ISSN 1000-9000(Print)
/1860-4749(Online)
CODEN JCTEEM

Vol.35 No.2 Mar. 2020

Journal of Computer Science & Technology

SPONSORED BY INSTITUTE OF COMPUTING TECHNOLOGY
THE CHINESE ACADEMY OF SCIENCES &
CHINA COMPUTER FEDERATION
SUPPORTED BY NSFC
CO-PUBLISHED BY SCIENCE PRESS &
SPRINGER

COMPUTER

Liu Y, Li Z, Huang W *et al.* Exploiting structural and temporal influence for dynamic social-aware recommendation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(2): 281–294 Mar. 2020. DOI 10.1007/s11390-020-9956-9

Exploiting Structural and Temporal Influence for Dynamic Social-Aware Recommendation

Yang Liu, Zhi Li, Wei Huang, Tong Xu*, and En-Hong Chen, *Fellow*, *CCF*, *Senior Member*, *IEEE*

Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China Hefei 230027, China

E-mail: {ly0330, zhili03, ustc0411}@mail.ustc.edu.cn; {tongxu, cheneh}@ustc.edu.cn

Received August 20, 2019; revised December 22, 2019.

Abstract Recent years have witnessed the rapid development of online social platforms, which effectively support the business intelligence and provide services for massive users. Along this line, large efforts have been made on the social-aware recommendation task, i.e., leveraging social contextual information to improve recommendation performance. Most existing methods have treated social relations in a static way, but the dynamic influence of social contextual information on users' consumption choices has been largely unexploited. To that end, in this paper, we conduct a comprehensive study to reveal the dynamic social influence on users' preferences, and then we propose a deep model called Dynamic Social-Aware Recommender System (DSRS) to integrate the users' structural and temporal social contexts to address the dynamic social-aware recommendation task. DSRS consists of two main components, i.e., the social influence learning (SIL) and dynamic preference learning (DPL). Specifically, in the SIL module, we arrange social graphs in a sequential order and borrow the power of graph convolution networks (GCNs) to learn social context. Moreover, we design a structural-temporal attention mechanism to discriminatively model the structural social influence and the temporal social influence. Then, in the DPL part, users' individual preferences are learned dynamically by recurrent neural networks (RNNs). Finally, with a prediction layer, we combine the users' social context and dynamic preferences to generate recommendations. We conduct extensive experiments on two real-world datasets, and the experimental results demonstrate the superiority and effectiveness of our proposed model compared with the state-of-the-art methods.

Keywords recommender system, social influence, sequential recommendation

1 Introduction

In the digital and informational era, people are overwhelmed by the glut of information. To deal with this problem, the recommender system has become an important application. Usually, recommender system techniques focus on modeling user behaviors, item attributes, and contexts to capture user preference, thus accomplishing personalized recommendation services. Along this line, large efforts have been made in both industry and academia to enhance the performance of recommender systems.

For a long time, collaborative filtering (CF) is one of the most popular techniques for building recommender systems. Among the CF-based algorithms, latent factor models have achieved significant success in many recommendation tasks [1–3]. In recent years, with the rapid development of online social platforms, many researchers have exploited the use of social contextual information to alleviate the sparsity issue and improve recommendation performance. Initially, social information is integrated into latent factor models for social-aware recommendation [4–9]. However, interactions between users and items are complex, which could be hardly captured by the linear operation in most latent factor methods. Fortunately, the deep learning technique has shown its capability in solving these problems and enhanc-

Regular Paper

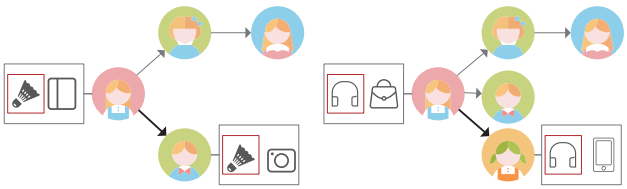
Special Section on Learning and Mining in Dynamic Environments

This work was partially supported by the National Key Research and Development Program of China under Grant No. 2018YFB1402600, the National Natural Science Foundation of China under Grant Nos. 61703386 and U1605251, and the MSRA (Microsoft Research Asia) Collaborative Research Project.

*Corresponding Author

ing the performance of recommendation task, such as CTR prediction [10, 11], top-k recommendation [12, 13] and session-based recommendation [14–16]. Existing studies have leveraged rich contextual information in deep learning based models, such as visual [17, 18] and textual [19, 20] contents of items. But the dynamic social influence, as an important information that reflects the evolving of the user profiles, has been largely unexploited.

Actually, there are still some unique challenges inherent in introducing dynamic social influence into the process of user preference modeling. Firstly, different from common auxiliary information such as pictures and texts, which are associated with users or items, social context refers to the correlation among users. Thereby most existing context-aware recommendation methods cannot be directly applied to model social contextual information. Secondly, interactions between users and items change over time, so do the social relations. It could be a difficult task to bridge dynamic information from these two domains. Thirdly, the social influence is complex and evolving over time. For example, Fig.1 shows a case that a user's preference is affected by his/her social relations. As illustrated in Fig.1, among all current social relations, different friends have impacts on the center user in different levels. Meanwhile, when the user builds a new social relation, he/she may show more interest in the items that his/her new friend likes. Therefore, the dynamic social context could help us to better understand the evolving user preference. However, how to capture dynamic influence from evolving social relations effectively is still an under-explored question.



(a) (b)

Fig.1. Illustration of a case that users' preferences are dynamically affected by their social relations.

Based on these intuitions, in this paper, we conduct a comprehensive study to exploit the dynamic social influence to enhance the performance of the sequential recommendations. We propose a hybrid model called Dynamic Social-Aware Recommender System (DSRS) to deeply explore the dynamic social influence and integrate social context to address the sequential recommendation task. DSRS is composed of two components, i.e., the social influence learning (SIL) and dynamic preference learning (DPL). Specifically, in the SIL module, we arrange social graphs in a sequential order and develop graph convolution networks (GCNs) to learn social context. Moreover, we design a structural-temporal attention mechanism to discriminatively model the social influence on structural and temporal aspects. Then, in the DPL part, we use recurrent neural networks (RNNs) to capture users' dynamic preferences. Finally, with a prediction layer, we integrate users' social context and dynamic preferences to generate recommendations. We conduct experiments on two real-world datasets, and the experimental results have clearly demonstrated the superiority of our proposed model. The contributions of this paper could be summarized as follows.

• We introduce the problem of dynamic social-aware recommendation, which focuses on revealing the impacts of dynamic social influence on user profiles.

• We propose a novel model DSRS to jointly capture dynamic social context and user preference. Moreover, we design a structural-temporal attention mechanism to discriminatively model social effects for the target users on structural and temporal aspects.

• We conduct comprehensive experiments on two real-world datasets, and the results demonstrate the effectiveness and superiority of our proposed model compared with several state-of-the-art methods.

2 Related Work

In this section, we will briefly review the studies that are related to our work. As we focus on exploring dynamic property and leveraging social contextual information in the recommendation task, the related studies mainly fall into two parts, i.e., social recommendation and sequential recommendation.

2.1 Social Recommendation

In social media platforms, users' behaviors can be divided into two types, i.e., consumption behaviors and social behaviors. With the knowledge of social influence theory that users' consumption behaviors are affected by their social relations [21], a large amount of efforts have been made to exploit social information for recommendation tasks. Traditionally, latent factor models have been widely used to model social contextual information [4–8]. For example, Jamali and Es-

ter proposed SocialMF, a social influence propagation mechanism that enables user latent representation to depend on possibly the entire social network [6]. Ma *et al.* designed SocialReg which adopts regularization method to force similar users to have similar latent representations [5]. Guo *et al.* proposed a trust-based matrix factorization model, which incorporates both rating and trust information [7]. With the advance of online social networks, latent factor based methods show its insufficiency in handling large-scale complex data. Lately, researchers have adopted neural networks, which have particular advantage in processing complex data with their deep non-linear operation, to implement social-aware recommendation [22–28]. Among those methods, Sun *et al.* attempted to incorporate social information into RNNs architecture [22], and Wu *et al.* devised an autoencoder-based approach for social embedding learning [23]. Considering that social relations are more suitable to be expressed as a graph, some studies take advantage of GCNs to extract social contextual information [24–26]. For example, Wu *et al.* proposed SocialGCN to model the diffusion process of social influence [24]. Qiu *et al.* proposed DeepInf [26], a graph-based learning framework for predicting social influence. Most previous studies process social relations with a static treatment, while we differ from these studies by considering sequential social behaviors and dynamic social influence.

