Collaborative Filtering with Social Exposure:
A Modular Approach to Social Recommendation

Menghan Wang, Xiaolin Zheng,[*] **Yang Yang**
College of Computer Science,
Zhejiang University, Hangzhou, 310027, China
{wangmengh, xlzheng, yangya}@zju.edu.cn

Kun Zhang
Department of Philosophy,
Carnegie Mellon University, Pittsburgh, USA
kunz1@cmu.edu

Abstract

This paper is concerned with how to make efficient use of social information to improve recommendations. Most existing social recommender systems assume people share similar preferences with their social friends. Which, however, may not hold true due to various motivations of making online friends and dynamics of online social networks. Inspired by recent causal process based recommendations that first model user exposures towards items and then use these exposures to guide rating prediction, we utilize social information to capture user exposures rather than user preferences. We assume that people get information of products from their online friends and they do not have to share similar preferences, which is less restrictive and seems closer to reality. Under this new assumption, in this paper, we present a novel recommendation approach (named SERec) to integrate social exposure into collaborative filtering. We propose two methods to implement SERec, namely *social regularization* and *social boosting*, each with different ways to construct social exposures. Experiments on four real-world datasets demonstrate that our methods outperform the state-of-the-art methods on top-N recommendations. Further study compares the robustness and scalability of the two proposed methods.

Introduction

Social recommendation has became a hot research area, and many social recommender systems have been proposed in recent years. Most existing social recommender systems assumed users who are connected are more likely to have similar preferences. This widely used assumption can be explained by social correlation theories such as social influence (Marsden and Friedkin 1993) and homophily (Mcpherson, Smithlovin, and Cook 2001). Under this assumption, social information is always utilized to better capture user latent preferences for collaborative filtering (CF), which is a popular technique used by recommender systems.

However, this assumption should be further examined because the increasing popularity of social media makes the situation more complex. The user characteristics change dynamically and why users interact in the social networks become diverse. It becomes easier to make friends today over
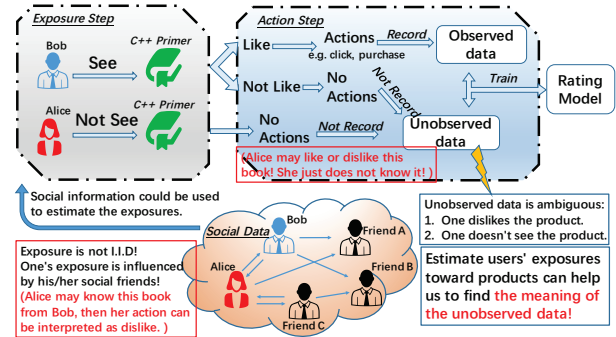
Figure 1: An example of user behaviors in a causal perspective.

the Internet and people make online friends for various reasons: e.g., alumni, colleagues, living in the same city, sharing similar interests. Online friends may not share similarities in preference. In fact, social network researchers have noticed similar phenomenon and studied dynamics of social networks such as information diffusion (Hu, Xu, and Shi 2015), social contagion (Yang et al. 2016), and social structural influence (Zhang et al. 2017). Clearly the "similar preference" assumption is too strong. To develop a more sensible recommender system, we make an attempt to explore the social information in a more natural way.

We come to the idea of utilizing social information on the exposure level rather than the preference level. Looking into the process of the user behaviors on websites, we can divide it in two steps: the *Exposure Step* and *Action Step*. In the *Exposure Step* users develop their own exposures on products. Then in the *Action Step* users take actions based on their exposures and preferences towards the products. That is to say, users have to see the products first, then they have the possibility to click or purchase the products based on their preferences. For example, in Figure 1, Bob has seen the *C++ Primer* book, then he takes actions if he likes it or does nothing if he does not like it. Alice does not see this book and does no action to the book. She may like and purchase this book if she has the opportunity to see this book. Note that in the model training, we only have observed data and unobserved data. We do not know which users have seen which products. So the unobserved

data is ambiguous with two meanings: 1) one does not like the product; 2) one does not know the product, which makes it hard to learning the user preference. One ideal approach is to model user exposure on each product so we can better capture user preference. Recent studies (Liang et al. 2016; Schnabel et al. 2016) mine exposures based on users' inner knowledge and cognitive bias. However, it is nature that user exposures are also influenced by their social friends. In reality social networks often provide plenty of opportunities for users to get information of products from their friends. So we assume that social information influences users on the exposure level, and then influences rating predictions by exposures. This idea gives a more detailed and sensible understanding of social recommendation: the social information is considered on the exposure level and is separated from the rating model. We can apply advanced social network analysis without constraints of rating models.

In this paper, we present a novel social recommendation approach, named "collaborative filtering with Social Exposure Recommendation" (SERec), to integrate social exposure into collaborative filtering. Different from traditional social recommendation which fuses the social matrix into the user-item matrix of the rating model, our SERec uses social information to model user exposures on items. The main contributions of this paper are listed as follows:

- Our SERec uses social information to estimate user exposures towards items, and then uses them to guide the rating prediction for recommendation. To the best of our knowledge, this is the first work to relax the "similar preference" assumption for social recommendation and exploit social information in a modular way.

- We propose two methods to implement SERec, namely *social regularization* and *social boosting*. *Social regularization* uses matrix factorization to model the social exposure, and utilizes the social information as a regularizer to constrain exposures between users and their friends. *Social boosting* makes use of a general function to compute the social exposure, which can be extend to advanced social network studies.

