

A novel social network hybrid recommender system based on hypergraph topologic structure

Xiaoyao Zheng^{1,2,3} · Yonglong Luo^{2,3} · Liping Sun^{2,3} ·
Xintao Ding^{2,3} · Ji Zhang⁴

Received: 20 July 2016 / Revised: 14 May 2017 / Accepted: 25 August 2017 /
Published online: 12 September 2017
© Springer Science+Business Media, LLC 2017

Abstract With the advent and popularity of social network, more and more people like to share their experience in social network. However, network information is growing exponentially which leads to information overload. Recommender system is an effective way to solve this problem. The current research on recommender systems is mainly focused on research models and algorithms in social networks, and the social networks structure of recommender systems has not been analyzed thoroughly and the so-called cold start problem has not been resolved effectively. We in this paper propose a novel hybrid recommender system called Hybrid Matrix Factorization(HMF) model which uses hypergraph topology to describe and analyze the interior relation of social network in the system. More factors including contextual information, user feature, item feature and similarity of users ratings are all taken into account based on matrix factorization method.

✉ Yonglong Luo
ylluo@ustc.edu.cn

Xiaoyao Zheng
zxiaoyao_2000@163.com

Liping Sun
slp620@163.com

Xintao Ding
xintaoding@163.com

Ji Zhang
Ji.Zhang@usq.edu.au

¹ College of Territorial Resources and Tourism, Anhui Normal University, Wuhu 241002, China

² School of Mathematics and Computer Science, Anhui Normal University, Wuhu 241002, China

³ Anhui Provincial Key Laboratory of Network and Information Security, Wuhu 241002, China

⁴ Faculty of Health, Engineering and Science, University of Southern Queensland, Queensland, Australia

Extensive experimental evaluation on publicly available datasets demonstrate that the proposed hybrid recommender system outperforms the existing recommender systems in tackling cold start problem and dealing with sparse rating datasets. Our system also enjoys improved recommendation accuracy compared with several major existing recommendation approaches.

Keywords Recommender system · Hypergraph · Hybrid approaches · Cold start problem

1 Introduction

With the exponential growth of network information, improving the utilization efficiency of information and alleviating the problem of information overload become a pressing research issue. Recommender systems provide a promising solution to the above problems [2], and plays an important role in e-commerce, information retrieval, e-tourism, online advertising, mobile applications and so on [3, 19, 24]. For example, according to a survey from VentureBeat, 35% sales of the online retail giant Amazon are materialized by its recommender system, and the researchers of the online DVD rental company Netflix pointed out in their 2011 report that 60% users are able to find their interested movies and videos from the company's recommender system.

With the development of online social network, people are accustomed to commenting items such as commodities that they have purchased or movies that they have seen before in online social network, and shared their experiences with friends. Through these comments, other people can assess the quality of items and choose suitable or interested items according to their own preference. From the perspective of sociology and psychology, the choice of item selection is affected by many factors. For example, before choosing a movie to watch, one may be affected by a variety of factors such as his own previous experience, movie trailers, his favorite types of movie or friends' recommendations, and even his own feelings and environment at that time. All these factors exert an influence to different degree on their final choice. In order to improve the accuracy of recommender systems, the recommended model used should take into account as many important factors as possible [8, 12, 25, 35], leading to a model with comprehensive information to improve the accuracy of the model.

Generally speaking, two types of recommender systems have been investigated in literature: the *content-based systems* and the *collaborative filtering-based systems*. The content-based recommendations originated from the area of information retrieval [51]. Content-based recommender systems use content information of items to find the match between items and users. In [52], the keywords of purchased books of a user are used to find other books that contain the same or similar keywords for recommendation. Content-based recommendations typically require external information that might not be available or easy to collect, which leads to a major disadvantage of the content-based systems that they may not have a satisfactory accuracy. Yet, the traditional content-based approaches have advantages in some aspects, such as data sparsity and new item problem [3].

Unlike the content-based recommendation methods, the collaborative recommender systems try to predict the utility of items for a particular user based on the items previously rated by other users. When users have supplied sufficient ratings, the collaborative recommender systems are generally more accurate than the content-based techniques, as evidenced by [23, 27, 29, 50, 51]. Several collaborative filtering-based systems have recently been proposed to improve recommendation accuracy by using social trust [12, 16, 17, 30, 48, 51, 52].

Yang et al. proposed to use the concept of inferred trust circle based on social network to recommend user favorite items [48]. In [17], M. Jamali et al. incorporated the mechanism of trust propagation into the recommendation model. In [12], a user model with trust and distrust networks was proposed to identify trustworthy users. Due to the preference similarity, user latent features should be similar to his/her friends based on the probabilistic matrix factorization model [33, 47, 49, 54]. In [47], Qian et al. proposed a personalized recommender model (PRM) combining user interpersonal interest similarity, interpersonal influence and personal interest factor. The main contribution of this paper is the proposal of a new method to make use of the category information of products, and user personal interest. In [49], Hu et al. proposed a factor model treating the data as an indication of positive and negative preference associated with vastly varying confidence levels. Zhao et al. proposed a user-service rating prediction approach by exploring social users rating behaviors [54]. In their work, four factors including user personal interest, interpersonal interest similarity, interpersonal rating behavior similarity and interpersonal rating behavior diffusion were fused into the matrix factorization model. Many recent studies have begun to mine text reviews to improve prediction accuracy for collaborative filtering recommender system. McAuley et al. proposed a HFT model which combines latent rating dimensions with latent review topics [31]. Lei and Qian et al. proposed a sentiment-based rating prediction method (RPS) to improve prediction accuracy in recommender systems which use LDA to mine the topic of reviews [25]. Jiang et al. proposed an author topic model-based collaborative filtering (ATCF) method to facilitate comprehensive points of interest (POIs) recommendations for social users [20]. Additionally, contextual information is exploited to reduce the prediction error in collaborative filtering-based system. In [26], Liu et al. handled contextual information by applying random decision trees to partition the original user-item rating matrix such that the ratings with similar contexts are grouped based on matrix factorization method. A. Akther et al. designed a new architecture for user personalization which combines both social network data and context data [4]. Zhao et al. proposed a LBRP model which make a full use of mobile users location sensitive characteristics to carry out rating prediction [53]. Their work demonstrates that human's rating behaviors are significantly affected by the contextual information of their geographical locations. A major appeal of the collaborative filtering-based systems is that they can deal with any kinds of content and recommend any items [1, 14, 32, 42], and they can address data aspects that are often elusive and difficult to profile when using the content-based systems. However, there are two critical issues in the collaborative recommender systems regarding new users and new items, which are termed the cold start problem. Because newly added users have few or no ratings in online social networks, it's difficult for recommender systems to quantify users' preferences.

In order to overcome the shortcomings of the content-based and collaborative filtering-based systems, several recommender systems use hybrid approaches by combining content-based and collaborative filtering-based mechanisms [7, 22, 26]. In [50], K. Yoshii et al. proposed an efficient hybrid music recommender system, and Wei et al. developed a market-based recommender system that allows multiple agents to compete with each other to present their best recommendations to users [35]. In [10], Cao et al. presented another novel idea and proposed a hybrid collaborative filtering algorithm for bidirectional Web service recommendation which confirms that the collaborative recommendation is more efficient once again. In [9], Braunhofer proposed a context-aware recommender system using various hybridization techniques to improve the prediction accuracy. In the area of e-commerce, Chen considered both the overall ratings and feature-level opinion values to identify reviewers preference homogeneity [11]. In [12, 14, 17, 23, 47], the cold start problem is solved

to different degree, but is still far from sufficient, because the existing methods only use social relations such as social trust, neighbor relations, etc, to help the recommender system to alleviate the cold start problem, without taking into account the useful content-based information about the features of users and items involved.

In this paper, we incorporate some important content-based characteristics into a collaborative approach based on matrix factorization method. We proposed a novel social network hybrid recommender system based on hypergraph topologic structure. For simplicity of presentation, we call it the HMF model (stands for Hybrid Matrix Factorization model). The technical contributions of this paper are summarized as follows:

1. We propose a novel HMF model which fuses important contextual information, including users ratings, item feature, user feature, into the matrix factorization approach;
2. We build a hypergraph topologic structure to represent the social network and theoretically analyze the basic concepts of the HMF model;
3. We use the k -Modes algorithm to cluster user-item datasets based on the contextual information which can effectively enhance the data correlation of sub-datasets and improve the recommendation accuracy;
4. Extensive experimental evaluation on publicly available datasets demonstrates that our proposed hybrid recommender system outperforms the existing recommender systems in tackling cold start problem and dealing with sparse rating datasets. Our system also enjoys improved recommendation accuracy compared with several major existing recommendation approaches.

The rest of the paper is organized as follows. We describe the related work and several typical recommender models in Section 2. In Section 3, we present a hypergraph structure in social networks and describe the relations through mathematical definitions. On the basis of the hypergraph structure introduced in Section 3, we build a comprehensive recommender model and a corresponding training method in Section 4. We present experimental evaluation results of our recommender system which involves a performance comparison with several major existing recommender models in Section 5. The paper is concluded with a summary and a discussion on future research directions in the final section.

2 Related work and preliminaries

In this section, we first present a survey on the recent related work. Then, several major recommendation approaches in the domain are introduced which are mostly relevant to our work and are all based on the technique of matrix factorization.

2.1 Related work

Recommender systems are generally classified into three categories: *content-based systems*, *collaborative filtering-based systems*, and *hybrid systems* [2]. Content-based recommender systems mainly select highly similar items with respect to user preference [14, 36, 40]. Collaborative filtering-based technology is a more efficient approach for recommender systems [1, 7, 34, 39, 47, 51]. Based on the rating profiles of the target user, collaborative filtering-based recommender systems calculate profile similarity between the target user and the existing users, and recommend the items of the interest of the similar existing users to the target user. The hybrid recommender systems are designed to overcome the limitations of single recommendation technology [5, 9, 50]. Such hybrid systems typically combine a

collaborative filtering-based approach with a content-based approach in order to produce more accurate recommendation.

In recent years, matrix factorization method is applied to collaborative filtering recommendation, which effectively improves the accuracy of the recommender systems. In [28], Lu et al. demonstrated that mine structural information of the original data can improve efficiency and lessen the interference of noise to some extent based on matrix factorization. So, matrix factorization techniques is employed by collaborative recommender systems, and these methods have become popular in recent years featuring good scalability [13, 17, 24, 29, 39, 42, 43]. In [33], Mnih et al. first presented a probabilistic matrix factorization model without taking into account any social factors. Subsequently, many researchers incorporate social factors such as social trust, interpersonal influence, geographical location, user sentiment and so on into matrix factorization model [6, 25, 30, 31, 47, 48, 53, 54]. These studies show that social factors can improve the accuracy of recommendations. Besides the aforementioned models, there are a body of variant models based on matrix factorization method. A distributed recommendation algorithm was proposed in [46]. This algorithm enjoys a better performance in terms of time and when dealing with the cold start problem. In addition, Liu et al. presented a recommender system which employed the matrix factorization techniques and aided context-aware information [26]. In [48], Yang et al. proposed circle-based recommendation models which is a novel idea based on matrix factorization method. The accuracy of the models mentioned above is impressive when an abundance of rating information is available. But when the users only have few ratings, which is the case for most real-world sparse datasets [21, 37, 44], the performance of these models becomes unsatisfactory. However, these methods metioned above need enough ratings data or text reviews to get the explicit or implicit social information. So they still trapped in the cold start to some extent.

