Attentive Recurrent Social Recommendation

Peijie Sun
Hefei University of Technology
sun.hfut@gmail.com

Le Wu*
Hefei University of Technology
lewu.ustc@gmail.com

Meng Wang
Hefei University of Technology
eric.mengwang@gmail.com

ABSTRACT

Collaborative filtering (CF) is one of the most popular techniques for building recommender systems. To alleviate the data sparsity issue in CF, social recommendation has emerged by leveraging social influence among users for better recommendation performance. In these systems, uses' preferences over time are determined by their temporal dynamic interests as well as the general static interests. In the meantime, the complex interplay between users' internal interests and the social influence from the social network drives the evolution of users' preferences over time. Nevertheless, traditional approaches either neglected the social network structure for temporal recommendation or assumed a static social influence strength for static social recommendation. Thus, the problem of how to leverage social influence to enhance temporal social recommendation performance remains pretty much open. To this end, in this paper, we present an attentive recurrent network based approach for temporal social recommendation. In the proposed approach, we model users' complex dynamic and general static preferences over time by fusing social influence among users with two attention networks. Specifically, in the dynamic preference modeling process, we design a dynamic social aware recurrent neural network to capture users' complex latent interests over time, where a temporal attention network is proposed to learn the temporal social influence over time. In the general static preference modeling process, we characterize each user's static interest by introducing a static social attention network to model the stationary social influence among users. The output of the dynamic preferences and the static preferences are combined together in a unified end-to-end framework for the temporal social recommendation task. Finally, experimental results on two real-world datasets clearly show the superiority of our proposed model compared to the baselines.

CCS CONCEPTS

• **Information Systems** → Information System Applications;

KEYWORDS

Recommendation, Social Network, Attention, Social Influence, Recurrent Network

*Le Wu is the corresponding author.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210023

ACM Reference Format:
Peijie Sun, Le Wu*, and Meng Wang. 2018. Attentive Recurrent Social Recommendation. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3209978.3210023

1 INTRODUCTION

Collaborative filtering is one of the most successful ways to build recommender systems and has received significant success in the past decades [1, 21]. Specifically, it infers each user's interests by collecting the user-item interaction history without any content information. Among all models of CF, latent factor based models have received great success in both academia and industry. These models try to characterize both users and items in a same low latent space inferred from the historical user-item interaction matrix [25, 27]. Then, the predicted preference of a user to an item could be reduced to comparing users and items in the low latent space. Despite the huge success, in the real-world systems, a user usually rates or experiences a small set of items in these applications, the data sparsity issue remains a key challenge for enhancing recommendation performance [1].

Luckily, the emergence of online social networks greatly improves users' initiative on the Internet, such as *Facebook*, *Twitter*, online social product review platform *Epinions*, and location based social network *Gowalla*. In these social applications, users like to spread their preferences for items to their social connections (e.g., friends in a undirected social network and followers in a directed social network), and social influence effect is well recognized as a main factor in these platforms [14]. The social influence theory argues that, users are influenced by the social connections in the social network, leading to the homophily effect of similar preferences among social neighbors [3, 4]. Thus, by leveraging the social influence among users, social recommendation has become a popular way to tackle the data sparsity issue in traditional recommender systems and has been extensively studied [16, 17, 24, 34]. E.g., Jamali et al. proposed a social influence propagation based latent factor models for social recommendation [16], and Ma et al. designed a latent factor based model with the social regularization of users' interests [24]. In summary, these works focused on how to push the social influence theory among users in the recommendation process. Usually, the social influence strength was set equally for the social connections [16] or relied on a predefined static function [17, 24].

Successful as they are, these social recommender systems assumed a general static assumption of users' interests over time. In the real world social recommendation scenarios, users' preferences for items are not always stationary but evolve over time. In fact, time has been recognized as an important type of information for modeling the dynamics of users' preferences in traditional recommender systems [9, 40, 44]. Researchers proposed tensor factorization or temporal user latent factor based models to capture users' dynamic interests over time. Instead of capturing users' temporal interests, other researchers argued that each user's preference is composed of two parts: a general static interest that is stationary and does not evolve over time, and a complex dynamic interest that is easily influenced by the external environments. Thus, some models have been proposed to tackle the temporal recommendation problem by combining static and dynamic interest modeling [20, 42]. These models showed better performance by simultaneously modeling users' static interests for temporal recommendation. To summarize, all these temporal models relied on the variants of latent factor based models to capture users' dynamic interest. Nevertheless, the inherent reasons for users' preference evolution are complex and non-linear, which are hard to be captured by these linear shallow latent factor based models. Therefore, the performance of those approaches are still not satisfactory.

Recently, Recurrent Neural Network (RNN) based approaches have shown promising potentials for capturing complex temporal patterns for time series tasks, such as sentence generation [19], and acoustic modeling [30]. Some pioneering works attempted to introduce RNNs for temporal recommendation [26, 38, 47]. These RNN based recommendation models modeled the latent structure of each user at each time with a hidden state. These hidden states over time are learned from the recurrent neural networks to model the complex temporal patterns, which advance previous shallow temporal recommendation models. Despite the success of applying RNNs for temporal recommendation, to the best of our knowledge, few research works have attempted to tackle the temporal social recommendation task with RNNs. Indeed, it is a non-trivial task due to the following two key challenges in this process. First, as the RNN based models are good at modeling the complex dynamic user interests, how to design a model that unifies both users' dynamic interests and the static interests? Second, in both the dynamic user interest and static user interest modeling part, different social connections would have different social influence strengths on users. This social influence strength issue is more challenging in the dynamic user interest modeling process due to the interplay between users' dynamic interests and the social influence. The dynamics of the social influence strength would affect the connected users' preferences, and users' evolving preferences would also affect their influence strengths to their social connections. Thus, how to model the complex interplay between social influence and users' interests over time becomes another challenge.

To solve the above technical challenges, in this paper, we present an attentive recurrent network based approach for the temporal social recommendation task. In the proposed approach, we model users' complex dynamic and general static preferences over time by leveraging social influence among users with two attention networks. Specifically, we use a recurrent neural network model as the base model for dynamic interest modeling, where each user's dynamic preference over time could be introduced as a hidden temporal state in the neural network structure. To capture the interplay between social influence and user dynamic interest over time, we build a dynamic social attention module in each hidden state to measure the temporal social influence strength. The dynamic attention network could selectively choose the social connections that have large influence for each user at each time, and a social aware recurrent neural network is proposed to capture users' complex latent interests over time. In the general static preference modeling process, we augment each user's static interest part by introducing a static social attention module to model the stationary social influence among users. Thus, both users' complex dynamic interests and general static interests are fused in a unified framework with the attentive social modeling networks. We summarize the contributions of this paper as follows.

