SoRec: Social Recommendation Using Probabilistic Matrix Factorization

Hao Ma, Haixuan Yang, Michael R. Lyu, Irwin King
Dept. of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{hma, hxyang, lyu, king}@cse.cuhk.edu.hk

ABSTRACT

Data sparsity, scalability and prediction quality have been recognized as the three most crucial challenges that every collaborative filtering algorithm or recommender system confronts. Many existing approaches to recommender systems can neither handle very large datasets nor easily deal with users who have made very few ratings or even none at all. Moreover, traditional recommender systems assume that all the users are independent and identically distributed; this assumption ignores the social interactions or connections among users. In view of the exponential growth of information generated by online social networks, social network analysis is becoming important for many Web applications. Following the intuition that a person's social network will affect personal behaviors on the Web, this paper proposes a factor analysis approach based on probabilistic matrix factorization to solve the data sparsity and poor prediction accuracy problems by employing both users' social network information and rating records. The complexity analysis indicates that our approach can be applied to very large datasets since it scales linearly with the number of observations, while the experimental results shows that our method performs much better than the state-of-the-art approaches, especially in the circumstance that users have made few or no ratings.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Algorithms, Experimentation

Keywords

Recommender Systems, Collaborative Filtering, Social Recommendation, Matrix Factorization

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

1. INTRODUCTION

Recommender Systems attempt to suggest items (movies, books, music, news, Web pages, images, etc.) that are likely to interest the users. Typically, recommender systems are based on *Collaborative Filtering*, which is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items. The underlying assumption of collaborative filtering is that the active user will prefer those items which the similar users prefer [13]. Based on this simple but effective intuition, collaborative filtering has been widely employed in some large, famous commercial systems, such as Amazon[1]. However, due to the nature of collaborative filtering, recommender systems based on this technique suffer from the following inherent weaknesses: (1) Due to the sparsity of the user-item rating matrix (the density of available ratings in commercial recommender systems is often less than 1% [19]), memory-based [10, 12, 13, 24] collaborative filtering algorithms fail to find similar users, since the methods of computing similarities, such as the Pearson Correlation Coefficient (PCC) or the Cosine method, assume that two users have rated at least some items in common. Moreover, almost all of the memory-based and model-based [8, 9, 18, 20] collaborative filtering algorithms cannot handle users who have never rated any items. (2) In reality, we always turn to friends we trust for movie, music or book recommendations, and our tastes and characters can be easily affected by the company we keep. Hence, traditional recommender systems, which purely mine the user-item rating matrix for recommendations, give somewhat unrealistic output.

Traditional recommender systems assume that users are *i.i.d.* (independent and identically distributed); this assumption ignores the social interactions or connections among users. But the fact is, offline, social recommendation is an everyday occurrence. For example, when you ask a friend for a recommendation of a movie to see or a good restaurant, you are essentially soliciting a verbal social recommendation. Sinha et al. in [22] have shown that, given a choice between recommendations from friends and those from recommender systems, in terms of quality and usefulness, friends' recommendations are preferred, even though the recommendations given by the recommender systems have high novelty factor. Friends are seen as more qualified to make good and useful recommendations compared to traditional recommender systems [1]. From this point of view, the traditional recommender systems that ignore the social network structure of users may no longer be suitable.

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

In the most recent research work conducted in [21], by analyzing the who talks to whom social network on the MSN instant messenger[2] over 10 million people with their related search records on the Live Search Engine[3], P. Singla and M. Richardson revealed that people who chat with each other (using instant messaging) are more likely to share interests (their Web searches are the same or topically similar). Therefore, to improve the recommendation accuracy, in modern recommender systems, both social network structure and user-item rating matrix should be taken into consideration.

In order to overcome the weaknesses mentioned above, based on the intuition that a user's social network will affect her/his personal behaviors on the Web, we propose to fuse a user's social network graph with the user-item rating matrix in order to make more accurate and personalized recommendations, which is called *Social Recommendation*. Actually, the method we develop is applicable not only to social recommendation, but also to social search and many other tasks in information retrieval and data mining.

To achieve this goal, this paper proposes a method integrating social network structure and the user-item rating matrix, based on probabilistic factor analysis. We connect these two different data resources through the shared user latent feature space, that is, the user latent feature space in the social network structure is the same in the user-item rating matrix. By performing factor analysis based on probabilistic matrix factorization, the low-rank user latent feature space and item latent feature space are learned in order to make social recommendations. The experimental results on the Epinions[4] dataset shows that our method outperforms the state-of-the-art collaborative filtering algorithms, especially when active users have very few ratings or even none at all. Moreover, the complexity analysis indicates that our approach can be applied to very large datasets since it scales linearly with the number of observations.

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 presents our work on social recommendation. The results of an empirical analysis are presented in Section 4, followed by the conclusions and future work in Section 5.

2. RELATED WORK

In this section, we review several major approaches for recommender systems, especially for collaborative filtering. Two types of collaborative filtering approaches are widely studied: memory-based and model-based.

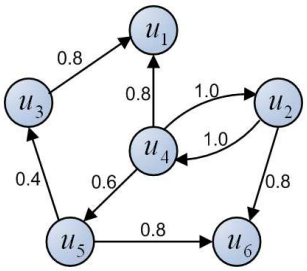
The memory-based approaches are the most popular prediction methods and are widely adopted in commercial collaborative filtering systems [12, 16]. The most analyzed examples of memory-based collaborative filtering include user-based approaches [2, 7, 10, 25] and item-based approaches [4, 12, 19]. User-based approaches predict the ratings of active users based on the ratings of similar users found, and 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 the PCC algorithm [16] and the VSS algorithm [2] as the similarity computation methods. PCC-based collaborative filtering generally can achieve higher performance than the other popular algorithm VSS, since it considers the differences of user rating style.

