A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks

Mohsen Jamali
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
mohsen_jamali@cs.sfu.ca

Martin Ester
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
ester@cs.sfu.ca

ABSTRACT

Recommender systems are becoming tools of choice to select the online information relevant to a given user. Collaborative filtering is the most popular approach to building recommender systems and has been successfully employed in many applications. With the advent of online social networks, the social network based approach to recommendation has emerged. This approach assumes a social network among users and makes recommendations for a user based on the ratings of the users that have direct or indirect social relations with the given user. As one of their major benefits, social network based approaches have been shown to reduce the problems with cold start users. In this paper, we explore a model-based approach for recommendation in social networks, employing matrix factorization techniques. Advancing previous work, we incorporate the mechanism of trust propagation into the model. Trust propagation has been shown to be a crucial phenomenon in the social sciences, in social network analysis and in trust-based recommendation. We have conducted experiments on two real life data sets, the public domain Epinions.com dataset and a much larger dataset that we have recently crawled from Flixster.com. Our experiments demonstrate that modeling trust propagation leads to a substantial increase in recommendation accuracy, in particular for cold start users.

Categories and Subject Descriptors

H.2.8.d [**Information Technology and Systems**]: Database Applications - Data Mining

General Terms

Algorithms, Design, Measurement, Experimentation

Keywords

Recommendation, Social Network, Trust, Graphical Model, Matrix Factorization

RecSys2010, September 26–30, 2010, Barcelona, Spain.
Copyright 2010 ACM 978-1-60558-906-0/10/09 ...$10.00.

1. INTRODUCTION

With the rapidly growing amount of information available on the WWW, it becomes necessary to have tools to help users to select the relevant part of online information. To satisfy this need, recommender systems have emerged, e.g. there are popular recommenders for movies[1], books[2], music[3], etc.

Typically in a recommender system, we have a set of *users* and a set of *items*. Each user u rates a set of items by some values. The recommender has the task to predict the rating for user u on a non-rated item i or to generally recommend some items for the given user u based on the ratings that already exist.

Generally two type of recommender systems have been investigated: Memory-based and Model-based recommenders. Memory based algorithms (collaborative filtering) [4] explore the user-item rating matrix and make recommendations based on the ratings of item i by a set of users whose rating profiles are most similar to that of user u. Model-based approaches learn the parameters of a model and store only those parameters. Hence they do not need to explore the rating matrix and only store the model parameters. Model-based approaches are very fast after the parameters of the model are learnt. The bottleneck for model-based approaches is the training phase, while in memory-based approaches there is no training, but the prediction (test) phase is slower.

With the advent of online social networks, the social network based approach to recommendation has emerged. This approach assumes a social network among users and makes recommendations for a user based on the ratings of the users that have direct or indirect social relations with the given user.

Collaborative filtering is most effective when users have expressed enough ratings to have common ratings with other users, but it performs poorly for so-called cold start user. Cold start users are new users who have expressed only a few ratings. Using similarity based approaches, it is unlikely to find similar users since the cold start users only have a few ratings. Social network based recommenders, however, can make recommendations as long as a new user is connected to a large enough component of the social network.

Exploiting social networks in recommendation works because of the effects of selection and social influence that have been postulated by sociologists for a long time. Selection means that people tend to relate to people with similar attributes, and due to social influence related people in a social network influence each other to become more similar[15]. The increasing availability of online social network data has finally allowed a verification of these sociological models. The results of experiments in [1] and of similar

[1] http://www.netflix.com
[2] http://www.amazon.com
[3] http://www.last.fm

work confirm that a social network provides an independent source of information which can be exploited to improve the quality of recommendations.

A social rating network is a social network in which each user expresses ratings on some items besides creating social relations to other users. A sample social rating network is illustrated in figure 1. Note that the terms "trust network" and "social network" are used as synonyms throughout this paper.

Some memory based approaches have been proposed for recommendation in social rating networks [11][3][5][16]. These methods typically explore the social network and find a neighborhood of users trusted (directly or indirectly) by a user and perform the recommendation by aggregating their ratings. These methods use the transitivity of trust and propagate trust to indirect neighbors in the social network. Since they have to explore the social network, memory based methods are slower in the test phase compared to model based approaches.

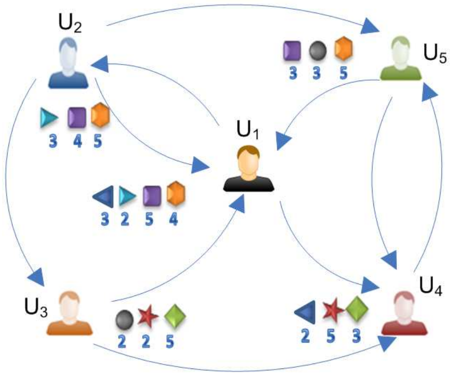

Figure 1: A sample social rating network.

Recently, the model based approach for recommendation in social rating networks has been investigated [9][10]. These methods exploit the matrix factorization technique to learn latent features for users and items from the observed ratings. Experimental results show better performance compared to state of the art memory-based approaches. However, these methods do not consider the propagation of trust. In this paper, we also propose a matrix factorization based model for recommendation in social rating networks, called SocialMF. We incorporate the propagation of trust in our model to improve the quality of recommendation. To inject social influence in our model, we make the features of every user dependent on the feature vectors of the his direct neighbors in the social network. Using this idea, latent features of users indirectly connected in the social network will be dependent and hence the trust gets propagated.

