Inferring Latent Network from Cascade Data for Dynamic Social Recommendation

Qin Zhang, Jia Wu, Peng Zhang, Guodong Long, Ivor W. Tsang, Chengqi Zhang
Centre for Quantum Computation and Intelligent Systems, FEIT,
University of Technology Sydney, NSW 2007, Australia
Qin.Zhang@student.uts.edu.au; Jia.Wu@uts.edu.au; zhangpeng04@gmail.com;
Guodong.Long@uts.edu.au; Ivor.Tsang@uts.edu.au; Chengqi.Zhang@uts.edu.au

***Abstract*—Social recommendation explores social information to improve the quality of a recommender system. It can be further divided into explicit and implicit social network recommendation. The former assumes the existence of explicit social connections between users in addition to the rating data. The latter one assumes the availability of only the ratings but not the social connections between users since the explicit social information data may not necessarily be available and usually are binary decision values (e.g., whether two people are friends), while the strength of their relationships is missing. Most of the works in this field use only rating data to infer the latent social networks. They ignore the dynamic nature of users that the preferences of users drift over time distinctly. To this end, we propose a new Implicit Dynamic Social Recommendation(IDSR) model, which infers latent social network from cascade data. It can sufficiently mine the information contained in time by mining the cascade data and identify the dynamic changes in the users in time by using the latest updated social network to make recommendations. Experiments and comparisons on three real-world datasets show that the proposed model outperforms the state-of-the-art solutions in both explicit and implicit scenarios.**

***Index Terms*—Social recommendation, Latent dynamic social network, Cascade data**

I. Introduction

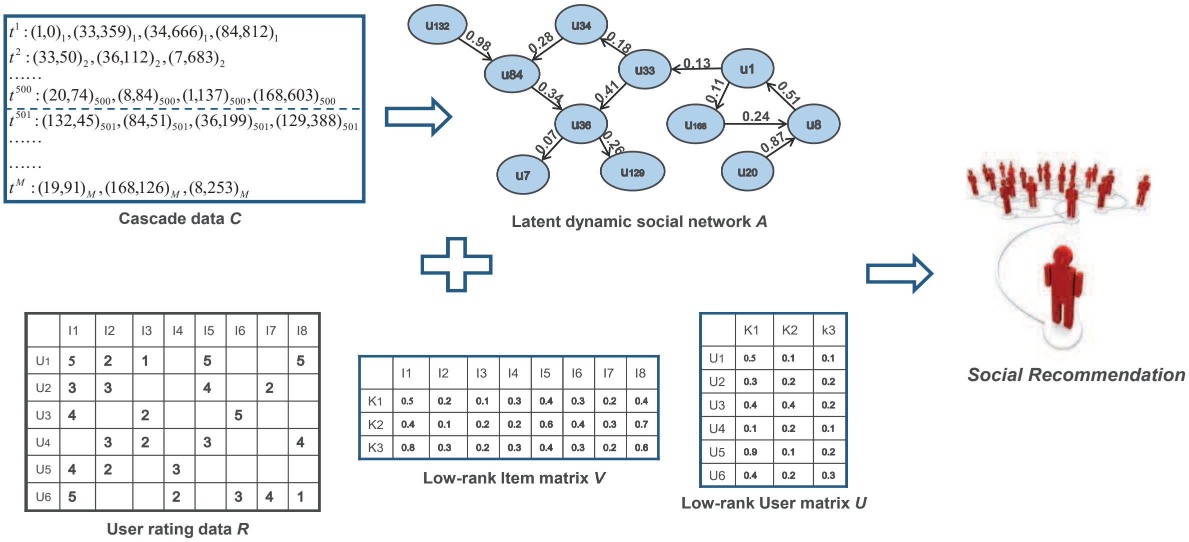
Social recommendation, a study aimed at incorporating the social information of users into a recommender system, has attracted much attention in recent years. It can further be divided into two types: explicit and implicit social recommendation. The existing and available social network information is often used to enhance the performance of a recommender system, i.e. explicit social recommendation [20] [11] [12] [19]. The most successful and common strategy is to integrate social information, either trust or friendship, into a collaborative filtering model in a certain way.

However, social information data may not necessarily be available for every recommendation scenario due to practical difficulties or privacy concerns. For example, Taobao, the most popular online shopping platform in China which would be greatly improved by a social recommender system, has not built a social network module for its users. On the other hand, most of the signals that a user provides about his preferences are implicit, such as watching a video or clicking on a link.

Furthermore, for most applications with disclosed social relationships, data are usually given as a binary decision value(e.g., whether two people are friends), while the strength of their relationship is missing. Knowing the strength of social relationships is very helpful for a recommender system, as it is reasonable to assume people have more trust in their close friends compared to their acquaintances. In addition, the quality of the given social information is sometimes questionable. Since most social data are collected from the web or social network services, inevitably they contain noise. Although it is generally believed that trust or friendship are positively correlated with the level of common-taste of people, the work in [1] shows that two users may not have similar rating tastes even though they strongly trust each other. Thus, an absolute acceptance of the given social connections can harm recommendation performance.

These concerns emerges another research direction named *implicit social recommendation*, which aims at mining implicit user social relationships from historical rating data for better recommendations. Implicit social recommendation can be further divided into two types, one is to determine the social connection strength of the existing binary social network [3] [4] to enhance the quality of recommendation based on the given rating data. The other is data to generate an implicit social network from given historical ratings without any explicit social data [6] [15] [16] [20]. The pseudo links and/or their strengths can then act as a surrogate of the explicit social network [5] [35] to be incorporated into any explicit social recommendation model.

