Personalizing Graph Neural Networks with Attention Mechanism for Session-based Recommendation

Shu Wu, Mengqi Zhang, Xin Jiang, Xu Ke and Liang Wang

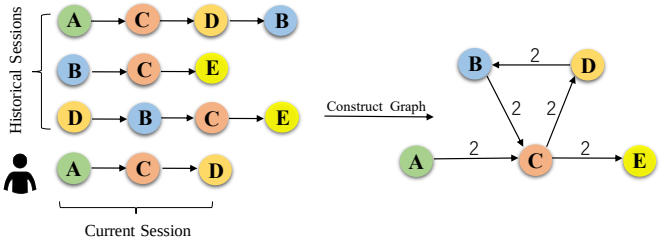
Abstract—The problem of personalized session-based recommendation aims to predict users' next click based on their sequential behaviors. Existing session-based recommendation methods only consider all sessions of user as a single sequence, ignoring the relationship of among sessions. Other than that, most of them neglect complex transitions of items and the collaborative relationship between users and items. To this end, we propose a novel method, named Personalizing Graph Neural Networks with Attention Mechanism, A-PGNN for brevity. A-PGNN mainly consists of two components: One is Personalizing Graph Neural Network (PGNN), which is used to capture complex transitions in user session sequence. Compared with the traditional Graph Neural Network (GNN) model, it also considers the role of users in the sequence. The other is Dot-Product Attention mechanism, which draws on the attention mechanism in machine translation to explicitly model the effect of historical sessions on the current session. These two parts make it possible to learn the multi-level transition relationships between items and sessions in user-specific fashion. Extensive experiments conducted on two real-world data sets show that A-PGNN significantly outperforms the state-of-the-art personalizing session-based recommendation methods consistently.

Index Terms—Graph Neural Networks, Attention, Session-based Recommendation

1 Introduction

With rapid growth of the amount of information on the Internet, recommender system have become fundamental technique to help users alleviate the problem of information overload. In different applications, e.g., searching engine, e-commerce, and streaming media sites, recommender systems help users to select interesting information. In many services, users' identifications are available, such as Reddit, Xing and YouTube. Users' behaviors in past session and profile can be utilized for next item recommendation under these platforms. In real world scenarios, users' most recent behaviors in the current session often reflect the short-term preference, while users' historical session sequences imply the evolution of users' long-term preference over time. Therefore, combining the short-term preference extracted from user's current session with the long-term preference extracted from the historical session sequence is critical in personalized session-based recommendation task.

Due to the highly practical value, many kinds of session-based recommendation methods have been developed. In recent years, many researches based on deep learning [1]–[4] apply Recurrent Neural Networks (RNNs) for session-based recommendation systems and obtain promising results. However, it is difficult to capture users' long-term preferences to make predictions by only relying on a sample RNN model. To consider both user's long-term and short-term preferences, some work [5], [6] design memory mechanisms combined with sequential model to capture user preference. Some attention models [7], [8] are also applied to the sequence recommendation scenarios to capture long and short-term preferences. But they ignore the hierarchy between user interest evolution: session and item. Recent work [9], [10] uses a Hierarchical RNN to capture the flow of user interest between sessions. Analogous to model Hierarchical RNN [9], II-RNN model [10] also utilizes multiple RNN to model interest relationships between user sessions.



(a) User interaction sessions (b) User behavior graph

Fig. 1. User behavior graph construction diagram. In (a), user's interaction sessions includes historical sessions and current session. (b) is a schematic diagram of the user behavior graph. The number on the side represents the number of times the side appears.

We argue that these state-of-the-art models based on sequence only considers the simple sequence relationship of the items within user's single session. They fail to model the global complex relationship between items among all the sessions. SR-GNN [11] utilize the Graph Neural Networks to capture the complex item transition relationships in session-based recommendation, but it ignore the role of user in the item transition relationship. For the sake of easy understanding, we simply illustrate the personalized recommendation scenario through Figure 1. Figure 1(a) represents all the user u's interaction sequences, including current and historical session sequences. The session sequences can be converted into a graph, Figure1(b), called user behavior

graph. We can find that the graph structure have stronger informative power than sequence structure. For example, in Figure 1(b), B, C and D compose a strongly connected component in user behavior graph, which reflects their dense link relationships. For this case, traditional sequence-based models are less capable in capturing this complex structural relationship. While our model, for each user, their user behavior graph reflects the user's personalized behavior, which makes it easier to personally model each user sequence than traditional graph model [11]. In addition, it is critical to model the impact of user historical sessions on user current session. Although these state-of-the-art work [9], [10] fuse the representations of all historical sessions into a vector to assist the current session in predicting, it is difficult to explicitly distinguish the effects of different historical sessions on current session.

To this end, we propose a novel method Personalizing Graph Neural Networks with Attention Mechanism (A-PGNN), which contains two main components: personalizing graph neural network (PGNN) and Dot-Product Attention mechanism. We first convert all sessions of each user to a graph and then feed it into PGNN to capture the global transition relationships of items. The Dot-Product Attention mechanism is used to explicitly model the effect of different historical sessions on the current session. Figure 2 illustrates the workflow of the proposed A-PGNN model. The details will be introduced in the section 3. Extensive experiments conducted on real-world representative data sets demonstrate the effectiveness of the proposed method over the state-of-art methods. The main contributions of this work are summarized as follows:

- We design a new graph neural network PGNN for personalized session-based recommendation scenario, which is able to capture complex item transitions in user-specific fashion.
- We use the Dot-Product attention mechanism to explicitly model the effect of user's historical interests on current session, which significantly improves the performance of personalized session-based recommendation task.
- We conduct empirical studies on two real-world data sets. Extensive experiments demonstrate the validity of different components of our model and show that A-PGNN evidently outperforms the state-of-art methods.

2 Related Work

In this section, we review some related work on session-based recommendation systems, including conventional methods, deep-learning-based methods. Then, we introduce some recent work on graph neural networks.

2.1 Conventional recommendation methods

Matrix factorization [12]–[14] is a general approach to recommendation systems. The basic objective is to factorize a user-item rating matrix into two low-rank matrices, each of which represents the latent factors of users or items. However, it is not very suitable for the session-based recommendation, because the user preference is only provided by some positive clicks. The item-based neighborhood methods [15] are natural solution, in which item similarities are calculated on the co-occurrence in the same session. However, These methods have difficulty in considering the sequential order of items and generate prediction merely based on the last click. Then, the sequential methods based on Markov chains are proposed, which predict users' next behavior based on the previous ones. Treating recommendation generation as a sequential optimization problem, [16] employ Markov decision processes (MDPs) for the solution. Via factorization of the personalized probability transition matrices of users, FPMC [17] models sequential behavior between every two adjacent clicks and provides a more accurate prediction for each sequence. However, the main drawback of Markov-chain-based models is that they combine past components independently. Such an independence assumption is too strong and thus confines the prediction accuracy.

2.2 Deep-learning-based methods

Recently, some prediction models, especially language models [18] are proposed based on neural networks. Among numerous language models, the recurrent neural network (RNN) has been the most successful one in modeling sentences [19] and has been applied in various natural language processing tasks, such as machine translation [20], conversation machine [21], and image caption [22]. RNN also has been applied successfully in numerous applications, such as the sequential click prediction [23], location prediction [24], and next basket recommendation [25].

