Learning to Recommend with Explicit and Implicit Social Relations

HAO MA, IRWIN KING, and MICHAEL R. LYU, The Chinese University of Hong Kong

29

Recommender systems have been well studied and developed, both in academia and in industry recently. However, traditional recommender systems assume that all the users are independent and identically distributed; this assumption ignores the connections among users, which is not consistent with the real-world observations where we always turn to our trusted friends for recommendations. Aiming at modeling recommender systems more accurately and realistically, we propose a novel probabilistic factor analysis framework which naturally fuses the users' tastes and their trusted friends' favors together. The proposed framework is quite general, and it can also be applied to pure user-item rating matrix even if we do not have explicit social trust information among users. In this framework, we coin the term *social trust ensemble* to represent the formulation of the social trust restrictions on the recommender systems. 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 show that our method outperforms state-of-the-art approaches.

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

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Recommender systems, social network, social trust ensemble, matrix factorization

ACM Reference Format:
Ma, H., King I., and Lyu, M. R. 20011. learning to recommend with explicit and implicit social relations. ACM Trans. Intell. Syst. Technol. 2, 3, Article 29 (April 2011), 19 pages.
DOI = 10.1145/1961189.1961201 http://doi.acm.org/10.1145/1961189.1961201

1. INTRODUCTION

As there has been exponential growth of information generated on the World Wide Web, *information filtering* techniques like *recommender systems* have become more and more important and popular. Recommender systems form a specific type of information filtering technique that attempts to suggest information items (movies, books, music, news, Web pages, images, etc.) that are likely to interest users. Typically, recommender systems are based on *collaborative filtering*, which is a technique that automatically

---

A short version of this article appears in the *proceedings of The 32th Annual ACM SIGIR Conference (SIGIR'09)* [Ma et al. 2009]. Different from the conference article, the new contents of this article include the following. (1) We extend our method to a more general framework to flexibly incorporate both explicit and implicit social connection information. (2) We reorganize and extend the relate work to include more recently work in the literature. (3) We compare our method with more approaches, like UserMean, ItemMean and NMF. (4) We add some experiments on a new MovieLens dataset to show the performance of our approach on traditional recommender systems.

The work described in this article was fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 4128/08E and CUHK 4154/09E) and a grant from Microsoft Research Asia Research Grant FY09-RES-OPP-103.

Authors' addresses: H. Ma (corresponding author), I. King, M. R. Lyu, The Chinese University of Hong Kong; emails: hma@cse.cuhk.edu.hk.

DOI 10.1145/1961189.1961201 http://doi.acm.org/10.1145/1961189.1961201

predicts the interest of an active user by collecting rating information from other similar users or items.

Although recommender systems have been widely studied in academia and deployed in industry, such as Amazon and eBay, most of these techniques suffer several inherent weaknesses. The first well-known challenge is the data sparsity problem. As reported in Sarwar et al. [2001], the density of the available ratings in commercial recommender systems is often less than 1%. Many collaborative filtering algorithms are impeded by the sparsity problem, hence cannot handle users who have rated few items. Secondly, traditional recommender systems ignore the social connections or trust relations among users. But the fact is, in the real world, we always turn to friends we trust for book, music, or restaurant recommendations, and our favors can easily be affected by the friends we trust. Therefore, traditional recommender systems, which purely mine the user-item rating matrix for recommendations, do not provide realistic output. Recently, trust-aware recommender systems have drawn lots of attention [Massa and Avesani 2004; 2007], but most of these methods are based on some ad hoc heuristics, and they still have the data sparsity and scalability problems. Moreover, the relationship between the user-item matrix and the users' trust network are not fully understood.

In this article, aiming at solving the preceding problems and modeling recommender systems more accurately and realistically, we make three assumptions based on our observations on real-world recommendation processes.

—Users have their own characteristics, and they have different tastes on different items, such as movies, books, music, articles, food, etc.
—Users can be easily influenced by the friends they trust, and prefer their friends' recommendations.
—One user's final decision is the balance between his/her own taste and his/her trusted friends' favors.

Based on the preceding intuitions, we endow a novel understanding to all the ratings in the user-item matrix R. We interpret the rating R_{ij} in the user-item matrix as the representation mixed by both the user u_i's taste and his/her trusted friends tastes on the item v_j. This assumption naturally employs both the user-item matrix and the users' social trust network for the recommendations.

In terms of the users' own tastes, we factorize the user-item matrix and learn two low-dimensional matrices, which are the user-specific latent matrix and item-specific latent matrix. For the social trust graph, based on the intuition that users always prefer items recommended by friends they trust, we infer and formulate the recommendation problem purely based on their trusted friends' favors. Then, by employing a probabilistic framework, we fuse the users and their trusted friends' tastes together by an ensemble parameter. Finally, by performing a simple gradient descent on the objective function, we learn the latent low-dimensional user-specific and item-specific matrices for the prediction of users' favors on different items.

The proposed framework in this article is quite general, and it can also be applied to pure user-item rating matrix even if we do not have explicit social connection information among users. By taking advantage of some well-known similarity functions, like Vector Space Similarity (VSS) and Pearson Correlation Coefficient (PCC), we can calculate the top-k similar users for every single user in the user-item rating matrix. These users can be treated as implicit social connections between users, and the relationships can be plugged in our framework to improve the prediction accuracy of traditional recommender systems.

The experimental results on Epinions and MovieLens datasets show that our method outperforms state-of-the-art collaborative filtering and social trust-based recommendation algorithms, especially when users have very few ratings. 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 article is organized as follows. In Section 2, we provide an overview of several major approaches for recommender systems and other related work. Section 3 presents our work on recommender systems with the social trust ensemble. In Section 4, we illustrate how to extend our work to traditional recommender systems which do not have social network information. The results of an empirical analysis are presented in Section 5, followed by the conclusions and future work in Section 6.

2. RELATED WORK

Recommendation techniques have been widely studied in research communities of information retrieval [Canny 2002; Hofmann 2003; 2004; Jin et al. 2004; Liu and Yang 2008; Yu et al. 2009; Zhang and Koren 2007], machine learning [Rennie and Srebro 2005; Salakhutdinov and Mnih 2008a; 2008b; Si and Jin 2003; Zhu et al. 2009], and data mining [Bell et al. 2007; Koren 2008; 2009]. In this section, we review several major approaches for recommender systems, including: (1) traditional recommender systems which are mainly based on collaborative filtering techniques, and (2) trust-aware recommender systems which have drawn lots of attention recently.