However, most of the existing implicit social recommendation research ignore the dynamic nature of the users. They only use the rating data to study the implicit social relationships, and ignore the dynamic information propagation. The dynamic nature of users' preferences means that they may drift over time in dynamic recommendation, resulting in users having different preferences for particular items at different times [29] [36]. For example, if the user gets married or moves to another city, his preferences are likely to change dramatically, and it is natural for him to turn to new friends for advice rather than old friends. A recommender system should take a user's actions into account instantaneously and adapt recommendations to the user's most recent preference. In fact, the fresher the feedback, the more informative it is on the user's current preference. Thus it is much desired that the dynamic social information can be effectively utilized to capture the dynamic drift of the users' preferences and



$t^1 : (1,0)_1, (33,359)_1, (34,666)_1, (84,812)_1$
$t^2 : (33,50)_2, (36,112)_2, (7,683)_2$
……
$t^{500} : (20,74)_{500}, (8,84)_{500}, (1,137)_{500}, (168,603)_{500}$
$t^{501} : (132,45)_{501}, (84,51)_{501}, (36,199)_{501}, (129,388)_{501}$
……
……
$t^M : (19,91)_M, (168,126)_M, (8,253)_M$

Cascade data *C*

	I1	I2	I3	I4	I5	I6	I7	I8
U1	5	2	1		5			5
U2	3	3			4		2	
U3	4		2			5		
U4		3	2		3			4
U5	4	2		3				
U6	5			2		3	4	1

User rating data *R*

	I1	I2	I3	I4	I5	I6	I7	I8
K1	0.5	0.2	0.1	0.3	0.4	0.3	0.2	0.4
K2	0.4	0.1	0.2	0.2	0.6	0.4	0.3	0.7
K3	0.8	0.3	0.2	0.3	0.4	0.3	0.2	0.6

Low-rank Item matrix *V*

	K1	K2	k3
U1	0.5	0.1	0.1
U2	0.3	0.2	0.2
U3	0.4	0.4	0.2
U4	0.1	0.2	0.1
U5	0.9	0.1	0.2
U6	0.4	0.2	0.3

Low-rank User matrix *U*

Figure 1. An example of Implicit Dynamic Social Recommendation (IDSR) framework. We infer both the structure and the strength of latent dynamic social network A from cascade data C to catch the dynamic changes of the users by using a generative probabilistic model. Based on the inferred latent dynamic social network A, IDSR model learns the low-rank item matrix V and user matrix U to predict the missing values in rating data R. By jointly learning the social relationships and missing ratings from both rating and cascade data, IDSR can absorb the potential damage brought up by noisy explicit social network and capture the most recent preferences of the users and give the exact recommendations.

give exact recommendations in time, and finally improve the recommendation performances.

To this end, in this paper, we propose a new **Implicit Dynamic Social Recommendation (IDSR)** model which uses cascade data to infer latent social network and attempts to identify the dynamic changes in the users' preferences to improve the performance of recommender system. We define cascade data as the observed action time stamps of users on certain items. We follow the idea of the generative probabilistic model to infer both the structure and the strength of the latent social network. By jointly learning the social relationships and ratings from the data, our model can absorb the potential damage caused by a noisy explicit social network. By exploring the information in time data, we capture the most recent preferences of the users and give exact recommendations. Figure 1 shows an example of the IDSR framework. Experiments show that the proposed solution outperforms the state-of-the-art models in both explicit and implicit scenarios. The **contributions** of the paper are summarized as follows:

1) We study a new problem of implicit dynamic social recommendation, which infers the latent social network from cascade data and uses rating data to make recommendations based on the inferred latent social network. It can sufficiently mine the dynamic information propagation and identify the dynamic changes of users' preferences in time.
2) We propose a new Implicit Dynamic Social Recommendation (IDSR) model to combine the learning of social network and rating prediction together as an unified optimization problem, which is different from most of the existing approaches of implicit social network recommendations which treat the learning of the social networks and recommendations as two independent tasks.
3) We use two new real-world datasets (Zomato and Douban movie data) we collected, and a public MovieLens dataset to evaluate the performance of the proposed model. Experiments show that the proposed model outperforms the state-of-the-art solutions in both explicit and implicit social recommendation scenarios on these three real-world datasets.

The rest of this paper is organized as follows: Section II surveys the related work. Sections III and IV introduce the problem and IDSR model in detail. Section V discusses the proposed algorithm. Section VI describes the experiments and Section VII concludes the paper.

II. Related work

In this section, we review several major approaches to social recommendations. Firstly we briefly introduce the principle of social recommendation, then we introduce the recent research on the two main categories: explicit social recommendation which uses external social networks and implicit social recommendation which uses the social networks inferred from the rating data.

Social recommendation [19] is defined as any recommendation with online social relations as additional input, i.e., augmenting an existing recommendation engine with additional social signals [28]. Social relations can be trust relations, friendships, memberships or following relations and so on [20] [10]. The underlying assumption is that users are correlated when they establish social relations [19] [21].

Most existing social recommender systems choose CF models as their basic models to build systems and propose approaches to capture social information based on the results of social network analysis [28]. Matrix factorization techniques are widely used in CF models. The general idea of matrix factorization is to model the user-item interactions with factors representing the latent characteristics of the users and items in the system, like the preference class of users and the category class of items [31]. Numerous social matrix factorization based RSs have recently been proposed to improve recommendation accuracy [33] [19] [32].

Explicit social recommendation uses rating data and all the information from external social networks. Early research [11] [21] searches the trust network to determine the recommended items. Later on, researchers started to bring social information into matrix factorization models with diversified assumptions for integration. SoRec [19] proposes shared user latent factors for both rating matrix and social matrix factorization. RSTE [17] predicts a user's ratings by the linear combination of the user and their trusted friends' latent factors vectors. SocialMF [12] defines that a user's latent factors should be close to the linear combination of his or her trusted friends' latent factors. Social Regularization [20] considers a pairwise assumption that two users who trust each other should have similar latent factors, and thus appends a regularization term to the classical matrix factorization model. Moreover, social influence is also considered in social recommendation, such as conditional random field [30] and probabilistic Poisson factorization [2]. TrustSVD [8] incorporates trust networks with SVD++, a variant of matrix factorization modeling implicit influence from user latent factors through observed ratings.

Implicit social recommendation attempts to extract latent social correlation between two users from historical rating behaviors. The generated information serves as the surrogate for explicit social networks in explicit social recommender systems. Since it is time-consuming to evaluate the quadratic number of pairwise users or item social relations, several studies assume that an explicit social network is available but the explicit edge strength information is missing. Fazeli et al. [4] survey and compare the performance of different trust strength metrics on an explicit social network combined with SocialMF. Fang et al. [3] use support vector regression with matrix factorization to learn both the ratings and strengths from an explicit trust network. However, in order to train this model, a binary social network is still required. Despite the quadratic time complexity of evaluating relations, Guo et al. [6] study user-based collaborative filtering that recommends items using a trust network generated from predefined trust metrics. With the existence of extra features, Lin et al. [15] (rating time) and Guo et al. [7] (text review) present methods to obtain implicit social networks. There are also several studies that apply matrix factorization techniques on implicit social networks [18]. And Social Regularization [16] reads implicit social networks generated by the evaluation of cosine similarity and Pearson correlation respectively together with predefined thresholds to determine the social connections.