For session-based recommendation, the work of [1] proposes the recurrent neural network approach, and then extends to an architecture with parallel RNNs [26] which can model sessions based on the clicks and features of the clicked items. After that, some work is proposed based on these RNN methods. An improved RNN [2] enhances the performance of recurrent model by using proper data augmentation techniques and taking temporal shifts in user behavior into account. The work [27] combine the recurrent method and the neighborhood-based method together to mix the sequential patterns and co-occurrence signals. In addition to its application in the field of computer vision, convolutional neural networks are also used in sequence recommendations. For example, model 3D-CNN [3] incorporates session clicks with content features, such as item descriptions and item categories, to generate recommendations by using 3-dimensional convolution neural networks. Besides, A list-wise deep neural network [28] models the limited user behavior within each session, and uses a list-wise ranking model to generate the recommendation for each session.

Furthermore, a neural attentive recommendation machine with an encoder-decoder architecture, i.e. NARM [4], employs the attention mechanism on RNN to capture users' features of sequential behavior and main purposes. Then, a short-term attention priority model (STAMP) [7] using simple MLP networks and an attentive net, is proposed to efficiently capture both users' general interests and current interests. Xu Chen [5] designed a memory-augmented neural network, which store and update users' historical records

by leveraging the external memory matrix. Jin Huang [6] proposed a model that integrates the RNN-based networks with Key-Value Memory Network to capture sequential user preference and attribute-level user preference. SHAN model [8] uses a two-layer hierarchical attention network, which takes the long- and short-term preference into account. But they do not model the relationship between users' historical session and current session, which is more important for capturing users' long-term preferences. Recent work [9], [10] uses a Hierarchical RNN to capture the flow of user interest between sessions. Analogous to model Hierarchical RNN [9], II-RNN model [10] also utilizes multiple RNN to model interest relationships between user sessions, but the limitation of RNN model limits the capture ability of long-term interests.

TABLE 1
Important notations

Notation	Description
V	the set of items
U	the set of users
$\mathcal{S}^u$	user u's all sessions set
S_i^u	the i-th session of user u's all sessions
$\mathcal{S}_{\mathrm{h}}^u$	the historical sessions of user u
$\mathcal{S}_{\mathrm{c}}^u$	the current session of user u
$\mathcal{G}^u$	the user behavior graph of user u
e_{v_i}	the embedding of item v_i
e_u	the embedding of user u
z_u	the unified representation of user u

2.3 Graph neural networks

Nowadays, neural network has been employed for generating representation for graph-structured data, e.g., social network and knowledge bases. Extending the word2vec [18], an unsupervised algorithm DeepWalk [29] is designed to learn representations of graph nodes based on random walk. Following DeepWalk, unsupervised network embedding algorithms LINE [30] and node2vec [31] are most representative methods. On the another hand, the classical neural network CNN and RNN are also deployed on graph-structured data. [32] introduces a convolution neural network that operates directly on graphs of arbitrary sizes and shapes. A scalable approach [33] chooses the convolution architecture via a localized approximation of spectral graph convolutions, which is an efficient variant and can operate on graphs directly as well. However, these methods can only be implemented on undirected graphs. Previously, in form of recurrent neural networks, Graph Neural Networks (GNNs) [34], [35] are proposed to operate on directed graphs. As a modification of GNN, gated GNN [36] uses gated recurrent units and employs back-propagation through time (BPTT) to compute gradients. Graph Attention Networks (GAT) [37] applies the attention mechanism to learn the weight of nodes and neighbor nodes. Recently GNN is broadly applied for the different tasks, e.g., script event prediction [38], situation recognition [39], recommender system [11] and image classification [40].

GNN has advantages in processing graph structure data and can be used to capture more abundant information in sequence data. SR-GNN [11] is the first model that utilize the Gate Graph Neural Networks to capture the complex item transition relationships in session-based recommendation, but it ignore the role of user in item transition relationship, it is also difficult to use user historical session information to improve recommendation performance. In this work, we proposed a model A-PGNN based on improved GGNN, whcih is more suitable for personalized session-based recommendation scenarios.

3 The Proposed Method

In this section, we introduce the proposed A-PGNN[1] which applies personalizing graph neural networks with attention mechanism for personalized session-based recommendation. First, the problem is formulated. Then we introduce the overview of our proposed method and more details.

3.1 Problem Formulation

Let $V = \{v_i\}_{i=1}^{|V|}$ and $U = \{u_i\}_{i=1}^{|U|}$ be the set of items and users in the system, respectively. We describe item v_i with embedding vector $e_{v_i} \in R^d$ and user u with embedding vector $e_u \in R^{d'}$, d and d' are the dimensions of the embedding of item and user. For each user u, his all session sequence can be represented as $\mathcal{S}^u = \{S_i^u\}_{i=1}^{n_u}$, where n_u stands for the total number of sessions for a user u, for convenience, n_u is abbreviated as n, historical session and current session sequence are denoted as $\mathcal{S}_{\mathrm{h}}^u = \{S_i^u\}_{i=1}^{n-1}$, $\mathcal{S}_{\mathrm{c}}^u = \{S_n^u\}$ respectively. Each session $S_i^u = \{v_{i,j}\}_{j=1}^{m_i} \in \mathcal{S}^u$ represents the sequence in which the user is interacting, $v_{i,j} \in V$ represents an interactive item of the user within the session S_i^u, and m_i is the number of items in session S_i^u. Sessions and items are all ordered by timestamps. Given users' all sessions $\mathcal{S}^u$, the goal of personalized session-based recommendation is to predict the next interactive item $v_{n,m+1}$ of the current session $\mathcal{S}_{\mathrm{c}}^u$. The notations used throughout this paper are summarized in Table 1.

3.2 Overview

Fig2 is the overview of our proposed method. Each user u's all sessions $\mathcal{S}^u$ can be modeled as a user behavior graph $\mathcal{G}^u$ (Section 3.3). We then input user behavior graph, item embedding and user embedding into Personalizing Graph Neural Network (PGNN) (Section 3.4) to capture transitions of items with respect to user u. After that, we use the max-pooling layer to get the session embedding, and the historical session embedding matrix can be obtained. Then, we calculate the explicit impact of the historical session on the current session through Dot-Product attention layer. Thereby, we can get users' dynamic interest representations (Section 3.5). In the end, we concatenate users' embedding with the dynamic interest representations to form unified representations of users (Section 3.6). Using the representations, we output probability $\hat{\mathbf{y}}$ for all candidate items, where the element $y_i \in \hat{\mathbf{y}}$ is the recommendation score of the corresponding item $v_i \in V$ (Section 3.7). The items with top-k values will be the candidate items for recommendation.