- We conduct extensive experiments on four public datasets to verify the efficacy of SERec. The experimental results demonstrate that SERec outperforms other baseline models. Further study on social exposure shows the effectiveness of social information to user's exposure. Meanwhile, we also compare the robustness and scalability of *social regularization* and *social boosting*.

Related Work

Social Recommendation utilizes social information to improve the performance of recommender systems. Existing social recommendation algorithms can be roughly divided into two categories: memory-based methods and model-based methods. Memory-based methods first obtain correlated users for a user and then aggregate ratings from the correlated users for the missing ratings (Tang, Hu, and Liu 2013). Researchers mainly focus on how to obtain the correlated users (e.g., MoleTrust (Massa and Avesani 2007), TrustWalker (Jamali and Ester 2009)). Model-based

methods often integrate social information into collaborative filtering for rating prediction. There are mainly two subgroups: ensemble methods (Ma, King, and Lyu 2009; Tang, Gao, and Liu 2012; Guo, Zhang, and Yorke-Smith 2015; Chaney, Blei, and Eliassi-Rad 2015) and regularization methods (Jamali and Ester 2010; Ma et al. 2011; Tang et al. 2013; Yang et al. 2017). The common rationale behind these methods is that users' preferences are similar to their friends. Different from all the above works, our work assumes users are influenced on the exposure level rather than the preference level by their friends due to the information spread by social networks.

Recommendation with causal inference is a newly emerging research area. It aims to alleviate the selection biases of training data for recommendation algorithms caused by the Missing Not At Random (MNAR) problem. Unlike traditional methods (Marlin and Zemel 2009; Ling et al. 2012; Hernández-Lobato, Houlsby, and Ghahramani 2014) that integrate rating prediction and missing data model into a unified Bayesian model with sophisticated approximate inference, causal inference based methods first compute exposures (or the propensity weights) for each user and then use them to guide rating prediction, which gives more appropriate description of the process as two separate modules. MF-IPS (Schnabel et al. 2016) proposes an empirical risk minimization approach to learning the unbiased estimators of user's preferences from biased rating data. ExpoMF (Liang et al. 2016) proposes a Bayesian model to capture propensity score by user exposure. These causal process based methods outperform the state-of-the-art traditional recommendation methods, but they do not take social information into consideration.

The Proposed Framework

Preliminary and Notation

First we describe some preliminary of our model and corresponding notation. Let $U = \{u_1, u_2, ..., u_n\}$ and $I = \{i_1, i_2, ..., i_m\}$ be the set of users and items respectively. For each user-item pair, we use α_{ui} to indicate whether user u has been exposed to item i, and use y_{ui} to indicate whether or not user u clicks item i. Following the definitions of (Liang et al. 2016), whether a user is exposed to an item follows from a Bernoulli. Conditional on the exposure variable α_{ui}, user's preference follows from a matrix factorization model, which computes y_{ui} by a multiplication of two latent factors: θ and β. The detailed notations are shown below:

$$
\begin{aligned}
\theta_u &\sim \mathcal{N}(0, \lambda_\theta^{-1} I_k), \\
\beta_i &\sim \mathcal{N}(0, \lambda_\beta^{-1} I_k), \\
\alpha_{ui} &\sim Bernoulli(\mu_{ui}), \\
p(y_{ui} &= 0 | \alpha_{ui} = 0) = 1, \\
y_{ui} | \alpha_{ui} &= 1 \sim \mathcal{N}(\theta_u^T \beta_i, \lambda_y^{-1}).
\end{aligned}
\tag{1}
$$

Where θ_u denotes user preference of user u, β_i denotes item attributes of item i; μ_{ui} is the prior probability of exposure; $\lambda_\theta, \lambda_\beta$, and λ_y are hyperparameters, denoting the inverse variances. $p(y_{ui} = 0 | \alpha_{ui} = 0) = 1$ means that when user u

Symbol	Description
y_{ui}	the rating user u gives to item i
θ_u	user preference vector of user u
β_i	item attribute vector of item i
α_{ui}	indicates whether user u has exposed to item i
μ_{ui}	prior probability of exposure variable α_{ui}
X_u	user exposure latent vector of user u
T_i	item exposure latent vector of item i
B_k	trustee specific vector of user k
γ_i	item exposure bias of item i
S	social connection matrix
$\Phi(S)$	social exposure function

Table 1: Notation

has not seen the item i, the probability for the user to click the item is zero. As we utilize social information to compute the exposure variable α_{ui}, we call it social exposure in this paper. We list the notation in Table 1.

SERec Framework

In the following, we describe our proposed framework, named "collaborative filtering with Social Exposure" Recommendation (SERec), which integrates social exposure into collaborative filtering for recommendation. In order to mine different aspects of social exposure, we propose two implementations of SERec and show their graphical representations in Figure 2. SERec has two main components: the *Rating Component* and the *Social Exposure Component*. The *Rating Component* is a graphical representation of equation (1). It is actually a matrix factorization model for rating prediction: the rating is predicted by taking a inner product of two latent vectors, i.e., $y_{ui} = \theta_u^T \beta_i$. The *Social Exposure Component* calculates the exposure priori μ_{ui} for each user-item pair. The social information is only used in this component. Unlike traditional social recommendations fusing social information into user latent factors, our SERec model utilizes social information to calculate the exposure variable and leave user latent factors θ_u "pure" – such factors are derived only from the rating information. This brings two merits: (1) The model is more scalable. In the *Rating Component* $\theta_u^T \beta_i$ contains essential information for recommendation; If needed, recent matrix factorization methods can be straightforwardly incorporated to improve recommendations, without interfering the *Social Exposure Component*. In the *Social Exposure Component* we can apply sophisticated social models with different assumptions and techniques as the only variable the social information can affect is μ_{ui}; (2) The modularity of SERec reduces the complexity of objective function, and the parameters are inferred into two steps. The separation of the two components makes parameter updates simple and easy to implement.

