World Wide Web (2018) 21:1141–1163
DOI 10.1007/s11280-017-0476-7

Joint user knowledge and matrix factorization for recommender systems

Yonghong Yu[1,2] **· Yang Gao**[2] **· Hao Wang**[2] **· Ruili Wang**[3]

Received: 13 February 2017 / Revised: 11 May 2017 / Accepted: 27 June 2017 /
Published online: 5 August 2017

Abstract Currently, most of the existing recommendation methods treat social network users equally, which assume that the effect of recommendation on a user is decided by the user's own preferences and social influence. However, a user's own knowledge in a field has not been considered. In other words, to what extent does a user accept recommendations in social networks need to consider the user's own knowledge or expertise in the field. In this paper, we propose a novel matrix factorization recommendation algorithm based on integrating social network information such as trust relationships, rating information of users and users' own knowledge. Specifically, since we cannot directly measure a user's knowledge in the field, we first use a user's status in a social network to indicate a user's

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering*
Guest Editors: Wojciech Cellary, Hua Wang, and Yanchun Zhang

This paper is an extension version of the WISE'2016 Long Presentation paper "Joint User Knowledge and Matrix Factorization for Recommender Systems" [40].

✉ Yonghong Yu
yuyh.nju@gmail.com

Yang Gao
gaoy@nju.edu.cn

Hao Wang
wanghao@nju.edu.cn

Ruili Wang
R.Wang@massey.ac.nz

1 TongDa College, Nanjing University of Posts and Telecommunications, Nanjing, People's Republic of China

2 State Key Lab for Novel Software Technology, Nanjing University, Nanjing, People's Republic of China

3 School of Engineering and Advanced Technology, Massey University, Palmerston North, New Zealand

knowledge in a field, and users' status is inferred from the distributions of users' ratings and followers across fields or the structure of domain-specific social network. Then, we model the final rating of decision-making as a linear combination of the user's own preferences, social influence and user's own knowledge. Experimental results on real world data sets show that our proposed approach generally outperforms the state-of-the-art recommendation algorithms that do not consider the knowledge level difference between the users.

Keywords Recommender systems · Social networks · User status · Matrix factorization

1 Introduction

With the rapid growth of information available on the World Wide Web, users are confronted with a serious information overload problem. In order to alleviate this issue, recommender systems [1] are proposed to provide users with *personalized* information, products or services to satisfy their tastes and preferences. Because of such attractive features, recommender systems have become more and more popular and are widely deployed in modern e-commerce applications, such as Amazon,[1] Youtube,[2] Netflix,[3] LinkedIn,[4] etc.

Collaborative filtering (CF) [3] is one of the most widely used techniques for building recommender systems and has achieved great success in e-commerce. CF methods discover hidden preferences of users from past activities of users, i.e., the user-item rating matrix, to make recommendations. However, CF approaches, including matrix factorization methods [11, 18], suffer from *data sparsity* and *cold start* issues [1]. For example, matrix factorization techniques cannot effectively learn the latent feature vectors for users with only a few ratings or newly added items. Nonetheless, matrix factorization techniques offer a flexible framework to incorporate additional sources of information to alleviate data sparsity and cold start issues.

With the increasing popularity of online social network services, social network platforms provide rich information for recommender systems. In order to improve the performance of traditional recommendation methods, several social-network-based recommendation approaches [6, 9, 14, 15, 33, 34] have been proposed to extend basic matrix factorization methods by exploiting rich social information. The underlying assumption of social-based recommendation approaches is that friends share common interests. A user is more likely to adopt the item recommendations from her friends than those from non-friends. However, most of existing social-network-based recommendation algorithms ignore the difference of users' knowledge levels in different categories. Intuitively, the factors that affect a user decision include (1) how much she trust her friends, and (2) the user's own knowledge in a field. For example, suppose user u is an expert in movies but has limited knowledge about cars, then user u may be less affected by other people's opinions/recommendations when she receives movie-related recommendations from her trusted friends. By contrast, the user u may be willing to accept car-related recommendations from her trusted friends who are familiar with cars. That is to say, the degree of social influence is strong for user u in the

[1] http://www.amazon.com

[2] https://www.youtube.com/

[3] http://www.netflix.com

[4] https://www.linkedin.com

"Car" category since user u is layman in this field. Hence, the user's knowledge needs to be considered when recommender systems provide recommendations for users.

Based on the above intuitions, we assume that the processes of rating decision-making are affected by three factors: users' own preferences, social influence and users' own knowledge, and thus we propose a novel matrix factorization recommendation algorithm based on integrating social network information such as trust relationships, rating information of users as well as users' own knowledge. Specifically, since we cannot directly measure a user's knowledge in the field, we first use a user's status in a social network to indicate a user's knowledge in a field. Moreover, we adopt two schemes to infer user's status: linear combination method and domain-specific PageRank [20] method. The linear combination method utilizes the distributions of users' ratings and followers across fields to compute users' status; the domain-specific PageRank method takes users of recommender systems as Web pages and computes users' status based on domain-specific social trust network. Then, we model the final rating of decision-making as a linear combination of users' own preferences, social influence and users' own knowledge. Experimental results on real world data sets show that our proposed approach can model the decision-making processes of users better, and outperforms the state-of-the-art recommendation algorithms that do not consider the knowledge level difference between the users.

The key contributions of our work are summarized as follows:

- We propose a novel matrix factorization based recommendation algorithm by considering user's own preferences, social influence and user's own knowledge.
- In a social network, we utilize a user's status to represent the user's knowledge level in a field since we cannot directly measure a user's knowledge in the field. Moreover, we adopt linear combination method and domain-specific PageRank method to infer users' status.
- We perform extensive experiments to evaluate our proposed method on real-life data sets. The results show that our proposed method outperforms the state-of-the-art recommendation algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews related work in recommender systems. Section 3 introduces some preliminary knowledge. Section 4 describes the details of our proposed user knowledge enhanced matrix factorization recommendation algorithm. Experiments are evaluated in Section 5. Finally, we conclude this paper and present some directions for future work in Section 6.

2 Related work

In this section, we review major related work for recommender systems, including the traditional collaborative filtering methods and the social-network-based recommendation approaches.

2.1 Collaborative filtering

Collaborative filtering (CF) [1, 3] approaches are widely deployed in modern E-commerce Web sites. Unlike content-based filtering methods [17] which need users' demographics and item descriptions and involve complicated natural language processing, CF methods are domain independent and only require users' past activities, i.e. a user-item rating matrix, to make recommendations. CF approaches can be divided into three main categories [1]: memory-based CF algorithms, model-based CF algorithms and hybrid algorithms.

Memory-based CF algorithms, also known as neighbor-based methods, use the entire user-item rating matrix to generate recommendations. Typical memory-based CF algorithms include user-based methods [3] and item-based methods [12, 25]. User-based approaches assume that if users u and v have similar preferences in past activities, then they will share common interests in the future. By contrast, item-based approaches suppose that users tend to prefer similar items, and provide predictions based on the ratings given by target users for the items which are similar to target items. The key to user-based and item-based methods is to adopt suitable similarity measures to compute pairwise similarity between users or items. Typical similarity measures include cosine similarity, Pearson correlation coefficient and adjusted cosine similarity [25]. Recently, several new similarity measures are proposed to alleviate the cold start problem, such as proximity impact popularity [2], heuristic similarity model [13] and coupled object similarity [30, 41].

In contrast to memory-based CF approaches, model-based CF methods first make use of statistical and machine learning techniques to learn a predictive model from training data. The predictive model characterizes the rating behaviors of target users. Then the model-based filtering approaches use the trained model rather than the original user-item matrix to compute predictions. Typical model-based filtering approaches include Bayesian networks [3], clustering model [29, 31, 39], latent semantic analysis [7, 8], restricted Boltzmann machines [22] and association rules [23]. Breese et al. [3] presented a CF algorithm based on Bayesian networks learned from training data. Hofmann [8] introduced latent class variables to discover user communities and prototypical interest profiles. Ungar and Foster [29] grouped similar users in the same class and made predictions according to the active users' neighbors that belong to the same class as active users. Sarwar et al. [23] applied association rule discovery algorithms to seek associations between co-purchased items and provided recommendations based on the strength of the associations.

In general, memory-based CF algorithms are easily implemented and produce reasonably high prediction quality. However, they present serious scalability problem since memory-based CF algorithms have to process the entire data set to compute a single prediction. As the numbers of users and items grow, these algorithms are not appropriate for online systems which need to respond to users in real time. Model-based CF algorithms tend to be faster in prediction time than the memory-based approaches, although building or updating models may require a considerable amount of time. However, many models are quite complex, as they have many parameters to estimate [4].

