What Can History Tell Us? Identifying Relevant Sessions for Next-Item Recommendation

Ke Sun
Wuhan University
China
sunke1995@whu.edu.cn

Tieyun Qian*
Wuhan University
China
qty@whu.edu.cn

Hongzhi Yin*
The University of Queensland
Australia
h.yin1@uq.edu.au

Tong Chen
The University of Queensland
Australia
tong.chen@uq.edu.au

Yiqi Chen
Wuhan University
China
yiqic16@whu.edu.cn

Ling Chen
University of Technology Sydney
Australia
ling.chen@uts.edu.au

ABSTRACT

Recommendation systems have been widely applied to many E-commerce and online social media platforms. Recently, sequential item recommendation, especially session-based recommendation, has aroused wide research interests. However, existing sequential recommendation approaches either ignore the historical sessions or consider all historical sessions without any distinction that whether the historical sessions are relevant or not to the current session, which motivates us to distinguish the effect of each historical session and identify relevant historical sessions for recommendation.

In light of this, we propose a novel deep learning based sequential recommender framework for session-based recommendation, which takes Nonlocal Neural Network and Recurrent Neural Network as the main building blocks. Specifically, we design a two-layer nonlocal architecture to identify historical sessions that are relevant to the current session and learn the long-term user preferences mostly from these relevant sessions. Besides, we also design a gated recurrent unit (GRU) enhanced by the nonlocal structure to learn the short-term user preferences from the current session. Finally, we propose a novel approach to integrate both long-term and short-term user preferences in a unified way to facilitate training the whole recommender model in an end-to-end manner. We conduct extensive experiments on two widely used real-world datasets, and the experimental results show that our model achieves significant improvements over the state-of-the-art methods.

KEYWORDS

Session-based Recommendation, Next-Item Recommendation, Deep Neural Networks

* Corresponding authors.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358050

ACM Reference Format:
Ke Sun, Tieyun Qian*, Hongzhi Yin*, Tong Chen, Yiqi Chen, and Ling Chen. 2019. What Can History Tell Us? Identifying Relevant Sessions for Next-Item Recommendation . In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3357384.3358050

1 INTRODUCTION

Recommendation systems have become an essential component in many E-commerce or social networking websites. There are two main types of recommendation systems: general recommenders and sequential recommenders. General recommenders like MF [22] and BPR [29] focus on modelling static user-item interactions. For instance, when a user living in *NewYork* travels to *Hawaii* for a holiday, general recommenders may still recommend points of interest (POIs) located in *NewYork* since they are unable to capture the dynamics of user preferences [39, 44]. As user's interactions with items can be viewed as a long sequence, sequential recommenders (e.g., Fossil [12], FPMC [30], CASER [34], RUM [5]) are proposed to exploit the sequential patterns or sequential dependencies of user-item interactions. The sequential recommenders can be further divided into two categories: traditional Markov chain based approaches (e.g, Fossil [12], FPMC [30]) and deep learning based approaches (e.g., CASER [34], RUM [5]). The sequence can be further partitioned into sessions where each session contains a set of interactions that occur within a given time window. Recently, session-based recommendation systems (SRS), as a subtask of sequential recommenders, have emerged and attracted much attention from both academia and industry [16, 23, 24, 26, 43, 45].

The key task in SRS is to predict the next item for a given user based on the user's current ongoing session. Due to the nature of SRS, i.e., assuming strong correlations among items within a session, most existing approaches on SRS, such as GRU4Rec [16], NARM [23], ATEM [38] and STAMP [26], only consider the current session and treat it as a short sequence. They mainly adopt recurrent neural network (RNN) or its variants and attention mechanism to characterize short-term user preferences. An obvious drawback of these methods is that they ignore long-term user preferences, i.e., the effect of historical sessions on the current session. On the other hand, some more recent works [24, 28, 45] try to exploit long-term user preferences to improve the performance of sequential

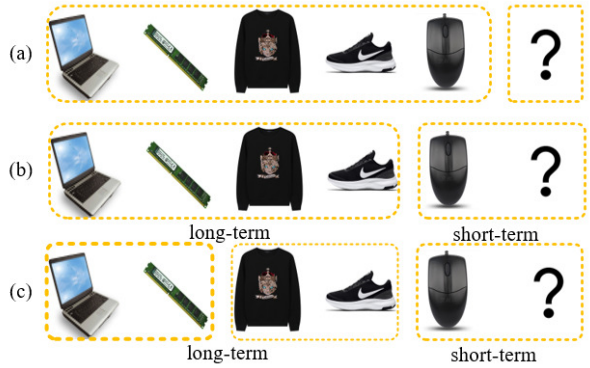


Figure 1: The difference between our model and the existing two types of methods. (a): Methods that do not distinguish long-term and short-term user preferences. (b): Methods that distinguish long-term and short-term user preferences, but do not identify relevant long-term user preferences. (c): Our method not only distinguishes long-term and short-term user preferences, but also identifies long-term user preferences relevant to the current session.

recommender systems. For example, SHAN [45] employs attention mechanism to learn an item weight vector for each user to represent personal long-term interests. BINN [24] models long-term user preferences by applying bi-directional LSTMs (Bi-LSTMs) to the whole historical interaction sequences. However, all these methods either assume all historical sessions are equally important without considering what users really need in the current session, or do not consider the session-based setting, i.e., they do not partition the long sequence into sessions at all. In light of this, we propose a novel deep learning-based approach to exploit and integrate both short-term user preferences and relevant long-term user preferences for session-based recommendation. Figure 1 summarizes the difference between our model and the existing two types of methods.

