Scalable Recommendation with Social Contextual Information

Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, *Fellow, IEEE*, and Shiqiang Yang, *Senior Member, IEEE*

Abstract—Exponential growth of information generated by online social networks demands effective and scalable recommender systems to give useful results. Traditional techniques become unqualified because they ignore social relation data; existing social recommendation approaches consider social network structure, but social contextual information has not been fully considered. It is significant and challenging to fuse social contextual factors which are derived from users' motivation of social behaviors into social recommendation. In this paper, we investigate the social recommendation problem on the basis of psychology and sociology studies, which exhibit two important factors: individual preference and interpersonal influence. We first present the particular importance of these two factors in online behavior prediction. Then we propose a novel probabilistic matrix factorization method to fuse them in latent space. We further provide a scalable algorithm which can incrementally process the large scale data. We conduct experiments on both Facebook style bidirectional and Twitter style unidirectional social network data sets. The empirical results and analysis on these two large data sets demonstrate that our method significantly outperforms the existing approaches.

Index Terms—Social recommendation, individual preference, interpersonal influence, matrix factorization

1 Introduction

Social network users generate large volumes of information, which makes it necessary to exploit highly accurate recommender systems to assist them in finding useful results. Traditional collaborative filtering techniques do not consider social relations, making them difficult to provide accurate recommendations [1]. Recently, Ma et al. [2], [3] proposed a framework of social recommender systems that made use of social relation data, from which friendship information is exploited to regularize the user latent space. However, in this work, the social contextual information was not fully considered. It is significant and challenging to discover social contextual factors from the contextual information and integrate them into a unified recommendation framework.

Fig. 1 shows the entire social contextual information which can be derived from links on social networks. Users typically examine items' content and information on senders. For example, in Twitter, when a user receives a tweet that is posted by one of his friends (the sender), he usually reads its content to see whether the item is interesting. We can get this knowledge from *item content* and *user-item interaction* information. In this case, the user cares about who the sender is and whether the sender is a close friend or authoritative. If more than one friend sends him the same tweet, he may read it more attentively. This knowledge can be learnt from *social relation* and *user-user interaction* information. Both of these aspects are important for the user to decide whether to adopt (e.g., share, retweet) the item. The above can be summarized as two contextual factors: (1) *individual preference* and (2) *interpersonal influence*.

Besides the experiential assumptions, psychological and sociological studies have proved that individual preference and interpersonal influence affect users' decisions on information adoption. In Bond's work [4], it is indicated that individuals are to some extent influenced by others' behaviors, rather than making decisions independently (i.e., purely preference driven). In [5], web-based experiments are designed for music adoption prediction. This work demonstrates that the introduction of interpersonal influence into the preference-driven decision process (as is the case in real social networks) makes user behaviors more complicated and thus increases the unpredictability of the item adoption. Therefore, only when individual preference and interpersonal influence are properly incorporated into recommendation, can the uncertainty be reduced and quality improved.

To address this problem, we propose a social contextual recommendation framework as shown in Fig. 2. This framework is based on a probabilistic matrix factorization method to incorporate individual preference and interpersonal influence to improve the accuracy of social recommendation. More specifically, we factorize the user-item interaction matrix into two intermediated latent matrices including user-item influence matrix and user-item preference matrix, which are generated from three objective latent matrices: user latent feature matrix, item latent feature matrix, and user-user influence matrix. Moreover, as we can partially observe individual preference and interpersonal influence based on previous user-item and user-user interaction data, we further utilize the observed social contextual factors to compute the three objective latent matrices. Furthermore, we provide a scalable algorithm, wherein the

- M. Jiang, P. Cui, W. Zhu and S. Yang are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: mjiang89@gmail.com, {cuip, wwzhu, yangshq}@tsinghua.edu.cn.
- F. Wang is with Healthcare Analytics Research Group, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598. E-mail: feiwang03@gmail.com.

Manuscript received 1 Aug. 2013; revised 31 Dec. 2013; accepted 2 Jan. 2014. Date of publication 15 Jan. 2014; date of current version 26 Sept. 2014. Recommended for acceptance by J. Pei.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2300487

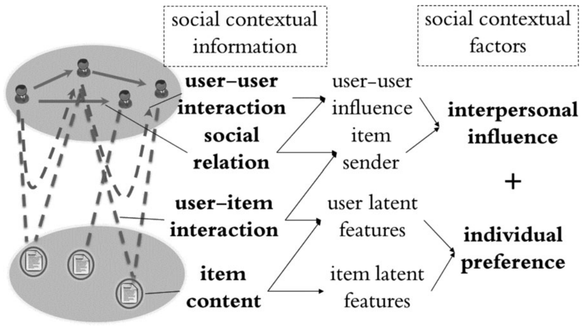


Fig. 1. A novel framework for social recommendation: it understands mechanism of user behavior on social networks, fully utilizes contextual information, and summarizes the knowledge as two social contextual factors.

time cost is linear to the size of recommended items and users, to incrementally process the large scale data so that it can achieve goal of the scalable recommendation and be used in large real applications.

We have conducted experiments on two real social network data sets. One is collected from Renren (www.renren.com), a Facebook style website in China; and the other is collected from Tencent Weibo (t.qq.com), a Twitter style website in China. The two data sets represent two typical social network structures: one for bidirectional social relations (mutual friends), and the other for unidirectional social relations (followers and followees). It is shown that social contextual factors can greatly boost the performance of recommender systems on social network data, and our method outperforms the previous algorithms by a large margin. We attribute this great performance to the incorporation of complete social contextual factors from both individual and interpersonal sides, which has been verified by the experiments.

This paper is organized as follows. In Section 2, we give introduction to relevant work. In Section 3, we illustrate the effectiveness of two contextual factors with studies on social data sets. In Section 4, we show the formulation on social contextual model. Section 5 is about the experimental result and key insights of social contextual factors on social recommendation problem. Section 6 comes to the conclusion.

2 Related Works

In this section, we review several major approaches to recommendation methods. Content-based filtering and collaborative filtering have been widely used to help users find out the most valuable information. With the emergence of social networks, researchers have designed trust-based and influence-based methods to take advantage of the knowledge from user relationships. Matrix factorization methods have been proposed for social recommendation due to their efficiency in dealing with large data sets. Although there are some mixture models of these methods, it is valuable to understand the mechanism in social recommendation problems and make the most use of social contextual information from the perspective of users' motivation of item adoption.

Content-based filtering introduces the basic idea of studying the item content for the ranking problem. With the emergence of topic modeling techniques like *LDA* [6], recent content-based approaches [7], [8], [9], [10] rank candidate items by how well they match the topic interest of the user as their preference. These approaches working on individual patterns is not able to learn user behavior patterns from user-item interaction data.

Collaborative filtering methods, which consists of memory-based and model-based methods, are widely used. The memory-based approaches [11], [12], [13], [14] calculate the similarity between all users based on their ratings of items. The model-based methods learn a model based on patterns recognized in the ratings of users. Liu et al. [15] build a model-based collaborative-filtering framework with three layers (user-interests-item) to help personalized ranking on recommender systems. Collaborative filtering only utilizes user-item

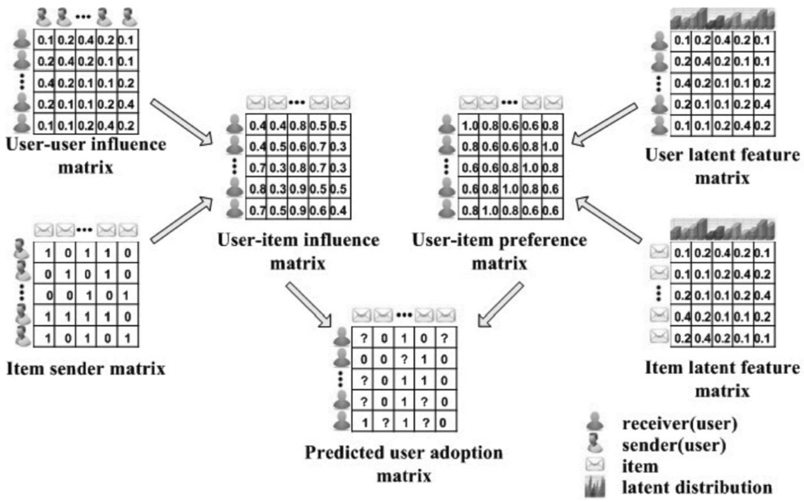
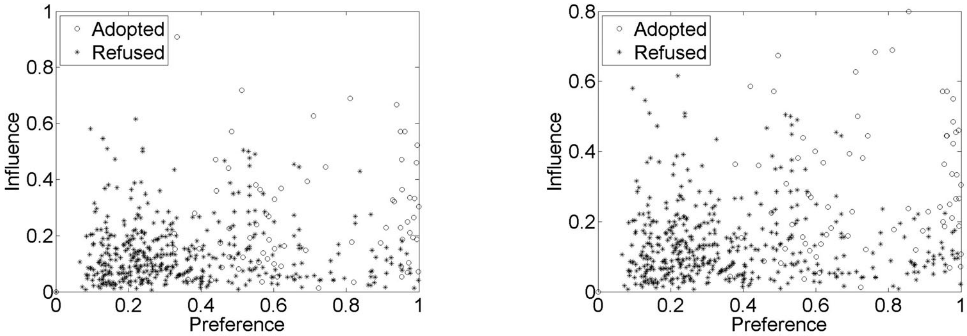


Fig. 2. Our social contextual recommendation model based on a probabilistic matrix factorization method: it incorporates interpersonal influence and individual preference. Sender is the user who generates an item (e.g., post, retweet, etc.). Receiver connects to the sender and thus receives the item.



(a) User behaviors on Renren social network (b) User behaviors on Tencent Weibo microblogging service

Fig. 3. Distribution of two contextual factors of user behaviors in (a) Renren and (b) Tencent Weibo: the adoption behaviors usually have higher individual preference and interpersonal influence than the refusion behaviors.

interaction information, but it is not able to make full use of social relation and rich social knowledge including user profiles and detailed item content.

Matrix factorization methods [16], [17], [18], [19], [20] have been proposed recently. The matrix approximation models all focus on representing the user-item rating matrix with low-dimensional latent vectors [21], [22], [23], [24], [25]. Recognizing that influence is a subtle force that governs the dynamics of social networks, influence-based recommendation [26], [27], [28], [29] involves interpersonal influence into social recommendation cases [30], [31], [32]. Trust-based approaches [33], [34], [35], [36] exploit the trust network among users and make recommendation based on the ratings of users who are directly or indirectly trusted. $SoRec$ [2] is proposed as a probabilistic factor analysis framework which fuses the users' tastes and their trusted friends' favors together. Aiming at improving recommender systems by incorporating users' social network information into both friend network and trust network, Ma et al. [3] propose a matrix factorization framework with social regularization. But this work only constrains users' individual features from interpersonal side but ignores users' individual side, which makes the framework lack of complete contextual information to further improve the performance. However, it is still an open issue about what factors motivate user adoption on recommended items and how they can be effectively integrated to further improve the accuracy.