For the *Rating Component*, the log joint probability of exposures α_{ui} and click y_{ui} for user u and item i is

$$log\, p(\alpha_{ui}, y_{ui} | \mu_{ui}, \theta_u, \beta_i, \lambda_y^{-1})$$
$$= Bernoulli(\alpha_{ui} | \mu_{ui}) + \alpha_{ui} log \mathcal{N}(y_{ui} | \theta_u^T \beta_i, \lambda_y^{-1}) \quad (2)$$
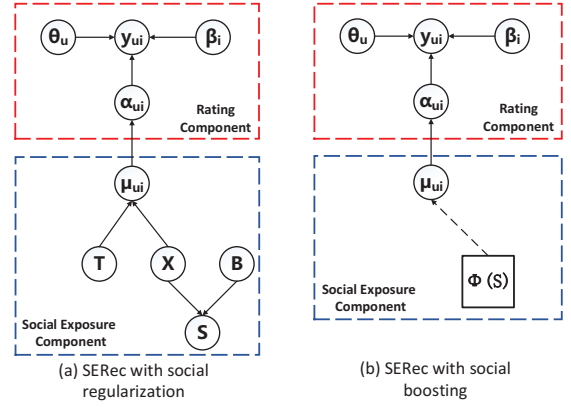$$+ (1 - \alpha_{ui}) log\, \mathrm{II}\, [y_{ui} = 0],$$

Figure 2: Two graphical models of SERec.

where $\mathrm{II}[b]$ is the indicator function which is 1 when b is true, and 0 otherwise. In the setting of a recommender system, we care about the prediction results of y_{ui} and equation (2) is the main objective function.

For the *Social Exposure Component*, we propose two methods to deal with the social information, namely *social boosting* and *social regularization*, each with different assumptions. The details of the two methods are shown in the next subsections.

Social Regularization

As regularization methods are effective in traditional social recommendation, a natural idea is to utilize social information as a regularizer in *Social Exposure Component*. Traditional social recommendation algorithms (Jamali and Ester 2010; Ma et al. 2011; Yang et al. 2017) assume friends share similarities in rating preferences and introduce social connections as regularization terms in user-item matrix factorizations. Differing from their ideas, we use user connections for social exposure constraints. Here we propose a novel matrix factorization based method to calculate μ with social information as regularization. The exposure prior is computed by $\mu_{ui} = X_u^T T_i + \gamma_i$, where X_u is the user exposure latent vector, T_i is the item exposure latent vector, and γ_i is the item bias. The social matrix S is mapped by two latent factors: truster-specific vector X and trustee-specific vector B. X and B characterize the behaviors of "to trust others" and "to be trusted by others". We set the X as a shared variable, so it has two folds of meanings: How a user trusts (or is affected by) others and how the same user has exposure on items. Therefore, we learn the latent factors by minimizing the following objective functions:

$$\mathcal{L}_{sr} = \sum_{ui}(X_u^T T_i + \gamma_i - \mu_{ui})^2 + \lambda_{sr} \sum_{uk}(X_u^T B_k - S_{uk})^2$$
$$+ \lambda_x \|X_u\|^2 + \lambda_t \|T_i\|^2 + \lambda_b \|B_k\|^2 + \lambda_\gamma \|\gamma_i\|^2, \quad (3)$$

where λ_{sr} controls the proportion; λ_x, λ_t, λ_b, and λ_γ are regularization parameters to avoid extreme values in the parameters. For those ui pairs such that $y_{ui} = 1$, we set

$\mu_{ui} = \frac{\#users\ who\ rated\ item\ i}{\#total\ users}$ as constraints. In this way, the estimated exposure component further takes social information into account by constraining X_u. For simplicity, we call this method $SERec_{regular}$.

Social Boosting

One potential drawback of $SERec_{regular}$ is that the matrix factorizations on social exposure prior and social matrix assume low rank decomposition structure, which may not well model practical situations. So we propose a more general method named "social boosting", which avoids this potential drawback and can be augmented with recent advanced social models. The intuition of social boosting is simple but reasonable: a consumer's knowledge of products is boosted by his/her friends. S/he gets information about products from friends' discussion and shared feelings. In other words, a consumer will have more exposure on a product if the consumer's friends have interacted with this product. We choose the Beta distribution $Beta(\alpha_1, \alpha_2)$ as the item-dependent conjugate prior for μ_{ui}. The general function of social boosting can be defined as follows:

$$\mu_{ui} = e_{ui} + \Phi(S), \qquad (4)$$

where e_{ui} is the inner exposure of user u toward item i; $\Phi(S)$ is the social exposure function that calculates the exposures user u get from his friends. e_{ui} can be learned by item popularity, text topics, or user locations (Liang et al. 2016). The $\Phi(S)$ serves as a plug-in function and it is flexible enough to meet different social situations. In this paper we use item popularity to calculate e_{ui} and set

$$\Phi(S) = \sum_{f \in Friends(u)} s \cdot \mu_{fi}, \qquad (5)$$

where s is the coefficient of social effects, $Friends(u)$ denotes the set of the friends of user u. For simplicity, we call this method $SERec_{boost}$.

