respectively. Therefore, the time complexity of Truster-MF is $O(td(|\Omega|+|\Psi|))$, linearly scaling to the numbers of observed ratings and trust links.

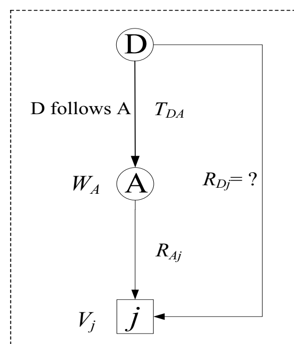

Figure 2: Trustee model $W^T V$: How others follow user A's rating on item j.

Trustee Model

Fig. 2 shows the proposed trustee model that is able to characterize how a user A's opinions affect the decisions of others who trust A by means of $W_A^T V_j$. Distinct from truster model, this time we choose the trustee-specific feature matrix W as the latent space commonly shared by R and T. In trustee model, vector W_A simultaneously characterizes twofold meanings: how a user A is trusted (or followed) by others and how the same user rates items. Again, the item-specific latent feature vector V_j depicts how an item j is rated by users. Putting together, the $W_A^T V_j$ indicates how other users follow user A to rate item j, which is the approximation of real score R_{ij} as well.

Similarly, one can learn the feature matrices W, V and B simultaneously by minimizing the following objective:

$$\mathcal{L} = \sum_{(i,j)\in\Omega} \left(g(W_i^T V_j) - R_{ij}\right)^2 + \lambda_T \sum_{(k,i)\in\Psi} \left(g(B_k^T W_i) - T_{ki}\right)^2 + \lambda \left(\sum_i (n_{w_i} + m_{w_i}) \|W_i\|_F^2 + \sum_j n_{v_j} \|V_j\|_F^2 + \sum_k m_{b_k} \|B_k\|_F^2\right) \quad (6)$$

where n_{w_i} and n_{v_j} denote the numbers of ratings given by user i and given to item j, respectively, m_{w_i} denotes the number of users who trust user i and m_{b_k} denotes the number of users who are trusted by user k.

As the model in Equation (6) uses trustee-specific feature matrix W as the commonly shared user latent space, we refer to corresponding learning algorithm as Trustee-MF.

2.4 Synthetic Influence of Trust Propagation

Individuals will affect each other during the process of rating. This point is emphasized in our work. That is to say, your decision on whether or not to rate something and how to rate them will be based on the opinions of your trustees. Meanwhile, your decisions will inevitably influence the choices of your trusters. Comprehensively, it is suggested that the observed ratings are actually generated according to the propagation of such twofold influence among users.

After independently training truster model and trustee model, one can obtain two sets of feature matrices. Let B_i^r and ${V_j}^r$ be the truster-specific vector and item-specific vector learned by algorithm Truster-MF. Let W_i^e, ${V_j}^e$ be the trustee-specific vector and item-specific vector learned by algorithm Trustee-MF. We suggest following synthetic strategy, named as TrustMF, to generate the approximation of real rating scores.

$$\hat{R}_{ij} = g\left(\left(\frac{B_i^r + W_i^e}{2}\right)^T \left(\frac{{V_j}^r + {V_j}^e}{2}\right)\right) \cdot R_{max} \quad (7)$$

2.5 Insights into the Truster and Trustee Models

The key idea behind the proposed truster and trustee models is to construct a bridge between ratings and trust by mapping them into the same d-dimension latent space, in which each dimention characterizes one of d latent features of items. Then we have, V_j (item-specific feature vector of item j), U_i (user-specific feature vector of user i), B_i (truster-specific feature vector of user i) and W_i (trustee-specific feature vector of user i) depict "what type item j is", "what types of items user i prefers", "what types of opinions (ratings or reviews) user i prefers to browse" and "what types of opinions user i prefers to generate" in terms of the same d latent features, respectively. Actually, browing/generating opinions exactly model how users follow/affect others. Note that, one tends to give high ratings to his preferred items and trust those whose opinions are relevant and helpful to him. Reasonably, observed ratings and trust can be approximated by measuring the coincidence of respective feature vectors in terms of inner products, i.e. $R_{ij} \approx U_i^T V_j$ and $T_{ik} \approx B_i^T W_k$.

We enable the system to learn user preference from their social activities. Specificially, truster model learns user preference from their browsing behaviors, in which U_i is approximated by B_i, implying that one will pay more attentions to the opinions about items he prefers. On the other hand, trustee model learns user preference from their writtings or ratings, in which U_i is approximated by W_i, implying that one will more likely generate opinions about items he prefers. As a result, users are encouraged to interact with the system, more social activities getting more accurate recommendations for them. The truster and trustee models favor those users having more browing and writing activities, respectively. While, TrustMF tries to provide high-quality recommendations for both kinds of users.