2.2 Sequential Recommendation

In realistic scenarios, various user behaviors are recorded with timestamps, which results in the sequence form that may reflect dynamic user preference. The importance of sequential information has gradually attracted researchers' attention. Initially, Markov-based methods are widely used in modeling temporal data, such as FPMC [29] and HRM [30]. These methods expose their deficiency in capturing complex long-term sequence dependency. To overcome the limitation of Markov-based methods and provide more effective solution to time series data mining, researchers have applied RNNs to sequential recommendation in various application scenarios [13, 14, 31–33]. For example, Hidasi *et al.* attempted to make gated recurrent unit (GRU) fit session-based recommendation by introducing session-parallel mini-batches [14]. Yu *et al.* proposed a recurrent neural model (DREAM) for next basket prediction [13]. Wu *et al.* provided RRN, a method based on recurrent neural networks that can model the user and the item dynamics synchronously [31]. Some scholars also attempted to alleviate the data sparsity problem and improve prediction accuracy by integrating rich contextual information, such as visual and textual content of items and external situations [34–36]. Under the background of dynamic modeling, social contextual information, as an important driven force of the evolving of user preference, is rarely used compared with other context. To fill this gap, in our work we dynamically model not only user preference but also social context.

3 Methodology

In this section, we will firstly give a problem definition of the dynamic social-aware recommendation task. Then, we will describe the technical details of the proposed Dynamic Social-Aware Recommender System (DSRS) model, which consists of two main components: an attentive GCNs part named Social Influence Learning (SIL) to acquire dynamic influence in social domain, and a Dynamic Preference Learning (DPL) module to capture dynamic preference of the target user in the consumption domain. Finally, we will analyze the model complexity of DSRS. The overview of our model architecture is presented in Fig.2, where Fig.2(a) is the SIL module and Fig.2(b) is the DPL module. The finial predictions come from the two modules.

3.1 Preliminaries

3.1.1 Problem Definition

In our recommendation scenario, there are a set of users $U = \{u_1, u_2, \ldots, u_{|U|}\}$ and a set of items $V = \{v_1, v_2, \ldots, v_{|V|}\}$. Without confusion, we use indexes u, u' to denote users and v, v' to denote items. Users can interact with items and build relationship with each other. As we focus on sequential behavior and implicit feedback, the transactions are recorded with timestamps and the rating values are transformed to binary. For user-item interactions, we use $B_t^{\boldsymbol{u}}$ to represent the set of items that u consumed at time t. For user-user relationships, let $N_t^{\boldsymbol{u}}$ be the set of neighbors that is established at time t. Our problem can be formulated as follows.

Definition 1 (Dynamic Social-Aware Recommendation). *Given a user u, the corresponding temporal consumption set $B^{\boldsymbol{u}} = \{B_1^{\boldsymbol{u}}, B_2^{\boldsymbol{u}}, \ldots, B_T^{\boldsymbol{u}}\}$, and the temporal social records $N^{\boldsymbol{u}} = \{N_1^{\boldsymbol{u}}, N_2^{\boldsymbol{u}}, \ldots, N_T^{\boldsymbol{u}}\}$, our task is to predict the user's future choices ($B_{T+1}^{\boldsymbol{u}}$) by leveraging both consumption and social information.*

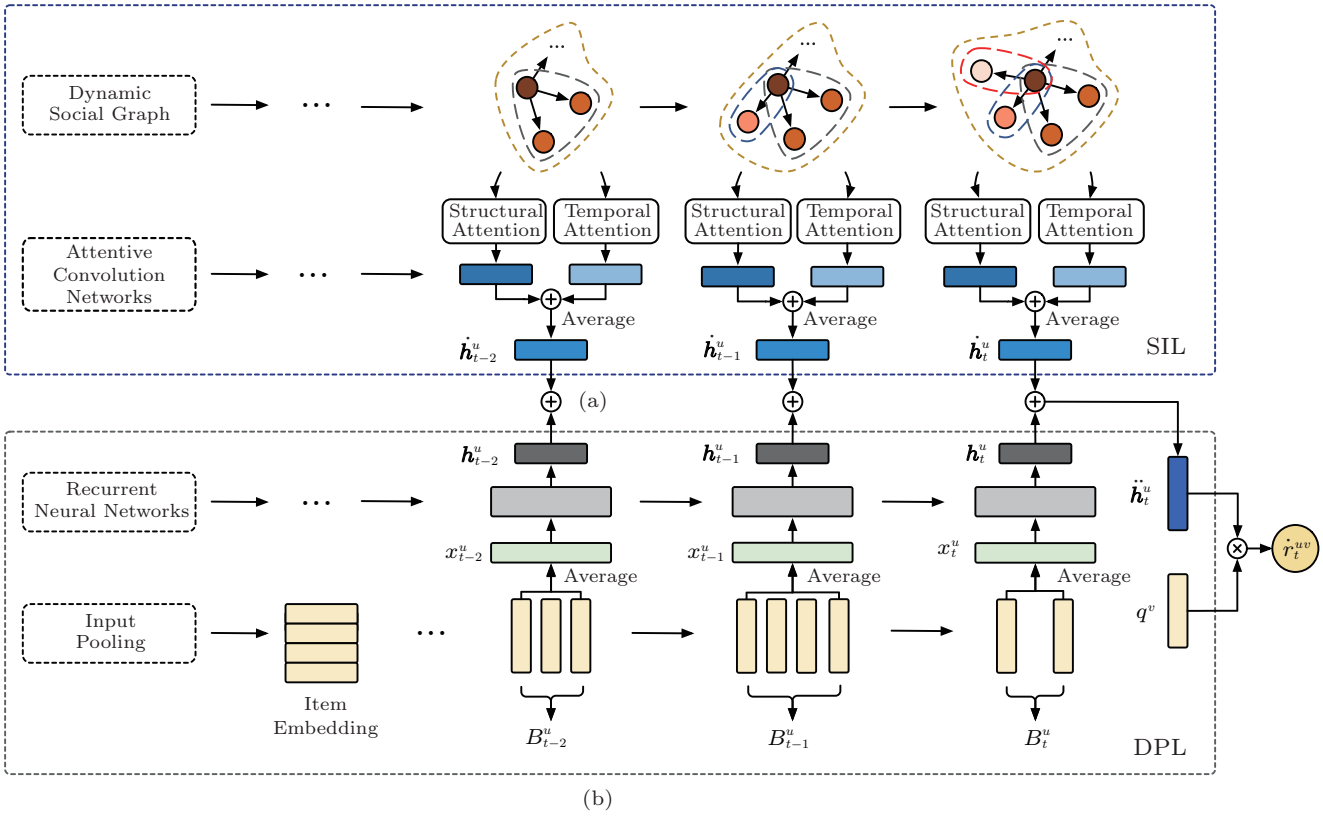


Fig.2. Overview of the proposed DSRS architecture. (a) Social Information Learning (SIL) module. (b) Dynamic Preference Learning (DPL).

3.1.2 *Dynamic Social Context*

In a social media platform, users can build relationships with each other. Social relations can be expressed as a directed graph, in which users act as nodes and social links act as edges. For a specific node, the nodes that it points to are regarded as its neighbors. As users can remove and add their social relations, the edges in the social graph are not static, but change over time. Therefore, for each specific user, the corresponding social context information is dynamic.

In our scenario, we consider the dynamic characteristics of social context information from two aspects: structural context and temporal context. As shown in Fig.3(a), links among users are changing over time, which evolves into a sequence of graphs (structural context). And Fig.3(b) illustrates the difference among all current neighbor nodes, which refers to that the durations of current social relations are different (temporal context). The evolving structure of social graph reflects that the contextual information in the social domain is dynamic. And the reason why we pay attention to temporal context is that time factor can reflect the peculiarity of social relations to a certain extent. Intuitively, newly-added social links affect users' current tastes more than the pre-existing ones. We take the two kinds of dynamic context into consideration and model them respectively in our Social Influence Learning (SIL) module. Next, we will discuss the design details of the SIL module.

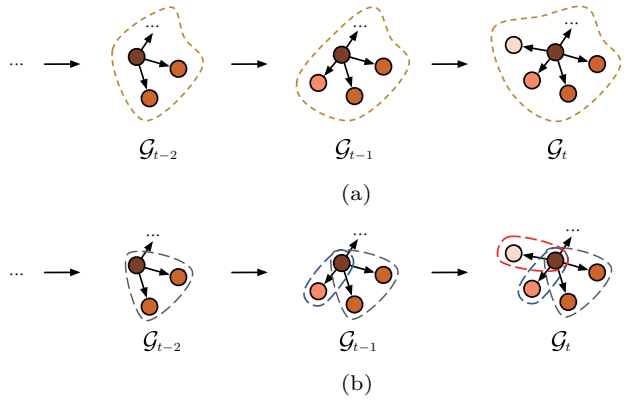


Fig. 3. Illustration of dynamic social context. (a) Structural context. Social structure changes over time, which formulates a sequence of social graphs. (b) Temporal context. Current social links come from different time steps. Links within a same small circle are established at the same time step.

