Learning to Recommend with Trust and Distrust Relationships

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

ABSTRACT

With the exponential growth of Web contents, *Recommender System* has become indispensable for discovering new information that might interest Web users. Despite their success in the industry, traditional recommender systems suffer from several problems. First, the sparseness of the user-item matrix seriously affects the recommendation quality. Second, traditional recommender systems ignore the connections among users, which loses the opportunity to provide more accurate and personalized recommendations. In this paper, aiming at providing more realistic and accurate recommendations, we propose a factor analysis-based optimization framework to incorporate the user trust and distrust relationships into the recommender systems. The contributions of this paper are three-fold: (1) We elaborate how user distrust information can benefit the recommender systems. (2) In terms of the trust relations, distinct from previous trust-aware recommender systems which are based on some heuristics, we systematically interpret how to constrain the objective function with trust regularization. (3) The experimental results show that the distrust relations among users are as important as the trust relations. The complexity analysis shows our method scales linearly with the number of observations, while the empirical analysis on a large Epinions dataset proves that our approaches perform 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, Trust, Distrust, Matrix Factorization

1. INTRODUCTION

Recommender systems are becoming increasingly indispensable nowadays since they focus on solving the information overload problem by providing users with more proactive and personalized information services. Examples of successful applications of recommender systems include product recommendations at Amazon, movie recommendations at Netflix, etc. Due to the potential commercial value and the great research challenges, recommendation techniques have drawn much attention in data mining [2, 11], information retrieval [15, 30] and machine learning [22, 25] communities. Recommendation algorithms suggesting personalized recommendations greatly increase the likelihoods of customers making the purchase online.

However, no matter what methods are employed, traditional recommender systems only utilize the user-item rating matrix for recommendations. Hence, in order to provide more personalized and accurate recommendations to users, researchers start to study the trust-aware recommender systems. Several trust-aware methods have been proposed to address the data sparsity and recommendation accuracy problems [1, 19, 20, 21].

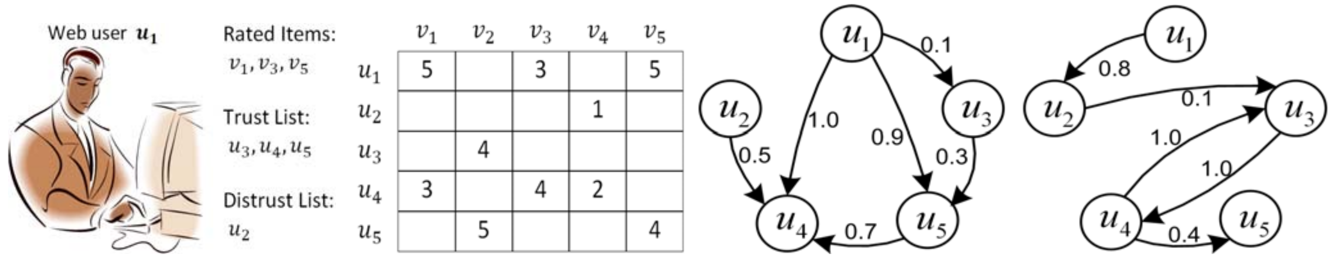
In [19], a trust-aware method for recommender system is proposed. In this work, the collaborative filtering process is informed by the reputation of users which is computed by propagating trust. Trust values are computed in addition to similarity measures between users. In [21], two computational models of trust are proposed and are incorporated into standard collaborative filtering frameworks in a variety of ways. The experimental analysis shows that these trust models can lead to improved predictive accuracy during recommendation.

Although these trust-aware methods move a nice step forward in the research of recommender systems, these methods have several inherent weaknesses. First of all, these methods are all memory-based methods which employ only heuristic algorithms to generate recommendations. Secondly, 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 most of them need to calculate pairwise user similarities and pairwise user trust scores. Lastly, these methods all ignore a very important information, i.e., distrust relations among users.

In this paper, aiming at providing solutions for the problems analyzed above, we propose a factor analysis framework with the constraints of trust and distrust relations among users. Our work is based on the following intuitions:

- Users' latent features can be extracted by factorizing the user-item rating matrix.
- Users' trust relations can be modeled as the "similar" relations due to the reason that user u_i trusts user u_t

RecSys'09, October 23–25, 2009, New York, New York, USA.



(a) A Web User (b) User-Item Rating Matrix (c) User Trust Graph (d) User Distrust Graph

Figure 1: A Toy Example

means that user u_i agrees with most of the opinions issued by u_t.