3 Experiments and Validations

3.1 Description of the Dataset

The data used in our experiments is Epinions, containing both ratings and trust relations as a benchmark for validating trust-based recommender methods, which is taken from the web site www.trustlet.org/wiki/Epinions. We use it to test and compare the performance of TrustMF with up-to-date model-based methods. This dataset is extremely sparse and imbalanced containing 49,289 users and 139,738 items. The total number of ratings is 664,823, and the number of trust relations between users is 487,183. Its density is 0.0097% and 0.0201% in terms of ratings and trust relations, respectively. The statistics of it is presented in Table 2.

3.2 Experimental Setup

Cross-validation

We use a 5-fold cross-validation for learning and testing. In each time we randomly select 80% of data as training set and the rest of 20% for test. For each experiment discussed below, we conduct five times and take the mean as the final result.

Statistics of Rating Matrix	**User**	**Item**
Avg. Num. of Ratings	13.49	4.76
Max. Num. of Ratings	1023	2026
Min. Num. of Ratings	0	1
Statistics of Trust Network	**Out-degree**	**In-degree**
Average	9.88	9.88
Maximum	1760	2589
Minimum	0	0

Table 2: Statistics of Epinions

Evaluate Metrics

The evaluation metrics used in our experiments are mean absolute error (MAE) and root mean square error (RMSE) which are defined as:

$$\text{MAE} = \frac{\sum_{i,j}\left|R_{ij} - \hat{R}_{ij}\right|}{N}, \qquad \text{RMSE} = \sqrt{\frac{\sum_{i,j}(R_{ij} - \hat{R}_{ij})^2}{N}}$$

where R_{ij} is the rating in the test set, $\hat{R}_{ij}$ is the predicted rating, and N is the number of ratings in the test set.

Comparison Methods

To comparatively evaluate the performance of our proposed methods, we select seven representative CF methods as competitors, which are given in Table 3.

Methods	Rationale
UserMean	mean rating of users
ItemMean	mean rating of items
PMF[Salakhutdinov and Mnih, 2007]	matrix factorization based upon user-item ratings
SoRec [Ma et al., 2008]	matrix factorization based upon both user-item ratings and trust relations
RSTE [Ma et al., 2009]	
SocialMF [Jamali and Ester, 2010]	
$SR2_{PCC}$ [Ma et al., 2011]	
TrustMF	

Table 3: Representative methods to be compared

Methods	Optimal Parameters
PMF	λ_U=λ_V=0.001
SoRec	λ_U=λ_V=λ_Z=0.001, λ_C=1
RSTE	λ_U=λ_V=0.001, α=0.4
SocialMF	λ_U=λ_V=0.001, λ_T=1
$SR2_{PCC}$	λ_1=λ_2=0.001, β=0.001
TrustMF	λ=0.001, λ_T=1

Tabel 4: Parameter settings of respective methods

For all methods to be validated, we set respective optimal parameters either according to corresponding references or based on our experiments. The main parameters of respective methods are given in Table 4. To focus on verifying models and fairly comparing them, for all matrix factorization based methods, we set the dimensionality of latent space as 5 and 10, respectively, and adopte the same initialization strategy, which randomly initialize all involved feature matrices with a uniform distribution within the interval [0, 1].

3.3 Experimental Results

Validation on Epinions

We now validate the performance of TrustMF and compare it with its competitors on Epinions. Table 5 shows that TrustMF performs the best of all. The improvements of TrustMF against respective competitors given in the table imply that our proposed truster, trustee and synthetic models properly catch on the influential propagation of trust on the generation of observed ratings, and hence get significant promotion on the quality of recommendations.

Validation on Cold Start Users

As mentioned in introduction section, cold start is one big challenge faced by CF methods. We now evaluate the capabilities of addressing cold start users by respective competitors. Conventionally, those who have rated five or fewer ratings are seemed as cold start users [Jamali and Ester, 2010]. 5-fold cross-validation is still used in the test but we only care about the accuracy of prediction for cold start users. Table 6 shows that TrustMF performs the best once again and gets remarkable improvements against the other CF methods with respect to addressing cold start users.

To further validate model-based methods against cold start users, we take users with many ratings (between 50 and 70). For each user, remove all but 5 ratings, compute the latent space mapping based on these ratings, then evaluate the performance on removed (unseen) ratings. The motivation is, when one evaluates with users who only have 5 or fewer ratings, the performance might be less indicative. The above method would allow us to evaluate prediction accuracy on 45 or more held-out ratings (actually 60863 ratings of 1145 users). As shown in Table 7 (d=5 in this case), TrustMF ranks top once again.