In the model-based approaches, training datasets are used to train a predefined model. Examples of model-based approaches include the clustering model [25], aspect models [8, 9, 20] and the latent factor model [3]. [11] presented an algorithm for collaborative filtering based on hierarchical clustering, which tried to balance 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, several matrix factorization methods [15, 17, 18, 23] have been proposed for collaborative filtering. These methods all focus on fitting the user-item rating matrix using low-rank approximations, and use it to make further predictions. The premise behind a low-dimensional factor model is that there is only a small number of factors influencing preferences, and that a user's preference vector is determined by how each factor applies to that user.

All the above methods for recommender systems are based on the assumption that users are independent and identically distributed, and ignores the social activities between users, which is not consistent with the reality that we normally ask friends for recommendations. Based on this intuition, many researchers have recently started to analyze trust-based recommender systems. In [14], a trust-aware collaborative filtering method for recommender systems 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 shows that this work increases the coverage (number of ratings that are predictable) while not reducing the accuracy (the error of predictions). Bedi et al. in [1] 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 uses the Web of trust to generate the recommendations. These methods are all memory-based methods which employ only heuristic algorithms to generate recommendations. There are several problems with this approach, however. The relationship between the trust network and the user-item matrix have not been studied systematically. Moreover, these methods are not scalable to very large datasets since they may need to calculate the pairwise user similarities and pairwise user trust scores.

In this paper, by conducting latent factor analysis using probabilistic matrix factorization, we learn the user latent feature space and item latent feature space by employing a user social network and a user-item matrix simultaneously and seamlessly. Although recently, similar factor analysis methods have been employed in [27, 28] for document retrieval and document classification, our approach has three essential differences compared with these methods: (1) Our method can deal with missing value problem, while their methods cannot. (2) Our method is interpreted using a probabilistic factor analysis model. (3) Complexity analysis shows that our method is more efficient than their methods and can be applied to very large datasets.

[2]http://www.msn.com
[3]http://www.live.com
[4]http://www.epinions.com



(a) Social Network Graph

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	5	2		3		4		
u_2	4	3			5			
u_3	4		2				2	4
u_4								
u_5	5	1	2		4	3		
u_6	4	3		2	4		3	5

(b) User-Item Matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	5	2	**2.5**	3	**4.8**	4	**2.2**	**4.8**
u_2	4	3	**2.4**	**2.9**	5	**4.1**	**2.6**	**4.7**
u_3	4	**1.7**	2	**3.2**	**3.9**	**3.0**	2	4
u_4	**4.8**	**2.1**	**2.7**	**2.6**	**4.7**	**3.8**	**2.4**	**4.9**
u_5	5	1	2	**3.4**	4	3	**1.5**	**4.6**
u_6	4	3	**2.9**	2	4	**3.4**	3	5

(c) Predicted User-Item Matrix

Figure 1: Example for Toy Data

3. SOCIAL RECOMMENDATION FRAMEWORK

In this section, we first demonstrate our social recommendation framework using a simple but illustrative toy example. Then we introduce the factor analysis method for social recommendation using probabilistic matrix factorization.

3.1 Toy Example

Let us first consider the typical social network graph in Fig. 1(a). There are 6 users in total (nodes, from u_1 to u_6) with 8 relations (edges) between users in this graph, and each relation is associated with a weight w_{ij} in the range $[0, 1]$ to specify how much user u_i knows or trusts user u_j. In an online social network Web site, the weight w_{ij} is often explicitly stated by user u_i. As illustrated in Fig. 1(b), each user also rates some items (from i_1 to i_8) on a 5-point integer scale to express the extent of favor of each item. The problem we study in this paper is how to predict the missing values of the user-item matrix effectively and efficiently by employing two different data sources. As mentioned in Section 1, motivated by the intuition that a user's social connections will affect this user's behaviors on the Web, we therefore factorize the social network graph and user-item matrix simultaneously and seamlessly using $U^T Z$ and $U^T V$, where the shared low-dimensional matrix U denotes the user latent feature space, Z is the factor matrix in the social network graph, and V represents the low-dimensional item latent feature space. If we use 5 dimensions to perform the matrix factorization for social recommendation, we obtain

$$U = \begin{bmatrix} 1.55 & 1.22 & 0.37 & 0.81 & 0.62 & -0.01 \\ 0.36 & 0.91 & 1.21 & 0.39 & 1.10 & 0.25 \\ 0.59 & 0.20 & 0.14 & 0.83 & 0.27 & 1.51 \\ 0.39 & 1.33 & -0.43 & 0.70 & -0.90 & 0.68 \\ 1.05 & 0.11 & 0.17 & 1.18 & 1.81 & 0.40 \end{bmatrix},$$

$$V = \begin{bmatrix} 1.00 & -0.05 & -0.24 & 0.26 & 1.28 & 0.54 & -0.31 & 0.52 \\ 0.19 & -0.86 & -0.72 & 0.05 & 0.68 & 0.02 & -0.61 & 0.70 \\ 0.49 & 0.09 & -0.05 & -0.62 & 0.12 & 0.08 & 0.02 & 1.60 \\ -0.40 & 0.70 & 0.27 & -0.27 & 0.99 & 0.44 & 0.39 & 0.74 \\ 1.49 & -1.00 & 0.06 & 0.05 & 0.23 & 0.01 & -0.36 & 0.80 \end{bmatrix},$$

where U_i and V_j are the column vectors and denote the latent feature vectors of user u_i and item v_j, respectively. Note that the solutions of U and V are not unique. Then we can predict the missing value w_{ij} in Fig. 1(b) using $U_i^T V_j$ (before prediction, we need to first transfer the value of $U_i^T V_j$ using logistic function $g(x)$ and another mapping function $f(x)$, which will be introduced in Section 3.2 and Section 3.3 respectively). Therefore, all the missing values can be predicted using 5-dimensional matrices U and V, as shown in Fig. 1(c). Note that even though user u_4 does not rate any items, our approach still can predict reasonable ratings.