3.2 Social Influence Learning

For learning social context, it is naturally to treat users' social networks as a graph. Indeed, there has been a surge of research into GCNs [37], which adapts neural networks to graph-structured data. The core idea of GCNs is to learn node embeddings by considering both node contents and the topology structure in a graph. To differentiate the importance degrees of neighbor nodes, some researchers have combined attention mechanism with GCNs [38–41]. Most graph-based methods process a large and static social graph, which leads to computational cost issues. Besides, the dynamic property of social relations is neglected among these methods. Instead of static treatment, we implement GCNs on temporal social graphs. In each step, the convolution operation is performed on a graph formed in one time interval ($\mathcal{G}_t$), rather than on an entire static graph. Moreover, we design an attention approach in our convolution step by considering both graph structure context and temporal context.

According to the social influence theory [21], a user's preference is affected by his/her social relations. Considering that the effects can be in different degrees, we adopt attention mechanism to evaluate neighbor weights in the SIL module. We design a structural-temporal attention approach, which is constituted of structural attention and temporal attention. Given a target user u and a social graph $\mathcal{G}_t$, the structural attention calculates neighbor weights based on the subgraph structure with u as the center node, and here how long each social relation exists is indiscriminate. Meanwhile, the temporal attention differentiates neighbor influence by considering when the relations are established (i.e., time factor). Social relations from different time contribute differently to the change of user preference. Therefore, the temporal attention is devised to capture time effect. The architecture of our structural-temporal attention is illustrated in Fig.4. Next, we will discuss our attention strategy in detail.

3.2.1 *Structural Attention*

In structural attention networks, the importance degrees of the neighbors to a user are evaluated based on their node features. The inputs to structural attention layer are a graph $\mathcal{G}_t$ and the node features $X_t = \{x_t^1, x_t^2, \ldots, x_t^{|U|}\}$, $x_t^i \in \mathbb{R}^F$, where $|U|$ is the number of nodes. The outputs are the attention weights. First, all feature vectors are transformed into intermediate representations by a shared weight matrix $\boldsymbol{W}_x \in \mathbb{R}^{F \times F'}$:

$$\hat{X}_t = X_t \boldsymbol{W}_x,$$

where $\hat{X}_t = \{\hat{x}_t^1, \hat{x}_t^1, \ldots, \hat{x}_t^{|U|}\}$ is the linear transformation of X_t. For a target node u and a neighbor node u', the relation between node u and u' is calculated as follows:

$$\boldsymbol{uu}'_t = \sigma(\boldsymbol{a}(\hat{\boldsymbol{x}}_t^{\boldsymbol{u}} \oplus \hat{\boldsymbol{x}}_t^{\boldsymbol{u}'})),$$

where $\boldsymbol{a}$ is a shared parameter vector and $\oplus$ is concatenation operation. The attention is masked, which means the calculation is only performed among u and its neighbors at present moment (i.e., $N_t^{\boldsymbol{u}}$). After the relations are obtained, the attention coefficients are the normalization of $\{\boldsymbol{uu}'_t, \boldsymbol{u}' \in N_t^{\boldsymbol{u}}\}$:

$$\alpha_t^{\boldsymbol{uu}'} = \mathrm{softmax}(\boldsymbol{uu}'_t) = \frac{\exp(\boldsymbol{uu}'_t)}{\Sigma_{\boldsymbol{u}' \in N_t^{\boldsymbol{u}}} \exp(\boldsymbol{uu}'_t)},$$

where $\alpha_t^{\boldsymbol{uu}'}$ indicates the importance degree of node $\boldsymbol{u}'$ to node $\boldsymbol{u}$.

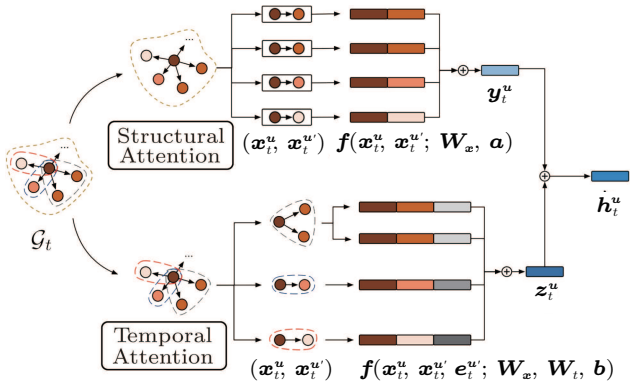


Fig.4. Illustration of structural-temporal attention architecture. Vectors in browns are node features and those in grays denote time factors.

3.2.2 *Temporal Attention*

In temporal attention networks, the calculation depends on not only node features but also associated time factors. We use $\boldsymbol{e}_t$ to denote the embedding vector of time factor. For each $\boldsymbol{u}'(\boldsymbol{u}' \in N_t^{\boldsymbol{u}})$, the corresponding time factor is obtained by embedding look-up. The calculation of attention weights here is similar to that of structural attention:

$$\hat{\boldsymbol{e}}_t = \boldsymbol{e}_t \boldsymbol{W}_t,$$
$$\hat{\boldsymbol{uu}}'_t = \sigma(\boldsymbol{b}(\hat{\boldsymbol{x}}_t^{\boldsymbol{u}} \oplus \hat{\boldsymbol{x}}_t^{\boldsymbol{u}'} \oplus \hat{\boldsymbol{e}}_t^{\boldsymbol{u}'})),$$
$$\beta_t^{\boldsymbol{uu}'} = \mathrm{softmax}(\hat{\boldsymbol{uu}}'_t) = \frac{\exp(\hat{\boldsymbol{uu}}'_t)}{\Sigma_{\boldsymbol{u}' \in N_t^{\boldsymbol{u}}} \exp(\hat{\boldsymbol{uu}}'_t)},$$

where $\boldsymbol{W}_t \in \mathbb{R}^{F \times F'}$ is a weight matrix for linear transformation of time vectors, which is used to map time vector to a proper vector space. And $\boldsymbol{b}$ is a weight vector for computing intermediate attention value. The output $\beta_t^{uu'}$ is the normalized attention coefficient, which indicates the influence of $\boldsymbol{u'}$ to $\boldsymbol{u}$ with time effect. Once obtained, the attention outputs $\alpha_t^{\boldsymbol{uu'}}$ and $\beta_t^{\boldsymbol{uu'}}$ are used to aggregate information from two aspects:

$$\begin{aligned}
\boldsymbol{y}_t^{\boldsymbol{u}} &= \Sigma_{\boldsymbol{u'} \in N_t^{\boldsymbol{u}}} \alpha_t^{\boldsymbol{uu'}} \hat{\boldsymbol{x}}_t^{\boldsymbol{u'}}, \\
\boldsymbol{z}_t^{\boldsymbol{u}} &= \Sigma_{\boldsymbol{u'} \in N_t^{\boldsymbol{u}}} \beta_t^{\boldsymbol{uu'}} \hat{\boldsymbol{x}}_t^{\boldsymbol{u'}}, \\
\dot{\boldsymbol{h}}_t^{\boldsymbol{u}} &= \sigma(\text{average}(\boldsymbol{y}_t^{\boldsymbol{u}}, \boldsymbol{z}_t^{\boldsymbol{u}})),
\end{aligned}$$

where the meaning of $\boldsymbol{y}_t^{\boldsymbol{u}}$ and $\boldsymbol{z}_t^{\boldsymbol{u}}$ is the social context vectors from two aspects (i.e., structural context and temporal context). After the aggregation operation, $\dot{\boldsymbol{h}}_t^{\boldsymbol{u}}$ serves as the output of the SIL module, which contains social contextual information with temporal influence. The above algorithm will be implemented to calculate dynamic social context vectors for all nodes in the graph.

3.3 Dynamic Preference Learning

The SIL module is applied to extract dynamic information in the social domain. To obtain users' preferences in the consumption domain, we use a sequential model to explore users' behavioral patterns. As RNNs are extensively used to process temporal sequence due to their ingeniously designed recurrent feedback mechanism, we utilize LSTMs [42], a variant of RNNs, to learn users' individual preferences. The architecture of our sequential behavior modeling part (i.e., the DPL module) is showed in Fig.2(b).