2.2 Compared algorithms

In this section, we will review several typical recommendation techniques based on matrix factorization method.

2.2.1 BaseMF model

To start with, we briefly review the basic low-rank matrix factorization (BaseMF) approach [33], which does not take any social factors into consideration. This method can efficiently handle large datasets. In comparison with the traditional collaborative filtering methods, BaseMF features a better performance. In details, BaseMF model is trained on the observed rating data by minimizing the following objective function:

$$\Psi(R, P, Q) = \frac{1}{2} \sum_{(u_i, s_j) \in T} (R_{ij} - \hat{R}_{ij})^2 + \frac{\lambda}{2} (\|P\|_F^2 + \|Q\|_F^2) \quad (1)$$

$$\hat{R} = r + PQ^T \quad (2)$$

Where T denotes the training dataset, R_{ij} is the real rating values for item s_j from user u_i in T , P and Q are the user and item latent feature matrices, λ is the regularization constant. $\|P\|_F$ and $\|Q\|_F$ are the Frobenius norm of matrices P and Q , respectively, and r denotes the mean rating value of T . This objective function can be minimized efficiently using gradient descent method. Once the low-rank matrices P and Q have been obtained, rating values can be predicted according to (2) for any user-item pair (u_i, s_j) .

2.2.2 STE model

The STE Model [30] is a linear combination of BaseMF and a social network based approach which taken the social trust into account. The objective function of STE Model is defined as follows.

$$\Psi(R, P, Q, S) = \frac{1}{2} \sum_{(u_i, s_j) \in T} \left(R_{ij} - g(\alpha P_i Q_j^T \hat{R}_{ij} + (1 - \alpha) \sum_{v \in N_i} S_{iv} P_i Q_j^T) \right)^2 + \frac{\lambda}{2} (\|P\|_F^2 + \|Q\|_F^2) \tag{3}$$

Where S_{iv} denotes the trust degree of u_i from his friends u_v and parameter α controls the effects of neighbors on predicted rating. $g(x)$ is the logistic function $g(x) = 1/(1 + \exp(-x))$, which makes it possible to bound the range of $P_i Q_j^T$ within the range of $[0, 1]$. Experiments show that STE model outperforms the BaseMF model in terms of accuracy.

2.2.3 SocialMF model

The SocialMF [17] incorporates the mechanism of trust propagation into the model based on the matrix factorization technique. This model outperforms BaseMF and STE [30] in terms of Root Square Mean Error (RSME) and the efficacy in dealing with the cold start problem. The objective function of SocialMF Model is defined as follows.

$$\Psi(R, P, Q, T) = \frac{1}{2} \sum_{(u_i, s_j) \in T} (R_{ij} - \hat{R}_{ij})^2 + \frac{\lambda}{2} (\|P\|_F^2 + \|Q\|_F^2) + \frac{\alpha}{2} \sum_T \left(\left(P_i - \sum_{u_v \in N_{u_i}} T_{iv} P_v \right) \left(P_i - \sum_{u_v \in N_{u_i}} T_{iv} P_v \right)^T \right) \tag{4}$$

Where T_{iv} denotes the degree of trust of user u_i on user u_v , T_{iv} is a positive value $T_{iv} \in [0, 1]$, and each row of the trust matrix T is normalized to 1. Compared with (1), the factor of trust propagation is enforced by the last term in (4), which means the user latent feature P_i should be similar to his/her friends latent feature P_v . The trade-off between the ratings and the trust propagation is determined by a weight $\alpha \geq 0$. If $\alpha = 0$, the influence of trust propagation is ignored and this model will degenerate back to BaseMF model as a result.

Equation (4) can be optimized by gradient descent approach as well. Once the model is trained, the latent feature matrix P is constrained by the last term in (4) and the degree of trust T_{iv} enhances the latent feature similarity between user u_i and u_v . The rating value for any user concerning any item can be predicted according to (2).

2.2.4 CircleCon model

The CircleCon model [48] uses inferred circle which is a trust sub-network derived from the trust relationship in social network, and the model applies each such circle to a single category of items. Therefore, the objective function of the CircleCon model is similar to

that of SocialMF, but differs in that the CircleCon model being trained in each category separately. The objective function of the CircleCon model is defined as follows.

$$\Psi^c (R^c, P^c, Q^c, T^{c*}) = \frac{1}{2} \sum_{(u_i, s_j) \in T} (R_{ij}^c - \hat{R}_{ij}^c)^2 + \frac{\lambda}{2} (\|P^c\|_F^2 + \|Q^c\|_F^2) + \frac{\alpha}{2} \sum_T \left(\left(P_i^c - \sum_{u_v \in N_{u_i}} T_{iv}^{c*} P_v^c \right) \left(P_i^c - \sum_{u_v \in N_{u_i}} T_{iv}^{c*} P_v^c \right)^T \right) \tag{5}$$

Where c is the category of items related to a inferred circle and T_{iv}^{c*} denotes the normalized interpersonal trust of u_v to u_i in category c . The CircleCon model enjoys a higher prediction accuracy compared with the SocialMF model.

2.2.5 ContextMF model

The ContextMF model is proposed in [18], which fused the social context including individual preference and interpersonal influence into the probabilistic matrix factorization method used. Individual preference represents the quantitative value for how strong a user likes different items, and interpersonal influence denotes degree of relationships between the user and the item senders. In the ContextMF model, the individual preference similarity is represented by matrix \mathbf{W} and interpersonal influence is denoted by matrix \mathbf{S} . The objective function of this model is defined as follows.

$$\Psi (R, P, Q, S^*, W^*) = \frac{1}{2} \sum_{(u_i, s_j) \in T} (R_{ij} - \hat{R}_{ij})^2 + \frac{\lambda}{2} (\|P\|_F^2 + \|Q\|_F^2) + \frac{\alpha}{2} \sum_T \left(\left(P_i - \sum_{u_v \in N_{u_i}} S_{iv}^* P_v \right) \left(P_i - \sum_{u_v \in N_{u_i}} S_{iv}^* P_v \right)^T \right) + \frac{\beta}{2} \sum_T (W_{iv}^* - P_i P_v^T)^2 \tag{6}$$

Where S_{iv}^* denotes the normalized interpersonal influence of u_v to u_i and W_{iv}^* denotes the normalized individual preference similarity between u_i and u_v . α and β are the tradeoff parameters which represent the strength of factors of individual preference and interpersonal influence.

2.2.6 PRM model

PRM model [47] takes more comprehensive factors into account. More specifically, it combines the advantages of different models including BaseMF model, CircleCon model [48] and ContextMF model [18]. Additionally, three social factors, i.e., personal interest,

interpersonal interest similarity and interpersonal influence, are incorporated into the PRM model which is shown in (7).

$$\begin{aligned}
 \Psi^c(R^c, U^c, P^c, S^{c*}, W^{c*}, Q^{c*}) = & \frac{1}{2} \sum_{u,i} (R_{u,i}^c - \hat{R}_{u,i}^c)^2 + \frac{\lambda}{2} (\|U^c\|_F^2 + \|P^c\|_F^2) \\
 & + \frac{\beta}{2} \sum_u \left(\left(U_u^c - \sum_v S_{u,v}^{c*} U_v^c \right) \left(U_u^c - \sum_v S_{u,v}^{c*} U_v^c \right)^T \right) \\
 & + \frac{\gamma}{2} \sum_u \left(\left(U_u^c - \sum_v W_{u,v}^{c*} U_v^c \right) \left(U_u^c - \sum_v W_{u,v}^{c*} U_v^c \right)^T \right) \\
 & + \frac{\eta}{2} \sum_{u,i} |H_u^{c*}| (Q_{u,i}^{c*} - U_u^c P_i^{cT})^2 \tag{7}
 \end{aligned}$$

In PRM model, items are first classified into several categories according to tree structure of items. Then, it uses the interpersonal trust and similarity of interest circle which both have effect on user latent feature matrix. Finally, it takes into account the influence of user personal interest for item latent feature. By doing above, a more sophisticated recommendation can be produced.

2.2.7 Our Hybrid matrix factorization(HMF) model

Our HMF model absorbs the advantage of hybrid recommendation and takes more social factors into consideration. It builds on the structure of hypergraph and consider not only the relation between user but also the relation between items, users preference for items and the contextual information when users are selecting items.

3 Basic concepts and definitions on hypergraph

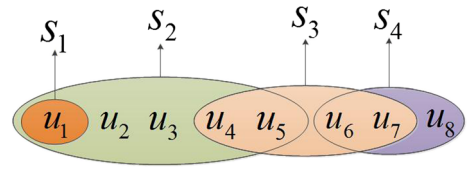
The key tasks of recommender systems are to establish binary relationship between users and items, and employ user ratings or similarity of users interest to identify potential interesting items for users so as to achieve personalized recommendation. In this paper, we use hypergraph to represent the binary relation between users and items [15]. In the following, we present some basic definitions based on hypergraph topological structure.

Definition 1 Item Centered Hypergraph. Suppose that a social network has m items and n users, the binary relationship between item and user can be expressed as $H = (S, E_S)$, where $S = \{s_1, s_2, \dots, s_m\}$ denotes the set of items and $E_S = \{es_1, es_2, \dots, es_m\}$ denotes

Table 1 Ratings for item from users

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
s_1	3							
s_2	4	5	4	5	5			
s_3				4	3	4	5	
s_4						5	5	2

Figure 1 Item Centered Hypergraph



hyperedge set and satisfies the condition $\bigcup_{i=1}^m es_i = U$, where $U = \{u_1, u_2, \dots, u_n\}$ denotes the set of users. We call this graph as item centered hypergraph. According to the data in Table 1, we have $H = (\{s_1, s_2, s_3, s_4\}, \{\{u_1\}, \{u_1, u_2, u_3, u_4, u_5\}, \{u_4, u_5, u_6, u_7\}, \{u_6, u_7, u_8\}\})$. The item centered hypergraph is shown in Figure 1.