In some cases, the existing explicit and implicit social recommendation cannot achieve appropriate performance, because they ignore the dynamic changes of uses' preferences. To address this issue, out model is proposed to learn the structure and the dynamic strength of social connection through cascade data simultaneously, which is more general because implicit signals such as clicks can be used to obtain the cascade data. Furthermore, by modeling the rating prediction and social strength learning as a joint optimization task, the proposed model mutually reinforces the quality of each to achieve better results.

III. Problem definition

The rating data is denoted by matrix $R \in \mathbb{R}^{N\times M}$, where N is the number of users and M is the number of items. An observed or non-missing entry R_{ij}, records a numerical rating score that user $i, 1 \le i \le N$, gives to item $j, 1 \le j \le M$, as a training instance. An adjacent matrix $A \in \mathbb{R}^{N\times N}$ denotes the social network, where entry A_{ij} is a positive value which represents the strength of the edge from node i to node j.

Time observations are also recorded on the N users and consist of a set C of cascades $\{t^1, \dots, t^M\}$. Each cascade t^c is a record of observed action time stamps within the population during a time interval of length T. A cascade is an N-dimensional vector $t^c := (t_1^c, \dots, t_N^c)$ recording when nodes' actions are observed, $t_k^c \in [0,T] \cup \{\infty\}$. The symbol ∞ labels users that are not observed acting during observation window $[0,T]$, which does not imply the nodes will never act. Each cascade is reset to start at 0 as we are not concerned about the specific time but the relative time. Lengthening the observation window T increases the number of observed infections within a cascade t^c and results in a more representative sample of the underlying dynamics [23].

Our goal is to mine the implicit dynamic social network A from cascade data C, and predict the missing value of the original rating matrix R simultaneously by solving an optimization problem.

IV. The Proposed IDSR Model

The formula of the **Implicit Dynamic Social Recommendation** (IDSR) model, which consists of dynamic social network inferring term and social recommendation regularization term, is shown in Eq. (1).

$$\begin{aligned} \min_{A,U,V} & -\frac{1}{2}\sum_{c\in C} logf(t^c, A) + \frac{\lambda_1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}\|U_i - U_j\|^2 \\ & + \frac{\lambda_2}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_3}{2}\|U\|_F^2 + \frac{\lambda_4}{2}\|V\|_F^2 \\ s.t. \quad & A \ge 0, U \ge 0, V \ge 0 \end{aligned} \tag{1}$$

where I_{ij} is the entry of indicator matrix I. I_{ij} is equal to 1 if user u_i rates item v_j and equals to 0 otherwise. Matrix $U \in \mathbb{R}^{K\times N}$ and $V \in \mathbb{R}^{K\times M}$ with $K << min(M,N)$ are two low-rank matrices to approximate the rating matrix R. Matrix $A \in \mathbb{R}^{N\times N}$ is the inferred social network and the entry A_{ij} represents the strength of the influence from user i

to user j. N is the number of users, M is the number of items as previously introduced. $t^c \in C$ is a N-dimensional cascade vector. $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are positive parameters to control the weight of each term.

Indeed, the model in Eq. (1) mainly comprises three terms. The first term $-\frac{1}{2}\sum_{c\in C} logf(t^c, A)$ is the dynamic social network inferring. The second term $\frac{\lambda_1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}\|U_i - U_j\|^2$ is the social recommendation regularization, and the third term $\frac{\lambda_2}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{ij} - U_i^T V_j)^2$ is the standard matrix factorization based recommender system, with the last two terms $\frac{\lambda_3}{2}\|U\|_F^2$ and $\frac{\lambda_4}{2}\|V\|_F^2$ been added to avoid overfitting. Our model conducts the learning of social network and rating prediction together as a unified framework. By using the dynamic social network inferring term, the proposed model can infer the latent social network from cascade data which sufficiently mine the dynamic information propagation and catch the latest dynamic changes of the users in time to improve the performance of recommendation.

A. Dynamic social network inference

In this section, we introduce the method of inferring dynamic social network from cascade data. We first introduce the pairwise transmission likelihood, then we give the formula of likelihood of a cascade, and finally show the social network inferring term.

• Pairwise transmission likelihood. With the cascades data, we calculate the pairwise transmission likelihood as follow. We assume that infections can occur at different rates over different edges of a network, and aim to infer the transmission rates between pairs of nodes in the network. Define $f(t_i|t_j, A_{j,i})$ as the conditional likelihood of transmission between a node j and node i. The transmission likelihood depends on the infection times (t_j, t_i) and a pairwise transmission rate $A_{j,i}$. A node cannot be infected by a node infected later in time. In this paper, we estimate the well-known exponential parametric likelihood model (2)

$$f(t_i|t_j; A_{j,i}) = \begin{cases} A_{j,i} \cdot e^{-A_{j,i}(t_i - t_j)}, & \text{if } t_j < t_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $A_{j,i} \geq 0$ is transmission rate. If $A_{j,i} \to 0$, the likelihood of infection tends to zero and the expected transmission time becomes arbitrarily long.

The cumulative density function $F(t_i|t_j; A_{j,i})$ is computed from the transmission likelihoods [27]. Given that node j is infected at time t_j, the survival function of edge $j \to i$ is the probability that node i is not infected by node j by time t_i:

$$S(t_i|t_j; A_{j,i}) = 1 - F(t_i|t_j; A_{j,i}) = e^{-A_{j,i}(t_i - t_j)} \tag{3}$$

The hazard function [27] [23], or instantaneous infection rate, of edge $j \to i$ is the ratio

$$H(t_i|t_j; A_{j,i}) = \frac{f(t_i|t_j; A_{j,i})}{S(t_i|t_j; A_{j,i})} = A_{j,i} \tag{4}$$

• Likelihood of a cascade. We compute the probability that a node survives uninfected until time T, given that some of its parents are already infected. Consider a cascade $t = (t_1, \dots, t_N)$ and a node i, which is not infected during the observation window (i.e., $t_i > T$). Since each infected node k may infect i independently, the probability that nodes $1 \dots N$ do not infect node i by time T is the product of the survival functions of the infected nodes $1 \dots N | t_k \leq T$ targeting i,

$$\prod_{t_k \leq T} S(T|t_k; A_{k,i}) \tag{5}$$

Since we assume infections are conditionally independent given the parents of the infected nodes, the likelihood factorization over nodes is

$$f(t^{\leq T}; A) = \prod_{t_i \leq T} f(t_i|t_1, \dots, t_N \backslash t_i; A) \tag{6}$$

where $t_{\leq T} = (t_1 \dots, t_N | t_i \leq T)$. Given an infected node i, we compute the likelihood of a potential parent j to be the first parent by applying Eq. (5),

$$f(t_i|t_j; A_{j,i}) \times \prod_{j \neq k, t_k < t_i} S(t_i|t_k; A_{k,i}) \tag{7}$$