With the great success of the Netflix Prize competition, matrix factorization methods [11, 18] have attracted a lot of attention since matrix factorization techniques can effectively and efficiently deal with a very large scale user-item rating matrix. Matrix factorization approaches make an assumption that only a few latent factors contribute to preferences of users and characteristics of items. Hence, matrix factorization approaches simultaneously embed both user and item feature vectors into a low-dimensional latent factor space, where the correlation between user preferences and item characteristics can be computed directly [18, 21, 24, 26, 27].

CF methods suffer from the cold start problem due to the sparsity of user-item rating information. For example, it is difficult for user-based methods to find the most similar neighbors when the target user has few ratings. Similarly, item-based methods may not able to seek a similar neighborhood for a target item which has been just added into recommender systems.

2.2 Social-network-based recommendation approaches

Recently, several social-network-based recommendation algorithms have been proposed to extend basic matrix factorization methods by leveraging rich social relations. Typical social-network-based recommendation algorithms include *SoRec* [14], *RSTE* [15], *SocialMF* [9], *TrustMF* [33] and *TrustSVD* [6].

Ma et al. [14] proposed *SoRec*, which integrates a user-item rating matrix and users' social relationships. Specifically, *SoRec* bridges these two different data resources by sharing the user latent feature matrix between them. However, as reported in [15], *SoRec* does not intuitively reflect the real world recommendation process (i.e., is hardly interpretable). In order to improve *SoRec*, Ma et al. [15] proposed *RSTE*, which fuses users' own tastes and their trusted friends' favors together by an ensemble parameter. In [9], Jamali et al. proposed *SocialMF*. Specifically, *SocialMF* incorporates trust propagation into probabilistic matrix factorization model [18] to improve the quality of recommendation. In order to model the multi-faceted friend relationships, Yang et al. [32] proposed a circle-based recommendation algorithm in social networks. In [33], Yang et al. proposed a novel social-network-based recommendation approach, named *TrustMF*, which fuses sparse ratings and sparse trust relationships. *TrustMF* is built on the observation that a user is likely to be affected by existing ratings or reviews from trusted others, and at the same time, this user's ratings or reviews will also influence others who trusts him/her. Technically, *TrustMF* maps users into two latent low-dimensional spaces, the truster space and the trustee space. Therefore, each user is simultaneously characterized by a truster-specific feature vector and a trustee-specific feature vector. More Recently, Guo et al. [6] proposed *TrustSVD* that not only takes explicit and implicit influence of ratings into account, but also considers the explicit and implicit influence of trust value among users in this recommendation model. All of the above approaches report that social network information is beneficial to social-unaware matrix factorization based recommendation methods.

Unlike the aforementioned methods which ignore the difference between users' knowledge of different fields, our proposed method assumes that the processes for rating decision-making are affected by three factors: users' own preferences, social influence and users' own knowledge. In other words, a user's knowledge in a particular field decides how much the user will accept recommendations. Hence, the strength of influence from different friends is different. Incorporating users' knowledge into matrix factorization is capable of improving the performance of traditional social-network-based recommendation approaches.

3 Preliminary knowledge

In this section, we introduce the preliminary knowledge related to our proposed social-network-based recommendation approach. We first describe the social-network-based recommendation problem in Section 3.1. Then, we briefly introduce matrix factorization in Section 3.2.

3.1 Problem description

There are two different types of information used in social-network-based recommender systems: user-item rating matrix and social network information. User-item rating matrix R comprises two entity sets: the set of N users $U = \{u_1, u_2, ..., u_N\}$ and the set of M items $I = \{v_1, v_2, ..., v_M\}$. Each entry R_{ui} represents the rating given by user u on item i. In principle, R_{ui} can be any real number, but usually ratings are integers and fall into [0, 5], in which 0 indicates that the user has not yet rated that item. A higher rating corresponds to better satisfaction. The set of items rated by the user u is denoted as I_u ($I_u \subseteq I$). Table 1 demonstrates an example of user-item rating matrix, which consists of five users (u_1 to u_5) and six items (v_1 to v_6) which are toys. In addition, each item belongs to one category, such as "Car", "Movie" etc. In Table 1, six items are classified into three categories, and items with same colors belong to the same category. In this paper, we use "domain" and "field" interchangeably.

In practice, the user-item rating matrix R is generally very sparse with many unknown entries since a typical user may have only rated a tiny percentage of items. This sparse nature of the user-item rating matrix leads to poor recommendation quality.

A social network is generally represented as a directed social trust graph $G = (U, E)$, where U indicates the set of all users and E represents social trust relations between users. For each edge e_{ij} connecting user u_i to u_j, there is a corresponding trust weight $T_{ij} \in [0, 1]$ which is the extent to which user u_i trusts user u_j, and $T_{ij} = 0$ means that user u_i does not trust u_j at all. All the trust values between users are constructed as a trust matrix $T = [T_{ij}]_{N \times N}$. Note that the trust matrix T is often asymmetric since the trust relationships are not mutual in the real world. Figure 1 shows an example of social network graph corresponding to the user-item rating matrix in Table 1.

The task of recommender systems is to predict the missing rating on the specified item i for an active user u, denoted by $\widehat{R}_{ui}$, by utilizing the observed users' ratings as well as the additional social network information.

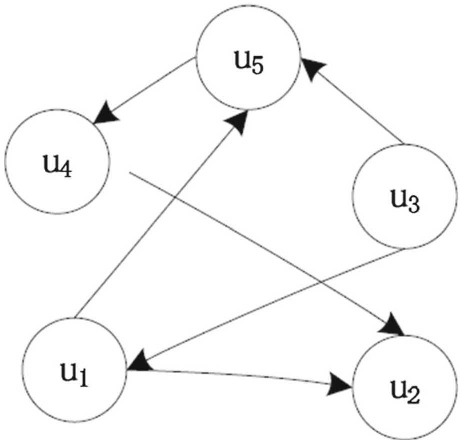
3.2 Matrix factorization

Matrix factorization is widely employed in recommender systems due to its effectiveness and efficiency for dealing with the large scale user-item rating matrix R. The goal of matrix

Table 1 A toy example of user-item rating matrix

	v_1	v_2	v_3	v_4	v_5	v_6
u_1	3	0	4	4	0	5
u_2	1	3	0	0	4	0
u_3	0	3	0	4	0	3
u_4	0	0	3	0	3	2
u_5	0	3	4	1	0	0

Figure 1 A toy example of social network graph



factorization is to learn the latent feature vectors of users and items from all known ratings, of which the inner products can be good estimators of unknown ratings. Formally, matrix factorization decomposes the user-item rating matrix R into two low-rank latent feature matrices $P \in \mathbb{R}^{K\times N}$ and $Q \in \mathbb{R}^{K\times M}$, where $K \ll \min\{N, M\}$, and then uses the product of P and Q to approximate the rating matrix R. As a result,

$$R \approx \widehat{R} = P^T Q = \begin{bmatrix} p_1^T \\ p_2^T \\ \cdots \\ p_N^T \end{bmatrix} \begin{bmatrix} q_1 & q_2 & \ldots & q_M \end{bmatrix}, \tag{1}$$

where the column vectors p_u and q_i represent the K-dimensional user and item feature vectors, respectively.

The latent feature matrices P and Q can be learned by minimizing the sum of squared errors with quadratic regularization terms. Formally,

$$\ell = \min_{P,Q} \frac{1}{2} \sum_{(u,i)\in\Omega} (R_{ui} - p_u^T q_i)^2 + \frac{\lambda_1}{2}||P||_F^2 + \frac{\lambda_2}{2}||Q||_F^2, \tag{2}$$

where $||\cdot||_F^2$ is the Frobenius norm [5], and Ω indicates the set of the (u, i) pairs for known ratings. λ_1 and λ_2 are regularization parameters. Usually, *stochastic gradient descent* (SGD) [19] is applied to seek a local minimum of the objective function given by (2).

In this paper, we extend the matrix factorization model by incorporating social trust information into the matrix factorization model and particularly considering user knowledge to improve the quality of recommendation.

4 Our approach

In this section, we present our proposed recommendation approach that fuses three factors: users' own tastes, social influence of trusted friends, and users' own knowledge. We first introduce the framework of our proposed social-networks based recommendation model in

Section 4.1. Then in Section 4.2, we present the user knowledge enhanced matrix factorization. Finally, we describe how to compute users' knowledge levels in a social network in Section 4.3.