Inference

We use expectation-maximization (EM) to find the maximum a posteriori of estimates of parameters θ_u, β_i, and μ_{ui}. Below we describe the details of each E step and M step.

E-step. We calculate the expected value of the following log likelihood function.

$$L = \sum_{u,i \in Y^-} log(\mu_{ui}\mathcal{N}(0|\theta_u^T\beta_i, \lambda_y^{-1}) + (1 - \mu_{ui}))$$
$$+ \sum_{u,i \in Y^+} log(\mu_{ui}\mathcal{N}(1|\theta_u^T\beta_i, \lambda_y^{-1})) \quad (6)$$
$$- \frac{\lambda_\theta}{2}\sum_u \theta_u^T\theta_u - \frac{\lambda_\beta}{2}\sum_i \beta_i^T\beta_i.$$

where Y^- denotes the set of unobserved user-item pairs, and Y^+ denotes the set of observed user-item pairs.

For the exposure variable α_{ui}, we compute the expectation $E[\alpha_{ui}]$ for every user-item pair. From the definition we

can see that $\alpha_{ui} = 1$ when the click action y_{ui} is observed. For those unobserved data, we apply the following equation:

$$E[\alpha_{ui}|y_{ui} = 0] = \frac{\mu_{ui}\mathcal{N}(0|\theta_u^T\beta_i, \lambda_y^{-1})}{\mu_{ui}\mathcal{N}(0|\theta_u^T\beta_i, \lambda_y^{-1}) + (1 - \mu_{ui})}. \quad (7)$$

α_{ui} belongs to a Bernoulli distribution, and y_{ui} is set to 0 when it is not observed. For convenience, we define $p_{ui} = E[\alpha_{ui}|y_{ui} = 0]$. We define $p_{ui} = 1$ if $y_{ui} = 1$.

M-step. In this step we aim to find the parameter that maximizes the objective log likelihood function in E-step. We use gradient descent to update the latent collaborative filtering factors θ_u and β_i:

$$\theta_u \leftarrow (\lambda_y \sum_i p_{ui}\beta_i\beta_i^T + \lambda_\theta I_K)^{-1}(\sum_i \lambda_y p_{ui}y_{ui}\beta_i), \quad (8)$$

$$\beta_i \leftarrow (\lambda_y \sum_u p_{ui}\theta_u\theta_u^T + \lambda_\beta I_K)^{-1}(\sum_u \lambda_y p_{ui}y_{ui}\theta_u), \quad (9)$$

Update for μ_{ui} with Social Regularization. Minimizing the objective functions $\mathcal{L}_{sr}$ needs enormous cost of computation. To speed up the learning procedure, we use stochastic gradient descent to learn the unknown parameters. At each step we randomly choose a triplet (i, u, k) and calculate the gradients for update as follows,

$$\frac{1}{2}\frac{\partial L}{\partial T_i} = (X_u^T T_i + \gamma_i - \mu_{ui})X_u + \lambda_t T_i,$$
$$\frac{1}{2}\frac{\partial L}{\partial X_u} = (X_u^T T_i + \gamma_i - \mu_{ui})T_i$$
$$+ \lambda_{sr}(X_u^T B_k - S_{uk})B_k + \lambda_x X_u, \quad (10)$$
$$\frac{1}{2}\frac{\partial L}{\partial B_k} = \lambda_{sr}(X_u^T B_k - S_{uk})X_u + \lambda_b B_k.$$
$$\frac{1}{2}\frac{\partial L}{\partial \gamma_i} = (X_u^T T_i + \gamma_i - \mu_{ui}) + \lambda_\gamma \gamma_i.$$

Then we repeat the process until a minimum is obtained. In practice, we find only updating μ with social regularization in one EM iteration could give comparable performance and save computation.

Update for μ_{ui} with Social Boosting. Maximizing the log likelihood with respect to μ_{ui} is equivalent to finding the mode of the complete conditional $Beta(\alpha_1 + \sum_{u'}^U p_{u'i} + (s-1)\sum_f^{Friends(u)} p_{fi}, \alpha_2 + U - \sum_{u'}^U p_{u'i})$, which is:

$$\mu_{ui} \leftarrow \frac{\alpha_1 + \sum_{u'}^U p_{u'i} + (s-1)\sum_f^{Friends(u)} p_{fi} - 1}{\alpha_1 + \alpha_2 + U + (s-1)\sum_f^{Friends(u)} p_{fi} - 2}, \quad (11)$$

where U is the number of users, $s \geq 1$ is the coefficient of social effects.