From psychological and sociological views, Bandura [37] gives a social cognitive theory of mass communication and argues that communication systems operate through two pathways. In the direct pathway, they promote changes by motivating and guiding participants to get what they prefer. In the socially-mediated pathway, participants' decisions are influenced by their friendship networks. Benjamin [38] shows the similar opinion that factors such as cognition, feeling, taste, interest and interpersonal relationship develop the structure of social behaviors and interactions. For social web, these two factors exactly represent individual preference and interpersonal influence. That motivates us to propose a social contextual recommendation framework to incorporate them by analyzing both user motivation and application mechanism to recommender systems for social networks. In our paper, we incorporate both individual preference and interpersonal influence in a principled manner.

3 Social Contextual Factors

In this section, we will demonstrate the existence and significance of social contextual factors (including individual preference and interpersonal influence) for social recommendation on real large data sets.

Given an item, the behavior of user adoption depends on individual preference to understand whether the user likes the item or not. Interpersonal influence tells whether the user has close relationships with the item senders (e.g., followees who post the tweet in Twitter). Based on previous data, we apply LDA to the content of web post (e.g., tweet) and extract topic-level distributions of these items. According to user behavior history, we summarize how much user u likes item a with a naïve preference measurement as

$$P_u(a) = T_a \cdot \left(\frac{1}{|A(u,a)|} \sum_{a' \in A(u,a)} T_{a'} \right),$$

where $A(u,a)$ is the set of items adopted by user u excluding a, and T_a is the topic distribution of item a.

To describe interpersonal influence from the perspective of user-user interactions on social web, we calculate the percentage of recommended items adopted by u from u's friends or followees who send the item a:

$$I_u(a) = \frac{1}{|V(u,a)|} \sum_{v \in V(u,a)} \frac{|\mathcal{S}(u,v) \cap \mathcal{A}(u)|}{|\mathcal{S}(u,v)|},$$

where $V(u,a)$ is the set of senders who send item a to user u, $\mathcal{S}(u,v)$ is the set of items sent from v to u, and $\mathcal{A}(u)$ is the set of items that u adopts.

We classify the items into "adopted" ones and "refused" ones according to user behaviors, and plot the pairs (u, a) as points, w.r.t., individual preference $P_u(a)$ and interpersonal influence $I_u(a)$ in Fig. 3, which shows that users intend to adopt items with better preference scores and from higher influential friends or followees in Facebook or Twitter style networks.

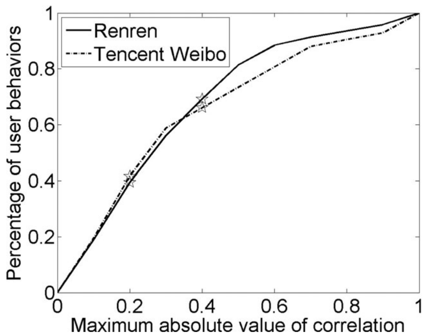


Fig. 4. The two contextual factors have little correlation: the absolute values of correlation between preference and influence is smaller than 0.2 for more than 40 percent cases, and smaller than 0.4 for more than 70 percent cases.

In order to demonstrate that individual preference and interpersonal influence are not only effective but also complementary social contextual factors, we compute their correlations in social recommendation cases. We use P and I to denote preference and influence of a user's adopted item. The Pearson correlation is defined as

$$\rho_{P,I} = \frac{cov(P,I)}{\sigma_P \sigma_I} = \frac{E[(P-\mu_P)(I-\mu_I)]}{\sigma_P \sigma_I}.$$

The correlation is 1 or -1 in the case of perfect positive or negative linear relationship, and zero if preference and influence are uncorrelated. In Fig. 4, the absolute correlation values of more than 40 percent users are less than 0.2 and the values of around 70 percent are less than 0.4. Thus we conclude that individual preference and interpersonal influence can be applied as two complementary social contextual factors in recommendation.

4 Model

4.1 Social Contextual Model *ContextMF*

In this section, we introduce details of our social contextual model based on matrix factorization ($ContextMF$). First, we formally define the problem of social recommendation. Suppose that we have M users with the ith user denoted as u_i, and N items with the jth item denoted as p_j. We denote the information adoption matrix as $\mathbf{R} \in \{0,1\}^{M\times N}$, with its (i,j)th entry

$$R_{ij} = \begin{cases} 1 & \text{if } u_i \text{ adopted } p_j \\ 0 & \text{otherwise.} \end{cases}$$

Then the social recommendation problem is converted to predict the unobserved entries in the information adoption matrix $\mathbf{R}$ based on the observed entries and other factors.

In our model, we suppose that whether a user adopts an item on social networks is determined by three aspects: (1) item content: what the item tells about, (2) user-item interaction: what items the user likes, and (3) social relation and user-user interaction: who the senders are.

Let $\mathbf{U} \in \mathbb{R}^{k\times M}$ be the latent user feature matrix, $\mathbf{V} \in \mathbb{R}^{k\times N}$ be the latent item feature matrix. $\mathbf{S} \in \mathbb{R}^{M\times M}$ is the interpersonal influence matrix, with each entry S_{ij} representing the degree of influence user u_i has on user u_j. It should be noted that $S_{ij} > 0$ if and only if u_i is the friend of u_j in social networks such as Facebook and Renren, or is followed by u_j in microblogging services such as Twitter and Tencent Weibo. $\mathbf{G} \in \mathbb{R}^{N\times M}$ is the item sender matrix, with entry $G_{ij} = 1$ meaning that u_j sends the item p_i and vice versa. Based on these denotations and the assumption that users can only receive items from their friends as social networks usually do ($G_{ii} = 0$), we can see the social recommendation problem is to find out $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$ so that $((\mathbf{S}\mathbf{G}^\top) \odot (\mathbf{U}^\top \mathbf{V}))$ can well approximate the observed entries in $\mathbf{R}$ without over-fitting, where $\odot$ is the Hadamard Product.

In our case, we know the item content, user behaviors over the items, and the interactions between users. From these previous data, we can derive the item content representation, individual preference, and interpersonal influence. We compute the user-user preference similarity matrix $\mathbf{W} \in \mathbb{R}^{M\times M}$, item-item content similarity matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$, and user-user interaction matrix $\mathbf{F} \in \mathbb{R}^{M\times M}$ as

$$\begin{aligned} W_{i,j} &= \frac{\sum_{a\in\mathcal{A}(u_i)} P_{u_i}(a)}{|\mathcal{A}(u_i)|} \cdot \frac{\sum_{a'\in\mathcal{A}(u_j)} P_{u_j}(a')}{|\mathcal{A}(u_j)|}, \\ C_{i,j} &= T_{a_i} \cdot T_{a_j}, \\ F_{i,j} &= \frac{|\mathcal{S}(u_i,u_j) \cap \mathcal{A}(u_i)|}{|\mathcal{S}(u_i,u_j)|}. \end{aligned}$$

Though the accuracy of similarity matrices $\mathbf{W}$ and $\mathbf{C}$ depends on how LDA performs on previous data, it is fair towards competing methods in experiments to share knowledge from these matrices.

With the hypothesis that the similarities in observed spaces are consistent with the latent spaces, we regularize the three latent spaces by observed matrices (social contextual factors) in that: (1) the users that are similar in user latent space $\mathbf{U}$ have similar preferences (derived from preference similarity matrix $\mathbf{W}$); (2) the items that are similar in item latent space $\mathbf{V}$ have similar descriptive contents (derived from content similarity matrix $\mathbf{C}$); (3) high interpersonal influence in the influence latent space $\mathbf{S}$ generates frequent interpersonal interactions $\mathbf{F}$; (4) the product of user latent space $\mathbf{U}$ and item latent space $\mathbf{V}$ corresponds to the users' individual preference on the items; (5) the Hadamard product of interpersonal influence and individual preference is proportional to the probability of item adoptions.

As the model performance is evaluated by root mean square error (RMSE) on the test set, we adopt a probabilistic linear model with Gaussian observation noise. Here we define the conditional distribution over the observed entries in $\mathbf{R}$ as

$$P(\mathbf{R}|\mathbf{S},\mathbf{U},\mathbf{V},\sigma_R^2) = \prod_{i=1}^{M}\prod_{j=1}^{N} \mathcal{N}(\mathbf{R}_{ij}|\mathbf{S}_i\mathbf{G}_j^\top \odot \mathbf{U}_i^\top \mathbf{V}_j, \sigma_R^2).$$

By incorporating the social contextual factors, we define the posterior distribution as

$$
\begin{aligned}
&P(\mathbf{S},\mathbf{U},\mathbf{V}|\mathbf{R},\mathbf{G},\mathbf{W},\mathbf{C},\mathbf{F},\Omega)\\
&\quad=\frac{P(\mathbf{R},\mathbf{W},\mathbf{C},\mathbf{F},\mathbf{G}|\mathbf{S},\mathbf{U},\mathbf{V},\Omega)P(\mathbf{S},\mathbf{U},\mathbf{V}|\Omega)}{P(\mathbf{R},\mathbf{G},\mathbf{W},\mathbf{C},\mathbf{F},\Omega)}\\
&\quad\propto P(\mathbf{R}|\mathbf{S},\mathbf{U},\mathbf{V},\Omega)P(\mathbf{W}|\mathbf{U},\Omega)P(\mathbf{C}|\mathbf{V},\Omega)P(\mathbf{F}|\mathbf{S},\Omega)\\
&P(\mathbf{S}|\Omega)P(\mathbf{U}|\Omega)P(\mathbf{V}|\Omega)\\
&\quad=\prod_{i,j}\mathcal{N}\big(R_{ij}|\mathbf{S}_i\mathbf{G}_j^\top\odot\mathbf{U}_i^\top\mathbf{V}_j,\sigma_R^2\big)\\
&\quad\prod_{p,q}\mathcal{N}\big(W_{pq}|\mathbf{U}_p^\top\mathbf{U}_q,\sigma_W^2\big)\prod_{m,n}\mathcal{N}\big(C_{mn}|\mathbf{V}_m^\top\mathbf{V}_n,\sigma_C^2\big)\\
&\quad\prod_{s,t}\mathcal{N}\big(F_{st}|S_{st},\sigma_F^2\big)\prod_{x}\mathcal{N}\big(\mathbf{S}_x|0,\sigma_S^2\big)\\
&\quad\prod_{y}\mathcal{N}\big(\mathbf{U}_y|0,\sigma_U^2\big)\prod_{z}\mathcal{N}\big(\mathbf{V}_z|0,\sigma_V^2\big),
\end{aligned}
$$

where Ω denotes that zero-mean spherical Gaussian priors are placed on latent feature vectors and observed matrices. Then,

$$
\begin{aligned}
&\ln P(\mathbf{S},\mathbf{U},\mathbf{V}|\mathbf{R},\mathbf{G},\mathbf{M},\mathbf{C},\mathbf{F},\Omega)\\
&\quad\propto-\frac{1}{2\sigma_R^2}\sum_{i,j}\big(R_{ij}-\mathbf{S}_i\mathbf{G}_j^\top\odot\mathbf{U}_i^\top\mathbf{V}_j\big)^2\\
&\quad-\frac{1}{2\sigma_W^2}\sum_{p,q}\big(W_{pq}-\mathbf{U}_p^\top\mathbf{U}_q\big)^2-\frac{1}{2\sigma_C^2}\sum_{m,n}\big(C_{mn}-\mathbf{V}_m^\top\mathbf{V}_n\big)^2\\
&\quad-\frac{1}{2\sigma_F^2}\sum_{s,t}(F_{st}-S_{st})^2-\frac{1}{2\sigma_S^2}\sum_{x}\mathbf{S}_x^\top\mathbf{S}_x\\
&\quad-\frac{1}{2\sigma_U^2}\sum_{y}\mathbf{U}_y^\top\mathbf{U}_y-\frac{1}{2\sigma_V^2}\sum_{z}\mathbf{V}_z^\top\mathbf{V}_z.
\end{aligned}
$$