- Users' distrust relations can be interpreted as the "dissimilar" relations since user u_i distrusts user u_d indicates that user u_i disagrees with most of the opinions issued by user u_d.

Based on the above intuitions, the trust and distrust relations between users can be easily modeled by adding the regularization terms into the objective functions of the user-item matrix factorization. By performing a simple gradient descent on the objective function, we can 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 shows that our method outperforms the state-of-the-art collaborative filtering and trust-aware recommendation algorithms. 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. Section 2 presents our work on recommender systems with trust and distrust constraints. The results of an empirical analysis are presented in Section 3. In Section 4, we provide an overview of several major approaches for recommender systems and other related work, followed by the conclusions and future work in Section 5.

2. RECOMMENDATION FRAMEWORK

Previous recommender system techniques only utilize the information of the user-item rating matrix for recommendations while ignoring the trust and distrust relationships among users. However, the fact is, trust and distrust information is very helpful in making the recommendations since to some extent, they represent the "similar" and "dissimilar" relationships. With the exponential growth of Web 2.0 Web sites, providing personalized recommendations and incorporating trust and distrust into traditional recommender systems are becoming more and more important.

In this section, we first describe the problem we study in Section 2.1, and then brief the matrix factorization technique for recommendation in Section 2.2. We provide solutions on how to incorporate the distrust and trust into recommendations in Section 2.3, Section 2.4 and Section 2.5. Finally, the complexity analysis is conducted in Section 2.6.

2.1 Problem Definition

Fig. 1(a) illustrates a typical Web user we will study in this paper. In this figure, user u_1 rated three items v_1, v_3 and v_5. In addition to the rating data, this user also maintains two lists: trust list and distrust list. The trust list stores all the users that user u_1 trusts while the distrust list includes all the users that user u_1 distrusts.

By integrating all the information from all the users, we summarize three different data sources: the user-item rating matrix shown in Fig. 1(b), the user trust graph shown in Fig. 1(c) and the user distrust graph shown in Fig. 1(d). In this example, totally, there are 5 users (from u_1 to u_5) and 5 items (from v_1 to v_5) with 6 trust relations (edges) and 5 distrust relations between users. Each relation is associated with a weight w_{ij} in the range $(0, 1]$ to specify how much user u_i trusts or distrusts user u_j. In an online social network Web site, the weight w_{ij} is often explicitly stated by user u_i. Typically, each user also rates some items on a 5-point integer scale to express the extent of the favor of each item (normally, 1, 2, 3, 4 and 5 represent "hate", "don't like", "neutral", "like" and "love", respectively).

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

2.2 Matrix Factorization for Recommendation

A common and popular approach to recommender systems is to fit a factor model to the user-item rating matrix, and use it in order to make further predictions [8, 18, 22, 25]. The premise behind a low-dimensional factor model is that there is only a small number of factors influencing the preferences, and that a user's preference vector is determined by how each factor applies to that user [22].

Consider an $m \times n$ user-item rating matrix R, the matrix factorization method employs a rank-l matrix $X = U^T V$ to fit it, where $U \in \mathbb{R}^{l \times m}$ and $V \in \mathbb{R}^{l \times n}$. From the above definition, we can see that the low-dimensional matrices U and V are unknown, and need to be estimated. Moreover, this feature representations have clear physical meanings. In this linear factor model, each factor is a preference vector, and a user's preferences correspond to a linear combination of these factor vectors, with user-specific coefficients. More specifically, each row of U performs as a "feature vector", and each row of V is a linear predictor, predicting the entries in the corresponding column of R based on the "features" in U.

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 $f(x) = x/R_{max}$. However, simply employing $U_i^T V_j$ to predict the missing value $R_{i,j}$ can make the prediction outside of the range of valid rating values. Hence, instead of using a simple linear factor model, in this paper, the inner product between user-specific and movie-specific fea-

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

ture vectors is mapped through a nonlinear logistic function $g(x) = 1/(1+\exp(-x))$, which bounds the range of the predictions into [0, 1].

Hence, by adding the constraints of the norms of U and V, we have the following optimization problem:

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

where 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, and $\|\cdot\|_F^2$ denotes the Frobenius norm.

The optimization problem in Eq. (1) minimizes the sum-of-squared-errors objective function with quadratic regularization terms. It also has a probabilistic interpretation with Gaussian observation noise, which is detailed in [25]. However, the same as many other collaborative filtering methods, this approach only utilizes the user-item rating matrix for the recommendations. In the following sections, we will introduce how to incorporate the distrust and trust information into the matrix factorization method.

2.3 Recommendation with Distrust Relations

In this section, we analyze how the distrust relationships can affect the recommendation processes.

