Recommender Systems with Social Regularization

Hao Ma
The Chinese University of
Hong Kong
Shatin, N.T., Hong Kong
hma@cse.cuhk.edu.hk

Dengyong Zhou
Microsoft Research
One Microsoft Way
Redmond, WA 98052
denzho@microsoft.com

Chao Liu
Microsoft Research
One Microsoft Way
Redmond, WA 98052
chaoliu@microsoft.com

Michael R. Lyu
The Chinese University of
Hong Kong
Shatin, N.T., Hong Kong
lyu@cse.cuhk.edu.hk

Irwin King*
AT&T Labs – Research
180 Park Ave
Forlham Park, NJ 07932
irwin@research.att.com

ABSTRACT

Although *Recommender Systems* have been comprehensively analyzed in the past decade, the study of social-based recommender systems just started. In this paper, aiming at providing a general method for improving recommender systems by incorporating social network information, we propose a matrix factorization framework with social regularization. The contributions of this paper are four-fold: (1) We elaborate how social network information can benefit recommender systems; (2) We interpret the differences between social-based recommender systems and trust-aware recommender systems; (3) We coin the term *Social Regularization* to represent the social constraints on recommender systems, and we systematically illustrate how to design a matrix factorization objective function with social regularization; and (4) The proposed method is quite general, which can be easily extended to incorporate other contextual information, like social tags, etc. The empirical analysis on two large datasets demonstrates that our approaches outperform other state-of-the-art methods.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Information Filtering; J.4 [Computer Applications]: Social and Behavioral Sciences

General Terms: Algorithm, Experimentation

Keywords: Recommender Systems, Collaborative Filtering, Social Network, Matrix Factorization, Social Regularization

---

*Irwin King is currently on leave from the Chinese University of Hong Kong, where the major part of this research was performed.

1. INTRODUCTION

As an indispensable type of *Information Filtering* technique, recommender systems have attracted a lot of attention in the past decade. Related recommendation techniques have been widely studied in research communities of information retrieval [5, 8, 12, 21, 36, 38, 39, 40], machine learning [28, 30, 31, 33, 41] and data mining [1, 3, 11, 14, 15]. Due to their great commercial value, recommender systems have also been successfully deployed in industry, such as product recommendation at Amazon, music recommendation at iTunes, movie recommendation at Netflix, etc.

In the past few years, the dramatic expanding of Web 2.0 Web sites and applications poses new challenges for traditional recommender systems. Traditional recommender systems always ignore social relationships among users. But in our real life, when we are asking our friends for recommendations of nice digital cameras or touching movies, we are actually requesting verbal social recommendations. Social recommendation is a daily occurrence, and we always turn to our friends for recommendations. Hence, in order to improve recommender systems and to provide more personalized recommendation results, we need to incorporate social network information among users.

Recently, based on the intuition that users' trust relations can be employed to enhance traditional recommender systems, a few trust-aware recommendation methods have been proposed [2, 22, 23, 25, 27]. These methods utilize the inferred implicit or observed explicit trust information to further improve traditional recommender systems. Trust-aware recommender systems move an important step forward in the research of recommender systems. However, to achieve the goal of "social recommendation", these approaches still have several inherent limitations and weaknesses that need to be addressed.

First of all, "trust relationships" are different from "social friendships" in many aspects. Typically, on the Web, when a user u_i likes a review issued by another user u_t, user u_i probably will add user u_t to his/her trust list. This process of trust generation is a unilateral action that does not require user u_t to confirm the relationship. This also indicates that user u_i does not need to even know user u_t in the real life. "Social friendships" refer to the cooperative and mutual relationships that surround us, such as classmates, colleagues, or relatives, etc. Lots of social networking Web sites, like

Facebook and Orkut, are designed for online users to interact and connect with their friends in the real life. From the definition, we can see that trust-aware recommender systems cannot represent the concept of "social recommendation", since the idea of "social recommendation" anticipates to improve recommender systems by incorporating a social friend network.

Secondly, trust-aware recommender systems are based on the assumption that users have similar tastes with other users they trust. This hypothesis may not always be true in social recommender systems since the tastes of one user's friends may vary significantly. Some friends may share similar favors with this user while other friends may have totally different tastes. Hence, trust-aware recommendation algorithms cannot be directly applied to generate recommendations in social recommender systems.

Thirdly, due to the rapid growth of Web 2.0 applications, online users spend more and more time on social network related applications since interacting with real friends is the most attractive activity on the Web. On the contrary, only few online systems, like Epinions, have implementations of trust mechanism. Thus, in order to provide more proactive and personalized recommendation results to online users, we should pay more attention to the research of social recommendation, in addition to the existing research of trust-aware recommendation.

In this paper, aiming at solving the problems mentioned above, we propose two social recommendation methods that utilize social information to improve the prediction accuracy of traditional recommender systems. More specifically, the social network information is employed in designing two social regularization terms to constrain the matrix factorization objective function. Moreover, friends with dissimilar tastes are treated differently in the social regularization terms in order to represent the taste diversity of each user's friends. Our proposed approaches are quite general, and they can also be applied to trust-aware recommender systems. The experimental analysis on two large datasets (one dataset contains a social friend network while the other dataset contains a social trust network) shows that our methods outperform other state-of-the-art algorithms.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of several major approaches for recommender systems and some related work. Section 3 describes the problem we study in this paper. Section 4 details the concept of low-rank matrix factorization. The social regularization framework is presented in Section 5. The results of an empirical analysis are presented in Section 6, followed by the conclusion and future work in Section 7.

2. RELATED WORK

In this section, we review several major approaches to recommender systems, including (1) traditional recommender systems which are mainly based on collaborative filtering techniques, (2) trust-aware recommender systems which have drawn lots of attention recently, and (3) social recommender systems which we study in this paper.