2.1. Traditional Recommender Systems

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.

Among all of these methods, memory-based approaches are the most popular methods and they are widely adopted in commercial collaborative filtering systems [Linden et al. 2003; Resnick et al. 1994]. These methods employ different strategies to find similar users and items for making the predictions, which are known as user-based approaches [Breese et al. 1998; Herlocker et al. 1999; Jin et al. 2004; Ma et al. 2007] and item-based approaches [Deshpande and Karypis 2004; Linden et al. 2003; Sarwar et al. 2001], respectively. To predict a rating R_{ij} of a given item v_j for an active user u_i, user-based methods search for other users similar to the user u_i and utilize their ratings to the item v_j for prediction, while item-based methods leverage the ratings of other items similar to the item v_j from the user u_i instead. In order to take advantage of these two types of methods, Wang et al. [2006] and Ma et al. [2007] proposed two fusion models to combine the user-based method with the item-based method.

In addition to memory-based methods, model-based approaches, which employ statistical and machine learning techniques to learn models from the data, also play an important role in collaborative filtering research. Examples of model-based approaches include the clustering model [Kohrs and Merialdo 1999], aspect models [Hofmann 2003; 2004; Si and Jin 2003], the latent factor model [Canny 2002], the Bayesian hierarchical model [Zhang and Koren 2007], and the ranking model [Liu and Yang 2008]. Kohrs and Merialdo [1999] 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. Hofmann [2003] proposed an algorithm based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables.

Recently, several matrix factorization methods [Rennie and Srebro 2005; Salakhutdinov and Mnih 2008a; 2008b; Srebro and Jaakkola 2003] have been proposed for collaborative filtering. These methods focus on factorizing the user-item rating matrix using low-rank representations, and then utilize them to make further predictions. The motivation behind a low-dimensional factorization model is that there is only a small

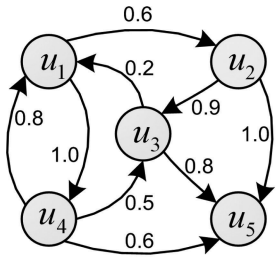
number of factors that are important, and a user's preference vector is determined by how each factor applies to that user.

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 Srebro and Jaakkola [2003]. In Srebro et al. [2004], the authors 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 Salakhutdinov and Mnih [2008b]. In Salakhutdinov and Mnih [2008a], the Gaussian-Wishart priors are placed on the user and item hyperparameters. Although low-dimensional methods are proved very effective and efficient, these methods still suffer several disadvantages that are unveiled. In the SVD method, as well as other well-known methods such as the weighted low-rank approximation method [Srebro and Jaakkola 2003], Probabilistic Principal Component Analysis (PPCA) [Tipping and Bishop 1999], Probabilistic Matrix Factorization (PMF) [Salakhutdinov and Mnih 2008b], and Constrained Probabilistic Matrix Factorization [Salakhutdinov and Mnih 2008b], the latent features are uninterpretable, and there is no range constraint bound on the latent features' vectors. The lack of interpretability results in the improper modeling of the latent factors, hence downgrades the recommendation accuracy. In Zhang et al. [2006], a nonnegative constraint is imposed on both user-specific features U and item-specific features V (nonnegative matrix factorization), but this work is also unable to interpret the physical meanings of the latent factors. Furthermore, low-rank approximation methods also suffer the data sparsity problem. Hence, in this article, we propose a novel matrix factorization method to solve the analyzed problems and remedy the aforementioned deficiencies.

2.2. Trust-Aware Recommender Systems

Recall that all the previous methods for recommender systems are based on the assumption that users are independent and identically distributed, and ignore the social trust relationships between users, which is not consistent with the reality that we normally ask trusted friends for recommendations. Based on this intuition, many researchers have recently started to analyze trust-based recommender systems [Andersen et al. 2008; Bedi et al. 2007; Ma et al. 2008; Massa and Avesani 2004; 2007].

Andersen et al. [2008] developed a set of five natural axioms that a trust-based recommendation system might be expected to satisfy, and then proved that no system can simultaneously satisfy all the axioms. Apparently, this work is out of the scope of this article since we focus on how to employ both the social trust network and user-item matrix to provide more accurate and realistic recommendations. In Massa and Avesani [2004; 2007], the authors studied trust-aware recommender systems. Their work replaces the similarity finding process with the use of a trust metric which is able to propagate trust over the trust network and to estimate a trust weight. The experiments on a large real dataset show that this work increases the coverage (number of ratings that are predictable) while not reducing accuracy (the error of predictions). Bedi et al. [2007] 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 recommendations. Trust-based methods have become a popular research topic recently, however, there are several problems with previous methods. Firstly, these approaches only employ some heuristics to generate recommendations while the relationship between the trust network and the user-item matrix has 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.



(a) Social Trust Graph

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

(b) User-Item Rating Matrix

Fig. 1. Example for trust-based recommendation.

In recent work proposed in Ma et al. [2008], the authors 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 nonsocial collaborative filtering algorithms. However, the disadvantage of this work is that although the users' social trust network is integrated into the recommender systems by factorizing the social trust graph, real, world recommendation processes are not reflected in the model. This drawback not only causes lack of interpretability in the model, but also affects the recommendation qualities. A more novel and realistic approach is needed to model the trust-aware recommendation problem.

3. RECOMMENDATION WITH SOCIAL TRUST ENSEMBLE

Traditional recommender system techniques, like collaborative filtering, only utilize the information of the user-item rating matrix for recommendations while ignoring the social trust relations among users. As online social networks have grown exponentially, incorporating social trust information into recommender systems is becoming more and more important. In this section, we first describe the trust-aware recommendation problem in Section 3.1, and then provide the solution in Sections 3.2, 3.3, and 3.4.

3.1. Problem Description

In the real world, the process of the recommendation scenario includes two central elements: the trust network and the favors of these friends, which can essentially be modeled by the examples of the trust graph in Figure 1(a) and the user-item rating matrix in Figure 1(b), respectively. In the trust graph illustrated in Figure 1(a), totally, 5 users (nodes, from u_1 to u_5) are connected with 9 relations (edges) between users, and each relation is associated with a weight S_{ij} in the range $(0, 1]$ to specify how much user u_i knows or trusts user u_j. Normally, the trust relations in the online trust network are explicitly stated by online users. As illustrated in Figure 1(b), each user also rated some items (from v_1 to v_6) 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 article is how to predict the missing values for the users effectively and efficiently by employing the trust graph and the user-item rating matrix.