Maximizing the posterior distribution is equivalent to minimizing the sum-of-squared errors function with hybrid quadratic regularization terms

$$
\begin{aligned}
\mathcal{J}=&\|\mathbf{R}-\mathbf{S}\mathbf{G}^\top\odot\mathbf{U}^\top\mathbf{V}\|_F^2+\alpha\|\mathbf{W}-\mathbf{U}^\top\mathbf{U}\|_F^2\\
&+\beta\|\mathbf{C}-\mathbf{V}^\top\mathbf{V}\|_F^2+\gamma\|\mathbf{F}-\mathbf{S}\|_F^2\\
&+\delta\|\mathbf{S}\|_F^2+\eta\|\mathbf{U}\|_F^2+\lambda\|\mathbf{V}\|_F^2,
\end{aligned}
$$

where $\alpha=\frac{\sigma_R^2}{\sigma_W^2},\beta=\frac{\sigma_R^2}{\sigma_C^2},\gamma=\frac{\sigma_R^2}{\sigma_F^2},\delta=\frac{\sigma_R^2}{\sigma_S^2},\eta=\frac{\sigma_R^2}{\sigma_U^2},\lambda=\frac{\sigma_R^2}{\sigma_V^2}$, and $\|.\|_F$ is the Frobenius norm.

We can adopt a block coordinate descent scheme to solve the problem. That is, starting from random initialization on $\mathbf{S},\mathbf{U},\mathbf{V}$, we solve each of them alternatively with the other two matrices fixed and proceed step by step until convergence. As the objective is obviously lower bounded by 0 and the alternating gradient search procedure will reduce it monotonically, the algorithm is guaranteed to be convergent. In this paper, we use the gradient search method to solve the problem. Specifically, the gradients of the objective with respect to the variables are

$$
\begin{aligned}
\frac{\partial\mathcal{J}}{\partial\mathbf{S}}&=-2\big(\mathbf{R}-\mathbf{S}\mathbf{G}^\top\odot\mathbf{U}^\top\mathbf{V}\big)\mathbf{G}\\
&\quad-2\gamma(\mathbf{F}-\mathbf{S})+2\delta\mathbf{S},\\
\frac{\partial\mathcal{J}}{\partial\mathbf{U}}&=-2\mathbf{V}\big(\mathbf{R}-\mathbf{S}\mathbf{G}^\top\odot\mathbf{U}^\top\mathbf{V}\big)^\top\\
&\quad-4\alpha\mathbf{U}\big(\mathbf{W}-\mathbf{U}^\top\mathbf{U}\big)+2\eta\mathbf{U},\\
\frac{\partial\mathcal{J}}{\partial\mathbf{V}}&=-2\mathbf{U}\big(\mathbf{R}-\mathbf{S}\mathbf{G}^\top\odot\mathbf{U}^\top\mathbf{V}\big)\\
&\quad-4\beta\mathbf{V}\big(\mathbf{C}-\mathbf{V}^\top\mathbf{V}\big)+2\lambda\mathbf{V}.
\end{aligned}
$$

Thus, we apply the following gradient-based approach to our social contextual model in Algorithm 1. $\mathcal{J}$ decreases the fastest in the direction of gradients during each iteration and the sequence $(\mathcal{J}^{(t)})$ converges to the desired minimum.

Algorithm 1 Social Contextual Model $ContextMF$

Require: $0<\alpha_S^{(t)},\alpha_U^{(t)},\alpha_V^{(t)}<1$, $t=0$. Initialization $\mathcal{J}^{(0)}=\mathcal{J}(\mathbf{S}^{(0)},\mathbf{U}^{(0)},\mathbf{V}^{(0)})$.
Ensure: $\mathcal{J}^{(0)}\ge 0$, $\mathcal{J}^{(t+1)}<\mathcal{J}^{(t)}$
  for $t=1,2,\cdots$ **do**
    Calculate $\frac{\partial\mathcal{J}}{\partial\mathbf{S}}^{(t-1)},\frac{\partial\mathcal{J}}{\partial\mathbf{U}}^{(t-1)},\frac{\partial\mathcal{J}}{\partial\mathbf{V}}^{(t-1)}$
    $\mathbf{S}^{(t)}\leftarrow\mathbf{S}^{(t-1)}-\alpha_S^{(t-1)}\cdot\frac{\partial\mathcal{J}}{\partial\mathbf{S}}^{(t-1)}$
    $\mathbf{U}^{(t)}\leftarrow\mathbf{U}^{(t-1)}-\alpha_U^{(t-1)}\cdot\frac{\partial\mathcal{J}}{\partial\mathbf{U}}^{(t-1)}$
    $\mathbf{V}^{(t)}\leftarrow\mathbf{V}^{(t-1)}-\alpha_V^{(t-1)}\cdot\frac{\partial\mathcal{J}}{\partial\mathbf{V}}^{(t-1)}$
    $\mathcal{J}^{(t)}\leftarrow\mathcal{J}(\mathbf{S}^{(t)},\mathbf{U}^{(t)},\mathbf{V}^{(t)})$
  end for

4.2 Model for Incremental Data Δ*ContextMF*

Our model can be applied in the real system to deal with incremental data, answering the following questions. First, how can we recommend items to new users with social relations and their previous behaviors? Second, how can we recommend new items to users with items' content and historical data? We give an incremental processing version $\Delta ContextMF$ based on $ContextMF$ to solve these two problems. It updates the influence matrix and latent feature matrices from the relationships between the increments and old matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$.

If ΔM new users come. Suppose we know the interactions and similarities between M users and the ΔM new users, we aim at learning the influence matrix $\Delta\mathbf{S}\in\mathbb{R}^{\Delta M\times M}$ and new users' latent feature matrix $\Delta\mathbf{U}\in\mathbb{R}^{k\times\Delta M}$. Let $\Delta\mathbf{F}\in\mathbb{R}^{\Delta M\times M}$ be the given incremental interaction matrix. $\Delta\mathbf{W}\in\mathbb{R}^{\Delta M\times M}$ is the incremental user-user similarity matrix. We obtain the objective functions $\mathcal{J}_{\Delta S}$ and $\mathcal{J}_{\Delta U}$ and their gradients to learn $\Delta\mathbf{S}$ and $\Delta\mathbf{U}$. Note that we ignore the high-order terms in the functions because of their small scales

$$
\begin{aligned}
\mathcal{J}_{\Delta S}&=\|\Delta\mathbf{F}-\Delta\mathbf{S}\|_F^2, &\frac{\partial\mathcal{J}}{\partial\Delta\mathbf{S}}&=-2\Delta\mathbf{F}+O(\Delta\mathbf{S})\\
\mathcal{J}_{\Delta U}&=\|\Delta\mathbf{W}-\Delta\mathbf{U}^\top\mathbf{U}\|_F^2, &\frac{\partial\mathcal{J}}{\partial\Delta\mathbf{U}}&=-2\mathbf{U}\Delta\mathbf{W}^\top+O(\Delta\mathbf{U}).
\end{aligned}
$$

Therefore, the predicted item adoption matrix $\Delta\mathbf{R}\in\mathbb{R}^{M\times\Delta N}$ can be computed as $\Delta\mathbf{R}=\Delta\mathbf{S}\mathbf{G}^\top\odot\Delta\mathbf{U}^\top\mathbf{V}$.

TABLE 1
Complexity Comparison (Suppose $M \gg \Delta M$, $N \gg \Delta N$)

	Incremental processing $\Delta ContextMF$	Offline recommendation $ContextMF^{\Delta}$
ΔM users	$O(k^2 L \Delta M M)$	$O(k^2 L M(M+N))$
ΔN items	$O(k^2 L \Delta N N)$	$O(k^2 L N(M+N))$

If ΔN new items come. Let $\mathbf{\Delta G} \in \mathbb{R}^{\Delta N \times M}$ be the given incremental item sender matrix. $\mathbf{\Delta C} \in \mathbb{R}^{\Delta N \times N}$ is the incremental item-item similarity matrix: the topic-level similarity between N items and the ΔN new items, that learnt from topic distributions of item content. We obtain $\mathcal{J}_{\Delta V}$ and the gradient to learn the incremental item latent feature matrix $\mathbf{\Delta V} \in \mathbb{R}^{k \times \Delta N}$,

$$\mathcal{J}_{\Delta V} = \|\mathbf{\Delta C} - \mathbf{\Delta V}^{\top}\mathbf{V}\|_F^2, \frac{\partial \mathcal{J}}{\partial \mathbf{\Delta V}} = -2\mathbf{V}\mathbf{\Delta C}^{\top} + O(\mathbf{\Delta V}).$$

Therefore, the predicted item adoption matrix $\mathbf{\Delta R} \in \mathbb{R}^{M \times \Delta N}$ can be computed as $\mathbf{\Delta R} = \mathbf{S}\mathbf{\Delta G}^{\top} \odot \mathbf{U}^{\top}\mathbf{\Delta V}$.