[1] https://github.com/CRIPAC-DIG/A-PGNN

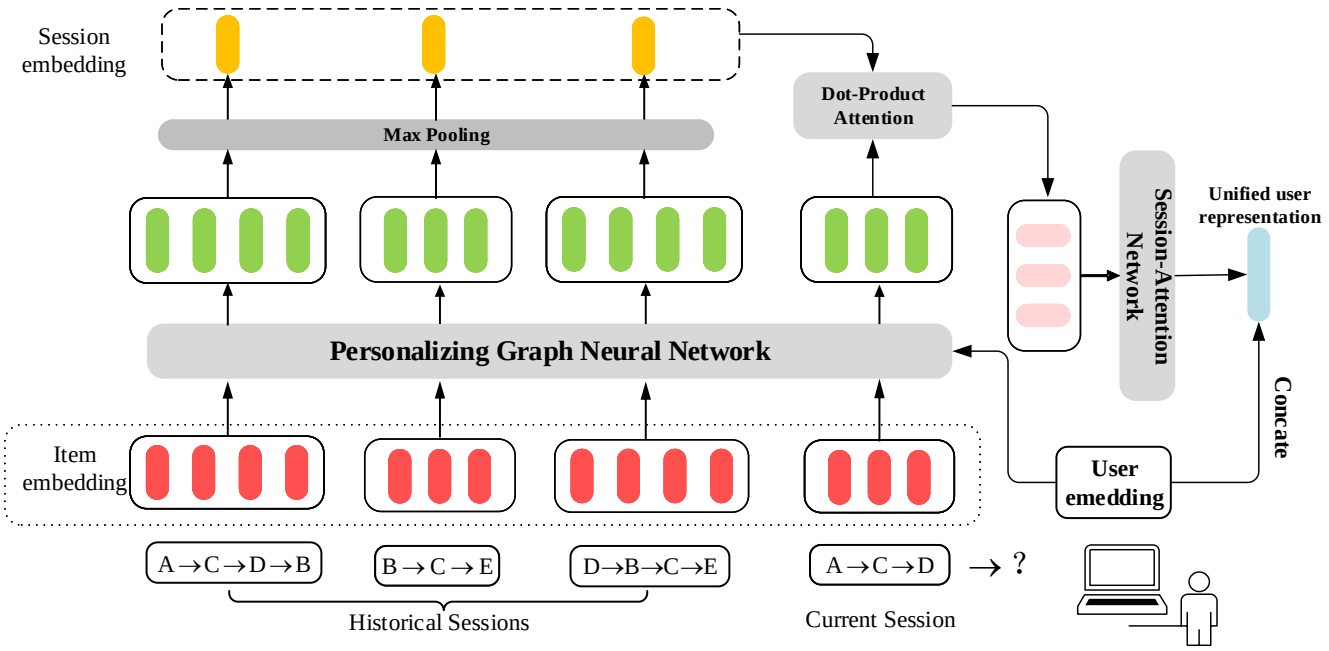


Fig. 2. The framework of A-PGNN. Based on the user's all sessions, we first construct a user behavior graph, where each node represents an item. We then input user behavior graph, items and user embedding vectors into PGNN to obtain the item representations that incorporates the global transition relationships of other items. Next, we utilize the max pooling layer to get the historical session embedding. Following this, the representation of the current session that incorporates user's historical preferences can be obtained by the Dot-product attention component. Finally, we put the obtained current session embedding into Session-Attention component to generate user's dynamic interest representation vector and combine it with user's static embedding to get a unified user representation for final prediction.

3.3 User Behavior Graph

The graph structure data contains richer information compared with the sequence structure data. To fully capture the transition relationships between items with respect to each user using the graph neural network, we construct graph $\mathcal{G}^u$ for each user. As shown in Fig1 (a) and (b), for each user u, we model all of his/her sessions $\mathcal{S}^u$ as a directed graph $\mathcal{G}^u = (\mathcal{V}^u, \mathcal{E}^u)$. In each user behavior graph $\mathcal{G}^u$, node i represents an item $v_i \in V$ that user u interacted with. The edge between nodes v_i and v_j represents a user interacts item v_i after v_j in one of his sessions. For this case, we assume that the effect of v_i on v_j and the effect of v_j on v_i are different, which produces two types of edges that represent two different transition relationships. One directed edge e_{ij}^{out} called outgoing edge with weights of ω_{ij}^{out} and the other directed edge called incoming edge e_{ji}^{in} with weights of ω_{ji}^{in}. Their weights can be computed by:

$$\omega_{ij}^{out} = \frac{\text{count}\left(v_i, v_j\right)}{\sum_{k \in \mathcal{N}_{out}(i)} \text{count}\left(v_i, v_k\right)}, \tag{1}$$

$$\omega_{ij}^{in} = \frac{\text{count}\left(v_j, v_i\right)}{\sum_{k \in \mathcal{N}_{in}(i)} \text{count}\left(v_k, v_i\right)}, \tag{2}$$

where function count(x, y) is used to calculate the number of occurrences that user interacts item x after interacts item y. $\mathcal{N}_{in}(i)$ is the set of predecessor nodes v_j with incoming edge e_{ji}^{in}. $\mathcal{N}_{out}(i)$ is the set of successor node v_j with outcoming edge e_{ij}^{out}. The topological structure of user behavior graph $\mathcal{G}^u$ can be represented by two adjacency matrices, which can be written as:

$$A_{\text{out}}[i, j] = \omega_{ij}^{out}, \tag{3}$$

$$A_{\text{in}}[i, j] = \omega_{ij}^{in}. \tag{4}$$

3.4 Personalizing Graph Neural Network

Here, we present personalizing graph neural networks (PGNN), which uses the improved graph neural network (GNN) to personally learn the complex transition relationships between each user's items, and then obtain the representation of items and users. The vanilla GNN is first proposed in [35], extending neural network methods for processing the graph-structured data. [36] [36] further introduce gated recurrent unites and propose gated graph neural network (GGNN). First applying the GNN for recommendation task, SR-GNN [11] obtains the session graphs considering rich node connections, and employs GNN for automatically extracting useful features of items and user behaviors. But the graph neural network in SR-GNN is not suitable for personalized recommendation scenarios, nor can it take advantage of the user's personalized information. In order to solve this limitation, we proposed the PGNN.

Different users have different behavior patterns, which results in different item transition relationships. We consider user factor in the item interaction when designing PGNN architecture. At each time of node update, we fuse user embedding e_u with the current representation of node item h_i^{t-1}. For example, at t time, the aggregated incoming and

outcoming information of node i can be formulated as:

$$a_{\text{in}_i}^{(t)} = A_{\text{in}_i}^{u\ \top}\left[\text{Concat}\left(h_1^{(t-1)}, e_u\right), \ldots, \text{Concat}\left(h_n^{(t-1)}, e_u\right)\right] \cdot W_{\text{in}}, \tag{5}$$

$$a_{\text{out}_i}^{(t)} = A_{\text{out}_i}^{u\ \top}\left[\text{Concat}\left(h_1^{(t-1)}, e_u\right), \ldots, \text{Concat}\left(h_n^{(t-1)}, e_u\right)\right] \cdot W_{\text{out}} \tag{6}$$

$$a_i^{(t)} = \text{Concat}\left(a_{\text{in}_i}^{(t)}, a_{\text{out}_i}^{(t)}\right), \tag{7}$$

where A_{out}^u and A_{in}^u represent adjacency matrix of outgoing and incoming edges in user behavior graph. $A_{\text{in}_i}^u$ and A_{out_i} extract the i^{th} row of matrix A_{in} and A_{out}, which indicates the adjacent relationship between node i and other nodes. Since $\mathcal{G}^u$ is bi-direction, we consider two parameters W_{in} and $W_{\text{out}} \in R^{(d+d')\times\hat{d}}$, which respectively encode the user and item connection vectors to two different $\hat{d}$-dimensional vectors for bidirectional propagation of information between nodes.