Cold start users are one of the most important challenges in recommender systems. Since cold start users are more dependent on the social network compared to users with more ratings, the effect of using trust propagation gets more important for cold start users. Moreover, in many real life social rating networks a very large portion of users do not express any ratings, and they only participate in the social network. Hence, using only the observed ratings does not allow to learn the user features. The SocialMF model forces the user feature vectors to be close to those of their neighbors to be able to learn the latent user features for users with no or very few ratings.

Due to the sensitive nature of social network data, there are only very few public social rating network datasets. We present a new large scale dataset that we crawled from Flixster.com. Flixster is a social networking service in which users can rate movies and create a social network. All experiments were performed on this dataset as well as on the existing Epinions dataset.

The list of contributions of this paper are as follows:

- We introduce the novel SocialMF model which incorporates trust propagation in the matrix factorization approach.
- SocialMF achieves significantly reduced recommendation error (RMSE) in particular for cold start users.
- We performed experiments on two real life data sets from Epinons.com and Flixster.com, the latter of which we crawled and prepared for research purposes.

The rest of this paper is organized as follows: The problem definition is presented in section 2. Some related work is discussed in section 3. We introduce our proposed model in section 4. We describe the real life data sets used in our experiments in section 5. Our experiments are reported in section 6. Finally, we conclude the paper and present some directions for future work.

2. PROBLEM DEFINITION AND PRELIMINARIES

In recommender systems we have a set of users $\mathbb{U} = \{u_1, \dots u_N\}$ and a set of items $\mathbb{I} = \{i_1, \dots i_M\}$. The ratings expressed by users on items are given in a rating matrix $R = [R_{u,i}]_{N \times M}$. In this matrix $R_{u,i}$ denotes the rating of user u on item i. $R_{u,i}$ can be any real number, but often ratings are integers in the range $[1, 5]$. In this paper, without loss of generality, we map the ratings *1, ..., 5* to the interval [0,1] by normalizing the ratings. In a social rating network, each user u has a set N_u of direct neighbors and $t_{u,v}$ denotes the value of social trust u has on v as a real number in $[0, 1]$. Zero means no trust and one means full trust. Binary trust networks are the most common trust networks (Amazon[4], eBay[5], ...). The trust values are given in a matrix $T = [T_{u,v}]_{N \times N}$. Non-zero cells $T_{u,v}$ in T denote the existence of a social relation from u to v. Note that T is asymmetric in general.

The social rating network can also be defined as a graph in which there is a node corresponding to each user, an edge corresponding to each social trust relation, and item ratings associated with nodes. An example of a graph representation of social rating network is demonstrated in figure 1.

The task of a recommender is as follows: Given a user $u \in \mathbb{U}$ and an item $i \in \mathbb{I}$ for which $R_{u,i}$ is unknown, predict the rating for u on item i using R and T.

In this paper, we employ matrix factorization techniques to learn the latent characteristics of users and items and predict the unknown ratings using these latent characteristics. Let $U \in \mathbb{R}^{K \times N}$ and $V \in \mathbb{R}^{K \times M}$ be latent user and item feature matrices, with column vectors U_u and V_i representing K-dimensional user-specific and item-specific latent feature vectors of users u and item i, respectively. The goal of matrix factorization is to learn these latent variables and exploit them for recommendation.

Traditional recommender systems, like collaborative filtering approaches [14][7][4], only utilize the information of the user-item rating matrix for recommendations but ignore the social relations among users. With the exponential growth of online social networks, incorporating social networks into recommender systems is becoming more and more important. In this section, we first review

[4] www.amazon.com

[5] www.ebay.com

the basic matrix factorization (MF) approach for recommendation using only user-item rating matrix . In section 4, we introduce our proposed model to employ matrix factorization (MF) for recommendation in social rating networks.

In order to learn the latent features of users and items, [14] employs matrix factorization to factorize the user-item matrix. The conditional probability of the observed ratings is defined as:

$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{N}\prod_{i=1}^{M}\Big[\mathcal{N}\Big(R_{u,i}|g(U_u^TV_i),\sigma_r^2\Big)\Big]^{I_{u,i}^R} \qquad (1)$$

where $\mathcal{N}(x|\mu,\sigma^2)$ is the normal distribution with mean μ and variance σ^2, and $I_{u,i}^R$ is the indicator function that is equal to 1 if u has rated i and equal to 0 otherwise. The function $g(x)$ is the logistic function $g(x) = 1/(1+e^{-x})$, which bounds the range of $U_u^TV_i$ within [0,1]. Also, zero mean Gaussian priors are assumed for user and item feature vectors:

$$p(U|\sigma_U^2) = \prod_{u=1}^{N}\mathcal{N}(U_u|0,\sigma_U^2\mathbf{I}),\ p(V|\sigma_V^2) = \prod_{i=1}^{M}\mathcal{N}(V_i|0,\sigma_V^2\mathbf{I}) \quad (2)$$

