Guo L, Ma J, Jiang HR *et al.* Social trust aware item recommendation for implicit feedback. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(5): 1039–1053 Sept. 2015. DOI 10.1007/s11390-015-1580-8

Social Trust Aware Item Recommendation for Implicit Feedback

Lei Guo [1,2] (郭 磊), Jun Ma [2,*] (马 军), *Senior Member, CCF, Member, ACM, IEEE*
Hao-Ran Jiang [3] (姜浩然), Zhu-Min Chen [2] (陈竹敏), *Senior Member, CCF, Member, ACM*
and Chang-Ming Xing [4] (邢长明)

[1] *School of Management Science and Engineering, Shandong Normal University, Jinan 250014, China*

[2] *School of Computer Science and Technology, Shandong University, Jinan 250101, China*

[3] *Bureau of Information Technology, Shandong Post Company, Jinan 250101, China*

[4] *School of Continuing Education, Shandong University of Finance and Economics, Jinan 250101, China*

E-mail: leiguo.cs@hotmail.com; majun@sdu.edu.cn; haoranjiang@live.com; chenzhumin@sdu.edu.cn
xingchm@126.com

Received November 10, 2014; revised July 15, 2015.

Abstract Social trust aware recommender systems have been well studied in recent years. However, most of existing methods focus on the recommendation scenarios where users can provide explicit feedback to items. But in most cases, the feedback is not explicit but implicit. Moreover, most of trust aware methods assume the trust relationships among users are single and homogeneous, whereas trust as a social concept is intrinsically multi-faceted and heterogeneous. Simply exploiting the raw values of trust relations cannot get satisfactory results. Based on the above observations, we propose to learn a trust aware personalized ranking method with multi-faceted trust relations for implicit feedback. Specifically, we first introduce the social trust assumption — a user's taste is close to the neighbors he/she trusts — into the Bayesian Personalized Ranking model. To explore the impact of users' multi-faceted trust relations, we further propose a category-sensitive random walk method CRWR to infer the true trust value on each trust link. Finally, we arrive at our trust strength aware item recommendation method SocialBPR$_{\mathrm{CRWR}}$ by replacing the raw binary trust matrix with the derived real-valued trust strength. Data analysis and experimental results on two real-world datasets demonstrate the existence of social trust influence and the effectiveness of our social based ranking method SocialBPR$_{\mathrm{CRWR}}$ in terms of AUC (area under the receiver operating characteristic curve).

Keywords social recommendation, matrix factorization, random walk, Bayesian personalized ranking

1 Introduction

With the exponential growth of information generated on the World Wide Web, recommender systems as one of the efficient information filtering techniques have attracted a lot of attentions in the last decade. Recommender systems focus on solving the information overload problem by suggesting the items that are potential of their interests to users. Typical recommender systems are based on collaborative filtering, which is a technique that can predict the preference of a given user by only collecting rating information from other similar users or items[1-2]. Examples of successful applications of recommender systems can be found in many industries, such as movie recommendation at Netflix① and product recommendation at Amazon②.

Regular Paper

Special Section on Social Media Processing

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61272240, 60970047, 61103151 and 71301086, the Doctoral Fund of Ministry of Education of China under Grant No. 20110131110028, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2012FM037, and the Excellent Middle-Aged and Youth Scientists of Shandong Province of China under Grant No. BS2012DX017.

*Corresponding Author

①https://www.netflix.com/, July 2015.

②http://www.amazon.com/, July 2015.

However, traditional recommender systems only utilize the user-item rating matrix for recommendation, and ignore the social connections or trust relations among users. But in our real life, we always turn to our friends we trust for recommendations of products, consultations, music and movies. The social trust relation helps us locate the items we are potentially interested in. Hence, with the advent of online social networks, social trust aware recommender systems have drawn lots of attentions. For example, Ma *et al.*[3-4] explored several ways to incorporate trust relations into the matrix factorization framework. Noel *et al.*[5] improved the existing social matrix factorization objective functions, and proposed a new unified framework for social recommendation.

Unfortunately, most of these existing trust aware recommendation methods are proposed for social networks with explicit feedback of users. In these cases, a user can tell us to what extent he/she likes a specific item by giving a real-valued rating, and we can explicitly know what he/she likes and hates. Nevertheless, explicit feedback is not always available. Most of the feedback in real social networks is not explicit but implicit. In implicit feedback social networks, we can only get a user's positive behaviors from the history of what he/she has clicked, purchased or connected, but never know to what degree he/she likes and what he/she does not like. The learning task for this kind of data is how to infer the user preferences from only positive observations. Rendle *et al.* explored this problem in [6], where they made use of partial order of items and presented a generic Bayesian optimization criterion for personalized ranking. Their work provides us a general way to learn users' interests from implicit data. However they did not consider the impact of social trust relations, which have been demonstrated in the rating prediction based tasks.

Moreover, most of the existing social trust aware recommendation methods assume the trust relationships among users are single and homogeneous. For example, Jamali and Ester[7] incorporated the mechanism of trust propagation into a matrix factorization method, and the binary social relations were considered. However, trust as a social concept is intrinsically multi-faceted and heterogeneous. Simply exploiting the raw values of trust relations cannot get satisfactory results. Intuitively, a user may trust different people in different domains/categories. For example, in multi-category recommender systems, a user may trust an expert in movies category but not trust him/her in cars category. Treating trust relationships of different categories equally will not capture the multi-faceted features hidden below the surface (especially when the social relations only have binary values).

Based on the above observations, we propose to learn a trust aware item ranking model on multi-category social networks. In order to learn from implicit feedback, we reconstruct the user-item rating matrix using the data policy proposed by Rendle *et al.*[6], and learn the user and item latent factors by making use of the partial order of items. To investigate the impact of social trust relations, we derive a social based personalized item ranking criterion from a Bayesian analysis of the problem, where we make the learned user feature vectors be close to those of their neighbors. To explore the impact of users' multi-faceted trust relations, we propose a category-sensitive random walk method to infer the category-specific trust value on each link. By replacing the original social trust value, we arrive at our final item ranking model. Data analysis and experimental results on real-world social networks demonstrate the existence of social influence, and the proposed item ranking method can better utilize users' social trust information in multi-category systems, where our methods can perform better than other state-of-the-art item recommendation methods in terms of AUC.

The primary contributions of this paper can be summarized as follows.

1) We derive a social based item ranking criterion from a Bayesian analysis of the problem for the implicit feedback.

2) We propose a category-sensitive random walk method CRWR to estimate the true trust relations.

3) We replace the raw binary social relation with inferred trust strength and learn the latent feature vectors on multi-category systems.

4) We conduct data analysis and experiments to demonstrate the existence of social influence and the effectiveness of our social based method SocialBPR$_{\mathrm{CRWR}}$.

The remaining of this paper is organized as follows. Section 2 discusses the related recommendation methods. Section 3 first derives the social based ranking criterion from the Bayesian analysis of the problem, and then proposes a category-sensitive random walk method CRWR to estimate the true trust relations. Section 4 conducts data analysis and experiments to demonstrate the existence of social influence and the effectiveness SocialBPR$_{\mathrm{CRWR}}$. Section 5 outlines some conclusions and directions for future work.

2 Related Work

Typical traditional recommender systems are based on collaborative filtering techniques, where two types of methods are well studied: memory-based and model-based. Memory-based methods mainly focus on employing different strategies to find similar users and items for making predictions, which are known as user-based[1] and item-based[2] approaches, respectively. Contrary to memory-based methods, which directly make predictions based on the original rating data, model-based methods focus on employing machine learning and statistical techniques to learn models from the data for making predictions. Algorithms in this category can be grouped as aspect models[8], latent factor models[9], ranking models[10] and the clustering models[11]. For example, Salakhutdinov and Mnih[9] proposed a probabilistic linear model with Gaussian observation noise, which provides a probabilistic way to learn latent matrices from the known rating data. Shi *et al.*[10] exploited a list-wise learning to rank techniques to improve the performance of the state-of-the-art matrix factorization techniques for the task of item ranking.