Since this example is a toy example, we cannot evaluate the accuracy of the prediction. However, the experimental analysis in Section 4 based on Epinions dataset tests the effectiveness of our approach. In the following sections, we will present the details of how we conduct factor analysis for social recommendation using probabilistic matrix factorization.

3.2 Social Network Matrix Factorization

Suppose we have a directed social network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V} = \{v_i\}_{i=1}^m$ represents all the users in a social network and the edge set $\mathcal{E}$ represents the relations between users. Let $C = \{c_{ik}\}$ denote the $m \times m$ matrix of $\mathcal{G}$, which is also called the social network matrix in this paper. For a pair of vertices, v_i and v_k, let $c_{ik} \in (0, 1]$ denote the weight associated with an edge from v_i to v_k, and $c_{ik} = 0$, otherwise. The physical meaning of the weight c_{ik} can be interpreted as how much a user i trusts or knows user k in a social network. Note that C is an asymmetric matrix, since in a social network, especially in a trust-based social network, user i trusting k does not necessary indicate user k trusts i.

The idea of social network matrix factorization is to derive a high-quality l-dimensional feature representation U of users based on analyzing the social network graph $\mathcal{G}$. Let $U \in R^{l \times m}$ and $Z \in R^{l \times m}$ be the latent user and factor feature matrices, with column vectors U_i and Z_k representing user-specific and factor-specific latent feature vectors, respectively. We define the conditional distribution over the observed social network relationships as

$$p(C|U, Z, \sigma_C^2) = \prod_{i=1}^{m} \prod_{k=1}^{m} \left[\mathcal{N}\left(c_{ik}|g(U_i^T Z_k), \sigma_C^2\right) \right]^{I_{ik}^C}, \quad (1)$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and I_{ik}^C is the indicator function that is equal to 1 if user i trusts or knows user k and equal to 0 otherwise. The function $g(x)$ is the logistic function $g(x) = 1/(1+\exp(-x))$, which makes it possible to bound the range of $U_i^T Z_k$ within the range $[0, 1]$.

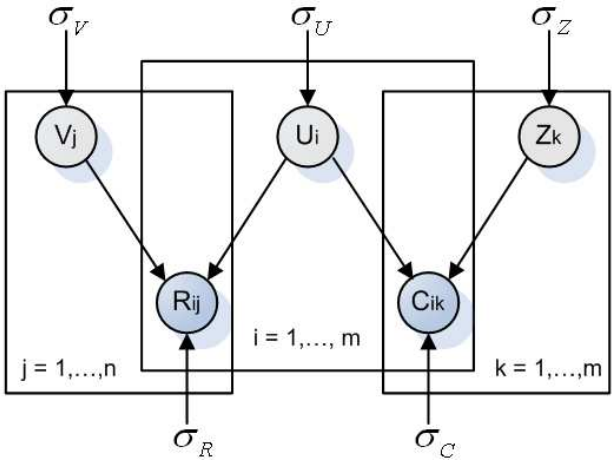


Figure 2: Graphical Model for Social Recommendation

We also place zero-mean spherical Gaussian priors [5, 18] on user and factor feature vectors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}),$$
$$p(Z|\sigma_Z^2) = \prod_{k=1}^{m} \mathcal{N}(Z_k|0, \sigma_Z^2 \mathbf{I}). \quad (2)$$

Hence, through a simple Bayesian inference, we have

$$\begin{aligned} p(U, Z|C, \sigma_C^2, \sigma_U^2, \sigma_Z^2) &\propto p(C|U, Z, \sigma_C^2) p(U|\sigma_U^2) p(Z|\sigma_Z^2) \\ &= \prod_{i=1}^{m}\prod_{k=1}^{m} \left[\mathcal{N}\left(c_{ik}|g(U_i^T Z_k), \sigma_C^2\right)\right]^{I_{ik}^C} \\ &\times \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \times \prod_{k=1}^{m} \mathcal{N}(Z_k|0, \sigma_Z^2 \mathbf{I}). \end{aligned} \quad (3)$$

In online social networks, the value of c_{ik}, which is mostly explicitly stated by user i with respect to user k, and it cannot accurately describe the relations between users since it contains noises and it ignores the graph structure information of social network. For instance, similar to the Web link adjacency graph in [26], in a trust-based social network, the confidence of trust value c_{ik} should be decreased if user i trusts lots of users; however the confidence of trust value c_{ik} should be increased if user k is trusted by lots of users. Hence, we employ the term c_{ik}^* which incorporates local authority and local hub values as a substitute for c_{ik} in Eq. (1),

$$p(C|U, Z, \sigma_C^2) = \prod_{i=1}^{m}\prod_{j=1}^{n} \left[\mathcal{N}\left(c_{ik}^*|g(U_i^T Z_k), \sigma_C^2\right)\right]^{I_{ik}^C},$$
$$c_{ik}^* = \sqrt{\frac{d^-(v_k)}{d^+(v_i) + d^-(v_k)}} \times c_{ik}, \quad (4)$$

where $d^+(v_i)$ represents the outdegree of node v_i, while $d^-(v_k)$ indicates the indegree of node v_k.

3.3 User-Item Matrix Factorization

Now considering the user-item matrix, suppose we have m users, n movies, and rating values within the range $[0, 1]$. Actually, most recommender systems use integer rating values from 1 to R_{max} to represent the users' judgements on items. In this paper, without loss of generality, we map the ratings $1, ..., R_{max}$ to the interval $[0, 1]$ using the function $f(x) = (x-1)/(R_{max}-1)$. Let r_{ij} represent the rating of user i for movie j, and $U \in R^{l\times m}$ and $V \in R^{l\times n}$ be latent user and movie feature matrices, with column vectors U_i and V_j representing user-specific and movie-specific latent feature vectors respectively. We define the conditional distribution over the observed ratings as