Distrust is one of the most controversial topics and issues to cope with, especially when considering trust metrics and trust propagation [34]. Although many researchers have already conducted comprehensive studies on the trust related applications, the understanding of distrust relations is still unclear to the researchers. Distrust is totally different with trust, hence the method employed in the trust-aware recommender systems cannot be simply transplanted to distrust-aware recommender systems. For example, the most popular method in trust-aware recommender systems is to improve the recommendation quality by the propagation of trust; however, we cannot simply use propagation methods to model distrust due to the reason that one person's enemy's enemy is not necessarily the enemy of this person.

However, we cannot ignore the distrust information since as reported in [6], experience with real-world implemented trust systems such as Epinions and eBay suggests that distrust is at least as important as trust.

In this paper, we employ a simple intuition to make positive influence using distrust information. If a user u_d is in the distrust list of a user u_i, most probably, it is because the user u_i thinks the user u_d's taste is totally different from him/her. Actually, this information is very useful on the recommender systems. We could interpret this problem using the following intuition: if user u_i distrusts user u_d, then we could assume that the features U_i and U_d will have a large distance in the feature space. Based on this assumption, for all the users in the user space, we summarize the following optimization function:

$$\max_U \frac{1}{2}\sum_{i=1}^{m}\sum_{d\in\mathcal{D}^+(i)} S_{id}^{\mathcal{D}}\|U_i - U_d\|_F^2, \qquad (2)$$

where $\mathcal{D}^+(i)$ is the set of users that user u_i distrusts, and $S_{id}^{\mathcal{D}} \in (0,1]$ is the weight of distrust score that user u_i gives to user u_d. The larger the value of $S_{id}^{\mathcal{D}}$ is, the more the user u_i distrusts the user u_d.

Based on Eq. (1) and Eq. (2), we define the recommendation with distrust relations as the following optimization problem:

$$\begin{aligned}\min_{U,V} \mathcal{L}_{\mathcal{D}}(R,S^{\mathcal{D}},U,V) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - g(U_i^T V_j))^2 \\ &+ \frac{\beta}{2}\sum_{i=1}^{m}\sum_{d\in\mathcal{D}^+(i)}(-S_{id}^{\mathcal{D}}\|U_i - U_d\|_F^2) \\ &+ \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2. \qquad (3)\end{aligned}$$

In the online opinion sharing or recommender systems, the distrust value $S_{id}^{\mathcal{D}}$ is typically issued by user u_i explicitly with respect to user u_d, and it cannot accurately describe the relations between users since it contains noises and ignores the graph structure information of distrust network. For instance, similar to the Web link adjacency graph in [32], in a distrust graph, the confidence of distrust value $S_{id}^{\mathcal{D}}$ should be decreased if user u_i distrusts lots of users; however, the confidence of distrust value $S_{id}^{\mathcal{D}}$ should be increased if user u_d is trusted by lots of users. Hence, we propose to smooth the term $S_{id}^{\mathcal{D}}$ by incorporating local authority and local hub values in Eq. (3),

$$S_{id}^{\mathcal{D}} = \frac{\nabla^-(u_d)}{\nabla^+(u_i) + \nabla^-(u_d)} \times S_{id}^{\mathcal{D}}, \qquad (4)$$

where $\nabla^+(u_i)$ represents the outdegree of user u_i in the distrust graph, while $\nabla^-(u_d)$ indicates the indegree of user u_d in the distrust graph.

A local minimum of the objective function given by Eq. (3) can be found by performing gradient descent in U_i, V_j,

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - R_{ij})V_j \\ &+ \beta\sum_{d\in\mathcal{D}^+(i)} S_{id}^{\mathcal{D}}(U_d - U_i) + \beta\sum_{p\in\mathcal{D}^-(i)} S_{pi}^{\mathcal{D}}(U_p - U_i) \\ &+ \lambda_U U_i, \\ \frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - R_{ij})U_i + \lambda_V V_j, \qquad (5)\end{aligned}$$

where $\mathcal{D}^-(i)$ is the set of users that distrust user u_i.

2.4 Recommendation with Trust Relations

In this section, we discuss how to incorporate the trust relationships into recommender systems. In order to model the trust relationships between users realistically, we first need to understand where the "trust" comes from. Actually, on the Web, it is not difficult to interpret the generation of trust relations. For example, in an opinion sharing Web site, if a user u_t is in the trust list of a user u_i, most probably, the underlying cause is that user u_i agrees with most of user u_t's opinions. Moreover, how much user u_i trusts user u_t depends on how much user u_i agrees with user u_t.