With Eq. (6), the likelihood of the infections in a cascade is shown in Eq. (8) by removing the condition $k \neq j$,

$$f(t^{\leq T}; A) = \prod_{t_i \leq T} \prod_{k: t_k < t_i} S(t_i|t_k; A_{k,i}) \times \sum_{j: t_j < t_i} \frac{f(t_i|t_j; A_{j,i})}{S(t_i|t_j; A_{j,i})} \tag{8}$$

Eq. (8) only considers infected nodes. We add the multiplicative survival term from Eq. (5) to include the information of the nodes which are not infected during the observation window and also replace the ratios in Eq. (8) with hazard functions,

$$\begin{aligned} f(t; A) = & \prod_{t_i \leq T} \prod_{t_m > T} S(T|t_i; A_{i,m}) \\ & \times \prod_{k: t_k < t_i} S(t_i|t_k; A_{k,i}) \sum_{j: t_j < t_i} H(t_i, t_j; A_{j,i}) \end{aligned} \tag{9}$$

Assuming independent cascades, the likelihood of a set of cascades $C = \{t^1, \dots, t^M\}$ is the product of the likelihoods of the individual cascades in Eq. (10)

$$\prod_{t^c \in C} f(t^c; A) \tag{10}$$

• Social Network inferring. We aim to find the strength $A_{j,i}$ of every pair of nodes such that the likelihood of an observed set of cascades $C = \{t^1, \dots, t^M\}$ is maximized. Thus, we can solve the following optimization problem (11) to obtain the latent social network from the given cascade data,

$$\begin{aligned} \min_{A} \quad & -\sum_{c \in C} logf(t^c; A) \\ s.t. \quad & A_{j,i} \geq 0, i, j = 1, \dots, N, i \neq j \end{aligned} \tag{11}$$

where $A := \{A_{i,j} | i, j = 1, \dots, n, i \neq j\}$ are the variables. The edges of the network are those pairs of nodes with transmission rates $A_{ij} > 0$.

B. *Social recommendation regularization term*

In the real world, it is usual to turn to friends for movie , music or book recommendations since we have confidence in the tastes of our friends. Due to the assumption that the closer the relationship between the users, the more similar preferences they have, we set the social regularization term in Eq. (12)

$$\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}\|U_i - U_j\|^2 \tag{12}$$

where N is the number of users, $A_{ij}, 1 \le i \le N, 1 \le j \le N$ is the inferred implicit social network matrix.

C. *Low-rank matrix factorization.*

An efficient and effective approach to recommender systems is to factorize the user-item rating matrix, and utilize the factorized user-specific and item-specific matrices to make further missing data prediction [20] [22] [24] [25] [34]. The premise behind a low-dimensional factor model is that there is only a small number of factors influencing the preferences, and that a user's preference vector is determined by how each factor applies to that user [22].

In this paper, we consider an $N \times M$ rating matrix R describing N users' numerical ratings on M items. A low-rank matrix factorization approach seeks to approximate the rating matrix R by a multiplication of K-rank factors, where $K << min(N, M)$, and the singular value decomposition method is traditionally utilized [20] to approximate a rating matrix R by minimizing $\frac{1}{2}\|R - U^T V\|_F^2$ where $U \in \mathbb{R}^{K\times N}, V \in \mathbb{R}^{K\times M}$, and $\|\cdot\|_F^2$ denotes the Frobenius norm. However, due to a large number of missing values contained in R, we only need to factorize the observed ratings in it. In order to avoid overfitting, two regularization terms are added. Hence, objective function is changed to

$$\min_{U,V} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\alpha}{2}\|U\|_F^2 + \frac{\beta}{2}\|V\|_F^2 \tag{13}$$

where $\alpha, \beta > 0$, and I_{ij} is the indicator function that is equal to 1 if user U_i rated item V_j and equal to 0 otherwise.

The optimization problem in Eq. (13) minimizes the sum-of-squared-errors objective function with quadratic regularization terms.

According to the above discussions, the implicit social network inferring term, the social recommendation regularization term and the traditional matrix factorization-based recommender system term, are unified in our IDSR model for implicit dynamic social recommendation.

V. Algorithm

In this section, we use a coordinate descent algorithm to solve the IDSR model. We first introduce the algorithm in Section V-A, and analyze its convergence and time complexity in Section V-B and Section V-C respectively.

Algorithm 1 Implicit Dynamic Social Recommendation

```
Require:
    $C = \{t^1, \dots, t^M\}$: Cascade data;
    $R \in \mathbb{R}^{N\times M}$: Rating data;
    $\lambda_1, \lambda_2, \lambda_3$ and $\lambda_4$: Tradeoff parameters;
    $K$: Latent feature space dimension parameter.
Ensure:
    $U \in \mathbb{R}^{K\times N}$: User matrix;
    $V \in \mathbb{R}^{K\times M}$: Item matrix;
    $A \in \mathbb{R}^{N\times N}$: Social network matrix.
Initialize: $U_0 = E, V_0 = E, A_0 = E$ where $E$ is the identity matrix;
while Not convergent do
    $A_{t+1} \leftarrow$ Apply Eq. (16) with $U_t$ to update $A_t$;
    $U_{t+1} \leftarrow$ Apply Eq. (20) with $A_{t+1}, V_t$ to update $U_t$;
    $V_{t+1} \leftarrow$ Apply Eq. (24) with $A_{t+1}, U_{t+1}$ to update $V_t$;
    $t \leftarrow t + 1$;
end while
return  $A^* = A_t$; $U^* = U_t$, $V^* = V_t$.
```

A. *Implicit Dynamic Social Recommendation algorithm*

In the coordinate descent algorithm, we iteratively update one variable by fixing the remaining two variables. The steps will be repeated until convergence. Algorithm 1 summarizes the steps.