- We propose the problem of temporal social recommendation. We argue that it is important to take users' complex dynamic interest and general static interest into consideration, where both the dynamic interest modeling part and the general interest modeling part needs to leverage the social influence in social networks.
- We propose RNN based structure to capture users' complex dynamic interest and design a dynamic social attention network to measure the dynamic social influence of social connections over time. We also extend the static user interest modeling part with a static social attention network to measure the stationary social influence among users. These two parts are fused together for the temporal social recommendation task.
- We conduct extensive experimental results on two real-world datasets. The experimental results clearly demonstrate the superiority of our proposed model compared to the baselines.

2 RELATED WORK

We summarize the related work into the following three categories: temporal recommendation, social recommendation and the attention mechanism.

2.1 Temporal Recommendation

Collaborative Filtering (CF) is one of the most popular techniques to build recommender systems by utilizing the collective behavior of users without any content information [12, 13, 21, 41]. Latent factor based models have dominated CF due to their relatively high performance in many CF tasks [25, 27, 39]. Most existing recommendation models neglected the time information in recommendation process, with the general static assumption of users' interests over time, i.e., the latent factor of each user is the same at each time slice [25, 27]. In the real world, users' preferences to items are not static but change over time. Thus, it is important to take the temporal dynamics of users' interests in the recommendation process [9, 18, 44]. Xiong et al. proposed to treat time as an additional dimension and designed a tensor factorization approach to capture the temporal dynamics of users' preferences over time [44]. Other works have expanded the latent factor based models to characterize the dependency and transition between users' current latent vector and that of next time period by manual feature engineering [9, 40]. Instead of capturing users' interests with either the general static interest or the dynamic interest, some studies argued that users' preferences are composed of two parts: a global static preference that do not change over time (e.g., the preference that is related to each user's gender and birthplace), and a temporal local preference that evolves over time (e.g., the preference that is related to the currently popular products). Given this assumption, some studies combined the static part and the dynamic part in a unified framework to further improve temporal recommendation performance [20, 28, 38].

In fact, all the above proposed temporal recommendation models relied on shallow linear latent factor based models and relied heavily on the manual efforts to define the temporal evolution patterns over time. Thus, they are hard to capture the inherent complex relationships of users' dynamic preferences. Recently, with the success of recurrent network based approaches for many temporal data, some pioneering works adapted the RNNs for temporal recommendation [26, 38, 47]. E.g., a dynamic recurrent basket based model is proposed to capture the dynamic preferences of users over time [47]. Instead of capturing users with temporal states, recurrent recommender network is proposed to endow both users and items with temporal hidden states with a recurrent neural networks [38]. Pei et al. proposed an attention-gated recurrent network to selectively memorize the previous time steps that are closely related to the current user interest [26]. Our work differs from these works by modeling the complex interplay between social influence and user interests in temporal social recommendation.

2.2 Social Influence and Social Recommendation

By leveraging the social network information, social recommendation provides an effective approach to alleviate data sparsity and improve recommendation performance. The underlying reason for social recommendation originates from the social influence theory, which states that users' behaviors are influenced or affected by others, leading to the similar behaviors (preferences) between socially connected users [2, 14, 36]. Social influence is a driving force for the prosperity of social platforms, and has a broad range of applications, such as node importance ranking [37] and social influence maximization [6]. In all these social network based applications, social influence strength modeling is a central problem [10, 11]. E.g., Goyal et al. designed a model to calculate influence strength from users' historical behaviors [11]. Various social recommendation algorithms have been proposed by pushing the social influence theory in the modeling process [16, 17, 24, 46]. E.g., Jamali et al. designed a social influence propagation based model in latent based recommendation models [16]. Ma et al. introduced a social correlation term to force similar users to have similar latent preferences [24]. All these works explicitly or implicitly modeled the social influence among users, where the social influence strength is assumed equal among social neighbors or with a simple metric from other sources (e.g., the strength between their interactions in the past). Despite the potentials of RNNs for temporal modeling, nevertheless, to the best of our knowledge, few has explored the neural networks for temporal social recommendation. In this paper, we try to model the interplay between users' temporal interest and the dynamic social influence in the social recommender systems.

2.3 Attention Mechanism

Our proposed technique is closely related to the attention mechanism that is widely adopted in neural network based approaches. The attention mechanism originates from the neural science studies by empirically demonstrating that human usually focus on specific parts of the input rather than using all available information [15]. In an attention based model, it automatically models and selects pertinent piece of information with the attentive weights from a set of inputs, with higher (lower) weights indicate the corresponding inputs more informative to generate the outputs. While attention is widely used for neural network based tasks, such as image captioning [45] and machine translation [33], it has recently been used in recommendation tasks [5, 26, 43]. E.g., an interacting attention-gated recurrent network is proposed for recommendation, which adopts the attention model to measure the relevance of each time [26]. Chen et al. designed an attentive collaborative filtering for multimedia recommendation with item and component level attention [5]. Intuitively, this attention technique could easily transferred to the social influence modeling task. In social influence modeling, each time a user decides to select an item, she would not take all social neighbors' opinions equally. Instead, she selects from informative friends and aggregates the attentive influence strengths. To the best of our knowledge, our proposed model is one of the first few attempts that adopts the attention networks for social influence strength modeling in social recommendation.

3 PROBLEM DEFINITION AND PRELIMINARIES

3.1 Problem Definition

In an online social service platform, there are a set of users U ($|U| = M$) and a set of items V ($|V| = N$). Users connect with each other to form a social network $S \in \mathbb{R}^{M \times M}$, with $s_{ba} = 1$ denotes that a follows b, otherwise it equals 0. Besides, users show preferences to items in this platform over time. Specifically, we represent users' preferences at time t as a matrix $R^t \in \mathbb{R}^{M \times N}$. As users usually implicitly express their behaviors of action or inaction (e.g., buy or not buy, like or not), in this paper we consider the more practical problem with implicit feedback of users. If user a likes item i at time t, then $r^t_{ai} = 1$. Otherwise it equals 0 indicating this user does not show any preference of the item at that time. As we consider the temporal evolution of users' preferences to items over time, we summarize users' preferences over time as a matrix sequence $R = [R^1, R^2, ..., R^T]$. Without confusion, we use a, b, c to represent users and i,j, k to denote items. Then, the problem we study in this paper could be defined as:

Definition 1. ***[Temporal Social Recommendation]*** *Given a user set U, an item set V, with user-user social network matrix S and user-item preference sequences from time 1 to time T as: $R = [R^1, R^2, ..., R^T]$, our goal is to predict each user's consumption behavior $\hat{R}^{T+1}$ at time $T+1$.*