2.1 Traditional Recommender Systems

As mentioned in [10], one of the most commonly-used and successfully-deployed recommendation approaches is collaborative filtering. In the field of collaborative filtering, two types of methods are widely studied: neighborhood-based approaches and model-based approaches. Neighborhood-based methods mainly focus on finding the similar users [4, 12] or items [6, 17, 32] for recommendations. User-based approaches predict the ratings of active users based on the ratings of similar users found, while item-based approaches predict the ratings of active users based on the computed information of items similar to those chosen by the active user. User-based and item-based approaches often use Pearson Correlation Coefficient (PCC) algorithm [29] and Vector Space Similarity (VSS) algorithm [4] as the similarity computation methods. PCC method can generally achieve higher performance than VSS approach, since the former considers the differences of user rating style.

In contrast to the neighborhood-based approaches, the model-based approaches to collaborative filtering use the observed user-item ratings to train a compact model that explains the given data, so that ratings could be predicted via the model instead of directly manipulating the original rating database as the neighborhood-based approaches do [19]. Algorithms in this category include the clustering model [13], the aspect models [8, 9, 33], the latent factor model [5], the Bayesian hierarchical model [40] and the ranking model [19]. [13] presented an algorithm for collaborative filtering based on hierarchical clustering, which tried to balance both robustness and accuracy of predictions, especially when few data were available. [8] proposed an algorithm based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables.

Recently, due to the efficiency in dealing with large datasets, several low-dimensional matrix approximation methods [28, 30, 31, 34] have been proposed for collaborative filtering. These methods all focus on fitting the user-item rating matrix using low-rank approximations, and employ the matrix to make further predictions. The Low-rank matrix factorization methods are very efficient in training since they assume that in the user-item rating matrix, only a small number of factors influences preferences, and that a user's preference vector is determined by how each factor applies to that user. Low-rank matrix approximations based on minimizing the sum-squared errors can be easily solved using Singular Value Decomposition (SVD), and a simple and efficient Expectation Maximization (EM) algorithm for solving weighted low-rank approximation is proposed in [34]. In [35], Srebro et al. proposed a matrix factorization method to constrain the norms of U and V instead of their dimensionality. Salakhutdinov et al. presented a probabilistic linear model with Gaussian observation noise in [31]. In [30], the Gaussian-Wishart priors are placed on the user and item hyperparameters.

2.2 Trust-aware Recommender Systems

Traditional recommender systems have been well studied and developed both in academia and in industry, but they are all based on the assumption that users are independent and identically distributed, and ignore the relationships among users. Based on this intuition, many researchers have recently started to analyze trust-based recommender systems [2, 22, 23, 24, 25, 27].

Bedi et al. in [2] proposed a trust-based recommender system for the Semantic Web; this system runs on a server with the knowledge distributed over the network in the form of ontologies, and employs the Web of trust to generate the recommendations. In [24], a trust-aware method for recom-
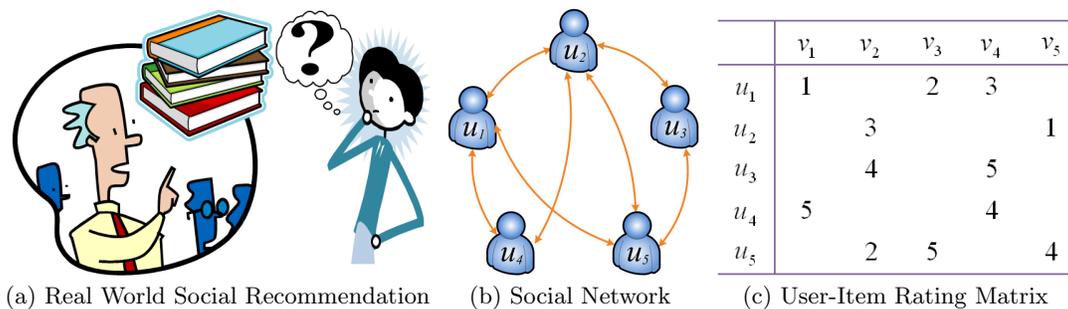
(a) Real World Social Recommendation (b) Social Network (c) User-Item Rating Matrix

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

Figure 1: A Toy Example

mender system is proposed. In this work, the collaborative filtering process is informed by the reputation of users, which is computed by propagating trust. Trust values are computed in addition to similarity measures between users. The experiments on a large real dataset show that this work increases the coverage (number of ratings that are predictable) while not reducing the accuracy (the error of predictions). In [27], two trust-aware methods are proposed to improve standard collaborative filtering methods. The experimental analysis shows that these trust information can help increase recommendation accuracy. In recent work proposed in [23], Ma et al. developed a factor analysis method based on the probabilistic graphical model, which fuses the user-item matrix with the users' social trust networks by sharing a common latent low-dimensional user feature matrix. The experimental analysis shows that this method generates better recommendations than the traditional collaborative filtering algorithms. The disadvantage of this work is that it lacks physical interpretations, which does not reflect the real world recommendation process. Hence, in order to model trust-aware recommender systems more realistically, in [22], the authors interpret one user's final rating decision as the balance between this user's own taste and his/her trusted users' favors. Finally, an ensemble probabilistic matrix factorization method is proposed to interpret this intuition. The experimental results illustrate that this approach can better model the problem.

The trust-aware recommender systems open a new direction in the research of recommender systems. However, as analyzed in Section 1, they have several inherent limitations.

2.3 Social Recommender Systems

How to utilize social network information has been extensively studied in many research tasks. In [20], in order to predict review quality, Lu et al. proposed a generic framework for incorporating social context information (about authors' identities and social networks) by adding regularization constraints to the text-based predictor. The experimental results show that incorporating social contextual information can help improve the accuracy of review quality prediction especially when the available training data is sparse. In [26], Mei et al. proposed a method combines topic modeling and social network analysis, and leveraged the power of both statistical topic models and discrete regularization. The proposed method can be applied to a wide range of text mining problems such as author-topic analysis, community discovery, and spatial text mining.