- **update A with fixed U and V**

With fixed U, V, model (1) degrades to problem (14)

$$\min_{A\ge 0} -\frac{1}{2}\sum_{c\in C} logf(t^c, A) + \frac{\lambda_1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{i,j}\|U_i - U_j\|^2 \tag{14}$$

where $f(t^c, A)$ is shown in Eq. (9) with t replaced by t^c. Problem (14) is convex and consistent [23]. For the fixed i and j, problem (14) degrades to problem (15)

$$\begin{aligned}\min_{A\ge 0} \frac{1}{2} &\sum_{c:t_i^c \le T, t_j^c > T} (T - t_i^c)A_{ij} + \frac{\lambda_1}{2}\|U_i - U_j\|^2 A_{ij} \\ +\frac{1}{2} &\sum_{c:t_i^c < t_j^c, t_j^c \le T} ((t_j^c - t_i^c)A_{ij} - logA_{ij})\end{aligned} \tag{15}$$

To set the derivative of A_{ij} equal to 0, we can obtain the update strategy of A_{ij}^{t+1} as Eq. (16)

$$A_{ij}^* = \frac{2N_{ij}^{\tilde{C}}}{\sum_{c\in\hat{C}_{ij}}(T - t_i^c) + \sum_{c\in\tilde{C}_{ij}}(t_j^c - t_i^c) + \lambda_1\|U_i - U_j\|^2} \tag{16}$$

where $\hat{C}_{ij} = \{c \in C : t_i^c \le T, t_j^c > T\}$, $\tilde{C}_{ij} = \{c \in C : t_i^c < t_j^c, t_j^c \le T\}$ and $N_{ij}^{\tilde{C}}$ is the number of the cascades in $\tilde{C}_{ij}$ for the fixed i, j.

- **update U with fixed A and V**

With fixed A, V, model (1) degrades to problem (17)

$$\begin{aligned}\min_{U\geq 0}\ & \frac{\lambda_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-U_iV_j^\top)^2+\frac{\lambda_3}{2}\|U\|_F^2\\ &+\frac{\lambda_1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{i,j}\|U_i-U_j\|^2\end{aligned} \tag{17}$$

Problem (17) is convex and consistent with variable U. It is equal to problem (18)

$$\begin{aligned}\min_{U\geq 0}\ & \frac{\lambda_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-\sum_{p=1}^{K}U_{ip}V_{jp})^2+\frac{\lambda_3}{2}\sum_{i=1}^{N}\sum_{p=1}^{K}U_{ip}^2\\ &+\frac{\lambda_1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{p=1}^{K} A_{ij}(U_{ip}-U_{jp})^2\end{aligned} \tag{18}$$

For the fixed i and p, the derivative of U_{ip} is in Eq. (19)

$$\begin{aligned}\frac{\nabla F}{\nabla U_{ip}}=&-\lambda_2\sum_{j=1}^{M} I_{ij}V_{jp}(R_{ij}-\sum_{q=1}^{K}U_{iq}V_{jq})+\lambda_3 U_{ip}\\ &+\lambda_1\sum_{j=1}^{N} A_{ij}(U_{ip}-U_{jp})\end{aligned} \tag{19}$$

Make the derivative equal to zero, we can obtain

$$U_{ip}^{*}=\frac{\lambda_2\sum_{j=1}^{M} I_{ij}V_{jp}(R_{ij}-\sum_{q=1,q\neq p}^{K}U_{iq}V_{jq})+\lambda_1\sum_{j=1}^{N}A_{ij}U_{jk}}{\lambda_2\sum_{j=1}^{M}I_{ij}V_{jp}^2+\lambda_3+\lambda_1\sum_{j=1}^{N}A_{ij}} \tag{20}$$

- **update V with fixed A and U**

With fixed A, U, model (1) degrades to problem (21)

$$\min_{V\geq 0}\ \frac{\lambda_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-U_iV_j^\top)^2+\frac{\lambda_4}{2}\|V\|_F^2 \tag{21}$$

Problem (21) is equal to problem (22)

$$\min_{V\geq 0}\frac{\lambda_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-\sum_{p=1}^{K}U_{ip}V_{jp})^2+\frac{\lambda_4}{2}\sum_{j=1}^{M}\sum_{p=1}^{K}V_{jp}^2 \tag{22}$$

For fixed j and p, the derivative of V_{jp} is

$$\frac{\nabla F}{\nabla V_{jp}}=-\lambda_2\sum_{i=1}^{N} I_{ij}U_{ip}(R_{ij}-\sum_{q=1}^{K}U_{iq}V_{jq})+\lambda_4 V_{jp} \tag{23}$$

Make the derivative equal to zero, we can obtain

$$V_{jp}^{*}=\frac{\lambda_2\sum_{i=1}^{N} I_{ij}U_{ip}(R_{ij}-\sum_{q=1,q\neq p}^{K}U_{iq}V_{jq})}{\lambda_4+\lambda_2\sum_{i=1}^{N}I_{ij}U_{ip}^2} \tag{24}$$

B. Convergence Analysis

Algorithm 1 is an typical coordinate descent algorithm, which will convergent to the local optimums. When updating A, U and V, in each step, the degraded optimization problem, which only contain one variable, is a convex optimization. We can obtain the temporary optimum in each step, which would make the objective function decrease persistently until convergent to one of its local optimums.

C. Complexity Analysis

In Algorithm 1, IDSR takes constant time for the initialization (line 1). When update social network matrix A, suppose the maximum number of cascades, in which $t_i < T$ contained, is I_c, it takes $O(I_cN^2)$ (line 3). When update matrix U, since rating matrix R is very sparse, it takes $O(\epsilon M + N)$ for each entry, where ϵ is the sparse rate. Thus, this part takes $O(\epsilon MNK + N^2K)$ (line 4). When update matrix V, it takes $O(\epsilon N)$ for each entry, so the actual time expense of line 5 is $O(\epsilon NMK)$. To sum up, the total worst-case time complexity of Algorithm 1 is $O(\epsilon MNKC + N^2KC + N^2I_cC)$ where C is the maximum number of iterations until convergence.

VI. Experiments

In the following, we first introduce the details of the datasets and metrics we use in Section VI-A and Section VI-B respectively. We then introduce the experimental setup and benchmark methods. Finally, we describe the comparison in detail with three aspects. Firstly, we evaluate the performance of the proposed algorithm with respect to MAEs and RMSEs compared with the five state-of-the-art benchmark methods in Section VI-D. Secondly, we explore the performance of the IDSR method in terms of data freshness (time distances between the target ratings and historical ratings) in Section VI-E, and finally we explore the performance of IDSR method in terms of observation window T in Section VI-F. All experiments are conducted on a Windows 8 machine with 3.00GHz CPU and 8GB memory.

A. Datasets

We evaluate the performance of the proposed IDSR model on three real-world datasets, *i.e.,* Zomato [1], MovieLens [2] and Douban movie [3] dataset. The statistics of the datasets are show in Table I. The sparsity shown in the table is defined as $1-\frac{\text{nonzero entries}}{\text{total entries}}$.