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^{m}\prod_{j=1}^{n} \left[\mathcal{N}\left(r_{ij}|g(U_i^T V_j), \sigma_R^2\right)\right]^{I_{ij}^R}, \quad (5)$$

where I_{ij}^R is the indicator function that is equal to 1 if user i rated movie j and equal to 0 otherwise. We also place zero-mean spherical Gaussian priors on user and movie feature vectors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}),$$
$$p(V|\sigma_V^2) = \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}). \quad (6)$$

Hence, similar to Eq. (3), through a Bayesian inference, we have

$$\begin{aligned} p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) &\propto p(R|U, V, \sigma_R^2) p(U|\sigma_U^2) p(V|\sigma_V^2) \\ &= \prod_{i=1}^{m}\prod_{j=1}^{n} \left[\mathcal{N}\left(r_{ij}|g(U_i^T V_j), \sigma_R^2\right)\right]^{I_{ij}^R} \\ &\times \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \times \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}). \end{aligned} \quad (7)$$

3.4 Matrix Factorization for Social Recommendation

As analyzed in Section 1, in order to reflect the phenomenon that a user's social connections will affect this user's judgement of interest in items, we model the problem of social recommendation using the graphical model described in Fig. 2, which fuses both the social network graph and the user-item rating matrix into a consistent and compact feature representation. Based on Fig. 2, the log of the posterior distribution for social recommendation is given by

$$\begin{aligned} \ln p(U, V, Z|C, R, \sigma_C^2, \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_Z^2) = \\ -\frac{1}{2\sigma_R^2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R (r_{ij} - g(U_i^T V_j))^2 \\ -\frac{1}{2\sigma_C^2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^C (c_{ik}^* - g(U_i^T Z_k))^2 \\ -\frac{1}{2\sigma_U^2}\sum_{i=1}^{m} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{n} V_j^T V_j - \frac{1}{2\sigma_Z^2}\sum_{k=1}^{m} Z_k^T Z_k \\ -\frac{1}{2}\left(\left(\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\right)\ln\sigma_R^2 + \left(\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^C\right)\ln\sigma_C^2\right) \\ -\frac{1}{2}\left(ml\ln\sigma_U^2 + nl\ln\sigma_V^2 + ml\ln\sigma_Z^2\right) + \mathcal{C}, \end{aligned} \quad (8)$$

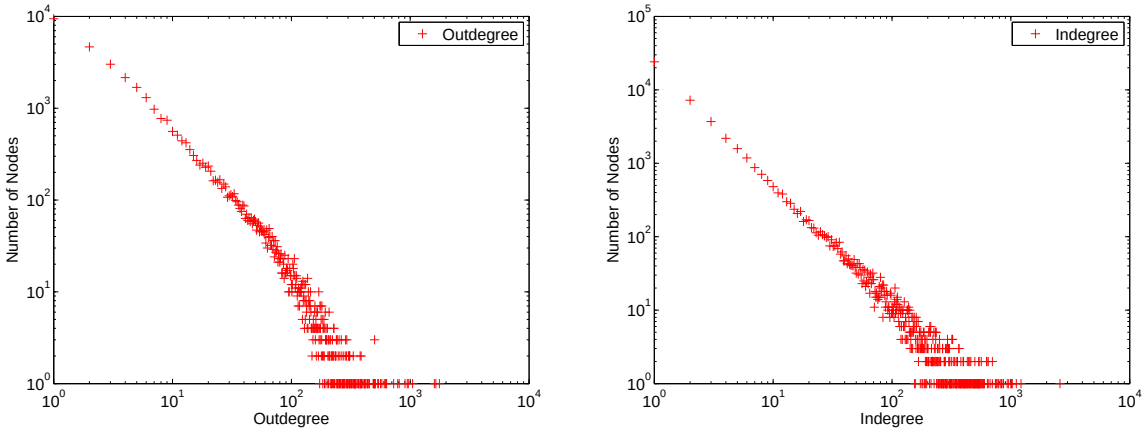


Figure 3: Degree Distribution of User Social Network

where $\mathcal{C}$ is a constant that does not depend on the parameters. Maximizing the log-posterior over three latent features with hyperparameters (i.e. the observation noise variance and prior variances) kept fixed is equivalent to minimizing the following sum-of-squared-errors objective functions with quadratic regularization terms:

$$\mathcal{L}(R, C, U, V, Z) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(r_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_C}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^{C}(c_{ik}^{*} - g(U_i^T Z_k))^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_Z}{2}\|Z\|_F^2, \quad (9)$$

where $\lambda_C = \sigma_R^2/\sigma_C^2$, $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, $\lambda_Z = \sigma_R^2/\sigma_Z^2$, and $\|\cdot\|_F^2$ denotes the Frobenius norm. A local minimum of the objective function given by Eq.(9) can be found by performing gradient descent in U_i, V_j and Z_k,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}^{R} g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})V_j \\
&\quad + \lambda_C \sum_{k=1}^{m} I_{ik}^{C} g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^{*})Z_k + \lambda_U U_i, \\
\frac{\partial \mathcal{L}}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}^{R} g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})U_i + \lambda_V V_j, \\
\frac{\partial \mathcal{L}}{\partial Z_k} &= \lambda_C \sum_{i=1}^{m} I_{ik}^{C} g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^{*})U_i + \lambda_Z Z_k, \quad (10)
\end{aligned}$$

where $g'(x)$ is the derivative of logistic function $g'(x) = \exp(x)/(1+\exp(x))^2$. In order to reduce the model complexity, in all of the experiments we conduct in Section 4, we set $\lambda_U = \lambda_V = \lambda_Z$.

3.5 Complexity Analysis

The main computation of gradient methods is evaluating the object function $\mathcal{L}$ and its gradients against variables. Because of the sparsity of matrices R and C, the computational complexity of evaluating the object function $\mathcal{L}$ is $O(\rho_R l + \rho_C l)$, where ρ_R and ρ_C are the numbers of nonzero entries in matrices R and C, respectively. The computational complexities for gradients $\frac{\partial \mathcal{L}}{\partial U}$, $\frac{\partial \mathcal{L}}{\partial V}$ and $\frac{\partial \mathcal{L}}{\partial Z}$ in Eq. (10) are $O(\rho_R l + \rho_C l)$, $O(\rho_R l)$ and $O(\rho_C l)$, respectively. Therefore, the total computational complexity in one iteration is $O(\rho_R l + \rho_C l)$, which indicates that the computational time of our method is linear with respect to the number of observations in the two sparse matrices. This complexity analysis shows that our proposed approach is very efficient and can scale to very large datasets.