3.2. User Features Learning

In order to learn the characteristics or features of the users, we employ matrix factorization to factorize the user-item matrix. The idea of user-item matrix factorization is to derive a high-quality l-dimensional feature representation U of users and V of

items based on analyzing the user-item matrix R. Suppose in a user-item rating matrix, we have m users, n items, 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 article, without loss of generality, we map the ratings 1, ..., R_{max} to the interval [0, 1] using the function $f(x) = x/R_{max}$. Let R_{ij} represent the rating of user u_i for item v_j, and $U \in R^{l\times m}$ and $V \in R^{l\times n}$ be latent user and item feature matrices, with column vectors U_i and V_j representing the l-dimensional user-specific and item-specific latent feature vectors of user u_i and item v_j, respectively. Note that the solutions of U and V are not unique. In Salakhutdinov and Mnih [2008b], the conditional distribution over the observed ratings is defined as:

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

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_{ij}^R is the indicator function that is equal to 1 if user u_i rated item v_j 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 $U_i^T V_j$ within the range [0, 1]. The zero-mean spherical Gaussian priors are also placed on user and item feature vectors.

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

Hence, through a Bayesian inference, we have

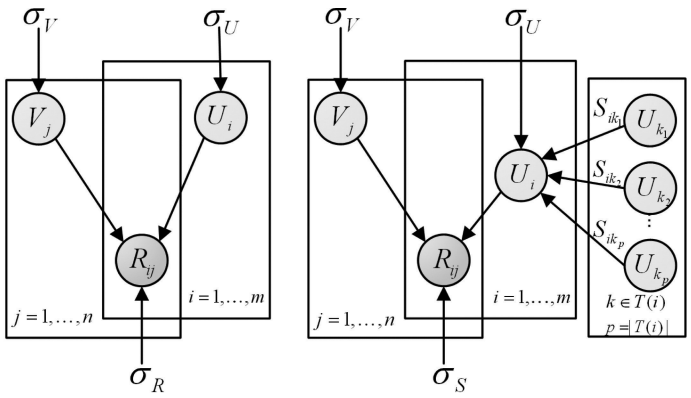
$$\begin{aligned} p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) &\propto p(R|U, V, \sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ &= \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathcal{N}\left(R_{ij}|g(U_i^T V_j), \sigma_R^2\right)\right]^{I_{ij}^R} \\ &\times \prod_{i=1}^{m}\mathcal{N}(U_i|0, \sigma_U^2\mathbf{I}) \times \prod_{j=1}^{n}\mathcal{N}(V_j|0, \sigma_V^2\mathbf{I}). \end{aligned} \tag{3}$$

The graphical model of Eq. (3) is shown in Figure 2(a). This equation represents the method on how to derive the users' latent feature space or users' characteristics purely based on the user-item rating matrix without considering the favors of users' trusted friends. In the next section, we will systematically illustrate how to recommend based on the tastes of trusted friends.

3.3. Recommendations by Trusted Friends

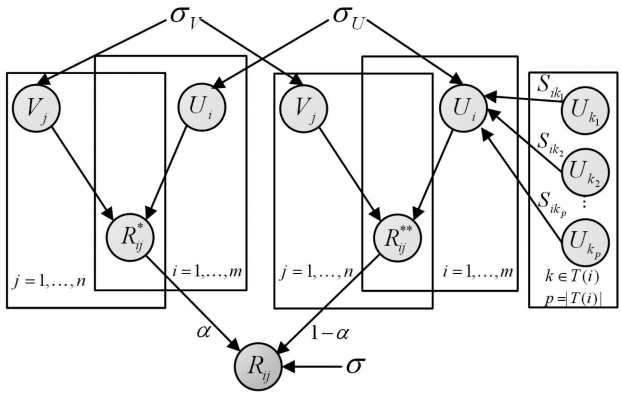
In this section, we analyze how our social trust networks affect our decisions or behaviors, and propose a method to recommend only by using the tastes of trusted friends.

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



(a) factorization of user-item matrix

(b) recommendations by trusted friends



(c) recommendations with social trust ensemble

Fig. 2. Graphical models.

As analyzed in Section 1, we always turn to our friends for recommendations since we trust our friends. We also believe that most probably we will like the items (books, music, movies, etc.) that our trusted friends recommend. Even if the recommended items are not the types we like, we still have a high probability to be influenced by our trusted friends. In the real world, suppose a user wants to see the movie *The Dark Knight* (suppose it is the item v_1 in Figure 1(b)), which is now playing at the theaters, but he/she knows nothing about the movie, like user u_1 in Figure 1(b). What this user, normally do is to take into account his/her trusted friends' recommendations. Among all of his/her trusted friends in Figure 1(a), u_2 and u_4 rated this movie as 4 and 5, and u_1 trusts u_4 (weight 1.0) more than u_2 (weight 0.6). Based on the information, there is a very high probability that u_1 will draw the conclusion that *The Dark Knight* is a very good movie worth watching.

From the preceding analysis, we can generalize the preceding social process as

$$\widehat{R}_{ik} = \frac{\sum\limits_{j \in \mathcal{T}(i)} R_{jk} S_{ij}}{|\mathcal{T}(i)|}, \tag{4}$$

where $\widehat{R}_{ik}$ is the prediction of the rating that user u_i would give item v_j, R_{jk} is the score that user u_j gave item v_k, $\mathcal{T}(i)$ is the friends set that user u_i trusts, and $|\mathcal{T}(i)|$ is the number of trusted friends of user u_i in the set $\mathcal{T}(i)$. $|\mathcal{T}(i)|$ can be merged into S_{ij} since it is the normalization term of trust scores. Hence, Eq. (4) can be simplified as

$$\widehat{R}_{ik} = \sum_{j \in \mathcal{T}(i)} R_{jk} S_{ij}. \tag{5}$$

Then the prediction of the ratings that user u_i gives to all the items can be inferred as

$$\begin{pmatrix} \widehat{R}_{i1} \\ \widehat{R}_{i2} \\ \dots \\ \widehat{R}_{in} \end{pmatrix} = \begin{pmatrix} R_{11} & R_{21} & \dots & R_{m1} \\ R_{12} & R_{22} & \dots & R_{m2} \\ \dots & \dots & \dots & \dots \\ R_{1n} & R_{2n} & \dots & R_{mn} \end{pmatrix} \begin{pmatrix} S_{i1} \\ S_{i2} \\ \dots \\ S_{im} \end{pmatrix}. \tag{6}$$

We can then infer that for all the users to obtain

$$\widehat{R} = SR, \tag{7}$$

where SR can be interpreted as the recommendations purely based on trusted friends' tastes.