3.2 Preliminary

Traditionally, different models have been proposed to tackle the time-series data, such as the Hidden Markov Model [7], and Conditional Random Fields [23]. The commonality of these models is that, they need to assume intimate knowledge of the dataset and explicitly define the correlation of time series in the model design process. However, the underlying reasons for the temporal changes are complex, and could not well captured by naive human feature engineering. Recently, the recurrent neural network based models provide an elegant way to model time-series data. Among all RNN based models, LSTM works tremendously well on a large variety of tasks, and are now widely used [30, 33]. Thus, we also consider to adopt the LSTM as a base module for modeling users' complex dynamic behaviors over time. Similar as many RNN modules, LSTMs have the form of a chain of repeating modules in neural networks. Instead of characterizing the repeating module with a simple structure (e.g., a single Tanh layer), LSTMs have a cell state that can remove or add information. Next, we introduce the key steps of LSTMs.

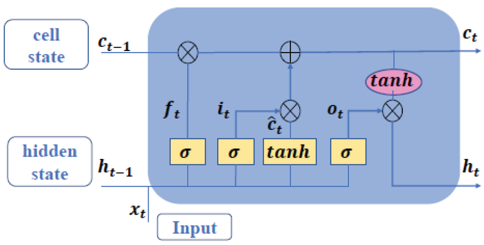


Figure 1: The overall architecture of LSTM.

Given an input x_t at time t with a fixed size of dimension, LSTM first decides what information to throw away with a forget layer as:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t]), \tag{1}$$

where h_{t-1} denotes the hidden state at the previous time $t-1$. Without confusion, we omit the bias terms in the following equations for simplicity. The next step is to decide what information to store based on a cell state as:

$$\begin{aligned} i_t &= \sigma(W_i \times [h_{t-1}, x_t]) \\ \widetilde{c}_t &= tanh(W_c \times [h_{t-1}, x_t]) \\ c_t &= f_t \times c_{t-1} + i_t \times \widetilde{c}_t, \end{aligned} \tag{2}$$

where c_t is the new cell state that forgets parts of previous state c_{t-1} and remembers some new parts of input data.

Given the cell state, the updated hidden state h_t is defined as

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t]) \\ h_t &= o_t \times tanh(c_t). \end{aligned} \tag{3}$$

Fig 1 shows the module of LSTM. By combining Eq.(1), Eq.(2), and Eq.(3), for each time t, the updated hidden state h_t in LSTM could be summarized as:

$$h_t = f_{LSTM}([h_{(t-1)}, x_t]). \tag{4}$$

In the following of this paper, we use f_{LSTM} to denote a LSTM module, and Θ_{LSTM} to summarize the parameters in this LSTM module.

4 THE PROPOSED MODEL

In this section, we first present the overall framework of our proposed model *A*ttentive *R*ecurrent *S*ocial r*E*commendation (ARSE) for temporal social recommendation. The proposed framework is composed of two parts: a complex *D*ynamic ARSE (DARSE) part that captures the dynamic preferences of users over time, and a general *S*tatic ARSE (SARSE) part that shows users' fixed interests that are stationary over time. Then, we will introduce these two parts in detail. After that, we show the model learning process of ARSE, which jointly optimizes the parameters in DARSE and SARSE in a tightly manner. For ease of explanation, Table 1 lists the mathematical notations used in this paper.

Table 1: Mathematical Notations

Notations	Description
U	Userset, $\lvert U \rvert = M$
V	Itemset, $\lvert V \rvert = N$
a,b,c,u	User
i,j,k,v	Item
$R^t \in \mathbb{R}^{M \times N}$	Rating matrix at time t
$S \in \mathbb{R}^{M \times M}$	Social network matrix, with s_{ba} denotes whether a follows b
$L_a^t \in V$	The item list that a likes at time t, $L_a^t = [i : \lvert r_{ai}^t = 1]$
$Q \in \mathbb{R}^{D \times N}$	Item latent matrix in the dynamic latent space
$W \in \mathbb{R}^{D \times N}$	Item latent matrix in the static latent space
$P \in \mathbb{R}^{D \times N}$	User base latent matrix in the static latent space
q_i	The dynamic embedding of item i in the dynamic latent space
w_i	The static embedding of item i in the static latent space
p_a	The static embedding of user a in the static latent space
x_a^t	The input vector of user a at time t
h_a^t	The dynamic latent vector of a at time t
α_{ab}^t	The dynamic influence strength of b to a at time t
β_{ab}	The static influence strength of b to a

4.1 The General Framework

Overall Prediction Function. In a temporal social recommender system, a natural assumption is that users' latent preferences over items are dynamic over time. To capture the time-evolving dynamic property of users' preference, we use a latent vector h_a^t to denote each user a's latent preference at each time t. Besides, we argue that even though a user's state can be time-varying, there is still some stationary components of each user such as the user profile, long-term preference. We use $\widetilde{p}_a$ to denote a's static latent vector. Then, the predicted preference $\hat{r}_{ai}^t$ of user a to item i at time t, could be predicted a combination of the dynamic effect and the static effect:

$$\hat{r}_{ai}^t = \hat{r}_{D,ai}^t + \hat{r}_{S,ai} = q_i' \times h_a^t + w_i' \times \widetilde{p}_a, \tag{5}$$

where $\hat{r}_{D,ai}^t$ denotes the predicted dynamic preference of user a to item i at time t, and $\hat{r}_{S,ai}$ is the predicted static preference score that does not evolve over time. q_i is the i-th column of Q that denotes the latent vector of item i in the dynamic latent space. Similarly, w_i is the i-th column of W that denotes the latent vector of item i in the static item space. We use two latent matrices to characterize items, as the dynamic effect and the static effect capture different latent characteristics of items. h_a^t denotes a's complex dynamic interest at time t, and $\widetilde{p}_a$ is a static latent factor of a that is stationary over time.