1) Zomato[1]*:* is a restaurant search and discovery service website founded in 2008. It operates in many countries, including the United States, Australia, India etc. It features restaurant information such as scanned menus and photos sourced by local street teams, as well as user reviews and ratings.

We collect the ratings (1-5) data and the time points of the posted reviews of 5336 users for the most popular 1012 restaurants in Sydney from September 2008 to April 2016. We also collect information on the existing social networks provided by the website, and the followers' and followees' data, which will be used in some of the benchmark methods for comparison. This dataset is very sparse with a 0.0031 sparsity value.

[1] https://www.zomato.com

[2] http://www.grouplens.org/node/73

[3] http://movie.douban.com

Table I
STATISTICS OF DATASETS

Dataset	Douban	MovieLens	Zomato
# of users	249408	943	5336
# of items	100	1682	1012
# of ratings	877572	100000	38367
sparsity	0.0330	0.0630	0.0031
Average cascade length	8755.70	59.45	38.06
Longest cascade length	29234	583	302
Shortest cascade length	480	1	3
Time interval	2015.1.1 -2016.4.30	1997.9.17 -1998.4.22	2008.9.21 -2016.4.30

2) MovieLens data[2] *:* is a web-based research recommender system that debuted in the autumn of 1997. Each week, hundreds of users visit MovieLens to rate and receive recommendations for movies. MovieLens is a popular dataset used in social recommendation researches. It was collected through the MovieLens website[4] during the seven-month period from September 19th, 1997 to April 22nd, 1998. The dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies and each user has rated at least 20 movies. This data has been cleansed that users who had less than 20 ratings or did not provide complete demographic information were removed from this data set. Detailed descriptions of the data file can be found in [9]. And the sparsity of this dataset is 0.0630.

3) Douban[3]*:* is one of the most popular rating website in China. It comprises several parts: Douban Movie, Douban Read and Douban Music, etc. Douban Movie provides the latest movie information, and users can record the movies they wish to watch and rate them after they have watched them. They can also share their reviews with their friends.

We collect the ratings and the time stamps of the rating data on the 100 most popular movies in Douban. It includes 877572 ratings (1-5) and the time stamps of 249408 users from January 2015 to April 2016. In this dataset, the data of time stamps are very dense, where the average length of cascades is 8755.7 with the longest one is 29234. The sparsity of this dataset is 0.0330.

B. Metrics

Recommender systems research has used several types of measures to evaluate the quality of recommender systems. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are the most popular metrics used to evaluate the deviation of recommendations from their true user-specified values. Specifically, RMSE and MAE are defined in Eqs. (25) and (26) respectively.

$$RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i,j} (R_{ij} - \hat{R}_{ij})^2} \tag{25}$$

$$MAE = \frac{1}{N_{test}} \sum_{i,j} |R_{ij} - \hat{R}_{ij}| \tag{26}$$

where R_{ij} denotes the rating user i gives to item j, $\hat{R}_{ij}$ denotes the rating user i gives to item j as predicted by the method, and N_{test} denotes the number of ratings in test set. From the definitions, we can see RMSE gives higher weights to larger errors as the errors are squared before taking their average. It is always larger or equal to MAE. The lower the values of RMSE and MAE, the more accurately the recommendation engine predicts.

C. Experimental setup

Training and testing data setup In many studies [26] [13], the training and testing data are randomly chosen for the experiments. But this is unsuitable for the evaluation of dynamic recommendation in which we can use only historical data but not future data for current predictions in real applications. Following [14] [29], we split the training and testing data based on time and evaluate the performance. Specifically, we sort the entire data set in normal time order, and use the earlier part as the training set to adjust all parameters in the recommendation algorithm. We run algorithms on the test set then and generate the estimated rating for each user-item pair and compare the estimated ratings with real rating to calculate RMSEs and MAEs.

Parameter settings focus on the meanings and settings of the parameters. We implement the proposed algorithm and compare it with benchmark methods under these parameters. Parameter K controls the dimension of the latent feature space. If K is too small, it is difficult for the model to make a distinction among 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. Here we choose the best value of K through grid search from 5 to 50 in steps of 5. The tradeoff parameters $\lambda_1, \lambda_2, \lambda_3$ and λ_4 in Eq. (1) control the weights of each part in the model and we choose the best value for them through grid search in $[10^{-8}, 10^{8}]$ with the step of 10^2.

Benchmark methods To evaluate the performance of our method, we compare the proposed IDSR with the following five approaches.

- **BaseMF** is the baseline matrix factorization approach proposed in [25], which does not take the social network into account;

- **SocialMF** [12] improves the recommendation accuracy of BaseMF by taking into account the social trust between users. It always uses all social links available in the dataset;

- **SR1**$_{pcc}$ [20] imports average-based regularization in social recommendation regularization term to form the social recommendation model;

[4] https://movielens.org/

Table II
PERFORMANCE COMPARISONS

Dataset	Training	Metrics	BaseMF	SocialMF	SR1$_{pcc}$	SR2$_{pcc}$	SR^{u+-}_{i+-}	IDSR
Douban	80%	MAE	0.9260	0.8019	0.7946	0.7831	0.7049	**0.6609**
		Improve	28.64%	17.59%	16.83%	15.61%	6.25%	
		RMSE	1.0525	1.0308	1.0257	1.0083	0.9710	**0.9484**
		Improve	9.90%	8.00%	7.54%	5.95%	2.33%	
	60%	MAE	0.9469	0.8618	0.8622	0.8497	0.7637	**0.7046**
		Improve	25.59%	18.24%	18.28%	17.08%	7.74%	
		RMSE	1.1059	1.0893	1.0414	1.0336	1.0641	**0.9500**
		Improve	14.09%	12.78%	8.77%	8.08%	10.72%	
Movielens	80%	MAE	1.0187	0.8984	0.8814	0.8772	0.7849	**0.7375**
		Improve	27.61%	17.91%	16.33%	15.93%	6.04%	
		RMSE	1.0964	1.0505	1.0587	1.0331	0.9909	**0.9511**
		Improve	13.25%	9.46%	10.16%	7.93%	4.01%	
	60%	MAE	1.0464	0.9386	0.9452	0.9335	0.8981	**0.8284**
		Improve	20.84%	11.75%	12.36%	11.26%	7.77%	
		RMSE	1.1962	1.1660	1.1536	1.1316	1.1297	**1.0067**
		Improve	15.84%	13.66%	12.73%	11.03%	10.88%	
Zomato	80%	MAE	0.9773	0.8505	0.8537	0.8317	0.7489	**0.7038**
		Improve	27.99%	17.25%	17.56%	15.38%	6.02%	
		RMSE	1.1027	1.0558	1.0333	1.0249	0.9870	**0.9319**
		Improve	15.49%	11.73%	9.81%	9.07%	5.58%	
	60%	MAE	0.9958	0.9117	0.9292	0.8948	0.8150	**0.7402**
		Improve	25.67%	18.81%	20.34%	17.28%	9.18%	
		RMSE	1.1136	1.1019	1.1042	1.0934	1.0562	**0.9934**
		Improve	10.79%	9.84%	10.03%	9.14%	5.95%	

- **SR2**$_{pcc}$ [20] is another social recommendation method where individual-based regularization is imported as its social recommendation regularization term;

- **SR**$^{u+-}_{i+-}$ [16] is an implicit social network recommendation approach which uses both implicit similar and dissimilar user information as well as similar and dissimilar item information in its recommendation model.