Based on the above interpretation, if user u_i trusts user u_t, we can assume that the feature representations U_i and U_d

of these two users are close in the feature space. Following this intuition, we minimize the objective function

$$\min_U \frac{1}{2}\sum_{i=1}^{m}\sum_{t\in\mathcal{T}^+(i)} S_{it}^{\mathcal{T}}\|U_i - U_t\|_F^2, \tag{6}$$

where $\mathcal{T}^+(i)$ is the set of users that user u_i trusts, and $S_{it}^{\mathcal{T}} \in (0, 1]$ is the degree indicates how much user u_i trusts user u_t. The larger the value of $S_{it}^{\mathcal{T}}$ is, the more the user u_i trusts the user u_t.

By employing Eq. (1) and Eq. (6), we define the recommendation problem with trust relations as the following optimization problems:

$$\begin{aligned}\min_{U,V} \mathcal{L}_{\mathcal{T}}(R, S^{\mathcal{T}}, U, V) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R (R_{ij} - g(U_i^T V_j))^2 \\ &+ \frac{\alpha}{2}\sum_{i=1}^{m}\sum_{t\in\mathcal{T}^+(i)} (S_{it}^{\mathcal{T}}\|U_i - U_t\|_F^2) \\ &+ \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2. \end{aligned} \tag{7}$$

Similar to Eq. (4), we also smooth the trust value $S_{it}^{\mathcal{T}}$ in Eq. (7) based on the following equation:

$$S_{it}^{\mathcal{T}} = \frac{\Delta^-(u_t)}{\Delta^+(u_i) + \Delta^-(u_t)} \times S_{it}^{\mathcal{T}}, \tag{8}$$

where $\Delta^+(u_i)$ represents the outdegree of user u_i in the trust graph, while $\Delta^-(u_t)$ indicates the indegree of user u_t in the trust graph.

In Eq. (7), by performing gradient descent in U_i, V_j, we have

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathcal{T}}}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - R_{ij})V_j \\ &+ \alpha \sum_{t\in\mathcal{T}^+(i)} S_{it}^{\mathcal{T}}(U_i - U_t) + \alpha \sum_{q\in\mathcal{T}^-(i)} S_{qi}^{\mathcal{T}}(U_i - U_q) \\ &+ \lambda_U U_i, \\ \frac{\partial \mathcal{L}_{\mathcal{T}}}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - R_{ij})U_i + \lambda_V V_j, \end{aligned} \tag{9}$$

where $\mathcal{T}^-(i)$ is the set of users that trust user u_i.

2.5 Prediction

After the low-dimensional latent feature spaces U and V are learned, the next step is to predict the ratings for the active users. For the given missing data R_{ij}, the value predicted by our method is defined as

$$\hat{R}_{ij} = g(U_i^T V_j). \tag{10}$$

We will evaluate the prediction quality in Section 3.

2.6 Complexity Analysis

The main computation of gradient methods is evaluating the object functions $\mathcal{L}_{\mathcal{D}}$, $\mathcal{L}_{\mathcal{T}}$ and their gradients against variables.

Because of the sparsity of matrices R, $S^{\mathcal{D}}$ and $S^{\mathcal{T}}$, the computational complexities of evaluating the objective functions $\mathcal{L}_{\mathcal{D}}$ are $\mathcal{L}_{\mathcal{T}}$ are $O(\rho_R l + m\overline{r}l)$ and $O(\rho_R l + m\overline{s}l)$, respectively, where ρ_R is the number of nonzero entries in the matrix R, l is the dimensions of the user feature, m is the number of users, $\overline{r}$ is the average number of users that a user distrusts, and $\overline{s}$ is the average number of friends that a user trusts. Since almost all of the online social network graphs fit the power-law distribution, a large long tail of users only have few trusted or distrusted users. This indicates that the values of $\overline{r}$ and $\overline{s}$ are relatively small. Generally, $m\overline{r} << \rho_R$ and $m\overline{s} << \rho_R$.

The computational complexities for the gradients $\frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial U}$ and $\frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial V}$ in Eq. (5) are $O(\rho_R l^2 + m(\overline{r} + \overline{r'})l)$ and $O(\rho_R l^2)$, respectively, where $\overline{r'}$ is the average number of users who distrust a user, which is also a small value. Actually, in a distrust network graph, the value of $\overline{r}$ is always equal to the value of $\overline{r'}$, which is 0.94 in the dataset we employ in the Section 3.