Definition 2 User Centered Hypergraph. Suppose that an item centered hypergraph is $H = (S, E_S)$, and its dual graph denoted by $H^* = (U, E_U)$, where $E_U = \{eu_1, eu_2, \dots, eu_n\}$ denotes hyperedge set and meets the condition $\bigcup_{i=1}^n eu_i = S$. We call this graph as user centered hypergraph. Again, according to the data in Table 1, we have

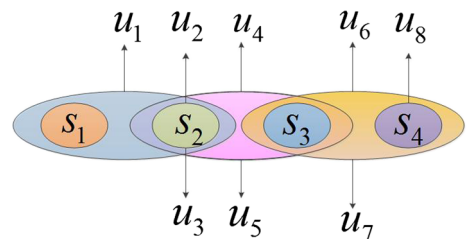
$$H^* = (\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}, \{\{s_1, s_2\}, \{s_2\}, \{s_2, s_3\}, \{s_2, s_3\}, \{s_3, s_4\}, \{s_3, s_4\}, \{s_4\}\})$$

The user centered hypergraph is shown in Figure 2.

Definition 3 Item Feature Space. It is described by a vector $\vec{f}_s = (f_{s_1}, f_{s_2}, \dots, f_{s_{l_s}})$, where l_s denotes the dimensionality of \vec{f}_s . The value of $f_{s_i} (i \leq l_s)$ uses quantitative value according to [24]. The item feature space of item s_j can be expressed as $\vec{f}_{s_j} = (f_{s_{j1}}, f_{s_{j2}}, \dots, f_{s_{j_{l_s}}})$. For example, the Item Feature Space of the movie Avatar can be something like $\vec{f}_s(\text{Avatar}) = (\text{science fiction movie, America, 2009, James Cameron, Twentieth Century Fox Film Corporation, Sam Worthington, Zoe Saldana, Oscar})$ which represents the item feature elements of film types, production area, publication year, production company, leading actor, leading actress and the award received, respectively.

Definition 4 User Feature Space. It is described by a vector $\vec{f}_u = (f_{u_1}, f_{u_2}, \dots, f_{u_{l_u}})$, where l_u denotes the dimensionality of \vec{f}_u . The value of $f_{u_i} (i \leq l_u)$ uses quantitative value according to [24]. The user feature space of user u_i can be expressed as $\vec{f}_{u_i} = (f_{u_{i1}}, f_{u_{i2}}, \dots, f_{u_{i_{l_u}}})$. For example, user u1 can be expressed as $\vec{f}_u(u_1) =$

Figure 2 User Centered Hypergraph



(male, 31, China, Engineer, Married) which represents the user feature elements of gender, age, nationality, job and marital status, respectively.

Definition 5 Rating Function. Assume that user u_i provides rating on item s_j , which is denoted by $r_{ij} = Rank(u_i, s_j)$, where $r_{ij} \in R$, and R is the rating value set, denoted by $R = \{1, 2, \dots, NUM\}$. Without losing generality, we set $NUM = 5$. Figure 3 shows a rating-weighted hypergraph, both of which are based on the data in Table 1.

Definition 6 User Neighbor. If user u_i and user u_v have rated some identical items, that is $eu_i \cap eu_v \neq \emptyset$ in $H^* = (U, E_U)$ with $eu_i \in E_U$ and $eu_v \in E_U$, then u_i and u_v is a user neighbor of each other. The neighbor of user u_i is denoted by N_{u_i} . For example, u_1 is a user neighbor of u_2 in Table 1 and vice versa.

Definition 7 Adjacent Item. Assume that the users who have rated item s_j are denoted by U_{s_j} . Then all the items which have been rated by the users in U_{s_j} is referred to as adjacent items of item s_j , which is denoted by M_{s_j} . For Instance, the adjacent items of s_3 in Table 1 is $\{s_2, s_4\}$.

Definition 8 Rating Contextual Information. It represents those related aspects of information when user u_i chooses item s_j , which is denoted by a vector $\vec{C} = (c_1, c_2, \dots, c_{l_c})$ with each vector component $c_t (t = 1, 2, \dots, l_c)$ representing one aspect of contextual information, such as temperature, time, location and so on.

Definition 9 Rating Contribution. The function of rating contribution for item s_j from user u_i is defined as $d_{ij} = Contribution(u_i, s_j) = \frac{Rank(u_i, s_j)}{\sqrt{\sum_{i=1}^n (Rank(u_i, s_j))^2}}$. We can employ rating contribution to build a rating contribution weighted hypergraph that is demonstrated in Figure 4 based on the data in Table 1.

Definition 10 User Rating Similarity. Assume that the item set $I_c = \{s_1, s_2, \dots, s_n\}$ which has both been rated by user u_i and user u_j . Then the user rating similarity between u_i and u_j

is defined as $Sim_C(u_i, u_j) = \frac{\sum_{k=1}^n (d_{ik} - \bar{d}_i)(d_{jk} - \bar{d}_j)}{\sqrt{\sum_{k=1}^n (d_{ik} - \bar{d}_i)^2 \sum_{k=1}^n (d_{jk} - \bar{d}_j)^2}}$, where $\bar{d}_i = (1/|I_c|) \sum_{s_j \in I_c} d_{ij}$

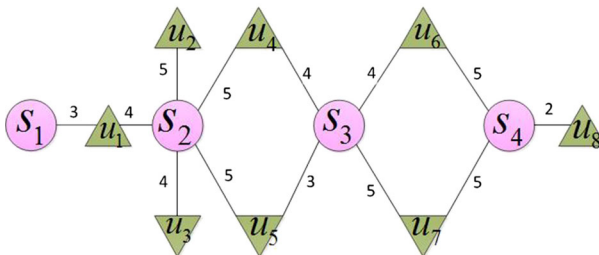


Figure 3 Rating-weighted hypergraph

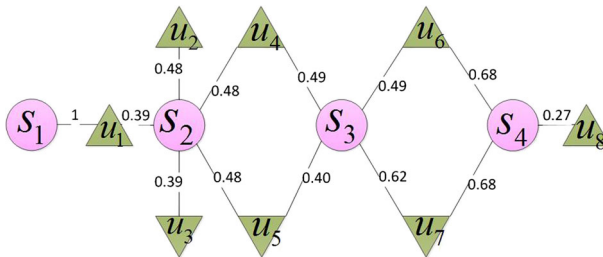


Figure 4 Rating Contribution Weighted Hypergraph

and the item set which has been rated by user u_i is denoted by I_c . If $I_c = \emptyset$, we will assign the value of \bar{d}_i as 0. In this paper, we use SC_{ij} to represent $Sim_C(u_i, u_j)$.

Definition 11 Item Feature Similarity. The function of item feature similarity between item s_i and item s_j defined as the following equation using cosine measure: $Sim_S(s_i, s_j) =$

$$\cos(\vec{f}_{s_i}, \vec{f}_{s_j}) = \frac{\sum_{k=1}^{I_s} f_{s_{ik}} f_{s_{jk}}}{\sqrt{\sum_{k=1}^{I_s} f_{s_{ik}}^2} \sqrt{\sum_{k=1}^{I_s} f_{s_{jk}}^2}}. \text{ In this paper, we use } SS_{ij} \text{ to represent } Sim_S(s_i, s_j).$$

Definition 12 User Feature Similarity. The user feature similarity between user u_i and user u_j is defined as follows using cosine measure: $Sim_U(u_i, u_j) = \cos(\vec{f}_{u_i}, \vec{f}_{u_j}) =$

$$\frac{\sum_{k=1}^{I_u} f_{u_{ik}} f_{u_{jk}}}{\sqrt{\sum_{k=1}^{I_u} f_{u_{ik}}^2} \sqrt{\sum_{k=1}^{I_u} f_{u_{jk}}^2}}. \text{ In this paper, we use } SU_{ij} \text{ to represent } Sim_U(u_i, u_j).$$

4 Our recommendation model

In our proposed model HMF, we employ matrix factorization method to obtain the optimal latent features for users and items using the stochastic gradient decent approach. The objective function used in BaseMF model, as in (1), only involves the item and user latent features, and it does not take into account other factors which are contextual information, relation between users and items, relation between items and relation between users. In this paper, we will combine various related factors, which are about three object entities and four relations shown in Figure 5, to produce hybrid recommendations based on BaseMF to achieve a better performance.

4.1 Fusing factors into HMF model

The proposed HMF model fuses four factors: contextual information, user rating similarity, item feature and user feature. Firstly, we introduce the approach by adding contextual information. Then we infer the objective function of the proposed HMF model by adding other three factors. In this way, we will represent our sophisticated model step by step in an incremental manner.

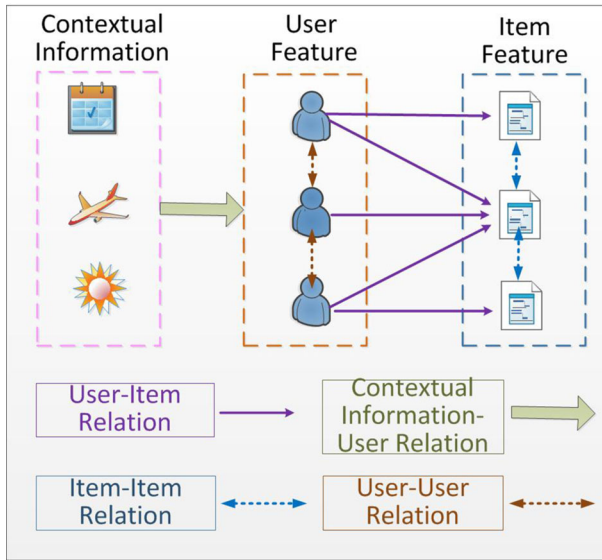


Figure 5 Three entities in recommender system, including contextual information, user feature and item feature. Four types relation between the entities which are user-item relation, item-item relation, user-user relation and contextual information-user relation

4.1.1 Stage 1: Adding contextual information

Contextual information has been proven to be valuable information for building accurate recommender system [26]. We handle contextual information by applying clustering algorithm to partition the original user item-rating matrix such that the ratings with similar contextual information are grouped. The contextual information is typically associated with the process when users are selecting items. The contextual information space in this paper is represented by vector $C = (c_1, c_2, \dots, c_{l_c})$, where each vector component c_t is binary taking the value of 0 or 1. For instance, c_1 may denote Weekend, c_2 may denote Daylight, and c_3 may denote Alone. In this way, the contextual information that Bob went to see the movie Avatar with his friends at 9:00 PM on Saturday can be expressed as (1,0,0). In this way, we can express contextual information in a unified form. We can then cluster the contextual information so that the contextual information which has high correlation can be grouped together in one cluster. In this paper, the k -Modes algorithm which has a good performance for discrete data is applied to cluster contextual information.