D. Comparisons

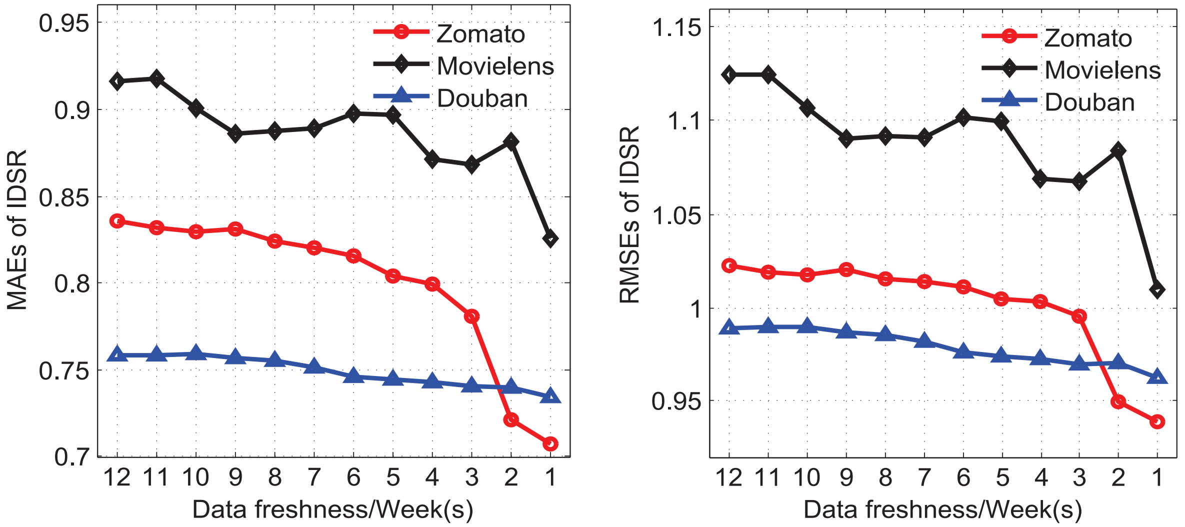
In this section, we compare the recommendation results of the proposed approach IDSR with all benchmark methods to evaluate the effectiveness.

For the three datasets, we use different training data settings (80% and 60%) to test the algorithms. Training data 80%, for example, means we use the first 80% of the cascade and rating data as the training data to predict the last 20% of the ratings. The experimental results are shown in Table II. For the MovieLens and Douban datasets, social network data are not provided, so we choose the five most similar users for each user as her/his friends and build the explicit social network which is used in the benchmark methods.

From the results, we can observe that our method consistently outperforms the other approaches in all the settings of the three datasets. Firstly, we can see our algorithm generates significantly better results than the BaseMF method which does not take the social network into account. There is an average 26.05% improvement in MAEs (max 28.63%, min 20.84%), and an average 13.23% improvement in RMSEs (max15.84%, min 9.90%) on all the three datasets. This observation illustrates that employing implicit user social information helps increase the recommendation quality.

Furthermore, our method performs better than the state-of-the-art explicit social recommendation algorithms SocialMF, SR1$_{pcc}$ and SR2$_{pcc}$ on all the three datasets. Specifically, the proposed IDSR achieves 16.93% improvement on average (max 18.81%, min 11.75%) in MAEs and 10.91% improvement on average (max 13.66%, min 8.00%) in RMSEs compared with SocialMF. Compared with the other two explicit social recommendation approaches SR1$_{pcc}$ and SR2$_{pcc}$, our method achieves on average 16.95% (max 20.34%, min 12.36%) and 15.42% (max 17.28%, min 11.26%) improvement in MAEs respectively and an average 9.84% (max 12.73%, min 7.54%) and 8.54% (max 11.03%, min 5.95) improvement in RMSEs respectively. These observations demonstrate that in many situations, implicit social networks can much more accurately describe a user's preferences compared with explicit social networks and avoid the noise contained in the explicit social networks.

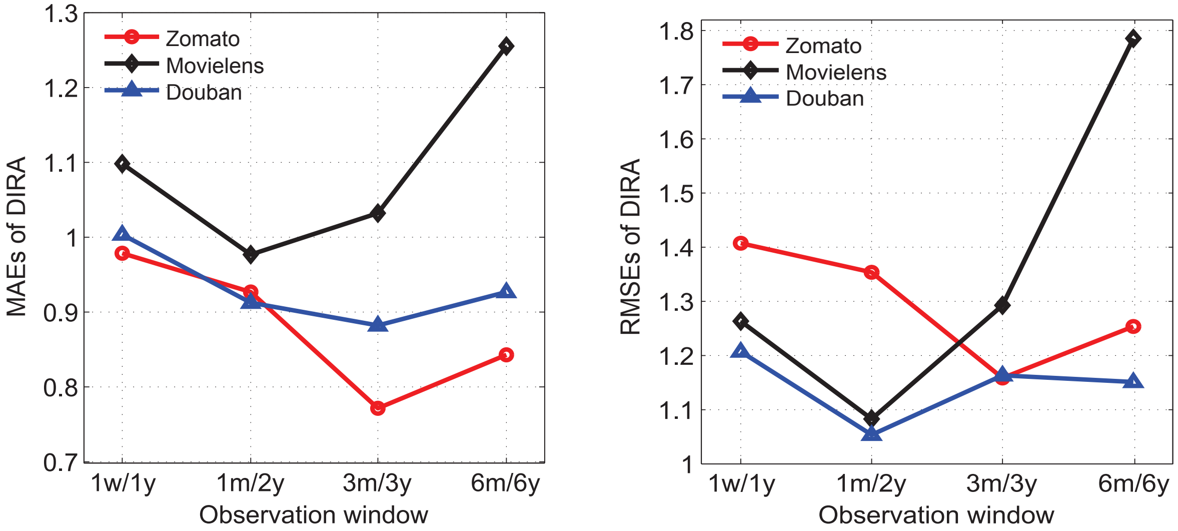
Compared with the implicit social recommendation approach SR^{u+-}_{i+-}, our algorithm also achieves better results an average 7.17% (max 9.18%, min 6.02%) improvement in MAEs and an average 6.58% (max 10.88%, min 2.33%) improvement in RMSEs. This observation shows that our implicit dynamic social recommendation model also outperforms the state-of-the-art static implicit social recommendation systems.