Meanwhile, offline recommendation $ContextMF^{\Delta}$ that merges new users/items into old ones can be applied. If ΔM new users come, $ContextMF^{\Delta}$ needs in each iteration $O(k^2(M+\Delta M)^2)$ to update $\mathbf{S}$ and $\mathbf{U}$, and $O(k^2(M+\Delta M)N)$ to update $\mathbf{V}$. However, $\Delta ContextMF$ needs only $O(k^2\Delta M(M+\Delta M))$ to compute $\mathbf{\Delta S}$ and $\mathbf{\Delta U}$. It is similar for the case of new items. $\Delta ContextMF$ outperforms $ContextMF^{\Delta}$ on both time and memory efficiency (see Table 1, in which L is number of iterations). Note that the complexity of incremental processing is *linear* to the size of users/items, while the existing systems [2], [3] usually cost *quadratic* time. In Section 5.7, with experimental results, we show significant recommendation performance of our incremental processing model $\Delta ContextMF$. Thus we demonstrate that our method has the capability of incremental processing on new data.

5 EXPERIMENTS

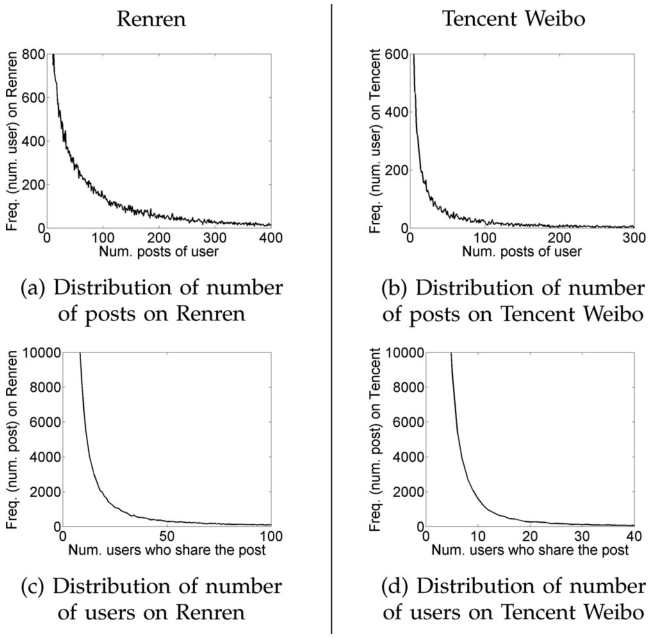
5.1 Data Sets Description

We conduct experiments on two large real data sets: Renren and Tencent Weibo. The statistics are summarized in Table 2. The density of Renren data set is 0.59 percent and the density of Tencent Weibo data set is 0.09 percent. The sparsity problem is typically serious in our case.

We collected data from Renren, a typical social networking service that enables users to put on their profiles and add friends. One of the most popular actions on Renren is to share blogs, photos and external video links (denoted as items in the paper). As an item is shared by a user, the item will be sent to the user's friends and appear in their pages in real time. We crawled relationships and shared items of nearly 1 million users from February 2007 to December 2009.

TABLE 2
Statistics of Data Sets

	Renren	Tencent Weibo
Num. users (M)	939,363	163,661
Num. items (N)	1,625,689	529,615
Num. adoption behaviors	5,829,368	1,566,609



(a) Distribution of number of posts on Renren
(b) Distribution of number of posts on Tencent Weibo
(c) Distribution of number of users on Renren
(d) Distribution of number of users on Tencent Weibo

Fig. 5. Long-tailed distributions of number of users' posts (a), (b) and number of users who share a post (c), (d) .

Meanwhile, we crawled data from Tencent Weibo, which allows users to follow and receive messages from other users. Like Twitter, it also enables users to spread information by retweeting the messages. We crawled tweets, retweets and user relationships from more than 100,000 accounts in January 2011.

To further demonstrate the problem caused by the sparsity of data, we take a look at some statistical analysis of Renren and Tencent Weibo data. In Figs. 5a and 5b, we plot the distribution of the number of users' shared or forwarded posts (calculated by $\sum_i G_{ij}$ for user u_j). In Figs. 5c and 5d, we plot the distribution of the number of users who share or forward them (calculated by $\sum_j G_{ij}$ for post p_i). We can see that all the four figures follow long-tailed distributions, which reflects that adoption behavior made by the majority of users over the majority of items is sparse on social networks.

5.2 Experimental Settings

We design our experiments with two typical tasks of recommendation [39]: (1) predict user behaviors; (2) rank received items. The first task requires the recommender to predict whether a specific user will adopt a specific item. Therefore, an appropriate measure is prediction error (smaller is better). The second task requires a more direct focus on actual recommendation and provides users with a ranked list of received items, along with predictions for how much the users would like them. We consider different ranking-based measures to show how successfully we put the most favorite items on the top of list (bigger is better). We will introduce all the measurements in the next section. Here we focus on experimental settings which should be fair for all comparative algorithms.

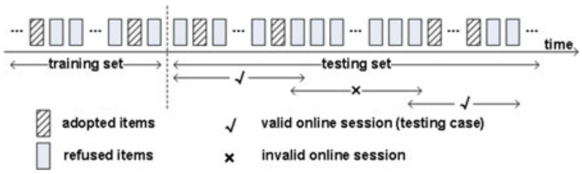


Fig. 6. Valid online sessions as testing cases for experiments: first we divide the data set into training and testing parts; second we select online sessions from the testing part with a window (Δt); at last we conduct experiments on sessions within at least two adopted items.

Different from held-out data experiments on data sets without time information, the recommendation on social items, e.g., tweets, should be evaluated in a temporal setting. The first reason is that users could make *different* decisions on the same items in *different* contexts. A user's desire to adopt or refuse an item when he receives it at time t_1 may be different from that when he receives the item at a later time t_2, if his friends or followers share or retweet the item during the time $\Delta t = t_2 - t_1$. The second reason is that our experiments demand both positive (user adopts item) and negative (user refuses item) instances of user behaviors. It is easy to detect the positive instances according to user adoption behaviors over items, but the negative instances cannot be simply detected. There are two categories among the items that users do not adopt. First, users were not online, thus did not read these items. Second, users read these items and refused them. Only the latter category could be considered as the negative instances. For the above reasons, we define *online sessions* for the duration that the user is online and active on this social networking service. We suppose the user reads all the items received from his relationships during this session.

Given a user, a valid online session should have these three properties: (1) the session length should be within Δt_{max} (like 5 minutes); (2) in this session the user should receive at least n_{min} items (like 15) from his friends or followees; (3) among the items in this session, the user adopts (shares or forwards) at least two items. Δt_{max} and n_{min} are called *online-session parameters*. In other words, in an online session as we defined, the user receives a number of messages in a short time and gives at least two positive behaviors (adoption). In Fig. 6, we show the difference between valid and invalid online sessions and how we use valid online sessions as testing cases in our experiments. We conduct all baseline algorithms and our method $ContextMF$ on those testing cases. Although it cannot be guaranteed that the estimation of online session is perfect, it is fair for all comparative algorithms to use the experimental settings. We demonstrate the advantages of our method $ContextMF$ if it can better accomplish the two tasks.

5.3 Comparative Algorithms

We implement the following baselines for comparison with our social contextual model ($ContextMF$).

- $ContentBased$ [7]. This method recommends similar items with ones that the receiver has shared or forwarded before. It only considers individual preference and utilizes item content information instead of social relation and interaction data.
- $ItemCF$ [11]. The standard item-based collaborative filtering assumes that users have common interests with their close friends or followers. It only utilizes user-item interaction information.
- $FeedbackTrust$ [34]. This method improves the basic trust-based recommendation algorithm [33] with feedback. It is accurate to compute user correlations in trust network, but it only utilizes user-user interactions information.
- $InfluenceBased$ [27]. This method estimates influence as social utility based on a gradient ascent algorithm. It uses information of user-user interaction, but fails to discover individual correlations between users and items.
- $SoRec$ [2]. This method jointly analyzes social relation and user-item interaction data by extracting a common latent factor from the shared mode, using *probabilistic matrix factorization*. User-user interaction information is not considered.
- $SoReg$ [3]. This method designs a matrix factorization objective function with *Social Regularization* to constraint user features. It does not consider item content information, which truly builds the users' individual preference. Both user and item features should be regularized with respect to the understanding of item content.

Meanwhile, we implement different configurations of our model to demonstrate the effectiveness of our proposed $ContextMF$.

- $InfluenceMF$. This method considers interpersonal influence, one of the social contextual factors in our social recommendation model. The adjusted function to minimize is

$$\mathcal{J} = \|\mathbf{R} - \mathbf{S}\mathbf{G}^\top\|_F^2 + \gamma\|\mathbf{S} - \mathbf{F}\|_F^2 + \delta\|\mathbf{S}\|_F^2.$$

- $PreferenceMF$. This method only considers individual preference. The degenerated function is

$$\mathcal{J} = \|\mathbf{R} - \mathbf{U}^\top\mathbf{V}\|_F^2 + \alpha\|\mathbf{W} - \mathbf{U}^\top\mathbf{U}\|_F^2 + \beta\|\mathbf{C} - \mathbf{V}^\top\mathbf{V}\|_F^2 + \eta\|\mathbf{U}\|_F^2 + \lambda\|\mathbf{V}\|_F^2.$$

5.4 Evaluation Measures

Generally, we evaluate recommendation performance of each algorithm with four typical groups of measurements: (1) prediction error: how accurately the algorithm works to predict user behaviors (for task 1); (2) top K performance: how successfully the algorithm offers top K recommendation service (for task 2); (3) ranking-based measure: how well the algorithm performs to rank items (also for task 2); (4) stability measure: how resistantly the gradient-based algorithm performs on the same piece of data for 100 times.

Prediction error. To measure the prediction quality of our proposed approach in comparison with other algorithms, we use two popular metrics including the mean absolute error (MAE) and the root mean square error (RMSE). The metric MAE is defined as

$$MAE = \frac{1}{|\mathcal{R}|} \sum_{R_{ij} \in \mathcal{R}} |\mathbf{R_{ij}} - \mathbf{S_i}\mathbf{G_j}^\top \odot \mathbf{U_i}^\top \mathbf{V_j}|,$$

where R_{ij} equals 1 if the ith user adopts the jth item and 0 if not. The metric RMSE is defined as

$$RMSE = \sqrt{\frac{1}{|\mathcal{R}|} \sum_{R_{ij} \in \mathcal{R}} (\mathbf{R_{ij}} - \mathbf{S_i}\mathbf{G_j}^\top \odot \mathbf{U_i}^\top \mathbf{V_j})^2}.$$

Therefore, a smaller MAE or RMSE value means better performance.

Top K performance. Compared to the prediction accuracy of user behaviors, the top K recommendation performance is also important because the recommendation space on page is limited and only the top K recommended items make sense in real applications. In an online session, each algorithm provides a list of K recommended items. We use Precision@K [11], [40] and NDCG@K [41] to evaluate the performance.