The concept "Social Recommender Systems" we mentioned in this paper is defined as using social friends network to improve recommender systems. There are also a few related work in the literature [7, 18, 23, 37] which studies social recommendation problems. However, there are some disadvantages of previous work. Although the method proposed in [37] is called "social recommendation", it essentially does not utilize any social network information. It only explores the similar users to generate recommendations. In [23], the proposed social recommendation method is actually a trust-aware method since it utilizes trust information in the experimental analysis. The approaches proposed in [7, 18] only either utilize very simple heuristics in making recommendations or use neighborhood-based method similar with the approach proposed in [24] to generate social recommendations. Hence, the social recommendation problem is not well studied in previous work. In this paper, we systematically analyze the social recommendation problem based on matrix factorization framework. We also elaborate the detailed differences between social recommender systems and trust-aware recommender systems.

3. PROBLEM DEFINITION

The problem we study in this paper is different from traditional recommender systems since the latter normally only considers the user-item rating matrix. In this paper, we will also incorporate users' social network information to improve recommender systems. Figure 1(a) shows the social process of recommendation in the real world. Actually, this scenario includes two central elements of social recommendation: the friends network and the favors of these friends, which can essentially be modeled by the examples of social network graph in Figure 1(b) and user-item matrix in Figure 1(c), respectively.

In this toy example, totally, there are 5 users (from u_1 to u_5) and 5 items (from v_1 to v_5) with user social relations (edges). Each edge represents the social connection between two users. Normally, the edges in social friend network are bidirectional. This is different with social trust network since user u_1 trusts user u_2 does not necessary mean user u_2 should trust user u_1. Typically, each user also rates some items on a 5-point integer scale to express the extent of the favor of each item (normally, 1, 2, 3, 4 and 5 represent "hate", "don't like", "neutral", "like" and "love", respectively).

The problem we study in this paper is how to effectively and efficiently predict the missing values of the user-item matrix by employing these different data sources.

4. LOW-RANK MATRIX FACTORIZATION

An efficient and effective approach to recommender systems is to factorize the user-item rating matrix, and utilize the factorized user-specific and item-specific matrices to make further missing data prediction [28, 30, 31, 39]. The premise behind a low-dimensional factor model is that there is only a small number of factors influencing the preferences, and that a user's preference vector is determined by how each factor applies to that user [28].

In this paper, we consider an $m \times n$ rating matrix R describing m users' numerical ratings on n items. A low-rank matrix factorization approach seeks to approximate the rating matrix R by a multiplication of l-rank factors,

$$R \approx U^T V, \tag{1}$$

where $U \in \mathbb{R}^{l \times m}$ and $V \in \mathbb{R}^{l \times n}$ with $l < \min(m, n)$. Since in the real world, each user only rates a very small portion of items, the matrix R is usually extremely sparse.

Traditionally, the Singular Value Decomposition (SVD) method is utilized to approximate a rating matrix R by minimizing

$$\frac{1}{2} \|R - U^T V\|_F^2, \tag{2}$$

where $\| \cdot \|_F^2$ denotes the Frobenius norm. However, due to the reason that R contains a large number of missing values, we only need to factorize the observed ratings in matrix R. Hence, we change Equation 2 to

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (R_{ij} - U_i^T V_j)^2, \tag{3}$$

where I_{ij} is the indicator function that is equal to 1 if user u_i rated item v_j and equal to 0 otherwise. In order to avoid overfitting, two regularization terms are added into Equation 3. Hence we have

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2, \tag{4}$$

where $\lambda_1, \lambda_2 > 0$. The optimization problem in Equation 4 minimizes the sum-of-squared-errors objective function with quadratic regularization terms. Gradient based approaches can be applied to find a local minimum. It also contains a nice probabilistic interpretation with Gaussian observation noise, which is detailed in [31]. The above algorithm is perhaps one of the most popular methods in collaborative filtering.

5. SOCIAL REGULARIZATION

In this section, we will systematically interpret how to model social network information as regularization terms to constrain the matrix factorization framework. Section 5.1 will detail the average-based regularization method while Section 5.2 introduces the individual-based regularization approach.

5.1 Model 1: Average-based Regularization

As mentioned in Section 1, we always turn to our friends for movie, music or book recommendations in the real world since we believe the tastes of our friends. Sometimes, in order to make a decision, we will consult lots of our friends for valuable suggestions. Based on this intuition, we propose

our first social recommendation model based on matrix factorization technique,

$$
\begin{aligned}
\min_{U,V} \mathcal{L}_1(R, U, V) \;=\; & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (R_{ij} - U_i^T V_j)^2 \\
& + \frac{\alpha}{2} \sum_{i=1}^m \|U_i - \frac{1}{|\mathcal{F}^+(i)|} \sum_{f \in \mathcal{F}^+(i)} U_f\|_F^2 \\
& + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2,
\end{aligned}
\tag{5}
$$

where $\alpha > 0$, $\mathcal{F}^+(i)$ is the set of friends of user u_i, and $|\mathcal{F}^+(i)|$ denotes the number of friends in the set $\mathcal{F}^+(i)$. Actually, we use the notation $\mathcal{F}^+(i)$ to denote u_i's outlink friends, and use the notation $\mathcal{F}^-(i)$ to represent u_i's inlink friends. In some networks, like the social network in Facebook, $\mathcal{F}^+(i)$ equals to $\mathcal{F}^-(i)$ since if u_i is in u_f's friend list then u_f must be in u_i's friend list. However, in some other networks, like trust network in Epinions, $\mathcal{F}^+(i)$ does not equal to $\mathcal{F}^-(i)$ since u_i trusts u_f does not indicate that u_f should also trust u_i.