From the social trust network aspect, we define the conditional distribution over the observed ratings as

$$p(R|S, U, V, \sigma_R^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} \left[\mathcal{N}\left(R_{ij} | g\left(\sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j \right), \sigma_S^2 \right) \right]^{I_{ij}^R}, \tag{8}$$

where S_{ik} is normalized by $|\mathcal{T}(i)|$, which is the number of trusted friends of user u_i in the set $\mathcal{T}(i)$. I_{ij}^R is the indicator function that is equal to 1 if user i rated item j and equal to 0 otherwise.

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

$$p(U, V|R, S, \sigma_S^2, \sigma_U^2, \sigma_V^2) \propto p(R|S, U, V, \sigma_S^2) p(U|S, \sigma_U^2) p(V|S, \sigma_V^2). \tag{9}$$

In Eq. (9), we can assume that S is independent with the low-dimensional matrices U and V, then this equation can be changed to

$$\begin{aligned} p(U, V|R, S, \sigma_S^2, \sigma_U^2, \sigma_V^2) \;\propto\; & p(R|S, U, V, \sigma_S^2) p(U|\sigma_U^2) p(V|\sigma_V^2), \\ = & \prod_{i=1}^{m} \prod_{j=1}^{n} \left[\mathcal{N}\left(R_{ij} | g\left(\sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j \right), \sigma_S^2 \right) \right]^{I_{ij}^R} \\ & \times \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \times \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}), \end{aligned} \tag{10}$$

where $p(U|\sigma_U^2)$ and $p(V|\sigma_V^2)$ are zero-mean spherical Gaussian priors on user and item feature vectors. This equation specifies the method to recommend purely based on users' trusted friends' tastes. The graphical model is shown in Figure 2(b).

3.4. Social Trust Ensemble

In Section 3.2, given the user-item rating matrix, the observed rating R_{ij} is interpreted by the user u_i's favor on item v_j, while in Section 3.3, given the user-item rating matrix and users' social trust network, the observed rating R_{ij} is realized as the favors on

item v_j of user u_i's trusted friends. Actually, both of the preceding assumptions are partially right since in the real-world situation, every user has his/her own taste and at the same time, every user may be influenced by his/her friends he/she trusts. Hence, in order to define the model more realistically, every observed rating in the user-item matrix should reflect both of these two factors. Based on this motivation, we model the conditional distribution over the observed ratings as

$$\begin{aligned} & p(U, V|R, S, \sigma^2, \sigma_U^2, \sigma_V^2) \\ & = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathcal{N}\left(R_{ij}|g\left(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j\right), \sigma^2\right)\right]^{I_{ij}^R} \\ & \quad \times \prod_{i=1}^{m}\mathcal{N}(U_i|0, \sigma_U^2\mathbf{I}) \times \prod_{j=1}^{n}\mathcal{N}(V_j|0, \sigma_V^2\mathbf{I}). \end{aligned} \tag{11}$$

In Eq. (11), the users' favors and the trusted friends' favors are smoothed by the parameter α, which naturally fuses an appropriate amount of real-world recommendation processes into the recommender systems. The parameter α controls how much users trust themselves or their trusted friends. The graphical model of our method is shown in Figure 3(c).

The log of the posterior distribution for the recommendations is given by

$$\begin{aligned} & \ln p(U, V|R, S, \sigma^2, \sigma_U^2, \sigma_V^2) \\ & = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\left(R_{ij} - g\left(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j\right)\right)^2 \\ & \quad -\frac{1}{2\sigma_U^2}\sum_{i=1}^{m} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{n} V_j^T V_j \\ & \quad -\frac{1}{2}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\right)\ln\sigma^2 - \frac{1}{2}(ml\ln\sigma_U^2 + nl\ln\sigma_V^2) + \mathcal{C}, \end{aligned} \tag{12}$$

where $\mathcal{C}$ is a constant that does not depend on the parameters. Maximizing the log-posterior over two 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

$$\begin{aligned} \mathcal{L}(R, S, U, V) = & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\left(R_{ij} - g\left(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j\right)\right)^2 \\ & + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2, \end{aligned} \tag{13}$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$, and $\|\cdot\|_F^2$ denotes the Frobenius norm.

A local minimum of the objective function given by Eq. (13) can be found by performing gradient descent in U_i, V_j,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial U_i} &= \alpha \sum_{j=1}^{n} I_{ij}^R g' \left(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j \right) V_j \\
&\quad \times \left(g(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j) - R_{ij} \right) \\
&\quad + (1-\alpha) \sum_{p \in \mathcal{B}(i)} \sum_{j=1}^{n} I_{pj}^R g' \left(\alpha U_p^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(p)} S_{pk} U_k^T V_j \right) \\
&\quad \times \left(g \left(\alpha U_p^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(p)} S_{pk} U_k^T V_j \right) - R_{pj} \right) S_{pi} V_j + \lambda_U U_i, \\
\frac{\partial \mathcal{L}}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}^R g' \left(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j \right) \\
&\quad \times \left(g \left(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j \right) - R_{ij} \right) \\
&\quad \times \left(\alpha U_i + (1-\alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T \right) + \lambda_V V_j, \qquad (14)
\end{aligned}$$

where $g'(x)$ is the derivative of logistic function $g'(x) = \exp(x)/(1+\exp(x))^2$ and $\mathcal{B}(i)$ is the set that includes all the users who trust user u_i. In order to reduce the model complexity, in all of the experiments we conduct in Section 5, we set $\lambda_U = \lambda_V$.