The above traditional model-based methods only utilize user-item rating matrix for recommendations and ignore the social trust and friend relations in social networks. But in real life, we always turn to our friends we trust for recommendations. Based on this intuition, many researchers have recently started to analyze trust-based recommender systems[4,12-14]. Jamali and Ester[12] proposed a random walk based recommendation method, where they considered not only the ratings of the target item, but also those of similar items. Ma *et al.*[4] introduced social trust restrictions into recommender systems, where they naturally fused the users' tastes and those of their friends together in a probabilistic factor analysis framework. Jiang *et al.*[14] proposed a social contextual recommendation framework to incorporate individual preference and interpersonal influence to improve the accuracy of social recommendation. Huang *et al.*[15] proposed a Hybrid Multigroup CoClustering recommendation framework, where they used user-item rating records, user social networks, and item features extracted from the DBpedia knowledge base to mine meaningful user-item groups. Ma[16] studied the problem of social recommendation with implicit social information and in his work, a general matrix factorization framework was employed to incorporate different implicit social information.

However, most of these existing social trust aware recommendation methods focus on the optimization of the squared error loss on user-item ratings, and few of them are directly optimized for ranking, especially for the task of ranking in implicit feedback social networks[17-19]. For example, Yang *et al.*[17] conducted a comprehensive study on improving the accuracy of top-k social recommendation by extending the training objective function to include both the observed ratings and the missing ratings. But their work resorts to optimizing the squared error, where the partial order of items is left out. Jamali and Ester[18] exploited a trust network to improve the top-n recommendation, where they first performed a random walk on the trust network, and then performed a weighted merge of the results from trust-based and collaborative filtering approach. Nonetheless, their work is conducted on explicit feedback social networks, and needs users to provide explicit ratings for items. Although there is some work directly optimized for ranking, none can handle the trust propagation phenomenon and the multi-faced trust relations. For example, Du *et al.*[20] extended the BPR (Bayesian Personalized Ranking) method, and proposed a user graph regularized pairwise matrix factorization to seamlessly integrate user information into pair-wise matrix factorization procedure. Krohn-Grimberghe *et al.*[21] formalized the problem of recommendation in social networks as a multi-relational learning problem and solved it by extending the BPR method to the multi-relational case. Pan and Chen[22] proposed a GBPR (Group Bayesian Personalized Ranking) method by introducing the richer interactions among users to relax the individual and independence assumptions made in the BPR method.

Moreover, most of these existing methods assume a single type of trust relations, and the role of multi-faceted trust for social recommendation has not been fully considered. Matsuo and Yamamoto[23] modeled the bidirectional effects between trust relations and product ratings, but they used binary trust relations. Tang *et al.*[24] studied the multi-faceted trust relations between users in product review sites. They demonstrated that people with trust relationships have more similar multi-faceted interests than those without trust relationships and proposed a fine-grained approach to capture multi-faceted trust relationships. Yang *et al.*[25] focused on inferring category-specific social trust circles from available rating data and social networks to improve recommendation accuracy. The key idea of their work is to determine the best subset of a user's friends for making recommendations in an item category.

As a conclusion, the social recommendation problem for implicit feedback in multi-faceted trust networks has not been well studied. In this paper, we systematically analyze this problem, and propose a social trust aware ranking method to improve recommendation results.

3 Our Approach

In this section, we first describe the social recommendation problem with only positive observations. Then we derive the social trust aware ranking criteria from the Bayesian analysis of the problem. Finally, we explore the influence of users' multi-faceted trust relations by inferring category-specific trust value on each link.

3.1 Problem Description

Let us consider the item recommendation problem (also called personalized ranking) in implicit feedback social networks, where we can only observe positive rating behaviors of users, that is, we can only know a user has rated an item, but do not know the exact rating value. Now the task in this scene is to provide a user with a ranked list of items that he/she is likely to rate. In real world, this process includes three central elements: users' trust network, the interests of users and their friends, and users' favorite categories/domains. The typical trust network can be shown in Fig.1(a), where six users (from u_1 to u_6) are connected by 11 relations (edges), and each relation is weighted by $s_{ij} \in (0, 1]$ to indicate to what extent user u_i trusts user u_j. Normally, the trust value will be domain-specific. Users in different domains (as shown in Fig.1(c)) will place different trust strengths to others, as there are naturally experts of different domains.

As we can see in Fig.1(b), each user also rated some items (from v_1 to v_3) to express his/her favors, but only positive behaviors can be observed. The remaining unknown data (denoted as ?) is a mixture of actually negative and missing values. We cannot use a common approach to learn user features directly from unobserved data, as they are unable to be distinguished from the two levels anymore. The problem we study in this paper is how to make use of trust relations and categories to predict the personalized ranking of items.

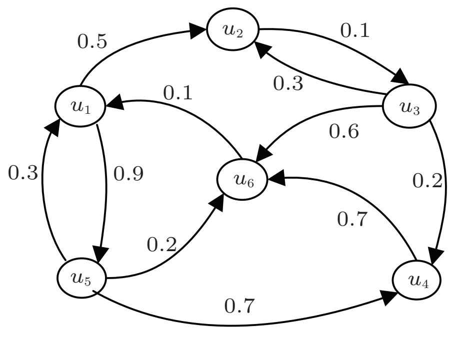
3.2 Review of BPRMF

Suppose we have an $m \times n$ rating behavior matrix $\boldsymbol{R} = (r_{ui})_{m\times n} \in \{1, ?\}^{m\times n}$ denoting m users' rating behaviors on n items, where $r_{ui} = 1$ denotes user u has rated item i in the past, and $r_{ui} =?$ denotes an unknown rating status of user u to item i. To learn from this implicit feedback, we reconstruct the user-item rating matrix using the following data policy proposed by Rendle *et al.*[6], that is, if an item i has been rated by user u, i.e., $(u, i) \in \boldsymbol{R}$, then we assume that u prefers i over all other non-rated items. But for the items that have been rated by the same user, we cannot infer any preference. The same is true for the items that a user has not rated yet. Formally, the training data $\mathcal{D}_{\mathcal{R}} : \mathcal{U} \times \mathcal{I} \times \mathcal{I}$ can be created by:

$$\mathcal{D}_{\mathcal{R}} = \{(u, i, j)|i \in \mathcal{I}_u^+ \wedge j \in \mathcal{I} \setminus \mathcal{I}_u^+\},$$

where $\mathcal{U}$ is the user set, $\mathcal{I}$ is the item set, and $\mathcal{I}_u^+$ and $\mathcal{I} \setminus \mathcal{I}_u^+$ are the observed item set and the missing item set associated with user u, respectively. The meaning of $(u, i, j) \in \mathcal{D}_{\mathcal{R}}$ is that user u is assumed to prefer item i over item j.

In order to find the correct personalized ranking for all items $i \in \mathcal{I}$, Rendle *et al.*[6] proposed the BPR method to solve this problem. BPR is derived by a



(a)

	v_1	v_2	v_3
u_1	1	1	?
u_2	1	?	1
u_3	?	1	?
u_4	1	?	1
u_5	?	1	1
u_6	1	1	?

(b)

	c_1	c_2	c_3
u_1	1	1	
u_2			1
u_3	1	1	
u_4	1		1
u_5		1	1
u_6	1	1	

(c)

Fig.1. Example for trust-based recommendation. (a) Social trust graph. (b) User-item rating matrix. (c) User-category matrix.

Bayesian analysis of the problem using the likelihood function for $p(i >_u j|\theta)$ and the prior probability for the model parameter $p(\theta)$ ($i >_u j$ denotes user u prefers item i over item j). The original objective function of BPR is written as:

$$BPR = -\sum_{(u,i,j)\in\mathcal{D}_\mathcal{R}} \ln\sigma(\hat{r}_{uij}(\theta)) + \lambda_\theta||\theta||^2,$$

where $\sigma(x)$ is the logistic sigmoid function $\sigma(x) = 1/(1+e^{-(x)})$, λ_θ is the regularization parameter of θ, and $\hat{r}_{uij}(\theta)$ is an arbitrary real-valued function of the model parameter θ. By decomposing the estimator $\hat{r}_{uij}$ as:

$$\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj},$$

any standard collaborative filtering model can be applied to predict the preference of user u to item i ($\hat{r}_{ui}$).