The computational complexities for the gradients $\frac{\partial \mathcal{L}_{\mathcal{T}}}{\partial U}$ and $\frac{\partial \mathcal{L}_{\mathcal{T}}}{\partial V}$ in Eq. (9) are $O(\rho_R l^2 + m(\overline{s} + \overline{s'})l)$ and $O(\rho_R l^2)$, respectively, where $\overline{s'}$ is the average number of friends who trust a user. In a trust network graph, the value of $\overline{s}$ is also equal to the value of $\overline{s'}$, which is 5.45 in the dataset we employ in the experiments.

Therefore, the total computational complexity in one iteration is $O(\rho_R l + \rho_R l^2)$, 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.

3. EXPERIMENTAL ANALYSIS

In this section, we conduct several experiments to compare the recommendation qualities of our approaches 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 do the model parameter α and β affect the accuracy of prediction?

3.1 Dataset Description

We choose Epinions as the data source for our experiments on trust and distrust-aware recommendations. 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 network of trust relationships between users, and a "block (distrust)" list which presents a network of distrust relationships. This network is called the "Web of trust", and is used by Epinions to re-order the product reviews such that a user first sees reviews by users that they trust. Epinions is thus an ideal source for experiments on social recommendation.

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

Statistics	User	Item
Min. Num. of Ratings	1	1
Max. Num. of Ratings	162169	1179
Avg. Num. of Ratings	102.07	17.79

Table 2: Statistics of Trust Network of Epinions

Statistics	Trust per User	Be Trusted per User
Max. Num.	2070	3338
Avg. Num.	5.45	5.45

The dataset used in our experiments consists of 131,580 users who have rated at least one of a total of 755,137 different items. The total number of ratings is 13,430,209. The density of the user-item matrix is 0.014%. We can observe that the user-item 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 reason that we choose the Epinions dataset is that user trust and distrust 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 trust network, the total number of issued trust statements is 717,129. The statistics of the this data source is summarized in Table 2. In the user distrust network, the total number of issued distrust statements is 123,670, and the statistics of the distrust data is summarized in Table 3.

We also observe a number of power-law distributions in these data sources, including items per user, trust relations per user (outdegree in the trust graph) and distrust relations per user (outdegree in the distrust graph). The distributions are shown in Fig. 2.

3.2 Metrics

We employ the Root Mean Square Error (RMSE) to measure the prediction quality of our proposed approaches in comparison with other collaborative filtering and trust-aware recommendation methods.

The metrics RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i,j}(r_{i,j} - \widehat{r}_{i,j})^2}{N}}. \quad (11)$$

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

3.3 Comparison

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

1. PMF (Probabilistic Matrix Factorization): this method is proposed by Salakhutdinov and Minh in [25]. It only uses user-item matrix for the recommendations.

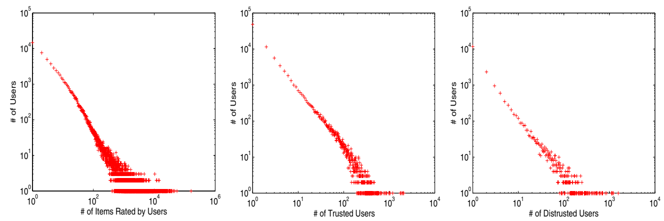
2. SoRec (Social Recommendation): this is the method proposed in [17]. It is a trust-aware recommendation method that factorizes the user-item rating matrix and users' trust network by sharing the same user latent space.

3. RWD (Recommendation With Distrust): this is a matrix factorization-based recommendation method with distrust constraints. It is proposed in Section 2.3 in this paper.

4. RWT (Recommendation With Trust): this is a matrix factorization-based recommendation method with trust constraints. It is proposed in Section 2.4 in this paper.

Table 3: Statistics of Distrust Network of Epinions

Statistics	Distrust per User	Be Distrusted per User
Max. Num.	1562	540
Avg. Num.	0.94	0.94



(a) (b) (c)

Figure 2: Power-Law Distributions of the Epinions Dataset. (a) Items per User Distribution. (b) Trust Graph Outdegree Distribution. (c) Distrust Graph Outdegree Distribution.

As to the training data, we employ three settings: 5%, 10% and 20% for training, where 20% means we randomly select 20% ratings as training data to predict the remaining 80% ratings.

In our RWD and RWT methods, there are totally four parameters need to be set, including α, β, λ_U and λ_V. Without loss of generality, in order to reduce the model complexity, we set $\lambda_U = \lambda_V = 0.001$ in all the experiments we conduct in this paper. We will discuss the influence of the parameters α and β in the experiments conducted in Section 3.4.