Now, through a Bayesian inference, the posterior probability of the latent variables U and V can be obtained as follows:

$$\begin{aligned} p(U,V|R,T,\sigma_R^2,\sigma_U^2,\sigma_V^2) &\propto p(R|U,V,\sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ &= \prod_{u=1}^{N}\prod_{i=1}^{M}\Big[\mathcal{N}\Big(R_{u,i}|g(U_u^TV_i),\sigma_r^2\Big)\Big]^{I_{u,i}^R} \\ &\quad \times \prod_{u=1}^{N}\mathcal{N}\Big(U_u|0,\sigma_U^2\mathbf{I}\Big) \times \prod_{i=1}^{M}\mathcal{N}\Big(V_i|0,\sigma_V^2\mathbf{I}\Big) \qquad (3) \end{aligned}$$

The corresponding graphical model is presented in figure 2. Using equation 3, we can learn the latent feature vectors of users and items purely based on the user-item rating matrix.

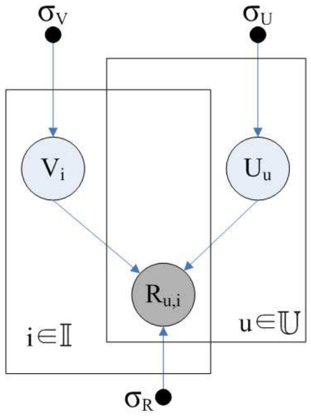

Figure 2: Graphical Model of the baseline factorization of user-item rating matrix.

3. RELATED WORK

In this section we review some related work on recommendation in social networks. Trust propagation in recommendation has been widely investigated in the memory based approaches. Hence, we first review some of the memory based methods for recommendation in social networks. Matrix factorization has been widely used in the model based recommendation [6][7][14]. However these models do not take into account the social network among users. Recently, some model-based approaches have been proposed which use matrix factorization for recommendation in social networks [9][10], however, these works do not consider the propagation of trust. In this section, after reviewing memory based approaches we discuss some model based approaches for recommendation in social networks.

TidalTrust[3] performs a modified breadth first search in the trust network to compute a prediction. Basically, it finds all raters with the shortest path distance from the source user and aggregates their ratings weighted by the trust between the source user and these raters. To compute the trust value between user u and v who are not directly connected, TidalTrust aggregates the trust value between u's direct neighbors and v weighted by the direct trust values of u and its direct neighbors.

[11] introduces MoleTrust. The ideas used in MoleTrust and TidalTrust are similar. But MoleTrust considers all raters up to a *maximum-depth* given as an input. *maximum-depth* is independent of any specific user and item. Also, to compute the trust value between u and v in MoleTrust, we perform a backward exploration. It means that the trust value from u to v is the aggregation of trust values between u and users directly trusting v weighted by the direct trust values.

The Advogato[8] maximum flow trust metric has been proposed in order to discover which users are trusted by members of an online community. The input for Advogato is given by an integer number n, the number of members to trust. To assign capacities to the edges of the network, they need to transform the network, so it needs to know the whole structure of the network. Moreover, it only computes the nodes to trust and does not compute different degrees of trust. Since the number of users to trust is independent of users and items and there is no distinction between the trusted users, this approach is not appropriate for trust-based recommendation.

In order to consider enough ratings without suffering from noisy data, [5] proposes a random walk method (TrustWalker) which combines trust-based and item-based recommendation. TrustWalker considers not only ratings of the target item, but also those of similar items. The probability of using the rating of a similar item instead of a rating for the target item increases with increasing length of the walk. Their framework contains both trust-based and item-based collaborative filtering recommendations as special cases. Their experiments show that their method outperforms other existing memory based approaches. The random walk model allows them to compute the confidence in the predictions.

The authors of [9] proposed a matrix factorization approach for social network based recommendation, called STE. Their method is a linear combination of basic matrix factorization approach [14] and a social network based approach. The graphical model for their proposed model is illustrated in figure 3[6]. The predicted rating of user u on item i is as follows:

$$\hat{R}_{u,i} = g(\alpha U_u^TV_i \ + \ (1-\alpha)\sum_{v\in N_u}T_{u,v}U_v^TV_i) \qquad (4)$$

where parameter α controls the effects of neighbors on the estimated rating.

[6] It should be noted that model in figure 3 is different from the graphical model presented in [9], but it correctly represents the joint probability distribution actually computed for the STE model.

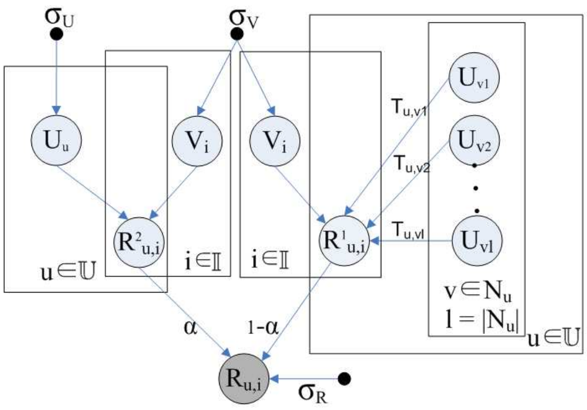

Figure 3: The STE model [9].