In the above objective function, we impose a social regularization term

$$\frac{\alpha}{2} \sum_{i=1}^m \|U_i - \frac{1}{|\mathcal{F}^+(i)|} \sum_{f \in \mathcal{F}^+(i)} U_f\|_F^2 \tag{6}$$

to minimize the tastes between user u_i and u_i's friends. More specifically, if user u_i's friend list is $\mathcal{F}^+(i)$, then we could assume that u_i's taste U_i (feature vector) should be close to the average tastes of all the friends in $\mathcal{F}^+(i)$, which is $\frac{1}{|\mathcal{F}^+(i)|} \sum_{f \in \mathcal{F}^+(i)} U_f$.

This social regularization term makes an assumption that every user's taste is close to the average taste of this user's friends. However, this may not always be true in the real world. For example, in Facebook, it is very normal that one user may have hundreds or even thousands of friends. Among all of these friends, some friends may have similar tastes with this user, while some other friends may have totally different tastes. Hence, a more realistic model should treat all friends differently based on how similar they are. We thus change the social regularization term in Equation 6 to

$$\frac{\alpha}{2} \sum_{i=1}^m \|U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f)}\|_F^2, \tag{7}$$

where $Sim(i, f) \in [0, 1]$ is the similarity function to indicate the similarity between user u_i and user u_f. Hence, we have the objective function

$$
\begin{aligned}
\min_{U,V} \mathcal{L}_1(R, U, V) = & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (R_{ij} - U_i^T V_j)^2 \\
& + \frac{\alpha}{2} \sum_{i=1}^m \|U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f)}\|_F^2, \\
& + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2.
\end{aligned}
\tag{8}
$$

The similarity function $Sim(i, f)$ allows the social regularization term to treat users' friends differently. If user u_i and user u_f are very similar, say $Sim(i, f) = 0.95$, then user u_f should contribute more in the average taste of u_i's friends. On the other hand, if these two users are dissimilar,

say $Sim(i, f) = 0.1$, then u_f should contribute less. The details on how to calculate the values of similarity function $Sim(i, f)$ will be explained in Section 5.3.

A local minimum of the objective function given by Equation 8 can be found by performing gradient descent in feature vectors U_i and V_j,

$$\frac{\partial \mathcal{L}_1}{\partial U_i} = \sum_{j=1}^{n} I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_1 U_i,$$
$$+ \alpha(U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f)})$$
$$+ \alpha \sum_{g \in \mathcal{F}^-(i)} \frac{-Sim(i, g)(U_g - \frac{\sum_{f \in \mathcal{F}^+(g)} Sim(g, f) \times U_f}{\sum_{f \in \mathcal{F}^+(g)} Sim(g, f)})}{\sum_{f \in \mathcal{F}^+(g)} Sim(g, f)},$$
$$\frac{\partial \mathcal{L}_1}{\partial V_j} = \sum_{i=1}^{m} I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_2 V_j. \quad (9)$$

5.2 Model 2: Individual-based Regularization

The first model we propose imposes a social regularization term to constrain user u_i's taste with the average taste of his/her friends. However, this approach is insensitive to those users whose friends have diverse tastes. This will cause information loss problem, which will result in inaccurate modeling of feature vector U_i. Hence, in order to tackle this problem, we propose another social regularization term to impose constraints between one user and their friends individually:

$$\frac{\beta}{2} \sum_{i=1}^{m} \sum_{f \in \mathcal{F}^+(i)} Sim(i, f)\|U_i - U_f\|_F^2, \quad (10)$$

where $\beta > 0$, and $Sim(i, f)$ is the same similarity function we use in Equation 8, which will be detailed in Section 5.3. A small value of $Sim(i, f)$ indicates that the distance between feature vectors U_i and U_f should be larger, while a large value tells that the distance between the feature vectors should be smaller.

Thus, our second social recommendation model can be formulated as:

$$\min_{U,V} \mathcal{L}_2(R, U, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2$$
$$+ \frac{\beta}{2} \sum_{i=1}^{m} \sum_{f \in \mathcal{F}^+(i)} Sim(i, f)\|U_i - U_f\|_F^2$$
$$+ \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_F^2. \quad (11)$$

Another advantage of this approach is that it indirectly models the propagation of tastes. More specifically, if user u_i has a friend u_f and user u_f has a friend u_g (suppose u_i and u_g are not friends), we actually indirectly minimize the distance between feature vectors U_i and U_g when we are minimizing the distances

$$Sim(i, f)\|U_i - U_f\|_F^2 \text{ and } Sim(f, g)\|U_f - U_g\|_F^2.$$

The propagation of tastes will reach a harmonic status when the values of the objective function converge in the learning phase.

Similar to the first model, a local minimum of the objective function given by Equation 11 can also be found by

performing gradient descent in latent feature vectors U_i and V_j,

$$\frac{\partial \mathcal{L}_2}{\partial U_i} = \sum_{j=1}^{n} I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_1 U_i$$
$$+ \beta \sum_{f \in \mathcal{F}^+(i)} Sim(i, f)(U_i - U_f)$$
$$+ \beta \sum_{g \in \mathcal{F}^-(i)} Sim(i, g)(U_i - U_g),$$
$$\frac{\partial \mathcal{L}_2}{\partial V_j} = \sum_{i=1}^{m} I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_2 V_j. \quad (12)$$

5.3 Similarity Function

In Section 5.1 and Section 5.2, the proposed two social regularization terms require the knowledge of similarities between users. Since we have the rating information of all the users, the evaluation of similarities between two users can be calculated by measuring the issued ratings of these two users. There are two very popular methods we can borrow in the literature, which are Vector Space Similarity (VSS) and Pearson Correlation Coefficient (PCC) [4]. VSS is employed to define the similarity between two users i and f based on the items they rated in common:

$$Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} R_{ij} \cdot R_{fj}}{\sqrt{\sum_{j \in I(i) \cap I(f)} R_{ij}^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} R_{fj}^2}}, \quad (13)$$

where j belongs to the subset of items which user i and user f both rated. R_{ij} is the rate user i gave item j. From the above definition, we can see that VSS similarity in $Sim(i, f)$ is within the range $[0, 1]$, and a larger value means users i and f are more similar.