Experiments

Datasets and Settings

We consider four public datasets for experiments: *Lastfm*, *Delicious*, *Douban*, and *Epinions*. These datasets are widely experimented by social recommendations. The details of each dataset are listed in Table 3. Note that S-Impact =

Effectiveness of models								
Dataset	Metrics	baseMF	RSTE	TrustMF	WMF	ExpoMF	SERec$_{regular}$	SERec$_{boost}$
Lastfm	recall@10	0.0004	0.0045	0.0731	0.1206	0.1483	0.2117	**0.2159**
	recall@50	0.0013	0.0199	0.1898	0.3081	0.3357	0.3990	**0.4381**
	MAP@100	0.0001	0.0018	0.0208	0.0406	0.0366	0.0501	**0.0527**
	NDCG@100	0.0008	0.0131	0.1261	0.2385	0.2200	0.2937	**0.3102**
Delicious	recall@10	0.0001	0.0005	0.0085	0.0614	0.1818	0.1385	**0.1934**
	recall@50	0.0011	0.0008	0.0160	0.2859	**0.4448**	0.3927	0.4442
	MAP@100	7.3E-5	0.0001	0.0006	0.0331	0.0502	0.0434	**0.0511**
	NDCG@100	0.0005	0.0005	0.0077	0.1635	0.2501	0.2193	**0.2542**
Douban	recall@10	0.0001	0.0011	0.0810	0.1157	0.0964	0.1132	**0.1161**
	recall@50	0.0002	0.0034	0.1878	0.2255	0.2057	0.2302	**0.2425**
	MAP@100	5E-6	0.0003	0.0397	0.0405	0.0403	0.0440	**0.0489**
	NDCG@100	0.0002	0.0016	0.1265	0.1820	0.1840	0.1913	**0.2015**
Epinions	recall@10	0.0006	0.0008	0.0304	0.0385	0.0491	0.0568	**0.0618**
	recall@50	0.0033	0.0031	0.0801	0.1072	0.1302	0.1347	**0.1556**
	MAP@100	0.0002	0.0002	0.0041	0.0071	0.0068	0.0079	**0.0088**
	NDCG@100	0.0018	0.0016	0.0385	0.0545	0.0634	0.0701	**0.0777**

Table 2: Performance of different models on four datasets.

Dataset	Epinions	Delicious	Lastfm	Douban
Users (U)	32,424	1,867	1,892	129,490
Items (V)	61,274	69,223	17,632	58,541
Ratings (R)	664,824	104,799	92,834	16,830,839
R-Density	0.03%	0.08%	0.01%	0.22%
Social (S)	487,145	15,328	25,434	1,692,952
S-Density	0.05%	0.44%	0.71%	0.01%
avg.S	15.02	8.23	13.44	13.07
S-Impact	3.67%	0.67%	3.72%	2.91%

Table 3: Data statistics. U, V, R, S show the counts of each feature; R/S-Density indicates ratings/social links density; avg.S is average social links per user; S-Impact is the average ratio of items with which users' social friends have interacted to the whole item set.

$\frac{S}{U} \times \frac{R}{U \times V}$, which is a measurement of social impact by computing the average ratio of items with which users' social friends have interacted to the whole item set. We use Recall, MAP, and NDCG as our metrics. They are common metrics for recommendation and their definitions can be found in (Liang et al. 2016). To avoid data biases, we randomly select 70% of each dataset for training, 20% of each dataset for validation, and the remaining data for testing. To accelerate the experiments, we choose parallel programming and use C++ with OpenMP.

We then choose various prevalent methods for comparison, including: (1) BaseMF (Salakhutdinov and Mnih 2007), the baseline matrix factorization approach, widely applied as a benchmark; (2) WMF (Hu, Koren, and Volinsky 2008), a standard factorization model for implicit data, which uses a simple heuristic where all unobserved user-item interactions are equally down weighted against the observed interactions; (3) RSTE (Ma, King, and Lyu 2009), which fuses the users' tastes and their friends' favors in a unified framework; (4) TrustMF (Yang et al. 2017), which utilizes social information as a regularization for recommendation, as one of the state-of-the-art social recommendation models; (5) ExpoMF (Liang et al. 2016), a new probabilistic approach that directly incorporates user exposure to items into collaborative filtering, which does not utilize social information.

Top-N Recommendation

We use grid search to tune the parameters to achieve the best performance. We set $K = 10$ for BaseMF, $K = 10$ and $\alpha = 0.7$ for RSTE, $K = 30$ and $\lambda_{sr} = 1$ for TrustMF, $K = 10$ and $\alpha = 0.4$ for WMF, and $\lambda_\theta = 0.01, \lambda_\beta = 0.01, \lambda_y = 0.01$, and $K = 100$ for ExpoMF. For our methods, we use the above values for $\lambda_\theta, \lambda_\beta$, and λ_y and set $K = 20, K_{sr} = 30, \lambda_x = 1, \lambda_t = 1, \lambda_b = 1$, and $\lambda_{sr} = 5$ for SERec$_{regular}$ and set $s = 5$ for SERec$_{boost}$.

We show the experimental results in Table 2. We can see that SERec$_{regular}$ and SERec$_{boost}$ outperform other methods on all the four datasets, according to all the metrics. PMF has the worst performances because it only uses click data and treats the unobserved and observed data equally. RSTE outperforms PMF slightly since it ensembles social information into matrix factorization with a linear combination. Due to the sparsity problem in both ratings and social links, the performance of RSTE is still rather poor. TrustMF utilizes social information as a regularization and clearly outperforms RSTE, which proves the superiority of regularization methods. Though WMF does not utilize social information, it is superior to TrustMF as WMF treats unobserved user-item interactions with a low confidence. ExpoMF computes exposure latent variable α_{ui} for every user-item pair, and WMF is a special case of ExpoMF by constraining the ExpoMF's exposure variable α_{ui} to be binary, indicating whether the data is unobserved or observed. ExpoMF is inferior to our models and it does not use social information, which proves the superiority of our SERec models. On the other hand, SERec also outperforms other ways of using social information (RSTE and TrustMF). This verifies that it is essential to find a proper way to exploit social information.