Experiments show that their model outperforms the basic matrix factorization based approach and existing trust based based approaches. However, in their model, the feature vectors of direct neighbors of u affect the ratings of u instead of affecting the feature vector of u. This model does not handle trust propagation. We use this method as our main comparison partners in our experiments and call it as *STE* model.

The same authors have also proposed another approach for trust based recommendation in their previous work [10]. In this paper, they developed a factor analysis method based on the probabilistic matrix factorization. In this model, they consider three sets of latent features: U for users, V for items, and F for factors. They factorize the rating matrix R using latent item features V and latent user features U. On the other hand, they factorize the trust matrix T using latent user features U and latent factor features F. They assume a factor latent vector for each user. The graphical model for this model is presented in figure 4.

As stated in [9], although the users' social network is integrated into the recommender systems by factorizing the social trust graph, the real world recommendation processes are not reflected in the model. Two sets of different feature vectors are assumed for users which makes the interpretability of the model very hard. Moreover, experiments in [9] show that the more recent STE model outperforms this model in terms of the RMSE values.

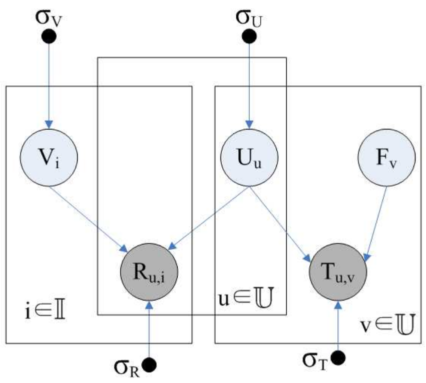

Figure 4: The graphical model of the model presented in [10].

4. THE SOCIALMF MODEL

In this section, we present our approach to incorporate trust propagation into a matrix factorization model for recommendation in social networks.

Due to social influence [2], the behavior of a user u is affected by his direct neighbors N_u. In other words, the latent feature vector of u is dependent on the latent feature vectors of all his direct neighbors $v \in N_u$. We formulate this influence as follows:

$$\widehat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}} = \frac{\sum_{v \in N_u} T_{u,v} U_v}{|N_u|} \qquad (5)$$

where $\widehat{U}_u$ is the estimated latent feature vector of u given the feature vectors of his direct neighbors. Since the social networks we are working with are all binary social networks, all none-zero values of $T_{u,v}$ are 1. We normalize each row of the trust matrix so that $\sum_{v=1}^{N} T_{u,v} = 1$. Now, we have:

$$\widehat{U}_u = \sum_{v \in N_u} T_{u,v} U_v \qquad (6)$$

The above equation indicates that the estimate of the latent feature vector of a user is the weighted average of the latent feature vectors of his direct neighbors. The estimated feature vector of user u can be inferred as:

$$\begin{pmatrix} \widehat{U}_{u,1} \\ \widehat{U}_{u,2} \\ \ldots \\ \widehat{U}_{u,K} \end{pmatrix} = \begin{pmatrix} U_{1,1} & U_{2,1} & \ldots & U_{N,1} \\ U_{1,2} & U_{2,2} & \ldots & U_{N,2} \\ \ldots & .. & \ldots & \ldots \\ U_{1,K} & U_{2,K} & \ldots & U_{N,K} \end{pmatrix} \begin{pmatrix} T_{u,1} \\ T_{u,2} \\ \ldots \\ T_{u,N} \end{pmatrix} \qquad (7)$$

Note that taking the social network into account does not change the equation for the conditional distribution of the observed ratings. It only affects the user latent feature vectors. So the conditional probability of observed rating is the same as the conditional probability in equation 1:

$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[\mathcal{N}\Big(R_{u,i} | g(U_u^T V_i), \sigma_r^2\Big) \right]^{I_{u,i}^R} \qquad (8)$$

For the user latent features, we have two factors: The zero-mean Gaussian prior to avoid over-fitting, and the conditional distribution of user latent features given the latent features of his direct neighbors. Therefore,

$$\begin{aligned} & p(U|T,\sigma_U^2,\sigma_T^2) \propto p(U|\sigma_U^2) \times p(U|T,\sigma_T^2) \\ & = \prod_{u=1}^{N} \mathcal{N}\Big(U_u | 0, \sigma_U^2 \mathbf{I}\Big) \times \prod_{u=1}^{N} \mathcal{N}\Big(U_u | \sum_{v \in N_u} T_{u,v} U_v, \sigma_T^2 \mathbf{I}\Big) \end{aligned} \qquad (9)$$

The above distribution is a normal distribution which is a product of two different normal distributions to keep the user feature vectors both small and close to the features of their direct neighbors.