4.1 The framework of user knowledge enhanced social-networks-based recommendation algorithm

Our proposed method is a user knowledge enhanced social-networks-based recommendation algorithm. The framework of our proposed recommendation algorithm is presented in Figure 2. Our proposed recommendation algorithm consists of four major components: (1) User-item rating matrix segmenting and domain-specific social networks inference; (2) User knowledge level learning; (3) User knowledge enhanced matrix factorization; (4) Ratings prediction.

In the component of user-item rating matrix segmenting and domain-specific social networks inference, we first split the original user-item rating matrix R into domain-specific user-item rating matrices $R^{(c_1)}$, $R^{(c_2)}$, ..., $R^{(c_n)}$ according to the category of items, where n is the number of categories in the recommender system. For example, in Table 1, since items with same colors belong to the same category, the original user-item rating matrix R is divided into three domain-specific user-item rating matrices, indicated as $R^{(c_1)}$, $R^{(c_2)}$ and $R^{(c_3)}$ in Figure 2. Then, we utilize the co-occurrence principle of ratings and social trust relationships to infer domain-specific social network information. Specifically, regarding domain c, the trust value $T_{uv}^{(c)}$ between user u and user v is defined as follows:

$$T_{uv}^{(c)} = \begin{cases} T_{uv}, & T_{uv} \neq 0 \wedge N_u^{(c)} > 0 \wedge N_v^{(c)} > 0 \\ 0, & otherwise, \end{cases} \tag{3}$$

where $N_u^{(c)}$ and $N_v^{(c)}$ indicate the numbers of ratings expressed by user u and user v in the domain c, respectively. T_{uv} is the trust value between user u and user v in the original social network. Equation (3) implies that if and only if both user u and user v has assigned ratings to items belonged to domain c as well as there is a trust relationship between user u and user v in the original social network, then the social trust value $T_{uv}^{(c)}$ is T_{uv}, otherwise,

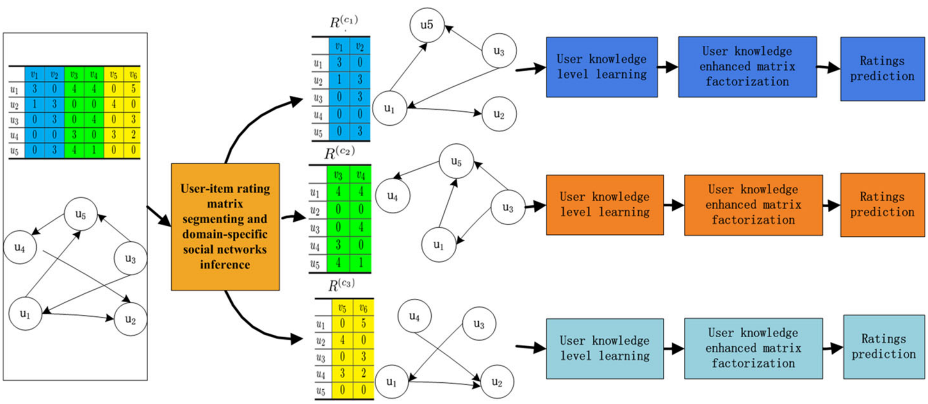


Figure 2 The framework of our proposed recommendation algorithm

$T_{uv}^{(c)}$ is equal to 0. For example, in Figure 2, although user u_5 trusts user u_4 in the original social network, there is not social trust relationship between u_5 and u_4 in the inferred social network regarding domain c_3 since user u_5 has not assigned any ratings to items in category c_3. For each domain c, all social trust relationships among users related to domain c are constructed as domain-specific social trust matrix $T^{(c)}$.

The goal of user knowledge level learning is to compute users' knowledge levels for each domain by utilized domain-specific user-item rating matrix and the inferred social network information. We describe the details of user knowledge learning in Section 4.3. The component of user knowledge enhanced matrix factorization integrates user knowledge with matrix factorization to learn user latent feature vectors and item latent feature vectors for each specific domain, which is introduced in the following section.

4.2 User knowledge enhanced matrix factorization

In traditional social-networks-based recommendation algorithms, whether a user likes an item is determined by the user's own tastes and the influence of her trusted friends. In general, user u's own tastes on item i is represented as the inner product of user u latent feature vector p_u and item i latent feature vector q_i:

$$\widehat{R}_{ui} = {p_u}^T q_i. \tag{4}$$

Since traditional social-networks-based approaches assume that users are mutually influenced, the predicted rating $\widehat{R}_{ui}$ is computed as follows:

$$\widehat{R}_{ui} = {p_u}^T q_i + \textstyle\sum_{k \in \mathcal{F}_u} (T_{uk} \cdot \widehat{R}_{ki}) = {p_u}^T q_i + \textstyle\sum_{k \in \mathcal{F}_u} (T_{uk} \cdot {p_k}^T q_i), \tag{5}$$

where $\mathcal{F}_u$ is the set of users whom user u trusts in the entire social network.

As described in Section 1, each user has different knowledge about different fields, which, to some extent, determines how much the user will accept recommendations. For a specific field, the social influence for users with limited knowledge will be larger than those with high knowledge levels. Hence, the way of predicting missing ratings using (5) does not reflect the real world recommendation process.

In contrast, we assume that the processes of rating decision-making are affected by three factors: users' own preferences, social influence and users' own knowledge with respect to a specific field. Suppose the knowledge level of user u in field c is a_u^c, then the predicted rating $\widehat{R}_{ui}^{(c)}$ is a linear combination of user own tastes, the social influence of trusted friends and user own knowledge level a_u^c. Formally,

$$\widehat{R}_{ui}^{(c)} = (\tau + a_u^c) {p_u^{(c)}}^T {q_i}^{(c)} + (1 - \tau - a_u^c) \sum_{k \in \mathcal{F}_u^{+(c)}} (T_{uk}^{(c)} \cdot {p_k^{(c)}}^T q_i^{(c)}), \tag{6}$$

where $p_u^{(c)}$ and $q_i^{(c)}$ indicate user u latent feature vector and item i latent feature vector in field c. $\mathcal{F}_u^{+(c)}$ is the set of users whom user u trusts in the field c. Parameter τ denotes the weight of user own tastes. User knowledge level a_u^c controls how much user u insists on her own opinions in the field c. Hence, the predicted rating $R_{ui}^{(c)}$ is a trade-off among user u's own tastes, the social influence of her trusted friends in the field c, and user u's own knowledge level. In order to facilitate the description of our proposed approach, we set $\psi_u^c = \tau + a_u^c$.

For each field c, by integrating user own tastes, social influence of trusted friends and user own knowledge into a unified model, we have the following objective function given user-item matrix $R^{(c)}$ and social trust matrix $T^{(c)}$.

$$\begin{aligned}\min_{P^{(c)},Q^{(c)}} \ & \tfrac{1}{2}\textstyle\sum_{(u,i)\in\Omega^{(c)}} \left(R_{ui}^{(c)} - \left(\psi_u^c {p_u^{(c)}}^T q_i{}^{(c)} + (1-\psi_u^c)\sum_{k\in\mathcal{F}_u^{+(c)}} T_{uk}^{(c)} \cdot {p_k^{(c)}}^T q_i^{(c)}\right)\right)^2 \\ & + \tfrac{\lambda_1}{2}\left\|P^{(c)}\right\|_F^2 + \tfrac{\lambda_2}{2}\left\|Q^{(c)}\right\|_F^2,\end{aligned} \tag{7}$$

where $\Omega^{(c)}$ is the set of (u, i) pairs with known ratings in field c.