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 S, the computational complexity of evaluating the object function $\mathcal{L}$ is $O(\rho_R l + \rho_R \overline{k} l)$, where ρ_R is the number of nonzero entries in the matrix R, and $\overline{k}$ is the average number of friends that a user trusts. Since almost all of the online social networks fit the power-law distribution, a large long tail of users only have few trusted friends. This indicates that the value of $\overline{k}$ is relatively small. The computational complexities for the gradients $\frac{\partial \mathcal{L}}{\partial U}$ and $\frac{\partial \mathcal{L}}{\partial V}$ in Eq. (14) are $O(\rho_R \overline{p}\, l + \rho_R \overline{p}\, \overline{k} l)$ and $O(\rho_R l + \rho_R \overline{k} l)$, respectively, where $\overline{p}$ is the average number of friends who trust a user, which is also a small value. Actually, in a social trust graph, the value of $\overline{k}$ is always equal to the value of $\overline{p}$, which is 9.91 in the dataset we employ in Section 5. Therefore, the total computational complexity in one iteration is $O(\rho_R \overline{p}\, l + \rho_R \overline{p}\, \overline{k} l)$, which indicates that theoretically, the computational time of our method is linear with respect to the number of observations in the user-item matrix R. This complexity analysis shows that our proposed approach is very efficient and can scale to very large datasets.

4. RECOMMENDATION WITH IMPLICIT RELATIONS

In Section 3, we illustrate how to improve recommender systems when both the user-item rating matrix and users' social graph are available. However, the users' social graph is not always available in recommender systems. Actually, our proposed framework is quite general and by taking advantage of some similarity calculation functions, we can find some "pseudo" trusted friends for every user.

4.1. Similarity Function

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) [Breese et al. 1998]. VSS is employed to define the similarity between two users i and f based on the items they rated in common. We have

$$Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} R_{ij} \cdot R_{fj}}{\sqrt{\sum_{j \in I(i) \cap I(f)} R_{ij}^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} R_{fj}^2}}, \tag{15}$$

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 previous 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

$$Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R}_i) \cdot (R_{fj} - \overline{R}_f)}{\sqrt{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R}_i)^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} (R_{fj} - \overline{R}_f)^2}}, \tag{16}$$

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 $[0, 1]$.

4.2. Implicit Friends

With the definition of similarity function, for every user in the user-item rating matrix, we can calculate the top-k similar users as implicit trusted friends. Then, we can use Eq. (13) to learn the latent features U and V. The weight S_{ik} in Eq. (13) can be defined as

$$S_{ik} = \frac{Sim(i, k)}{\sum_{f \in \mathcal{T}(i)} Sim(i, f)}. \tag{17}$$

5. EMPIRICAL ANALYSIS

In this section, we conduct several experiments to compare the recommendation qualities of our Recommendation with Social Trust Ensemble (RSTE) approach with other

state-of-the-art collaborative filtering and trust-aware recommendation methods. Our experiments are intended to address the following questions.

(1) How does our approach compare with published state-of-the-art collaborative filtering and trust-aware recommendation algorithms?
(2) How does the model parameter α 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 few observed rating records?
(5) Is our algorithm efficient when training the model?

5.1. Metrics

We use two 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 metrics MAE is defined as

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r}_{i,j}|}{N}, \tag{18}$$

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 a method, and N denotes the number of tested ratings. The metrics RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{i,j}(r_{i,j} - \widehat{r}_{i,j})^2}{N}}. \tag{19}$$

5.2. Experiments with Explicit Social Information

5.2.1. Dataset Description. We choose Epinions as the data source for our experiments on recommendation with explicit social trust information. Epinions.com is a well-known knowledge sharing site and review site which 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 about to decide whether a product is worth buying or a movie is worth watching. Every member of Epinions maintains a "trust" list which presents a social network of trust relationships between users. Epinions is thus an ideal source for experiments on social trust recommendation.

The dataset used in our experiments is collected by crawling the Epinions.com site on January 2009. It consists of 51,670 users who have rated a total of 83,509 different items. The total number of ratings is 631,064. The density of the user-item rating matrix is less than 0.015%. We can observe that the user-item rating matrix of Epinions is very sparse, since the densities for the two most famous collaborative filtering datasets Movielens (6,040 users, 3,900 movies, and 1,000,209 ratings) and Eachmovie (74,424 users, 1,648 movies, and 2,811,983 ratings) are 4.25% and 2.29%, respectively. Moreover, an important factor for why we choose the Epinions dataset is that user social trust network information is not included in the Movielens and Eachmovie datasets. The statistics of the Epinions user-item rating matrix is summarized in Table I. As to the user social trust network, the total number of issued trust statements is 511,799. The statistics of this data source is summarized in Table II.

Table I. Statistics of User-Item Rating Matrix of Epinions

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

Table II. Statistics of Social Trust Network of Epinions

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

Table III. Performance Comparisons (A Smaller MAE or RMSE Value Means a Better Performance)

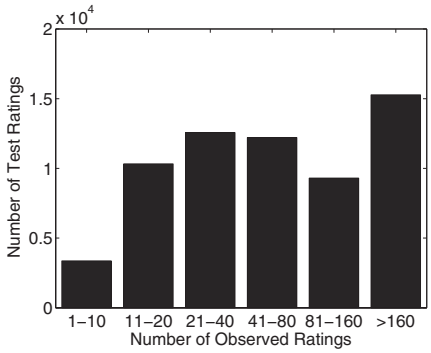
Training Data	Metrics	Dimensionality = 5							
		UserMean	ItemMean	NMF	PMF	TCF	Trust	SoRec	RSTE
90%	MAE	0.9134	0.9768	0.8738	0.8676	0.9005	0.9054	0.8442	**0.8377**
	RMSE	1.1688	1.2375	1.1649	1.1575	1.1697	1.1959	1.1333	**1.1109**
80%	MAE	0.9285	0.9913	0.8975	0.8951	0.9044	0.9221	0.8638	**0.8594**
	RMSE	1.1817	1.2584	1.1861	1.1826	1.1761	1.2140	1.1530	**1.1346**
Training Data	Metrics	Dimensionality = 10							
		UserMean	ItemMean	NMF	PMF	TCF	Trust	SoRec	RSTE
90%	MAE	0.9134	0.9768	0.8712	0.8651	0.9005	0.9039	0.8404	**0.8367**
	RMSE	1.1688	1.2375	1.1621	1.1544	1.1697	1.1917	1.1293	**1.1094**
80%	MAE	0.9285	0.9913	0.8951	0.8886	0.9044	0.9215	0.8580	**0.8537**
	RMSE	1.1817	1.2584	1.1832	1.1760	1.1761	1.2132	1.1492	**1.1256**

5.2.2. Comparison. In this section, in order to show the performance improvement of our RSTE approach, we compare our method with the following approaches.