Similar to equation 3, through a Bayesian inference, we have the following equation for the posterior probability of latent feature vectors given the rating and social trust matrices:

$$
\begin{aligned}
&p(U,V|R,T,\sigma_R^2,\sigma_T^2,\sigma_U^2,\sigma_V^2) \propto \\
&\quad p(R|U,V,\sigma_R^2)p(U|T,\sigma_U^2,\sigma_T^2)p(V|\sigma_V^2) \\
&\quad = \prod_{u=1}^{N}\prod_{i=1}^{M}\Big[\mathcal{N}\Big(R_{u,i}|g(U_u^T V_i),\sigma_r^2\Big)\Big]^{I_{u,i}^R} \\
&\qquad \times \prod_{u=1}^{N}\mathcal{N}\Big(U_u|\sum_{v\in N_u} T_{u,v}U_v,\sigma_T^2\mathbf{I}\Big) \\
&\qquad \times \prod_{u=1}^{N}\mathcal{N}\Big(U_u|0,\sigma_U^2\mathbf{I}\Big) \times \prod_{i=1}^{M}\mathcal{N}\Big(V_i|0,\sigma_V^2\mathbf{I}\Big) \qquad (10)
\end{aligned}
$$

The graphical model corresponding to equation 9 is shown in figure 5. Note that the trust matrix in the above equation is not explicitly shown in the figure. However, the edges among the latent feature vectors of users are representatives of the trust network among users and the degree of trust of user u on user v is $T_{u,v}$.

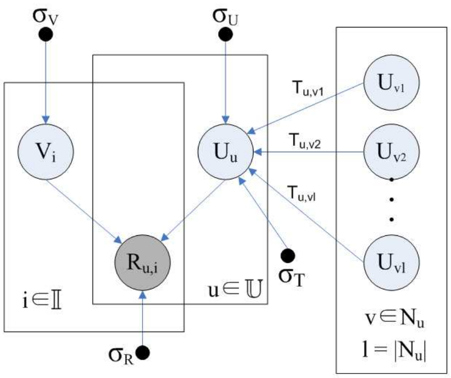

Figure 5: Proposed Graphical Model to consider the social network in the factorization of user-item rating matrix.

The log of the posterior probability can be computed as follows:

$$
\begin{aligned}
&lnp(U,V|R,T,\sigma_R^2,\sigma_T^2,\sigma_U^2,\sigma_V^2) = \\
&\quad -\frac{1}{2\sigma_R^2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^R(R_{u,i}-g(U_u^T V_i))^2 \\
&\quad -\frac{1}{2\sigma_U^2}\sum_{u=1}^{N} U_u^T U_u - \frac{1}{2\sigma_V^2}\sum_{i=1}^{M} V_i^T V_i \\
&-\frac{1}{2\sigma_T^2}\sum_{u=1}^{N}\Big((U_u-\sum_{v\in N_u}T_{u,v}U_v)^T(U_u-\sum_{v\in N_u}T_{u,v}U_v)\Big) \\
&\quad -\frac{1}{2}(\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^R)\ln\sigma_R^2 \\
&-\frac{1}{2}((N\times K)\ln\sigma_U^2 + (M\times K)\ln\sigma_V^2 + (N\times K)\ln\sigma_T^2) + \mathcal{C}
\end{aligned}
\qquad (11)
$$

Keeping the parameters(observation noise variance and prior variance) fixed, maximizing the log-posterior over latent features of users and items is equivalent to minimizing the following objective function, which is a sum of squared errors with quadratic regularization terms:

$$
\begin{aligned}
\mathcal{L}(R,T,U,V) &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^R(R_{u,i}-g(U_u^T V_i))^2 \\
&\quad + \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u^T U_u + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i^T V_i \\
&+ \frac{\lambda_T}{2}\sum_{u=1}^{N}\Big((U_u-\sum_{v\in N_u}T_{u,v}U_v)^T(U_u-\sum_{v\in N_u}T_{u,v}U_v)\Big) \qquad (12)
\end{aligned}
$$

In the above equation, $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, and $\lambda_T = \sigma_R^2/\sigma_T^2$.

We can find a local minimum of the objective function in equation 4 by performing gradient descent on U_u and V_i for all users u and all items i.

$$
\begin{aligned}
\frac{\partial\mathcal{L}}{\partial U_u} &= \sum_{i=1}^{M} I_{u,i}^R V_i g'(U_u^T V_i)(g(U_u^T V_i)-R_{u,i}) + \lambda_U U_u \\
&+\lambda_T(U_u-\sum_{v\in N_u}T_{u,v}U_v)) - \lambda_T\sum_{\{v|u\in N_v\}} T_{v,u}\Big(U_v-\sum_{w\in N_v}T_{v,w}U_w\Big)
\end{aligned}
\qquad (13)
$$

$$
\frac{\partial\mathcal{L}}{\partial V_i} = \sum_{u=1}^{N} I_{u,i}^R U_u g'(U_u^T V_i)(g(U_u^T V_i)-R_{u,i}) + \lambda_V V_i \qquad (14)
$$

where $g'(x)$ is the derivative of logistic function and is equal to $g'(x) = e^{-x}/(1+e^{-x})^2$. To reduce the model complexity, we set $\lambda_U = \lambda_V$ in all our experiments. The initial values of U and V are samples from normal noises with zero mean. In each iteration, U and V are updated based on the latent variables from the previous iteration.

4.1 Desirable properties of the proposed model

In this section, we discuss some desirable properties of SocialMF and compare it against the closely related STE model [9].

The SocialMF model addresses the transitivity of trust in social networks. In other words, our model takes the trust propagation into account. According to the graphical model, the feature vector of any user is dependent on the feature vectors of his direct neighbors. Recursively, the feature vector of each direct neighbor is dependent on the feature vector of his direct neighbors. This effect is shown in the conditional distributions by considering the feature vector of a user being a normal distribution around the average of the feature vectors of his neighbors. On the other hand, the STE model [9] does not support trust propagation and they list trust propagation as future work.