After the information propagation between the nodes is defined, we employ gated recurrent units (GRUs) [20] to incorporate information from other nodes with hidden states of the previous timestep, and update each node's hidden state:

$$z_i^{(t)} = \sigma\left(W_z a_i^{(t)} + U_z h_i^{(t-1)}\right), \tag{8}$$

$$r_i^{(t)} = \sigma\left(W_r a_i^{(t)} + U_r h_i^{(t-1)}\right), \tag{9}$$

$$\widetilde{h_i^{(t)}} = \tanh\left(W_o a_i^{(t)} + U_o\left(r_i^{(t)} \odot h_i^{(t-1)}\right)\right), \tag{10}$$

$$h_i^{(t)} = \left(1 - z_i^{(t)}\right) \odot h_i^{(t-1)} + z_i^{(t)} \odot \widetilde{h_i^{(t)}}, \tag{11}$$

where z_i^t and r_i^t are update and reset gate, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ is the element-wise multiplication operator.

After a total of T propagation steps, the final hidden state vector $h_i^{(T)}$ of each node i can be obtained in graph $\mathcal{G}^u$. For convenience, we use h_i instead of $h_i^{(T)}$. The final hidden state of each node not only contains its node features, but also aggregates the information from its T-order neighbors.

3.5 Generating User's Dynamic Interest Representation via Attention Networks

Some methods [9], [11] only consider the singe-level sequence models or mix all historical session embedding into one embedding to predict users' next interactions based on their current session. However, they omits the explicit impacts of the historical sessions on the current session. For example, if a user previously browsed or clicked a digital camera on a shopping site, his current clicked items are SD and Micro-SD cards. In this case, there is a strong relationship between the historical session and the current session item. If his current interaction is with automotive products, the camera and automotive fall into two unrelated categories, which shows the historical sessions have a minor effect on the current session. In conclusion, explicit modeling of the relationship between historical sessions and current session projects is of great significance for capturing user behavior patterns. In this section, we discuss how to apply the attention mechanism to model the impact of historical sessions on the current session.

3.5.1 Calculating the impact of historical sessions on the current session

We resort to Transformer network [41] which is widely-used in some popular neural machine translation models to address the problem discussed above. In our model, we use the scaled dot-product attention mechanism which is the core of Transformer network, to complete the calculation of the impact of historical sessions on current session.

Dot-Product Attention: The input of the scaled dot-product attention consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the *query* with all *keys*, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weight on values. The scaled dot-product attention is formally defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{12}$$

where Q, K, V represent the queries, keys, and values respectively, and the scale factor $\sqrt{d}$ is to avoid exceedingly large dot products and speed up convergence.

We model the historical session sequence $\mathcal{S}_{\text{h}}^u$ and current session S_{c}^u by using hidden state vector $h_i, \forall v_i \in V$ of nodes. The embedding vector of historical session $S_i^u = \{v_{i,1}, v_{i,2}, \ldots, v_{i,m_i}\}$ in $\mathcal{S}_{\text{h}}^u$ can be represented as $f_i^u \in \mathbb{R}^d$, which can be calculated by max-pooling:

$$f_{i,j}^u = \max_{1\le j\le d}\left(h_{1,j}, h_{2,j}, \ldots, h_{m_i,j}\right). \tag{13}$$

Therefore, historical session sequence $\mathcal{S}_{\text{h}}^u = \{S_1^u, S_2^u, \ldots, S_{n-1}^u\}$ can be expressed as an embedded matrix $F^u = \left[f_1^u, f_2^u, \ldots, f_{n-1}^u\right]$. For current session S_{c}^u, we simply denote the embedding matrix as $H^u = [h_1, h_2, \ldots, h_m]$. In our context, we use current session embedding to query historical session embedding, where the queries Q are determined by H^u, the keys and values are determined by F^u. In specical, we project F^u and H^u to the same latent space through nonlinear transformation:

$$\begin{aligned} Q^u &= \text{Relu}\left(H^u W^Q\right), \\ K^u &= \text{Relu}\left(F^u W^K\right), \\ V^u &= \text{Relu}\left(F^u W^V\right), \end{aligned} \tag{14}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d\times d}$ are the projection matrices. The effect of the historical sessions on current session can be calculated by

$$H_h = \text{Attention}\left(Q^u, K^u, V^u\right). \tag{15}$$

After the effect of history session on each item in the current session sequence is calculated, we then compute the embedding of the current session as follows:

$$H^{u'} = H_h + H^u. \tag{16}$$

Then, the current session embedding $H^{u'}$ can be rewritten as $\left[h_1', h_2', \ldots, h_m'\right]$.

3.5.2 Generating user dynamic representation

The current embedding matrix $H^{u'}$ combines long- and short-term interests of users. In the following part, we describe how to encode the embedding matrix to the user dynamic representation vector for next-item recommendation

task. Similar to the model SR-GNN [11], we use the attention mechanism to encode the current embedding matrix to global representation and local representation respectively, where global representation denotes the user's general interest and local representation denotes the user's new interest. The global representation is defined as:

$$z_g = \sum_{i=1}^{n} \alpha_i h'_i, \tag{17}$$

$$\alpha_i = W_0 \sigma \left(W_1 h'_n + W_2 h'_i + b_c \right), \tag{18}$$

where parameters $W_0 \in \mathbb{R}^d$, $W_1, W_2 \in \mathbb{R}^{d \times d}$ control the weights of item embedding vectors, $b_c \in R^d$ is a bias vector, $\sigma(\cdot)$ denotes the sigmoid function and weighted coefficient α_i determine the weights of item of current session when making predictions. The local representation can be simply defined as the embedding of last clicked item, and can be written as:

$$z_l = h'_n. \tag{19}$$

After that, we compute the user's dynamic representation as:

$$z_d = \text{Concat}\left(z_g, z_l\right). \tag{20}$$

3.6 Generating Unified User Representation

The embedding e_u implies the inherent attributes of users and can be regarded as the static representation of users. So, we concatenate the dynamic and the static representation of users in one vector, and then get the unified representation of users through linear transformation:

$$z_u = \text{Concat}\left(z_d, e_u\right) \cdot B, \tag{21}$$

where matrix $B \in \mathbb{R}^{(2d+d') \times d}$ compresses two combined embedding vectors into the latent space $\mathbb{R}^d$, and d, d' are the dimension of item and user embedding respectively.

3.7 Making Recommendation

After obtained the unified representation of user u, we compute the score $\hat{z}_i$ for each candidate item $v_i \in V$ by multiplying its embedding v_i by the user representation S, which can be defined as:

$$\hat{z}_i = {z_u}^\top e_{v_i}. \tag{22}$$

Then we apply a softmax function to get the output vector of the model:

$$\hat{\mathbf{y}} = \text{softmax}(\hat{z}), \tag{23}$$

where $\hat{z} \in R^{|V|}$ denotes the recommendation scores over all candidate items and $\hat{y}$ denotes the probabilities that items will be interacted by user u next time in the current session S_c^u.

For any given user behavior graph, the loss function is defined as the cross-entropy of the prediction and the ground truth. It can be written as follows:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{|V|} y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right), \tag{24}$$

where y denotes the one-hot encoding vector of the ground truth item. Finally, we use the back-propagation through time (BPTT) algorithm to train the proposed A-PGNN method.

4 Experiments

In this section, we first present the experimental datasets, baselines, evaluation metrics and parameter settings. Then we assess the proposed method with compared methods, and compare the performance of different ablation model. Besides, we analyze the performance results with different lengths of current sessions and different numbers of user history sessions. In this section, we intend to answer the following questions through experiments.