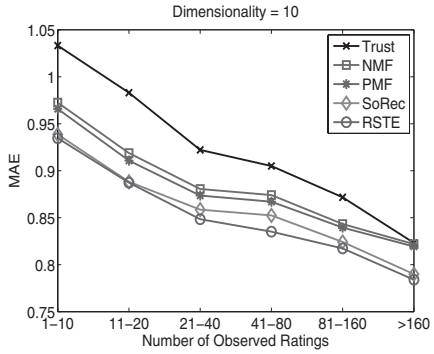
(1) *UserMean*. This method uses the mean value of every user to predict the missing values.
(2) *ItemMean*. This method utilizes the mean value of every item to predict the missing values.
(3) *NMF*. This method is originally proposed in Lee and Seung [1999] for image analysis. However, it is widely used in collaborative filtering recently. It only uses a user-item matrix for recommendations.
(4) *PMF*. This method is proposed by Salakhutdinov and Minh [2008b]. It only uses the user-item matrix for the recommendations, and it is based on probabilistic matrix factorization.
(5) *TCF*. This method is a well-known trust-aware collaborative filtering method proposed in Massa and Avesani [2004].
(6) *Trust*. This is the method purely using trusted friends' tastes in making recommendations. It is proposed in Section 3.3 in this article. It is also a special case of RSTE when $\alpha = 0$.
(7) *SoRec*. This is the method proposed in Ma et al. [2008]. It is a social trust-aware recommendation method that factorizes the user-item rating matrix and users' social trust network by sharing the same user latent space.

We use different amounts of training data (90%, 80%) to test the algorithms. Training data 90%, for example, means we randomly select 90% of the ratings from the Epinions dataset as the training data to predict the remaining 10% of ratings. Random selection was carried out 5 times independently. The experimental results using 5 and 10 dimensions to represent the latent features are shown in Table III.

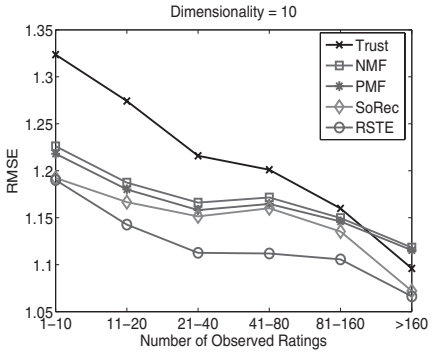
The parameter settings of our approach are $\alpha = 0.4$ for both 90% training data and 80% training data, $\lambda_U = \lambda_V = 0.001$, and in all the experiments conducted in the following sections, we set all of the parameters λ_U, λ_V equal to 0.001. From Table III, we can



(a) distribution of testing data (90% as training data)



(b) MAE comparison on different user rating scales (90% as training data)



(c) RMSE comparison on different user rating scales (90% as training data)

Fig. 3. Performance comparison on different users.

observe that our approach RSTE outperforms the other methods. In general, two social trust recommendation approaches SoRec and RSTE all perform better than the NMF and PMF methods (only using the user-item matrix for recommendations). However, the trust method performs worse than the NMF and PMF method, which indicates purely utilizing trusted friends' tastes to recommend is not applicable. Among these three trust-aware recommendation methods, our RSTE method generally achieves better performance than the SoRec and trust methods on both MAE and RMSE. This demonstrates that our interpretation on the formation of the ratings is realistic and reasonable.

5.2.3. Performance on Different Users. One challenge of the recommender systems is that it is difficult to recommend items to users who have very few ratings. Hence, in order to compare our approach with the other methods thoroughly, we first group all 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 Figure 3. Users are grouped into 6 classes: "$1-10$", "$11-20$", "$21-40$", "$41-80$", "$81-160$", and ">160", denoting how many ratings users have rated.

Figure 3(a) summarizes the distributions of testing data according to groups in the training data (90% as training data). For example, there are a total 3,360 user-item pairs to be predicted in the testing dataset in which the related users in the training

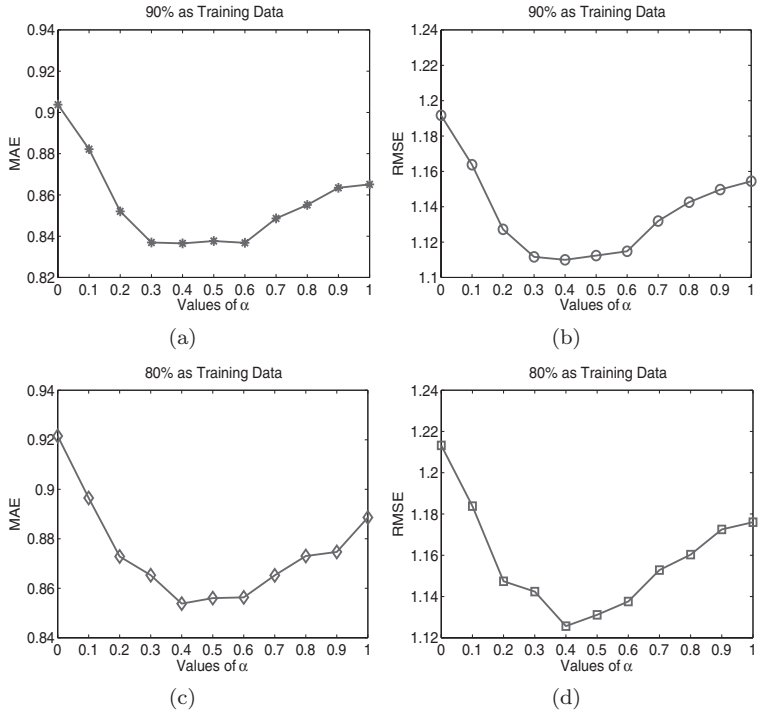


Fig. 4. Impact of parameter α (dimensionality = 10).

dataset have rating numbers from 1 to 10. In Figure 3(b) and Figure 3(c), we observe that our RSTE algorithm consistently performs better than other methods, especially when few user ratings are given. When users' rating records are ranging from 1 to 80, our RSTE method performs much better than the trust, PMF and SoRec approaches.