Actually, the similarity calculation in VSS does not consider that different users may have different rating styles. Some users may potentially give a higher ratings to all the products while some other users probably tend to issue lower ratings. Hence, PCC is proposed to solve this problem:

$$Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R}_i) \cdot (R_{fj} - \overline{R}_f)}{\sqrt{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R}_i)^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} (R_{fj} - \overline{R}_f)^2}}, \quad (14)$$

where $\overline{R}_i$ represents the average rate of user i. From this definition, user similarity $Sim(i, f)$ is ranging from $[-1, 1]$, and a larger value means users i and f are more similar. For consistency with VSS similarities, we employ a mapping function $f(x) = (x + 1)/2$ to bound the range of PCC similarities into $[0, 1]$.

5.4 Extensions

Our proposed method is quite general, and it can be easily extended to incorporate other contextual information, including social tags issued by users, movie genres, user demographic information, etc. By taking advantages of these information, we can use Cosine similarity of other similarity calculation methods to compute the affinities between users or items. Then we can plug in those similar users or items

Table 1: Statistics of User-Item Matrix of Douban

Statistics	User	Item
Min. Num. of Ratings	1	1
Max. Num. of Ratings	6,328	49,504
Avg. Num. of Ratings	129.98	287.51

Table 2: Statistics of Friend Network of Douban

Statistics	Friends per User
Max. Num.	986
Avg. Num.	13.07

Table 3: Statistics of User-Item Matrix of Epinions

Statistics	User	Item
Max. Num. of Ratings	1960	7082
Avg. Num. of Ratings	12.21	7.56

Table 4: Statistics of Trust Network of Epinions

Statistics	Trust per User	Be Trusted per User
Max. Num.	1763	2443
Avg. Num.	9.91	9.91

into our social regularization framework to further improve the performance of recommender systems. We do not discuss the details in this paper since our focus of this paper is to illustrate how to incorporate social friends information.

6. EXPERIMENTAL ANALYSIS

In this section, we conduct several experiments to compare the recommendation qualities of our approaches with other state-of-the-art recommendation methods.

6.1 Datasets

Our proposed models are quite general, and can be utilized to both social recommender systems and trust-aware recommender systems. Hence, in this paper, we evaluate our methods on two datasets: one dataset contains a social friend network while another one has a trust network.

The first data source we choose is Douban[1]. Douban, launched on March 6, 2005, is a Chinese Web 2.0 Web site providing user rating, review and recommendation services for movies, books and music. It is also the largest online Chinese language book, movie and music database and one of the largest online communities in China. Users can assign 5-scale integer ratings (from 1 to 5) to movies, books and music. It also provides Facebook-like social networking services, which allows users to find their friends through their email accounts[2]. This means that most of the friends on Douban actually know each other offline. Hence, Douban is an ideal source for our research on social recommendation.

Users on Douban can join different interesting groups. Currently, there are more than 700 groups under the "Movie" subcategory. We crawled all the users in these groups, and used these users as seeds to further crawl their social networks with their movie ratings. Finally, we obtain 129,490 unique users and 58,541 unique movies with 16,830,839 movie ratings. As to the social friend network, the total number of friend links between users is 1,692,952. The statistics of the Douban user-item rating matrix and social friend network are summarized in Table 1 and Table 2, respectively.

The second dataset we employ for evaluation is Epinions[3]. Epinions is a well-known general consumer review site that was established in 1999. At Epinions, visitors can read reviews about a variety of items to help them decide on a

---

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

[2] At the time when we were crawling the Douban dataset, Douban only allowed Facebook-like relationship building approach. Now Douban also supports Twitter-like following mechanism.

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

purchase or they can join for free and begin writing reviews that may earn them reward and recognition. To post a review, members need to first rate the product or service on a rating scale from 1 to 5 stars. Every member of Epinions maintains a "trust" list which presents a network of trust relationships between users, and a "block (distrust)" list which presents a network of distrust relationships. This network is called the "Web of trust", and is used by Epinions to reorder the product reviews such that a user first sees reviews by users that they trust.

The dataset we collected from Epinions consists of 51,670 users who have rated a total of 83,509 different items. The total number of ratings is 631,064. As to the user social trust network, the total number of issued trust statements is 511,799. Other statistics of this dataset are respectively shown in Table 3 and Table 4.

6.2 Metrics

We use two popular metrics, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), to measure the prediction quality of our proposed approach in comparison with other collaborative filtering and trust-aware recommendation methods.

The metric MAE is defined as:

$$MAE = \frac{1}{T} \sum_{i,j} |R_{ij} - \widehat{R}_{ij}|, \tag{15}$$

where R_{ij} denotes the rating user i gave to item j, $\widehat{R}_{ij}$ denotes the rating user i gave to item j as predicted by a method, and T denotes the number of tested ratings. The metric RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{T} \sum_{i,j} (R_{ij} - \widehat{R}_{ij})^2}. \tag{16}$$

From the definitions, we can see that a smaller MAE or RMSE value means a better performance.

6.3 Comparisons

In this section, in order to show the effectiveness of our proposed recommendation approaches, we compare the recommendation results of the following methods:

1. **UserMean**: this method uses the mean value of every user to predict the missing values.

2. **ItemMean**: this method utilizes the mean value of every item to predict the missing values.