In the baseline MF approach [14] and the STE model [9], the features are being learnt based only on the observed ratings. However, in real life social rating networks, a huge portion of users have expressed no ratings and they participate only in the social network. So their features can not be learnt based on their observed ratings. However, our model can handle these users very well. The SocialMF model learns to tune the latent features of these users close to their neighbors. So, despite not having any expressed ratings, the feature vectors of these users will be learnt to be close to their neighbors. Basically, the social trust relations among users is an observed dependency among the feature vectors of users. It should be noted that since evaluating the learnt features is typically based on the withheld observed ratings, we are currently not able to evaluate the features learnt for users with no expressed ratings.

4.2 Complexity analysis of parameter learning

The main cost in learning the parameters is computing $\mathcal{L}$ and its gradients against feature vectors of users and items. Assuming the average number of ratings per user is $\overline{r}$, and the average number of direct neighbors per user is $\overline{t}$, the complexity of evaluation of $\mathcal{L}$ is $O(N\overline{r}K + N\overline{t}K)$. Since both the rating matrix R and trust matrix T are very sparse, $\overline{t}$ and $\overline{r}$ are relatively small. So the computation of the objective function $\mathcal{L}$ is very fast and linear with respect to the number of users in the social rating network. The computational complexity of computing the gradients is $O(N\overline{r}K + N\overline{t}^2K)$ which is linear with respect to the number of users in the social rating network. Note that the cost of computing the gradient in STE [9] is $O(N\overline{r}\overline{t}^2K)$. So SocialMF is $\frac{\overline{r}\overline{t}^2}{\overline{r}+\overline{t}^2}$ times faster than STE in computing the gradient in each iteration of parameter learning process.

For each rating estimation, the model proposed in [9] needs to take the average of estimated ratings for direct neighbors which makes it slower in prediction compared to SocialMF proposed in this paper.

5. DATASETS

One of the bottlenecks to research in social network analysis in general, and in social rating networks in particular, has been the lack of publicly available datasets. To the best of our knowledge, so far the Epinions.com dataset is the only publicly available social rating network dataset. In this section we briefly introduce the data sets we use in our experiments: the Flixster dataset, we crawled and prepared for this research, and the public domain Epinions dataset.

5.1 Flixster dataset

For this research, we crawled a large scale social rating network dataset from the Flixster website[7]. Flixster is a social networking service in which users can rate movies. Users can also add some users to their friend list and create a social network. Note that social relations in Flixster are undirected.

Flixster has many ways to gather ratings from users. First, users can go online, log into Flixster.com and rate some movies or make friendship connections to other users. In addition, Flixster uses some applications to gather ratings from users. There are two Flixster applications which are very famous in Facebook[8] and Myspace[9]. When users of facebook or myspace install these applications on their profile, they will be asked to rate different movies. Initially, they are presented with a fixed set of 50 movies to rate. Many users of Flixster rate only these movies. As a results the rating counts for these 50 movies are extraordinarily high. Hence, we remove the ratings for these movies to reduce the bias of the dataset. Although this information could be useful in a real recommender system, we ignore it to have a less biased dataset in our research.

Possible rating values in Flixster are 10 discrete numbers in the range [0.5,5] with step size 0.5. Users are also allowed to rate the items with two other nominal values: "Want To See" and "Not Interested". These two types of rating also include some information and may be useful for some research purposes. However, nominal ratings are not considered in the rest of this paper since we can not easily convert them to numerical rating values. Almost half of the ratings are numerical values. The following statistics are only for the numerical ratings.

General statistics of the Flixster dataset are shown in table 1. According to this table, on average each user has 27 friends and each user has rated 8.2 movies. However, if we ignore the many users who have not rated any movies and only consider users with at least one rating, each user has rated 55 movies on average.

Statistics	**Flixster**	**Epinions**
Users	1M	71K
Social Relations	26.7M	508K
Ratings	8.2M	575K
Items	49K	104K
Users with Rating	150K	47K
Users with Friend	980K	60K

Table 1: General statistics of the Flixster and Epinions

The Flixster dataset contains ratings expressed by users during the period from November 2005 to November 2009. The data set is now publicly available at www.sfu.ca/∼sja25/datasets/ .

5.2 Epinions dataset

To the best of our knowledge, the Epinions[10] dataset is the only social rating network dataset publicly available.

We used the version of the Epinions dataset[11] published by the authors of [13]. Table 1 also shows the statistics of the Epinions dataset. Each user has on average 8.1 expressed ratings and 7.2 direct neighbors. Unlike Flixster, the social relations in Epinions are directed. The distribution of the number of ratings per user follows a power law. It should also be noted that unlike Flixster, the items in Epinions are from different categories such as cameras, dvd players, music, etc, while all the items in the Flixster dataset are movies.

Based on the statistics presented in table 1, the crawled Flixster dataset is denser than the Epinions dataset. The number of ratings and friends per user is larger in Flixster dataset. A large portion of users in Flixster dataset have no expressed ratings, but most of them have social relations. Users without any ratings are also important. They may not be useful to compute the prediction for other users based on their own ratings, but they may allow us to connect indirectly to other users who have rated items.