5.2.4. Impact of Parameter α. In our method proposed in this article, the parameter α balances the information from the users' own characteristics and their friends' favors. It controls how much our method should trust users themselves and their friends. If $\alpha = 1$, we only mine the user-item rating matrix for matrix factorization, and simply employ users' own tastes in making recommendations. If $\alpha = 0$, we only extract information from the social trust graph to predict users' preferences purely from the friends they trust. In other cases, we fuse information from the user-item rating matrix and the user social trust network for probabilistic matrix factorization and, furthermore, to predict ratings for the users.

Figure 4 shows the impacts of α on MAE and RMSE. We observe that the value of α impacts the recommendation results significantly, which demonstrates that fusing the users' own tastes with their friends' favors greatly improves the recommendation accuracy. No matter whether using 90% training data or 80% training data, as α increases, the MAE and RMSE decrease (prediction accuracy increases) at first, but when α surpasses a certain threshold, the MAE and RMSE increase (prediction accuracy decreases) with further increase of the value of α. This phenomenon confirms the intuition that purely using the user-item rating matrix or purely using the users' social trust network for recommendations cannot generate better performance than fusing these two factors together.

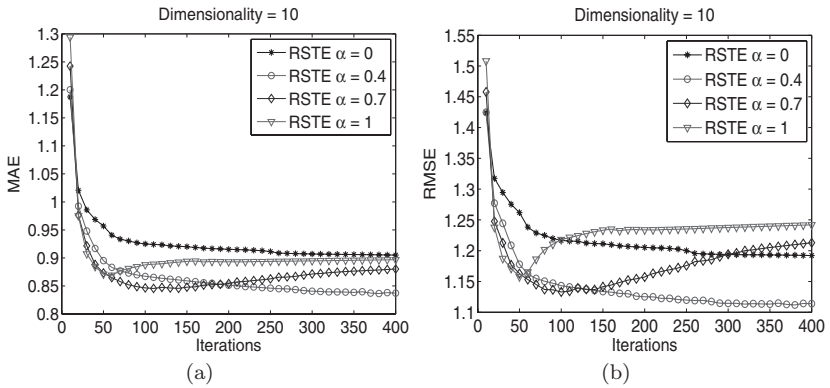


Fig. 5. Efficiency analysis (90% as training data).

From Figure 4(a) and Figure 4(b), when using 90% ratings as training data, we observe that our RSTE method achieves the best performance when α is around 0.4, while smaller values like $\alpha = 0.1$ or larger values like $\alpha = 0.7$ can potentially degrade the model performance. This indicates that we need to trust more about the tastes of users' trusted friends than users' own tastes, since the training data of the user-item matrix is very sparse, which can hardly learn accurate characteristics of the users. In Figure 4(c) and Figure 4(c), when using 80% ratings as training data, the optimal value of α is also around 0.4. However, less ratings for users will lead to an overall degradation of the recommendation results.

5.2.5. Training 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 shows 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 90% of the data as training data, our method needs less than 400 iterations for training, and each iteration only requires less than 20 seconds. All the experiments are conducted on a normal personal computer containing an Intel Pentium D CPU (3.0 GHz, Dual Core) and 1G memory.

Figure 5(a) and Figure 5(b) show the performance (MAE and RMSE) changes with the iterations. We observe that when using a large value of α, such as $\alpha = 1$ or $\alpha = 0.7$, at the end of the training, the model begins to overfit (especially for the RMSE), while a relatively smaller α, such as $\alpha = 0$ or $\alpha = 0.4$, does not have the overfitting problem. These experiments clearly demonstrate that in this dataset, an approach ignoring the social trust information can cause the overfitting problem, and that predictive accuracy can be improved by incorporating an appropriate amount of social trust information.

5.3. Experiments with Implicit Social Information

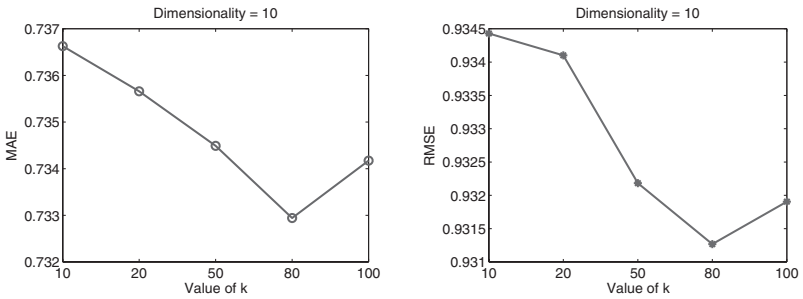
In this section, we illustrate the performance of our framework on traditional user-item rating matrix.

5.3.1. Dataset Description. We evaluate our algorithms on the MovieLens dataset, which is commonly used in previous work. The MovieLens dataset contains 943 users, 1,682 movies, and 100,000 ratings in the scale of one to five. We map the ratings 1,2,3,4, and 5 to the interval $[0, 1]$ using the linear function $t(x) = x/5$.

5.3.2. Comparison. We compare our method with UserMean, ItemMean, NMF, and PMF. The experimental results are shown in Table IV. In our RSTE method, we set

Table IV. Performance Comparisons on MovieLens Dataset (A Smaller MAE or RMSE Value Means a Better Performance)

Training Data	Metrics	Dimensionality = 10				
		UserMean	ItemMean	NMF	PMF	RSTE
80%	MAE	0.839	0.827	0.770	0.745	**0.732**
	RMSE	1.047	1.036	0.981	0.942	**0.931**



(a) (b)

Fig. 6. Impact of parameter k.

$\alpha = 0.3$ and $k = 80$, which means we choose 80 most similar users for every user. We can observe that our method performs better than other methods, which shows that our method not only works well in trust-aware recommender systems, but can also generate very good performance in traditional recommender systems.

5.3.3. Impact of Parameter k. Basically, the parameter α on the MovieLens dataset shares a similar trend with the Epinions dataset. Hence, we do not analyze the details in this section. Instead, we will show the impact of parameter k. Parameter k is introduced to specify how many similar users should be selected for every user. These similar users perform as implicit social friends for targeted users. A small value of k probably is not enough for representing friends' tastes while a large value of k will introduce some noise (some computed friends are no longer similar with the targeted user). From Figure 6, we can see that the best k value is around 80, and larger or smaller k values will hurt the recommendation performance, which confirms our intuition.