Table 1: Statistics of User-Item Rating Matrix of Epinions

Statistics	User	Item
Min. Num. of Rated	1	1
Max. Num. of Rated	1022	2018
Avg. Num. of Rated	16.55	4.76

4. EXPERIMENTAL ANALYSIS

In this section, we conduct several experiments to compare the recommendation quality of our social recommendation approach with other state-of-the-art collaborative filtering methods. Our experiments are intended to address as the following questions:

1. How does our approach compare with the published state-of-the-art collaborative filtering algorithms?
2. How does the model parameter λ_C affect the accuracy of prediction?
3. What is the performance comparison on users with different observed ratings?
4. Can our algorithm achieve good performance even if users have no observed ratings?
5. Is our algorithm efficient for large datasets?

In the following, Section 4.3 gives answers to question 1, Section 4.4 addresses question 2, Section 4.5 describes experiments for questions 3 and 4, and lastly, Section 4.6 shows the analysis of question 5.

4.1 Description of the Epinions Dataset

A tremendous amount of data has been produced on the Internet every day over the past decade. Millions of people influence each other implicitly or explicitly through online

Table 2: MAE comparison with other approaches (A smaller MAE value means a better performance)

Training Data	Dimensionality = 5				Dimensionality = 10			
	MMMF	PMF	CPMF	SoRec	MMMF	PMF	CPMF	SoRec
99%	1.0008	0.9971	0.9842	**0.9018**	0.9916	0.9885	0.9746	**0.8932**
80%	1.0371	1.0277	0.9998	**0.9321**	1.0275	1.0182	0.9923	**0.9240**
50%	1.1147	1.0972	1.0747	**0.9838**	1.1012	1.0857	1.0632	**0.9751**
20%	1.2532	1.2397	1.1981	**1.1069**	1.2413	1.2276	1.1864	**1.0944**

social network services, such as Facebook[5]. As a result, there are many online opportunities to mine social networks for the purposes of social recommendations.

We choose Epinions as the data source for our experiments on social recommendation. Epinions.com is a well known knowledge sharing site and review site that was established in 1999. In order to add reviews, users (contributors) need to register for free and begin submitting their own personal opinions on topics such as products, companies, movies, or reviews issued by other users. Users can also assign products or reviews integer ratings from 1 to 5. These ratings and reviews will influence future customers when they are deciding whether a product is worth buying or a movie is worth watching. 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 re-order the product reviews such that a user first sees reviews by users that they trust. Epinions is thus an ideal source for experiments on social recommendation. Note that in this paper, we only employ trust statements between users while ignoring the distrust statements, for the following two reasons: (1) The distrust list of each user is kept private in Epinions.com in order to protect the privacies of users, hence it is not available in our dataset. (2) As presented in [6], the understanding of distrust is more complicated than trust, which indicates that the user trust latent feature space may not be the same as the user distrust latent feature space. The study of distrust-based social recommendation will be conducted in future work.

The dataset used in our experiments consists of 40,163 users who have rated at least one of a total of 139,529 different items. The total number of reviews is 664,824. The density of the user-item matrix is

$$\frac{664824}{40163 \times 139529} = 0.01186\%.$$

We can observe that the user-item matrix of Epinions is relatively very sparse, since the densities for the two most famous collaborative filtering datasets Movielens[6] (6,040 users, 3,900 movies and 1,000,209 ratings) and Eachmovie[7] (74,424 users, 1,648 movies and 2,811,983 ratings) are 4.25% and 2.29%, respectively. In particular, in the Movielens dataset all the users are guaranteed to have voted on at least 20 items, while in our Epinions dataset, 18,826 users, representing 46.87% of the population, submitted fewer than or equal to 5 reviews. Moreover, an important reason that we choose the Epinions dataset is that user social network information is not included in the Movielens and Eachmovie datasets. The statistics of the Epinions user-item rating matrix are summarized in Table 1.

As to the user social network, the total number of issued trust statements is 487,183. The indegree and outdegree distributions of this social network fit with a power-law distribution, as has been found in many social networks. The degree distributions of the Epinions social network are shown in Fig. 3.

[5]http://www.facebook.com

[6]http://www.cs.umn.edu/Research/GroupLens.

[7]http://www.research.digital.com/SRC/EachMovie. It is now retired by Hewlett-Packard (HP).

4.2 Metrics

We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approach in comparison with other collaborative filtering methods. MAE is defined as:

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r}_{i,j}|}{N}, \tag{11}$$

where $r_{i,j}$ denotes the rating user i gave to item j, $\widehat{r}_{i,j}$ denotes the rating user i gave to item j as predicted by our approach, and N denotes the number of tested ratings.