The prediction accuracies evaluated by RMSE are shown in Table 4. In our proposed distrust-aware recommendation method RWD, the parameter β is set to be 0.00001 while in our trust-aware recommendation method RWT, the parameter α is set to be 0.001.

From Table 4, we can observe that our RWD and RWT approaches constantly performs better than the other methods in all the settings. When we use 20% as training data, we find that our method generates much better performance than PMF and SoRec. This demonstrates the advantages of trust and distrust-aware recommendation algorithms.

In Fig. 3 and Fig. 4, we also plot the percentages of performance increase of our RWT algorithm against PMF, SoRec as well as our RWD algorithms in terms of RMSE. From these figures, we observe an interesting phenomenon: as the sparsity of the data decreases, the percentages of performance increase against PMF and SoRec keep increasing. This observation is reasonable since in the very spare training settings like 5% and 10%, the user features cannot be accurately learned since the training sample is very sparse.

Table 4: RMSE Comparison with other popular algorithms. The reported values are the RMSE on the Epinions Dataset achieved from dividing the data into 5%, 10%, and 20% for training data, respectively.

Dataset	Traning Data	Dimensionality	PMF	SoRec	RWD	RWT
Epinions	5%	5D	1.228	1.199	1.186	**1.177**
		10D	1.214	1.198	1.185	**1.176**
	10%	5D	0.990	0.944	0.932	**0.924**
		10D	0.977	0.941	0.931	**0.923**
	20%	5D	0.819	0.788	0.723	**0.721**
		10D	0.818	0.787	0.723	**0.720**

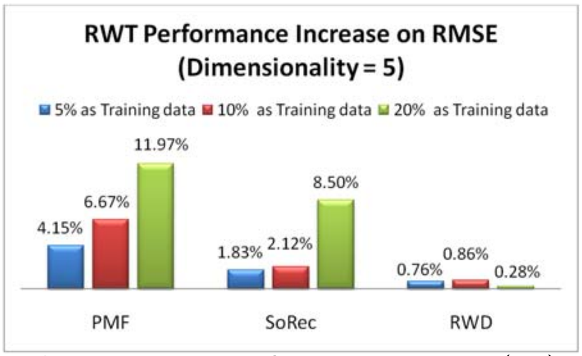


Figure 3: RWT Performance Increase (5D)

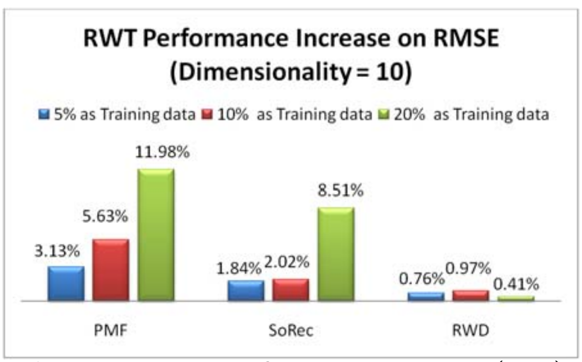
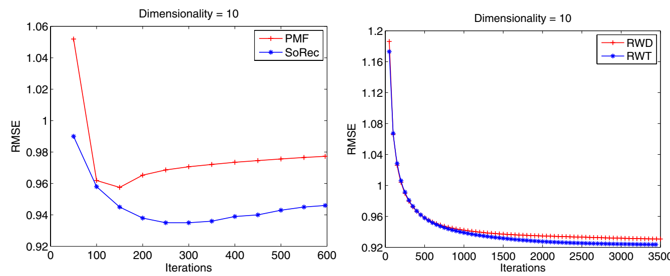


Figure 4: RWT Performance Increase (10D)

Hence our optimization methods cannot maximize the influences of the trust and distrust constraints. But as the increase of the training data, RWD and RWT performs better and better.

We also observe another phenomenon worthy of studying. We find that the distrust-based method RWD performs almost as good as the trust-based method RWT (Please notice that in Table 2 and Table 3, in average, every user only has 0.94 distrusted users while has 5.45 trusted users). This observation proves that the distrust information among users is as important as the trust information in the recommender systems.

In Fig. 5, we plot the performance (RMSE) changes with the iterations. We observe that in the PMF and SoRec methods, at the end of the training, the models begin to overfit, as shown in Fig. 5(a), while our RWD and RWT methods do not have the overfitting problem, as illustrated in Fig. 5(b). These experiments clearly demonstrate that in this dataset, the employ of our trust and distrust regularization terms not only generates better performance than other methods, but also avoids the overfitting problem.