Recall that in (1), the optimization of the objective function Ψ is carried out based on the entire training dataset. In order to improve the optimization performance of the object function Ψ , we first group the training dataset into k clusters using the k -Modes Algorithm. Then, the object function Ψ can be modified accordingly as follows:

$$\Psi \left(R^{T_x}, P^{T_x}, Q^{T_x} \right) = \frac{1}{2} \sum_{(u_i, s_j) \in T_x} \left(R_{ij}^{T_x} - \hat{R}_{ij}^{T_x} \right)^2 + \frac{\lambda}{2} \left(\left\| P^{T_x} \right\|_F^2 + \left\| Q^{T_x} \right\|_F^2 \right) \quad (8)$$

$$\hat{R}_{ij}^{T_x} = r^{T_x} + P^{T_x} \left(Q^{T_x} \right)^T \quad (9)$$

Where T_x denotes the x^{th} cluster of training dataset. $R_{ij}^{T_x}$ and $\hat{R}_{ij}^{T_x}$ represent the actual and predicted rating for item s_j from user u_i . r^{T_x} denotes the mean value of user rating in the x^{th} cluster of the training dataset.

After partitioning the training data into k clusters, we then use objective function in (8) to get the optimal latent features for users and items iteratively. There are several salient advantages for this approach. Firstly, the high density rating data is gathered into a sub-matrix, whereby the irrelevant rating data or those with low degree of correlation are removed. This can contribute to improving the accuracy of the recommender system. Secondly, through data clustering, the training data is divided into k clusters which can effectively reduce the complexity of matrix operations.

4.1.2 Stage 2: Adding user rating similarity

In real-life scenarios, many users choose to purchase an item or see a movie through friends' recommendations. Therefore, using friends circle recommendation [41] can improve the accuracy of recommender systems and resolve the limited content problem in content-based recommendation models. The experiment results in [48] show that the circle-based recommendation model outperforms the BaseMF model in terms of RMSE. The difference of preference and interest between friends will affect the user latent feature. In this paper, we extract the user rating similarity under different contextual scenarios to optimize the mode in (8). Using inferred social circles of those friends exhibiting similar taste, we incorporated the user rating similarity into our model.

According to [17], zero mean Gaussian priors are assumed for user and item latent feature vectors: $p(P|\sigma_p^2) = \prod_{u_i} N(P_i|0, \sigma_p^2 I)$, $p(Q|\sigma_Q^2) = \prod_{u_i} N(Q_j|0, \sigma_Q^2 I)$ By adding user rating similarity to the user latent feature vectors, we can have the conditional distributions given the user latent features of his neighbors:

$$p\left(P^{T_x} | SC^{T_x}, \Omega^{T_x}\right) = \prod_{u_i} N\left(P_i^{T_x} \mid \sum_{u_v \in N_{u_i}^{T_x}} SC_{iv}^{T_x*} P_v^{T_x}, \Omega^{T_x}\right) \tag{10}$$

Where Ω^{T_x} denotes the zero-mean spherical Gaussian priors in x^{th} cluster, $N_{u_i}^{T_x}$ denotes user u_i neighbors in the x^{th} cluster and $SC_{iv}^{T_x*}$ is the normalization of $SC_{iv}^{T_x}$.

According to [17], through a Bayesian inference, the posterior probability of the latent feature vectors can be obtained as follows:

$$\begin{aligned} & p\left(P^{T_x}, Q^{T_x} | R^{T_x}, SC^{T_x*}, \Omega^{T_x}\right) \\ & \propto p\left(R^{T_x} | P^{T_x}, Q^{T_x}, \Omega^{T_x}\right) p\left(P^{T_x} | SC^{T_x*}, \Omega^{T_x}\right) p\left(Q^{T_x} | \Omega^{T_x}\right) p\left(Q^{T_x} | \Omega^{T_x}\right) \\ & = \prod_{u_i} \prod_{s_j} \left[N\left(R_{ij}^{T_x} | P_i^{T_x} (Q_j^{T_x})^T, \Omega^{T_x}\right) \right]^{I_{ij}^{T_x}} \times \prod_{u_i} N\left(P_i^{T_x} \mid \sum_{u_v \in N_{u_i}^{T_x}} SC_{iv}^{T_x*} P_v^{T_x}, \Omega^{T_x}\right) \\ & \quad \times \prod_{u_i} N\left(P_i^{T_x} | 0, \Omega^{T_x}\right) \times \prod_{s_j} N\left(Q_j^{T_x} | 0, \Omega^{T_x}\right) \end{aligned} \tag{11}$$

where $I_{ij}^{T_x}$ is the indicator function that equals to 1 if user u_i has rated item s_j and equals to 0 otherwise. The training objective function according to (11) is improved as follows:

$$\Psi \left(R^{T_x}, P^{T_x}, Q^{T_x}, SC_{iv}^{T_x*} \right) = \frac{1}{2} \sum_{(u_i, s_j) \in T_x} \left(R_{ij}^{T_x} - \hat{R}_{ij}^{T_x} \right)^2 + \frac{\lambda}{2} \left(\|P^{T_x}\|_F^2 + \|Q^{T_x}\|_F^2 \right) + \frac{\alpha}{2} \sum_{T_x} \left(\left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SC_{iv}^{T_x*} P_v^{T_x} \right) \times \left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SC_{iv}^{T_x*} P_v^{T_x} \right)^T \right) \quad (12)$$

Where α is tradeoff parameter which controls the factor of user rating similarity by the last term in (12).

4.1.3 Stage 3: adding user and item feature similarity

The survey of sociology shows that certain users have common interests [1]. For example, housewives like watching Korean teleplay. Moreover, some kinds of items become more popular at a certain time of the year. For example, people more likely to buy turkey on Thanksgiving Day, and on June 1 the children’s day, childrens movies are popular. Therefore, in the case of contextual information clustering, we supplement the restricting terms for item and user feature similarity on the basis of (12) which can enhance our models practicability.

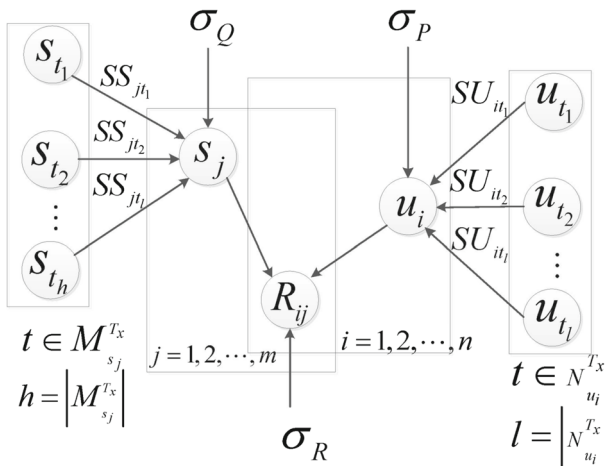


Figure 6 Recommendation Schematic Drawings by adding user and item feature similarity. n denotes the total number of users in x^{th} cluster, m denotes the total number of items in x^{th} cluster, $|N_{u_i}^{T_x}|$ denotes the total number of neighbors of user u_i in x^{th} cluster. $|M_{s_j}^{T_x}|$ denotes the total number of adjacent items of s_j

Similar to (10), we have the conditional distributions given the user and item latent features. Equations (13) and (14) in a graphical form are shown in Figure 6.

$$p\left(P^{T_x} | SU^{T_x}, \Omega^{T_x}\right) = \prod_{u_i} N\left(P_i^{T_x} \mid \sum_{u_v \in N_{u_i}^{T_x}} SU_{i_v}^{T_x*} P_v^{T_x}, \Omega^{T_x}\right) \tag{13}$$

$$p\left(Q^{T_x} | SS^{T_x}, \Omega^{T_x}\right) = \prod_{s_j} N(Q_j^{T_x} \mid \sum_{s_v \in M_{s_j}^{T_x}} SS_{j_v}^{T_x*} Q_v^{T_x}, \Omega^{T_x}) \tag{14}$$

Through a Bayesian inference, the posterior probability of the latent feature vectors P^{T_x} and Q^{T_x} can be obtained as follows based on (11).

$$\begin{aligned} p\left(P^{T_x}, Q^{T_x} | R^{T_x}, SC^{T_x*}, SU^{T_x*}, SS^{T_x*}, \Omega^{T_x}\right) &\propto p\left(R^{T_x} | P^{T_x}, Q^{T_x}, \Omega^{T_x}\right) p\left(P^{T_x} | SC^{T_x*}, \Omega^{T_x}\right) \\ &p\left(P^{T_x} | SU^{T_x*}, \Omega^{T_x}\right) p\left(Q^{T_x} | SS^{T_x*}, \Omega^{T_x}\right) p\left(P^{T_x} | \Omega^{T_x}\right) p\left(Q^{T_x} | \Omega^{T_x}\right) \\ &= \prod_{u_i} \prod_{s_j} \left[N\left(R_{ij}^{T_x} | P_i^{T_x} (Q_j^{T_x})^T, \Omega^{T_x}\right) \right]^{I_{ij}^{T_x}} \times \prod_{u_i} N\left(P_i^{T_x} \mid \sum_{u_v \in N_{u_i}^{T_x}} SC_{i_v}^{T_x*} P_v^{T_x}, \Omega^{T_x}\right) \\ &\times \prod_{u_i} N\left(P_i^{T_x} \mid \sum_{u_v \in N_{u_i}^{T_x}} SU_{i_v}^{T_x*} P_v^{T_x}, \Omega^{T_x}\right) \times \prod_{s_j} N\left(Q_j^{T_x} \mid \sum_{s_v \in M_{s_j}^{T_x}} SS_{j_v}^{T_x*} Q_v^{T_x}, \Omega^{T_x}\right) \\ &\times \prod_{u_i} N\left(P_i^{T_x} | 0, \Omega^{T_x}\right) \times \prod_{s_j} N\left(Q_j^{T_x} | 0, \Omega^{T_x}\right) \end{aligned} \tag{15}$$

The objective function which supplements restrict term of user feature similarity and item feature similarity to the (12) is shown as follows:

$$\begin{aligned} \Psi\left(R^{T_x}, P^{T_x}, Q^{T_x}, SC^{T_x*}, SU^{T_x*}, SS^{T_x*}\right) &= \frac{1}{2} \sum_{(u_i, s_j) \in T_x} \left(R_{ij}^{T_x} - \hat{R}_{ij}^{T_x} \right)^2 + \frac{\lambda}{2} \left(\|P^{T_x}\|_F^2 + \|Q^{T_x}\|_F^2 \right) \\ &+ \frac{\alpha}{2} \sum_{T_x} \left(\left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SC_{i_v}^{T_x*} P_v^{T_x} \right) \left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SC_{i_v}^{T_x*} P_v^{T_x} \right)^T \right) \\ &+ \frac{\beta}{2} \sum_{T_x} \left(\left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SU_{i_v}^{T_x*} P_v^{T_x} \right) \left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SU_{i_v}^{T_x*} P_v^{T_x} \right)^T \right) \\ &+ \frac{\gamma}{2} \sum_{T_x} \left(\left(Q_j^{T_x} - \sum_{s_v \in M_{s_j}^{T_x}} SS_{j_v}^{T_x*} Q_v^{T_x} \right) \left(Q_j^{T_x} - \sum_{s_v \in M_{s_j}^{T_x}} SS_{j_v}^{T_x*} Q_v^{T_x} \right)^T \right) \end{aligned} \tag{16}$$

Where β and γ are tradeoff parameters which control the factor of user and item feature similarity respectively.