Note that most users in the social networks express a few or even no ratings. In other words, there are lots of cold start users in a social network. The objective function defined by (7) fails to work for these cold start users. In order to alleviate the cold start problem, we introduce a social regularization term, which is similar to the regularization term described in [16], to constrain the process of matrix factorization. The social regularization term is defined as:

$$\frac{\beta}{2}\sum_{u\in U^{(c)}}\sum_{v\in\mathcal{F}_u^{+(c)}} T_{u,v}^{(c)}\left\|p_u^{(c)} - p_v^{(c)}\right\|_F^2, \tag{8}$$

where β is a regularization parameter for controlling the effect of social trust, and $U^{(c)}$ is the set of users who are interested in the field c. Specifically, a small value of $T_{u,v}^{(c)}$ means that the distance between two user latent feature vectors should be relatively large, and vice versa. Hence, this social regularization term makes two user latent feature vectors closer if they have more common interests. In addition, this social regularization term incorporates the propagation mechanism of trust into the recommendation model, which makes our recommendation approach more reasonable. Suppose user u trusts user v and user v trusts user w with trust value $T_{uv}^{(c)}$ and $T_{vw}^{(c)}$, respectively. If we simultaneously minimize two distances of latent feature vectors, i.e., $T_{uv}^{(c)}||p_u^{(c)} - p_v^{(c)}||_F^2$ and $T_{vw}^{(c)}||p_v^{(c)} - p_w^{(c)}||_F^2$, we actually minimize the distance of user latent feature vectors between user u and w in category c, which models the propagation mechanism of trust.

In this paper, without loss of generality, we map the ratings $R_{ui}^{(c)}$ to the interval [0,1] using the function $f(x) = (x - minRating)/(maxRating - minRating)$, where $maxRating$ and $minRating$ are the maximum and minimum ratings in recommender systems, respectively. Meanwhile, we use logistic function $g(x) = 1/(1 + e^{-x})$ to limit the predicted ratings $\widehat{R}_{ui}^{(c)}$ within the range [0,1]. Finally, the objective function of our method is defined as follows.

$$\begin{aligned}\mathcal{L}^* = & \min_{P^{(c)},Q^{(c)}} \tfrac{1}{2}\textstyle\sum_{(u,i)\in\Omega^{(c)}} \left(R_{ui}^{(c)} - g\left(\psi_u^c {p_u^{(c)}}^T q_i{}^{(c)} + (1-\psi_u^c)\sum_{k\in\mathcal{F}_u^{+(c)}} T_{uk}^{(c)} \cdot {p_k^{(c)}}^T q_i^{(c)}\right)\right)^2 \\ & + \tfrac{\beta}{2}\textstyle\sum_{u\in U^{(c)}}\sum_{v\in\mathcal{F}_u^{+(c)}} T_{u,v}^{(c)}\left\|p_u^{(c)} - p_v^{(c)}\right\|_F^2 + \tfrac{\lambda_1}{2}\left\|P^{(c)}\right\|_F^2 + \tfrac{\lambda_2}{2}\left\|Q^{(c)}\right\|_F^2.\end{aligned} \tag{9}$$

We seek a local minimum of $\mathcal{L}^*$ by applying the SGD method. The derivatives of $\mathcal{L}^*$ with respect to $p_u^{(c)}$ and $q_i^{(c)}$ are computed as:

$$\begin{aligned}\frac{\partial\mathcal{L}^*}{\partial p_u^{(c)}} &= \psi_u^c g'(y_{ui})\left(g(y_{ui}) - R_{ui}^{(c)}\right)q_i^{(c)} + \lambda_1 p_u^{(c)} \\ &\quad + (1-\psi_w^c)\textstyle\sum_{w\in\mathcal{F}_u^{-(c)}} g'(y_{wi})\left(g(y_{wi}) - R_{wi}^{(c)}\right)T_{wu}^{(c)}q_i^{(c)} \\ &\quad + \beta\textstyle\sum_{v\in\mathcal{F}_u^{+(c)}} T_{uv}^{(c)}(p_u^{(c)} - p_v^{(c)}) + \beta\sum_{g\in\mathcal{F}_u^{-(c)}} T_{gu}^{(c)}(p_u^{(c)} - p_g^{(c)}), \\ \frac{\partial\mathcal{L}^*}{\partial q_i^{(c)}} &= g'(y_{ui})\left(g(y_{ui}) - R_{ui}^{(c)}\right) \times \left(\psi_u^c p_u^{(c)} + (1-\psi_u^c)\textstyle\sum_{k\in\mathcal{F}_u^{+(c)}} T_{uk}^{(c)}p_k^{(c)}\right) + \lambda_2 q_i^{(c)}.\end{aligned} \tag{10}$$

We set $y_{ui} = \psi_u^c {p_u^{(c)}}^T {q_i}^{(c)} + (1-\psi_u^c) \sum_{k \in \mathcal{F}_u^{+(c)}} T_{uk}^{(c)} {p_k^{(c)}}^T q_i^{(c)}$ and $y_{wi} = \psi_w^c {p_w^{(c)}}^T {q_i}^{(c)} + (1-\psi_w^c) \sum_{k \in \mathcal{F}_w^{+(c)}} T_{wk}^{(c)} {p_k^{(c)}}^T q_i^{(c)}$ for simplified presentation in (10), $g'(x) = e^{-x}/(1+e^{-x})^2$ is the derivative of logistic function $g(x)$ and $\mathcal{F}_u^{-(c)}$ is the set of users who trust user u in field c.

4.3 User knowledge level learning

User knowledge is a key component of our proposed approach since it determines the degree to which a user is influenced by her social relationships. In this section, we describe how to compute users' knowledge levels based on users' status in a social network.

In a social network, a user u may be interested in several item fields, such as digital cameras, computer, books, music, etc. However, she may only be professional in a subset of fields of interest. In such fields, she will be relatively confident; as a consequence, she will tend to express more ratings in these fields than in the other ones. Moreover, a user may often have more followers in familiar fields than those in unfamiliar fields because she could provide valuable information for others. From this perspective, a user's status (e.g., the number of followers or the number of ratings one has done) reflects the user's knowledge level. Hence, we infer a user's knowledge from the distributions of ratings and followers across fields.

The knowledge level of user u in the field c, denoted as a_u^c, consists of two components: the number of ratings that user u has expressed in field c and the number of followers who trust user u in field c. The rating distribution of user u over all fields is denoted as $\mathcal{DR}_u$,

$$\mathcal{DR}_u = \left(\frac{N_u^{(1)}}{N_u}, \frac{N_u^{(2)}}{N_u}, ..., \frac{N_u^{(n)}}{N_u} \right), \tag{11}$$

where N_u is the total number of ratings given by user u across all fields. From the definition of $\mathcal{DR}_u$, we can see that $\mathcal{DR}_u$ represents the knowledge level distribution of user u. Hence, the first component of user knowledge level a_u^c is defined as $\frac{N_u^{(c)}}{N_u}$.

From the perspective of social network, both the number of followers and the number of followees contribute to the status of users. The status of user u in field c should grow as the number of followers grows, and decline if user u has lots of followees. Formally, the knowledge level distribution of user u over all fields concerning the distribution of followers is defined as,

$$\mathcal{DF}_u = \left(\frac{|{\mathcal{F}_u}^{-(1)}|}{|{\mathcal{F}_u}^{-(1)}| + |\mathcal{F}_u^{+(1)}|}, ..., \frac{|{\mathcal{F}_u}^{-(n)}|}{|{\mathcal{F}_u}^{-(n)}| + |\mathcal{F}_u^{+(n)}|} \right), \tag{12}$$

where $|{\mathcal{F}_u}^{-(c)}|$ and $|\mathcal{F}_u^{+(c)}|$ indicate the numbers of followers and followees of user u in field c, respectively. Thus, the second component of user knowledge level a_u^c is computed as $\frac{|{\mathcal{F}_u}^{-(c)}|}{|{\mathcal{F}_u}^{-(c)}|+|\mathcal{F}_u^{+(c)}|}$.

Fusing both components, the user knowledge level of user u in field c is the linear addition of these two components:

$$a_u^c = \frac{1}{2} \left(\frac{N_u^{(c)}}{N_u} + \frac{|{\mathcal{F}_u}^{-(c)}|}{|{\mathcal{F}_u}^{-(c)}| + |\mathcal{F}_u^{+(c)}|} \right). \tag{13}$$

In fact, for the task of learning user knowledge level, the above linear combination method does not deeply utilize the structure of social network, and just exploits the distribution of followers in social network. Meanwhile, there are many algorithms to compute the relevance ranking of Web pages based on hyperlinks, such as PageRank [20] and HITS [10]. Taking users of recommender systems as Web pages, the social trust network is analogous to Web links graph. Hence, as an alternative to the linear combination method, we adopt one of most widely used algorithms PageRank to compute users' status. For each domain c, we use PageRank algorithm to compute domain-specific rank score for each user based on domain-specific social trust matrix $T^{(c)}$. Let $PR^{(c)} \in \mathbb{R}^{N^{(c)}}$ be ranking scores of all users in the social trust matrix $T^{(c)}$, the ranking score $PR^{(c)}(u)$ of user u in domain c is defined as:

$$PR^{(c)}(u) = (1-\epsilon) \sum_{v \in \mathcal{F}_u^{-(c)}} \frac{PR^{(c)}(v)}{|\mathcal{F}_v^{+(c)}|} + \epsilon E^{(c)}(u), \tag{14}$$

where ϵ is the teleport probability and $E^{(c)}$ is the teleport vector, whose each entry is set to $1/N^{(c)}$. In our experiments, we follow the PageRank algorithm and set $\epsilon = 0.15$. After applying the power iteration algorithm to compute ranking scores $PR^{(c)}$, we take $PR^{(c)}(u)$ as the user knowledge level of user u.