(a) (b)

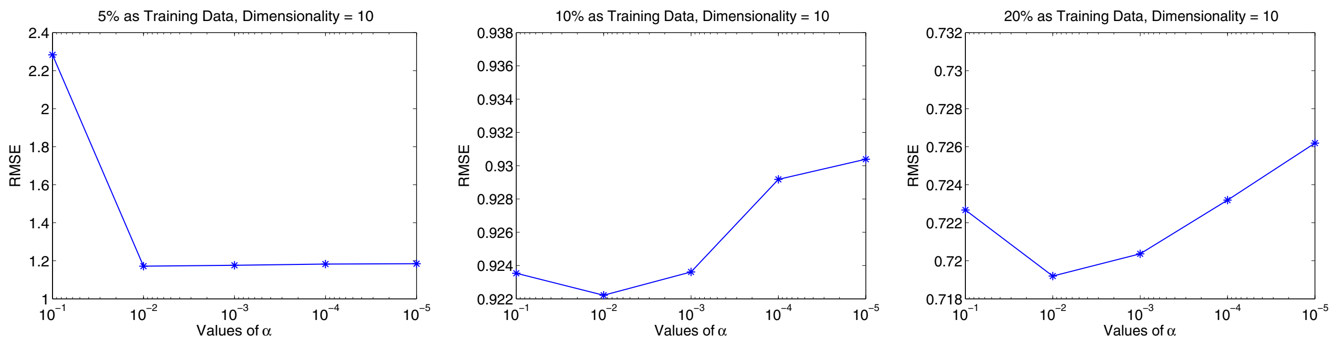
Figure 5: Efficiency Analysis (10% as Training Data). (a) RMSEs of PMF and SoRec Change with Iterations. (b) RMSEs of RWD and RWT Change with Iterations ($\alpha = 0.001$, $\beta = 0.00001$).

3.4 Impact of Parameters α and β

In our method proposed in this paper, the parameters α and β play very important roles. They control how much our method should use the information of trusted or distrusted users. In the extreme case, if we use a very small value of α or β, we only mine the user-item rating matrix for matrix factorization, and simply employ users' own tastes in making recommendations. On the other side, if we employ a very large value of α or β, the trust or distrust information will dominate the learning processes. In normal cases, we integrate information from the user-item rating matrix and the users' trust or distrust network for matrix factorization and, furthermore, to predict ratings for the users.

Fig. 6 shows the impacts of α on RMSE. We observe that the value of α impacts the recommendation results significantly, which demonstrates that incorporating the trust information greatly improves the recommendation accuracy. No matter using 5% training data, 10% training data or 20% training data, as α increases, the RMSE decrease (prediction accuracy increases) at first, but when α surpasses a certain threshold like 0.01, the RMSE increase (prediction accuracy decreases) with further increase of the value of α. The existence of the yielding point confirms with the intuition that purely using the user-item rating matrix or purely using the users' trust information for recommendations cannot generate better performance than appropriately integrating these two sources together.

The impact of β generally shares the same trend as the impact of α. The difference is that we should choose a relatively small value of β, since if we choose a large value, the optimization problem in Eq. (3) will become unbounded, hence we cannot find the solutions.



(a) 5% as Training Data (b) 10% as Training Data (c) 20% as Training Data

Figure 6: Impact of Parameter α

4. 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) trust-aware recommender systems which have drawn lots of attention recently.

Generally, traditional recommender systems can be divided into two categories: memory-based and model-based methods. Memory-based recommender systems, also known as neighborhood-based methods, are the most popular prediction methods and are widely adopted in commercial collaborative filtering systems [13, 23]. Memory-based methods mainly focus on finding the similar users [3, 9] or items [5, 13, 26] for recommendations. User-based approaches predict the ratings of active users based on the ratings of similar users found, and item-based approaches predict the ratings of active users based on the computed information of items similar to those chosen by the active user. User-based and item-based approaches often use the PCC algorithm [23] and the VSS (Vector Space Similarity) algorithm [3] as the similarity computation methods. Recently, a set of related work considers how to utilize the user-based and item-based approaches together [15, 31]. Ma et al. in [15] proposed a method to use the information of users and items to fill in the missing value first before prediction.