	Metrics	UserMean	ItemMean	PMF	SoRec	RSTE	SocialMF	$SR2_{PCC}$	T-r	T-e	T
d=5	MAE	0.9420	0.9512	0.8440	0.9197	0.8635	0.8826	0.8450	0.8329	0.8345	**0.8212**
	Improve	12.8%	13.7%	2.7%	10.7%	4.9%	7.0%	2.8%	1.4%	1.6%	
	RMSE	1.2166	1.2363	1.0828	1.1510	1.1071	1.1107	1.0977	1.0729	1.0751	**1.0585**
	Improve	13.0%	14.4%	2.2%	8.0%	4.4%	4.7%	3.6%	1.3%	1.5%	
d=10	MAE	0.9420	0.9512	0.8382	0.9152	0.8572	0.8567	0.8671	0.8267	0.8291	**0.8148**
	Improve	13.5%	14.3%	2.8%	11.0%	4.9%	4.9%	6.0%	1.4%	1.7%	
	RMSE	1.2166	1.2363	1.1146	1.1773	1.1483	1.1113	1.1199	1.0932	1.0970	**1.0771**
	Improve	11.5%	12.9%	3.4%	8.5%	6.2%	3.1%	3.8%	1.5%	1.8%	

Table 5: Experimental results on Epinions (T-r, T-e, T refer to Truster-MF, Trustee-MF and TrustMF, respectively)

	Metrics	UserMean	ItemMean	PMF	SoRec	RSTE	SocialMF	$SR2_{PCC}$	T-r	T-e	T
d=5	MAE	1.1524	0.9312	0.9145	0.9124	0.9143	1.0954	0.9250	0.9094	0.9075	**0.8901**
	Improve	22.8%	4.4%	2.7%	2.4%	2.6%	18.7%	3.8%	2.1%	1.9%	
	RMSE	1.5054	1.2192	1.1412	1.1461	1.1439	1.3094	1.1888	1.1332	1.1308	**1.1089**
	Improve	26.3%	9.0%	2.8%	3.2%	3.1%	15.3%	6.7%	2.1%	1.9%	
d=10	MAE	1.1524	0.9312	0.8531	0.8499	0.8293	1.0296	0.9833	0.8368	0.8424	**0.8260**
	Improve	28.3%	11.3%	3.2%	2.8%	0.4%	19.8%	16.0%	1.3%	1.9%	
	RMSE	1.5054	1.2192	1.1608	1.1717	1.1621	1.2772	1.2370	1.1306	1.1369	**1.1143**
	Improve	26.0%	8.6%	4.0%	4.9%	4.1%	12.8%	9.9%	1.4%	2.0%	

Table 6: Experimental results on testing cold start users

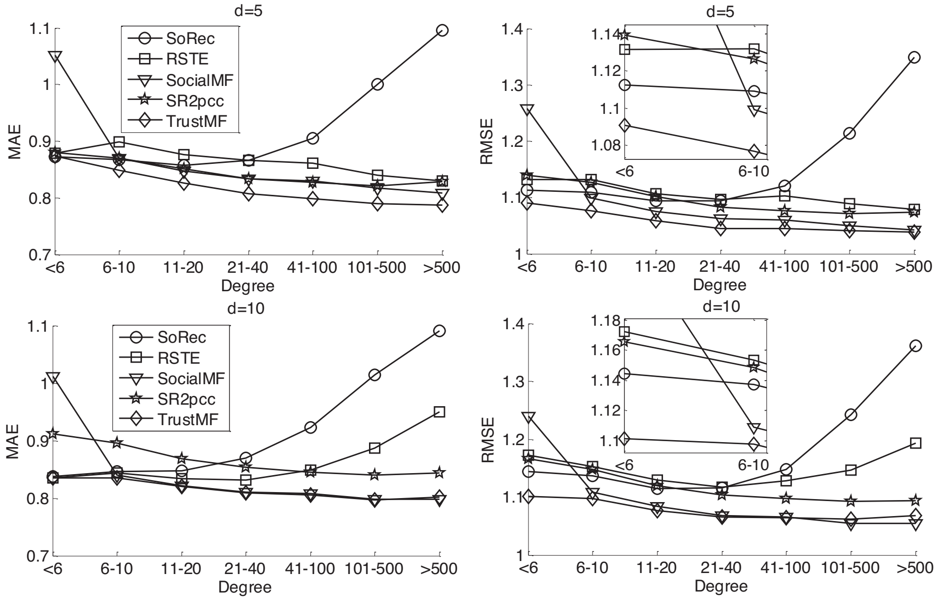

Figure 3: The predict error on users with different degrees

The above two experiments indicate that, compared with current CF methods, TrustMF has demonstraed the ability to provide high quality recommendations to cold start users.