Note that we only apply the linear combination method or PageRank algorithm to infer users' knowledge levels. Since this work focuses on incorporating users' own knowledge into recommender systems, we leave more sophisticated user knowledge level learning methods to future work.

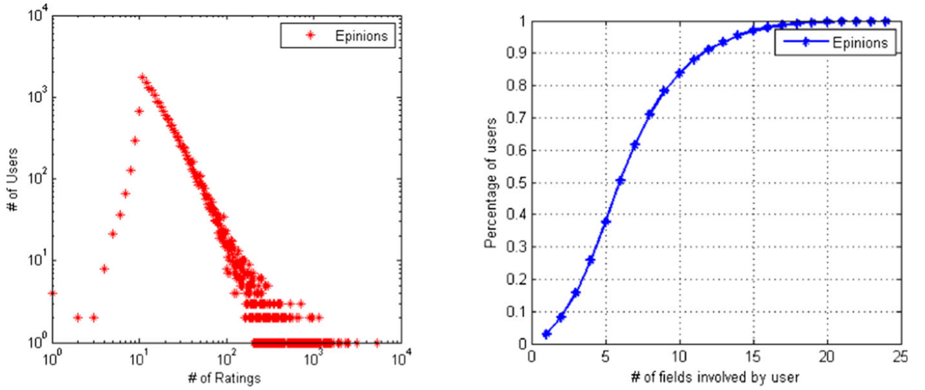
5 Experiments

In this section, we conduct several experiments on real data sets to compare the performance of our proposed recommendation algorithms with other state-of-the-art methods. As users' knowledge levels can be derived from linear combination method or PageRank algorithm, we refer to these user knowledge enhanced matrix factorization recommendation methods as UKMF_DS and UKMF_PR, respectively.

5.1 Data set and evaluation metrics

We choose the Epinions data set to evaluate the performance of our proposed methods since Epinions contains rating information, social network relations and item field. In Epinions, common users can express their opinions on various items in terms of ratings and reviews. Items are divided into several fields, such as "Digital Camera", "Toys", "Movies" etc. The value of each rating is an integer ranging from 1 to 5. Moreover, each user has a set of trusted users, which is formulated by trust relations between users. It should be noted that the trust relations are directed and the trust values are binary.

The Epinions data set used in our experiments was published by the authors of [28]. It contains 922,267 ratings from 22,166 users and 296,277 items. The sparsity level is $1 - \frac{922267}{22166*296277} = 99.986\%$. We plot the distribution of the number of ratings in Figure 3a and then observe that it generally exhibits a power-law distribution. In Figure 3b, the x-axis indicates the number of fields involved by users, and the y-axis is the accumulative percentage of users who involved with different number of fields. From Figure 3b,



(a) Power-law distribution of Epinions (b) Accumulative percentage of users

Figure 3 Power-law distribution and accumulative percentage distribution of users

we can observe that only a small portion of users are interested in a large portion of fields. In Epinions, only 21.7% of the users expressed ratings in more than 10 fields. In other words, most users are only interested in a few fields. This observation implies that the existence of a trust relation between two users does not mean that they have common interests in a specific field. Hence, it is necessary for recommender systems to take the difference of user knowledge regarding different fields into consideration when making recommendations, which confirms our motivation of this paper. Items are divided into 27 fields and the ratings distribution across all fields is plotted in Figure 4. The total number of trust statements issued is 355,813. The average number of trust links is 16.05 per user.

In this paper, we select the top 10 fields in terms of the scale of ratings to evaluate our proposed methods. General statistics about the Epinions and its top 10 subfields data sets are summarized in Table 2. In Table 2, $\bar{r}$ and $\bar{t}$ denote the average number of ratings per user and the average number of trust links per user, respectively.

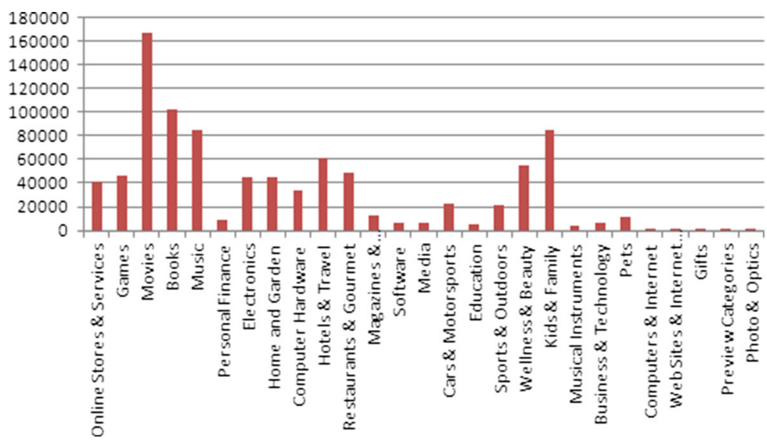


Figure 4 Rating distribution across 27 fields

Table 2 Statistics of Epinions and top 10 fields

DataSet	N	M	$\|R\|$	Density	$\bar{r}$	$\|T\|$	$\bar{t}$
Epinions	22,166	296,277	922,267	1.4×10^{-4}	41.6	355,813	16.05
Movies	14,180	28,616	167,261	4.1×10^{-4}	11.80	153,773	10.84
Books	10,731	59,129	102,975	1.6×10^{-4}	9.59	104,706	9.57
Music	9,010	34,541	85,419	2.7×10^{-4}	9.48	61,383	6.81
Kids Family	8,606	24,124	85,113	4.0×10^{-4}	9.88	68,697	7.98
Hotels Travel	10,660	15,421	61,503	3.7×10^{-4}	5.77	99,997	9.38
Wellness Beauty	7,209	22,575	55,087	3.3×10^{-4}	7.64	51,435	7.13
Restaurants Gourmet	8,376	16,642	48,112	3.4×10^{-4}	5.74	62,638	7.47
Games	8,169	8,306	45,730	6.7×10^{-4}	5.59	50,840	6.22
Electronics	11,385	15,639	45,459	2.5×10^{-4}	3.99	102,844	9.03
Home Garden	8,074	20,159	45,113	2.7×10^{-4}	5.58	62,805	7.77

5.2 Evaluation metrics

We choose two popular metrics: *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)*, to measure the recommendation quality of our proposed methods. Formally,

$$MAE = \frac{\sum_{(u,i)\in R_{test}} |\widehat{R}_{ui} - R_{ui}|}{|R_{test}|}, \tag{15}$$

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} |\widehat{R}_{ui} - R_{ui}|^2}{|R_{test}|}}, \tag{16}$$

where R_{ui} and $\widehat{R}_{ui}$ are the actual and predicted ratings, respectively, and R_{test} denotes the set of user-item pairs in test data set.

From the (15) and (16), we can see that the lower the *MAE* or *RMSE*, the better the recommendation algorithm.

5.3 Baseline approaches

In order to evaluate the effectiveness of our proposed methods, we compare our proposed methods with the following state-of-the-art approaches:

1. PMF: This method is proposed by Mnih and Salakhutdinov [18] and can be viewed as a probabilistic extension of the SVD [24] model. PMF represents the latent user and item feature vector by means of a probabilistic graphic model with Gaussian observation noise. PMF learns the latent user and item feature vector based on rating information alone.
2. SoRec: This method is proposed by Ma et al. [14]. SoRec simultaneously factorizes user-item rating matrix and social network trust matrix by sharing the latent user feature matrix. Unlike PMF, SoRec is a social network based recommendation algorithm, which fuses user-item ratings and social network trust information.
3. RSTE: This method is proposed in [15]. RSTE takes users' own tastes and their trusted friends' favors into account. It fuses both user-item ratings and social network trust information for recommendations. RSTE assumes that users' final decisions compromise between their own preferences and their trusted friends' tastes.