In the model-based approaches, training datasets are used to train a predefined model. Examples of model-based approaches include the clustering model [10], aspect models [7, 8, 27], the latent factor model [4], the Bayesian hierarchical model [30] and the ranking model [14]. [10] presented an algorithm for collaborative filtering based on hierarchical clustering, which tried to balance robustness and accuracy of predictions, especially when few data were available. [7] proposed an algorithm based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables. Recently, several matrix factorization methods [22, 24, 25, 28, 33] have been proposed for collaborative filtering. These methods all focus on fitting the user-item rating matrix using low-rank approximations, and use it to make further predictions. The matrix factorization methods or low-dimensional factor models are very efficient in training since they assume that in the user-item rating matrix, only a small number of factors influences preferences, and that a user's preference vector is determined by how each factor applies to that user. In order to take advantages of both the factor models and the neighborhood models, Koren et al. in [2, 11] proposed an interesting idea which merges the factor and neighborhood models, thereby building a more accurate combined model.

Traditional recommender systems have been well studied and developed both in academia and in industry, but they are all based on the assumption that users are independent and identically distributed, and ignore the relationships between users. Based on this intuition, many researchers have recently started to analyze trust-based recommender systems [1, 16, 17, 19, 20, 21].

In [19], a trust-aware method for recommender system is proposed. In this work, the collaborative filtering process is informed by the reputation of users which is computed by propagating trust. Trust values are computed in addition to similarity measures between users. The experiments on a large real dataset shows that this work increases the coverage (number of ratings that are predictable) while not reducing the accuracy (the error of predictions). Bedi et al. in [1] proposed a trust-based recommender system for the Semantic Web; this system runs on a server with the knowledge distributed over the network in the form of ontologies, and uses the Web of trust to generate the recommendations. These methods are all memory-based methods which employ only heuristic algorithms to generate recommendations. There are several problems with this approach, however. Firstly, 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 most of them need to calculate the pairwise user similarities and pairwise user trust scores. Lastly, these methods all ignore a very important information, i.e., distrust relations among users.

In recent work proposed in [17], Ma et al. developed a factor analysis method based on the probabilistic graphical model which fuses the user-item matrix with the users' social trust networks by sharing a common latent low-dimensional user feature matrix. The experimental analysis shows that this method generates better recommendations than the traditional collaborative filtering algorithms. However, this method also failed to model the distrust information since most probably, the users' trust space and distrust space are not the same space, hence cannot simply factorize the trust graph and distrust graph by sharing the same latent feature space.

As reported in [6, 12, 29], distrust also performs a very important role in social networks. In this work, we also investigate how to incorporate distrust information to improve recommender systems.

5. CONCLUSION AND FUTURE WORK

In this paper, we systematically study how to effectively and efficiently incorporate the trust and distrust information into the recommender systems. Our proposed framework is based on matrix factorization with regularization terms constraining the trust and distrust relations between users. The complexity of our proposed optimization framework is linear with the observations of the ratings, and the experimental analysis on a large Epinions dataset shows that our RWD and RWT methods outperforms other state-of-the-arts algorithms. Based on the experimental analysis, we also draw the conclusion that the distrust information is at least as important as the trust information. This observation brings a major contribution to the research of trust and distrust-aware applications since it proves that the distrust information can also be utilized to influence online applications in a positive fashion.

In this paper, the trust and distrust constraints are regularized separately. In order to generate better prediction quality, a possible improvement is to fuse these two data sources into the same objective function. The most direct method is simply attaching the constraints in Eq. (2) and Eq. (6) to the objective function in Eq. (1). However, this will increase the model complexity, hence a more flexible and efficient method needs to be designed in the future.

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 or distrusted "friends" to influence the users' search results or query suggestions. This would be an interesting search phenomenon to explore in social networks.

6. REFERENCES

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

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

[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, Tampere, Finland, 2002.

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

[6] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of WWW '04*, pages 403–412, New York, NY, USA, 2004.

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

[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, Sheffield, United Kingdom, 2004.

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

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

[12] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proc. of WWW '09*, pages 741–750, Madrid, Spain, 2009.

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

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

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

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

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

[18] B. Marlin and R. S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *Proc. of ICML '04*, page 73, Banff, Alberta, Canada, 2004.

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

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

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

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

[23] 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.

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

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

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

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

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

[29] W. Wei, K. T. Chan, I. King, and J. H.-M. Lee. Rate: A review of reviewers in a manuscript review process. In *Proc of WI '08*, pages 204–207, 2008.

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

[31] Z. Zheng, H. Ma, M. R. Lyu, and I. King. WSRec: A collaborative filtering based web service recommender system. In *Proc. of ICWS '09*, pages 437–444, 2009.

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

[33] T. C. Zhou, H. Ma, I. King, and M. R. Lyu. TagRec: Leveraging tagging wisdom for recommendation. In *Proc. of IEEE International Symposium on Social Intelligence and Networking*, 2009.

[34] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.