Learning to Recommend with Social Trust Ensemble

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

ABSTRACT

As an indispensable technique in the field of *Information Filtering*, *Recommender System* has been well studied and developed both in academia and in industry recently. However, most of current recommender systems suffer the following problems: (1) The large-scale and sparse data of the user-item matrix seriously affect the recommendation quality. As a result, most of the recommender systems cannot easily deal with users who have made very few ratings. (2) The 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 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. 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 performs better than the state-of-the-art approaches.

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

General Terms: Algorithm, Experimentation

Keywords: Recommender Systems, Social Network, Social Trust Ensemble, Matrix Factorization

1. INTRODUCTION

As the exponential growth of information generated on the World Wide Web, the *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 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.

Although recommender systems have been widely studied in the academia and deployed in the 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 [20], 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 [14, 15], 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 paper, aiming at solving the above problems and modeling the recommender systems more accurately and realistically, we make three assumptions based on our observations on the 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 above 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

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...$10.00.

are user-specific latent matrix and item-specific latent matrix. For the social trust graph, based on the intuition that users always prefer the items recommended by the 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 experimental results on a large Epinions[1] dataset show that our method outperforms the state-of-the-art collaborative filtering and social trust-based recommendation algorithms, especially when the 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 paper 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 system with social trust ensemble. 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, including (1) traditional recommender systems which are mainly based on collaborative filtering techniques, and (2) social trust-based recommender systems which have drawn lots of attention recently.

Traditional collaborative filtering algorithms mainly focus on the user-item matrix. Among all of these methods, the memory-based approaches are the most popular methods and they are widely adopted in commercial collaborative filtering systems [10, 17]. These methods employ different strategies to find similar users and items for making the predictions, which are known as user-based approaches [3, 6, 9, 12] and item-based approaches [5, 10, 20], 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 advantages of these two types of methods, Wang et al. in [23] and Ma et al. in [12] proposed two fusion models to combine user-based method with item-based method. In addition to the 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 the collaborative filtering research. Examples of model-based approaches include aspect models [7, 8, 21], the latent factor model [4], the Bayesian hierarchical model [24] and the ranking model [11]. Recently, several matrix factorization methods [16, 18, 19, 22] 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.

Recall that all the above methods for recommender systems are based on the assumption that users are independent and identically distributed, and ignores 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 [1, 2, 13, 14, 15].

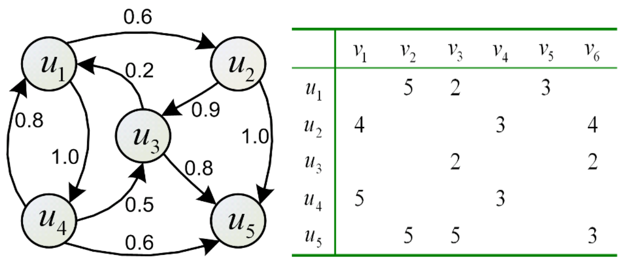
Andersen et al. in [1] developed a set of five natural axioms that a trust-based recommendation system might be expected to satisfy, and then proved that no system can simultaneously satisfy all the axioms. Apparently, this work is out of the scope of this paper since we focus on how to employ both social trust network and user-item matrix to provide more accurate and realistic recommendations. In [14, 15], Massa and Avesani studied the 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 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 [2] proposed a trust-based recommender system for the Semantic Web; this system runs on a server with the knowledge distributed over the network in the form of ontologies, and uses the Web of trust to generate the recommendations. The 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.

In recent work proposed in [13], Ma et al. developed a factor analysis method based on the probabilistic graphical model which fuses the user-item matrix with the users' social trust networks by sharing a common latent low-dimensional user feature matrix. The experimental analysis shows that this method generates better recommendations than the non-social 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, the 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 ignore the social trust relations among users. As the exponential growth of online social networks, incorporating social trust information into recommender systems is becoming more and more important. In this section, we first describe the trust-aware

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



	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

(a) Social Trust Graph (b) User-Item Rating Matrix

Figure 1: Example for Trust based Recommendation

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 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 Fig. 1(a) and the user-item rating matrix in Fig. 1(b), respectively. In the trust graph illustrated in Fig. 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 Fig. 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 paper 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 paper, 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 [19], the conditional distribution over the observed ratings is defined as:

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

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_{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:

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

Hence, through a Bayesian inference, we have

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

The graphical model of Eq. (3) is shown in Fig. 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.