BPRMF[6] is the model that delegates the task of modeling the relationship between u, i and j to matrix factorization (MF) method[9]. The objective function of BPRMF can be achieved as:

$$BPRMF = -\sum_{(u,i,j)\in\mathcal{D}_\mathcal{R}} \ln\sigma(\hat{r}_{uij}) + \frac{\lambda_{\boldsymbol{U}}}{2}||\boldsymbol{U}||_F^2 + \frac{\lambda_{\boldsymbol{V}}}{2}||\boldsymbol{V}||_F^2,$$

where $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj} = \boldsymbol{U}_u{}^{\mathrm{T}}\boldsymbol{V}_i - \boldsymbol{U}_u{}^{\mathrm{T}}\boldsymbol{V}_j$, $\lambda_{\boldsymbol{U}}$ and $\lambda_{\boldsymbol{V}}$ are the regularization parameters, $\boldsymbol{U} \in \mathbb{R}^{l\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{l\times n}$ are the latent user and item feature matrices, with column vectors $\boldsymbol{U}_u$ and $\boldsymbol{V}_i$ or $\boldsymbol{V}_j$ representing user-specific and item-specific latent feature vectors respectively. As in MF, the zero-mean spherical Gaussian priors[9] are placed on user and item feature vectors:

$$p(\boldsymbol{U}|\sigma_{\boldsymbol{U}}^2) = \prod_{u=1}^{m}\mathcal{N}(\boldsymbol{U}_u|0, \sigma_{\boldsymbol{U}}^2\boldsymbol{I}),$$
$$p(\boldsymbol{V}|\sigma_{\boldsymbol{V}}^2) = \prod_{k=1}^{n}\mathcal{N}(\boldsymbol{V}_k|0, \sigma_{\boldsymbol{V}}^2\boldsymbol{I}), \quad (1)$$

where $\mathcal{N}(\boldsymbol{x}|\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2 and $\boldsymbol{I}$ is the identity matrix.

3.3 Social Trust Aware Ranking Criteria

3.3.1 Social Trust Assumption

Suppose we have a directed social trust graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{U}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node in $\mathcal{U}$ represents a user in the network and each edge in $\mathcal{E}$ represents a trust relation between two users. Let $\boldsymbol{S} = (s_{uv})_{m\times m}$ denote the $m \times m$ social trust matrix of graph $\mathcal{G}$.

BPRMF makes an assumption, that is, for any two users u and v, the pair-wise preference of user u is independent of that of user v. But in real world, we always turn to our friends we trust for recommendations. In the theory of social influence[26], the behavior of a user u is affected by his/her direct neighbors N_u. User u's taste is close to those of the neighbors that he/she trusts. In other words, the latent feature vector of u is close to the latent feature vectors of all his/her trusted neighbors $v \in N_u$. This influence can be formulated as follows[7,27]:

$$\hat{\boldsymbol{U}}_u = \frac{\sum_{v\in N_u} s_{uv}\boldsymbol{U}_v}{\sum_{v\in N_u} s_{uv}} = \sum_{v\in N_v} s^*_{uv}\boldsymbol{U}_v,$$

where $\hat{\boldsymbol{U}}_u$ is the estimated latent feature vector of u, s_{uv} is the trust value of how much user u trusts user v. In most cases, the statement of trust relations only takes on positive values ($s_{uv} = 1$), which will not fully reflect the true relationship between two users. In Subsection 3.4, we will investigate how to estimate the true trust strengths in multi-category systems. $s^*_{uv} = s_{uv}/\sum_{v\in N_u} s_{uv}$ is the row normalization form of the trust matrix $\boldsymbol{S} = (s_{uv})_{m\times m}$, so that $\sum_{v=1}^{m} s^*_{uv} = 1$.

The user latent feature vector $\boldsymbol{U}$ now has two factors: the zero-mean spherical Gaussian prior, and the conditional distribution of $\boldsymbol{U}$ given the latent features of his/her trusted neighbors. Hence,

$$\begin{aligned} & p(\boldsymbol{U}|\boldsymbol{S}, \sigma_{\boldsymbol{U}}^2, \sigma_{\boldsymbol{S}}^2) \propto p(\boldsymbol{U}|\sigma_{\boldsymbol{U}}^2)p(\boldsymbol{U}|\boldsymbol{S}, \sigma_{\boldsymbol{S}}^2) \\ = & \prod_{u=1}^{m}\mathcal{N}(\boldsymbol{U}_u|0, \sigma_{\boldsymbol{U}}^2\boldsymbol{I}) \times \\ & \prod_{u=1}^{m}\mathcal{N}(\boldsymbol{U}_u|\sum_{t\in N_u} s^*_{ut}\boldsymbol{U}_t, \sigma_{\boldsymbol{S}}^2\boldsymbol{I}). \end{aligned} \quad (2)$$

3.3.2 Bayesian Inference

Like in BPR, the Bayesian formulation of finding the correct personalized ranking for all items $i \in \mathcal{I}$ is to maximize the posterior probability over the user and item latent feature vectors:

$$\begin{aligned} & p(\boldsymbol{U}, \boldsymbol{V}|\mathcal{D}_\mathcal{R}, \boldsymbol{S}, \sigma_{\boldsymbol{S}}^2, \sigma_{\boldsymbol{U}}^2, \sigma_{\boldsymbol{V}}^2) \\ \propto\ & p(\mathcal{D}_\mathcal{R}|\boldsymbol{U}, \boldsymbol{V})p(\boldsymbol{U}|\boldsymbol{S}, \sigma_{\boldsymbol{S}}^2, \sigma_{\boldsymbol{U}}^2)p(\boldsymbol{V}|\sigma_{\boldsymbol{V}}^2). \end{aligned}$$

Note that taking the trust network into account does not change the equation for the conditional distribution of the observed item pairs. It only affects the user feature vectors. The Bernoulli distribution over the binary random variable $\delta((u, i, j) \in \mathcal{D}_\mathcal{R})$ can also be used:

$$
\begin{aligned}
& p(\mathcal{D}_{\mathcal{R}}|\boldsymbol{U},\boldsymbol{V}) \\
= & \prod_{(u,i,j)\in\mathcal{U}\times\mathcal{I}\times\mathcal{I}} p(i>_u j|\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j)^{\delta((u,i,j)\in\mathcal{D}_{\mathcal{R}})} \times \\
& (1-p(i>_u j|\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j))^{(1-\delta((u,i,j)\in\mathcal{D}_{\mathcal{R}}))},
\end{aligned}
$$

where δ is the indicator function[6]. Due to the totality and anti-symmetry of a sound pair-wise ordering scheme[6], the above formula can be simplified to:

$$
p(\mathcal{D}_{\mathcal{R}}|\boldsymbol{U},\boldsymbol{V}) = \prod_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}} p(i>_u j|\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j). \quad (3)
$$

In order to get a personalized total order, we resort to the MF model by defining the individual probability that a user really prefers item i over item j as:

$$
p(i>_u j|\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j) = \sigma(\hat{r}_{uij}(\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j)),
$$

where $\hat{r}_{uij}(\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j) = {\boldsymbol{U}_u}^{\mathrm{T}}\boldsymbol{V}_i - {\boldsymbol{U}_u}^{\mathrm{T}}\boldsymbol{V}_j$ captures the partial order relationship between user u, item i and item j. For convenience, in the following, we will skip the arguments $\boldsymbol{U}_u,\boldsymbol{V}_i,\boldsymbol{V}_j$ from $\hat{r}_{uij}$.

To complete the Bayesian inference of the personalized ranking task, we also place zero-mean spherical Gaussian prior on item feature vector $\boldsymbol{V}$ (see (1)). Based on (2) and (3), the posterior probability of latent feature vectors can be achieved as follows:

$$
\begin{aligned}
& p(\boldsymbol{U},\boldsymbol{V}|\mathcal{D}_{\mathcal{R}},\boldsymbol{S},\sigma_{\boldsymbol{S}}^2,\sigma_{\boldsymbol{U}}^2,\sigma_{\boldsymbol{V}}^2) \\
\propto & \; p(\mathcal{D}_{\mathcal{R}}|\boldsymbol{U},\boldsymbol{V})p(\boldsymbol{U}|\boldsymbol{S},\sigma_{\boldsymbol{S}}^2,\sigma_{\boldsymbol{U}}^2)p(\boldsymbol{V}|\sigma_{\boldsymbol{V}}^2) \\
= & \prod_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}} \hat{r}_{uij} \times \prod_{u=1}^{m}\mathcal{N}(\boldsymbol{U}_u|\sum_{t\in N_u} s^*_{ut}\boldsymbol{U}_t,\sigma_{\boldsymbol{S}}^2\boldsymbol{I}) \times \\
& \prod_{u=1}^{m}\mathcal{N}(\boldsymbol{U}_u|0,\sigma_{\boldsymbol{U}}^2\boldsymbol{I}) \times \prod_{k=1}^{n}\mathcal{N}(\boldsymbol{V}_k|0,\sigma_{\boldsymbol{V}}^2\boldsymbol{I}).
\end{aligned}
$$