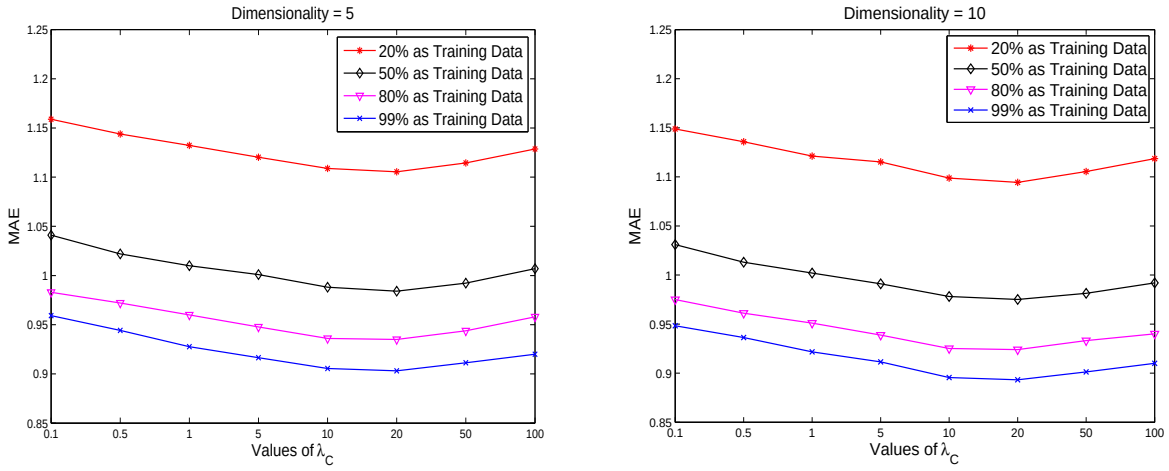
4.3 Comparison

In this section, in order to show the performance improvement of our Social Recommendation (SoRec) algorithm, we compare our algorithm with some state-of-the-art algorithms: Maximum Margin Matrix Factorization (MMMF) [15], Probabilistic Matrix Factorization (PMF) [18], and Constrained Probabilistic Matrix Factorization (CPMF) [18].

We use different amounts of training data (99%, 80%, 50%, 20%) to test all the algorithms. Training data 99%, for example, means we randomly select 99% of the ratings from Epinions dataset as the training data to predict the remaining 1% of ratings. The random selection was carried out 5 times independently. The experimental results are shown in Table 2. The parameter settings of our approach are $\lambda_C = 10$, $\lambda_U = \lambda_V = \lambda_Z = 0.001$, and in all the experiments conducted in the following sections, we set all of the parameters λ_U, λ_V and λ_Z equal to 0.001. From Table 2, we can observe that our approach outperforms the other methods. On average, our approach improves the accuracy by 11.01%, 9.98%, and 7.82% relative to MMMF, PMF and CPMF, respectively. The improvements are significant, which shows the promising future of our social recommendation approach.

4.4 Impact of Parameter λ_C

The main advantage of our social recommendation approach is that it incorporates the social network information, which helps predict users' preferences. In our model, parameter λ_C balances the information from the user-item rating matrix and the user social network. If $\lambda_C = 0$, we only mine the user-item rating matrix for matrix factorization, and if $\lambda_C = \inf$, we only extract information from the



(a) Dimensionality=5 (b) Dimensionality=10

Figure 4: Impact of Parameter λ_C

social network to predict users' preferences. In other cases, we fuse information from the user-item rating matrix and the user social network for probabilistic matrix factorization and, furthermore, to predict ratings for active users.

Fig. 4 shows the impacts of λ_C on MAE. We observe that the value of λ_C impacts the recommendation results significantly, which demonstrates that fusing the user-item rating matrix with the user social network greatly improves the recommendation accuracy. As λ_C increases, the prediction accuracy also increases at first, but when λ_C surpasses a certain threshold, the prediction accuracy decrease with further increase of the value of λ_C. This phenomenon coincides with the intuition that purely using the user-item rating matrix or purely using the user social network cannot generate better performance than fusing these two resources together. From Fig. 4, no matter using 5-dimension or 10-dimension representation, we observe that for this Epinions dataset, our social recommendation method achieves the best performance when $\lambda_C \in [10, 20]$, while smaller values like $\lambda_C = 0.1$ or larger values $\lambda_C = 100$ can potentially degrade the model performance. Moreover, the insensitivity of the optimal value of λ_C shows that the parameter of our model is easy to train.

4.5 Performance on Different Users

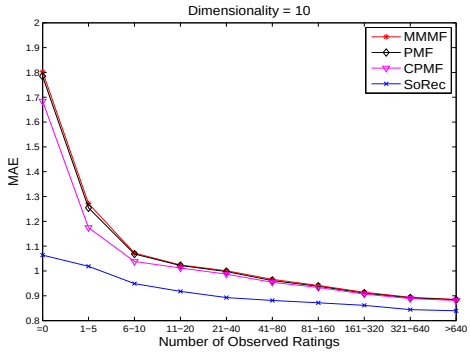
One main task we target in this paper is to provide accurate recommendations when users only supply a few ratings or even have no rating records. Although previous work always notices this critical problem, few approaches perform well when few user ratings are given. Hence, in order to compare our approach with the other methods thoroughly, we first group all the users based on the number of observed ratings in the training data, and then evaluate prediction accuracies of different user groups. The experimental results are shown in Fig. 5. Users are grouped into 10 classes: "$= 0$", "$1 - 5$", "$6 - 10$", "$11 - 20$", "$21 - 40$", "$41 - 80$", "$81 - 160$", "$160 - 320$", "$320 - 640$", and "> 640", denoting how many ratings users have rated.

Fig. 5(b), Fig. 5(d), Fig. 5(f) and Fig. 5(h) summarize the distributions of testing data according to groups in the training data. For an example, in Fig. 5(d), there are a total of 1,089 user-item pairs needing to be predicted in the testing dataset in which the related users in the training dataset have no rating records ("$= 0$"). Actually, Fig. 5(f) does not have the label "> 640" since no user has rated more than 640 items; Fig. 5(h) does not have the labels "$320 - 640$" and "> 640" for the same reason. In Fig. 5(a), Fig. 5(c), Fig. 5(e), and Fig. 5(g) we observe that our algorithm generally performs better than other methods, especially when few user ratings are given. When users have no rating records ("$= 0$"), our method performs much better than MMMF, PMF and CPMF, and increase the performance more than 36.75%, 40.82%, and 41.75%, respectively. Although as the number of users' observed ratings increases, the performances of all the algorithms converge, our model still generates better predictions than the other methods at all level.