In our proposed session-based recommender, there are three deep neural network based components to learn relevant long-term user preferences from historical sessions, learn short-term user preferences from the current session, and fuse these two types of user preferences, respectively. Specifically, we design a two-layer nonlocal architecture [41] to learn long-term user preferences from relevant historical sessions, inspired by the observation that sessions in the sequence are correlated, and different historical sessions have distinct effect on the short-term user preferences in the current session. For example, assuming a user's goal in the current session is mainly on purchasing computer accessories, it would be more helpful to exploit the user's preferences on computer brands from her/his previous relevant sessions, e.g., she/he prefers MacBook to Dell. In such a case, this user's sessions on purchasing clothes could be neglected. In addition to the nonlocal architecture for finding relevant historical sessions, we also integrate the nonlocal structure with a gated recurrent unit (GRU) to learn short-term user preferences with subtle sequential and non-consecutive patterns from the current session. Finally, we propose a novel approach to integrate both long-term and short-term user preferences in a unified way to facilitate training the whole recommender model in an end-to-end manner. We conduct extensive experiments on two publicly available real-world recommendation datasets, and the experimental results show that our model achieves significant improvements over the state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, we review the related work. In Section 3, we present our proposed model. In Section 4, we show the experimental evaluation. We conclude the paper in Section 5.

2 RELATED WORK

In this session, we summarize the related research background, including the conventional recommendation systems, deep learning-based recommendation systems, and session-based recommendation systems.

2.1 Conventional Recommendation Systems

Conventional recommendation systems can be categorized into two types: general recommenders and sequential recommenders.

The general recommenders, represented by the classic collaborative filtering (CF) technique [15], aims to explore the entire purchase history of users to build their general static interests. Specifically, matrix factorization [22] is the most popular method along this line due to its strong power of reconstructing the user-item matrix. Following the success of MF, many optimized methods such as BPR [29], SVD [8], WRMF [31], SVD++ [21], eALS [14] have been introduced into this field. The general recommenders mainly focus on the stability of users' preferences and ignore the dynamics and evolutions of users' interests.

The second type leverages users' historical records in a sequential manner to model the dynamics and evolutions of users' preferences. The main task is to predict the next item. Instead of using a common matrix in MF, the classic sequential recommender method FPMC [30] builds a personal transition matrix for each user by integrating Markov chain into MF. Besides dynamic interests, other studies take users' general tastes into account. For example, HRM [37] builds a two-layer aggregation structure to combine users' general preferences and their sequential behavior in a nonlinear way. Similarly, SPORE [40] proposes a novel latent variable topic model to fuse sequential influence with personal interests. Fossil [12] models users' long and short-term preferences by extending Markov chain with similarity-based methods.

2.2 Deep Learning-based Recommendation Systems

Deep learning has become pervasive in recommendation systems in recent years. In general recommenders, impressive progress has been made by incorporating deep neural networks [4, 6, 9, 13, 35, 42]. NeuMF [13] models latent features of users and items in a high level of non-linearities by jointly learning a matrix factorization and a multi-layer neural network. Based on NeuMF [13], DELF [6] is proposed by constructing an additional item/user-based embedding for each item/user before neural interaction layers. LRML [35] adopts an attentive memory module to generate latent relation vectors which can improve the interpretability of users' interactions.

Various deep neural networks like recurrent neural networks (RNNs) have been applied to sequential recommenders to characterize the temporal dependency [2, 5, 19, 25, 27, 34, 46]. DREAM [46] puts users' transaction sequences into a single RNN layer and treats

the hidden states as the current preferences. CA-RNN [25] imports the adaptive context-specific inputs and transition matrices to model current contextual information. RUM [5] and KSR [19] maintain an external memory matrix for each user to store historical hidden states by reading and updating the matrix. CASER [34] adopts a convolutional neural network (CNN) to capture union-level and point-level patterns, as well as the skip behaviors.