The log of the posterior probability can be computed as follows:

$$
\begin{aligned}
& \ln p(\boldsymbol{U},\boldsymbol{V}|\mathcal{D}_{\mathcal{R}},\boldsymbol{S},\sigma_{\boldsymbol{S}}^2,\sigma_{\boldsymbol{U}}^2,\sigma_{\boldsymbol{V}}^2) \\
= & \sum_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}} \ln\sigma(\hat{r}_{uij}) - \\
& \frac{1}{2\sigma_{\boldsymbol{S}}^2}\sum_{u=1}^{n}\left(\boldsymbol{U}_u - \sum_{v\in N_u} s^*_{uv}\boldsymbol{U}_u\right)^{\mathrm{T}} \cdot \\
& \left(\boldsymbol{U}_u - \sum_{v\in N_u} s^*_{uv}\boldsymbol{U}_u\right) - \frac{1}{2\sigma_{\boldsymbol{U}}^2}\sum_{u=1}^{n}{\boldsymbol{U}_u}^{\mathrm{T}}\boldsymbol{U}_u - \\
& \frac{1}{2\sigma_{\boldsymbol{V}}^2}\sum_{k=1}^{n}{\boldsymbol{V}_k}^{\mathrm{T}}\boldsymbol{V}_k - \frac{1}{2}((n\times l)\ln\sigma_{\boldsymbol{U}}^2 + \\
& (m\times l)\ln\sigma_{\boldsymbol{V}}^2 + (n\times l)\ln\sigma_{\boldsymbol{S}}^2) + C,
\end{aligned}
$$

where C is a constant that does not depend on the parameters. Maximizing the log-posterior over user and item features with hyper-parameters kept fixed is equivalent to minimizing the following objective function L_1:

$$
\begin{aligned}
& L_1(\mathcal{D}_{\mathcal{R}},\boldsymbol{S},\boldsymbol{U},\boldsymbol{V}) \\
= & -\sum_{u=1}^{m}\sum_{i\in\mathcal{I}_u^+}\sum_{j\in\mathcal{I}\setminus\mathcal{I}_u^+}\ln\sigma(\hat{r}_{uij}) + \\
& \frac{\beta}{2}\sum_{u=1}^{m}\left(\boldsymbol{U}_u - \sum_{v\in N_u} s^*_{uv}\boldsymbol{U}_v\right)^{\mathrm{T}} \cdot \\
& \left(\boldsymbol{U}_u - \sum_{v\in N_u} s^*_{uv}\boldsymbol{U}_v\right) + \frac{\lambda_{\boldsymbol{U}}}{2}\|\boldsymbol{U}\|_F^2 + \frac{\lambda_{\boldsymbol{V}}}{2}\|\boldsymbol{V}\|_F^2,
\end{aligned}
$$

where $\lambda_{\boldsymbol{U}} = 1/\sigma_{\boldsymbol{U}}^2$, $\lambda_{\boldsymbol{V}} = 1/\sigma_{\boldsymbol{V}}^2$, $\beta = 1/\sigma_{\boldsymbol{S}}^2$ are the social regularization parameters determining the trade-off between the user rating data and the social trust network information, and $v \in N_u$ is the direct neighbor set user u trusts. Gradient-based approaches can be applied to find a local minimum.

Note that our Social Based Bayesian Personalized Ranking method (SocialBPR) is a pair-wise approach, and its training object is not to directly predict the list of items. Instead, it cares about the relative order between two items. Compared with the point-wise or rating-based approach, it is closer to the concept of "ranking" as it does not focus on accurately predicting the rating of each item. However, the output of SocialBPR is still the user and item latent factors $\boldsymbol{U}$ and $\boldsymbol{V}$. The main difference is that we optimize them by correctly ranking item pairs instead of scoring single items. Another advantage of our approach is that it models the trust propagation in social networks. More specifically, for any user u, the feature vector of u is dependent on the feature vectors of his/her trusted neighbors N_u. Recursively, the feature vector of each trusted neighbor $v \in N_u$ is also dependent on the feature vector of v's trusted neighbors N_v.

3.4 Learning the Strengths of Multi-Faceted Trust Relations

SocialBPR assumes the social trusts in different categories have the same influence on the target user and the trust relations only take on values from either 1 or 0, which do not truly reflect their impact on user behaviors. However, in real-world social networks, trust is multi-faceted, that is, a user places different trusts in different categories. Hence, it is desirable to understand the true trust strengths from multi-category systems.

In multi-category systems, each user can have multiple favorite categories, and will make social trusts based on his/her category interests. Let $\boldsymbol{W} = (w_{ij})_{m\times m}$ denote the $m \times m$ category similarity matrix between users, where $w_{ij} \in (0, 1]$ denotes the similarity value associated with the edge from u_i to u_j in trust graph $\mathcal{G}$, and $w_{ij} = 0$, otherwise. The physical meaning of a similarity value w_{ij} can be interpreted as how much a user u_i is similar to user u_j in different categories.

Let $\boldsymbol{P}_u \in \mathbb{R}^{1\times h}$ be the user category preference vector③ of user u, with each entry p_{uk} denoting the probability that user u has interest in category k, which is formally defined as:

$$p_{uk} = \frac{n_{uk}}{n_u},$$

where n_u is the total number of items rated by user u, and n_{uk} is the number of items in category k rated by user u. The ordered set of categories is determined by their preferences p_{uk} over the categories. And the category interest similarity w_{uv} between users u and v can be measured by the Kendall rank correlation coefficient[28]:

$$w_{uv} = 1 - \frac{4 \times \sum_{i,j\in C_u\cap C_v} I^{-}((p_{ui} - p_{uj})(p_{vi} - p_{vj}))}{|C_u \cap C_v| \times (|C_u \cap C_v| - 1)},$$

where C_u is the observed category set associated with user u and $I^{-}(x)$ is the indicator function defined as:

$$I^{-}(x) = \begin{cases} 1, & \text{if } x < 0, \\ 0, & \text{otherwise.} \end{cases}$$

The approach of Kendall will take values between -1 and $+1$, where -1 is obtained when one order is the exact reverse of the other order and $+1$ is obtained when both orders are identical. In our experiments, we only keep positive values ($w_{ij} > 0$), as a negative w_{ij} denotes that the ordered set of categories is negatively correlated.

Given the social trust network and category similarity, we propose a Category-Sensitive Random Walk with Restart (CRWR) method to learn the intrinsic relevance between two users. For a standard random walk with restart approach, a random walker starts from the i-th vertex, iteratively with probability $1 - \alpha$ jumps to other vertices according to transition probabilities $q_i = \{q_{i1}, \cdots, q_{in}\}$, and with probability α jumps back to itself. After reaching the steady state, the probability of the random walker staying at the j-th vertex corresponds to the relevance score of vertex i to vertex j.

To reflect users' multi-category interests, we bias the transition matrix $\boldsymbol{Q}$ by users' category similarity matrix $\boldsymbol{W}$, which can be defined as follows:

$$\boldsymbol{Q} = \boldsymbol{D}^{-1} \cdot \boldsymbol{W}, \tag{4}$$

where $\boldsymbol{D}$ is the degree matrix of trust graph $\mathcal{G}$. In this scene, when the random walker jumps to other vertices, the walker will not only consider the graph structure but also consider the category similarity between two users. The final steady-state probability matrix can be obtained by iterating the following updates:

$$\boldsymbol{B}(t + 1) = (1 - \alpha)\boldsymbol{Q}\boldsymbol{B}(t) + \alpha\boldsymbol{I}, \tag{5}$$

where $\boldsymbol{B}(t)$ and $\boldsymbol{B}(t+1)$ are the state probability matrices at time t and time $t+1$ respectively. The iterations will finally converge when $t \to \infty$. For any two vertexes i and j, the value of steady-state probability b_{ij} represents how well user i trusts/knows user j in social graph.

Algorithm 1 summarizes the whole procedure of the category-sensitive random walk method for estimating the strength of social trusts.