Besides, SERec$_{boost}$ outperforms SERec$_{regular}$ on four datasets. We then compare our two methods by exploring how well the number of social friends influences the recommendation performance. We divide the users into four
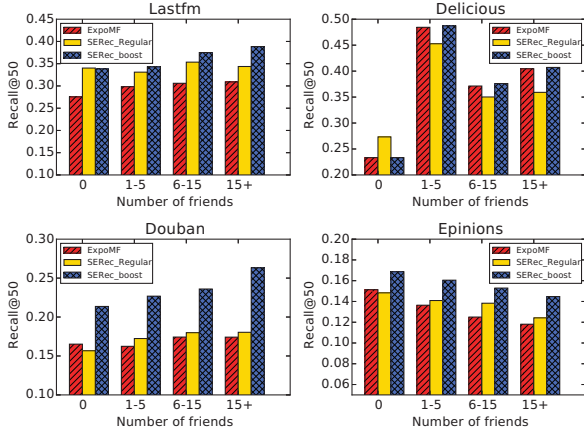
Figure 3: Performances of users with different number of friends.

Figure 4: Case study of two users: the left column for User A and the right for User B. The Top row shows the exposure prior μ and the bottom row shows the expectation of exposure variable α.

groups (namely, *0*, *1-5*, *6-15* and *15+*) based on how many friends they have. We compare the $recall@50$ among ExpoMF, SERec$_{boost}$, and SERec$_{regular}$ in Figure 3. ExpoMF serves as a baseline because it does not utilize social information. SERec$_{boost}$ outperforms ExpoMF especially for users who have more friends. This conforms to our expectation that users have more exposure when they have more friends (see equation (4)), which increases the possibility of click. Meanwhile, SERec$_{regular}$ performs more stable across the four groups. Compared to SERec$_{boost}$, we can see that SERec$_{regular}$ is inferior especially on group *6-15* and *15+*. There is an "exception" that in dataset *Delicious* SERec$_{regular}$ is worse than ExpoMF except for group 0. Note that RSTE and TrustMF also perform very poorly in *Delicious* compared to other datasets. Checking the data distribution of *Delicious* in Table 3, the average links of users is 8.23 and the number of items is 37.07 times that of users, which accounts for the low S-Impact of 0.67%. The S-Impact of *Delicious* is much smaller than that of the other datasets (3.67%, 3.72%, 2.91% respectively), indicating the "social quality" of *Delicious* is lower. This is also a sparsity problem: users' friends have few recorded interactions, and accordingly, utilizing them may limit the performance of recommendations. On the other hand, even though the social quality is poor, our methods still outperforms traditional social recommendation methods (RSTE and TrustMF), which proves our ways of utilizing social information.

Analysis of Social Exposure

In SERec, the rating behavior is controlled by the social exposure. We explore the effectiveness of social exposure variable α and its prior μ with two specific users in dataset Lastfm, namely, User A and User B. User A has 14 friends while User B has no friends. The reason of selection lies in that the avg.links is 13.44 and the social density is 0.71% (shown in Table 3). So User A and User B represent two typical kinds of users: users with average number of friends and users with few or no friends.
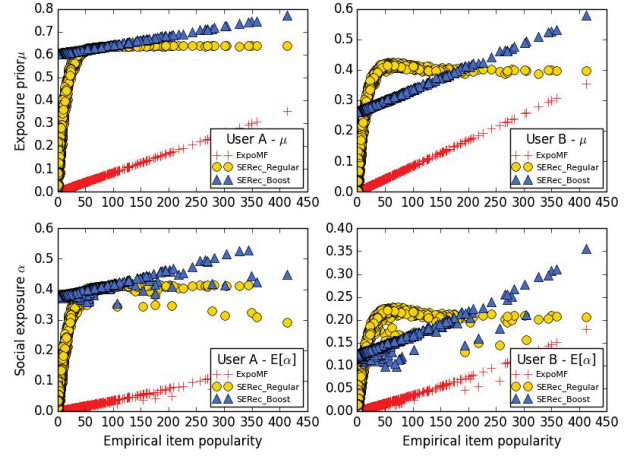
In the Figure 4, we draw the μ and $E[\alpha_{ui}]$ distributions for the two users. The ExpoMF only utilizes item popularity, so its μ curve is a "perfect" linear function of item popularity. And this μ curve is the same across all users, including User A and User B. For SERec$_{regular}$, there is a rapid increase of μ in Users A and B when the item popularity is smaller than 50. After that μ becomes stable. We can see the μ curve is always above that of ExpoMF, and the μ values of User A is much larger than those of User B. For SERec$_{boost}$, its μ curve is always above that of ExpoMF, and the two curves are parallel for User B, because User B has no friends and the increase of μ is caused by the increase of p_{ui} in equation (10). Meanwhile, the μ curve of User A is less steep than that of User B. This is because: when item popularity is small, $\sum_u^U p_{ui}$ is small and the social effect $(s-1)\sum_f^{Friends(u)} p_{fi}$ contributes more to the exposure; when item popularity increases, the $\sum_u^U p_{ui}$ increases and the effect of social information is weakened. Accordingly, for the users who have more friends, SERec$_{boost}$ has a larger promotion on their exposures than those who have few friends, and it is more effective for those items with small popularity. Compared to SERec$_{regular}$, SERec$_{boost}$ performs worse for User B when item popularity varies from 10 to 200. But for User A SERec$_{boost}$ always performs better. As User A has more friends than User B, such observations also demonstrate that SERec$_{boost}$ is more effective when users have more friends.