Dataset	Training	Metrics	UserMean	ItemMean	NMF	PMF	RSTE	SR1$_{vss}$	SR1$_{pcc}$	SR2$_{vss}$	SR2$_{pcc}$
Douban	80%	MAE	0.6809	0.6288	0.5732	0.5693	0.5643	0.5579	0.5576	0.5548	**0.5543**
		Improve	18.59%	11.85%	3.30%	2.63%	1.77%				
		RMSE	0.8480	0.7898	0.7225	0.7200	0.7144	0.7026	0.7022	0.6992	**0.6988**
		Improve	17.59%	11.52%	3.28%	2.94%	2.18%				
	60%	MAE	0.6823	0.6300	0.5768	0.5737	0.5698	0.5627	0.5623	0.5597	**0.5593**
		Improve	18.02%	11.22%	3.03%	2.51%	1.84%				
		RMSE	0.8505	0.7926	0.7351	0.7290	0.7207	0.7081	0.7078	0.7046	**0.7042**
		Improve	17.20%	11.15%	4.20%	3.40%	2.29%				
	40%	MAE	0.6854	0.6317	0.5899	0.5868	0.5767	0.5706	0.5702	0.5690	**0.5685**
		Improve	17.06%	10.00%	3.63%	3.12%	1.42%				
		RMSE	0.8567	0.7971	0.7482	0.7411	0.7295	0.7172	0.7169	0.7129	**0.7125**
		Improve	16.83%	10.61%	4.77%	3.86%	2.33%				
Epinions	90%	MAE	0.9134	0.9768	0.8712	0.8651	0.8367	0.8290	0.8287	0.8258	**0.8256**
		Improve	9.61%	15.48%	5.23%	4.57%	1.33%				
		RMSE	1.1688	1.2375	1.1621	1.1544	1.1094	1.0792	1.0790	1.0744	**1.0739**
		Improve	8.12%	13.22%	7.59%	6.97%	3.20%				
	80%	MAE	0.9285	0.9913	0.8951	0.8886	0.8537	0.8493	0.8491	0.8447	**0.8443**
		Improve	9.07%	14.83%	5.68%	4.99%	1.10%				
		RMSE	1.1817	1.2584	1.1832	1.1760	1.1256	1.1016	1.1013	1.0958	**1.0954**
		Improve	7.30%	12.95%	7.42%	6.85%	2.68%				

3. **NMF**: this method is originally proposed in [16] for image analysis. However, it is widely used in collaborative filtering recently. It only uses user-item matrix for recommendations.

4. **PMF**: this method is proposed by Salakhutdinov and Minh in [31]. It also only uses user-item matrix for recommendations.

5. **RSTE**: this approach is proposed in [22]. It is a trust-aware recommendation method that models one user's ratings as the balance between this user's own favors and the tastes of his/her trusted users.

As we mentioned in the related work, there is no representative or suitable social recommendation method we can compare with in the literature. Hence, among all the above algorithms, we include a state-of-the-art trust-aware recommendation algorithms RSTE for experimental study.

We call our method proposed in Section 5.1 SR1, and we name the approach proposed in Section 5.2 SR2. Since we also have two different similarity functions in Section 5.3 that can be used in SR1 and SR2, we thus use the name SR1$_{vss}$ to represent SR1 method with VSS similarity function and SR1$_{pcc}$ to indicate SR1 method with PCC similarity function. The notations SR2$_{vss}$ and SR2$_{pcc}$ also have similar meanings.

For the Douban dataset, we use different training data settings (80%, 60% and 40%) to test the algorithms. Training data 80%, for example, means we randomly select 80% of the ratings from user-item rating matrix as the training data to predict the remaining 20% of ratings. For the Epinions dataset, since the dataset is much sparser, we use 90% and 80% as the training data settings. The random selection was carried out 5 times independently, and we report the average results. In all the experiments conducted in this paper, the values of λ_1 and λ_2 are set to a trivial value 0.001. The experimental results using 10 dimensions to represent the latent features are shown in Table 5. In this table, parameters α and β in our two models are set to 0.001 on Douban dataset and 0.01 on Epinions dataset. The standard deviations of the results generated of our method are all around 0.004.

From the results, we can observe that our methods consistently outperform other approaches in all the settings of both datasets. The percentages in Table 5 are the improvements of our SR2$_{PCC}$ method over the corresponding approaches. Our methods can generate better results than the state-of-the-art trust-aware recommendation algorithm RSTE. The MAE and RMSE values generated by all the methods on Epinions dataset are much higher than those on Douban dataset. This indicates that the Epinions trust network probably contains lots of noises. Since our algorithms treat one user's trusted users differently based on how similar they are, we can reduce some of the errors that are caused by those noises.

We also notice totally different characteristics of these two datasets. We find that in Douban dataset, the method Item-Mean outperforms UserMean while in Epinions dataset, the method UserMean outperforms ItemMean. This can be explained by the statistics in Table 1 and Table 3. In Douban dataset, the average number of ratings per item is much higher than the average number of ratings per user. In this case, item-based method can generally generate better results since every item has more information to use. However, in Epinions dataset, the average number of ratings per item is lower than the average number of ratings per user. This is the possible cause of why the user-based method has better performance.