Algorithm 1. CRWR($\mathcal{G}$, $\boldsymbol{D}$, $\boldsymbol{W}$, α, ϵ, $\boldsymbol{B}$, t)

```
Initialize B, t, and the stop condition ϵ
Compute transition probabilistic matrix Q using (4)
repeat
    Update the state matrix B using (5)
until |B(t) − B(t − 1)| < ϵ
return the steady state matrix B*
```

3.5 Learning Algorithm

Using the (normalized) estimated trust strength $\boldsymbol{B}^*$ to replace users' direct trust matrix $\boldsymbol{S}$, we arrive at our final item recommendation method SocialBPR$_{\text{CRWR}}$ and train the following objective function:

$$\begin{aligned} & L_2(\boldsymbol{R}, \boldsymbol{B}^*, \boldsymbol{U}, \boldsymbol{V}) \\ = & -\sum_{u=1}^{m} \sum_{i\in\mathcal{I}_u^+} \sum_{j\in\mathcal{I}\setminus\mathcal{I}_u^+} \ln\sigma(\hat{r}_{uij}) + \\ & \frac{\beta}{2}\sum_{u=1}^{m}\left(\boldsymbol{U}_u - \sum_{v\in N_u} b^*_{uv}\boldsymbol{U}_v\right)^{\mathrm{T}}\left(\boldsymbol{U}_u - \sum_{v\in N_u} b^*_{uv}\boldsymbol{U}_v\right) + \\ & \frac{\lambda_{\boldsymbol{U}}}{2}\|\boldsymbol{U}\|_F^2 + \frac{\lambda_{\boldsymbol{V}}}{2}\|\boldsymbol{V}\|_F^2, \end{aligned} \tag{6}$$

where b^*_{uv} denotes the trust value from user u to user v estimated by CRWR.

③h is the number of categories.

As the optimization criterion derived by (6) is differentiable, gradient descent based algorithms are an obvious choice for minimization. But as we can see, due to the huge number of preference pairs ($O(|\boldsymbol{R}||\mathcal{I}|)$), standard gradient descent is expensive to update the latent features over all pairs. To solve this issue, we exploit the strategy proposed in the BPR[6] method, which is a stochastic gradient descent algorithm based on bootstrap sampling of the training triples (u, i, j). Then, the corresponding latent factors can be updated by the following gradients:

$$\begin{aligned}\frac{\partial L_2}{\partial \boldsymbol{U}_u} =& -\frac{e^{-\hat{r}_{uij}}}{1+e^{-\hat{r}_{uij}}}(\boldsymbol{V}_i - \boldsymbol{V}_j) + \lambda_{\boldsymbol{U}} \boldsymbol{U}_u + \\ & \beta \left(\boldsymbol{U}_u - \sum_{v \in N_u} b^*_{uv} \boldsymbol{U}_v \right) - \\ & \beta \sum_{\{v | u \in N_v\}} b^*_{vu} \left(\boldsymbol{U}_v - \sum_{w \in N_v} b^*_{vw} \boldsymbol{U}_w \right), \quad (7)\end{aligned}$$

$$\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{V}_i} = -\frac{e^{-\hat{r}_{uij}}}{1+e^{-\hat{r}_{uij}}} \boldsymbol{U}_u + \lambda_{\boldsymbol{V}} \boldsymbol{V}_i, \quad (8)$$

$$\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{V}_j} = -\frac{e^{-\hat{r}_{uij}}}{1+e^{-\hat{r}_{uij}}} \boldsymbol{U}_u + \lambda_{\boldsymbol{V}} \boldsymbol{V}_j. \quad (9)$$

The learning algorithm of estimating the latent low-rank matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ is described in Algorithm 2.

Algorithm 2. LEARN($\boldsymbol{R}$, $\boldsymbol{B}^*$, η, $\lambda_{\boldsymbol{U}}$, $\lambda_{\boldsymbol{V}}$, β, ϵ)

```
Initialize U, V, step t, and the stop condition ϵ
repeat
    Draw (u, i, j) from U × I × I
    r̂_uij ← r̂_ui − r̂_uj
    Update U_u, the u-th row of U according to (7)
    Update V_i, the i-th row of V according to (8)
    Update V_j, the j-th row of V according to (9)
    Compute training loss L(t) using (6)
until |L(t) − L(t − 1)| < ϵ
return U and V
```

3.6 Complexity Analysis

The main cost of training the objective function of SocialBPR$_{\text{CRWR}}$ is to compute the loss function L_2 and its gradients against feature vectors. Assuming the average number of direct neighbors per user is $\overline{s}$. Then, the complexity of evaluation of L_2 is $O(|\boldsymbol{R}||\mathcal{I}|l + m\overline{s}l)$, where l is the dimension of latent feature vectors. Since the estimated trust matrix $\boldsymbol{B}^*$ is very sparse, $\overline{s}$ is relatively small. Therefore, the time complexity of computing the objective function L_2 mainly depends on the number of training triples $O(|\boldsymbol{R}||\mathcal{I}|)$, which does not increase the time complexity much (compared with the BPRMF method). The computational complexities for gradients $\frac{\partial L_2}{\partial \boldsymbol{U}}$, $\frac{\partial L_2}{\partial \boldsymbol{V}}$ are $O(N\overline{s}^2 l)$ and $O(Nl)$, respectively (N is the number of sampled triples). Therefore, the total computational complexity for gradients is $O(N\overline{s}^2 l)$, which is linear with respect to the number of sampled triples.

4 Data Analysis and Experimental Results

In this section, we first investigate the relationship between trust networks and users' preferences on real-world social networks, and then conduct several experiments to compare the recommendation performance of our approach with other state-of-the-art collaborative filtering methods.

4.1 Datasets

We make use of two datasets Epinions and Ciao[29] as the data source for our experiments. The Epinions dataset is from the online review site Epinions.com, which is a well-known consumer review site that was established in 1999. The purpose of this website is to help people to be informed of buying decisions from other consumer reviews. To post a review, users need to first rate the product or service using a 5-scale integer (from 1 to 5). Note that Epinions.com groups products into different categories (e.g., Movies, Books, and Music). Members of Epinions can maintain a "trust" list, which presents a trust network between users. This trust network is used to determine in which order product views are shown to visitors.

The second dataset Ciao is a European based online-shopping portal, which provides a forum for users to write reviews and give their opinions on products with different categories to help others make decisions. The registered user in Ciao can also give an overall rating from 1 star (poor) to 5 stars (excellent) for a particular product, and can nominate others to join his/her trusted network when he/she finds a member's reviews consistently interesting and helpful. Typically, both of these two websites are binary trust networks, where users can only express trust or distrust (0 or 1). As we want to solve an implicit feedback task, we get rid of the rating scores from these two datasets. Now the task is to predict a personalized ranked list starting with the item that the user is most likely to rate next. Table 1 presents some descriptive statistics of the two datasets.

Table 1. Statistics of User-Item Rating Matrix

Statistics	Epinions	Ciao
# of users	4 276	2 016
# of items	6 031	4 361
# of ratings	190 093	74 852
# of categories	27	28
# of trust relations	21 101	39 885
Min. # of ratings per user	20	15
Min. # of ratings per item	20	15
Rating sparsity	99.416	99.327
Trust network density	0.002 7	0.018 9

4.2 Analysis of Preference and Social Trust

In the view of the theory of homophily and social influence, similar users are more likely to establish trust relations, and users with trust relationships are supposed to have similar interests. In this subsection, we investigate the existence of homophily in social networks and answer the following question: "Do people with trust relationships have similar rating behavior to those without trust relationships?" To answer this question, we need to define how to measure the interest similarity between a pair of users from their rating behaviors.

Let $A(u)$ be the set of items rated by user u, $A(v)$ be the set of items rated by user v. The similarity between users u and v can be simply computed by the Cosine similarity:

$$x_{uv} = \frac{|A(u) \cap A(v)|}{\sqrt{|A(u)||A(v)|}},$$

where $|A(u)|$ denotes the length of set $A(u)$. However, the basic Cosine similarity function ignores the influence of popular items, that is, when two users rate many common popular items, we cannot deduce that they have similar interests. But when two users rate many common cold items, we can say they have similar interests with high confidence. Hence, Breese *et al.*[30] adjusted the Cosine similarity function as follows:

$$x_{uv} = \frac{\sum_{i \in A(u) \cap A(v)} \frac{1}{\log(1+A(i))}}{\sqrt{|A(u)||A(v)|}}, \tag{10}$$

where $A(u) \cap A(v)$ is the set of items user u and user v both have rated. The term $1/\log(1+(A(i))$ punishes the influence of popular items.

For each user u, we compute two similarities, i.e., $x_t(u)$ and $x_r(u)$, where $x_t(u)$ is the average similarity between user u and his/her trust network, and $x_r(u)$ is the average similarity between user u and randomly chosen users who are not in the trust network of user u. The randomly chosen users have the same size with user u's trust network.

For a visual comparison, in Fig.2, we plot the Kernel-smoothing density estimations[4] based on vectors $\boldsymbol{x}_t$ and $\boldsymbol{x}_r$. For both Epinions and Ciao, we can observe that compared with $\boldsymbol{x}_t$, $\boldsymbol{x}_r$ has smaller similarity values. This evidence from Fig.2 suggests a positive answer to our question: users with trust relations have more similar interests than those without trust relations.