4. SocialMF: This method is presented by Jamali et al. [9]. It incorporates the mechanism of trust propagation into basic probabilistic matrix factorization to improve the quality of recommendation.
5. TrustMF: This method is presented in [33]. By factorizing the social trust matrix, TrustMF maps users into two latent low-dimensional spaces: truster space and trustee space, which explicitly describe the way users affect or follow the opinions of others.

5.4 Experiment settings

The main parameters settings of all comparison methods are listed in Table 3. Note that, in order to make a fair comparison, we set parameters of each method according to respective references or based on our experiments. Under these parameters settings, each method achieves its best performance. In addition, since all comparison methods use a stochastic gradient descent algorithm to optimize respective objective functions, we set the learning rate η involved in the stochastic gradient descent algorithm to be 0.03 for PMF and 0.001 for other social-based recommendation approaches. The number of dimensions K of latent feature vectors is set to 10 in all our experiments.

For each field in Epinions, we conduct a five-fold cross validation by randomly extracting different training and test sets at each time, which accounts for 80% and 20%, respectively. Finally, we report the average results on test sets for each field.

We use a PC with an Intel Xeon CPU@3.2GHz processor, 8GB memory, Windows2007 operating system and J2SE 1.7, to conduct all our experiments.

5.5 Recommendation quality

In this section, we evaluate the effectiveness of our proposed methods by comparing the performance of our proposed methods with all selected methods.

Table 4 reports recommendation quality of all comparison algorithms on top 10 fields in Epinions. We bold the best performance and use superscripts to denote the rank of our approaches among all comparison methods. It is can be seen from Table 4 that social network based approaches outperform PMF, which only utilizes user-item ratings to learn latent feature vectors. This observation is consistent with the results reported in [9, 14, 15, 33]. Except for our proposed recommendation approaches, SocialMF generally achieves the best performance among comparison methods. In addition, comparing the recommendation quality of selected baselines on "Movie", "Hotels & Travel" and "Home & Garden", we can observe that as the density of user-item rating matrix and the average number of trust links per user $\bar{t}$ simultaneously decreases, the performance rank of RSTE rises from the 5th

Table 3 Parameter settings of comparison methods

Methods	Parameter Settings
PMF	$\lambda_U = \lambda_V = 0.001$
SoRec	$\lambda_U = \lambda_V = \lambda_Z = 0.001, \lambda_C = 1$
RSTE	$\lambda_U = \lambda_V = 0.001, \alpha = 0.4$
SocialMF	$\lambda_U = \lambda_V = 0.001, \lambda_T = 1$
TrustMF	$\lambda = 0.001, \lambda_T = 1$
UKMF_DS and UKMF_PR	$\lambda_1 = \lambda_2 = 0.001, \tau = 0.6, \beta = 0.01$

Table 4 Performance comparison

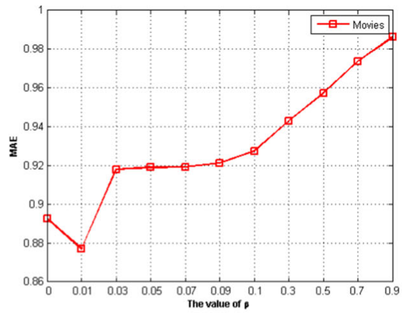
Dataset	Metric	PMF	SoRec	RSTE	SocialMF	TrustMF	UKMF_DS	UKMF_PR
Movies	*MAE*	1.2584	0.8806	0.9468	0.8781	0.9070	0.8770[2]	**0.8755**[1]
	RMSE	1.6708	1.1380	1.2240	1.1310	1.1804	1.1239[2]	**1.1201**[1]
Books	*MAE*	2.3331	0.7504	0.8291	0.7268	0.7261	0.7299[4]	**0.7256**[1]
	RMSE	2.7019	0.9918	1.0520	0.9748	0.9863	0.9598[2]	**0.9579**[1]
Music	*MAE*	1.8545	0.7593	0.8353	0.7300	0.7359	0.7380[4]	**0.7283**[1]
	RMSE	2.3021	0.9995	1.0516	0.9737	0.9881	0.9890[4]	**0.9516**[1]
Kids	*MAE*	1.5680	0.8389	0.8589	0.8103	0.8157	0.8052[2]	**0.8050**[1]
Family	*RMSE*	2.0408	1.1063	1.1132	1.0808	1.1058	1.0738[2]	**1.0545**[1]
Hotels	*MAE*	1.6338	0.8314	0.8372	0.8031	0.8165	0.8020[2]	**0.8013**[1]
Travel	*RMSE*	2.0774	1.0987	1.0959	1.0698	1.1069	1.0605[2]	**1.0435**[1]
Wellness	*MAE*	1.9871	0.9339	0.9512	0.9123	0.9577	0.9079[2]	**0.9051**[1]
Beauty	*RMSE*	2.4021	1.2250	1.2375	1.2148	1.3404	1.1948[2]	**1.1642**[1]
Restaurants	*MAE*	1.8798	0.8870	0.9150	0.8616	0.8727	0.8583[2]	**0.8580**[1]
Gourmet	*RMSE*	2.2982	1.1562	1.1899	1.1344	1.1591	1.1209[2]	**1.1186**[1]
Games	*MAE*	1.3849	0.8408	0.8388	0.8212	0.8542	0.8252[3]	**0.8130**[1]
	RMSE	1.8189	1.1091	1.0901	1.0882	1.1370	1.0896[3]	**1.0568**[1]
Electronics	*MAE*	2.4476	0.8732	0.8764	0.8501	0.8785	0.8385[2]	**0.8370**[1]
	RMSE	2.7600	1.1831	1.1540	1.1599	1.2207	1.1341[2]	**1.1195**[1]
Home	*MAE*	2.3177	0.8619	0.8657	0.8259	0.8305	0.8238[2]	**0.8225**[1]
Garden	*RMSE*	2.6986	1.1769	1.1633	1.1441	1.1857	1.1439[2]	**1.1248**[1]

place to the 3rd in terms of *RMSE*. In fact, RSTE can outperforms SocialMF when both the density of rating matrix and $\bar{t}$ further decrease. This observation implies that RSTE can generate better recommendations than others when both the rating matrix and the social trust matrix are sparse, while SocialMF generally outperforms other comparison methods when both the rating matrix and the social trust matrix are relatively dense.

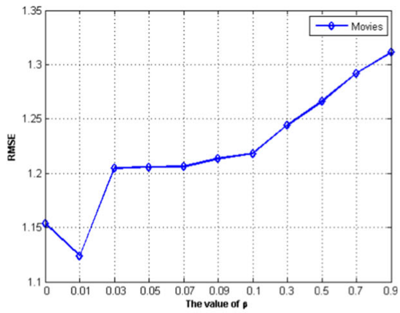
Moreover, our approaches are generally superior to other comparison methods. UKMF_DS outperforms baseline methods on 8 fields such as "Movies", "Kids & Family", "Hotel & Travel" in terms of *RMSE*. Furthermore, taking RSTE as our main comparison method since the first part of our objective function $\mathcal{L}^*$ is partly similar to that of RSTE, UKMF_DS is consistently better than RSTE on 10 fields and the improvements over RSTE are significant. This observation confirms the assumption that ignoring users' knowledge can not accurately reflect the item adoption process, while incorporating users' own knowledge levels into the processes of recommendation decision-making can largely improve the recommendation quality. In addition, UKMF_MF achieves better performance than UKMF_DS on top 10 fields. A possible reason is that user knowledge level learning by utilizing PageRank algorithm is more accurate than that by exploiting the distributions of ratings and followers across fields.