ARSE is such a neural network structure that models users' preferences over time with the prediction function in Eq.(5). The overall structure of ARSE is shown in Fig. 2, where the left part shows DARSE that captures the dynamic effect, and the right part depicts SARSE part that captures the static effect. In each part, the user-user social network is sent to a social attention layer to get the influence strength of users. Specifically, given user a, item i, and time t, we use α_{ab}^t to denote the dynamic social influence degree of user b to user a ($s_{ba} = 1$), and β_{ab} to denote the static social influence degree of b to a. A naive idea is to set $\alpha_{ab}^t = \beta_{ab} = \frac{1}{|S_a|}$, denoting each social connection influences user a equally over time. This naive social influence strength is widely used in many social recommender systems [16]. However, it fails

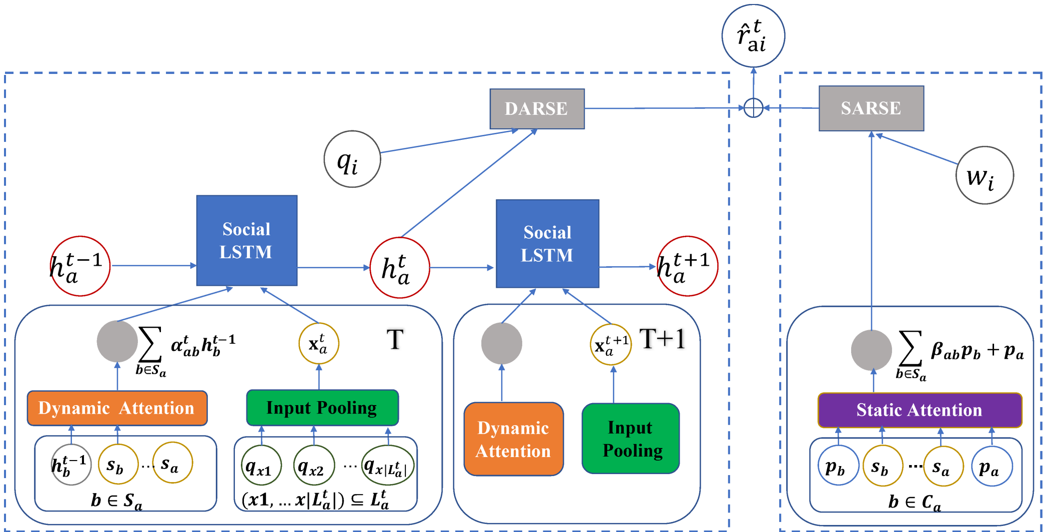


Figure 2: The overall architecture of ARSE, where the left part models the dynamic effect and the right part shows the static effect.

to consider the influence strength of different social connections. Accurately modeling the social influence strength is crucial for better social recommendation performance. Hence, we borrow the ideas of attention mechanism and design two attention networks to model the static and dynamic social influence strengths. Specifically, the static social influence β_{ab} is learned from a static social attention network. In DARSE, the dynamic influence strength α^t_{ab} is learned from a dynamic attention network with users' preferences over time, and the learned attention scores are summarized with social connections' latent vectors to obtain each user's future dynamic latent vector h^{t+1}_a of each user. In this way, we can accurately model the interplay between users' dynamic interests and the evolving social influence.

Thus, the proposed framework unifies the strength of classical latent factor based models to capture users' static preferences and the recurrent neural networks that capture the non-linear complex user interest drift over time. In the meantime, the two social attention parts in this framework could alleviate the data sparsity in recommendation and adaptively select the influential connections, thus further improve social recommendation performance. Next, we would introduce the two parts of ARSE in detail.

4.2 Dynamic Attentive Social Recurrent Recommendation

In the DARSE part, we adapt an LSTM structure to capture each user's complex temporal latent vector over time. We choose LSTM as the base model due to its effectiveness in modeling the complex temporal data. We would show how to transform the user-item rating data over time into a valid input of the LSTM framework, and how to embed the dynamic social influence of users over time to help better model each user's latent vector over time. Specifically, for each user a, given her previous hidden state h^{t-1}_a and the current preference list L^t_a at time t, DARSE tries to model each user's current state h^t_a by leveraging the social network at the same time. DARSE is mainly composed of four modules: an input pooling layer that generates the input, a dynamic attention layer that models the dynamic influence strength over time, a social aware LSTM part that enriches the LSTM with dynamic social contextual information, and finally an output layer that generates the predicted dynamic preferences of users over time. Next, we introduce these parts in detail.

Input Pooling Layer At each time, an LSTM receives input of fixed size. The input pooling layer transforms each user's liked itemset L^t_a into a valid input of an LSTM. Specifically, given a's consumed item set L^t_a with varying sizes of consumed items, DRASE first adopts an average pooling operation that transforms this variable length list into a fixed-size latent representation $\mathbf{x}^t_a \in \mathbb{R}^D$ as:

$$\mathbf{x}^t_a = Pooling(Q(:, L^t_a)) \tag{6}$$

where Q is the latent matrix of items in the dynamic space. $Q(:, L^t_a)$ denotes choosing all item latent vectors that appear in L^t_a. Since L^t_a changes for each user at each time, we generate the input $\mathbf{x}^t_a$ through an pooling operation. Specifically, pooling is a basic operation that appears in many neural networks. It refers the operation that combines the outputs of neuron clusters at one layer into a neuron in the next layer [22]. By treating each row in $Q(:, L^t_a)$ as a neuron cluster, the pooling would generate a fixed size representation of $\mathbf{x}^t_a$ with size D. Commonly used pooling operations include the max pooling and the average pooling [22]. E.g., for the average pooling operation, the l-th element $\mathbf{x}^t_a(l)$ in $\mathbf{x}^t_a$ is:

$$\forall l = 1, .., D, \forall i \in L^t_a, \mathbf{x}^t_a(l) = \frac{\sum_{i \in L^t_a} Q(l, i)}{|L^t_a|}. \tag{7}$$

In practice, we find there are not significant differences in choosing different pooling operations, we select the average pooling operation in the following experiments.

Dynamic Attentive Network The goal of the dynamic attention layer is to select influential social connections for each user over time, and then summarizes these social connections' states with a social contextual vector $\widetilde{h}^t_a$. This contextual vector could be further sent to an LSTM to better model the social influence of each user's latent vectors over time. Given user a and one of her social connection b ($s_{ba}=1$), let α^t_{ab} denote the influence strength of b to a at time t, we use a two-layered subnetwork to capture the dynamic attentive score $m^t(a,b)$ as:

$$m^t(a,b) = ReLU(A_5 \times ReLU(A_1 \times h_a^{t-1} + A_2 \times h_b^{t-1} + A_3 \times e_a + A_4 \times e_b)) \quad (8)$$

where h_a^{t-1} denotes the latent vector of a at time $t-1$. We use $\Theta_A = [A_1, A_2, A_3, A_4, A_5]$ to denote the parameters in the dynamic attention network, where the first four elements are the parameters in the first layer of the attention network, and the last parameter (i.e., A_5) is the parameter in the second layer. $ReLU$ denotes the ReLU activation function. e_a and e_b are the social embeddings of a and b from the social network structure. In fact, various social embedding techniques have been proposed to extract meaningful embeddings from S. Since the focus of this paper is not to devise more sophisticated techniques for social embedding, we simply use a popular unsupervised deep learning model, i.e, denoising autoEnoder, to model each user's social embedding [35]. This simple technique is empirically proved to have good performance for capturing hidden structures in a social network [48].