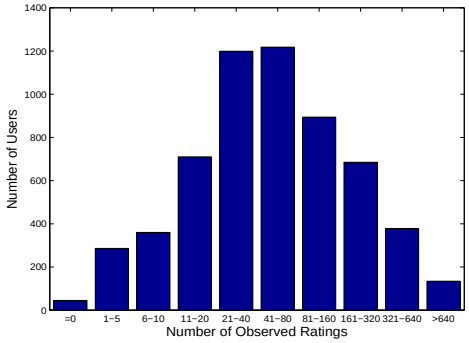
4.6 Efficiency Analysis

The complexity analysis in Section 3.5 states that the computational complexity of our approach is linear with respect to the number of ratings, which proves that our approach is scalable to very large datasets. Actually, our approach is very efficient even when using a very simple gradient descent method. In the experiments using 99% of the data as training data, each iteration only needs less than 1 second. Also, as shown in Fig. 6, when using 99% of the data as training data (Fig. 6(a)), our method needs less than 1,200 iterations to converge, which only needs approximately 18 minutes. When using 20% of the data as training data (Fig. 6(d)), we only need less than 5 minutes to train the model. All the experiments are conducted on a normal personal computer containing an *Intel Pentium D* CPU (3.0 GHz, *Dual Core*) and 1 *Giga* byte memory.

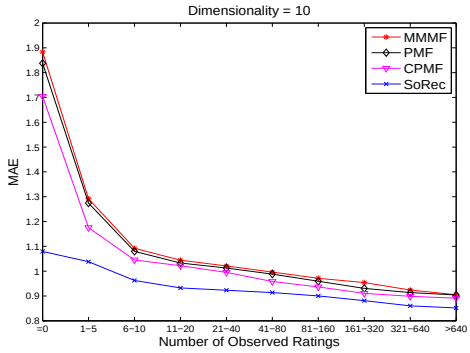
From Fig. 6, we also observe that when using a small value of λ_C, such as $\lambda_C = 0.1$ or $\lambda_C = 1$, after 200 to 300 iterations, the model begins to overfit, while a larger λ_C, such as $\lambda_C = 10$, does not have the overfitting problem. These experiments clearly demonstrate that in this Epinions dataset, using little social network information can cause overfitting problem, and that the predictive accuracy can be improved by incorporating more social network information.



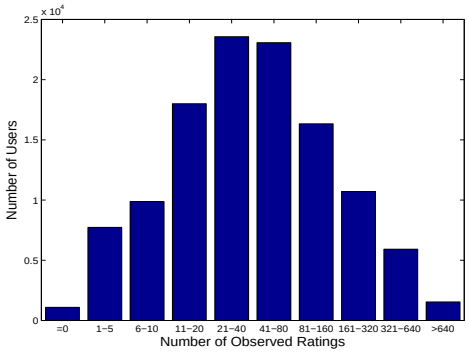
(a) Performance Comparison on Different User Rating Scales (99% as Training Data)



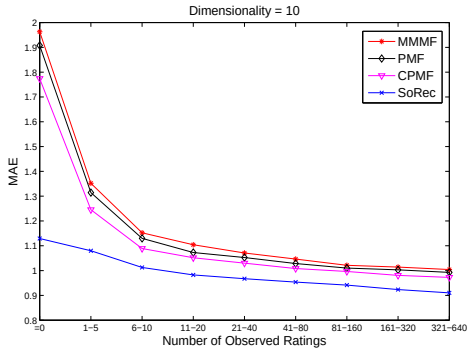
(b) Distribution of Testing Data (99% as Training Data)



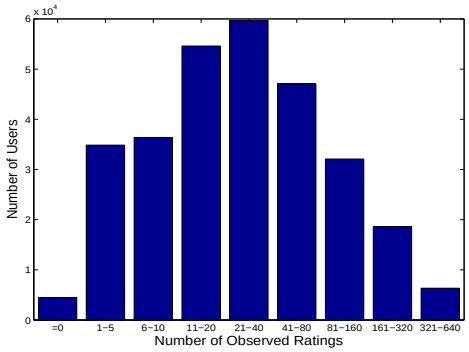
(c) Performance Comparison on Different User Rating Scales (80% as Training Data)



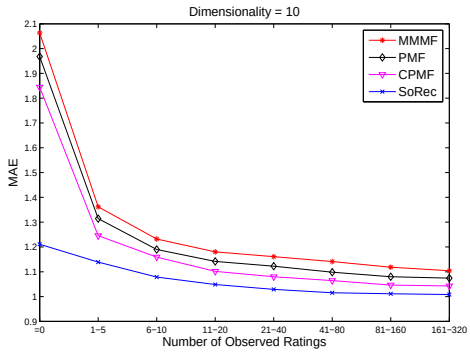
(d) Distribution of Testing Data (80% as Training Data)



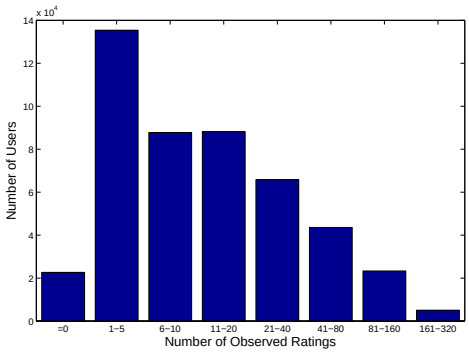
(e) Performance Comparison on Different User Rating Scales (50% as Training Data)



(f) Distribution of Testing Data (50% as Training Data)