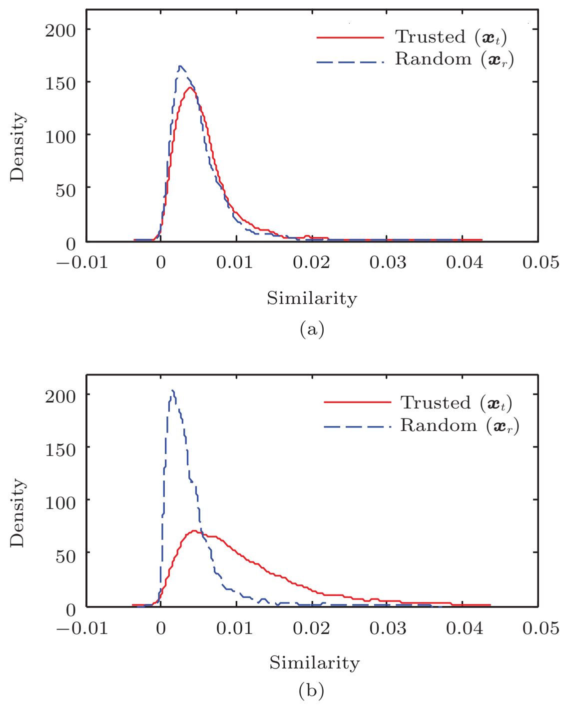


Fig.2. Density estimates of users' similarity. (a) Epinions. (b) Ciao.

4.3 Analysis of Multi-Faceted Trust Relations

In this subsection, we investigate the multi-faceted trust relations among users and want to answer the following two questions. 1) For each facet, do users with trust relationships have more similar interests than those without trust relationships? 2) Do users trust their friends differently in multi-category systems? To answer the first question, we randomly choose six categories from Epinions and Ciao, and study the preference similarity between users in each category. For each user u in a specific category c, we compute two similarities based on (10), i.e., $x_c(u)$ and $x_r(u)$, where $x_c(u)$ is the average similarity between user u and his/her trust networks in category c, and $x_r(u)$ is the average similarity

[4] http://en.wikipedia.org/wiki/Kernel_density_estimation/, July 2015.

between user u and randomly chosen users. The Kernel-smoothing density estimations based on the similarity vectors of six categories and $\boldsymbol{x}_r$ are plotted in Fig.3. For both datasets, $\boldsymbol{x}_r$ has smaller concentrated values compared with the similarity vectors in the six categories, which indicates the answer for the first question: for each category, users with trust relationships have more similar preferences than those without trust relationships.

To answer the second question, for each user, we further calculate the variance⑤ of similarities with her/his trust network in the six categories and randomly chosen users. Let $v_c(u)$ denote the variance of the similarities with the trust network of user u in category c, and $\overline{v}_c$ denote the average of variances of all users in category c. $\overline{v}_r$ is the average variance of the similarities with randomly choose users. Table 2 shows the results in Epinions and Ciao datasets, where $\overline{v}_c$ is always larger than $\overline{v}_r$. This answers our second question: in multi-category systems, users trust their friends differently and they have greater trust in some friends than in others, that is, it demonstrates the existence of multi-faceted trust relationships between users.

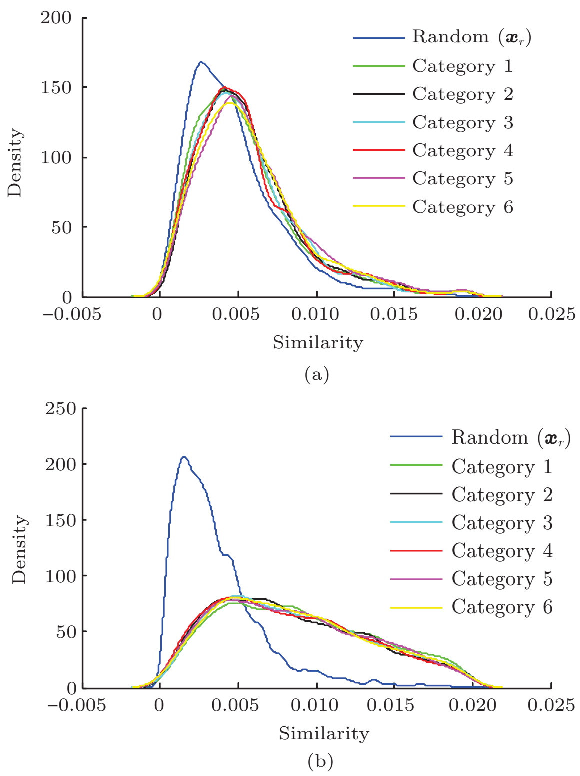


Fig.3. Density estimates of users' category similarity. (a) Epinions. (b) Ciao.

⑤http://en.wikipedia.org/wiki/Variance/, July 2015.

Table 2. Average of Variances of Category Similarities

Category	Epinions	Ciao
1	3.8825×10^{-5}	8.3637×10^{-5}
2	1.0704×10^{-4}	1.1210×10^{-4}
3	7.0867×10^{-5}	1.0894×10^{-4}
4	4.0905×10^{-5}	9.1194×10^{-5}
5	3.8954×10^{-5}	1.1743×10^{-4}
6	3.6831×10^{-5}	1.3456×10^{-4}
$\overline{v}_r$	3.4742×10^{-5}	6.6196×10^{-5}

4.4 Metric

We use the popular metric, the average area under the ROC curve (AUC), to measure the personalized ranking performance of our proposed approach. The metric AUC[6] is defined as:

$$AUC = \frac{1}{|\mathcal{U}|}\sum_{u}\frac{1}{|E(u)|}\sum_{(i,j)\in E(u)}\delta(\hat{x}_{ui} > \hat{x}_{uj}),$$

where $\hat{x}_{ui}$ is the predicted preference value of user u to item i, and $\delta(x)$ is the Heaviside loss function[31]. $E(u)$ is the evaluation item pairs with respect to user u in test dataset:

$$E(u) = \{(i,j)|(u,i) \in \mathcal{I}_u^+ \wedge (u,j) \in \mathcal{I} \setminus \mathcal{I}_u^+\}.$$

The AUC of a random guess method is 0.5 and the best performance that can be achieved is 1. A higher value of the AUC indicates a better performance.

4.5 Comparisons

In order to evaluate the recommendation performance of our proposed approach, we compare the recommendation results with the following methods.

Random. This method provides the basic recommendation result in our experiments, which ranks the items randomly for the users in the test set.

MostPopular. This method sorts the items based on how often they have been rated in the training data, where the popularity of the items determines the order of the ranked list. This simple method is supposed to have reasonable performance, since many people tend to access the popular items.

WRMF. This method was proposed by Pan *et al.*[32] and Hu *et al.*[33] for item prediction in implicit feedback, which extends the matrix factorization method and adds weights in the error function to increase the impact of positive feedback.

BPRMF. This method was introduced in Subsection 3.2, which only utilizes the user-item rating matrix for item ranking.

MR-BPR. This method was proposed by Krohn-Grimberghe *et al.*[21], which extends the BPR method to the multi-relational case. This method is a state-of-the-art method for implicit feedback recommendation with social information.

SocialBPR. This is our social item ranking method proposed in Subsection 3.3, where the social influence from trusted neighbors is considered.

SocialBPR$_{CRWR}$. This is our social item ranking method with multi-faceted trust relations (see more details in Subsections 3.4 and 3.5).

In our experiments, we split the training data with different ratios to test the above algorithms. For example, training data 80% means we select randomly 80% actions (80% user-item pairs) of each user for training, and predict the remaining 20% actions. The random selection is conducted five times independently. The parameter settings of our approaches are: in both 80% and 70% training data, for Epinions, we set the social regularization parameter $\beta = 0.0004$, and for Ciao, we set $\beta = 0.15$. The regularization parameters⑥ of latent factors are set as 0.008.

Table 3 shows the comparison results with setting the latent factor dimension l as 5 and 10 respectively. We can observe that, although the MostPopular method just ranks the items based on their popularity, this simple method has reasonable performance, since many people tend to focus on popular items. WRMF as the state-of-the-art matrix factorization method for item recommendation using binary information achieves a better AUC value than the MostPopular method, but does worse than the BPRMF method. BPRMF achieves a substantial AUC improvement over WRMF, which indicates optimizing the pairwise rank criteria directly in item rank problem is more reasonable. In this work, we also include comparisons with one state-of-the-art implicit feedback recommendation method considering social information (MR-BPR). We find our social network based methods (SocialBPR and SocialBPR$_{\mathrm{CRWR}}$) outperform MR-BPR in both Eipnions and Ciao datasets. This result indicates social trusts play an important role in users' decision process and our methods can model this information more effectively. From the results, we can observe that in Ciao dataset, SocialBPR improves more obviously than that in the Epinions data, since in Ciao, users are more likely to trust others and make decisions based on their trust network (as we have plotted in Fig.2).