- **RQ1:** How does our model A-PGNN perform compared with other state-of-the-art models?
- **RQ2:** What is the influence of various components in our model?
- **RQ3:** What are the influence of different hyper-parameter settings (the number of PGNN layers and the maximum user historical session length) for our model A-PGNN?
- **RQ4:** What are the effect of current and historical session length on recommendation tasks?

4.1 Datasets

We used two different real-word datasets for our experiments. The first is the Xing data [42], which is collection from RecSys Challenge 2016[2], we named this dataset Xing. The second is a dataset [10] of user activity on the social news aggregation and discussion website, the Reddit[3] data.

Xing. The Xing Data contains interactions on job postings for 770k users over a 80-days period. In this data, user behaviors include click, bookmark, reply and delete. Similar to [9], We split the Xing data into session by 30-minute idle threshold, and discarded repeated interactions having type "delete" and repeated interactions of the same type within sessions. We removed items with support less than 20 and users with session length < 2 and kept > 5 sessions to have sufficient cross-session information for proper modeling of return users.

Reddit. The Reddit dataset contains tuples of user name, a subreddit where the user made a comment to a thread, and a timestamp for the interaction. We also had to split each user's records into sessions manually for the Reddit dataset. We used the same approach as of the Xing dataset, but here we used 60-minutes as the time threshold.

Then we preprocessed both datasets as follows: For each user, 80% of his sessions were placed in the training set. The test set is build with the remaining session. We further partitioned the training set with the same procedure to tune the hyper-parameters of the algorithms. The statistics of two datasets after the preprocessing steps are shown in Table 2. Due to the principle of the algorithm, we segment each user's sessions $\mathcal{S}^u$ into a series of sequences and labels. For example, for an input $\mathcal{S}^u$ of user u's all sessions, where $\mathcal{S}^u_{\text{h}} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}, v_{3,3}\}\}$,

[2] http://2016.recsyschallenge.com/
[3] https://www.kaggle.com/colemaclean/subreddit-interactions

TABLE 2
Statistics of datasets after preprocessing

Dataset	Xing	Reddit
users	11479	18271
items	59121	27452
Sessions	91683	1135488
Average session length	5.78	3.02
Sessions per user	7.99	62.15
Train sessions	69135	901161
Test sessions	22548	234327

$\mathcal{S}^u_c = \{\{v_{c,1}, v_{c,2}, v_{c,3}\}\}$, we generate historical sessions, current sessions and labels:

$$\mathcal{S}^u_{h_1} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}\},$$
$$\mathcal{S}^u_{c_1} = \{\{v_{2,1}\}\}, \text{label}_1 : v_{2,2};$$
$$\mathcal{S}^u_{h_2} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}\},$$
$$\mathcal{S}^u_{c_2} = \{\{v_{3,1}\}\}, \text{label}_2 : v_{3,2};$$
$$\mathcal{S}^u_{h_3} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}\},$$
$$\mathcal{S}^u_{c_3} = \{\{v_{3,1}, v_{3,2}\}\}, \text{label}_3 : v_{3,3};$$
$$\mathcal{S}^u_{h_4} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}, v_{3,3}\}\},$$
$$\mathcal{S}^u_{c_4} = \{\{v_{c,1}\}\}, \text{label}_4 : v_{c,2};$$
$$\mathcal{S}^u_{h_5} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}, v_{3,3}\}\},$$
$$\mathcal{S}^u_{c_5} = \{\{v_{c,1}, v_{c,2}\}\}, \text{label}_5 : v_{c,3};$$

where the label is the next interaction item for current session.

4.2 Compared Methods

To evaluate the performance of the proposed method, we compare it with the following representative methods:

POP recommend the top-K frequent items in the training set.

Item-KNN [15] recommends items similar to the previously clicked ones in the session, where similarity is defined as the cosine similarity between the vector of sessions.

FPMC [17] is a sequential prediction method based on the personalized markov chain considering the items in the current session.

GRU4Rec [1] uses RNNs to model user sequences for the session-based recommendation.

H-RNN [10], [42] are similar in model architecture, and they are all hierarchical RNNs for personalized session-based recommendations, which are based on GRU4Rec and adds an additional GRU layers to model information across the user's sessions for tracking the evolution of the user's interest. Therefore, we only select the best results of the two models as a comparison, collectively referred to as H-RNN.

SR-GNN [11] utilize the Gate GNN model to capture the complex transition relationships of items for the session-based recommendation.

4.3 Evaluation Metrics

Following metrics are used to evaluate compared methods, which are also widely used in other related works [10], [42].

Recall@K (Precision) is widely used as a measure of predictive accuracy. It represents the proportion of correctly recommended items amongst the top-K items.

MRR@K (Mean Reciprocal Rank) is the average of reciprocal ranks of the correctly-recommended items. The reciprocal rank is set to 0 when the rank position exceeds K. The MRR measure considers the order of recommendation ranking, where large MRR value indicates that correct recommendations are in the top of the ranking list.

We used Recall@K and MRR@K with $K = 5, 10, 20$ to evaluate all compared methods.

4.4 Parameter Setup

We set the dimension of item $d = 100$ for Xing data, $d = 50$ for Reddit data, and set user embedding dimension $d' = 50$ for both data set. According to the data processing method of the Section4.1, the maximum length of current session is 20. For Xing and Reddit data, we set the maximum historical session length to 50 and 30 respectively. For the PGNN's propagation step, we set the Xing and Reddit data sets to 1 and 3 respectively. All parameters are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. The model is trained with Adam [43] optimizer, with learning rate 0.001, L2 penalty 0 and batch size 100. In particular, we use batch normailzation [44] between dot-attention layer and session-attention layer to prevent overfitting for smaller Xing data. For the baseline methods, we use the default hyperparameters in addition to the dimension.

4.5 Performance Comparison (RQ1)

First, for question **RQ1**, to demonstrate the overall performance of the proposed model, we compare it with other state-of-the-art personalized recommendation methods. The overall performance in terms of Recall@K and MRR@K is shown in Table 3. It is obvious that the proposed graph-based model A-PGNN method achieves the best performance among all methods on the two datasets in terms of Recall@K and MRR@K. The performance on different datasets can reflect different superiorities of our model.

As prior study [42] on dataset Xing have already shown, users' activity within and across sessions has a high degree of repetitiveness. From the results, we can see that A-PGNN and SR-GNN significantly outperform GRU4Rec and H-RNN by large margins (up to 4% better Recall and 30% better MRR) which highlights the superiority of graph-based models in this scenario. Based on graphs converted by sessions of each user, graph-based models are capable of capturing transition relationships of items and generating accurate item embedding vectors correspondingly, which are difficult to be revealed by the conventional sequential methods, like MC-based and RNN-based methods.

On another dataset Reddit, A-PGNN and H-RNN perform best. We attribute the success of these two models to their excellent ability to model the effect of userâĂŹs historical interests on current session. But they act in an explicit or implicit way. As for H-RNN, it fuses the representations of all historical sessions into a single vector to assist the current session in predicting which implicitly treats all historical sessions equally and uses all previous

TABLE 3
Recommendation performance comparison on Xing data and Reddit data. The best performance on each dataset is highlighted.