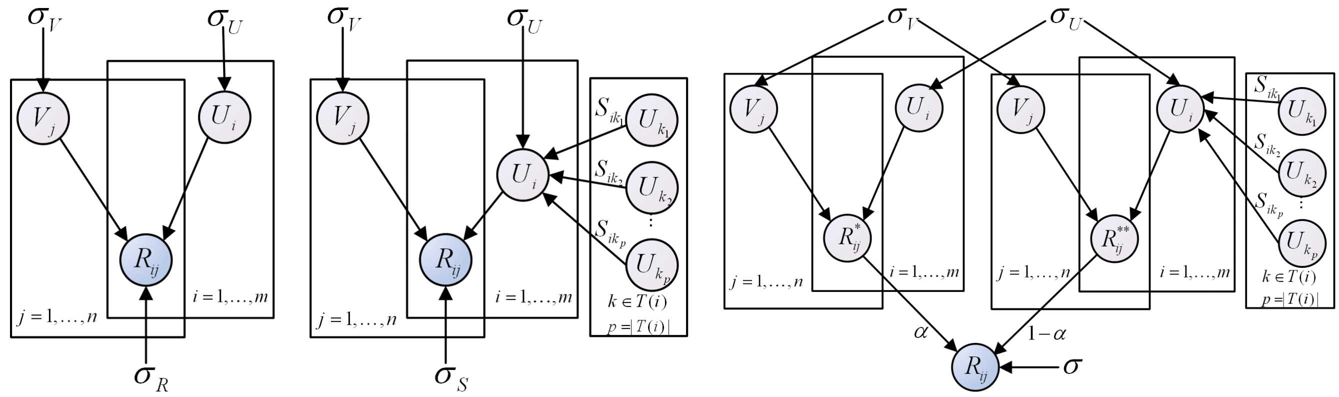
3.3 Recommendations by Trusted Friends

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 paper. 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.

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 Fig. 1(b)), which is now playing at the theaters, but he/she knows nothing about the movie, like user u_1 in Fig. 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 Fig. 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 of watching.

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



(a) Factorization of User-Item Matrix

(b) Recommendations by Trusted Friends

(c) Recommendations with Social Trust Ensemble

Figure 2: Graphical Models

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

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

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

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

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

We can then infer that for all the users to obtain

$$\widehat{R} = SR, \qquad (7)$$

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

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

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

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

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

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

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

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 Fig. 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 above 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:

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

In Eq. (11), the users' favors and the trusted friends' favors are smoothed by the parameter α, which naturally fuses appropriate amount of real world recommendation processes into the recommender systems. The parameter α controls how much do users trust themselves or their trusted friends. It is also the reason we call our approach Recommendation with Social Trust Ensemble (RSTE). The graphical model of RSTE is shown in Fig. 2(c).

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

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

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:

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

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,

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

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 4, 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 the Section 4. 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.

Table 1: 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 2: 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

4. EMPIRICAL ANALYSIS

In this section, we conduct several experiments to compare the recommendation qualities of our 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 the 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?