The final dynamic social influence score α^t_{ab} is obtained by normalizing the above attention scores as:

$$\alpha^t_{ab} = \frac{exp(m^t(a,b))}{\sum_{c \in S_a} exp(m^t(a,c))}. \quad (9)$$

where S_a denotes all users that a follows in the social network. The learned value α^t_{ab} shows the influence of b on a at time t. Given each user a's hidden state at time t, then the social contextual information of a, denoted as $\widetilde{h}^t_a$, is represented as a weighted dynamic social influence from social neighbors as:

$$\widetilde{h}^t_a = \sum_{b \in S_a} \alpha^t_{ab} \times h^t_b. \quad (10)$$

Social LSTM Layer In a social platform, a user's current dynamic latent vector is largely influenced by her social connections' previous latent vectors. After obtaining each user's social contextual vector $\widetilde{h}^{t-1}_a$, the Social LSTM part takes each user a's input $\mathbf{x}^t_a$, her previous state h^{t-1}_a, and the enriched social contextual information $\widetilde{h}^{t-1}_a$ as input, and predicts the hidden state h^t_a as:

$$h^t_a = f_{LSTM}([\mathbf{x}^t_a, h^{t-1}_a, \widetilde{h}^{t-1}_a]), \quad (11)$$

where $f_{LSTM}(x)$ is an LSTM network as depicted in Fig.1. In the social LSMT network, each user a's input is composed of three parts: an input representation $\mathbf{x}^t_a$ that encodes user a's consumed items, a previous hidden state h^{t-1}_a that represents her hidden state at previous time $t-1$, and a dynamic social contextual input $\widetilde{h}^{t-1}_a$. Thus, different from the traditional LSTM part (Eq.(4)) that only considers each user a's previous input $\mathbf{x}^t_a$ and her previous hidden state h^{t-1}_a, it also encompasses each user's dynamic social contextual representation to infer a user's dynamic future latent vector. Specifically, at each time t, the social contextual information $\widetilde{h}^{t-1}_a$ captures the dynamic social influence of a's social neighbors and varies over time. In such a way, the social neighbors' dynamic influences on each user at previous time t are naturally fused to infer user a's dynamic latent vector h^t_a.

Dynamic Output Layer After getting each user's hidden representation h^t_a at time t, the output of the DARSE is defined as:

$$\hat{r}^t_{D,ai} = q'_i \times h^t_a. \quad (12)$$

4.3 Static Attentive Social Recurrent Recommendation

Besides capturing each user's time-evolving preferences with DARSE, we also argue that each user remains a static interest that does not evolve over time. E.g., the latent factors that are correlated to a user's profile of gender and birthplace, and a user's long term interest. In this part, we introduce the SARSE part that depicts users' stationary interests by leveraging the social network. Similar as DARSE, we also use an attention network to select social neighbors that have large influence on each user for the recommendation. Different from DARSE, as we focus on the static user preference over time, this part models the static social influence among users that do not evolve over time.

Input With the static assumption, let P and W denote the corresponding user and item base latent matrix. Then, given the input of user a's latent vector p_a and item i's latent vector w_i, the predicted rating $\hat{r}^t_{ai}$ of user a to item i at time t is usually defined as:

$$\hat{r}^t_{S,ai} = w'_i \times p_a. \quad (13)$$

In fact, the above simple prediction function is the foundation of many classical latent factor based models [25, 27]. However, as the user-item rating matrix is very sparse, the prediction performance may be limited by the sparse rating data [24]. Next, we introduce the static social attention part that summarizes the social influences in a social network.

Static Social Attention The goal of the static attention layer is to select the stationary influential social connections for each user, and then summarizes these social connections' stats with a social contextual vector $\widetilde{p}_a$. This contextual social vector could enhance the prediction results. Specifically, given user a and one of her social connection b ($s_{ba}=1$), let β_{ab} denotes the static influence strength of b to a that does not evolve over time, we use a two-layered subnetwork to capture the static attentive score $n(a,b)$ as:

$$n(a,b) = ReLU(B_5 \times ReLU(B_1 \times p_a + B_2 \times p_b + B_3 \times e_a + B_4 \times e_b)). \quad (14)$$

where p_a denotes the static latent vector of a. e_a and e_b are the social embeddings of a and b from the social network structure. We use $\Theta_B = [B_1, B_2, B_3, B_4, B_5]$ to denote the parameters in the static attention network, where the first four elements are the parameters in the first layer of the attention network, and the last parameter (i.e., B_5) is the parameter in the second layer. Then, the final static social influence score β_{ab} is obtained by normalizing the above attention scores as:

$$\beta_{ab} = \frac{exp(n(a,b))}{\sum_{c \in S_a} exp(n(a,c))}. \quad (15)$$

After that, we could get the enriched static social latent vector $\widetilde{p}_a$ as:

$$\widetilde{p}_a = \sum_{b \in S_a} \beta_{ab} \times p_b + p_a. \quad (16)$$

Static Output After getting each user's enriched representation $\widetilde{p}_a$, the output of the SARSE part is defined as:

$$\hat{r}^t_{S,ai} = w'_i \times \widetilde{p}_a. \quad (17)$$

Compared to the prediction function in Eq.(13), the above static output function summarizes the static social influence from neighbors for the recommendation. Thus, it can partially solve the data sparsity issue in recommendation.

4.4 Model Learning

In practice, we jointly train the two parts of ARSE, i.e., DARSE and SARSE, in a unified loss function. Specifically, given the user preference sequence from time 1 to time T, our goal is to learn the model parameter set $\Theta = [\Theta_A, \Theta_B, \Theta_{LSTM}, P, Q, W]$ with an objective loss function. Specifically, Θ_A and Θ_B are the parameters in the dynamic and static attention networks, and Θ_{LSTM} is the parameters in the LSTM module. Since we focus on the implicit feedback of users, we adopt the widely used log loss function, which is defined as:

$$L_\Theta(\mathbf{R}, \hat{\mathbf{R}}) = -\sum_{t=1}^{T}\sum_{a=1}^{M}\sum_{i=1}^{N}[r_{ai}^t log(\hat{r}_{ai}^t) + (1 - r_{ai}^t)log(1 - \hat{r}_{ai}^t)]. \quad (18)$$

In fact, the above log loss function is equal to the negative log-likelihood of the binary outputs with a Bernoulli distribution.