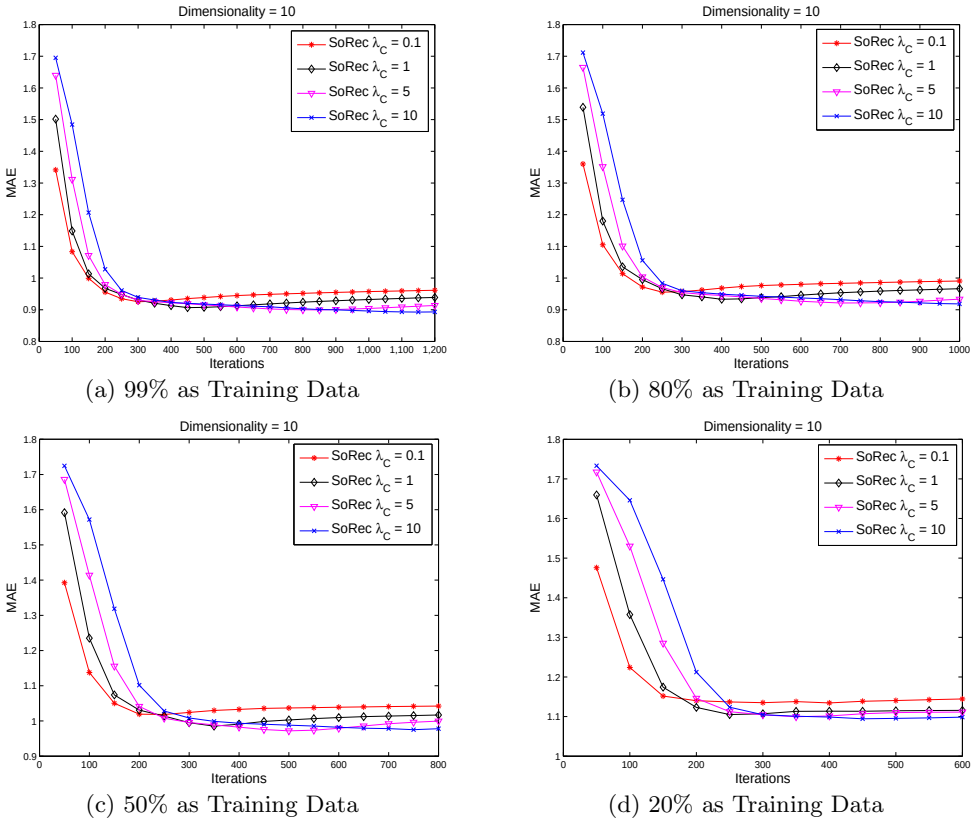


(g) Performance Comparison on Different User Rating Scales (20% as Training Data)



(h) Distribution of Testing Data (20% as Training Data)

Figure 5: Performance Comparison on Different Users



(a) 99% as Training Data

(b) 80% as Training Data

(c) 50% as Training Data

(d) 20% as Training Data

Figure 6: Efficiency Analysis

5. CONCLUSIONS AND FUTURE WORK

In this paper, based on the intuition that a user's social network will affect this user's behaviors on the Web, we present a novel social recommendation framework fusing a user-item rating matrix with the user's social network using probabilistic matrix factorization. The experimental results show that our approach outperforms the other state-of-the-art collaborative filtering algorithms, and the complexity analysis indicates it is scalable to very large datasets. Moreover, the data fusion method using probabilistic matrix factorization we introduce in this paper is not only applicable to social recommendation, but also can be applied to other popular research topics, such as social search and many other tasks in information retrieval and data mining.

In this paper, we employ the inner product of two vectors to fit the observed data; this approach assumes that the observed data is a linear combination of several latent factors. Although we use the logistic function to constrain the inner product, a more natural and accurate extension for this assumption is to use a kernel representation for the two low-dimensional vectors, such as a Gaussian Kernel or a Polynomial Kernel, which map the the relations of two vectors into a nonlinear space, and thus would lead to an increase in the model's performance.

We only use inter-user trust information in this paper, but in many online social networks, the distrust information is also stated by many users. Because a user trust feature space may not be consistent with the corresponding user distrust feature space, we cannot simply incorporate the distrust information into our model. In the future, we need to investigate the following two problems: whether the distrust information is useful to increase the prediction quality, and how to incorporate it.

When fusing the social network information, we ignore the information diffusion or propagation between users. A more accurate approach is to consider the diffusion process between users. Hence, we need to replace the social network matrix factorization with the social network diffusion processes. This consideration will help alleviate the data sparsity problem and will potentially increase the prediction accuracy.

6. ACKNOWLEDGMENTS

The authors appreciate the anonymous reviewers for their extensive and informative comments for the improvement of this paper. The work described in this paper was fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4150/07E and GRF #412507).

7. REFERENCES

[1] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In *IJCAI'07: Proceedings of International Joint Conferences on Artificial Intelligence*, pages 2677–2682, 2007.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI'98: Proceedings of Uncertainty in Artificial Intelligence*, 1998.

[3] J. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, New York, NY, USA, 2002. ACM.

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

[5] D. Dueck and B. Frey. Probabilistic sparse matrix factorization. In *Technical Report PSI TR 2004-023*, Dept. of Computer Science, University of Toronto, 2004.

[6] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.

[7] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM.

[8] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, New York, NY, USA, 2003. ACM.

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

[10] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344, New York, NY, USA, 2004. ACM.

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

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

[13] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46, New York, NY, USA, 2007. ACM.

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

[15] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML '05: Proceedings of the 22th International Conference on Machine Learning*, 2005.

[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM Conference on Computer Supported Cooperative Work*, 1994.

[17] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, 2008.

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

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[20] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, 2003.

[21] P. Singla and M. Richardson. Yes, there is a correlation - from social networks to personal behavior on the web. In *WWW '08: Proceedings of the 17th international conference on World Wide Web*, pages 655–664, New York, NY, USA, 2008. ACM.

[22] R. R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[23] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, pages 720–727, 2003.

[24] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM.

[25] 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 *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121, New York, NY, USA, 2005. ACM.

[26] D. Zhou, B. Scholkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In *Advances in Neural Information Processing Systems*, volume 17, 2005.

[27] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles. Learning multiple graphs for document recommendations. In *WWW '08: Proceedings of the 17th international conference on World Wide Web*, pages 141–150, New York, NY, USA, 2008. ACM.

[28] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 487–494, New York, NY, USA, 2007. ACM.