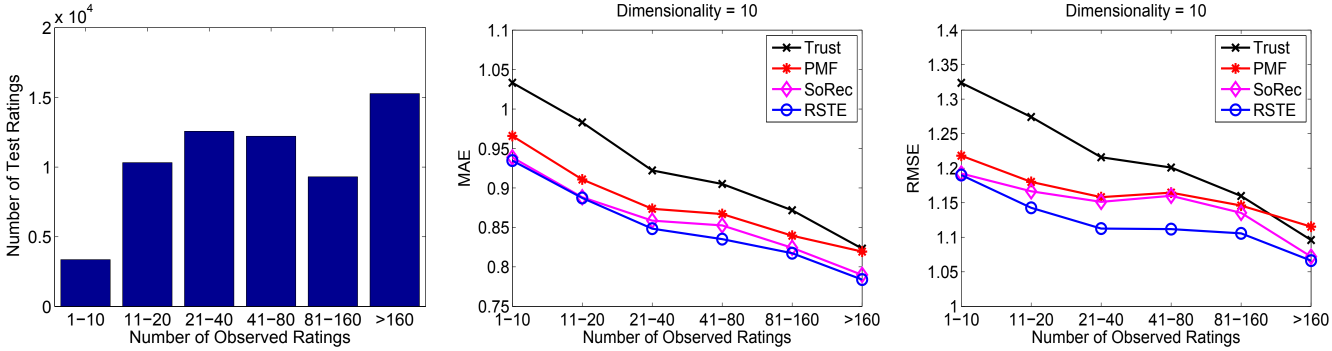
4.1 Dataset Description

We choose Epinions as the data source for our experiments on recommendation with social trust ensemble. 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 Jan 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

Table 3: Performance Comparisons (A Smaller MAE or RMSE Value Means a Better Performance)

Training Data	Metrics	Dimensionality = 5				Dimensionality = 10			
		Trust	PMF	SoRec	RSTE	Trust	PMF	SoRec	RSTE
90%	MAE	0.9054	0.8676	0.8442	**0.8377**	0.9039	0.8651	0.8404	**0.8367**
	RMSE	1.1959	1.1575	1.1333	**1.1109**	1.1917	1.1544	1.1293	**1.1094**
80%	MAE	0.9221	0.8951	0.8638	**0.8594**	0.9215	0.8886	0.8580	**0.8537**
	RMSE	1.2140	1.1826	1.1530	**1.1346**	1.2132	1.1760	1.1492	**1.1256**



(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)

Figure 3: Performance Comparison on Different Users

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 that 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 1. 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 2.

4.2 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:

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r}_{i,j}|}{N}, \tag{15}$$

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:

$$RMSE = \sqrt{\frac{\sum_{i,j} (r_{i,j} - \widehat{r}_{i,j})^2}{N}}. \tag{16}$$

4.3 Comparison

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

1. PMF: this method is proposed by Salakhutdinov and Minh in [19]. It only uses user-item matrix for the recommendations, and it is based on probabilistic matrix factorization.

2. Trust: this is the method purely uses trusted friends' tastes making recommendations. It is proposed in Section 3.3 in this paper. It is also a special case of RSTE when $\alpha = 0$.

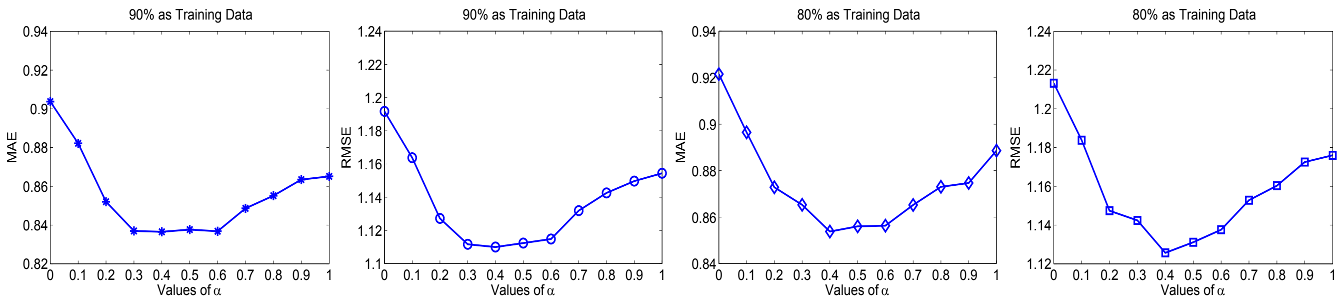
3. SoRec: this is the method proposed in [13]. It is a social trust-aware recommendation method that factorizes the user-item rating matrix and users' social trust network by sharing the same user latent space.