Consider the bottom panels of Figure 4, we can find that the shape of the curve $E[\alpha_{ui}]$ are similar to that of μ. As a high $E[\alpha_{ui}]$ only indicates a high possibility for user u to see the item i, it does not guarantee that the user will take actions (e.g., click, purchase) on this item. In other words, we can not infer specific recommendation results based only on the social exposure curves. In this section we can conclude that our SERec models improve the user exposure by utilization of social information. SERec$_{regular}$ and SERec$_{boost}$ perform differently due to their different mechanisms, and

Performances of SERec$_{regular}$ and SERec$_{boost}$ on the two datasets					
Lastfm	Metric	P$_s$ = 1	P$_s$ = 0.6	P$_s$ = 0.2	decay ratio
Regular.	Recall@10	0.2110	0.2071	0.1782	15.55%
	Recall@50	0.3971	0.3910	0.3233	18.58%
	MAP@100	0.0501	0.0484	0.0417	16.77%
	NDCG@100	0.2937	0.2824	0.2439	16.96%
Boost.	Recall@10	0.2142	0.1701	0.1553	27.49%
	Recall@50	0.4386	0.3933	0.3801	13.34%
	MAP@100	0.0527	0.0501	0.0407	22.77%
	NDCG@100	0.3102	0.2958	0.2225	28.27%
Delicious	Metric	P$_s$ = 1	P$_s$ = 0.6	P$_s$ = 0.2	decay ratio
Regular.	Recall@10	0.1385	0.0802	0.0701	49.39%
	Recall@50	0.3927	0.2398	0.2020	48.56%
	MAP@100	0.0434	0.0344	0.0286	34.10%
	NDCG@100	0.2193	0.1207	0.1007	54.08%
Boost.	Recall@10	0.1934	0.1693	0.1426	26.27%
	Recall@50	0.4442	0.3866	0.3739	15.83%
	MAP@100	0.0511	0.0429	0.0399	21.92%
	NDCG@100	0.2542	0.2406	0.2112	16.92%

Table 4: Analysis of robustness. P$_s$ is the proportion of used social information. Decay ratio means the decay from P$_s$ = 1 to P$_s$ = 0.2.

SERec$_{boost}$ generates higher exposures than SERec$_{regular}$ when users have more friends.

Comparison of Robustness and Scalability

As the *Rating Component* of our two methods is the same, here we focus more on the *Social Exposure Component*.

Robustness We now explore the robustness of SERec to different proportions of social information exploited. We randomly prune some social links between users and use the remaining links in our methods, and observe the effectiveness of recommendation performance. We select two datasets, *Lastfm* and *Delicious*, and show in Table 4 the performance of SERec$_{boost}$ and SERec$_{regular}$ with 100%, 60%, and 20% of social information utilized. *Lastfm* has a S-Impact of 3.72%, while *Delicious* is of low quality with a S-Impact of 0.67%. We also examine the robustness across different datasets.

In Table 4 we can see that the performance of SERec$_{boost}$ has almost a linear positive correlation with the percentage of utilized social information both in *Lastfm* and *Delicious*. And the decay ratio is about 20% in both datasets. SERec$_{boost}$ is an incremental method that utilizes the linear sum of one's social connections. It reduces to ExpoMF if no social information is available. But in SERec$_{regular}$ the performance varies across datasets. It has a low decay ratio around 17% in *Lastfm* and a high decay ratio around 50% in *Delicious*, due to the different S-Impact of two datasets. When the amount of utilized social information decreases, the effectiveness of regularization is weakened and the model may suffer the overfitting issue, accounting for the decrease in the performance of SERec$_{regular}$. To conclude, it is flexible for SERec$_{boost}$ to deal with different amount of social information, and its performance is more stable across different datasets. SERec$_{boost}$ is more robust than SERec$_{regular}$ to how much social information is used.