Dataset	Xing						Reddit					
	Recall@5	Recall@10	Recall@20	Mrr@5	Mrr@10	Mrr@20	Recall@5	Recall@10	Recall@20	Mrr@5	Mrr@10	Mrr@20
Pop	0.21	0.26	0.58	0.08	0.09	0.11	13.22	19.46	26.47	8.50	9.32	9.82
Item-KNN	8.79	11.85	14.67	5.01	5.42	5.62	21.71	30.32	38.85	11.74	12.88	13.49
FPMC	1.70	2.42	3.27	0.61	0.50	0.37	29.91	34.31	44.32	8.78	6.56	4.54
GRU4Rec	10.35	13.15	15.30	5.94	6.36	6.69	33.72	41.73	50.04	24.36	25.42	26.00
SR-GNN	13.38	16.71	19.25	8.95	9.39	9.64	34.96	42.38	50.33	25.90	26.88	27.44
H-RNN	10.74	14.36	17.64	7.22	7.78	8.83	44.76	53.44	61.80	32.13	33.29	33.88
A-PGNN	**14.38**	**17.06**	**19.98**	**10.36**	**10.71**	**10.91**	**49.19**	**59.43**	**68.00**	**33.54**	**34.92**	**35.52**

sessions to obtain a vector representing the user's general interest. When a user's behavior is aimless, or his interests drift quickly in the current session, H-RNN is ineffective to cope with noisy sessions. On the contrary, A-PGNN not only overcomes the limitation of sequence-based model but also utilizes the dot-product attention mechanism to explicitly calculate the impact of historical sessions on current session to generate a more accurate current session embedding therefore better unified user representation. We consider this kind of strategy results in 9.9% - 40.7% recall gain and 4.4% - 29.5% MRR gain over the best performing baselines.

It is also worth noting that the performance of SR-GNN is inferior to A-PGNN by large margins on Reddit (up to 40.7% worse Recall and 29.5% worse MRR) but such huge difference is not showed on Xing. One possible explanation is that, compared with Xing, there is less repeativeness in Reddit which makes simple GNN-based model loses its advantages. To be specific, compared with SR-GNN, A-PGNN further considers transitions between items in user's all sessions and thereby models all sessions as a single graph, which can capture more complex and implicit connections between user clicks. Whereas the graph neural network in SR-GNN is not suitable for personalized recommendation scenarios, nor can it take advantage of the userâĂŹs personalized information.

4.6 Ablation Study (RQ2)

Next, we turn to **RQ2**. To verify the effectiveness of the Dot-Product Attention mechanism and PGNN in our model, we compare our method with different variants.

A-PGNN(-U): A-PGNN without user embedding utilized in gated graph neural network only uses the most trivial GGNN model.

A-PGNN(-H): A-PGNN without Dot-Attention mechanisms, that is, it does not consider the impact of the historical sessions on the current session.

A-PGNN(-U-H): A-PGNN has neither user representation nor Dot-Product attention mechanism, that is, the user's information is not used at all.

A-PGNN(-GNN): A-PGNN has no PGNN component, but retains the user's embedding and Dot-Product attention mechanism.

The performances on Xing and Reddit data are shown in Table 4. Compared with A-PGNN(-U-H), A-PGNN(-U) and A-PGNN(-H) utilize the user's historical information and inherent information, respectively, which makes the performance of these two models on most indicators improved. From Table 4, we have the following conclusions based on performance of the ablation model on Xing:

(1) The performance of ablation model A-PGNN(-H) and A-PGNN(-U-H) are relatively worse than others, which illustrates user's historical session sequence information need to be captured for next-item recommendation.

(2) Model A-PGNN and A-PGNN(-U) work better than model A-PGNN(-GNN), which demonstrates that the Graph Neural Network can better capture the complex transition relationship between items in the user sequence, this is also in line with our motivation of using GGNN.

(3) Although model A-PGNN and A-PGNN(-U) both can capture the transition relationships between items, model A-PGNN works better than model A-PGNN(-U), which also show that the use of user embedding makes the GGNN model more capable of capturing relationships between items in user-specific fashion. For Reddit data, we can draw similar conclusions with Xing data. But unlike Xing, the influence of historical sequence on the current session is less, which is reflected by the smaller improvement of A-PGNN than A-PGNN(-H). From the aspect of reality, Xing is an employment-oriented website, user interest drift is small, and historical interests are often reflected in the current session. Reddit is an entertainment, social and news website, which means that the purpose of the user's browsing behavior is often unclear and is susceptible to drift of interest due to the content posted on the website. Therefore, a longer period of historical sequence has less influence on the current session. In Section 4.8.2, we will elaborate on the influence of the length of user historical session sequence on the prediction results. From the results of different ablative methods, the validity of the two components in our model can be proved.

4.7 Hyper-Parameter Study (RQ3)

User historical information and PGNN layer play an pivotal role in our model. In this section, we perform experiments how the maximum historical session and number of pgnn layers influence the performance. In each of the following experiments, we only change on hyper-parameter and keep the others the same. We conduct several experiments on dataset Xing and Reddit. The performance pattern in terms of Recall@5,10,20 and Mrr@5,10,20 are similar, so we only use Recall@5 and Mrr@5 to compare the results.

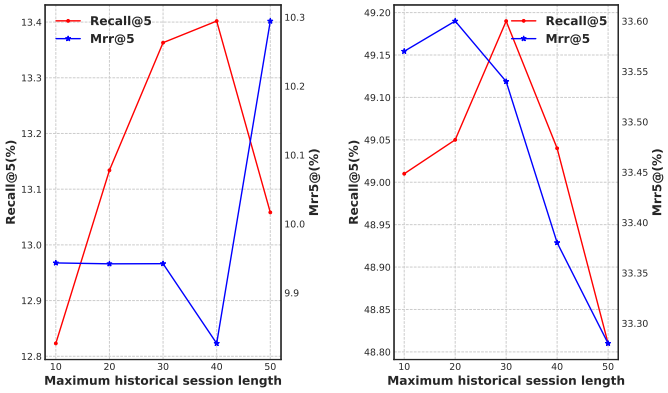
4.7.1 Influence of maximum historical session length

We first investigate how the performance change with the maximum historical session length. The maximum historical

TABLE 4
Recommendation performance comparison of ablation models on Xing data and Reddit data. The best performance on each dataset is highlighted.

Datasets	Xing						Reddit					
	Recall@5	Recall@10	Recall@20	Mrr@5	Mrr@10	Mrr@20	Recall@5	Recall@10	Recall@20	Mrr@5	Mrr@10	Mrr@20
A-PGNN(**-U-H**)	11.84	14.67	17.68	8.25	8.63	8.84	48.71	58.90	67.57	33.11	34.47	35.07
A-PGNN(**-U**)	14.20	16.69	19.44	10.29	10.62	10.81	48.97	59.24	67.77	33.37	34.75	35.35
A-PGNN(**-H**)	11.89	14.74	17.91	8.08	8.46	8.68	49.04	59.31	67.84	33.35	34.73	35.33
A-PGNN(**-GNN**)	13.84	16.67	19.73	9.67	10.05	10.26	48.77	58.75	67.25	33.46	34.79	35.39
A-PGNN	**14.38**	**17.06**	**19.98**	**10.36**	**10.71**	**10.91**	**49.20**	**59.43**	**67.96**	**33.54**	**34.92**	**35.52**

session length is an important criterion for measuring the amount of user historical information. Figure 3 shows the variation of evaluation values with the maximum historical session length M. Higher M doesn't lead to better results always. For dataset Xing, the best M is 50. For Reddit, the model reaches the highest result when M is 10. This shows that the increase in user historical information does not necessarily lead to an increase in model performance.