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 Epinions dataset as the training data to predict the remaining 10% of ratings. The 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 3.

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 3, we can observe that our approach RSTE outperforms the other methods. In general, two social trust recommendation approaches SoRec and RSTE all perform better than the PMF method (only uses the user-item matrix for recommendations). However, the Trust method performs worse than the 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.

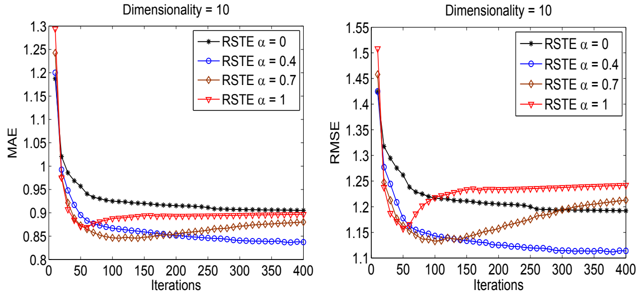
4.4 Performance on Different Users

One challenge of the recommender systems is that it is difficult to recommend items to users who have very few



(a) (b) (c) (d)

Figure 4: Impact of Parameter α (Dimensionality = 10)



(a) (b)

Figure 5: Efficiency Analysis (90% as Training Data)

ratings. 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. 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.

Fig. 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 dataset have rating numbers from 1 to 10. In Fig. 3(b) and Fig. 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.

4.5 Impact of Parameter α

In our method proposed in this paper, 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.

Fig. 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 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 with 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 favors together.

From Fig. 4(a) and Fig. 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 their own tastes, since the training data of user-item matrix is very sparse, which can hardly learn the accurate characteristics of users. In Fig. 4(c) and Fig. 4(d), 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.

4.6 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 only 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.

Fig. 5(a) and Fig. 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 the predictive accuracy can be improved by incorporating appropriate amount of social trust information.

5. CONCLUSIONS AND FUTURE WORK

This paper 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 the recommender systems. Experimental analysis on the Epinions dataset shows the promising future of our proposed method. Moreover, the method introduced in this paper by using probabilistic matrix factorization is not only working in trust-aware recommender systems, but also applicable to other popular research topics, such as social search, collaborative information retrieval, and social data mining.

In this paper, although we employ the trusted friends' opinions in the social trust network to make recommendations for the users, we do not consider the possible diffusions of trusts 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 the 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 the user u_i thinks the user u_j's taste is totally different from him/her. Actually, this information is very useful on the recommender systems. Unfortunately, to the best of our knowledge, no previous work can employ this information well into recommender systems. The understanding of distrust relations is still unclear to the 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 the distrust relations, and explicitly model them in the 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.

6. ACKNOWLEDGMENTS

We thank Dr. Haixuan Yang, Mr. Shikui Tu and Mr. Tom Chao Zhou for many valuable discussions on this topic. The work described in this paper is supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No.: CUHK 4128/08E and CUHK 4158/08E). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

7. REFERENCES

[1] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proc. of WWW '08*, pages 199–208, New York, NY, USA, 2008. ACM.

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

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

[4] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proc. of SIGIR '02*, pages 238–245, New York, NY, USA, 2002. ACM.

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

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of SIGIR '99*, pages 230–237, New York, NY, USA, 1999. ACM.

[7] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. of SIGIR '03*, pages 259–266, New York, NY, USA, 2003. ACM.

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

[9] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proc. of SIGIR '04*, pages 337–344, New York, NY, USA, 2004. ACM.

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

[11] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proc. of SIGIR '08*, pages 83–90, New York, NY, USA, 2008. ACM.

[12] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *Proc. of SIGIR '07*, pages 39–46, New York, NY, USA, 2007. ACM.

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

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

[15] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proc. of RecSys*, pages 17–24, New York, NY, USA, 2007. ACM.

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

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

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

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

[20] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.

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

[22] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proc. of ICML '03*, 2003.

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

[24] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proc. of SIGIR '07*, pages 47–54, New York, NY, USA, 2007. ACM.