As all the modules in ARSE with the above loss function are analytically differentiable, ARSE could be trained in an end-to-end manner with gradient descent based methods. Specifically, we learn all the parameters in the DARSE part and the SARSE part simultaneously, which enable the two parts in ARSE to reinforce each other. In practice, we use TensorFlow to implement our proposed model and use Adam to adaptively update the learning rate, which has been proven especially effective for training neural networks. Since the rating data is very sparse, if we consider all the missing values in the optimization function as 0, then this problem turns to an especially unbalance prediction problem with much more 0s than 1s. We borrow a widely used under sampling technique for the implicit feedback. Specifically, in each iteration in the training process, for each observed consumption at each time (i.e., $r_{ai}^t = 1$), we sample m unobserved ratings as the pseudo negative consumption with a weight of $\frac{1}{m}$. Since the sampling process is random, each pseudo negative sample gives very weak signals in the learning process. Besides, since our model heavily relies on the classical LSTMs, to prevent overfitting, we use dropout [32] technique to randomly drop hidden units of LSTM in each iteration during the training process.

After obtaining the model parameter set, the predicted rating of $\hat{r}_{ai}^{T+1}$ could be approximated as:

$$\hat{r}_{ai}^{T+1} = Sig(\hat{r}_{S,ai}^{T+1} + \hat{r}_{D,ai}^{T+1}) \approx Sig(q_i' \times h_a^{T+1} + w_i' \times \widetilde{p}_a), \quad (19)$$

where $Sig(x)$ is a sigmoid function that constraints the results within the range of $[0, 1]$.

Table 2: The statistics of the two datasets.

Dataset	Epinions	Gowalla
Users	4,630	21,755
Items	26,991	71,139
Time Windows	12	4
Total Links	78,356	257,550
Training Ratings	62,872	278,154
Test Ratings	2,811	52,448
Link Density	0.35%	0.053%
Rating Density	0.050%	0.018%

5 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of ARSE on two datasets. We aim to answer the following research questions:

RQ1: Does our proposed model outperforms the state-of-the-art baselines for the recommendation task?

RQ2: Does the combination of the dynamic and static user preferences and two social attention networks make sense in recommendation applications?

5.1 Experimental Settings

Datasets. We briefly introduce the two datasets that we use.

Epinions: Epinions is a who-trust-whom directed online social network that provides product rating and review service. Users can rate and review products in this website with rating values from 1 to 5. Also, users link to others that they trust. We used the public available Epinions dataset provided by Richardson et al. [29] and treated each month as a time window. Since we focus on the implicit feedback of users over time, in Epinions dataset, we transform the detailed ratings into a value of 0 or 1 indicating whether the user has rated the item.

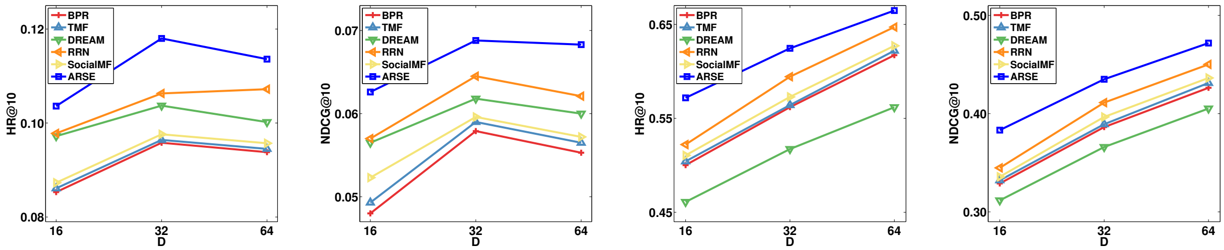
Gowalla: People make friends and share locations on this location-based social network. In this paper, we adopted the dataset provided by Scellato [31]. Specifically, it contains 4 snapshots, with each month as a time window. Since we focus on the implicit feedback of users over time, in Gowalla dataset, if a user checked in a location at that time, the rating value is marked as 1, otherwise it equals 0.

In both datasets, we filtered out users that have less than 2 rating records and 2 social links. We also removed those items that have been rated less than 2 times. Table 2 shows the statistics of the two datasets after pruning. In data splitting process, we use the data till time T for model training, i.e., $T = 11$ (T=3) in Epinions (Gowalla). To tune the parameters, we randomly select 10% from the training data as validation data, which are used for parameter tuning.

Evaluation Metrics. To evaluate the performance of the models for item recommendation, we adopt two widely used evaluation metrics for top-K ranking performance, i.e., Hit Ratio(HR) and Normalized Discounted Cumulative Gain(NDCG) [8]. The HR measures the percentage of the liked items that are presented in the ranking list. And NDCG considers the ranking positions of the hit items in the ranking list. For both metrics, the larger the value, the better the performance. Since there are much more unrated items than the rated items in the test data, similar as other works, for each test record, we randomly select 1000 items that each user did not rate before as the negative samples [40]. The positive sample and the 1000 negative samples are mixed together for ranking. Then, the final model performance is measured by averaging the ranking scores of all the test records in the test data.

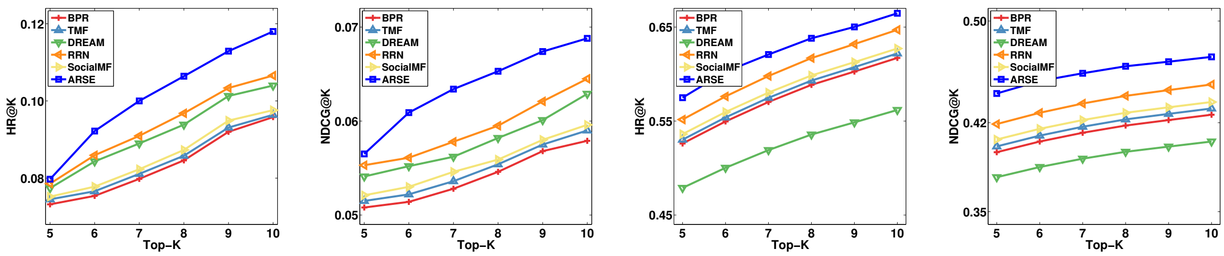
Baselines. We compare our proposed model with the following baselines:

- **BPR**:It is a competing ranking based static latent factor model for the implicit feedback. Specifically, it assumed the predicted rating is an inner product of the corresponding user and item latent vector, and a pair-wise loss function is adopted for model learning [27].



(a) Epinions HR@10 (b) Epinions NDCG@10 (c) Gowalla HR@10 (d) Gowalla NDCG@10

Figure 3: Overall performance under different latent dimension size D.