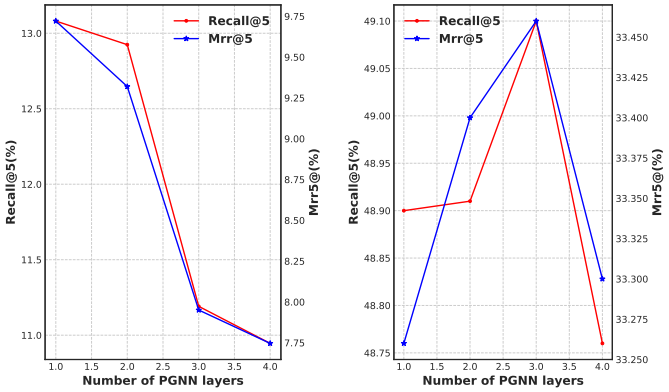


(a) Xing data (b) Reddit data

Fig. 3. The influence of maximum historical session length. As a hyperparameter, maximum historical session lengths have a significant impact on model performance. Different lengths have large differences on different data sets.

4.7.2 *Influence of PGNN layers*

PGNN is the core component of our model. Thus we are interested in the effect of the number of PGNN layers which is the propagation step T mentioned in Section 3.4. The results on datasets are shown in Figure 4. On Xing data, We can see that the performance of model reaches the highest value when T is 1, and then gradually decreases with the increase of T. As for Reddit data, the model performance increases along the increasing of T until it reaches 3, after that the performance starts to decrease. From the perspective of Graph Neural Networks, T represents the order of the central node to aggregate neighbors. It is reasonable since the dependencies between adjacent items in the sequence of xing data is very strong and $T = 1$ is enough for capture the item transition relationships. On Reddit data, the purpose of users' browsing is often not very clear, so we need to capture higher-order neighborhood relationships.



(a) Xing data (b) Reddit data

Fig. 4. The influence of PGNN layers amount. On Xing and Reddit data, one-layer and three-layer PGNN achieves the best result, respectively.

4.8 The Effects of Current and Historical Session Length on Recommendation Tasks (RQ4)

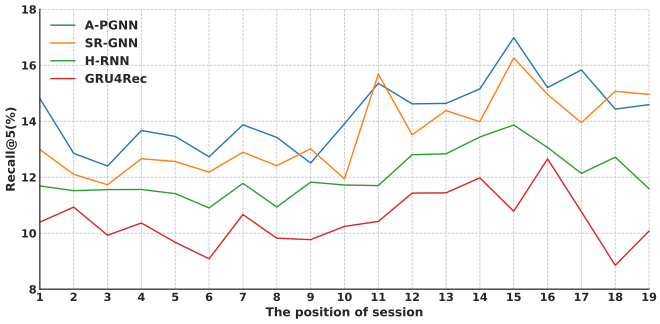
To answer **RQ4**, we further analyze the capability of model to cope with user sessions with different lengths and number of historical sessions. When evaluating, we grouped them according to the current session and the historical session lengths, respectively. Similarly, for Section 4.7, we use Recall@5 and Mrr@5 to compare the results.

4.8.1 *Analysis with user current session length*

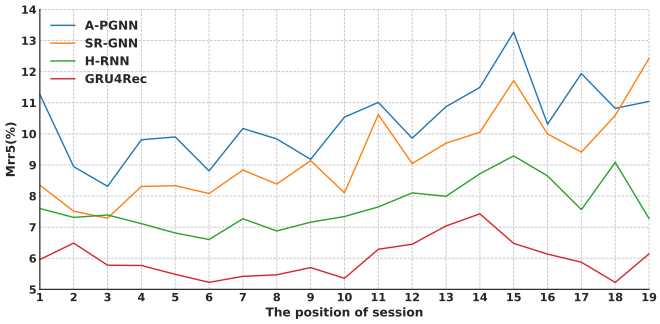
First, we analyze the capability of different models to cope with sessions of different lengths. Since the session length of Reddit data is relatively short, we only give the analysis of sessions whose length are ≥ 5 in Xing. For comparison, we evaluate recommendation performance of our A-PGNN with SR-GNN, H-RNN and GRU4Rec on every position of current sessions, respectively. Figure 5 shows the Recall@5 and Mrr@5 results. Since the maximum length of session is 20, the evaluation position ranges from 1 to 19. From the results, some interesting conclusion can be drawn:

(1) A-PGNN significantly and consistently outperforms the best baseline model almost in every position, further underpinning the superiority of A-PGNN to explicitly model the impact of historical sessions on current session by dot-attention mechanism and utilizes PGNN to model the complex transition relationships between different items in user-specific fashion.

(2) After the initial stage of the new session, the evaluation values promotion of personalized model A-PGNN begins to decrease compared with the state-of-the-art session model SR-GNN. The session-level dynamics start to prevail

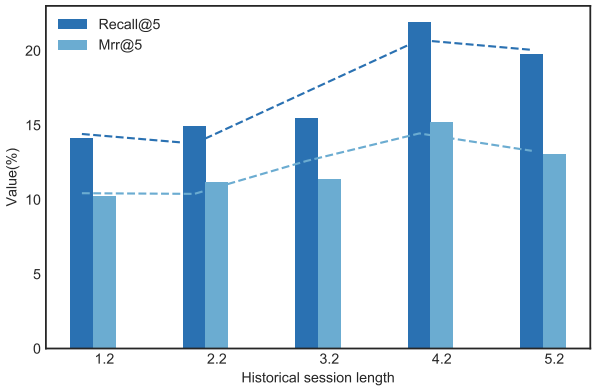


(a) Recall@5 on Xing Data

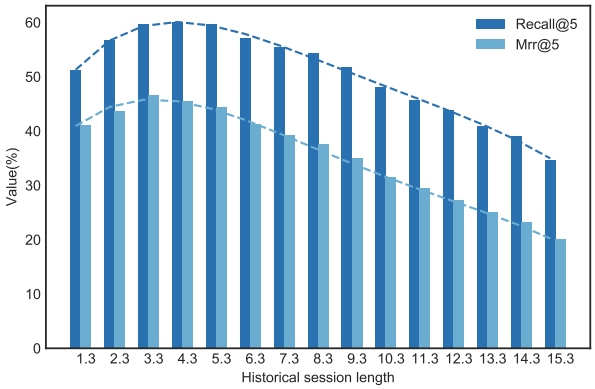


(b) Mrr@5 on Xing Data

Fig. 5. Performance comparison in terms of Recall@5 and Mrr@5 with different positions of current sessions on Xing data. As shown in Figure, our model has achieved the best results in almost all positions of the session



(a) Performance on Xing Data



(b) Performance on Reddit Data

Fig. 6. Performance with different length historical sessions on Xing data and Reddit data.

over longer-term user interest dynamics, making personalizing model less effective. However, A-PGNN model still provides superior recommendation quality within the session.

(3) A-PGNN significantly outperforms baseline models by a large margin in the initial position. This conforms with the intuition that user's past session information can be effectively used in A-PGNN to predict the first actions of a user in the forthcoming session with greater accuracy. On the contrary, GRU4Rec and SR-GNN do not incorporate any user's past session information when user starts a new session, which increases the difficulty of capturing user's preference when she/he just clicks a few items.