Methods	MAE (rank)	RMSE (rank)
PMF	0.8787 (4)	1.1039 (3)
SoRec	1.0060 (6)	1.2351 (6)
RSTE	0.8763 (3)	1.1118 (4)
SocialMF	0.8684 (2)	1.0960 (2)
$SR2_{PCC}$	0.8827 (5)	1.1341 (5)
TrustMF	**0.8224 (1)**	**1.0757 (1)**

Table 7: Cold start testing by removing existing ratings

Validation on Users with Different Degrees

In previous experiments, the trust link number (outboud plus inbound) on average of cold start user is 5.39. In order to further check the capabilities of respective methods to utilize sparse trust data for recommending, we conduct a special experiment. Distinct from previous validations focusing on comparing global quality of recommendations in terms of the average accuracy over all users, here we are particularly interested in testing the performance of respective social CF methods in regard to different categories of users. We first group all users into several groups according to their degrees in trust network, and then calculate prediction error over each group, respectively.

Fig. 3 shows following observations. The performance of all five compared methods varies at different extent with respect to different groups. TrustMF performs quite stably and demonstrates the best quality for almost all groups, especially for the group with no more than 6 trust relations (62.3% of total users) and the group with 6 to 10 trust relations (12.5% of total users) in terms of RMSE. This strongly indicates TrustMF can utilize sparse trust data more effectively compared with other social CF methods.

4 Conclusions

Collaborative filtering methods are facing the difficulty of data sparsity and cold start. Aiming at addressing them by utilizing additional trust data more effectively, we have proposed a novel social CF method named TrustMF, which is motivated by the heuristic that individuals will affect each other during the process of reviewing. In properly catching on a twofold influence of trust propagation on the generation of observed opinions, a truster model and a trustee model have been proposed to map users into the same latent feature spaces but with different implications that can explicitly describe the feedback how users affect or follow the opinions of others. Moreover, the two models are naturally synthetized to one fusing model simultaneously fitting available ratings and trust ties. As has been verified that the TrustMF gets remarkable improvements against the state-of-the-art social CF methods according to the validations and comparisons on a real-world dataset. Especially, for the cold start users who have few ratings or occupy only a few trust ties, TrustMF performs significantly better than its competitors.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grants 60873149, 61133011, and 61170092, the Program for New Century Excellent Talents in University under grant NCET-11-0204, Hong Kong Research Grants Council under grant RGC/HKBU211212, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University.

References

[Granovetter, 1973] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, May 1973.

[Jamali and Ester, 2010] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 135–142, Barcelona, Spain, September 2010. Association for Computing Machinery.

[Kautz *et al*., 1997] Henry Kautz, Bart Selman and Mehul Shah. ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, March 1997.

[Ma *et al*., 2008] Hao Ma, Haixuan Yang, Michael R. Lyu and Irwin King. SoRec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 931–940, Napa Valley, California, USA, October 2008. Association for Computing Machinery.

[Ma *et al*., 2009] Hao Ma, Irwin King and Michael R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 203–210, Boston, Massachusetts, USA, July 2009. Association for Computing Machinery.

[Ma *et al*., 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 287–296, Hong Kong, China, February 2011. Association for Computing Machinery.

[Resnick and Varian, 1997] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.

[Salakhutdinov and Mnih, 2007] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 22nd Annual Conference on Advances in Neural Information Processing Systems*, pages 1257–1264, Vancouver, Canada, December 2007. Association for Computing Machinery.

[Sinha and Swearingen, 2001] Rashmi Sinha and Kirsten Swearingen. Comparing recommendations made by online systems and friends. In *Proceedings of the Joint DELOS-NSF Workshop on Personalization and Recommender systems in Digital Libraries*, pages 64–69, Dublin, Ireland, June 2001.

[Yang *et al.*, 2012] Bo Yang, Pengfei Zhao, Shuqiu Ping and Jing Huang. Improving the recommendation of collaborative filtering by fusing trust network. In *Proceedings of the 7th International Conference on Computational Intelligence and Security*, pages 195–199, Guangzhou, China, November 2012. Institute of Electrical and Electronics Engineers.

[Zhou *et al*., 2008] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber and Rong Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *Proceedings of 4th International Conference on Algorithmic Aspects in Information and Management*, pages 337–348, Shanghai, China, June 2008. Springer-Verlag.

[Ziegler and Lausen, 2004] Cai N. Ziegler and Georg Lausen. Analyzing correlation between trust and user similarity in online communities. In *Proceedings of the Second International Conference on Trust Management*, Page 251–265, Oxford, UK, March 2004. Springer-Verlag.