(a) Epinions HR@K (b) Epinions NDCG@K (c) Gowalla HR@K (d) Gowalla NDCG@K

Figure 4: Overall performance under different values of top-K size K.

- **TMF**:This model extended classical latent factor models by considering the dynamics of users' preferences over time. Specifically, it introduced a temporal latent vector and modeled users' preferences over time as a tensor factorization task [44].
- **SocialMF**: This is a classical model for social recommendation. Specifically, this model incorporated the social influence among users into classical latent factor models, where the influence strength is simply set equally for all social connections [16].
- **DREAM**: Dynamic REcurrent bAsket Model (DREAM) is a recurrent network based model for temporal recommendation. Specifically, DREAM could be seen as a special case of our dynamic social recommendation part DARSE in ARSE without any dynamic social influence modeling[47].
- **RRN**: It is a state-of-the-art model that adopts recurrent neural network for temporal recommendation. RRN endowed both users and items with an LSTM autoregressive model that captured dynamics over time [38].

Parameter Settings. For all models that are based on latent factor models, we randomly initialize the latent factors with a Gaussian distribution (with a mean of 0 and standard deviation of 0.01). As to our proposed model that has an LSTM structure, the initial LSTM cell is set as a zero matrix, and the rest parameters of our model are initialized with the same Gaussian distribution as mentioned before. Since all models relied on the gradient descent based methods, we use Adam as the optimizing method for all models, with a batch size of 1024 at each iteration. To avoid overfitting, we also adopt the dropout in training process. We test different dropout ratios and find that when the dropout ratio is set as 0.2, the performance is the best. Thus, we set dropout ratio as 0.2. Please note that, there are several other parameters in the baselines, we tune all these parameters to ensure the best performance of the baselines for fair comparison.

5.2 Overall Comparison(RQ1)

In this section, we compare the overall performance of various models under different parameters. Specifically, Fig. 3 shows the HR@10 and NDCG@10 for both datasets with varying latent dimension size D. As can be seen from this figure, on both datasets, our model outperforms all the other models with different values of D for the two ranking metrics. E.g, when $D = 32$, the improvement of HR over the best baseline is 11.01% on Epinions. With regard to the baselines, the performance of SocialMF and TMF improves over BPR, since it extended BPR by considering the social influence and temporal effect. RRN always performs the best among all the baselines by modeling the users' dynamic latent interests with the LSTM structure. This also validates the superiority of modeling users' complex temporal interests with deep neural models compared to the classical linear models. However, DREAM performs quite well on Epinions while it does not perform well on Gowalla. We guess a possible reason is that, the Epinions dataset exhibits more dynamics while the Gowalla data shows more static properties (e.g., most users can only checkin at locations nearby). As DREAM does not considered the static user interest, it fails on the Gowalla data. Last but not least, as the latent dimension size increases from 16 to 64, the performances of all models on Gowalla increase. This is because the larger dimensions could capture more hidden factors of users and items, thus achieves better performance. On Epinions, as the latent dimension size increases from 32 to 64, the performances begin to decrease. Besides, Fig. 4 shows the HR@K and NDCG@K on both datasets with varying top-K recommendation size K. We find the performance trends are similar as the trends in Fig. 3, with our proposed model ARSE always shows the best performance. Based on the overall experimental results, we could empirically conclude that our proposed ARSE model outperforms all the baselines under different ranking metrics and different parameters. In the following experiments, without loss of generality, we set $D = 32$ and $K = 10$ for all models on Epinions dataset, and $D = 64$ and $K = 10$ for all models on Gowalla.

Table 3: The improvement of the temporal and static attention subnetworks in ARSE, where AVG denotes the average social influence strength and ATT represents our proposed attention network.

Model	Sub model		Epinions		Gowalla	
	DARSE	SARSE	HR	NDCG	HR	NDCG
ARSE	AVG	-	-	-	-	-
	-	AVG	-3.30%	-2.60%	5.97%	4.55%
	AVG	AVG	2.23%	8.31%	6.86%	7.24%
	ATT	AVG	3.75%	11.19%	7.78%	8.87%
	AVG	ATT	2.50%	8.80%	8.90%	11.87%
	ATT	ATT	**5.36%**	**12.78%**	**9.78%**	**12.76%**

5.3 Attention Analysis(RQ2)

A key characteristic in our proposed model ARSE is the two designed attention networks for social influence modeling: a dynamic attention part of DARSE that captures the dynamics of social influence over time, and a static attention part of SARSE that models the static social influence among users. In this part, we show the effectiveness of the two attention subnetworks. We present the different attention network strategies in Table 3. In this table, AVG means adopting an average social influence weight among users, which is not learned from the attention network. This average social influence resembles an average pooling operation in neural networks. And ATT means modeling the social influence with the designed attention networks. E.g., in this table, (DARSE=AVG, SARSE=-) means we use the average social influence strength in DARSE, and the SARSE part without attention modeling. With this setting(DARSE=AVG, SARSE=-) as a baseline, we show the improvement of each attention strategy compared to it. Correspondingly, we do not show the results that are correlated with the setting (DARSE=AVG, SARSE=-) in the first row (marked as - in the first row).

From this table, we first compare the static part and the dynamic part alone (the first two rows). On Epinions, the performance of (DARSE=-, SARSE=AVG) is worse than the baseline. In contrast, the performance of (DARSE=-, SARSE=AVG) shows better results than the baseline on Gowalla. We guess a possible reason is that, users exhibit more static preferences on Gowalla, while users show more dynamic preferences on Epinions. By observing the results from the third to the sixth rows, it is obvious that combining dynamic part and static part together could enhance either part alone. Furthermore, each attention network improves the results of the corresponding model that uses the average social influence strength. For example, on Epinions, the dynamic attention network in ARSE (i.e., DARSE=ATT, SARSE=AVG) improves the average (DARSE=AVG, SARSE=AVG) social influence strength with 2.66% on NDCG metric. Combining both attention networks, our model further improves 7.67% compared to the results of average social influence strength modeling. Thus, the best performance achieves when both the static and dynamic user interests are modeled with the attention networks.

Table 4: The contribution of different parts in ARSE. Please refer to the explainations in Section 5.4 for the detailed meanings of these submodules.