3.3.1 Input Pooling

In each time step, the LSTMs take current consumption records as inputs. The corresponding outputs serve as two purposes, i.e., users' current representations and the hidden states of the next step. Note that a user can interact with multiple items in a time window, and the LSTM takes a fixed-size vector as input at each time step. To generate valid model inputs, a pooling operation will be performed. Before input pooling, we first use an embedding layer to project the sparse representations of items to dense vectors. The item embedding set is denoted as $Q = \{q^{v_1}, q^{v_2}, \ldots, q^{v_{|V|}}\}$. For a target user u and a specific time step t, from the consumption records set $B_t^{\boldsymbol{u}}$ we can obtain the corresponding embeddings $Q_t^{\boldsymbol{u}} = \{q^{v_i} | v_i \in B_t^{\boldsymbol{u}}\}$. Then a pooling operation will be applied to aggregate input vectors. We choose the commonly used function average pooling as the aggregation function. Among all the vectors to be aggregated, average pooling takes the average value of every dimension. The calculation of input vector is formulated as follows:

$$\boldsymbol{x}_t^{\boldsymbol{u}} = average(\boldsymbol{Q}_t^{\boldsymbol{u}}) = \frac{\Sigma_{v_i \in B_t^{\boldsymbol{u}}} \boldsymbol{q}^{v_i}}{|\boldsymbol{Q}_t^{\boldsymbol{u}}|}.$$

3.3.2 Dynamic Behavior Modeling

Based on the current input $\boldsymbol{x}_t^{\boldsymbol{u}}$ and the previous hidden state $\boldsymbol{h}_{t-1}^{\boldsymbol{u}}$, the LSTM unit calculates the updated hidden state $\boldsymbol{h}_t^{\boldsymbol{u}}$ which represents the current preference of u. The compact forms of the equations for the forward pass of an LSTM unit are as follows:

$$\begin{aligned}
\boldsymbol{f}_t &= \sigma(\boldsymbol{W_f} \times [\boldsymbol{h}_{t-1}^{\boldsymbol{u}}, \boldsymbol{x}_t^{\boldsymbol{u}}]), \\
\boldsymbol{i}_t &= \sigma(\boldsymbol{W_i} \times [\boldsymbol{h}_{t-1}^{\boldsymbol{u}}, \boldsymbol{x}_t^{\boldsymbol{u}}]), \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W_o} \times [\boldsymbol{h}_{t-1}^{\boldsymbol{u}}, \boldsymbol{x}_t^{\boldsymbol{u}}]), \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \sigma(\boldsymbol{W_c} \times [\boldsymbol{h}_{t-1}^{\boldsymbol{u}}, \boldsymbol{x}_t^{\boldsymbol{u}}]), \\
\boldsymbol{h}_t^{\boldsymbol{u}} &= \boldsymbol{o}_t \circ \sigma(\boldsymbol{c}_t),
\end{aligned}$$

where $\boldsymbol{f}_t, \boldsymbol{i}_t, \boldsymbol{o}_t$ represent the forget gate, the input gate and the output gate respectively. The initial values c_0 and h_0 are generally set to zero. All W_* are learn-able parameters and σ is the active function. The output $\boldsymbol{h}_t^{\boldsymbol{u}}$ serves as the preference vector of u in consumption domain.

3.4 Prediction Layer

The hybrid model DSRS combines the two parts (i.e., SIL and DPL) to obtain final representation of user preference. As shown in Fig.2, the SIL module extracts social contextual information from a temporal social graph dynamically, and the DPL module adapts LSTMs to capture the evolving user preference in the consumption domain. In each time step, the social context vector and the user preference vector are combined as follows:

$$\ddot{\boldsymbol{h}}_t^{\boldsymbol{u}} = \dot{\boldsymbol{h}}_t^{\boldsymbol{u}} + \boldsymbol{h}_t^{\boldsymbol{u}}.$$

Note that the input features to SIL and DPL are both calculated from item embedding; thus the outputs $\dot{\boldsymbol{h}}_t^{\boldsymbol{u}}$ and $\boldsymbol{h}_t^{\boldsymbol{u}}$ can be combined by vector addition operation. When predicting the preference that a user gives to an item, the predicted rating value equals the dot product of $\ddot{\boldsymbol{h}}_t^{\boldsymbol{u}}$ and $\boldsymbol{q}^v$:

$$\dot{r}_t^{uv} = dot(\ddot{\boldsymbol{h}}_t^{\boldsymbol{u}}, \boldsymbol{q}^v).$$

3.5 Model Training

We adopt Bayesian Personalized Ranking (BPR) [3] framework for model learning. BPR is a widely used pairwise ranking framework for implicit feedback. The basic assumption of BPR is that a user prefers a positive item more than a negative one. Based on the above, the following probability needs to be maximized:

$$p(u, t, v \succ v') = \sigma(\dot{r}_t^{u,v} - \dot{r}_t^{u,v'}), \tag{1}$$

where v and v' denote a positive sample and a negative sample respectively, and $\sigma(\cdot)$ is the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$. With this functional form, a higher score is expected to be given on the positive item in comparison with the negative one. Since our goal is to predict user preference in the future, at time step t, we treat items that show in $t+1$ as the positive samples.

In the training step, for each positive sample (v), we randomly choose an item that the user has not interacted with before as the corresponding negative sample (v'). The objective function is adding up the log likelihood of (1) and the regularization term:

$$J = \Sigma \ln(1 + e^{-(\dot{r}_t^{u,v} - \dot{r}_t^{u,v'})}) + \frac{\lambda}{2}\|\Theta\|^2,$$

where λ is a parameter to control the power of regularization and Θ denotes all the parameters to be estimated. In practice, we choose Adam [43] as the optimizer, which has proved to be especially effective for training neural networks. The model parameters updating procedure is repeated iteratively until the convergence is achieved.

3.6 Model Analysis

3.6.1 Space Complexity

All the model parameters come from two parts: the parameters $\Theta_1 = [\boldsymbol{W}_f, \boldsymbol{W}_i, \boldsymbol{W}_o, \boldsymbol{W}_c]$ in LSTMs and the parameters $\Theta_2 = [\boldsymbol{W}_x, \boldsymbol{W}_t, \boldsymbol{a}, \boldsymbol{b}]$ in attentive GCNs. For Θ_1, the space complexity grows linearly with the layer of LSTMs. In most RNNs-based models the layer number is set to 1 or 2. The parameter sharing mechanism in RNNs enables its space complexity to be independent of the number of users. As for Θ_2, it is lighter than Θ_1, and parameters in Θ_2 are shared by all users. Therefore, the total space complexity of DSRS is reasonable.

3.6.2 Time Complexity

Compared with the basic RNN model, the additional time cost of DSRS mainly lies in the attentive convolution operation. If the user number is N, the average neighbor size per time is M, and for T time steps the time complexity of single layer convolution calculation is $O(2NMT)$. In the training datasets, T equals 12 in Epinions and 3 in Gowalla. The average number of neighbors per time step is 7 in Epinions and 6 in Gowalla. Hence the additional time cost is acceptable.

4 Experiments

We conduct experiments on the proposed model and other compared methods. To verify the model performance, our experiments are designed mainly to answer the following questions.

*RQ*1. Compared with the state-of-the-art methods, how does the proposed model perform?

*RQ*2. How do the two modules SIL and DPL perform when they are separately used?

*RQ*3. Is the attention mechanism helpful in our recommendation task? What roles do the two attention strategies play?

*RQ*4. How do the hyper parameters (e.g., embedding size) influence model performance?

4.1 Datasets

We conduct experiments on two real-world datasets, i.e., Epinions① and Gowalla②. Both the two datasets contain temporal consumption records and social relations records. Next, we will briefly introduce the two datasets, and then we will describe the data processing in our experimental implementation.

• *Epinions* [44]. It is a who-trust-whom online social network of a general consumer review site. Members of this platform can read new and old reviews about a variety of items to help them decide on a purchase, and they can also decide whether to "trust" each other. We use the public Epinions dataset provided by Richardson *et al.* [44] In this dataset, users' rating and social actions are timestamped.

• *Gowalla* [45]. It is a location sharing social networking website. Users on this platform are able to check in at "Spots" in their local vicinity, and they can also build social relations with each other. We use

①http://www.epinions.com/, Jan. 2020.

②https://blog.gowalla.com/, Jan. 2020.

the Gowalla dataset provided by [45]. In this dataset, checking-in actions and social behaviors are both timestamped.

In both the two datasets, we treat one month as a time window. There are total 13 time windows in Epinions and 4 in Gowalla. Since implicit feedback is the main point of focus, we transform the concrete rating values to binary. Consumption data is recorded in the form of (u, v, t) triple ID, which indicates that user u interacted with item v at time t. Similarly, the social relation is also recorded as (u, u', t) triple ID, which indicates that user u built a social relation with user v' at time t. In our sequential prediction task, the goal is to predict users' future preferences. Therefore, the records in the last time window will be used for model testing (i.e., $T = 13$ in Epinions and $T = 4$ in Gowalla) and the others for training (i.e., $T = 1$–12 in Epinions and $T = 1$–3 in Gowalla). we randomly select 10% from the training data as validation data for parameter tuning. We filter out users who have less than two time windows. After data pruning, there are 3 282 users and 26 991 items in Epinions. In the Gowalla dataset, there are 7 035 users and 71 139 items. Table 1 lists the statistics of the two datasets.

4.2 Baselines

We compare our proposed model with the following baselines.

• *BPR*[3]. It is a generic optimization criterion and learning algorithm for personalized ranking. BPR predicts ratings by calculating the inner product of the user and item latent vectors. With the assumption that users prefer positive items to the negative ones, it adopts a pair-wise loss function for model learning.