6. EXPERIMENTS

We performed experiments on the Epinions and the Flixster data sets described in section 5. In this section, we report our experimental results and compare the results with existing methods. Also we present the results for different settings of model parameters.

6.1 Experimental Setup

We perform 5-fold cross validation in our experiments. In each fold we have 80% of data as the training set and the remaining 20% as the test data.

The evaluation metric we use in our experiments is RMSE which is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,i)|R_{test}} (r_{u,i} - \widehat{r}_{u,i})^2}{|R_{test}|}} \tag{15}$$

where R_{test} is the set of all pairs (u, i) in the test data.

To evaluate the performance of our method we consider three comparison partners:

[7] www.flixster.com

[8] http://apps.facebook.com/flixster/

[9] http://www.myspace.com/flixstermovies

[10] www.epinions.com

[11] http://alchemy.cs.washington.edu/data/epinions/

- BaseMF: This method is the baseline matrix factorization approach proposed in [14], which does not take the social network into account.

- STE: This is the model proposed in [9], which takes into account the social network in a way different from SocialMF. We set $\alpha = 0.4$ for STE in our experiments which is the optimum value according to the results of experiments in [9].

- CF: This is the well-known user based collaborative filtering method which is a memory based approach.

In our experiments, we refer to our proposed model as *SocialMF*. In all our experiments, we set $\lambda_U = \lambda_V = 0.1$.

6.2 Experimental Results

Table 2 reports the RMSE values of all comparison partners on the Epinions data set. The parameter λ_T is set to 5 for experiments on Epinions. Table 2 shows that SocialMF outperforms existing methods. Note that since collaborative filtering has no latent features, there is no dimensionality K associated with it and hence the result for different values of K are the same.

SocialMF improves the RMSE of the state-of-the-art method STE by 6.2%. for K=5 and by 5.7% for K=10. To show how significant our gain is, note that the gain of STE over the baseline MF method is 2.5% and the gain of SocialMF over STE is more than 2 times that gain. As another evidence for the significance of these RMSE reductions, note that in the Netflix prize competition[12], there was a \$1 Million reward for a reduction of the RMSE by 10%.

Method	K=5	K=10
CF	1.180	1.180
BaseMF	1.175	1.195
STE	1.145	1.150
SocialMF	1.075	1.085

Table 2: RMSE values for comparison partners on Epinions with different settings of dimensionality K.

RMSE values for Flixster are presented in table 3. Again, SocialMF clearly outperforms existing methods. In Flixster, the improvement of the RMSE for SocialMF over STE is 5% which more than 3 times of the gain of STE over baseline MF (1.5%).

It should be noted that the results for Flixster are generally better than the results for Epinions for all methods, possibly because of the fact that the items in Epinions are from multiple categories such as DVD players, cameras, printers, laptops, ..., while the items in Flixster are all movies, which makes the recommendation easier in general. Another explanation for the better results on Flixster could be that it Flixster is a richer dataset since there are more social relations and ratings per user in Flixster compared to Epinions.

Method	K=5	K=10
CF	0.911	0.911
BaseMF	0.878	0.863
STE	0.864	0.852
SocialMF	0.821	0.815

Table 3: RMSE values for comparison partners on Flixster with different settings of dimensionality K.

Intuitively, increasing K should add more flexibility to the model and hence should improve the results. However, comparing results of tables 2 and 3 for different values of K shows that increasing K in Epinions did not improve the results, while increasing K in Flixster improved the results. We believe that these counter-intuitive results for Epinions are due to the fact that the Epinions data set is smaller than Flixster and increasing K leads to more parameters in the model which leads to overfitting. Flixster, on the other hand, is a huge data set, and increasing K to 10 does not lead to overfitting.

6.3 Impact of λ_T on the results

Parameter λ_T controls the influence of the social network on the behavior of users. Larger values of λ_T in the objective function of equation 12 indicate more impact of the social network on the behavior of users. Very small values of λ_T makes our model close to the baseline MF approach. However, very large values of λ_T lead to a model in which having feature vectors close to those of direct neighbors dominating having lower squared error in the training phase.

Figures 6 and 7 compare the RMSE of our model for different ranges of values for λ_T in both data sets. As shown in these figures, SocialMF has its best results on Epinions for $\lambda_T = 5$, and $\lambda_T = 1$ for Flixster.

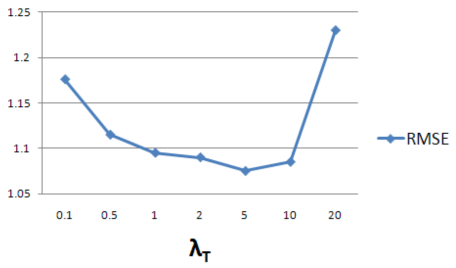

Figure 6: Impact of different values of λ_T on the performance of prediction in Epinions.