4.2 Model training

After the rating data is divided into k subsets based on the clustering result of k -Modes algorithm on the contextual information, the model in (16) is used to train P^{T_x} and Q^{T_x} in each cluster separately. P^{T_x} and Q^{T_x} are considered as variables and gradient decent method can be used to obtain the optimal latent feature vectors. An improved variable step gradient descent algorithm is proposed which is more efficient than PRM [47].

$$\begin{aligned} \frac{\partial \psi}{\partial P_i^{T_x}} = & \sum_{(u_i, s_j) \in T_x} \left(\hat{R}_{ij}^{T_x} - R_{ij}^{T_x} \right) Q_j^{T_x} + \lambda P_i^{T_x} + \alpha \left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SC_{iv}^{T_x*} P_v^{T_x} \right) \\ & - \alpha \sum_{u_v: u_i \in N_{u_v}^{T_x}} SC_{vi}^{T_x*} \left(P_v^{T_x} - \sum_{u_w \in N_{u_v}^{T_x}} SC_{vw}^{T_x*} P_w^{T_x} \right) \\ & + \beta \left(P_i^{T_x} - \sum_{u_v \in N_{u_i}^{T_x}} SU_{iv}^{T_x*} P_v^{T_x} \right) \\ & - \beta \sum_{u_v: u_i \in N_{u_v}^{T_x}} SU_{vi}^{T_x*} \left(P_v^{T_x} - \sum_{u_w \in N_{u_v}^{T_x}} SU_{vw}^{T_x*} P_w^{T_x} \right) \end{aligned} \tag{17}$$

$$\begin{aligned} \frac{\partial \psi}{\partial Q_j^{T_x}} = & \sum_{(u_i, s_j) \in T_x} \left(\hat{R}_{ij}^{T_x} - R_{ij}^{T_x} \right) P_i^{T_x} + \lambda Q_j^{T_x} + \gamma \left(Q_j^{T_x} - \sum_{s_v \in M_{s_j}^{T_x}} SS_{jv}^{T_x*} Q_v^{T_x} \right) \\ & - \gamma \sum_{s_v: s_j \in M_{s_v}^{T_x}} SS_{vj}^{T_x*} \left(Q_v^{T_x} - \sum_{s_w \in M_{s_v}^{T_x}} SS_{vw}^{T_x*} Q_w^{T_x} \right) \end{aligned} \tag{18}$$

Algorithm 1 An improved variable step gradient descent algorithm (IVGDA)

Input: Number of iterations N , initialize $P_i^{T_x}(0), Q_j^{T_x}(0), l_0, t = 0, m, \mu, \varepsilon$.

Output: $P_i^{T_x}(*), Q_j^{T_x}(*)$

Algorithm steps:

- 1: if $t < N$, then goto Step 2, else goto Step 7;
 - 2: Calculate $\frac{\partial \psi}{\partial P_i^{T_x}}(t), \frac{\partial \psi}{\partial Q_j^{T_x}}(t)$;
 - 3: Calculate $p_i^{T_x}(t+1) = p_i^{T_x}(t) - l_t \frac{\partial \psi}{\partial P_i^{T_x}}(t), q_j^{T_x}(t+1) = q_j^{T_x}(t) - l_t \frac{\partial \psi}{\partial Q_j^{T_x}}(t)$;
 - 4: if $\Psi(t+1) < \Psi(t)$, then $Q_j^{T_x}(*), P_i^{T_x}(*), P_i^{T_x}(*) = P_i^{T_x}(t+1)$; else goto Step 5;
 - 5: if $\psi(t) < \varepsilon$, then goto Step 7;
 - 6: Calculate $\alpha_t = 1 - \left[\frac{t-1}{t} \right]^m, l_{t+1} = 4\alpha_t \mu(1 - \mu)$, goto Step 1;
 - 7: Output $P_i^{T_x}(*), Q_j^{T_x}(*)$.
-

Before running the optimization algorithm, we first set the values of P^{T_x} and Q^{T_x} as 0. According to (17) and (18), the user and item latent feature vectors P^{T_x} and Q^{T_x} are updated as follows as in (19) and (20) based on previous values to ensure the fastest improvement of the objective function by each iteration:

$$P_i^{T_x}(t+1) = P_i^{T_x}(t) - l \frac{\partial \psi}{\partial P_i^{T_x}}(t) \quad (19)$$

$$Q_j^{T_x}(t+1) = Q_j^{T_x}(t) - l \frac{\partial \psi}{\partial Q_j^{T_x}}(t) \quad (20)$$

Where l is the learning rate. Our improved variable step gradient descent algorithm is presented in Algorithm 1. In the algorithm, we set a chaos variable μ which has micro different value in each iteration, making it possible for the objective function to converge steadily.

The main computational overhead of the gradient methods lies in the evaluation of the objective function Ψ and its gradients against variables. Because of the sparsity of matrices R^{T_x} , SC^{T_x} , SU^{T_x} and SS^{T_x} , the computational complexity of evaluating the object function Ψ is $O(l_n \omega_{R^{T_x}} + l_n \omega_{SC^{T_x}} + l_n \omega_{SU^{T_x}} + l_n \omega_{SS^{T_x}})$, where $\omega_{R^{T_x}}$, $\omega_{SC^{T_x}}$, $\omega_{SU^{T_x}}$ and $\omega_{SS^{T_x}}$ are the numbers of nonzero entries in matrices R^{T_x} , SC^{T_x} , SU^{T_x} and SS^{T_x} , respectively, and l_n is the number of latent factors. The computational complexity for gradients $\frac{\partial \psi}{\partial P_i^{T_x}}$, $\frac{\partial \psi}{\partial Q_i^{T_x}}$ in (17), (18) are $O(l_n \omega_{R^{T_x}} + l_n \omega_{SC^{T_x}} + l_n \omega_{SU^{T_x}})$, $O(l_n \omega_{R^{T_x}} + l_n \omega_{SS^{T_x}})$, respectively. Thus, for each iteration, the total computational complexity is $O(l_n \omega_{R^{T_x}} + l_n \omega_{SC^{T_x}} + l_n \omega_{SU^{T_x}} + l_n \omega_{SS^{T_x}})$, which indicates that the computational time of our method is linear with respect to the number of observations in the four sparse matrices. In addition, we divide the training data into groups by clustering, which further helps improve the efficiency of our method. This complexity analysis shows that our proposed approach is very efficient and can well scale to very large datasets.

5 Experimental evaluation

In this section, we will report the extensive experiments that we have conducted to evaluate the performance of our proposed recommender system. We will start with a discussion on the experimental setup of our evaluation, which will cover the information about datasets, performance measurements, the recommendation algorithms to be used for comparative study and the parameter setting. The detailed experimental results will be reported in the second half of this section.

5.1 Experimental setup

5.1.1 Datasets

Three commonly used real-life datasets are used to validate our model and perform comparative study with existing models in our work. These datasets are MovieLens,¹ Epinions² and Douban³ which are chosen due to the use of item feature, user feature and contextual

¹<http://www.grouplens.org/node/12>

²<http://www.epinions.com>

³<http://www.douban.com>

Table 2 MovieLens dataset

Category	User count	Item count	Rating count	Sparsity	r
Movies	6040	3900	1000209	4.25E-2	3.571

information in our model. The Moivelens dataset includes 3,900 moives, 6,040 users and 1,000,209 ratings. Epinions is a consumer opinion website where users can review items and also assign them numeric ratings in the range of 1 to 5. We use the version of the Epinions dataset published by the authors of [55]. This dataset is made up of ratings from 75,888 users who rated a total of 149,943 different items. The total number of ratings is 598,329, it includes 25 categories and 240 subcategories. In our work, we selected 5 representative categories in the Epinions dataset and experimented based on them. Douban is one of the most popular social networks in China. It includes several parts: reading, movie, music, city circle and so on. In Douban, users can rate the book which they have read and share the reviews to their friends. We crawled nearly 30,000 users and 400,000 items from July 2012 to January 2014. The details of these three datasets are shown in Tables 2, 3 and 4.

5.1.2 Performance Measures

In our work, we use 70% of the data as the training dataset and the remaining 30% as the test dataset in the above three datasets respectively. The performance measures we use in our experiments are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) which are the most popular accuracy measures. RMSE and MAE are defined as follows respectively.

$$RMSE = \sqrt{\frac{\sum_{(u_i,s_j) \in T_{test}} (R_{ij} - \hat{R}_{ij})^2}{|T_{test}|}} \tag{21}$$

$$MAE = \frac{\sum_{(u_i,s_j) \in T_{test}} |R_{ij} - \hat{R}_{ij}|}{|T_{test}|} \tag{22}$$

Where T_{test} is the training dataset, R_{ij} is the real rating of item from user, and \hat{R}_{ij} is the corresponding predicted rating.

5.1.3 Competitive algorithms for comparison

The following recommendation models are involved in our experimental evaluation for performance comparative study.

Table 3 Epinions dataset

Category	User count	Item count	Rating count	Sparsity	r
Books	14176	226022	40231	1.256E-5	4.27
Music	10895	15313	45356	2.719E-4	4.28
Toys	6203	3547	27119	1.233E-3	4.13
Software	8196	1434	18989	1.612E-3	4.01
Cars	11202	3011	18808	5.576E-4	4.15

Table 4 Douban dataset

Category	User count	Item count	Rating count	Sparsity	r
Books	7204	189923	631724	4.62E-4	4.25
Music	19782	150436	173269	5.82E-5	4.23
Movies	4780	47906	1595667	6.97E-3	3.766

- (1) BaseMF. This model uses basic matrix factorization techniques without considering any social factors.
- (2) SocialMF. This mode improves the recommendation accuracy compared with the BaseMF as it makes a use of the factor of social trust as the social links in the dataset.
- (3) PRM. This model integrates the user interest, interpersonal interest similarity and interpersonal influence into the model. According to the properties of items, the dataset was divided into subcategories to improve the recommendation accuracy of the model.
- (4) HMF. Our model considers four types relation which are user-item relation, item-item relation, user-user relation and contextual information-user relation based on the topological structure of hypergraph. The model classifies the training dataset into several groups based on the contextual information which helps enhance the intrinsic link among features in the latent space, and three important factors (user feature, item feature and users ratings) are incorporated into the model.

5.1.4 Parameter settings

Table 5 presents the details about the parameters used in all methods including their meanings and the default values.