Among our two models, the second model proposed in Section 5.2 outperforms the first model proposed in Section 5.1. We also notice that PCC-based methods are slightly better than VSS-based methods, which shows that PCC similarity function is generally more accurate in capturing the similarities between two users.
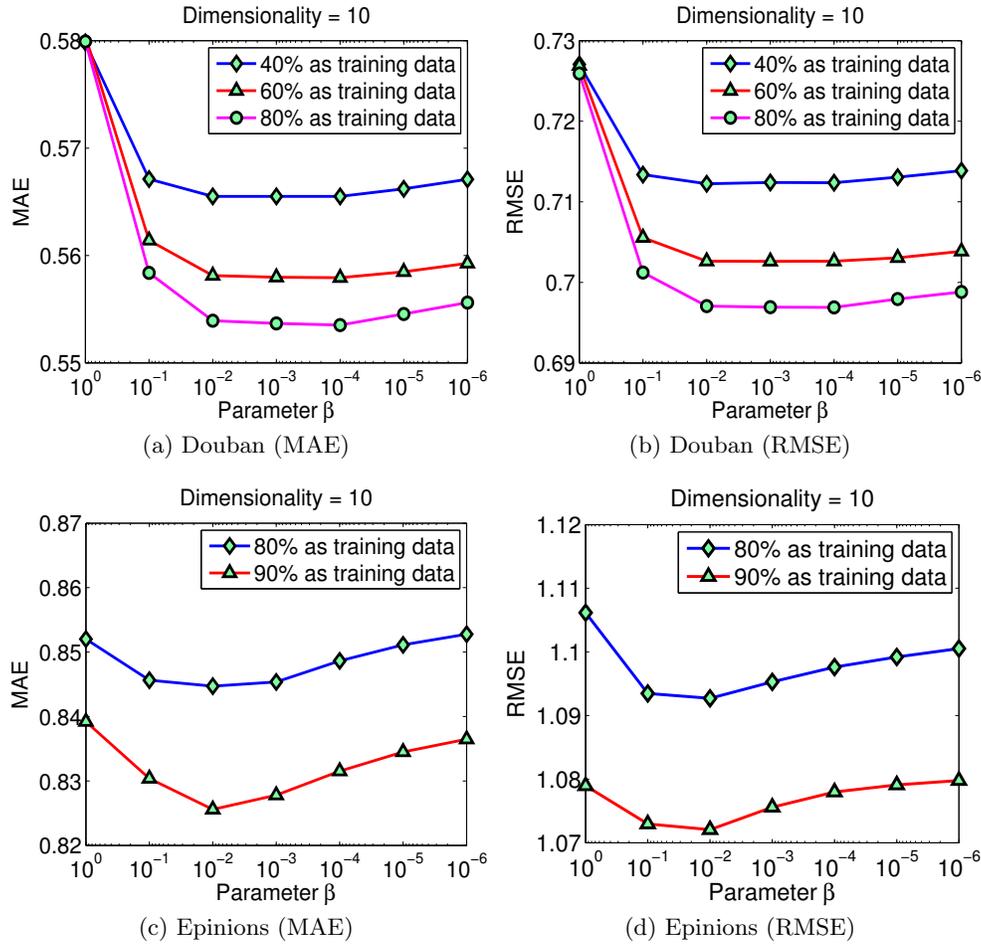
Figure 2: Impact of Parameter β (Dimensionality = 10)

6.4 Impact of Parameters α and β

In our methods proposed in this paper, the parameters α and β play very important roles. They control how much our methods should incorporate the information of the social network. In the extreme case, if we use a very small value of α or β, we only mine the user-item rating matrix for matrix factorization, and simply employ users' own tastes in making recommendations. On the other side, if we employ a very large value of α or β, the social network information will dominate the learning processes. In many cases, we do not want to set α and β to these extreme values since they will potentially hurt the recommendation performance. In this section, we analyze how the changes of α and β can affect the final recommendation accuracy.

The impact of α generally shares the same trend as the impact of β. Hence we only illustrate the results of β here due to the space limitation. Figure 2 shows the impacts of β on MAE and RMSE in our second model. We observe that the value of β impacts the recommendation results significantly, which demonstrates that incorporating the social network information greatly improves the recommendation accuracy. From the results, we can see that no matter using which dataset and which training data setting, as β increases, the MAE and RMSE values decrease (prediction accuracy increases) at first, but when β goes below

a certain threshold like 0.01 on Epinions dataset, the MAE and RMSE values increase (prediction accuracy decreases) with further increase of the value of β. The existence of the yielding point confirms with the intuition that purely using the user-item rating matrix or purely using the social network information for recommendations cannot generate better performance than appropriately integrating these two sources together.

6.5 Impact of Similarity Functions

The similarity function $Sim(i, f)$ measures how similar users u_i and u_f are. It can help distinguish friends with different tastes. In this paper, we employ two popular similarity functions: PCC and VSS. In order to examine how much the similarity function $Sim(i, f)$ contributes to the whole social regularization framework, we also conduct an experimental analysis on some special similarity settings. We are especially interested in two cases:

1. What is the performance of our social regularization framework if we discard the similarity calculation function $Sim(i, f)$? This case is also equivalent to setting all the similarities between friends to 1.

2. What if we assign a random similarity to any pair of friendship?

Table 6: Similarity Analysis (Dimensionality = 10)

Dataset	Training	Metrics	SR2 Sim=1	SR2 Sim=Ran	$SR2_{vss}$	$SR2_{pcc}$
Douban	80%	MAE	0.5579	0.5592	0.5548	0.5543
		RMSE	0.7034	0.7047	0.6992	0.6988
	60%	MAE	0.5631	0.5643	0.5597	0.5593
		RMSE	0.7083	0.7098	0.7046	0.7042
	40%	MAE	0.5724	0.5737	0.5690	0.5685
		RMSE	0.7195	0.7209	0.7129	0.7125
Epinions	90%	MAE	0.8324	0.8345	0.8258	0.8256
		RMSE	1.0794	1.0809	1.0744	1.0739
	80%	MAE	0.8511	0.8530	0.8447	0.8443
		RMSE	1.1002	1.1018	1.0958	1.0954

We evaluate the results on our second social regularization model SR2, and the comparison is shown in Table 6. In this table, Sim=1 means we set all the similarities between friends to 1 while Sim=Ran indicates that all the similarities are replaced by random values within the range [0,1]. We can see that SR2 with these two settings all perform worse than Sim_{vss} and Sim_{pcc}. This observation demonstrates the importance of similarity function $Sim(i, f)$. We cannot either discard this function from our framework or simply use some random values to represent the friend relationships.