(a) MAE

(b) RMSE

Figure 2. MAEs and RMSEs obtained by IDSR algorithm in terms of different data freshness (time distances between the target ratings and historical ratings) values on the three datasets.



(a) MAE

(b) RMSE

Figure 3. MAEs and RMSEs obtained by IDSR algorithm in terms of observation window on the three datasets.

E. Performance in terms of data freshness

In this section, we explore the performance of the proposed model in terms of data freshness. All the three datasets contain abundant rating records and cascade data which are collected over a reasonable period of time and are different in composition and dynamic in nature. We define the time distances between the target ratings and historical ratings as *data freshness*. We manually assign twelve different data freshness values by classify the cascades into multiple phases, i.e., within 1 week, within 2 weeks,..., within 12 weeks.

Figure 2 shows the MAEs and RMSEs of our IDSR using the data in different setting of data freshness on the three datasets. It is clear that the more fresh the data, the better the performance. Such observation means using closer data sources to train recommender system, the difference between the original ratings and the predicted ones becomes smaller.

F. Performance in terms of observation window

In this section, we explore the performance of the IDSR model in terms of observation window T. We use different sized training sets. For MovieLens and Douban datasets, we rebuild four new training datasets based on the original one. In the new training datasets, the rating data were proposed within 1 week, 1 month, 3 months and 6 months respectively while for the Zomato dataset we rebuild four new training datasets in 1 year, 2 years, 3 years and 6 years due to its high sparsity.

Figure 3 shows the results of IDSR with different observation windows. We can see in most cases, MAEs and RMSEs are large when the observation windows are very short. With longer observation windows, MAEs and RMSEs decrease at first and then increase again, which means the predicted results become better at first and then become worse later. This is because the longer cascades may contain more noise which harms the predicted results. However, if the cascades are too short, they do not contain enough information to infer the whole latent social network, which also harms the algorithm performances.

VII. Conclusion

In this paper, we formulate a new implicit dynamic social recommendation problem where the inferred latent dynamic social network is used for enhancing the social recommendation performance. We proposed a new model IDSR which uses cascade data to infer implicit dynamic social networks. IDSR addresses the commonly existing preference drafting issues in real social recommendation studies, and identifies dynamic changes in the users closely by taking advantage of the information contained in time. This is different from most of the existing implicit social recommendation approaches which only use rating data to infer implicit social networks. The proposed IDSR model also undertakes the learning of social structures and rating predictions together in a unified optimization problem, rather than treating the learning of social networks and recommendations as two independent tasks. The experiments demonstrate that the proposed IDSR model outperforms the state-of-the-art models in both explicit and implicit social recommendation scenarios.

Acknowledgments

This work was supported by Australia ARC Discovery Project (No. DP140100545 and No. DP140102206), Australia Linkage Projects (No. LP150100671 and No. LP160100630), ARC Future Fellowship (No. FT130100746), and CSC Foundation (No. 201206410056). J. Wu is the corresponding author.

References

[1] C.-m. Au Yeung and T. Iwata. Strength of social influence in trust networks in product review sites. In *WSDM*, pages 495–504, 2011.

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

[3] H. Fang, Y. Bao, and J. Zhang. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. In *AAAI*, pages 30–36, 2014.

[4] S. Fazeli, B. Loni, A. Bellogin, H. Drachsler, and P. Sloep. Implicit vs. explicit trust in social matrix factorization. In *RecSys*, pages 317–320, 2014.

[5] L. Gao, J. Wu, H. Yang, Z. Qiao, C. Zhou, and Y. Hu. Semi-data-driven network coarsening. In *IJCAI*, pages 1483–1489, 2016.

[6] G. Guo, J. Zhang, D. Thalmann, A. Basu, and N. Yorke-Smith. From ratings to trust: an empirical study of implicit trust in recommender systems. In *SAC*, pages 248–253, 2014.

[7] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith. Etaf: An extended trust antecedents framework for trust prediction. In *ASONAM*, pages 540–547, 2014.

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

[9] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2015.

[10] Z. Huang, H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):116–142, 2004.

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

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

[13] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[14] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *WSDM*, pages 297–306, 2011.

[15] C. Lin, R. Xie, X. Guan, L. Li, and T. Li. Personalized news recommendation via implicit social experts. *Information Sciences*, 254:1–18, 2014.

[16] H. Ma. An experimental study on implicit social recommendation. In *SIGIR*, pages 73–82, 2013.

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

[18] H. Ma, I. King, and M. R. Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):29:1–29:19, 2011.

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

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

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

[22] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, pages 713–719, 2005.

[23] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697*, 2011.

[24] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, pages 880–887, 2008.

[25] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, volume 20, pages 1–8, 2011.

[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[27] R. E. Schafer. Statistical models and methods for lifetime data. *Technometrics*, 25(1):111–112, 1983.

[28] J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[29] X. Tang and J. Zhou. Dynamic personalized recommendation on sparse data. *IEEE Transactions on Knowledge and Data Engineering*, 25(12):2895–2899, 2013.

[30] X. Xin, I. King, H. Deng, and M. R. Lyu. A social recommendation framework based on multi-scale continuous conditional random fields. In *CIKM*, pages 1247–1256, 2009.

[31] X. Yang, Y. Guo, Y. Liu, and H. Steck. A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41:1–10, 2014.

[32] X. Yang, H. Steck, Y. Guo, and Y. Liu. On top-k recommendation using social networks. In *RecSys*, pages 67–74, 2012.

[33] X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *KDD*, pages 1267–1275, 2012.

[34] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *SIGIR*, pages 211–218, 2009.

[35] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang. Unsupervised feature learning from time series. In *IJCAI*, pages 2322–2328, 2016.

[36] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang, and X. Wu. Online learning from trapezoidal data streams. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 28(10):2709–2723, 2016.