5.2 Experiments results

5.2.1 Recommendation accuracy

In this section, we first evaluate the recommendation accuracy of different models in terms of both RMSE and MAE. Then we compare their classification accuracy based on precision,

Table 5 Description and default value of parameters

Symbol	Description	Default value
l_n	The dimension of latent feature space	$l_n = 10$
l_u	The dimension of user feature space	$l_u = 5$
l_s	The dimension of item feature space	$l_s = 10$
l_c	The dimension of contextual information space	$l_c = 5$
k	The number of clusters which were divided by k -Modes algorithm	$k = 8$
λ	The regularization constant	$\lambda = 0.1$
α	The tradeoff parameters play the role of adjusting the strengths of user rating similarity in object function (16)	$\alpha = 1$
β	The tradeoff parameters play the role of adjusting the strengths of user feature similarity in object function (16)	$\beta = 1$
γ	The tradeoff parameters play the role of adjusting the strengths of item feature similarity in object function (16)	$\gamma = 1$

Table 6 Performance comparison on the movielens dataset

Dataset	BaseMF		SocialMF		PRM		HMF	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Movies	1.437	1.120	1.158	0.953	1.125	0.944	1.089	0.932
	22.8%	16.8%	6.0%	7.1%	3.2%	1.3%		

recall and F-measure. We conduct a series of experiments on MovieLens, Epinions and Douban under the default experimental settings set forth in Table 6. The results of experiments are shown in Tables 6, 7 and 8. The percentage numbers in each cell of the tables are the relative improvements of the HMF model over the other comparative models. We can see from the results that the accuracy of our model is better than PRM and outperforms BaseMF and SocialMF by a larger margin. More specifically, as shown in Table 6, our model increases recommendation accuracy by 17.2%, 6.0% and 3.2% in terms of RMSE, and by 16.8%, 7.1% and 1.3% in terms of MAE over BaseMF, SocialMF and PRM, respectively. Our model achieves an even better accuracy performance when dealing with Epinions and Douban datasets, as shown in Tables 7 and 8. This is because that the contextual information was taken into account in our model and the training dataset was divided into 8 groups using k -Modes clustering algorithm. The MovieLens dataset doesn't have the contextual information, so the k -Modes clustering algorithm is not applicable on this dataset.

In order to further test the overall performance of the HMF model, we utilize three additional measures including precision, recall, and their harmonic mean F1 for experimental evaluation. For the conduct of this experiment, we must transform the item ratings into a binary scale (i.e., relevant and non-relevant) by converting each rating of four or five to relevant and all ratings from one to three to not-relevant. Thus, the precision is the ratio of recommended items (i.e., the items whose predicted ratings are larger than three) that are relevant. The recall is the fraction of relevant items that are recommended. Finally, the F1 score is the harmonic mean of the precision and recall. We use MovieLens dataset and set numbers of recommendation range from 2 to 30. From Figure 7a and b, we can see that our model is superior to other models in terms of precision, recall and F1.

Table 7 Performance comparison on the epinions dataset

Dataset	BaseMF		SocialMF		PRM		HMF	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Books	1.437	1.162	1.203	1.040	1.136	0.962	1.096	0.943
	23.7%	18.8%	8.9%	9.3%	3.5%	2.0%		
Music	1.392	1.145	1.185	0.997	1.114	0.941	1.091	0.925
	21.6%	19.2%	7.9%	7.2%	2.1%	1.7%		
Toys	1.377	1.133	1.175	1.009	1.088	0.950	1.007	0.891
	26.9%	21.4%	14.3%	11.7%	7.4%	6.2%		
Software	1.421	1.190	1.193	1.064	1.125	0.980	1.032	0.886
	27.4%	25.5%	13.5%	16.7%	8.3%	9.6%		
Cars	1.479	1.267	1.212	1.112	1.134	1.007	1.089	0.979
	26.4%	22.7%	10.1%	12.0%	4.0%	2.8%		

Table 8 Performance comparison on the douban dataset

Dataset	BaseMF		SocialMF		PRM		HMF	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Books	1.494 27.7%	1.204 23.3%	1.192 9.4%	1.083 14.7%	1.134 4.8%	0.977 5.4%	1.080	0.924
Music	1.452 28.7%	1.193 23.0%	1.191 13.1%	1.015 9.5%	1.170 11.5%	1.005 8.6%	1.035	0.919
Movies	1.470 26.1%	1.213 25.4%	1.186 8.3%	1.019 11.2%	1.116 2.6%	0.933 3.0%	1.087	0.905

5.2.2 Performance on sparse datasets

In order to analyze the performance of various models when dealing with sparse datasets, we take out some rating data from the datasets randomly to generate new datasets with desired sparsity for the experiments, and evaluate the recommendation accuracy of each model against RMSE and MAE in different scenarios. To ensure the fairness of the experiment, we only conduct this experiment using MovieLens with default parameter settings for different models. Figure 8a and b present the experiment results of different models in terms of MAE and RMSE under 7 different cases of data sparsity, which are 1E-2, 5E-3, 1E-3, 5E-4, 1E-4, 5E-5, 1E-5. For instance, 1E-2 denotes 1×10^{-2} . X coordinates represents sparsity which is defined as: $Sparsity = N_r / (N_u \times N_s)$, where N_r denotes the number of ratings, N_u and N_s denote the number of users and items, respectively. We can see that our model is superior to other models compared from 1E-2 to 1E-5 in terms of both MAE and RMSE. Because HMF model uses both user and item feature whilst the competitive models do not use, it demonstrates the user feature and item feature are effectively integrated into our model.

5.2.3 Sensitivity study

In this subsection, we conducted experiments to test the sensitivity of our model with regard to various parameters to understand the effect posed by them on the recommendation accuracy of our model.

(a) The effect of the number of iterations

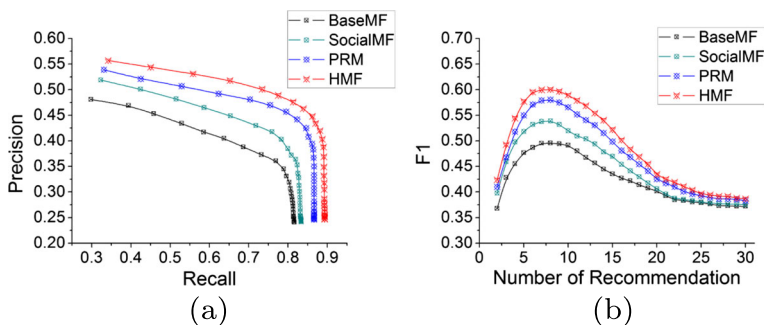


Figure 7 Results on dataset (MovieLens) in terms of precision/recall and F1

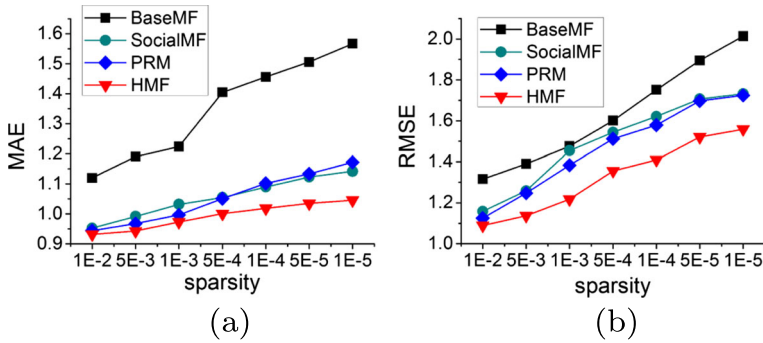


Figure 8 Results on sparse dataset (MovieLens) in terms of MAE and RMSE

In our recommendation model, we use the improved variable step gradient descent algorithm (IVGDA) to do experiment to get the optimal user and item latent feature vectors. Because PRM use the MovieLens dataset, in order to compare with the other algorithm fairly, in this experiment, the MovieLens is used and we set $m = 2$, $\mu = 0.5$ and $\varepsilon = 0.001$ in Algorithm 1. We compare the performance of IVGDA with the gradient descend algorithm (GDA) used by PRM Model [47]. From Figure 9a and b, we can observe that both MAE and RMSE decrease steadily when the number of iterations increases for the two algorithms, but IVGDA features a noticeably better convergence than GDA.

(b) The effect of cluster number (k)

In order to evaluate the effect of number of clusters on recommendation accuracy, we conduct experiments on Epinions and Douban datasets, because these two datasets have the contextual information, but MovieLens does not. We set the value of k from 1 to 32, and set other parameters as their respective default value. As shown in Figure 10a and b for Epinions dataset, we see that subcategories such as software and toys have a lower predicted error when k is set to relatively small values, and the value of MAE and RMSE grow quickly when k is bigger than 8. In Figure 11a and b for Douban dataset, we can see that the predicted error reaches the minimum when the value of k gets closer to 16. In fact, the rating count in Douban is 10 times larger than that in Epinions, so we can conclude that k should be generally set to a larger value when the rating count in the dataset is higher. However, the prediction accuracy will decrease

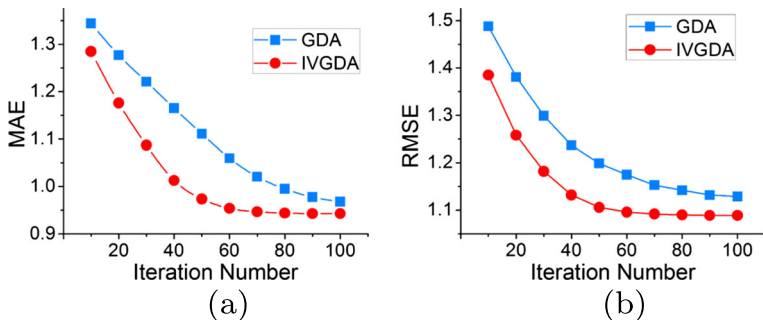


Figure 9 The effect of the number of iterations (MovieLens)

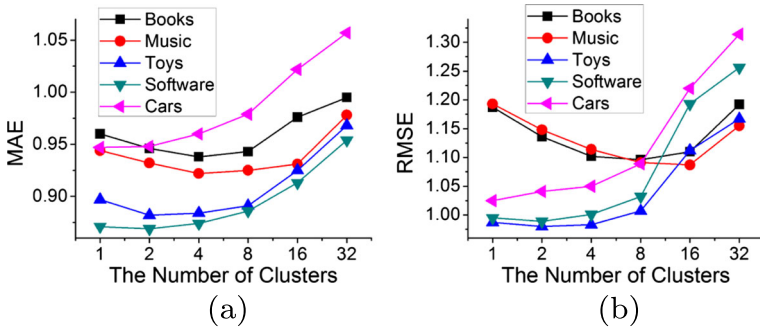


Figure 10 The number of clusters experiments with MAE and RMSE on Epinions

when k is greater than a threshold. Experimentally, when k value is 8 or 16, the HMF has a better recommendation accuracy in our experimental datasets.