The precision at rank K (Precision@K) is defined as the ratio of adopted items in the K-length recommended list to K. Therefore, a higher Precision@K shows better performance. NDCG is a normalization of the Discounted Cumulative Gain (DCG) measure. DCG is a weighted sum of "relevancy" of the ranked items. The DCG at rank K (DCG@K) for a give session is computed as:

$$DCG@K = \sum_{r=1}^{K} \frac{y(r)}{log(r+1)},$$

where the "relevancy" of the item at rank r is $y(r)$ and the logarithmic discount is $\frac{1}{log(1+r)}$. The "relevancy" $y(r)$ is a mapping from the item's rank to a finite set $\mathcal{Y} = \{0, 1\}$, where 1 corresponds to an adoption behavior, i.e., the rth item is shared or forwarded by the user, and 0 corresponds to the opposite. The Ideal DCG (IDCG) is simply the maximum value of DCG results, i.e., DCG measure of the best ranking result. NDCG normalizes DCG by IDCG and thus it is always a number in $[0, 1]$. A bigger number of NDCG@K is better for the algorithm to provide a top K recommendation service.

Ranking-based measure. We use two kinds of ranking-based measures: (1) $\hat{\tau}$ and $\hat{\rho}$, which are Kendall's and Spearman's ranking coefficients to measure order accuracy; (2) ERR, which is suggested by Sanderson et al. [42] as one of the best measurements for ranking problem of user preference. All these four measures share the same property that a bigger number means better performance.

The ranking coefficients $\hat{\tau}$ and $\hat{\rho}$ start by defining this intuitive statistics, that is, the number of ranking order switches, which means how many of the pairs in the testing case are ordered incorrectly by the model

$$T = \sum_{r<s} I(y(r) > y(s)),$$

where (r, s) is a pair of orders of ranked items, and $I(x)$ is a mapping function that returns 1 if event x is true and 0 if x is false. $y(r)$ has defined formerly as a relevancy function. The weighted sum of order switches, which weighs the incorrect ordered pairs by the ranking difference

$$R = \sum_{r<s} (s - r) \cdot I(y(r) > y(s)).$$

These two measures are then transformed linearly into the range [−1,1], where 1 corresponds to perfect model performance and −1 corresponds to the worst case, thus attaining perfect reverse ranking. We have the non-parametric correlation prevalent data analysis tools here

$$\hat{\tau} = 1 - \frac{4T}{n(n-1)},$$
$$\hat{\rho} = 1 - \frac{12R}{n(n-1)(n+1)},$$

where the number of items to rank is n.

Chapelle et al. [43] propose expected reciprocal rank (ERR), a metric for graded relevancy based on a cascade model, and demonstrate that this metric is better than DCG, modeling user satisfaction. It can be computed as follows:

$$ERR = \sum_{r=1}^{n} \frac{1}{r} \prod_{i=1}^{r-1} (1 - y(i)) y(r).$$

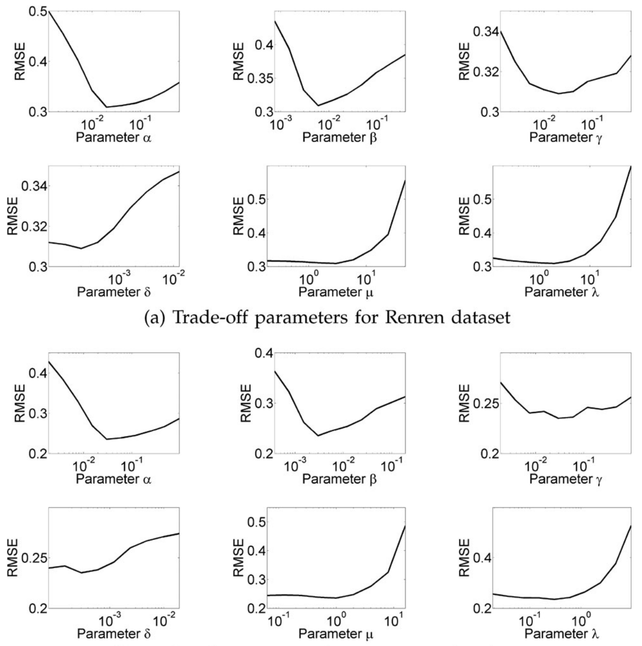
From the equation, we know that the value of ERR is in range $[0, 1]$, and an algorithm returns a better ranking list of items if its ERR result is bigger.

Stability measure. An algorithm that gives higher probabilities on adopted items than refused items will better help the recommender systems. In order to demonstrate the distinguish ability, we conducted a group of T-tests to compare numerical gaps between good recommendations and bad ones for each method. When the value of T-test is bigger, the classification is more accurate on whether user adoption will happen.

5.5 Parameter Settings

The parameters in our model are meaningful and necessary but not difficult to set. We tune the parameters of our $ContextMF$ model and all baseline algorithms to reach their best performance. In this condition, both the experiment settings and parameter settings are fair so that the experimental results are reasonable. Moreover, we introduce how to easily and automatically set appropriate parameters for our $ContextMF$ model from insights of their tuning processes.

Trade-off parameters. The trade-off parameters α, β, γ, δ, η and λ in our model are supposed to adjust the strengths of different terms in the objective function: (1) α and β regularize the terms of latent features of users and items with user-user similarity and item-item similarity on topical distribution, so they determine the weight of individual preference in our recommendation model; (2) γ regularizes the term of user-user influence with their interaction frequency, so this parameter determines the weight of interpersonal influence; (3) δ, η and λ make sure that the scales of user-user influence



(a) Trade-off parameters for Renren dataset

(b) Trade-off parameters for Tencent Weibo dataset

Fig. 7. We tune and find the best settings of trade-off parameters α, β, γ, δ, η and λ in our $ContextMF$ model. We use the same training data, control for a single parameter, plot the RMSE curve and choose the least.

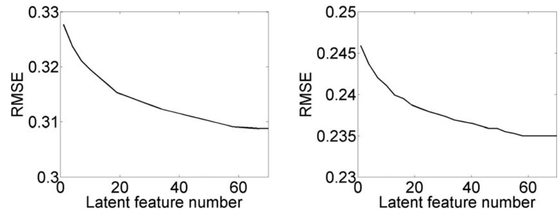
$\mathbf{S}$, user latent features $\mathbf{U}$ and item latent features $\mathbf{V}$ change the objective function little in our model.

We tune trade-off parameters α, β, γ, δ, η and λ for our $ContextMF$ model with the "Controlling for a variable" method to reach the best performance. As shown in Fig. 7, the RMSE can be reduced to the minimum when the parameters are neither too big nor too small. We suggest a way of automatic parameter settings for social recommendation model $ContextMF$ as insights from this observation

$$\begin{aligned}
&\alpha \leftarrow 10^{-2} \times \frac{\|\mathbf{R} - \mathbf{S}\mathbf{G}^\top \odot \mathbf{U}^\top\mathbf{V}\|_F^2}{\|\mathbf{W} - \mathbf{U}^\top\mathbf{U}\|_F^2} \propto 10^{-2} \times \frac{N}{M},\\
&\beta \leftarrow 10^{-2} \times \frac{M}{N}, \gamma \leftarrow 10^{-2} \times \frac{N}{M},\\
&\delta \leftarrow 10^{-4} \times \frac{\|\mathbf{R} - \mathbf{S}\mathbf{G}^\top \odot \mathbf{U}^\top\mathbf{V}\|_F^2}{\|\mathbf{S}\|_F^2} \propto 10^{-4} \times \frac{N}{M},\\
&\mu \leftarrow 10^{-4} \times \frac{N}{k}, \lambda \leftarrow 10^{-4} \times \frac{M}{k},
\end{aligned}$$

where M and N are the number of users and items (see Table 2) and k is the number of latent features. We set $k = 60$ in this process while we introduce how to determine k as follows.

Number of latent features. We train $\mathbf{U}$, $\mathbf{V}$ to find the proper number of latent features k for users and items. If k is too small, the recommender system cannot make a distinction between any users or items. If k is too large, users and items will be too unique for the system to calculate their similarities and the complexity will considerably increase. Therefore, we conduct experiments with k ranging from 3 to 80 on both Renren and Tencent Weibo data sets. The results are shown in Fig. 8, from which we can find that with the latent feature number k increasing, RMSE reduces gradually. It obviously shows that when $k \geq 60$, RMSE decreases rather slow. Considering the recommendation effect and time efficiency, we choose $k = 60$ as the latent space dimension in our experiments.

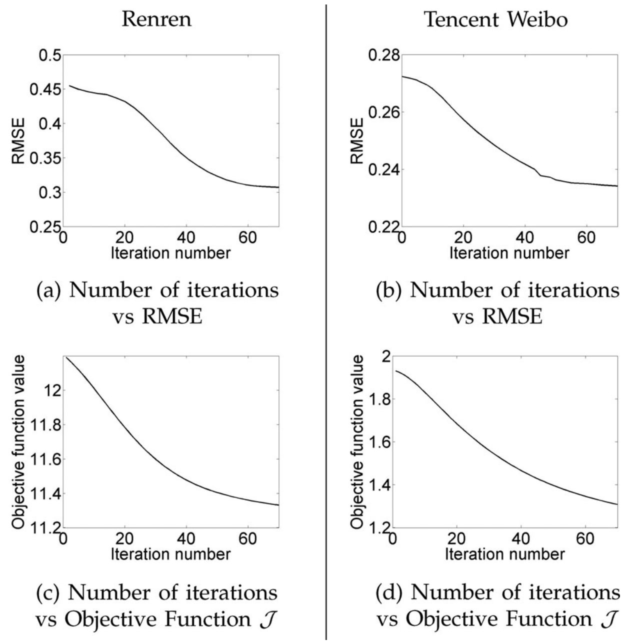


(a) Renren (b) Tencent Weibo

Fig. 8. RMSE decreases with number of latent features k and converges when the number is 60 in (a) Renren and (b) Tencent Weibo data sets.

Number of iterations. In Fig. 9, we observe that both RMSE and the objective function value $\mathcal{J}$ decrease gradually with the number of iterations increasing. It shows that, by incorporating effective regularizers, our method successfully avoids the overfitting problem which often occurs in gradient algorithms. On both data sets, it is better to run 60 iterations in order to reach a converged result with an acceptable time cost. We tune parameters following the same method as above in our $ContextMF$ model for $PreferenceMF$ and $InfluenceMF$. For other comparative algorithms, we also search for the best configurations while applying for our real data sets. It is fair to report their best recommendation performance in the next section.