• *FPMC*[29]. It is a traditional sequential model for next basket recommendation. By combining MC and MF, FPMC can capture general interest of users and sequential effects between every two adjacent time intervals.

• *SocialMF*[6]. This is a classical model for recommendation with social influence. SocialMF incorporates social information into the basic MF model. Specifically, when updating a user's latent vector, it fuses the latent vectors of corresponding neighbors.

• *DREAM*[13]. We adapt DREAM, an RNNs-based method, as one of our baselines. For more clearly comparison, both our basic sequential prediction module and DREAM use two-layer LSTMs as the building block.

• *SocialGCN*[24]. It is a GCNs-based recommendation algorithm, which captures social influence by an information diffusion process. The designers of SocialGCN leverage rich user and item attributes in their experiment and propose a general version for the scenario where no attributes are available. We employ the general version SocialGCN which is applicable to our recommendation scenario.

As we aim to tackle the problem of bridging temporal social influence and sequential prediction, we choose methods by considering two aspects: sequence-aware methods and social-aware methods. Besides, we also consider three variants of the proposed method, and the variants are listed as follows.

• *DSRS-avg.* As a simplified version of DSRS, this method also leverages social information and it adopts an average operation rather than attentive aggregation.

• *DSRS-s.* This is a variant of the proposed method which uses structural attention only in the attentive convolution step.

• *DSRS-t.* This is a variant of the proposed method where only temporal attention mechanism is used in the attentive convolution step.

4.3 Metrics and Setups

4.3.1 Metrics

In order to measure the performance, we adopt two frequently used evaluation metrics hit ratio (HR) and normalized discounted cumulative gain (NDCG) for top-k recommendation task, as applied in previous literature[12, 22, 24, 34]. $HR@k$ measures the percentage of the positive samples presented in the top-k ranking list. For a single user, the calculation of $HR@k$ is as follows:

$$HR@k = \frac{Numberof Hit@k}{|GT|},$$

where $|GT|$ is the length of positive samples in test set. And $NDCG@k$ is sensitive to the positions of the hit

Table 1. Statistics of the Two Datasets

Dataset	Users	Items	Time Windows	Social Links	Train Records	Test Records	Link Density (%)	Rating Density (%)
Epinions	3 282	26 991	13	106 076	174 325	8 056	0.980	0.190
Gowalla	7 035	71 139	4	47 864	180 944	18 736	0.096	0.036

positive samples in the ranking list. The calculation of $NDCG@k$ is as follows:

$$DCG@k = \Sigma_{i=1}^{k} \frac{2^{rel_i - 1}}{\log_2(i+1)},$$
$$NDCG@k = \frac{DCG@k}{IDCG@k},$$

where i in DCG is the position index and rel_i is the relevance score, $IDCG$ is the ideal value of DCG, i.e., the DCG value where items in the predicted top-k list are all positive samples. When correctly predicted samples rank highly on the list, NDCG reaches a higher value. For both $HR@k$ and $NDCG@k$, the larger the value, the better the performance.

4.3.2 *Setups*

In the training step, all methods are optimized with mini-batch and the batch size is set to 512. The learning rate and optimizer are searched for each model according to their peculiarity. To prevent the neural networks-based models from overfitting, we employ dropout and set the ratio to 0.5. For models that are based on latent factor, the latent vectors are randomly initialized with Gaussian distribution with mean 0 and standard deviation 0.01. We tune all the parameters to ensure the best performance of the baselines for fair comparison. In the testing step, we generate top-k ranking list and evaluate all benchmarks with $HR@k$ and $NDCG@k$. For negative sampling, we randomly select 500 items and each user has not interacted with as the negative samples. The negative samples are mixed with the positive ones for ranking. We compare all methods with different latent factor dimensions $d = \{32, 64, 128, 256\}$ and set $k = 10$ for top-k. Experiments are conducted on a Linux server with four 2.0 GHz Intel® Xeon® E5-2620 CPUs and a Tesla K80 GPU.

4.4 Results and Analysis

4.4.1 *Overall Comparison (RQ1)*

We first compare the performance of the proposed DSRS with the baselines. All the methods are evaluated by same metrics (i.e., $HR@10$ and $NDCG@10$). Table 2 and Table 3 show the results on datasets Epinions and Gowalla respectively. We bold the results of DSRS for better comparing our method with other baselines.

1) From the results we can see that sequential models give more favorable performance than the static ones. Among sequence-aware methods, DSRS and DREAM achieve better performance than FPMC, as the former can capture multi-step dependency among users' preferences and the latter is in a pair-wise way. Both being neural networks based sequential models, DSRS performs better than DREAM. As a static latent factor model, BPR lags behind the sequential methods.

2) By comparing the results of SocialMF and BPR, the observation highlights the importance of social information. These two methods are both static la-

Table 2. Overall Performance Comparison on Dataset Epinions

Method	Dimension							
	$d = 32$		$d = 64$		$d = 128$		$d = 256$	
	$HR@10$	$NDCG@10$	$HR@10$	$NDCG@10$	$HR@10$	$NDCG@10$	$HR@10$	$NDCG@10$
BPR [3]	0.158	0.045 1	0.163	0.046 3	0.165	0.046 6	0.159	0.045 4
FPMC [29]	0.167	0.046 3	0.175	0.049 0	0.189	0.051 7	0.185	0.051 5
SocialMF [6]	0.166	0.045 8	0.168	0.046 4	0.167	0.045 1	0.163	0.049 3
DREAM [13]	0.182	0.044 5	0.210	0.051 5	0.221	0.057 0	0.213	0.056 5
SocialGCN [24]	0.186	0.046 0	0.184	0.048 8	0.173	0.046 2	0.169	0.046 0
Proposed DSRS	**0.201**	**0.048 3**	**0.220**	**0.053 6**	**0.233**	**0.060 8**	**0.230**	**0.061 1**

Table 3. Overall Performance Comparison on Dataset Gowalla

Method	Dimension							
	$d = 32$		$d = 64$		$d = 128$		$d = 256$	
	$HR@10$	$NDCG@10$	$HR@10$	$NDCG@10$	$HR@10$	$NDCG@10$	$HR@10$	$NDCG@10$
BPR [3]	0.480	0.154	0.495	0.160	0.498	0.161	0.488	0.157
FPMC [29]	0.511	0.151	0.527	0.167	0.532	0.172	0.536	0.173
SocialMF [6]	0.481	0.157	0.516	0.173	0.552	0.182	0.578	0.189
DREAM [13]	0.527	0.151	0.557	0.168	0.563	0.175	0.558	0.165
SocialGCN [24]	0.484	0.158	0.537	0.177	0.515	0.161	0.493	0.161
Proposed DSRS	**0.569**	**0.170**	**0.623**	**0.192**	**0.635**	**0.198**	**0.663**	**0.206**

tent factor based, and the difference is that SocialMF leverages social information for recommendation. As a social-aware deep model, SocialGCN does not give significant upgrades as expected. The ordinary performance of SocialGCN lies in two aspects. First, the temporal information is neglected in SocialGCN. Second, it does not differentiate the important degrees of all neighbors, and the diffusion process in SocialGCN brings more information meanwhile it causes more disturbance. DSRS, by contrast, processes sequential social graphs and uses attention mechanism to differentiate neighbor weights, and it outperforms SocialGCN.

3) Generally, the proposed model outperforms all other methods in most cases. Compared with SocialMF and SocialGCN, DSRS extracts social information dynamically by constructing a graph sequence. The benefit of such an approach is that we do not have to process a large social graph at one time. Moreover, the designed structural-temporal attention mechanism can eliminate redundancy and maintain quality social information effectively. Compared with all the baselines, on the Epinions dataset, DSRS achieves 8.06%–44.65% increase in HR@10 and 4.07%–34.81% in $NDCG$@10, and on the Gowalla dataset, the improvements are 7.97%–35.86% in HR@10 and 7.59%–31.21% in $NDCG$@10.

4.4.2 *Module Performance* (*RQ2*)

To test the performance of the two modules (i.e., SIL and DPL) in DSRS, we conduct comparative experiments and show the results in Table 4 and Table 5, where DSRS-SIL utilizes only social context information to make predictions and DSRS-DPL only learns user personal preference. By comparing model performance on the two datasets, we can see that social information plays different roles in different scenarios. On the Epinions dataset, DPL performs better than SIL. On the Gowalla dataset, SIL is superior to DPL. The above results indicate that in dataset Gowalla social context gives more information than consumption records, and it is the reverse in dataset Epinions. By combining SIL and DPL, DSRS reaches the best performance in all cases.

4.4.3 *Attention Effect* (*RQ3*)

To investigate what role the structural-temporal attention mechanism plays, we conduct experiments on three variants of DSRS. In this subsection, we will first compare performance of all variant methods and then we will analyze the working mechanism of the two attention strategies with examples.