(c) Effect of weight coefficients in our model

In this section, we conducted a series experiments on Douban dataset to test the impact of different weight coefficients used in our model on its prediction error. Specifically, we evaluate the effect of parameters λ , α , β and γ . Parameters α , β , γ control the influence of user rating similarity, user feature similarity and item feature similarity on the prediction error respectively, and λ , being the regularization constant, plays a role in adjusting the strengths of different terms in the objective function (16). Larger values of the weight coefficients in the objective function in (16) indicate a stronger impact of the respective factor on the user and item latent feature. In order to analyze each parameter individually, we every time only change the value of the target parameter while keeping the values of others fixed.

In Figure 12, we set λ from 0.01 to 20, and other parameters were set as default. As shown in Figure 12a and b, our model achieves its better performance on Douban when λ ranges from 0.05 to 1, and the prediction error will increase when λ is set to a too small or too large value.

As shown in Figure 13a–f, the accuracy of our model reaches its highest level on Douban when $\alpha = 5$, β ranges from 0.5 to 2 and γ falls in the range of (1,5), which provides a good guidance to set their values in practice.

To ensure the fairness of our experiments, we set $\lambda = 0.1$, the same configuration as in the competitive model. α , β and γ are set to 1 as an empirical value according to previous

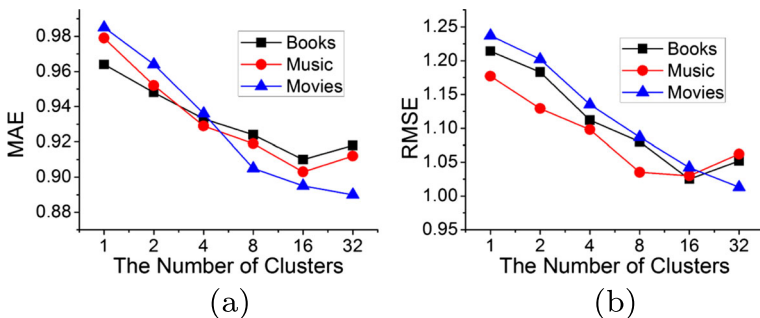


Figure 11 The number of clusters experiments with MAE and RMSE on Douban

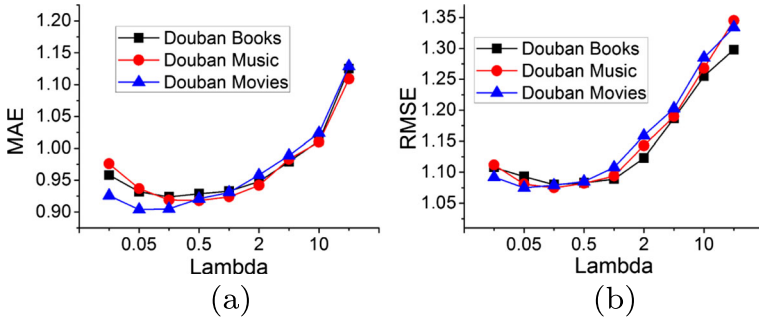


Figure 12 The value of λ influence on prediction error which based on Douban dataset

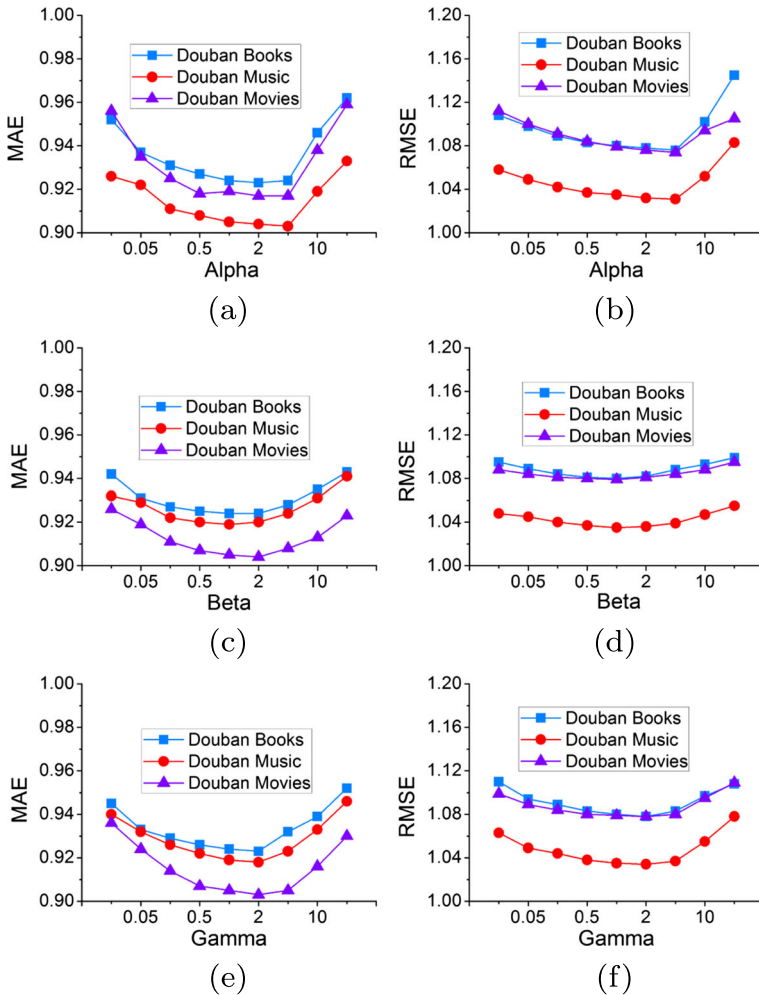


Figure 13 The value of tradeoff parameters influence on prediction error which based on Douban dataset

Table 9 Parameters combination performance on Douban dataset

Parameters combination	RMSE	MAE	Parameters combination	RMSE	MAE
$\lambda = 0.05, \alpha = 1, \beta = 1, \gamma = 1$	1.091	0.906	$\lambda = 0.5, \alpha = 1, \beta = 1, \gamma = 1$	1.097	0.921
$\lambda = 0.05, \alpha = 3, \beta = 1, \gamma = 2$	1.085	0.901	$\lambda = 0.5, \alpha = 3, \beta = 1, \gamma = 2$	1.092	0.910
$\lambda = 0.05, \alpha = 5, \beta = 2, \gamma = 2$	1.081	0.899	$\lambda = 0.5, \alpha = 5, \beta = 2, \gamma = 2$	1.085	0.903
$\lambda = 0.05, \alpha = 5, \beta = 2, \gamma = 5$	1.084	0.901	$\lambda = 0.5, \alpha = 5, \beta = 2, \gamma = 5$	1.091	0.911
$\lambda = 0.1, \alpha = 1, \beta = 1, \gamma = 1$	1.087	0.905	$\lambda = 1, \alpha = 1, \beta = 1, \gamma = 1$	1.108	0.932
$\lambda = 0.1, \alpha = 3, \beta = 1, \gamma = 2$	1.078	0.895	$\lambda = 1, \alpha = 3, \beta = 1, \gamma = 2$	1.107	0.931
$\lambda = 0.1, \alpha = 5, \beta = 2, \gamma = 2$	1.062	0.893	$\lambda = 1, \alpha = 5, \beta = 2, \gamma = 2$	1.101	0.925
$\lambda = 0.1, \alpha = 5, \beta = 2, \gamma = 5$	1.077	0.894	$\lambda = 1, \alpha = 5, \beta = 2, \gamma = 5$	1.110	0.933

works. In order to obtain a more optimized combination of parameters, we choose the appropriate value for parameters combining from a single parameter optimal interval. Then, the experimental results of 16 representative combinations are presented in Table 9.

Besides evaluating the effect of each weight coefficient individually on our recommendation model, we also explore different binary combinations of them to see how different combinations of factors (user rating similarity, user feature similarity and item feature similarity) affect the performance of our model. As shown in Figure 14, NULL denotes the model where α, β and γ values are zero (i.e., the model itself is reduced to BaseMF). UF denotes the model using only the user feature similarity ($\alpha = 0, \beta = 1, \gamma = 0$), IF denotes

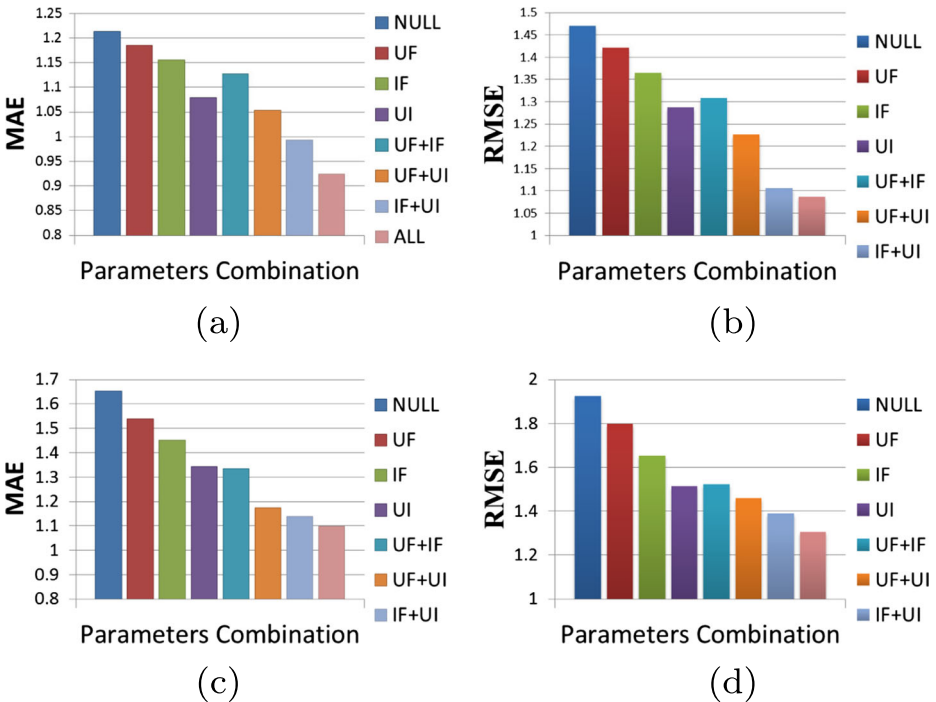


Figure 14 Different combinations of tradeoff parameters influence on MAE and RMSE which test on Douban Movies Dataset. **a** and **b** are done under normal circumstance, **c** and **d** are done under cold start circumstances

Table 10 Cold start performance comparison on Douban dataset

Dataset	BaseMF		SocialMF		PRM		HMF	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Books	2.015 34.1%	1.67 33.9%	1.521 12.7%	1.206 8.5%	1.413 6.0%	1.131 2.4%	1.328	1.104
Music	2.128 38.1%	1.730 36.4%	1.505 12.5%	1.214 9.4%	1.412 6.7%	1.155 4.8%	1.317	1.100
Movies	1.924 32.3%	1.652 33.5%	1.498 13.0%	1.197 8.3%	1.398 6.8%	1.150 4.5%	1.303	1.098

the model using only the item feature similarity ($\alpha = 0, \beta = 0, \gamma = 1$) and UI denotes the model using only the user rating similarity ($\alpha = 1, \beta = 0, \gamma = 0$). UF+If, UF+UI, If+UI correspond to the cases of using the combination of two appropriate factors, and finally ALL is a case where all the three factors are considered. Furthermore, two experiments were conducted under the normal and cold star circumstances. The results are shown in Figure 14. The settings under the cold start scenario are the same as those in Section 5.2.4 where the number of target user's ratings is no more than 5. When each factor is used in the model individually, the symbol UI shows the highest prediction accuracy. The results suggest that the similarity of user ratings is the most important factor compared to others. From those experiments, we also obtained an interesting finding that the gap in the prediction accuracy using various factors become larger when dataset becomes increasingly sparse. Figure 14 clearly demonstrates the benefits of using the three different factors as a combination, which can apparently help improve the recommendation accuracy of our model in the two scenarios.