Scalability We analyze the scalability of SERec$_{boost}$ and SERec$_{regular}$ in three aspects. (1) **Time cost**: The time complexity of the *Rating Component* in SERec$_{regular}$ and SERec$_{boost}$ is the same. The cost is $O(UVt)$, where U is the number of users, V is the number of items, and t is the required iterations. For the *Social Exposure Component*, the complexity of SERec$_{regular}$ is $O(td(R+S))$, where d is the dimensionality of feature vectors, and R and S are the numbers of observed ratings and observed social links, respectively. The complexity of SERec$_{boost}$ is $O(UVf)$, where f is the average number of friends. Because of the severe sparsity problem shown in Table 2, we can see that $(R+S) <<$ UV. Besides, SERec$_{regular}$ uses stochastic gradient decent for inference. So we can conclude that SERec$_{regular}$ is more computationally efficient. Meanwhile, the above analysis assumes that our methods run in a sequential fashion, which makes $O(UV)$ a big overhead. But due to the independence of users in the EM procedure, our methods can be easily devised in a parallel form. In experiments we achieved a $30\times$ speed up with C++ and OpenMP. Table 5 shows the elapsed time for training SERec$_{regular}$ and SERec$_{boost}$, and SERec$_{regular}$ is indeed faster. (2) **Space cost**: SERec$_{regular}$ calculates μ_{ui} with a matrix factorization method, and the memory overhead is $O(Ud + Vd)$, where d ($d << U, V$) is the dimension of feature vectors. But SERec$_{boost}$ has to store μ_{ui} for every possible user-item pair. Accordingly its space cost is $O(UV)$, which will lead to a big memory overhead when there are millions of users and items. How to reduce its space cost is worth further exploring. Compared to SERec$_{boost}$, the cost of SERec$_{regular}$ is relatively small. (3) **Model extension**: Potential extensions of SERec$_{regular}$ may make use of alternative methods for matrix factorization. SERec$_{boost}$ does not rely on matrix factorization and can handle more sophisticated situations given that $\Phi(S)$ is properly defined. For example, in this paper we assume that all friends are equally close; we can find a way to measure the closeness among friends as closer friends tend to have more influence regarding item exposure. Social information based exposure may be improved by further resorting to recent social network analysis techniques (e.g., social contagion (Yang et al. 2016) and social structural influence (Zhang et al. 2017)). To summarize, SERec$_{boost}$ costs more space and time, but it performs better and allows straightforward extensions. SERec$_{regular}$ is more competitive in space and time cost.

Runtime Comparison (In Seconds)				
Method	Lastfm	Delicious	Douban	Epinions
Regular.	66 ± 0.66	234 ± 3.66	122,206 ± 33.0	2,339 ± 13.5
Boost.	124 ± 2.5	549 ± 5.0	330,431± 45.0	4,738 ± 19.0

Table 5: Runtime comparison of two methods of SERec.

Conclusion and Future Work

In this paper we proposed a novel approach to social recommendation, called SERec. It integrates social exposure into collaborative filtering for recommendation in a sensible, modular way. We utilized social information in two ways,

namely, social regularization and social boosting. Empirical results on four benchmark real-world datasets show that our methods clearly outperform alternatives. Further experimental results demonstrate the influence of social information on user exposures and the usefulness of such information for better recommendation. We also compared the robustness and scalability of the two proposed implementations of SERec. In the future, we will investigate how to leverage alternative social analysis models to better capture users' exposures. Another future line of research is to discover and make use of other sources of information that helps better model the process of users' decision making.

Acknowledgments

References

Chaney, A. J.; Blei, D. M.; and Eliassi-Rad, T. 2015. A probabilistic model for using social networks in personalized item recommendation. In *ReSys*, 43–50.

Guo, G.; Zhang, J.; and Yorke-Smith, N. 2015. Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *AAAI*, 123–129.

Hernández-Lobato, J. M.; Houlsby, N.; and Ghahramani, Z. 2014. Probabilistic matrix factorization with non-random missing data. In *ICML*, 1512–1520.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*, 263–272.

Hu, C.; Xu, W.; and Shi, P. 2015. Information diffusion in online social networks: models, methods and applications. In *WAIM*, 65–76.

Jamali, M., and Ester, M. 2009. Trustwalker : a random walk model for combining trust-based and item-based recommendation. In *KDD*, 397–406.

Jamali, M., and Ester, M. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, 135–142.

Liang, D.; Charlin, L.; Mcinerney, J.; and Blei, D. M. 2016. Modeling user exposure in recommendation. In *WWW*, 951–961.

Ling, G.; Yang, H.; Lyu, M. R.; and King, I. 2012. Response aware model-based collaborative filtering. In *UAI*, 501–510.

Ma, H.; Zhou, D.; Liu, C.; Lyu, M. R.; and King, I. 2011. Recommender systems with social regularization. In *WSDM*, 287–296.

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

Marlin, B. M., and Zemel, R. S. 2009. Collaborative prediction and ranking with non-random missing data. In *RecSys*, 5–12.

Marsden, P. V., and Friedkin, N. E. 1993. Network studies of social influence. *Sociological Methods & Research* 22(1):127–151.

Massa, P., and Avesani, P. 2007. Trust-aware recommender systems. In *RecSys*, 17–24.

Mcpherson, M.; Smithlovin, L.; and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27(1):415–444.

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *NIPS*, 1257–1264.

Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*, 1670–1679.

Tang, J.; Gao, H.; Hu, X.; and Liu, H. 2013. Exploiting homophily effect for trust prediction. In *WSDM*, 53–62.

Tang, J.; Gao, H.; and Liu, H. 2012. mtrust:discerning multifaceted trust in a connected world. In *WSDM*, 93–102.

Tang, J.; Hu, X.; and Liu, H. 2013. Social recommendation: a review. *SNAM* 3(4):1113–1133.

Yang, Y.; Jia, J.; Wu, B.; and Tang, J. 2016. Social role-aware emotion contagion in image social networks. In *AAAI*, 65–71.

Yang, B.; Lei, Y.; Liu, J.; and Li, W. 2017. Social collaborative filtering by trust. *TPAMI* 39(8):1633–1647.

Zhang, J.; Tang, J.; Zhong, Y.; Mo, Y.; Li, J.; Song, G.; Hall, W.; and Sun, J. 2017. Structinf: Mining structural influence from social streams. In *AAAI*, 73–80.