Online-session parameters. Fig. 10 shows the standard variance of RMSE (σ_{RMSE}) according to the length of time window Δt_{max} and the number of adopted items n_{min}, after we conduct the experiments for 100 times. If Δt_{max} is too big, users may have stopped the session and been offline; if it is too small, the user has little time to adopt items. If



(a) Number of iterations vs RMSE

(b) Number of iterations vs RMSE

(c) Number of iterations vs Objective Function $\mathcal{J}$

(d) Number of iterations vs Objective Function $\mathcal{J}$

Fig. 9. RMSE (a), (b) and Objective Function Value $\mathcal{J}$ (c), (d) decrease with the number of iterations and converge at around 60 in Renren and Tencent Weibo.

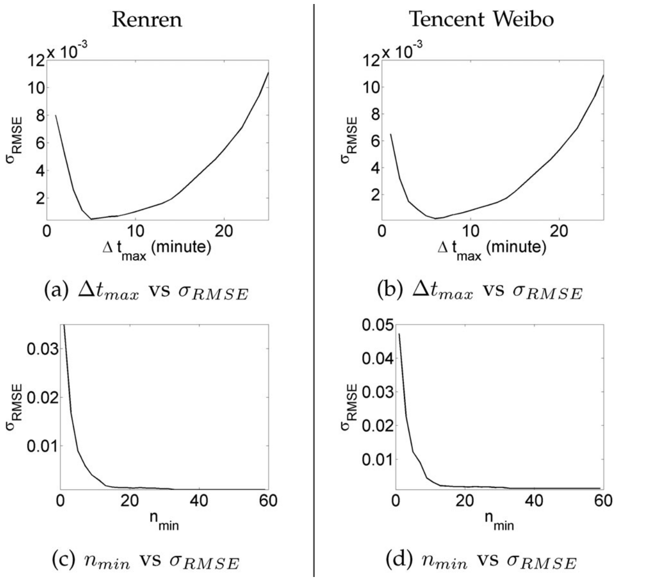


Fig. 10. We choose Δt_{max} and n_{min} that determine *online session* for the least standard variance of RMSE.

n_{min} is too small, the user is also inactive in the time windows; if it is too big, the data set is held out too much. Therefore, we choose values of the two parameters when σ_{RMSE} reaches the least. In our experiments, we choose $\Delta t_{max} = 5$ mins and $n_{min} = 15$.

5.6 Recommendation Performance

We judge recommendation performance of the mentioned models and algorithms in three ways: (1) performance on user behavior prediction; (2) top K recommendation; (3) stability.

First, we evaluate our $ContextMF$ model and comparative algorithms with the measurements including precision errors (MAE and RMSE), ranking coefficients ($\hat{\tau}$, $\hat{\rho}$ and ERR) and stability measure (T-test). As shown in Table 4, our social contextual model recommends items based on matrix factorization algorithm with social contextual factors. It provides reasonably accurate recommendations that are much better than baselines. On Renren and Tencent Weibo data sets, we decrease the MAE by 19.1 and 12.8 percent, RMSE by 24.2 and 20.7 percent, and increase ERR by 19.7 and 11.4 percent over $SoReg$, a state-of-the-art social recommendation algorithm with social regularization. $ContextMF$ improves the recommendation performance with large margins over both $PreferenceMF$ and $InfluenceMF$: on Renren and Tencent Weibo, it decreases MAE by 25.2 and 39.7 percent, RMSE by 21.7 and 31.5 percent; it increases Kendall's ranking coefficient by 12.1 and 2.27 percent, Spearman's ranking coefficient by 12.2 and 6.04 percent, ERR by 46.5 and 31.6 percent. All these numbers prove that if a social recommendation model considers both contextual factors (individual preference and interpersonal influence), it outperforms the version that only considers one of them. To compare the distinguish-ability of our method with baselines, we report average value of prediction of positive and

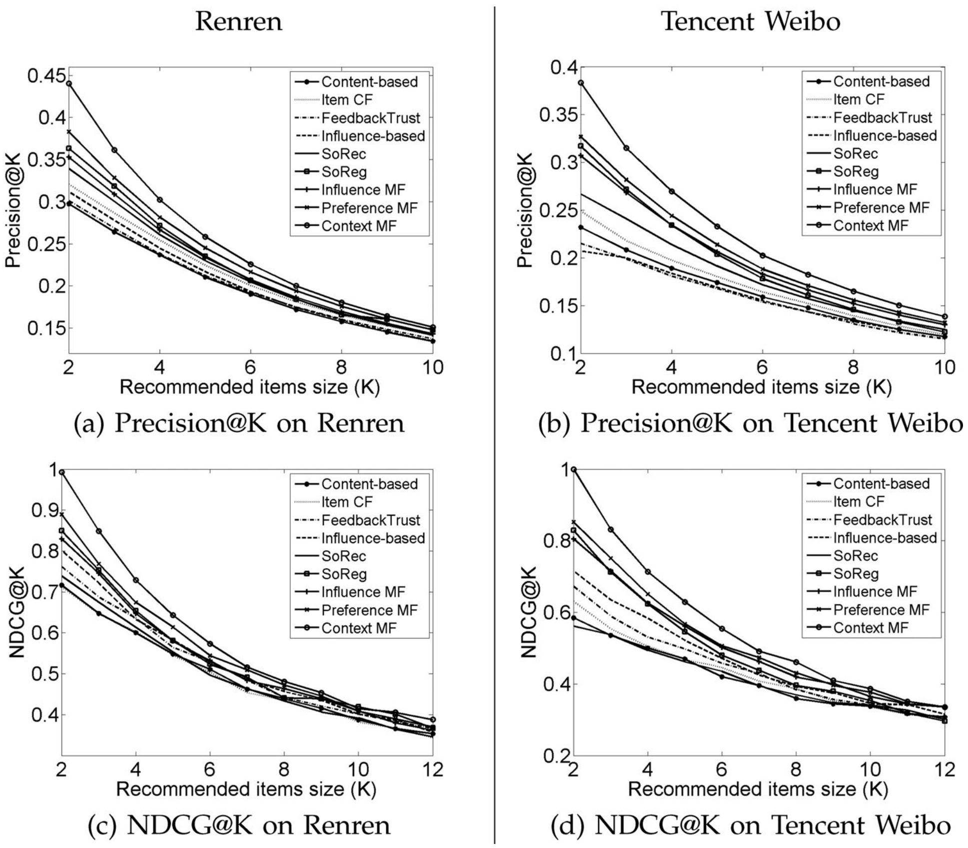


Fig. 11. Top K recommendation performance on Renren and Tencent Weibo data sets: compared with baselines, from (a), (b) we know Precision@5 increases by 21.7 percent on Renren and 12.3 percent on Tencent Weibo. Also from (c), (d), we know NDCG@5 increases by 4.7 percent on Renren and 10.8 percent on Tencent Weibo.

TABLE 3
Recommendation Stability on Two Data Sets

	MAE	RMSE	$\hat{\tau}$	$\hat{\rho}$	ERR	T-test
Renren Dataset						
$\overline{x}$	0.2416	0.3086	0.7783	0.7897	0.6987	4.2437
σ	0.0001	0.0001	0.0006	0.0006	0.0008	0.6
Tencent Dataset						
$\overline{x}$	0.1514	0.2348	0.8571	0.8686	0.7529	13.989
σ	0.0001	0.0002	0.0002	0.0001	0.001	0.8

negative instances, and thus report their ratio, i.e., T-test results. Our model gives the highest T-test (1.78 and 1.26 times of the best baseline), which shows the social contextual model has better distinguish-ability.

It should be noticed that: (1) $PreferenceMF$ and $InfluenceMF$ achieve better performance than $SoRec$, which demonstrates the effectiveness of introducing either individual preference or interpersonal influence. (2) The large improvement margin achieved by $ContextMF$ over both $PreferenceMF$ and $InfluenceMF$ demonstrates the importance of incorporating complete contextual information from both individual and interpersonal sides for social recommendation. We further give showcases and discuss *insights* from this improvement in Section 6, with real instances of unique users and items. (3) The fact that our proposed $ContextMF$ performs better than $SoReg$ proves the effectiveness of incorporating the two social contextual factors from users' motivations on item adoption, instead of adding average-based or individual-based regularization to user latent vectors.

Second, we compare our $ContextMF$ model and other algorithms on top K recommendation performance (Precision@K and NDCG@K) in Fig. 11. The performance increases regularly as we decreases the size of recommended items K. Compared with the best of baselines $SoReg$, our Precision@5 (precision of top 5 recommended items) increases by 21.7 percent and Precision@10 increases by 10.8 percent on Renren data set. Similarly, on Tencent Weibo data set, Precision@5 increases by 12.3 percent and Precision@10 increases by 6.85 percent. Also we take a look at NDCG@K: NDCG@5 increases by 4.7 percent on Renren data set and 10.8 percent on Tencent Weibo data set. The advantage of our method $ContextMF$ is much more obvious when K is small. As the user adoption behavior is very sparse, it is difficult to distinguish excellent methods when K is rather large. That's why all the baseline algorithms tend to converge as K becomes larger.

Third, we conduct experiments to test the stability of our model with different random starts of the gradients for 100 times. As shown in Table 3, the low variances of MAE and RMSE (less than 0.001) show that our algorithm not only performs well on both social networking and microblogging data sets, but also runs without big fluctuation.