5.2.4 Performance on cold start problem

Some users in social network provide a lot of ratings, but most users only provide very limited number of ratings [24, 38]. In this paper, we define the users who have provided no more than 5 ratings as cold start users [16]. We conduct an experiment on the cold start users by using Douban dataset, and the comparative results for different models are shown in Table 10. Our model, by considering both user feature and item feature, is able to decrease the recommendation error by more than 30%, 10%, 6% on RMSE and by more than 30%, 8%, 2% on MAE against BaseMF, SocialMF and PRM.

6 Conclusion and future work

With the advent of online social networks, exploiting the information hidden in the social networks to predict the behavior of users has become very valuable. Recommender systems have become very useful tools to mine the knowledge from the social networks to improve recommendation accuracy. In this paper, we presented a hybrid recommender system based on hypergraph topologic structure in social networks, and four types of relations, which are user-item relation, item-item relation, user-user relation and contextual information-user relation, have been taken into account in our recommendation model to improve its accuracy. We also conducted extensive experiments on three larger real-life datasets, and the experimental results show that our model enjoys higher recommendation accuracy over

the existing major recommendation models. Although it has some advantages in its recommendation effectiveness, our model still has some limitations and there is room for further improvement in some aspects. Firstly, our HMF model requires access to more information in social networks such as user feature, item feature and contextual information. Secondly, our HMF model faces an increased computational complexity when user feature and item similarity are calculated.

There will be several interesting directions to explore for our future work. We want to extend the model to handle recommendation across multiple social networks as it is not uncommon that many users register in several different social networks. Just as presented in [45], shilling attack is one of great threat to recommender system, attack detection is also an important direction for our next work. Furthermore, since the information of user feature and context will be used and exploited, the issues of privacy preservation and protection in recommendation should be considered and further investigated.

Acknowledgments This work was supported by the Natural Science Foundation of China (No. 61772034, No. 61672039, No. 61602009, No. 61370050), University Natural Science Research Project of Anhui Province (No. KJ2015A067), Natural Science Foundation of Anhui Province (No. 1608085MF145), and Wuhu Science and Technology Plan Projects (No. 2015cxy10).

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the Web. *Soc. Netw.* **25**(3), 211–230 (2003)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
3. Adomavicius, G., Sankaranarayanan, R., Sen, S., et al.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst. (TOIS)* **23**(1), 103–145 (2005)
4. Akther, A., Alam, K.M., Heung-Nam, K., et al.: Social network and user context assisted personalization for recommender systems. In: 2012 International Conference on Innovations in Information Technology (IIT), pp. 95–100 (2012)
5. Chu, W.T., Tsai, Y.L.: A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web* **20**(6), 1313–1331 (2017)
6. Bao, Y., Fang, H., Zhang, J.: Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 2–8 (2014)
7. Bennett, P.N. et al.: Overview of the special issue on contextual search and recommendation. *ACM Trans. Inf. Syst. (TOIS)* **33**(1), 1–7 (2015)
8. Bobadilla, J., Ortega, F., Hernando, A., et al.: Recommender systems survey. *Knowledge-Based Systems* **46**, 109–132 (2013)
9. Braunhofer, M.: Hybridisation techniques for cold-starting context-aware recommender systems. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 405–408, Foster City (2014)
10. Cao, J., Wu, Z., Wang, Y., et al.: Hybrid collaborative filtering algorithm for bidirectional Web service recommendation. *Knowl. Inf. Syst.* **36**(3), 607–627 (2013)
11. Chen, L., Wang, F.: Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowl.-Based Syst.* **50**, 44–59 (2013)
12. Chen, C.C., Wan, Y.-H., Chung, M.-C., et al.: An effective recommendation method for cold start new users using trust and distrust networks. *Inf. Sci.* **224**, 19–36 (2013)
13. Zhang, Q., Wu, J., Zhang, Q., et al.: Dual influence embedded social recommendation. *World Wide Web*. <https://doi.org/10.1007/s11280-017-0486-5> (2017)
14. Forsati, R. et al.: Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Trans. Inf. Syst. (TOIS)* **32**(4), 17 (2014)
15. Gao, J., Zhao, Q., Ren, W., et al.: Dynamic Shortest Path Algorithms for Hypergraphs. *IEEE/ACM Trans. Networking* **PP**(99), 1–1 (2014)

16. Jamali, M., Ester, M.: TrustWalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 397–406, Paris (2009)
17. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 135–142 (2010)
18. Jiang, M., Cui, P., Liu, R., et al.: Social contextual recommendation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 45–54. Maui (2012)
19. Jiang, S., Qian, X., Mei, T., et al.: Personalized travel sequence recommendation on multi-source big social media. *IEEE Trans. Big Data* **2**(1), 43–56 (2016)
20. Jiang, S., Qian, X., Shen, J., et al.: Author topic model-based collaborative filtering for personalized POI recommendations. *IEEE Trans. Multimed.* **17**(6), 907–918 (2015)
21. Kazienko, P., Musial, K., Kajdanowicz, T.: Multidimensional Social Network in the Social Recommender System. *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans* **41**(4), 746–759 (2011)
22. Kim, Y.S., Yum, B.-J., Song, J., et al.: Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites. *Expert Systems with Applications* **28**(2), 381–393 (2005)
23. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
24. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
25. Lei, X., Qian, X., Zhao, G.: Rating prediction based on social sentiment from textual reviews. *IEEE Trans. Multimed.* **18**(9), 1910–1921 (2016)
26. Liu, X., Aberer, K.: SoCo: a social network aided context-aware recommender system. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 781–802, Rio de Janeiro (2013)
27. Liu, N.N., Zhao, M., Yang, Q.: Probabilistic latent preference analysis for collaborative filtering. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 759–766, Hong Kong (2009)
28. Lu, X., Wu, H., Yuan, Y.: Double Constrained NMF for Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.* **52**(5), 2746–2758 (2014)
29. Luo, X., Xia, Y., Zhu, Q.: Applying the learning rate adaptation to the matrix factorization based collaborative filtering. *Knowl.-Based Syst.* **37**, 154–164 (2013)
30. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 203–210, Boston (2009)
31. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: ACM Conference on Recommender Systems, pp. 165–172 (2013)
32. Mirbakhsh, N., Ling, C.X.: Improving Top-N recommendation for cold-start users via cross-domain information. *ACM Trans. Knowl. Discov. from Data (TKDD)* **9**(4), 33 (2015)
33. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2007)
34. Nathanson, T., Bitton, E., Goldberg, K.: Eigentaste 5.0: constant-time adaptability in a recommender system using item clustering. In: Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 149–152. Minneapolis (2007)
35. Nguyen, T.T.: Improving recommender systems: user roles and lifecycles. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 417–420, USA (2014)
36. Pazzani, M., Billsus, D.: Learning and revising user profiles: The identification of interesting Web sites. *Mach. Learn.* **27**(3), 313–331 (1997)
37. Pliakos, K., Kotropoulos, C.: Personalized and geo-referenced image recommendation using unified hypergraph learning and group sparsity optimization. In: 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), pp. 306–309 (2014)
38. Rashid, A.M., Albert, I., Cosley, D., et al.: Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the 7th International Conference on Intelligent user Interfaces, pp. 127–134. San Francisco (2002)
39. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61–70. Edmonton (2002)
40. Salton, G.: Automatic text processing. Addison Wesley, Boston (1989)

41. Sarwat, M., Levandoski, J.J., Eldawy, A., et al.: LARS*: An Efficient and Scalable Location-Aware Recommender System. *IEEE Trans. Knowl. Data Eng.* **26**(6), 1384–1399 (2014)
42. Shi, Y., Larson, M., Hanjalic, A.: Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Trans. Intell. Syst. Technol.* **4**(1), 1–19 (2013)
43. Shi, Y., Larson, M., Hanjalic, A.: Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation. *Inf. Sci.* **229**, 29–39 (2013)
44. Theodoridis, A., Kotropoulos, C., Panagakis, Y.: Music recommendation using hypergraphs and group sparsity. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 56–60 (2013)
45. Wu, Z., Wu, J., Cao, J., et al.: HySAD: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–993 (2012)
46. Xiwang, Y., Yang, G., Yong, L.: Bayesian-Inference-Based Recommendation in Online Social Networks. *IEEE Trans. Parallel Distrib. Syst.* **24**(4), 642–651 (2013)
47. Xueming, Q., He, F., Guoshuai, Z., et al.: Personalized recommendation combining user interest and social circle. *IEEE Trans Knowl Data Eng* **26**(7), 1763–1777 (2014)
48. Yang, X., Steck, H., Liu, Y.: Circle-based recommendation in online social networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1267–1275, Beijing (2012)
49. Yifan, H., Koren, Y., Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. In: 2008. ICDM '08. Eighth IEEE International Conference on Data Mining, pp. 263–272 (2008)
50. Yoshii, K., Goto, M., Komatani, K., et al.: An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Trans. Audio, Speech, Lang. Process.* **16**(2), 435–447 (2008)
51. Yu, K., Schwaighofer, A., Tresp, V., et al.: Probabilistic memory-based collaborative filtering. *IEEE Trans Knowl Data Eng* **16**(1), 56–69 (2004)
52. Yuan, W., Shu, L., Chao H. C., et al.: ITARS: trust-aware recom-mender system using implicit trust networks. *Commun. IET* **4**(14), 1709–1721 (2010)
53. Zhao, G., Qian, X., Kang, C.: Service Rating Prediction by Exploring Social Mobile Users' Geographic Locations. *IEEE Transactions on Big Data* **3**(1), 67–78 (2017)
54. Zhao, G., Qian, X., Xie, X.: User-service rating prediction by exploring social users' rating behaviors. *IEEE Trans. Multimed.* **18**(3), 496–506 (2016)
55. Zhong, E., Fan, W., Wang, J., et al.: ComSoc: adaptive transfer of user behaviors over composite social network. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 696–704. Beijing (2012)