5.6 Impact of parameter β

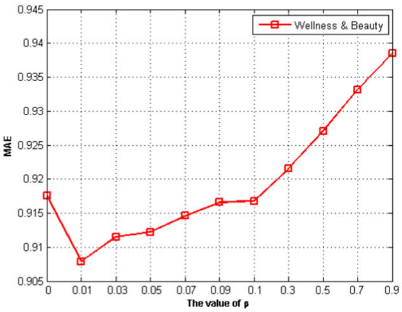
In this section, we perform a group of experiments to investigate the impact of β on the accuracy of recommendation by changing the values of β from 0 to 1. Figure 5 shows the



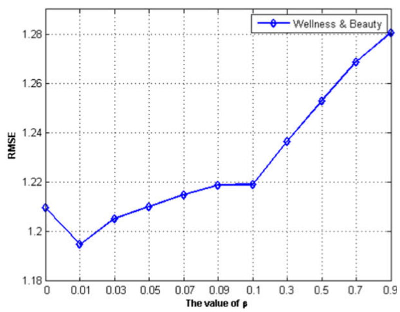
(a) Impact of β on MAE in Movies



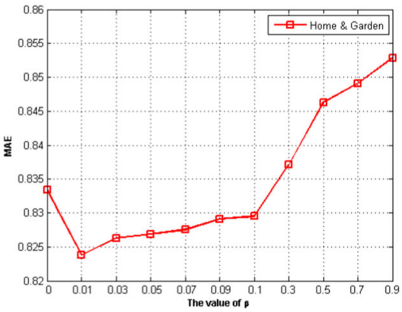
(b) Impact of β on RMSE in Movies



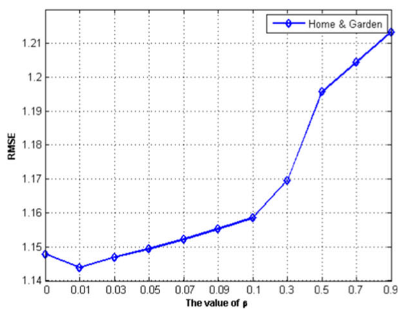
(c) Impact of β on MAE in Wellness & Beauty



(d) Impact of β on RMSE in Wellness & Beauty



(e) Impact of β on MAE in Home & Garden



(f) Impact of β on RMSE in Home & Garden

Figure 5 Impact of β on MAE and RMSE

impact of parameter β of UKMF_DS on *MAE* and *RMSE* for three fields, i.e., "Movies", "Wellness & Beauty", and "Home & Garden", which are representatives of large, middle and small scale data sets, respectively. For UKMF_PR, the change trends of *MAE* and *RMSE* on three data sets are similar to those of UKMF_DS.

From Figure 5, we can see that the value of β affects the recommendation quality significantly, which indicates that integrating the social regularization term greatly improves the accuracy of recommendation. As β increases, the values of *MAE* and *RMSE* firstly decrease, so the recommendation accuracy is accordingly improved. After β reaches a certain threshold, *MAE* and *RMSE* scores begin to increase as β increases, which means that the performance degrades when β is too large. These findings indicate that neither abandoning the social regularization term nor relying heavily on the social regularization term can produce high-quality recommendations. In addition, comparing Table 4 and Figure 5, we also observe that UKMF_DS outperforms RSTE even when $\beta = 0$ on these three fields, which demonstrates the promising future of our proposed recommendation approaches. On these three fields, our approaches achieve the best recommendation performance when β is around 0.01.

5.7 Impact of parameter τ

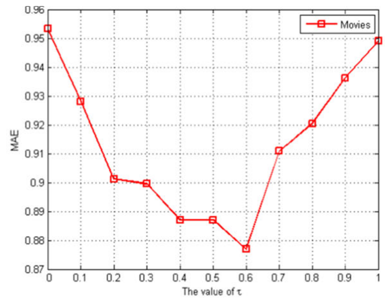
Parameter τ is another important parameter in our proposed methods, which indicates the weight of user own preferences in the processes of ratings decision-making. In order to explore the impact of τ on our proposed methods, we conduct another group of experiments and observe the changes in recommendation quality by varying the value of τ from 0 to 1. In this group of experiments, we set $\beta = 0.01$ and $K = 10$. Since UKMF_DS and UKMF_PR show similar trends, we only plot the experimental results of UKMF_DS on the aforementioned three fields in Figure 6.

As shown in Figure 6, parameter τ does have a significant impact on UKMF_DS, which indicates that it is needed to trade off between user own preferences and social influence when we decide the rating for a target item. The recommendation quality of UKMF_DS first moves toward optimal values, and then degrades as the values of τ continually increase. When τ is around 0.6, UKMF_DS achieves the best recommendation performance: *MAE*=0.8769 for "Movies", *MAE*=0.8238 for "Home & Garden" and *MAE*=0.9079 for "Wellness & Beauty". In addition, from Figure 6, we can observe that completely ignoring user own preferences or relying excessively on them cannot generate better recommendations. Moreover, in the processes of rating decision-making, the weight of user own preferences is relatively larger than those of social influence and user own knowledge since UKMF_DS gains the best recommendation quality when τ is around 0.6.

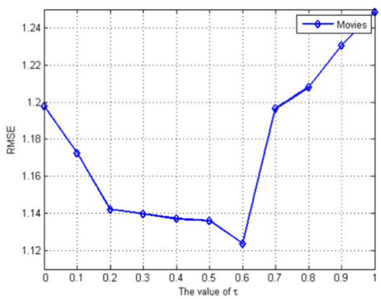
5.8 Performance on cold start users

Since some users expressed a few or no ratings in the recommender systems, recommendation algorithms generally suffer serious cold start problem. In this section, we perform another group of experiments to evaluate the effectiveness of our proposed methods on solving the cold start problem. We consider users who rated less than 5 items as cold start users. The ratios of cold start users in "Movies", "Wellness & Beauty", and "Home & Garden" are shown in Figure 7. As shown in Figure 7, more than 60% of users are cold start users in each selected dataset. In this group experiments, only cold start users are included in the test set, and we set parameters of all comparison methods according to Table 3. The experimental results are reported in Table 5.

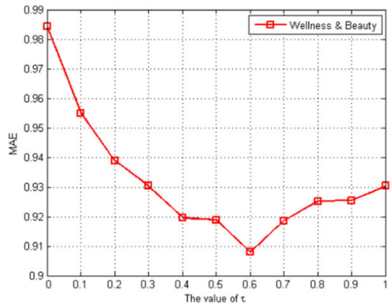
For solving the cold start problem, Table 5 demonstrates that PMF achieves the worst performance among all compared methods, which indicates that social network information is beneficial for alleviating the cold start problem. Except for our proposed methods, TrustMF is superior to other compared methods. In addition, our proposed methods generally outperform other comparison methods. In terms of *RMSE*, the improvements of UKMF_DS are 3.8%, 3.7%, 6.2% over TrustMF on "Movies", "Wellness & Beauty" and "Home & Garden", respectively. UKMF_PR improves the *RMSE* of TrustMF by 4.1%, 5.8%, and



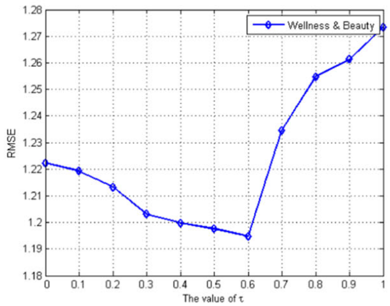
(a) Impact of τ on MAE in Movies



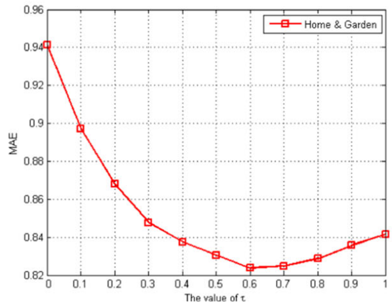
(b) Impact of τ on RMSE in Movies



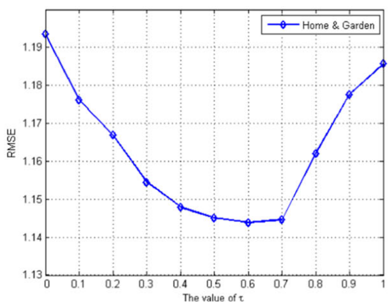
(c) Impact of τ on MAE in Wellness & Beauty



(d) Impact of τ on RMSE in Wellness & Beauty



(e) Impact of τ on MAE in Home & Garden



(f) Impact of τ on RMSE in Home & Garden

Figure 6 Impact of β on MAE and RMSE

6.3% on three selected datasets, respectively. These observations indicate that our proposed recommendation algorithms can cope with the cold start problem more better than other state-of-art methods. We argue that the main reason for the improvements is the consideration of domain-specific social relationships as well as the integration of social regularization term, which makes two users latent feature vectors closer if the corresponding users have more common interests in a specific domain. Moreover, the performance of UKMF_PR is superior to UKMF_DS on three selected datasets. Hence, we argue that PageRank algorithm is more robust than the linear combination method for the task of cold start user knowledge level learning.