4.8.2 *Analysis with user historical Session length*

According to the experimental results of section 4.6, one can find that the information of historical sessions play an important role in predicting user's next click. Is it the more historical sessions we feed into PGNN (which means it has more past information to extract), the better the model will be? To answer this question, we compute the evaluation index values of our model under different historical session lengths of all users. To facilitate analysis and observation, we pack several historical sessions as a unit to feed into our model and make predictions. We regard ten historical sessions as a single unit, so the user's historical session length can be represented as follows:

$$\underbrace{S_1, \cdots, S_{10}}_{1}, \underbrace{S_{11}, \cdots, S_{20}}_{2}, \cdots, \underbrace{S_{31}, \cdots, S_{40}}_{4}, \cdots,$$

where S_i represents user's session i. As for the evaluation method under different historical session length, we take the historical length as 2 unit as an example, the evaluation method is as follows:

$$\begin{aligned} &S_1, \cdots, S_{10} \rightarrow S_{11} \\ &S_1, \cdots, S_{10}, S_{11} \rightarrow S_{12} \\ &\quad \vdots \\ &S_1, \cdots, S_{10}, S_{11}, \cdots, S_{19}, S_{20} \rightarrow S_{21} \end{aligned}$$

As formula showed above, the left side of the arrow is user's historical sessions, and the right side is the corresponding current session. Then calculate the average Recall@K and MRR@K for all current sessions. As shown in the x-axis of Figure 6, on Xing dataset, the maximum number of users' historical sessions is 50, so the maximum length of users' historical sessions is 5 units. Similarly, the number of sessions in the longest history session sequence of Reddit dataset is 150, and the maximum length of user's historical session sequence can be represented as 15 units.

As a consequence, the performances of A-PGNN under different historical unit lengths can be obtained. The results are shown in Figure 6, the change curves of evaluation metrics values along with the increase of user history session length were given. We can observe that with the increase of number of historical session units fed into A-PGNN, our model perform better and better. However,

as the number of historical sessions increases to a certain extent, the performance begins to deteriorate. The reason is that the user's interest drifts greatly with the increase of time, which means that the interest of a long period in the past may bring noise to the prediction of the current session. Furthermore, we find that the prediction of current session in Xing data depends on the historical session data over a long period of time. Compared with Xing data, Reddit data has a shorter dependence on historical sessions. The Xing results in terms of Recall@5 and Mrr@5 begin to decrease when the number of historical sessions is greater than four unit lengths. But in the Reddit data, when the length of units is greater than three, the value of MRR@5 begin to decrease. The reasons for these phenomena are related to the different characteristics of Xing and Reddit websites, as mentioned in 4.6. In summary, the length of the user's historical sessions has a significant impact on the prediction accuracy of next-item. This disproves the theory that "more is better". But we can select the appropriate length of the historical session sequence as hyper-parameter to train and test model in the actual scenarios according to the characteristic of the dataset.

5 Conclusions

In this paper, we have proposed a novel architecture Personalizing Graph Neural Networks with Attention Mechanism, abbreviated as A-PGNN for personalized recommendation scenario, which achieves the target by leveraging user information to improve recommendation performance. The proposed model not only captures the complex transition relationships between items in each user by the improved graph model personalizing graph neural networks, but also uses the Dot-Attention mechanism in Transformer [41] to explicitly model the effect of historical sessions on the current session, which make it easy to capture the user's long-term performance. Comprehensive experiments on two public datasets verify the effectiveness of different component in our model and confirm that the proposed model can consistently outperform other state-of-art models. For future work, we will further improve A-PGNN by incorporating the dynamic graph neural networks to improve the flexibility and scalability of PGNN. In addition, we are also interested in exploring more effective attention mechanisms to integrate users' long and short-term interests.

References

[1] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proceedings of the 2016 International Conference on Learning Representations*, ser. ICLR '16, 2016.

[2] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, ser. DLRS 2016. New York, NY, USA: ACM, 2016, pp. 17–22. [Online]. Available: http://doi.acm.org/10.1145/2988450.2988452

[3] T. X. Tuan and T. M. Phuong, "3d convolutional networks for session-based recommendation with content features," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys '17. New York, NY, USA: ACM, 2017, pp. 138–146. [Online]. Available: http://doi.acm.org/10.1145/3109859.3109900

[4] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 1419–1428. [Online]. Available: http://doi.acm.org/10.1145/3132847.3132926

[5] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, ser. WSDM '18. New York, NY, USA: ACM, 2018, pp. 108–116. [Online]. Available: http://doi.acm.org/10.1145/3159652.3159668

[6] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR '18. New York, NY, USA: ACM, 2018, pp. 505–514. [Online]. Available: http://doi.acm.org/10.1145/3209978.3210017

[7] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: Short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 1831–1839. [Online]. Available: http://doi.acm.org/10.1145/3219819.3219950

[8] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *the 27th International Joint Conference on Artificial Intelligence*, 2018.

[9] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys '17. New York, NY, USA: ACM, 2017, pp. 130–137. [Online]. Available: http://doi.acm.org/10.1145/3109859.3109896

[10] M. Ruocco, O. S. L. Skrede, and H. Langseth, "Inter-session modeling for session-based recommendation," pp. 24–31, 2017.

[11] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 346–353.

[12] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2007, pp. 1257–1264.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[14] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Springer, 2011, pp. 145–186.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW '01, 2001.

[16] G. Shani, R. I. Brafman, and D. Heckerman, "An mdp-based recommender system," in *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 453–460. [Online]. Available: http://dl.acm.org/citation.cfm?id=2073876.2073930

[17] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 811–820.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Annual Conference on Neural Information Processing Systems*, 2013, pp. 3111–3119.

[19] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, vol. 2, 2010, p. 3.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

[21] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, pp. 3776–3784.

[22] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *International Conference on Learning Representations*, 2015.

[23] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, "Sequential click prediction for sponsored search with recurrent neural networks," in *AAAI Conference on Artificial Intelligence*, 2014, pp. 1369–1376.

[24] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *AAAI Conference on Artificial Intelligence*, 2016, pp. 194–200.

[25] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent basket recommendation model," in *Proceedings of the 39nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2016.

[26] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: ACM, 2016, pp. 241–248. [Online]. Available: http://doi.acm.org/10.1145/2959100.2959167

[27] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys '17. New York, NY, USA: ACM, 2017, pp. 306–310. [Online]. Available: http://doi.acm.org/10.1145/3109859.3109872

[28] C. Wu and M. Yan, "Session-aware information embedding for e-commerce product recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 2379–2382. [Online]. Available: http://doi.acm.org/10.1145/3132847.3133163

[29] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 701–710. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623732

[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077. [Online]. Available: https://doi.org/10.1145/2736277.2741093

[31] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 855–864. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939754

[32] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15, 2015.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 2016 International Conference on Learning Representations*, ser. ICLR '16, 2016.

[34] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, July 2005, pp. 729–734 vol. 2.

[35] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan 2009.

[36] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *Proceedings of the 2015 International Conference on Learning Representations*, ser. ICLR '15, vol. abs/1511.05493, 2015.

[37] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[38] Z. Li, X. Ding, and T. Liu, "Constructing narrative event evolutionary graph for script event prediction," 05 2018.

[39] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation recognition with graph neural networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 4183–4192.

[40] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, July 2017, pp. 20–28. [Online]. Available: doi.ieeecomputersociety.org/10.1109/CVPR.2017.10

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[42] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *the Eleventh ACM Conference*, 2017, pp. 130–137.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.