To explore the influence of users' multi-faceted trust relationships, we conduct experiments on these two datasets by utilizing the category information, where the method SocialBPR$_{\mathrm{CRWR}}$ outperforms ScoialBPR. This result demonstrates that simply taking the row values of trust relations will not get satisfactory results, and it is beneficial to model trust strength in recommender systems. Note that, with small value of latent factor dimensions, the results of MF-based methods, WRMF, BPRMF, MR-BPR, SocialBPR and SocialBPR$_{\mathrm{CRWR}}$, can achieve a reasonable AUC value, which can significantly reduce the computational complexity. Hence, in our following experiments, we set the latent factor dimension to 5.

Table 3. Performance Comparisons with Different Training Datasets in the Measure of AUC (Training = 80%)

Dimensionality	Dataset	Random	MostPopular	WRMF	BPRMF	MR-BPR	SocialBPR	SocialBPR$_{\mathrm{CRWR}}$
5	Epinions	0.4998	0.7141	0.8212	0.8354	0.8309	0.8431	0.8481
	Ciao	0.5038	0.6722	0.7892	0.7889	0.7918	0.8064	0.8094
10	Epinions	0.4994	0.7142	0.8320	0.8357	0.8327	0.8456	0.8497
	Ciao	0.5033	0.6724	0.7958	0.7890	0.8013	0.8109	0.8120

4.6 Impact of Parameter β

In SocialBPR, the social regularization parameter β plays an important role, which balances the information from users' interests and the preferences of their trusted friends. It controls the degree that our trust-aware method SocialBPR should depend on the interests of users' trusted friends. If $\beta = 0$, we will only utilize users' own history rating matrix for recommendation. If $\beta \to \infty$, we will derive the latent feature vectors only from those of direct neighbors. In other cases, we make recommendations from users' own rating behaviors as well as the preferences of their trusted neighbors.

⑥ $\lambda_{\boldsymbol{U}} = \lambda_{\boldsymbol{V}} = 0.008$.

Fig.4 illustrates how the changes of β affect the recommendation results on the measure AUC. We notice that the value of parameter β affects the recommendation results significantly, which denotes that incorporating the preferences of users' trusted neighbors considerably improves the recommendation accuracy. To get an appropriate β value, we use a 5-fold cross-validation for learning and testing. For each experiment, we conduct five times and take the mean value as the final result. In both Epinions and Ciao datasets, as β increases, the values of AUC increase (recommendation accuracy increases) at first, but when β surpasses a certain threshold, the values of AUC decrease (recommendation accuracy decreases) with the further increase of the value of β. The experimental results are consistent with the intuition that purely utilizing users' own rating history or purely utilizing the interests of their direct trusted neighbors for recommendations cannot make better results than fusing them together.

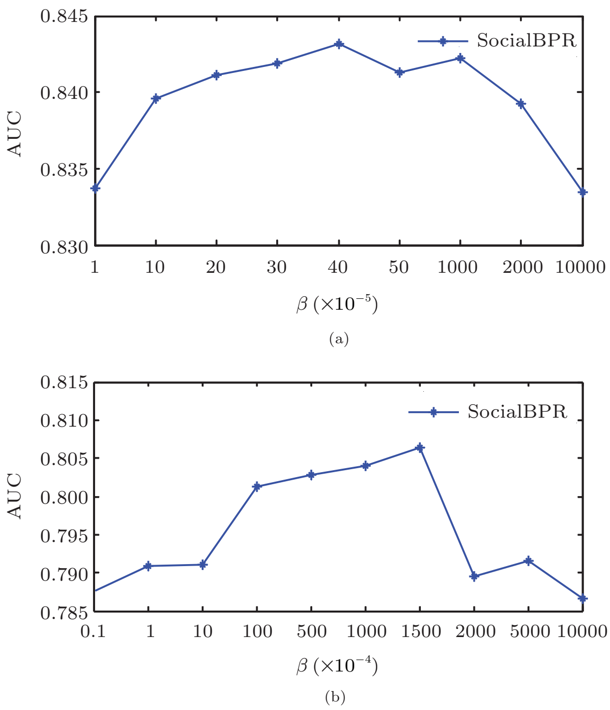


Fig.4. Impact of social regularization parameter β on the performance of recommendation in (a) Epinions and (b) Ciao.

4.7 Recommendation on Cold Start Users

To explore the recommendation performance on cold start users, we evaluate a cold-start scenario in which the social trust network is already available for some users, but the training data does not contain any information about the users in the test set. For Epinions, we create four split percentages from 50% to 80% of the rating data being used for training. For Ciao, we use 10% to 40% for training (a setting in which only a little of rating information remains available).

The recommendation performance is shown in Fig.5. BPRMF and MR-BPR as two representative state-of-the-art rank methods have done better than WRMF in Epinions, but worse in Ciao. Our trust-based ranking method SocialBPR$_{\mathrm{CRWR}}$ can do better than BPRMF and MR-BPR in both two datasets, which indicates that our method is also suitable to model the social influence of cold start users. However, we also notice that, as the rating data used for training decreases, the improvement of our method becomes less obvious, since the available trust information has also decreased. The more rating and social trust information can be used, the better performance our method SocialBPR$_{\mathrm{CRWR}}$ can achieve.

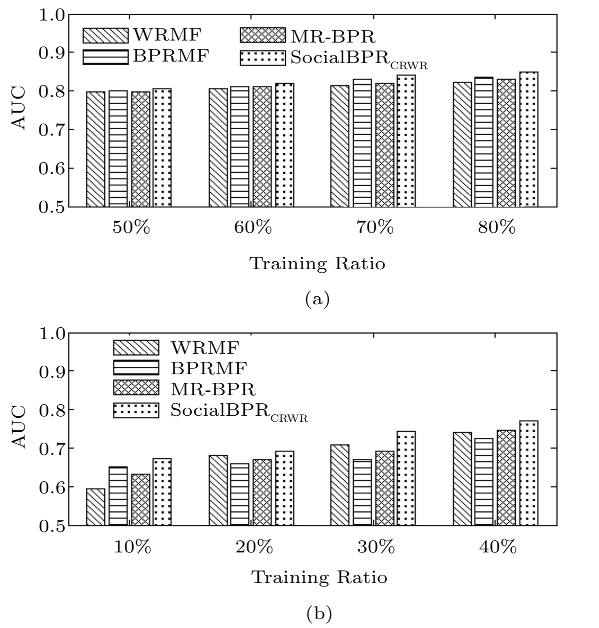


Fig.5. Impact of different training ratios on the performance of recommendation in (a) Epinions and (b) Ciao.

4.8 Convergence Analysis

We further compare the convergence of the SocialBPR and the BPRMF methods. Fig.6 shows the comparison results on Epinions and Ciao data. For these two methods, in each iteration, we select the same number of instances for training and set the learning rate both as 0.07. From the results, we see that both BPRMF and SocialBPR can converge within 50 iterations in Epinions data, and within 60 iterations in

Ciao data. Incorporating social trust information does not slow down the convergence rate of SocialBPR, but makes it achieve a higher AUC value than BPRMF.

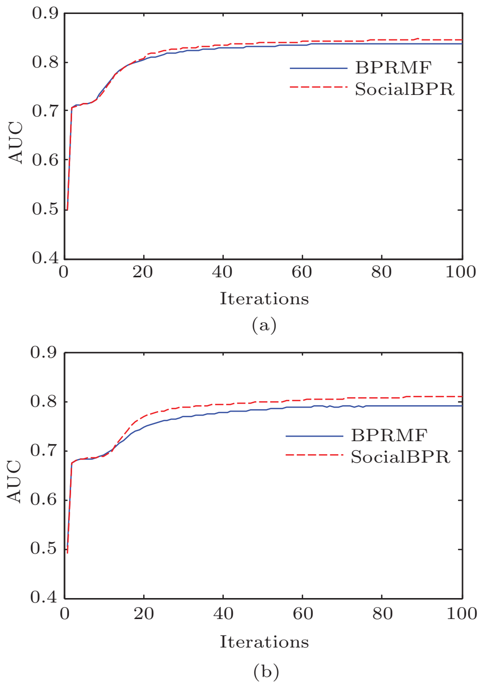


Fig.6. Convergence analysis on (a) Epinions and (b) Ciao. Learning rate = 0.07.