TABLE 5
Data Sets for Comparison between Incremental Processing and Offline Recommendation

Dataset	Resource	ΔM	M_0=M-ΔM	N_0=N
RΔM1000	Renren	1,000	938,363	1,625,689
RΔM10000	Renren	10,000	929,363	1,625,689
TΔM1000	Tencent Weibo	1,000	162,661	529,615
TΔM10000	Tencent Weibo	10,000	153,661	529,615
		ΔN	M_0=M	N_0=N-ΔN
RΔN1000	Renren	1,000	939,363	1,624,689
RΔN10000	Renren	10,000	939,363	1,615,689
TΔN1000	Tencent Weibo	1,000	163,661	528,615
TΔN10000	Tencent Weibo	10,000	163,661	519,615

5.7 Incremental Capability Analysis

In this section, we analyze incremental capability of our $ContextMF$ model. We create 8 copies of data sets with combinations of the following properties, as shown in Table 5: (1) Renren or Tencent Weibo; (2) new users or new items; and (3) the number of new users ΔM and new items ΔN, 1,000 or 10,000. For example, in data set RΔM1000, we first randomly select $\Delta M = 1000$ users as *new users* from M Renren users. We hide the historical

TABLE 4
Recommendation Performance on Two Data Sets

Method	Prediction error		Ranking measure			T-test statistics		
	MAE	RMSE	$\hat{\tau}$	$\hat{\rho}$	ERR	Adopted	Refused	T-test
Renren Dataset								
$ContentBased$ [7]	0.384	0.477	0.541	0.540	0.325	0.702	0.665	1.06
$ItemCF$ [11]	0.360	0.451	0.590	0.599	0.397	0.360	0.268	1.34
$FeedbackTrust$ [34]	0.376	0.468	0.543	0.547	0.378	0.363	0.343	1.06
$InfluenceBased$ [27]	0.386	0.469	0.539	0.545	0.365	0.641	0.590	1.09
$SoRec$ [2]	0.328	0.413	0.617	0.620	0.452	0.473	0.347	1.37
$SoReg$ [3]	0.299	0.354	0.709	0.714	0.561	0.523	0.336	1.56
$InfluenceMF$	0.310	0.377	0.686	0.701	0.477	0.351	0.213	1.65
$PreferenceMF$	0.303	0.376	0.694	0.704	0.465	0.132	0.056	2.38
$ContextMF$	**0.242**	**0.309**	**0.778**	**0.790**	**0.699**	0.456	0.107	**4.24**
Tencent Weibo Dataset								
$ContentBased$ [7]	0.258	0.364	0.773	0.778	0.476	0.417	0.276	1.51
$ItemCF$ [11]	0.238	0.337	0.787	0.805	0.544	0.637	0.244	2.62
$FeedbackTrust$ [34]	0.283	0.389	0.709	0.712	0.492	0.792	0.610	1.30
$InfluenceBased$ [27]	0.265	0.381	0.716	0.728	0.491	0.800	0.392	2.04
$SoRec$ [2]	0.226	0.333	0.797	0.806	0.555	0.495	0.058	8.53
$SoReg$ [3]	0.200	0.296	0.839	0.842	0.667	0.552	0.060	9.174
$InfluenceMF$	0.218	0.321	0.818	0.82	0.572	0.522	0.062	8.42
$PreferenceMF$	0.211	0.309	0.838	0.845	0.568	0.576	0.052	11.1
$ContextMF$	**0.151**	**0.235**	**0.857**	**0.896**	**0.753**	0.812	0.058	**14.0**

TABLE 6
Recommendation Performance of Incremental Method $\Delta ContextMF$ and Offline Recommendation $ContextMF^{\Delta}$

Dataset	RMSE (smaller is better)			ERR (bigger is better)			Time cost	
	$SoReg$	$\Delta ContextMF$	$ContextMF^{\Delta}$	$SoReg$	$\Delta ContextMF$	$ContextMF^{\Delta}$	$\Delta ContextMF$	$ContextMF^{\Delta}$
RΔM1000	0.342	0.263	0.257	0.555	0.610	0.636	172s	41.7h
RΔM10000	0.502	0.464	0.444	0.481	0.542	0.559	1610s	41.7h
TΔM1000	0.168	0.122	0.105	0.652	0.764	0.783	54.2s	2.42h
TΔM10000	0.342	0.333	0.317	0.534	0.611	0.651	531s	2.42h
RΔN1000	0.335	0.276	0.276	0.570	0.663	0.680	97.3s	41.7h
RΔN10000	0.546	0.478	0.465	0.514	0.587	0.609	941s	41.7h
TΔN1000	0.218	0.192	0.173	0.726	0.824	0.864	17.8s	2.42h
TΔN10000	0.427	0.376	0.355	0.658	0.720	0.751	160s	2.42h

data of the new users and then with the data of the remaining $M_0 = M - \Delta M$ users we train the interpersonal influence matrix $\mathbf{S}$ and latent feature matrices $\mathbf{U}$, $\mathbf{V}$. Third, we apply our incremental version $\Delta ContextMF$, retrain the offline model $ContextMF^{\Delta}$ and the baseline $SoReg$, and compare their performances on solving the cold-start problem.

In Table 6, running time of incremental processing method $\Delta ContextMF$ is much less than that of offline recommendation $ContextMF^{\Delta}$: it is reduced from the level of hour to that of second. Improving efficiency will reduce the effectiveness because the high-order terms are ignored in $\Delta ContextMF$. However, RMSE of $\Delta ContextMF$ is only 2.33 percent bigger (worse) than that of $ContextMF^{\Delta}$ on the data sets from Renren, which is applicable for real cases. On the other hand, the performance of $\Delta ContextMF$ still outperforms that of $SoReg$ because it fully learns the social contextual information: On Renren, RMSE of $\Delta ContextMF$ is 18.5 percent smaller (better) than that of $SoReg$ on average; on Tencent Weibo, RMSE of $\Delta ContextMF$ is 16.9 percent smaller. Also, ERR of $\Delta ContextMF$ is 11.7 and 11.9 percent bigger (better) on Renren and Tencent than that of $SoReg$. We demonstrate that on real social networks, our $\Delta ContextMF$ model, carefully designed for incremental data, has significant performance on both social network data sets.

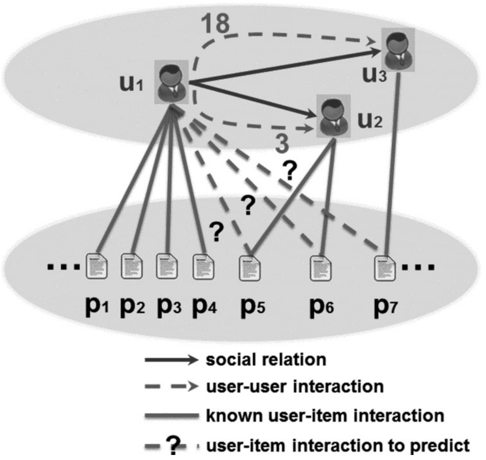


Fig. 12. An example of social recommendation case on Tencent Weibo: user u_1 follows u_2 and u_3 and thus is able to receive messages from them; before time t, (1) u_1 retweeted 18 messages from u_3 before but only 3 from u_2; (2) u_1 adopted posts $p_1, \ldots, p_4$. At the time t, user u_1 receives p_5 and p_6 from u_2 and p_7 from u_3. In this case, our task is to predict whether u_1 will adopt them nor not.

5.8 Insights

Besides the above numbers, we provide unique instances (users and items) to demonstrate the importance of incorporating all kinds of social contextual information: social relation, item content, user-user interaction and user-item interaction. An example of social recommendation case on Tencent Weibo microblogging service is shown in Fig. 12 and Table 7. In this scenario, user u_1 follows u_2 and u_3, and thus it is able to receive messages from them: (1) Before time t, u_1 adopted (retweeted) 18 messages from u_3 before but only 3 from u_2. Our $ContextMF$ learns interpersonal influence between them: u_1 prefers interacting with u_3; but $PreferenceMF$ does not. (2) The user u_1 adopted posts including $p_1, \ldots, p_4$: p_1, p_2 and p_3 have consistently high numbers on the 8th topical distribution because their contents are mainly about programming language, coding and computer engineering; p_4 tells about love and life and has a peak on the 3rd topic. Our model also learns his preference as the second contextual factor: u_1 loves content in these unique fields; but $InfluenceMF$ does not. In this condition, at time t, u_1 receives p_5 and p_6 from u_2 and p_7 from u_3. We focus on recommendation task for user u_1, i.e., ranking these web posts with adoption behavior prediction of users.

In Table 8, we give predicted values of user-item links from user u_1 to the posts p_5, p_6 and p_7, i.e., probability of event that user u_1 adopts these posts. Post p_5 is about programming language java that u_1 likes. From the past behaviors of u_1, $ContextMF$ predicts that the probability of his adopting is 88.4 percent, while $InfluenceMF$ just knows that he does not adopts many messages before from p_5's sender u_2 and gives its answer 19.0 percent. Post p_6 and p_7

TABLE 7
Topic Distribution and Content of Posts in our Example of Social Recommendation Case (Fig. 12)

Post ID	Topic t_3	Topic t_8	Content
p_1	0.00	0.86	I love java, I like code!!!
p_2	0.00	0.72	Have you ever read this? The Zen of Python by Tim Peter
p_3	0.02	0.91	We want a web developer: 1. know java 2. know HTML,CSS, XML,AJAX,JavaScript (Beijing)
p_4	0.65	0.09	Love starts with a smile develops with a kiss and ends with a tear.
p_5	0.12	0.68	I met...Exception in thread main me.love.NoGirlFriendError
p_6	0.71	0.00	I miss you. But I missed you.
p_7	0.68	0.00	if you leave me please don't comfort me because each sewing has to meet stinging pain

TABLE 8
Predicted Values of Links from User u_1 to Items p_5, p_6, p_7

	$\mathcal{R}(u_1,p_5)$	$\mathcal{R}(u_1,p_6)$	$\mathcal{R}(u_1,p_7)$
Ground truth	1	0	1
$ContextMF$	**0.884**	**0.112**	**0.845**
$PreferenceMF$	0.901	**0.354**	**0.323**
$InfluenceMF$	**0.190**	0.094	0.854

have similar topic-level distributions. However, since p_7 is adopted by user u_3 who has bigger impact on u_1 than p_6's sender u_2 does (u_1 has retweeted u_3's 18 messages), both $ContextMF$ and $InfluenceMF$ predict that the user u_1 will prefer p_7, but $PreferenceMF$ does not. The prediction results of $ContextMF$ are much closer to ground truth than those of $PreferenceMF$ and $InfluenceMF$ because $ContextMF$ fuses all the social contextual information into a single model.

6 Conclusions

We proposed $ContextMF$, a novel social recommendation model utilizing social contextual factors, i.e., individual preference and interpersonal influence. We conducted extensive experiments on two large real-world social network data sets, and showed that social contextual information can greatly boost the performance of recommendation on social network data sets. In particular, we have gained growth of 24.2 and 20.7 percent in prediction accuracy and 21.7 and 12.3 percent in recommendation Precision@K upon previous approaches on these social networks, respectively. Also, the proposed algorithm is general and can be easily adapted according to different real-world recommendation scenarios.

Acknowledgments

This work was supported by National Natural Science Foundation of China, No. 61370022, No. 61003097, No. 60933013, and No. 61210008; International Science and Technology Cooperation Program of China, No. 2013DFG12870; National Program on Key Basic Research Project, No. 2011CB302206. Thanks for the support of NExT Research Center funded by MDA, Singapore, under the research grant, WBS:R-252-300-001-490.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, June 2005.

[2] H. Ma, H. Yang, M.R. Lyu, and I. King, "SoRec: Social Recommendation Using Probabilistic Matrix Factorization," *Proc. 17th ACM Conf. Information and Knowledge Management (CIKM '08)*, pp. 931-940, 2008.

[3] H. Ma, D. Zhou, C. Liu, M.R. Lyu, and I. King, "Recommender System with Social Regularization," *Proc. Fourth ACM Int'l Conf. Web Search and Data Mining (WSDM '11)*, pp. 287-296, 2011.