7. CONCLUSION AND FUTURE WORK

In this paper, we focus on the social recommendation problem which is rarely studied in the literature before. Based on the intuition that users' social friend information can help improve the prediction accuracy of recommender systems, we propose two social recommendation algorithms that impose social regularization terms to constrain matrix factorization objective functions. The proposed methods are quite general since they can also be applied to solve the trust-aware recommendation problems. The experimental analysis on two large real-world datasets shows that our proposed approaches indicate a promising future.

In our proposed social regularization methods, we actually utilize all the social connections of each user. But the existence of some social connections may hurt the recommendation performance. Moreover, in the real world, we normally will not consult all our friends before making decisions. We may ask some friends for movie recommendations since these friends are experts on reviewing movies. At the same time, we may ask another group of friends for some other recommendation tasks. Hence, in order to model the social recommender systems more realistically, in the future, we need to design an effective algorithm to identify the most suitable group of friends for different recommendation tasks. This is a research direction worthy of further exploration, and it may need to develop a scalable and effective user clustering method.

The similarity function in this paper plays as an important role in distinguishing different tastes between users. Currently, we only utilize the traditional PCC and VSS for similarity computations. However, if we have more information about the users, such as the users' clicking behaviors, purchasing histories or tagging records, we can design a more complete and accurate similarity calculation function.

In this paper, we only constrain user feature vectors while ignoring the item side. Actually, if we can find some correlations between the items, we can also incorporate item regularization terms into our framework to further improve the prediction accuracy. We can achieve this if we have social tagging data in the recommender systems. The tags associated with each item can be treated as features of each item. For every item, we then find some similar items using similarity calculation functions, like Cosine method. Finally, these similar items can be added as item regularization terms in our matrix factorization framework. As analyzed in Section 5.4, we can further incorporate more contextual information to improve recommender systems if we have sufficient data.

As the rapid growth of online social network sites continues, we believe the social-based research will become more and more popular and important. We also plan to develop similar techniques to allow one user's friends to influence this user's search results, query suggestions. This would be an interesting social search problem to explore in the future.

Acknowledgments

8. REFERENCES

[1] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent dirichlet allocation. In *Proc. of WSDM '10*, pages 91–100, New York, New York, USA, 2010.

[2] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In *Proc. of IJCAI'07*, pages 2677–2682, 2007.

[3] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proc. of KDD '07*, pages 95–104, San Jose, California, USA, 2007.

[4] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of UAI '98*, 1998.

[5] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proc. of SIGIR '02*, pages 238–245, Tampere, Finland, 2002.

[6] M. Deshpande and G. Karypis. Item-based top-n recommendation. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

[7] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proc. of RecSys '09*, pages 53–60, New York, New York, USA, 2009.

[8] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. of SIGIR '03*, pages 259–266, Toronto, Canada, 2003.

[9] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[10] Z. Huang, H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, 2004.

[11] M. Jahrer, A. Töscher, and R. Legenstein. Combining predictions for accurate recommender systems. In *Proc. of KDD '10*, pages 693–702, Washington, DC, USA, 2010.

[12] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proc. of SIGIR '04*, pages 337–344, Sheffield, United Kingdom, 2004.

[13] A. Kohrs and B. Merialdo. Clustering for collaborative filtering applications. In *Proceedings of CIMCA*, 1999.

[14] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of KDD '08*, pages 426–434, Las Vegas, Nevada, USA, 2008.

[15] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. of KDD '09*, pages 447–456, Paris, France, 2009.

[16] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct. 1999.

[17] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, pages 76–80, Jan/Feb 2003.

[18] F. Liu and H. J. Lee. Use of social network information to enhance collaborative filtering performance. *Expert Syst. Appl.*, 37(7):4772–4778, 2010.

[19] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proc. of SIGIR '08*, pages 83–90, Singapore, Singapore, 2008.

[20] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *Proc. of WWW '10*, pages 691–700, Raleigh, North Carolina, USA, 2010.

[21] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *Proc. of SIGIR '07*, pages 39–46, Amsterdam, The Netherlands, 2007.

[22] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proc. of SIGIR '09*, pages 203–210, Boston, MA, USA, 2009.

[23] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: Social recommendation using probabilistic matrix factorization. In *Proc. of CIKM '08*, pages 931–940, Napa Valley, USA, 2008.

[24] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In *Proceedings of CoopIS/DOA/ODBASE*, pages 492–508, 2004.

[25] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proc. of RecSys '07*, pages 17–24, Minneapolis, MN, USA, 2007.

[26] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proc. of WWW '08*, pages 101–110, Beijing, China, 2008.

[27] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proc. of IUI '05*, pages 167–174, San Diego, California, USA, 2005.

[28] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of ICML '05*, 2005.

[29] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of CSCW '94*, 1994.

[30] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proc. of ICML '08*, 2008.

[31] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[32] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW '01*, pages 285–295, Hong Kong, Hong Kong, 2001.

[33] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *Proc. of ICML '03*, 2003.

[34] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proc. of ICML '03*, pages 720–727, 2003.

[35] N. Srebro, J. D. M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *NIPS*, 2004.

[36] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proc. of SIGIR '06*, pages 501–508, Seattle, Washington, USA, 2006.

[37] X. Xin, I. King, H. Deng, and M. R. Lyu. A social recommendation framework based on multi-scale continuous conditional random fields. In *Proc. of CIKM '09*, pages 1247–1256, Hong Kong, China, 2009.

[38] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proc. of SIGIR '05*, pages 114–121, Salvador, Brazil, 2005.

[39] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proc. of SIGIR '09*, pages 211–218, Boston, MA, USA, 2009.

[40] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proc. of SIGIR '07*, pages 47–54, Amsterdam, The Netherlands, 2007.

[41] S. Zhu, K. Yu, and Y. Gong. Stochastic relational models for large-scale dyadic data using mcmc. In *Advances in Neural Information Processing Systems*, volume 21, 2009.