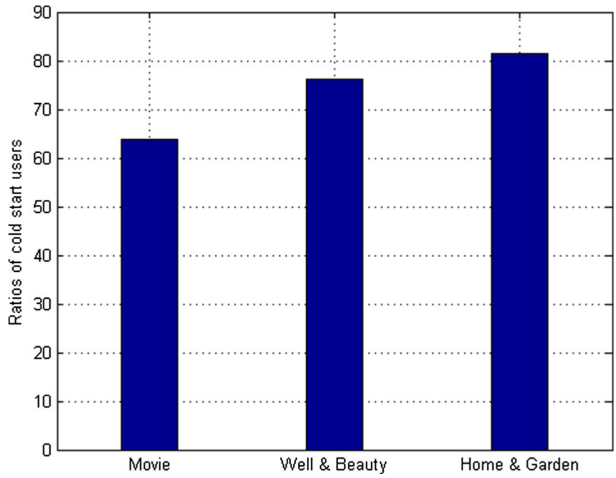


Figure 7 The ratios of cold start users

5.9 Efficiency comparison

In order to evaluate the efficiency of our proposed methods, we compare the runtime of model building of our proposed methods with those of selected baselines. In this group of experiments, the parameter settings of all comparison methods is same as Table 3. The experimental results on the aforementioned three fields are presented in Table 6.

From Table 6, we can see that the runtime of model building of PMF is minimum. This is due to the fact that PMF only utilizes ratings to learn latent feature vectors and ignores the effect of social relationships. Among social-networks-based recommendation methods, SoRec and TrustMF generally outperform RSTE, SocialMF and our proposed methods in terms of the runtime of model building. These observations indicate that the recommendation models of sharing latent user feature vectors between user-item rating matrix and social trust matrix (i.e., SoRec, TrustMF) are more efficient than the recommendation models of constraining the latent user feature vectors with social trust relationships (i.e., RSTE, SocialMF and our proposed methods). In terms of the runtime of model building, our proposed recommendation methods are superior to RSTE and SocialMF, but are inferior to SoRec and TrustMF.

Table 5 Performance comparison on cold start users

Dataset	Metric	PMF	SoRec	RSTE	SocialMF	TrustMF	UKMF_DS	UKMF_PR
Movies	*MAE*	1.6273	0.9657	1.0245	0.9662	0.8936	0.8853	0.8774
	RMSE	2.0744	1.3334	1.3035	1.3401	1.2007	1.1549	1.1513
Wellness	*MAE*	2.3602	0.9448	0.9686	0.9358	0.9098	0.9056	0.9046
Beauty	*RMSE*	2.7264	1.3633	1.2910	1.3596	1.2781	1.2308	1.2036
Home	*MAE*	2.5543	0.9603	0.9716	0.9534	0.9220	0.9113	0.9112
Garden	*RMSE*	2.8801	1.3933	1.3115	1.3910	1.3247	1.2432	1.2411

Table 6 Efficiency comparison (hours:minutes:seconds)

Dataset	PMF	SoRec	RSTE	SocialMF	TrustMF	Our Methods
Movies	00:00:03	00:02:21	01:57:29	55:02	00:02:52	**00 : 45 : 06**
Wellness Beauty	00:00:0.90	00:01:07	01:15:34	00:53:12	00:01:59	**00 : 09 : 39**
Home Garden	00:00:0.80	00:00:31	00:40:22	00:52:25	00:00:59	**00 : 07 : 37**

6 Conclusion

In this paper, assuming that the degree of social influence is different for users with different levels of knowledge, and that users' own knowledge affects the processes of their rating-making, we propose a novel recommendation algorithms by integrating the social network information and rating information as well as considering each user's knowledge. Specifically, since we cannot directly measure a user's knowledge in the field, we first use a user's status in a social network to indicate a user's knowledge in a field, and users' status is inferred from the distributions of users' ratings and followers across fields or the structure of domain-specific social network. Then, we model the final rating decision-making as a linear combination of users' own preferences, social influence and users' own knowledge. Experimental results on real-world data sets show that our proposed approach generally outperforms the state-of-the-art recommendation algorithms that ignore the knowledge level difference between the users.

We only employ simple schemes to infer users' knowledge levels. A more accurate method that computes users' knowledge levels will further improve the performance of our proposed method. Moreover, the decision-making process is complicated and is affected by several factors, such as user own preference, friends' influence, and item characteristics. At present, most research works concern the influence derived from friends and ignore the user's own personality. In fact, some users who are arrogant seldom follow others' suggestions. On the other hand, laymen are easily influenced by others. Hence, analyzing an individual's personality will be helpful for recommender systems to provide better personalized recommendations for users. Finally, location-based recommendation algorithms [35–38] have become increasingly popular in location-based social networks, applying user knowledge enhanced recommendation approach to location-based recommendation problem would be an interesting direction.

Acknowledgements The authors would like to acknowledge the support for this work from NSFC (Grant Nos. 61432008, 61503178, 61403208) and NUPTSF (Grant No. NY217114).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. TKDE **17**(6), 734–749 (2005)
2. Ahn, H.J.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inform. Sci. **178**(1), 37–51 (2008)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI (1998)
4. Cacheda, F., Carneiro, V., Fernández, D., Formoso, V.: Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. TWEB **5**(1), 2 (2011)

5. Golub, G.H., Van Loan, C.F.: Matrix Computations, vol. 3. JHU Press (2012)
6. Guo, G., Zhang, J., Yorke-Smith, N.: Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI (2015)
7. Hofmann, T.: Collaborative filtering via Gaussian probabilistic latent semantic analysis. In: SIGIR (2003)
8. Hofmann, T.: Latent semantic models for collaborative filtering. TOIS **22**(1), 89–115 (2004)
9. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys (2010)
10. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM (JACM) **46**(5), 604–632 (1999)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
12. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. Internet Comput. IEEE **7**(1), 76–80 (2003)
13. Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X.: A new user similarity model to improve the accuracy of collaborative filtering. Knowl.-Based Syst. **56**, 156–166 (2014)
14. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: CIKM (2008)
15. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: SIGIR, pp. 203–210 (2009)
16. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296 (2011)
17. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: AAAI/IAAI
18. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: NIPS (2007)
19. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIOPT **19**(4), 1574–1609 (2009)
20. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
21. Rennie, J.D., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: ICML (2005)
22. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: ICML (2007)
23. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce (2000)
24. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of dimensionality reduction in recommender system – a case study. In: WebKDD Workshop (2000)
25. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW (2001)
26. Seung, D., Lee, L.: Algorithms for non-negative matrix factorization. In: NIPS (2001)
27. Srebro, N., Rennie, J., Jaakkola, T.S.: Maximum-margin matrix factorization. In: NIPS (2004)
28. Tang, J., Gao, H., Liu, H., Das Sarma, A.: eTrust: understanding trust evolution in an online world. In: KDD, pp. 253–261 (2012)
29. Ungar, L.H., Foster, D.P.: Clustering methods for collaborative filtering. In: AAAI Workshop on Recommendation Systems (1998)
30. Wang, C., Dong, X., Zhou, F., Cao, L., Chi, C.H.: Coupled attribute similarity learning on categorical data. TNNLS **26**(4), 781–797 (2015)
31. Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: SIGIR (2005)
32. Yang, X., Steck, H., Liu, Y.: Circle-based recommendation in online social networks. In: KDD, pp. 1267–1275 (2012)
33. Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: IJCAI, pp. 2747–2753 (2013)
34. Yin, H., Cui, B., Chen, L., Hu, Z., Zhou, X.: Dynamic user modeling in social media systems. TOIS **33**(3), 1–44 (2015)
35. Yin, H., Cui, B., Huang, Z., Wang, W., Wu, X., Zhou, X.: Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 819–822. ACM (2015)
36. Yin, H., Cui, B., Chen, L., Hu, Z., Zhang, C.: Modeling location-based user rating profiles for personalized recommendation. TKDD **9**(3), 1–41 (2015)

37. Yin, H., Cui, B., Zhou, X., Wang, W., Huang, Z., Sadiq, S.: Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. TOIS **35**(2), 1–44 (2016)
38. Yu, Y., Chen, X.: A survey of point-of-interest recommendation in location-based social networks. In: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
39. Yu, Y., Wang, C., Gao, Y., Cao, L., Chen, X.: A coupled clustering approach for items recommendation. In: PAKDD, pp. 365–376 (2013)
40. Yu, Y., Gao, Y., Wang, H., Wang, R.: Joint user knowledge and matrix factorization for recommender systems. In: WISE, pp. 77–91. Springer (2016)
41. Yu, Y., Wang, C., Wang, H., Gao, Y.: Attributes coupling based matrix factorization for item recommendation. Appl. Intell. **46**(3), 521–533 (2017)