[4] R. Bond and P.B. Smith, "Culture and Conformity: A Meta-Analysis of Studies Using Asch's (1952b, 1956) Line Judgment Task," *Psychological Bull.*, vol. 119, no. 1, pp. 111-137, 1996.

[5] M.J. Salganik, "Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market," *Science*, vol. 311, pp. 854-856, 2006.

[6] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

[7] M. Balabanović and Y. Shoham, "FAB: Content-Based, Collaborative Recommendation," *Comm. ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[8] O. Phelan, K. McCarthy, and B. Smyth, "Using Twitter to Recommend Real-Time Topical News," *Proc. Third ACM Conf. Recommender Systems (RecSys '09)*, pp. 385-388, 2009.

[9] K. Stefanidis, E. Pitoura, and P. Vassiliadis, "Managing Contextual Preferences," *Information System*, vol. 36, no. 8, pp. 1158-1180, 2011.

[10] H. Zhu, E. Chen, K. Yu, H. Cao, H. Xiong, and J. Tian, "Mining Personal Context-Aware Preferences for Mobile Users," *Proc. IEEE 12th Int'l Conf. Data Mining (ICDM '12)*, pp. 1212-1217, 2012.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l Conf. World Wide Web (WWW '01)*, 2001.

[12] L. Si and R. Jin, "Unified Filtering by Combining Collaborative Filtering and Content-Based Filtering via Mixture Model and Exponential Model," *Proc. 13th ACM Int'l Conf. Information and Knowledge Management (CIKM '04)*, pp. 156-157, 2004.

[13] Y. Koren, "Collaborative Filtering with Temporal Dynamics," *Comm. ACM*, vol. 53, no. 4, pp. 89-97, 2010.

[14] Y. Chen and J.F. Canny, "Recommending Ephemeral Items at Web Scale," *Proc. 34th Int'l ACM SIGIR Conf. Research and Development in Information (SIGIR '11)*, pp. 1013-1022, 2011.

[15] Q. Liu, E. Chen, H. Xiong, C. Ding, and J. Chen, "Enhancing Collaborative Filtering by User Interest Expansion via Personalized Ranking," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 1, pp. 218-233, Feb. 2012.

[16] Y. Koren, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[17] D. Kong, C. Ding, and H. Huang, "Robust Nonnegative Matrix Factorization Using L21-Norm," *Proc. 20th ACM Int'l Conf. Information and Knowledge Management (CIKM '11)*, pp. 673-682, 2011.

[18] F. Wang, H. Tong, and C.Y. Lin, "Towards Evolutionary Nonnegative Matrix Factorization," *Proc. Assoc. Advancement of Artificial Intelligence (AAAI '11)*, pp. 501-506, 2011.

[19] Z. Zhang, K. Zhao, and H. Zha, "Inducible Regularization for Low-Rank Matrix Factorizations for Collaborative Filtering," *Neurocomputing*, vol. 97, pp. 52-62, 2012.

[20] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang, "Social Contextual Recommendation," *Proc. 21st ACM Int'l Conf. Information and Knowledge Management (CIKM '12)*, pp. 45-54, 2012.

[21] M. Jiang, P. Cui, F. Wang, Q. Yang, W. Zhu, and S. Yang, "Social Recommendation Across Multiple Relational Domains," *Proc. 21st ACM Int'l Conf. Information and Knowledge Management (CIKM '12)*, pp. 1422-1431, 2012.

[22] M. Ou, P. Cui, F. Wang, J. Wang, W. Zhu, and S. Yang, "Comparing Apples to Oranges: A Scalable Solution with Heterogeneous Hashing," *Proc. 19th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '13)*, pp. 230-238, 2013.

[23] X. Liu and K. Aberer, "SoCo: A Social Network Aided Context-Aware Recommender System," *Proc. 22nd Int'l Conf. World Wide Web (WWW '13)*, pp. 781-802, 2013.

[24] J. Tang, X. Hu, and H. Liu, "Social Recommendation: A Review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113-1133, 2013.

[25] J. Tang, X. Hu, H. Gao, and Huan Liu, "Exploiting Local and Global Social Context for Recommendation," *Proc. 23rd Int'l Joint Conf. Artificial Intelligence (IJCAI '13)*, 2013.

[26] J. Leskovec, A. Singh, and J. Kleinberg, "Patterns of Influence in a Recommendation Network," *Proc. 10th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD '06)*, pp. 380-389, 2006.

[27] J. Huang, X. Cheng, J. Guo, H. Shen, and K. Yang, "Social Recommendation with Interpersonal Influence," *Proc. 19th European Conf. Artificial Intelligence (ECAI '10)*, 2010.

[28] A. Goyal, F. Bonchi, and L.V.S. Lakshmanan, "Learning Influence Probabilities in Social Networks," *Proc. Third ACM Int'l Conf. Web Search and Data Mining (WSDM '10)*, pp. 241-250, 2010.

[29] Y. Shen, R. Jin, D. Dou, N.A. Chowdhury, J. Sun, B. Piniewski, and D. Kil, "Socialized Gaussian Process Model for Human Behavior Prediction in a Health Social Network," *Proc. IEEE 12th Int'l Conf. Data Mining (ICDM '12)*, pp. 1110-1115, 2012.

[30] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun, "Who Should Share What? Item-Level Social Influence Prediction for Users and Posts Ranking," *Proc. 34th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '11)*, pp. 185-194, 2011.

[31] P. Cui, F. Wang, S. Yang, and L. Sun, "Item-Level Social Influence Prediction with Probabilistic Hybrid Factor Matrix Factorization," *Proc. Assoc. Advancement of Artificial Intelligence (AAAI '11)*, 2011.

[32] F.C.T. Chua, H.W. Lauw, and E. Lim, "Generative Models for Item Adoptions Using Social Correlation," *IEEE Trans. Knowledge and Data Eng.*, vol. 25, no. 9, pp. 2036-2048, Sept. 2013.

[33] P. Massa and P. Avesani, "Trust-Aware Recommender Systems," *Proc. ACM Conf. Recommender Systems (RecSys '07)*, pp. 17-24, 2007.

[34] S. Moghaddam, M. Jamali, M. Ester, and J. Habibi, "FeedbackTrust: Using Feedback Effects in Trust-Based Recommendation Systems," *Proc. ACM Conf. Recommender Systems (RecSys '09)*, pp. 269-272, 2009.

[35] M. Jamali and M. Ester, "TrustWalker: A Random Walk Model for Combining Trust-Based and Item-Based Recommendation," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '09)*, pp. 397-406, 2009.

[36] B. Carminati, E. Ferrari, and J. Girardi, "Trust and Share: Trusted Information Sharing in Online Social Networks," *Proc. IEEE 28th Int'l Conf. Data Eng. (ICDE '12)*, 2012.

[37] A. Bandura, "Social Cognitive Theory of Mass Communication," *Media Psychology*, vol. 3, pp. 265-299, 2001.

[38] L.S. Benjamin, "Structural Analysis of Social Behavior," *Psychological Rev.*, vol. 81, no. 5, pp. 392-425, 1974.

[39] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Information Systems*, vol. 22, no. 1, pp. 5-53, 2004.

[40] D. Agarwal and M. Gurevich, "Fast Top-k Retrieval for Model Based Recommendation," *Proc. Fifth ACM Int'l Conf. Web Search and Data Mining (WSDM '12)*, pp. 483-492, 2012.

[41] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to Rank Using Gradient Descent," *Proc. 22nd Int'l Conf. Machine Learning (ICML'05)*, pp. 89-96, 2005.

[42] M. Sanderson, M.L. Paramita, P. Clough, and E. Kanoulas, "Do User Preferences and Evaluation Measures Line Up?" *Proc. 33rd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '10)*, 2011.

[43] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected Reciprocal Rank for Graded Relevance," *Proc. 18th ACM Conf. Information and Knowledge Management (CIKM '09)*, pp. 621-630, 2009.

Meng Jiang received the BE degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, in 2010. He is currently working toward the PhD degree and his main research interests include data mining and social network analysis. He was sponsored by China Scholarship Council to visit Carnegie Mellon University, from August 2012 to May 2013. He has published papers on social recommendation in top conferences of the relevant field.

Peng Cui received the PhD degree in computer science from Tsinghua University in 2010. He is an assistant professor at Tsinghua University. He has vast research interests in data mining, multimedia processing, and social network analysis. Until now, he has published more than 20 papers in conferences such as SIGIR, AAAI, ICDM, etc. and journals such as *IEEE TMM*, *IEEE TIP*, *DMKD*, etc. Now his research is sponsored by National Science Foundation of China, Samsung, Tencent, etc. He also serves as guest editor, cochair, PC member, and reviewer of several high-level international conferences, workshops, and journals.

Fei Wang is a research staff member in Healthcare Analytics Research Group, IBM T.J. Watson Research Center. Before his current position, he has been a postdoctoral researcher in the same group from 2010 to 2011, and at the Department of Statistical Science, Cornell University, from 2009 to 2010. His major research interests include data mining, machine learning, and how to make use of those technologies in heathcare and social informatics. He has been serving on the Program Committee Members of major data mining and machine learning conferences such as ICDM, SDM, KDD, CIKM, and IJCAI, and referee of major journals such as *JMLR*, *Machine Learning journal*, *AI journal*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Knowledge Discovery from Data*, *Data Mining and Knowledge Discovery*.

Wenwu Zhu received the PhD degree from Polytechnic Institute, New York University in 1996. He is currently a professor at Tsinghua University. His research interest include wireless/Internet multimedia communication and computing. He worked at Bell Labs during 1996 to 1999. He was with Microsoft Research Asia's Internet Media Group and Wireless and Networking Group as research manager from 1999 to 2004. He was the director and chief scientist at Intel Communication Technology Lab, China. He was also a senior researcher at the Internet Media Group at Microsoft Research Asia. He has published more than 200 referred papers and led 40 patents. He participated in the IETF ROHC WG on robust TCP/IP header compression over wireless links and IEEE 802.16m WG standardization. He currently serves as the chairman of IEEE Circuits and System Society Beijing Chapter and advisory board on the *International Journal of Handheld Computing Research*. He is a fellow of the IEEE.

Shiqiang Yang received the BE and ME degrees from the Department of Computer Science and Technology, Tsinghua University, in 1977 and 1983, respectively. He is currently a professor at Tsinghua University. His research interests include multimedia technology and systems, video compression and streaming, content-based retrieval for multimedia information, multimedia content security, and digital right management. He has published more than 100 papers and MPEG standard proposals. He has organized many conferences as program Chair or TPC member, including PCM'05, PCM'06 Workshop on ACM Multimedia'05, MMM'06, ICME'06, MMSP'05, ASWC'06, etc. He is a senior member of the IEEE.