Model	Sub modules	Epinions		Gowalla	
		HR	NDCG	HR	NDCG
ARSE	SRSE	0.0961	0.0537	0.6167	0.4276
	SARSE	0.0994	0.0575	0.6201	0.4302
	DRSE	0.1067	0.0626	0.5618	0.4003
	DARSE	0.1139	0.0666	0.5702	0.4052
	DRSE+SRSE	0.1102	0.0663	0.6391	0.4452
	DARSE+SRSE	0.1156	0.0683	0.6411	0.4532
	DRSE+SARSE	0.1112	0.0673	0.6501	0.4658
	ARSE	**0.1180**	**0.0688**	**0.6647**	**0.4718**

5.4 Contribution Analysis(RQ2)

In our proposed ARSE, we adopt a combination of the DARSE module and SARSE module to capture both users' dynamic interests and their static preferences. In both the DARSE part and SARSE part, we propose an attention network to model the corresponding social influence. In this subsection, we would like to show the contribution of each part for the overall performance. Specifically, we use DARSE and SARSE to denote the two parts in ARSE. If DARSE disregards any dynamic attentive influence modeling (i.e, $\widetilde{h}_a^t = 0$), it degenerates to a Dynamic Recurrent Social rEcommendation (DRSE). Similar, if SARSE neglects any static attentive influence modeling (i.e., $\widetilde{p}_a = p_a$), it degenerates to SRSE.

Table 4 shows the contribution analysis of each part in ARSE. As can be seen from the first four rows of this table, combing the social attention in either part would enhance the corresponding performance. The fifth row shows that it is necessary to consider both the dynamic and static user interests for better modeling users' preferences. Finally, if we model the dynamic interest and static interest with two attention networks, the expressiveness of our proposed ARSE model reaches the best performance. These results clearly show the effectiveness of each module in ARSE, and modeling them together could largely improve the recommendation performance.

6 CONCLUSION

In this paper, we proposed an ARSE model for temporal social recommendation under the recurrent neural network structure. We argued that users' preferences over time are driven by their temporal complex dynamic interests and the static interests, and modeled these two kinds of interests by leveraging social influence among users with two attention networks. Specifically, in the dynamic preference modeling process, we designed a temporal attention network to model the temporal social influence over time, and proposed a dynamic social aware recurrent neural network to capture users' complex latent interests over time. In the general static preference modeling process, we augmented each user's static interest part by introducing a static social attention module to model the stationary social influence among users. Extensive experimental results on two real-world datasets clearly showed the improvement of our proposed model, e.g., the improvement of our proposed ARSE model over the best baseline is more than 11% on Epinions dataset with the HR metric.

ACKNOWLEDGEMENTS

This research was partially supported by grants from the National Key Research and Development Program (Grant No. 2017YFB0803301), the Natural Science Foundation of China(Grant No. 61432019, 61732008, 61725203,61602147,61632007), the Anhui Provincial Natural Science Foundation(Grant No. 1708085QF155), and the Fundamental Research Funds for the Central Universities(Grant No. JZ2016HGBZ0749).

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE* 17, 6 (2005), 734–749.
[2] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. 2008. Influence and correlation in social networks. In *SIGKDD*. ACM, 7–15.
[3] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The role of social networks in information diffusion. In *WWW*. WWW, 519–528.
[4] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. 2012. A 61-million-person experiment in social influence and political mobilization. *Nature* 489, 7415 (2012), 295–298.
[5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. ACM, 335–344.
[6] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *SIGKDD*. ACM, 1029–1038.
[7] Sean R Eddy. 1996. Hidden markov models. *Current opinion in structural biology* 6, 3 (1996), 361–365.
[8] Aleksandr Farseev, Ivan Samborskii, Andrey Filchenkov, and Tat-Seng Chua. 2017. Cross-domain recommendation via clustering on multi-layer graphs. In *SIGIR*. ACM, 195–204.
[9] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*. ACM, 93–100.
[10] Eric Gilbert and Karrie Karahalios. 2009. Predicting tie strength with social media. In *SIGCHI*. ACM, 211–220.
[11] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *WSDM*. ACM, 241–250.
[12] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. ACM, 355–364.
[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. WWW, 173–182.
[14] Herminia Ibarra and Steven B Andrews. 1993. Power, social influence, and sense making: Effects of network centrality and proximity on employee perceptions. *Administrative science quarterly* (1993), 277–303.
[15] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *PAMI* 20, 11 (1998), 1254–1259.
[16] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*. ACM, 135–142.
[17] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2012. Social contextual recommendation. In *CIKM*. ACM, 45–54.
[18] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*. ACM, 79–86.
[19] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*. IEEE, 3128–3137.
[20] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Communications* 53, 4 (2010), 89–97.
[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. NIPS, 1097–1105.
[23] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*. IMLS, 282–289.
[24] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. ACM, 287–296.
[25] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NIPS*. NIPS, 1257–1264.
[26] Wenjie Pei, Jie Yang, Zhu Sun, Jie Zhang, Alessandro Bozzon, and David MJ Tax. 2017. Interacting Attention-gated Recurrent Networks for Recommendation. In *CIKM*. ACM, 1459–1468.
[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI, 452–461.
[28] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. WWW, 811–820.
[29] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. 2003. Trust management for the semantic web. *Web-ISWC*, 351–368.
[30] Haşim Sak, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*. 338–342.
[31] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. 2011. Exploiting place features in link prediction on location-based social networks. In *SIGKDD*. ACM, 1046–1054.
[32] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.
[33] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. NIPS, 3104–3112.
[34] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. *SNAM* 3, 4 (2013), 1113–1133.
[35] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR* 11, Dec (2010), 3371–3408.
[36] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *SIGIR*. ACM, 185–194.
[37] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*. ACM, 261–270.
[38] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. ACM, 495–503.
[39] Le Wu, Enhong Chen, Qi Liu, Linli Xu, Tengfei Bao, and Lei Zhang. 2012. Leveraging tagging for neighborhood-aware probabilistic matrix factorization. In *CIKM*. ACM, 1854–1858.
[40] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. 2017. Modeling the Evolution of Users' Preferences and Social Links in Social Networking Services. *TKDE* 29, 6 (2017), 1240–1253.
[41] Le Wu, Qi Liu, Enhong Chen, Nicholas Jing Yuan, Guangming Guo, and Xing Xie. 2016. Relevance meets coverage: A unified framework to generate diversified recommendations. *TIST* 7, 3 (2016), 39.
[42] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. 2010. Temporal recommendation on graphs via long-and short-term preference fusion. In *SIGKDD*. ACM, 723–732.
[43] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *IJCAI*. 3119–3125.
[44] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*. SIAM, 211–222.
[45] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. IMLS, 2048–2057.
[46] Xiwang Yang, Yang Guo, Yong Liu, and Harald Steck. 2014. A survey of collaborative filtering based social recommender systems. *Computer Communications* 41 (2014), 1–10.
[47] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *SIGIR*. ACM, 729–732.
[48] Shuangfei Zhai and Zhongfei Zhang. 2015. Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs. In *SDM*. SIAM, 451–459.