5 Conclusions

With the rapid growth of online social networks, the social based recommender systems have become more and more popular and important. In this work, we focused on the social item recommendation problem in the implicit feedback and proposed a novel social item ranking method called SocialBPR$_{\mathrm{CRWR}}$. We derived the optimization criterion of SocialBPR$_{\mathrm{CRWR}}$ from a Bayesian analysis of the problem, where we introduced the social trust interactions among users from the theory of social influence to improve the performance of item recommendation. To understand the true social trust relations, we further proposed a category-sensitive random walk method CRWR to estimate the multi-faceted trust strengths. By replacing the direct trust matrix of the original social network, we arrived at our final strength aware social item recommendation method SocialBPR$_{\mathrm{CRWR}}$. Data analysis and experimental results on real-world datasets demonstrated the existence of trust influence and the effectiveness of our social based ranking method SocialBPR$_{\mathrm{CRWR}}$.

In our work, we mainly focused on the social trust relation in multi-category systems, but we did not differentiate the categories of social trust relations, such as friends or colleagues. As future work, we plan to develop new trust category aware algorithms to further improve our social based ranking methods.

References

[1] Jin R, Chai J Y, Si L. An automatic weighting scheme for collaborative filtering. In *Proc. the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2004, pp.337-344.

[2] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th International Conference on World Wide Web*, May 2001, pp.285-295.

[3] Ma H, Yang H, Lyu M R, King I. SoRec: Social recommendation using probabilistic matrix factorization. In *Proc. the 17th ACM Conference on Information and Knowledge Management*, October 2008, pp.931-940.

[4] Ma H, King I, Lyu M R. Learning to recommend with social trust ensemble. In *Proc. the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2009, pp.203-210.

[5] Noel J, Sanner S, Tran K N, Christen P, Xie L, Bonilla E V, Abbasnejad E, Penna N D. New objective functions for social collaborative filtering. In *Proc. the 21st International Conference on World Wide Web*, April 2012, pp.859-868.

[6] Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. the 25th Conference on Uncertainty in Artificial Intelligence*, June 2009, pp.452-461.

[7] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proc. the 4th ACM Conference on Recommender Systems*, September 2010, pp.135-142.

[8] Hofmann T. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *Proc. the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 28-August 1, 2003, pp.259-266.

[9] Salakhutdinov R, Mnih A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, Platt J C, Koller D, Singer Y, Roweis S T (eds.), Curran Associates, Inc., 2008, pp.1257-1264.

[10] Shi Y, Larson M, Hanjalic A. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proc. the 4th ACM Conference on Recommender Systems*, September 2010, pp.269-272.

[11] Xue G, Lin C, Yang Q, Xi W, Zeng H, Yu Y, Chen Z. Scalable collaborative filtering using cluster-based smoothing. In *Proc. the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2005, pp.114-121.

[12] Jamali M, Ester M. TrustWalker: A random walk model for combining trust-based and item-based recommendation. In *Proc. the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, June 28-July 1, 2009, pp.397-406.

[13] Yang B, Lei Y, Liu D, Liu J. Social collaborative filtering by trust. In *Proc. the 23rd International Joint Conference on Artificial Intelligence*, August 2013, pp.2747-2753.

[14] Jiang M, Cui P, Liu R, Yang Q, Wang F, Zhu W, Yang S. Social contextual recommendation. In *Proc. the 21st ACM International Conference on Information and Knowledge Management*, October 29-November 2, 2012, pp.45-54.

[15] Huang S, Ma J, Cheng P, Wang S. A hybrid multigroup coclustering recommendation framework based on information fusion. *ACM Transactions on Intelligent Systems and Technology*, 2015, 6(2): 27:1-27:22.

[16] Ma H. An experimental study on implicit social recommendation. In *Proc. the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 28-August 1, 2013, pp.73-82.

[17] Yang X, Steck H, Guo Y, Liu Y. On top-k recommendation using social networks. In *Proc. the 6th ACM Conference on Recommender Systems*, September 2012, pp.67-74.

[18] Jamali M, Ester M. Using a trust network to improve top-N recommendation. In *Proc. the 3rd ACM Conference on Recommender Systems*, October 2009, pp.181-188.

[19] Yuan Q, Chen L, Zhao S. Factorization vs. regularization: Fusing heterogeneous social relationships in top-N recommendation. In *Proc. the 5th ACM Conference on Recommender Systems*, October 2011, pp.245-252.

[20] Du L, Li X, Shen Y. User graph regularized pairwise matrix factorization for item recommendation. In *Lecture Notes in Computer Science 7121*, Tang J, King I, Chen L, Wang J (eds.), Springer Berlin Heidelberg, 2011, pp.372-385.

[21] Krohn-Grimberghe A, Drumond L, Freudenthaler C, Schmidt-Thieme L. Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In *Proc. the 5th ACM International Conference on Web Search and Data Mining*, February 2012, pp.173-182.

[22] Pan W, Chen L. GBPR: Group preference based Bayesian personalized ranking for one-class collaborative filtering. In *Proc. the 23rd International Joint Conference on Artificial Intelligence*, August 2013, pp.2691-2697.

[23] Matsuo Y, Yamamoto H. Community gravity: Measuring bidirectional effects by trust and rating on online social networks. In *Proc. the 18th International Conference on World Wide Web*, April 2009, pp.751-760.

[24] Tang J, Gao H, Liu H. mTrust: Discerning multi-faceted trust in a connected world. In *Proc. the 5th ACM International Conference on Web Search and Data Mining*, February 2012, pp.93-102.

[25] Yang X, Steck H, Liu Y. Circle-based recommendation in online social networks. In *Proc. the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2012, pp.1267-1275.

[26] Friedkin N E. A Structural Theory of Social Influence. Cambridge University Press, 1998.

[27] Ma H, Zhou D, Liu C, Lyu M R, King I. Recommender systems with social regularization. In *Proc. the 4th ACM International Conference on Web Search and Data Mining*, February 2011, pp.287-296.

[28] Marden J I. Analyzing and Modeling Rank Data. CRC Press, 1996.

[29] Tang J, Gao H, Liu H, Sarma A D. eTrust: Understanding trust evolution in an online world. In *Proc. the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2012, pp.253-261.

[30] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. the 14th Conference on Uncertainty in Artificial Intelligence*, July 1998, pp.43-52.

[31] Herschtal A, Raskutti B. Optimising area under the ROC curve using gradient descent. In *Proc. the 21st International Conference on Machine Learning*, July 2004.

[32] Pan R, Zhou Y, Cao B, Liu N N, Lukose R, Scholz M, Yang Q. One-class collaborative filtering. In *Proc. the 8th IEEE International Conference on Data Mining*, December 2008, pp.502-511.

[33] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In *Proc. the 8th IEEE International Conference on Data Mining*, December 2008, pp.263-272.

Lei Guo received his Ph.D. degree in computer architecture from Shandong University, Jinan, in 2015. Now he works at the School of Management Science and Engineering, Shandong Normal University, Jinan. His research interests include information retrieval, social network and recommender system.

Jun Ma received his B.S., M.S. and Ph.D. degrees in computer science from Shandong University, Jinan, Ibaraki University, Hitachi City, and Kyushu University, Fukuoka City, respectively. He worked as a senior researcher at the Department of Computer Science of Ibaraki University in 1994 and the German National Research Center for Information Technology (GMD) in Germany from Sept. 2000 to Dec. 2003. At present, he is a professor in the School of Computer Science and Technology, Shandong University. He is a senior member of CCF and a member of ACM and IEEE. His research interests include algorithms, AI, parallel computing, knowledge management and Web technology.

Hao-Ran Jiang received her M.S. degree in computer science and technology from Shandong Normal University, Jinan, in 2012. Now, she is a software engineer in Shandong Post Company, Jinan. Her research interests include grid computing and social computing.

Zhu-Min Chen received his Ph.D. degree in computer science and technology from Shandong University, Jinan, in 2008. Currently, he is an associate professor and master supervisor in the School of Computer Science and Technology, Shandong University. He is a senior member of CCF and a member of ACM. His research interests include Web information retrieval, data mining and social network analysis.

Chang-Ming Xing received his Ph.D. degree in management science and engineering from Shandong Normal University, Jinan, in 2010. He is an associate professor in Shandong University of Finance and Economics, Jinan. His research interests include social network analysis and grid computing.