1) To investigate the impact of the designed attention networks, we test the performance of three variants of DSRS on the two datasets. In this part of experiments, we fix the vector size and set it $d = 32$. Table 6 shows the results and the improvements of all the variant methods. The values in "Improv." column indicate the lifting percentages with DSRS-avg as benchmark. Among all the variants, DSRS-s and DSRS-t are both single layer attention networks and they are designed for learning structural weights and temporal weights respectively. As shown in Table 6, the attentive methods achieve better performance than DSRS-avg, which aggregates social information by a simple average sum operation. Besides, we can observe that the temporal attention (DSRS-t) attains a slight increase over the structural attention (DSRS-s) in both HR@10 and $NDCG$@10. It indicates that time factor could contribute to the extraction of dynamic social information.

Table 4. Module Performance Comparison on Dataset Epinions

Method	Dimension							
	$d = 32$		$d = 64$		$d = 128$		$d = 256$	
	HR@10	$NDCG$@10	HR@10	$NDCG$@10	HR@10	$NDCG$@10	HR@10	$NDCG$@10
DSRS-SIL	0.181	0.044 4	0.186	0.047 1	0.209	0.054 5	0.194	0.051 8
DSRS-DPL	0.189	0.044 6	0.215	0.051 0	0.225	0.059 0	0.220	0.057 3
DSRS	**0.201**	**0.048 3**	**0.220**	**0.053 6**	**0.233**	**0.060 8**	**0.230**	**0.061 1**

Table 5. Module Performance Comparison on Dataset Gowalla

Method	Dimension							
	$d = 32$		$d = 64$		$d = 128$		$d = 256$	
	HR@10	$NDCG$@10	HR@10	$NDCG$@10	HR@10	$NDCG$@10	HR@10	$NDCG$@10
DSRS-SIL	0.560	0.167	0.600	0.181	0.616	0.188	0.623	0.190
DSRS-DPL	0.553	0.161	0.583	0.174	0.603	0.182	0.625	0.195
DSRS	**0.569**	**0.170**	**0.623**	**0.192**	**0.635**	**0.198**	**0.663**	**0.206**

Table 6. Effect of Structural and Temporal Attention Mechanism

Method	Dataset							
	Epinions				Gowalla			
	HR@10	Improv. (%)	*NDCG*@10	Improv. (%)	*HR*@10	Improv. (%)	*NDCG*@10	Improv. (%)
DSRS-avg	0.188	–	0.044 7	–	0.554	–	0.163	–
DSRS-s	0.191	+1.60	0.045 4	+1.57	0.560	+1.08	0.167	2.45
DSRS-t	0.195	+3.72	0.045 7	+2.24	0.565	+1.99	0.168	+3.07
DSRS	0.201	+6.91	0.048 3	+8.05	0.569	+2.71	0.170	+4.29

As a hybrid approach, DSRS reaches the best results.

2) With the learned attention weights, we can track back what and when the social information dominates the prediction, which helps us to understand the recommendation strategy. The structural attention estimates which neighbors among current social relations are more important. The temporal attention measures the effects of time when relations are established. Final neighbor weights come from comprehensive results of the two attention strategies. For better demonstrating the working mechanism of our attention networks, we show the tendency of structural and temporal attention weights during three time steps in Fig.5. We randomly choose five center users (the real IDs of user 1–user 5 are 4 003, 8 003, 12 003, 16 003, 21 003 respectively) from dataset Gowalla, and draw the learned attention weights of their certain neighbors in three time steps. As shown in Fig.5, when getting closer to current time step ($t = 3$ in this example), the overall values of temporal attention weights are on a decreasing trend. The explanation behind this is: as time progresses, the influence of old neighbors will decrease and their position could be replaced by newly-added neighbors.

4.4.4 *Hyper Parameter Investigation (RQ4)*

The common hyper parameter of all methods is the embedding size d. As shown in Table 2 and Table 3, different methods reach their best results at different d due to their unique property. A larger vector size does not always bring a significant improvement, and the time consumption will become more expensive when the vector size increases. Hence the determination of a proper d comes from the trade off. In DSRS, there is another hyper parameter, i.e., the embedding size d_t for time factor. In the overall comparison stage, we set the default value of d_t equal to d. To explore the impact of time factor size, we test model performance by setting d to a fixed value and changing the value of d_t. Fig.6 illustrates the results on Epinions and Gowalla, where in each sub-figure, the curve in lighter

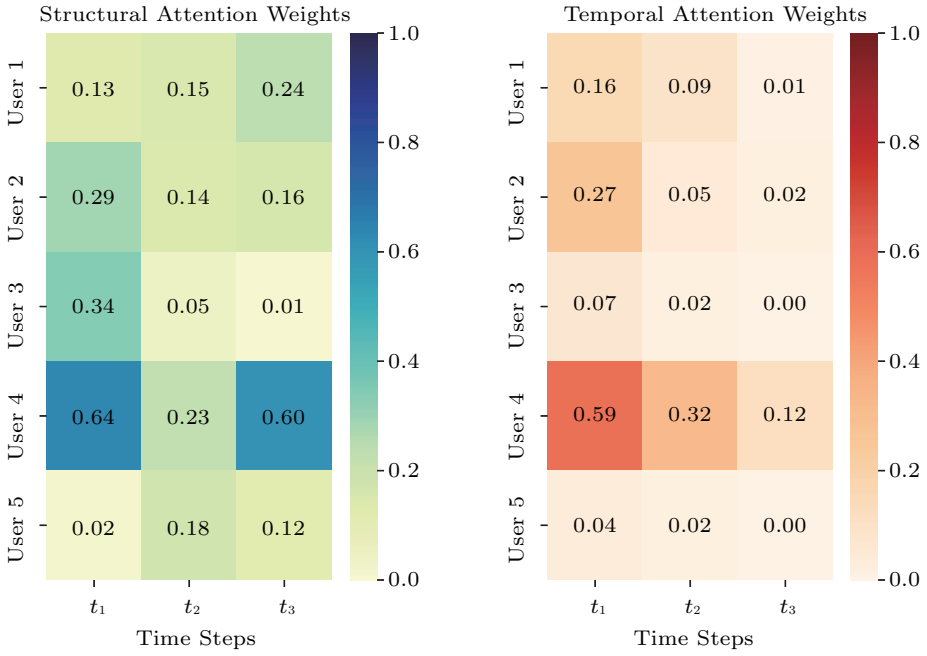


Fig.5. Visualization of the learned neighbor weights by structural and temporal attention during three time steps.

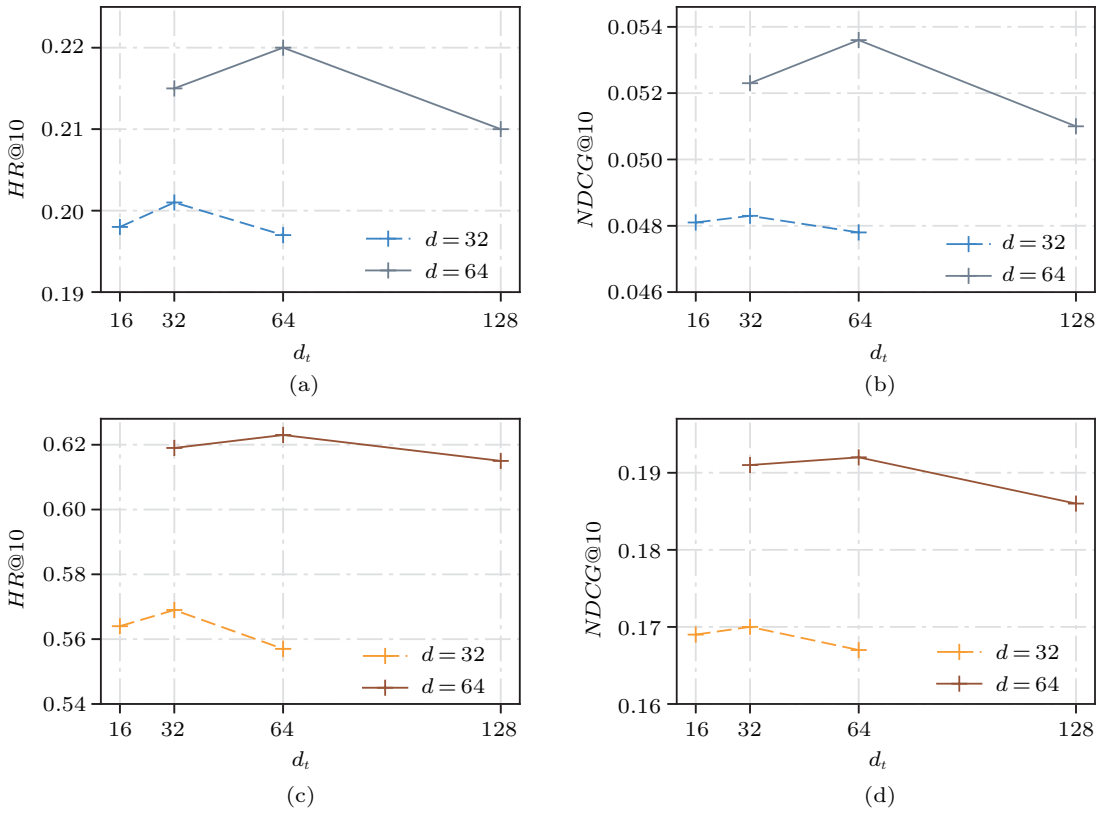


Fig.6. Changing tendency of DSRS performance when changing the dimension of time factor. (a) Epinions-HR. (b) Epinions-NDCG. (c) Gowalla-HR. (d) Gowalla-NDCG.

color indicates the results with $d = 32, d_t = \{16, 32, 64\}$ and the curve in darker color shows the results with $d = 64, d_t = \{32, 64, 128\}$. Generally, the variations on the two metrics are not obvious, but we can get the observation that keeping d_t no larger than d is a more appropriate choice. Time factor is a kind of auxiliary information, for which a larger size is superfluous.