6. CONCLUSIONS AND FUTURE WORK

This article is motivated by the fact that a user's trusted friends on the Web will affect this user's online behavior. Based on the intuition that every user's decisions on the Web should include both the user's characteristics and the user's trusted friends' recommendations, we propose a novel, effective, and efficient probabilistic matrix factorization framework for recommender systems. Experimental analysis on the Epinions and MovieLens datasets shows the promising future of our proposed method. Moreover, the method introduced in this article, by using probabilistic matrix factorization, is not only working in trust-aware recommender systems, but also applicable to traditional recommender systems.

In this article, although we employ trusted friends' opinions in the social trust network to make recommendations for users, we do not consider the possible diffusions of trust between various users. Under the circumstance that both the user-item rating matrix and the trust relations of a social network are very sparse, the diffusions of trust relations become inevitable, since this consideration will help to alleviate the data sparsity problem and will potentially increase prediction accuracy. We plan to employ the diffusion processes in our future work.

In many popular applications on the Web, users not only can keep a list of trust relationships, but also have the rights to establish a list of distrust or block relationships. If a user u_j is in the distrust list of a user u_i, most probably, it is because user u_i thinks user u_j's taste is totally different from him/her. Actually, this information is very useful on recommender systems. Unfortunately, to the best of our knowledge, no previous work can employ this information well in recommender systems. The understanding of distrust relations is still unclear to researchers: We cannot use diffusion methods to model it due to the reason that one person's enemy's enemy is not necessarily the enemy of this person. In the future, we plan to study the formation and nature of distrust relations, and explicitly model them in recommender systems.

As the exponential growth of online social network sites continues, the research of social search is becoming more and more important. We also plan to develop similar techniques to allow users' trusted friends to influence the users' search results or query suggestions. The intuition behind this is that if a large number of our friends are searching for something, it's likely that we may be interested in that topic too. This would be an interesting search phenomenon to explore in social networks.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers and editor for their helpful comments.

REFERENCES

ANDERSEN, R., BORGS, C., CHAYES, J., FEIGE, U., FLAXMAN, A., KALAI, A., MIRROKNI, V., and TENNENHOLTZ, M. 2008. Trust-Based recommendation systems: An axiomatic approach. *In Proceedings of the International World Wide Web Conference (WWW'08).* 199–208.

BEDI, P., KAUR, H., and MARWAHA, S. 2007. Trust based recommender system for semantic web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'07).* 2677–2682.

BELL, R., KOREN, Y., and VOLINSKY, C. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the International SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'07).* 95–104.

BREESE, J. S., HECKERMAN, D., and KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI'98).*

CANNY, J. 2002. Collaborative filtering with privacy via factor analysis. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02).* 238–245.

DESHPANDE, M. and KARYPIS, G. 2004. Item-Based top-n recommendation. *ACM Trans. Inf. Syst. 22,* 1, 143–177.

HERLOCKER, J. L., KONSTAN, J. A., BORCHERS, A., and RIEDL, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99).* 230–237.

HOFMANN, T. 2003. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03).* 259–266.

HOFMANN, T. 2004. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. 22*, 1, 89–115.

JIN, R., CHAI, J. Y., and SI, L. 2004. An automatic weighting scheme for collaborative filtering. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04).* 337–344.

KOHRS, A. and MERIALDO, B. 1999. Clustering for collaborative filtering applications. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'99).*

KOREN, Y. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the International SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'08).* 426–434.

KOREN, Y. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the International SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09).* 447–456.

LEE, D. D. and SEUNG, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, 6755, 788–791.

LINDEN, G., SMITH, B., and YORK, J. 2003. Amazon.com recommendations: Item-to-Item collaborative filtering. *IEEE Internet Comput. 7,* 1, 76–80.

LIU, N. N. and YANG, Q. 2008. Eigenrank: A ranking-oriented approach to collaborative filtering. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08).* 83–90.

MA, H., KING, I., and LYU, M. R. 2007. Effective missing data prediction for collaborative filtering. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07).* 39–46.

MA, H., KING, I., and LYU, M. R. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09).* 203–210.

MA, H., YANG, H., LYU, M. R., and KING, I. 2008. SoRec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM'08).* 931–940.

MASSA, P. and AVESANI, P. 2004. Trust-Aware collaborative filtering for recommender systems. In *Proceedings of the CoopIS/DOA/ODBASE Conference*. 492–508.

MASSA, P. and AVESANI, P. 2007. Trust-Aware recommender systems. In *Proceedings of the ACM International Conference on Recommender Systems (RecSys'07).* 17–24.

RENNIE, J. D. M. and SREBRO, N. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine Learning (ICML'05).*

RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., and RIEDL, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94).*

SALAKHUTDINOV, R. and MNIH, A. 2008a. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the International Conference on Machine Learning (ICML'08).*

SALAKHUTDINOV, R. and MNIH, A. 2008b. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems.* Vol. 20.

SARWAR, B., KARYPIS, G., KONSTAN, J., and REIDL, J. 2001. Item-Based collaborative filtering recommendation algorithms. In *Proceedings of the International World Wide Web Conference (WWW'01).* 285–295.

SI, L. and JIN, R. 2003. Flexible mixture model for collaborative filtering. In *Proceedings of the International Conference on Machine Learning (ICML'03).*

SREBRO, N. and JAAKKOLA, T. 2003. Weighted low-rank approximations. In *Proceedings of the International Conference on Machine Learning (ICML'03).*

SREBRO, N., RENNIE, J. D. M., and JAAKKOLA, T. 2004. Maximum-Margin matrix factorization. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'04).*

TIPPING, M. E. and BISHOP, C. M. 1999. Probabilistic principal component analysis. *J. Roy. Statist. Soc. B61,* 3, 611–622.

WANG, J., DE VRIES, A. P., and REINDERS, M. J. T. 2006. Unifying user-based and item- based collaborative filtering approaches by similarity fusion. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06).* 501–508.

YU, K., ZHU, S., LAFFERTY, J., and GONG, Y. 2009. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09).* 211–218.

ZHANG, S., WANG, W., FORD, J., and MAKEDON, F. 2006. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the SIAM International Conference on Data Mining (SDM'06).*

ZHANG, Y. and KOREN, J. 2007. Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07).* 47–54.

ZHU, S., YU, K., and GONG, Y. 2009. Stochastic relational models for large-scale dyadic data using mcmc. In *Advances in Neural Information Processing Systems*. Vol. 21.

Received March 2010; revised July 2010; accepted September 2010