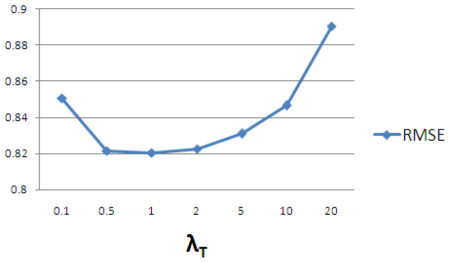

Figure 7: Impact of different values of λ_T on the performance of prediction in Flixster.

6.4 Performance on cold start users

Some users in a social rating network express a lot of ratings, but most users express a few ratings. We consider users who have expressed less than 5 ratings as cold start users [12][5]. In both Flixster and Epinions more than 50% of users are cold start users[13]. Hence efficiency of any recommendation algorithm for cold start users becomes very important.

Table 4 shows that for cold start user, proposed SocialMF outperforms other methods. The improvement of the RMSE for cold

[12] http://www.netflixprize.com

[13] In Flixster, we do not take into account the users with no ratings in this statistics.

Method	Epinions	Flixster
CF	1.361	1.228
BaseMF	1.352	1.213
STE	1.295	1.152
SocialMF	1.159	1.057

Table 4: RMSE values on cold start users (K=5).

start users compared to STE is 11.5% for Epinions and 8.5% for Flixster. The gain for cold start users is more than the gain for all users which we discussed in previous subsection. This implies that SocialMF handles cold start users better than STE. We believe this is mainly due to the consideration of trust propagation in our model.

6.5 Analysis of learning runtime

In section 4, we analyzed the runtime complexity and compared the complexity of computing the gradients of the objective function $\mathcal{L}$ in SocialMF and STE. In this section we compare the actual runtimes, conducting experiments on a Core2 Duo 2.16 GHz machine with Windows XP and 2 GB of memory.

Table 5 compares the actual runtime of a single iteration of the training phase for both models.

Note that according to our discussion in section 3.2, SocialMF should theoretically be $\frac{\bar{r}\bar{t}^2}{\bar{r}+\bar{t}^2}$ times faster than STE. Since the Flixster data set is denser than the Epinions data set, the improvement of runtime efficiency for Flixster is stronger than for Epinions.

Model	Epinions	Flixster
SocialMF	2.8 sec	29 sec
STE	37 sec	27 min

Table 5: Runtime comparison of a single iteration in training.

It should also be noted that the average number of iterations for SocialMF to converge is around 700, while the number of iterations for STE is around 550. Table 6 shows the total time required to learn the parameters for each model. As shown in this table, the STE model is much slower than SocialMF. The training phase of SocialMF is 40 times faster than that of the STE model for Flixster and more than 7 times faster for Epinions.

Model	Epinions	Flixster
SocialMF	40 min	5.5 hr
STE	5 hr	9 days

Table 6: Total time required to learn the parameters of models.

7. CONCLUSIONS AND FUTURE WORK

Recommender systems are emerging as tools of choice to select the online information relevant to a given user. Collaborative filtering is the most popular approach to building recommender systems and has been successfully employed in many applications. With the advent of online social networks, exploiting the information hidden in the social network to predict the behavior of users has become very important.

In this paper we proposed a novel model based approach for recommendation in social networks. Our model is a matrix factorization based approach. Similar to the STE model presented in [9], SocialMF learns the latent feature vectors of users and items. Different from STE, the feature vector of each user is dependent on the feature vectors of his direct neighbors in the social network.

This allows SocialMF to handle the transitivity of trust and trust propagation, which is not captured by the STE model. Trust propagation has been shown to be a crucial phenomenon in the social sciences, in social network analysis and in trust-based recommendation. Also if a user has not expressed any ratings, his feature vectors can be learnt as long as he is connected to the social network via a social relation. Thus SocialMF deals better with cold start users than existing methods. Note that if a cold start user is not connected to the social network, then social network based approaches have no additional information to improve the quality of recommendation for that user.

Experiments on two real life data sets from Epinions and Flixster demonstrate that SocialMF outperforms existing methods for social network based recommendation. This work suggests several interesting directions for future work. We want to extend the model to handle negative trust relations, since some social networks allow users to express distrust towards other users. In this paper, we performed experiments on different ranges of λ_T. As future work, we investigate the possibility of tuning λ_T automatically. Finally, cold start items have not been addressed in this paper, and it should be explored how the model can be extended so that the feature vectors of cold start items are also learnt efficiently.

8. REFERENCES

[1] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD 2008*.

[2] N. E. Friedkin. *A Structural Theory of Social Influence*. Cambridge University Press, 1998.

[3] J. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland College Park, 2005.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 1992.

[5] M. Jamali and M. Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *KDD 2009*.

[6] Y. Koren. Factorization meets the neighborhood a multifaceted collaborative filtering model. In *KDD 2008*, Las Vegas, USA.

[7] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42:30–37, 2009.

[8] Levien and Aiken. Advogato's trust metric. online at http://advogato.org/trust-metric.html, 2002.

[9] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR 2009*, pages 203–210.

[10] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM 2008*, pages 931–940. ACM, 2008.

[11] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys 2007*, USA.

[12] P. Massa and P. Avesani. Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(1), 2007.

[13] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD 2002*.

[14] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS 2008*, volume 20.

[15] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge Univ. Press, 1994.

[16] C. N. Ziegler. *Towards Decentralized Recommender Systems*. PhD thesis, University of Freiburg, 2005.