5 Conclusions

In this paper, we aimed at exploring user social context to enhance the performance of recommender systems. To that end, we conducted a comprehensive study to reveal the dynamic social influence on users' preferences, and then we proposed a deep model called Dynamic Social-Aware Recommender System (DSRS) to address the dynamic social-ware recommendation task. DSRS contains two main components, i.e., Social Influence Learning (SIL) and Dynamic Preference Learning (DPL). Specifically, we firstly arranged user social status in a sequential order and developed graph convolution networks to learn social context of the target users in SIL. Moreover, we designed a structural-temporal attention mechanism to discriminatively model the social influence on structural and temporal aspects. Then, we modeled the users' individual dynamic preferences by DPL. Finally, with a prediction layer, we integrated the users' social context and dynamic preferences to generate personalized recommendations. We conducted quantitative and qualitative experiments on two real-world datasets and compared the proposed DSRS with several state-of-the-art methods. Experimental results clearly demonstrated the rationality and effectiveness of DSRS.

References

[1] Mnih A, Salakhutdinov R R. Probabilistic matrix factorization. In *Proc. the 21st Annual Conference on Neural Information Processing Systems*, December 2007, pp.1257-1264.

[2] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2008, pp.426-434.

[3] Rendle S, Freudenthaler C, Gantner Z *et al.* BPR: Bayesian personalized ranking from implicit feedback. In *Proc. the 25th Conference on Uncertainty in Artificial Intelligence*, June 2009, pp.452-461.

[4] Ma H, Yang H, Lyu M R *et al.* SoRec: Social recommendation using probabilistic matrix factorization. In *Proc.*

the 17th ACM Conference on Information and Knowledge Management, October 2008, pp.931-940.

[5] Ma H, Zhou D, Liu C *et al*. Recommender systems with social regularization. In *Proc. the 4th ACM International Conference on Web Search and Data Mining*, February 2011, pp.287-296.

[6] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proc. the 4th ACM Conference on Recommender Systems*, September 2010, pp.135-142.

[7] Guo G, Zhang J, Yorke-Smith N. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Proc. the 29th AAAI Conference on Artificial Intelligence*, January 2015, pp.123-129.

[8] Jiang M, Cui P, Wang F *et al*. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering*, 2014, 26(11): 2789-2802.

[9] Biao C, Tong X, Qi L *et al*. Study on information diffusion analysis in social networks and its applications. *International Journal of Automation and Computing*, 2018, 15(4): 377-401.

[10] Huang Z, Pan Z, Liu Q *et al*. An Ad CTR prediction method based on feature learning of deep and shallow layers. In *Proc. the 2017 ACM on Conference on Information and Knowledge Management*, November 2017, pp.2119-2122.

[11] Guo H, Tang R, Ye Y *et al*. DeepFM: A factorization-machine based neural network for CTR prediction. arXiv:1703.04247, 2017. https://arxiv.org/abs/1703.04247, Dec. 2019.

[12] He X, Liao L, Zhang H *et al*. Neural collaborative filtering. In *Proc. the 26th International Conference on World Wide Web*, April 2017, pp.173-182.

[13] Yu F, Liu Q, Wu S *et al*. A dynamic recurrent model for next basket recommendation. In *Proc. the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2016, pp.729-732.

[14] Hidasi B, Karatzoglou A, Baltrunas L *et al*. Session-based recommendations with recurrent neural networks. arXiv:1511.06939, 2015. https://arxiv.org/abs/1511.06939, Dec. 2019.

[15] Li J, Ren P, Chen Z *et al*. Neural attentive session-based recommendation. In *Proc. the 2017 ACM Conference on Information and Knowledge Management*, November 2017, pp.1419-1428.

[16] Hidasi B, Karatzoglou A. Recurrent neural networks with top-k gains for session-based recommendations. In *Proc. the 27th ACM International Conference on Information and Knowledge Management*, October 2018, pp.843-852.

[17] Lei C, Liu D, Li W *et al*. Comparative deep learning of hybrid representations for image recommendations. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.2545-2553.

[18] Hou M, Wu L, Chen E *et al*. Explainable fashion recommendation: A semantic attribute region guided approach. arXiv:1905.12862, 2019. https://arxiv.org/pdf/1905.12862.pdf, Dec. 2019.

[19] Kim D, Park C, Oh J *et al*. Convolutional matrix factorization for document context-aware recommendation. In *Proc. the 10th ACM Conference on Recommender Systems*, September 2016, pp.233-240.

[20] Wang Q, Li S, Chen G. Word-driven and context-aware review modeling for recommendation. In *Proc. the 27th ACM International Conference on Information and Knowledge Management*, October 2018, pp.1859-1862.

[21] Aral S, Muchnik L, Sundararajan A. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 2009, 106(51): 21544-21549.

[22] Sun P, Wu L, Wang M. Attentive recurrent social recommendation. In *Proc. the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2018, pp.185-194.

[23] Wu L, Sun P, Hong R *et al*. Collaborative neural social recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. doi:10.1109/TSMC.2018.2872842.

[24] Wu L, Sun P, Hong R *et al*. SocialGCN: An efficient graph convolutional network based model for social recommendation. arXiv:1811.02815, 2018. https://arxiv.org/abs/1811.02815, Dec. 2019.

[25] Fan W, Ma Y, Li Q *et al*. Graph neural networks for social recommendation. arXiv:1902.07243, 2019. https://arxiv.org/pdf/1902.07243.pdf, Dec. 2019.

[26] Qiu J, Tang J, Ma H *et al*. DeepInf: Social influence prediction with deep learning. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2018, pp.2110-2119.

[27] Wang H, Xu T, Liu Q *et al*. MCNE: An end-to-end framework for learning multiple conditional network representations of social network. In *Proc. the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, August 2019, pp.1064-1072.

[28] Xu T, Zhu H S, Zhong H *et al*. Exploiting the dynamic mutual influence for predicting social event participation. *IEEE Transactions on Knowledge and Data Engineering*, 2019, 31(6): 1122-1135.

[29] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. In *Proc. the 19th International Conference on World Wide Web*, April 2010, pp.811-820.

[30] Wang P, Guo J, Lan Y *et al*. Learning hierarchical representation model for next basket recommendation. In *Proc. the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2015, pp.403-412.

[31] Wu C Y, Ahmed A, Beutel A *et al*. Recurrent recommender networks. In *Proc. the 10th ACM International Conference on Web Search and Data Mining*, February 2017, pp.495-503.

[32] Li Z, Zhao H, Liu Q *et al*. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2018, pp.1734-1743.

[33] Sordoni A, Bengio Y, Vahabi H *et al*. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proc. the 24th ACM International on Conference on Information and Knowledge Management*, October 2015, pp.553-562.

[34] Manotumruksa J, Macdonald C, Ounis I. A contextual attention recurrent architecture for context-aware venue recommendation. In *Proc. the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2018, pp.555-564.

[35] Smirnova E, Vasile F. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proc. the 2nd Workshop on Deep Learning for Recommender Systems*, August 2017, pp.2-9.

[36] Manotumruksa J, Macdonald C, Ounis I. A contextual attention recurrent architecture for context-aware venue recommendation. In *Proc. the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2018, pp.555-564.

[37] Bruna J, Zaremba W, Szlam A *et al*. Spectral networks and locally connected networks on graphs. arXiv:1312.6203, 2013. https://arxiv.org/abs/1312.6203, Dec. 2013.

[38] Feng C, Liu Z, Lin S *et al*. Attention-based graph convolutional network for recommendation system. In *Proc. the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2019, pp.7560-7564.

[39] Zhang J, Shi X, Xie J *et al*. GaAN: Gated attention networks for learning on large and spatiotemporal graphs. arXiv:1803.07294, 2018. https://arxiv.org/pdf/1803.07294.pdf, Dec. 2019.

[40] Veličković P, Cucurull G, Casanova A *et al*. Graph attention networks. arXiv:1710.10903, 2017. https://arxiv.org/abs/1710.10903, Dec. 2019.

[41] Wu L K, Li Z, Zhao H K *et al*. Estimating early fundraising performance of innovations via graph-based market environment model. arXiv:1912.06767, 2019. https://arxiv.org/abs/1912.06767v1, Dec. 2019.

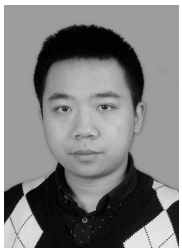
[42] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735-1780.

[43] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014. https://arxiv.org/abs/1412.6980, Dec. 2019.

[44] Richardson M, Agrawal R, Domingos P. Trust management for the semantic Web. In *Proc. the 2nd International Semantic Web Conference*, October 2003, pp.351-368.

[45] Scellato S, Noulas A, Mascolo C. Exploiting place features in link prediction on location-based social networks. In *Proc. the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2011, pp.1046-1054.

Yang Liu received her B.E. degree in information security from University of Science and Technology of China (USTC), Hefei, in 2016. She is currently working toward her Master's degree in the Anhui Province Key Laboratory of Big Data Analysis and Application, USTC, Hefei. Her research interests include deep learning and its application in recommender systems.



Zhi Li received his B.E. degree in software engineering from the Xi'an Jiaotong University, Xi'an, in 2015. He is currently pursuing his Ph.D. degree with the School of Data Science, University of Science and Technology of China, Hefei. His current research interest includes data mining, recommender systems and industrial intelligence. He has published papers in in refereed journals and conference proceedings, such as Journal of Computer Science and Technology, ACM SIGKDD, AAAI and IJCAI.

Wei Huang received his B.E. degree in software engineering from Sichuan University (SCU), Chengdu, in 2017. He is currently working toward his Ph.D. degree in the School of Data Science, University of Science and Technology of China (USTC), Hefei. His research interests include data mining, deep learning, natural language processing and applications in text classification, such as patent annotation.

Tong Xu received his Ph.D. degree in computer science in University of Science and Technology of China (USTC), Hefei, in 2016. He is currently working as an associate researcher of the Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China (USTC), Hefei. He has authored more than 40 journal and conference papers in the fields of social network and social media analysis, including TKDE, TMC, KDD, AAAI, ICDM, SDM, etc.

En-Hong Chen is a professor and vice dean of the School of Computer Science, University of Science and Technology of China (USTC), Hefei. His general area of research includes data mining and machine learning, social network analysis, and recommender systems. He has published more than 100 papers in refereed conferences and journals, including Nature Communications, IEEE/ACM Transactions, KDD, NIPS, IJCAI and AAAI, etc. He was on program committees of numerous conferences including KDD, ICDM, and SDM. He received the Best Application Paper Award on KDD-2008, the Best Research Paper Award on ICDM-2011, and the Best of SDM-2015. His research is supported by the National Science Foundation for Distinguished Young Scholars of China.

JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

Volume 35, Number 2, March 2020

Special Section on Learning and Mining in Dynamic Environments

Preface ..... *Min-Ling Zhang, Yu-Feng Li, and Qi Liu* (231)

Incremental Multi-Label Learning with Active Queries ..... *Shen-Jun Huang, Guo-Xiang Li, Wen-Yu Huang, and Shao-Yuan Li* (234)

Joint Label-Specific Features and Correlation Information for Multi-Label Learning ..... *Xiu-Yi Jia, Sai-Sai Zhu, and Wei-Wei Li* (247)

Discrimination-Aware Domain Adversarial Neural Network ..... *Yun-Yun Wang, Jian-Min Gu, Chao Wang, Song-Can Chen, and Hui Xue* (259)

Efficient Multiagent Policy Optimization Based on Weighted Estimators in Stochastic Cooperative Environments ..... *Yan Zheng, Jian-Ye Hao, Zong-Zhang Zhang, Zhao-Peng Meng, and Xiao-Tian Hao* (268)

Exploiting Structural and Temporal Influence for Dynamic Social-Aware Recommendation ..... *Yang Liu, Zhi Li, Wei Huang, Tong Xu, and En-Hong Chen* (281)

Semi-Supervised Classification of Data Streams by BIRCH Ensemble and Local Structure Mapping ..... *Yi-Min Wen and Shuai Liu* (295)

Sequential Recommendation via Cross-Domain Novelty Seeking Trait Mining ..... *Fu-Zhen Zhuang, Ying-Min Zhou, Hao-Chao Ying, Fu-Zheng Zhang, Xiang Ao, Xing Xie, Qing He, and Hui Xiong* (305)

Finding Communities by Decomposing and Embedding Heterogeneous Information Network ..... *Yue Kou, De-Rong Shen, Dong Li, Tie-Zheng Nie, and Ge Yu* (320)

Exploiting Multiple Correlations Among Urban Regions for Crowd Flow Prediction ..... *Qiang Zhou, Jing-Jing Gu, Chao Ling, Wen-Bo Li, Yi Zhuang, and Jian Wang* (338)

You Are How You Behave – Spatiotemporal Representation Learning for College Student Academic Achievement ..... *Xiao-Lin Li, Li Ma, Xiang-Dong He, and Hui Xiong* (353)

PetroKG: Construction and Application of Knowledge Graph in Upstream Area of PetroChina ..... *Xiang-Guang Zhou, Ren-Bin Gong, Fu-Geng Shi, and Zhe-Feng Wang* (368)

Special Section of ChinaSys 2019

Preface ..... *Wen-Guang Chen, Ying-Wei Luo, and Guang-Yu Sun* (379)

Optimistic Transaction Processing in Deterministic Database ..... *Zhi-Yuan Dong, Chu-Zhe Tang, Jia-Chen Wang, Zhao-Guo Wang, Hai-Bo Chen, and Bin-Yu Zang* (382)

MPI-RCDD: A Framework for MPI Runtime Communication Deadlock Detection ..... *Hong-Mei Wei, Jian Gao, Peng Qing, Kang Yu, Yan-Fei Fang, and Ming-Lu Li* (395)

Interference Analysis of Co-Located Container Workloads: A Perspective from Hardware Performance Counters ..... *Wen-Yan Chen, Ke-Jiang Ye, Cheng-Zhi Lu, Dong-Dai Zhou, and Cheng-Zhong Xu* (412)

IMPULP: A Hardware Approach for In-Process Memory Protection via User-Level Partitioning ..... *Yang-Yang Zhao, Ming-Yu Chen, Yu-Hang Liu, Zong-Hao Yang, Xiao-Jing Zhu, Zong-Hui Hong, and Yun-Ge Guo* (418)

Huge Page Friendly Virtualized Memory Management ..... *Sai Sha, Jing-Yuan Hu, Ying-Wei Luo, Xiao-Lin Wang, and Zhenlin Wang* (433)

Bigflow: A General Optimization Layer for Distributed Computing Frameworks ..... *Yun-Cong Zhang, Xiao-Yang Wang, Cong Wang, Yao Xu, Jian-Wei Zhang, Xiao-Dong Lin, Guang-Yu Sun, Gong-Lin Zheng, Shan-Hui Yin, Xian-Jin Ye, Li Li, Zhan Song, and Dong-Dong Miao* (453)

A Machine Learning Framework with Feature Selection for Floorplan Acceleration in IC Physical Design ..... *Shu-Zheng Zhang, Zhen-Yu Zhao, Chao-Chao Feng, and Lei Wang* (468)

SIES: A Novel Implementation of Spiking Convolutional Neural Network Inference Engine on Field-Programmable Gate Array ..... *Shu-Quan Wang, Lei Wang, Yu Deng, Zhi-Jie Yang, Sha-Sha Guo, Zi-Yang Kang, Yu-Feng Guo, and Wei-Xia Xu* (475)

JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

《计算机科学技术学报》

Volume 35 Number 2 2020 (Bimonthly, Started in 1986)

Indexed in: **SCIE, Ei, INSPEC, JST, AJ, MR, CA, DBLP**

Edited by:

THE EDITORIAL BOARD OF JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

Guo-Jie Li, Editor-in-Chief, P.O. Box 2704, Beijing 100190, P.R. China

Managing Editor: Feng-Di Shu E-mail: jcst@ict.ac.cn http://jcst.ict.ac.cn Tel.: 86-10-62610746

Sponsored by: Institute of Computing Technology, CAS & China Computer Federation

Supervised by: Chinese Academy of Sciences

Undertaken by: Institute of Computing Technology, CAS

Published by: Science Press, Beijing, China

Printed by: Beijing Kexin Printing House

Distributed by:

China: All Local Post Offices

Other Countries: Springer Nature Customer Service Center GmbH, Tiergartenstr. 15, 69121 Heidelberg, Germany

Available Online: https://link.springer.com/journal/11390

ISSN 1000-9000

国内统一刊号: CN11-2296/TP 国内邮发代号: 2-578 RMB ¥150.00