2.3 Session-Based Recommendation Systems

Session-based recommendation [10, 16, 28] is one kind of emerging recommenders. Considering a session as a short sequence, the deep neural networks used in sequential recommenders can also be employed in session-based recommenders. For example, GRU4Rec [16] makes use of GRU (a variant of RNN) to model sequential patterns within a short time frame. NARM [23], ATEM [38] and STAMP [26] adopt attention mechanisms to emphasize the main intentions in current session. Following this line of work, a number of optimized methods have been introduced. For example, the basic GRU4Rec [16] method is later improved by incorporating side information such as dwell time [3] and items' features [17]. Techniques including data augmentation, model pre-training, and distillation [32] are introduced to enhance the RNN based model. A significant improvement is made in [20] by using a weighted combination of GRU4Rec [16] and kNN methods.

The aforementioned models ignore users' long-term preferences. To tackle this problem, several methods consider both short and long-term interests. For example, HRNN [28] adds an additional user-GRU based on GRU4Rec [16] to propagate information across sessions, thus tracking the evolution of users' long-term interests. SHAN [45] models users' evolving general tastes with a hierarchical attention network, while BINN [24] applies a Bi-LSTM on users' entire historical interaction sequences to generate vector representations of static preferences.

While HRNN [28], SHAN [45], and BINN [24] show improvements over previous methods, they neglect the relationship between users' current session and previous sessions when extracting general interests from historical interactions. In contrast, our proposed model can identify the relevant sessions from the history. Furthermore, our model well exploits both long- and short-term preferences to improve the performance of next-item recommendation.

3 THE PROPOSED MODEL

In this section, we first formulate the session-based next-item recommendation problem, and then introduce the details of our model.

3.1 Preliminaries

3.1.1 Problem Definition. Let U and I be the user and item set, respectively. Each user $u \in U$ has a session sequence denoted as $S = \{S_1, S_2, ..., S_n\}$, where n is the index for the current session. For $m = 1, 2, ..., n$, each session $S_m \in S$ consists of a sequence of items $i \in I$ clicked or visited by the user, i.e., $S_m = \{i_1, i_2, ..., i_{|S_m|}\}$. The session-based next-item recommendation problem is defined as: for a target user $u \in U$, within the user's current session $S_n = \{i_1, i_2, ..., i_{t-1}\}$ where i_{t-1} is the most recent item u has interacted with, given a historical session sequence $\{S_1, S_2, ..., S_{n-1}\}$ of u, the task is to predict the next item $i_t \in I$ that u is most likely to access at the next time step t in the present session S_n.

3.1.2 Nonlocal Neural Network. The basic idea of nonlocal operation [41] is to represent each position of the input signal (image, sequence, or video) by a weighted sum of the features at all positions such that long-range dependencies can be taken into account. Taking a real-valued feature sequence $X = \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{|X|}\}$ as an example, each position $\mathbf{x}_i \in X$ of the sequence is encoded as $\mathbf{y}_i \in Y$ ($|X| = |Y|$) by:

$$\mathbf{y}_i = \frac{1}{C(X)} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j), \tag{1}$$

where $C(X) = \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j)$ is the normalization factor, and j is the index that enumerates all positions of the input $\mathbf{x}$, and $g(\cdot)$ computes the latent representation for $\mathbf{x}_j$. The pairwise function $f(\cdot)$ is used to calculate a scalar (representing the similarity) between $\mathbf{x}_i$ and $\mathbf{x}_j$. We will detail our design of $g(\cdot)$ and $f(\cdot)$ for session-based recommendation in the following sections.

3.2 Model Overview

In this paper, we propose a novel sequential recommender for session-based next-item recommendation by considering both users' short-term and long-term preferences. The architecture of our model is shown in Figure 2. Our model contains three key components: long-term preference modelling, short-term preference modelling, and personalized representation integration. Specifically, to capture users' long-term preferences, we design a two-layer nonlocal neural network which learns representations from historical sessions relevant to the current session and the most recently visited item. To model users' short-term preferences, we present a hybrid structure containing a GRU and a nonlocal neural network to model users' present intentions from the current session. Finally, for each user, we integrate the user's long- and short-term preference representations in a personalized way based on her/his interaction history to estimate the next item to recommend.

3.3 Long-Term Preference Modelling

A user's click or check-in history contains rich information about the user's interests, and each session can be regarded as a short clip of the user's long-term interaction history. When recommending an item within a certain session, though the limited information from a single session can hardly indicate a user's long-term preferences, the historical sessions related to the present session can offer substantial signals that lead to the user's final decision. Hence, a smart recommender should be capable of learning to selectively gather the most useful information from the history to infer each user's long-term preferences.

To this end, we propose a two-layer nonlocal neural network to lay more emphasis on the influence of relevant sessions and weaken that of irrelevant ones. As demonstrated in Figure 2, for a session sequence of length n, the first layer learns a session-level long-term representation $\mathbf{s}_n^*$ indicating a user's general preference. Then, by leveraging the information of the most recent item visited by the user, the second layer further refines $\mathbf{s}_n^*$ into $\mathbf{s}_n^+$, which is the fine-grained long-term representation containing both item-level and session-level information.

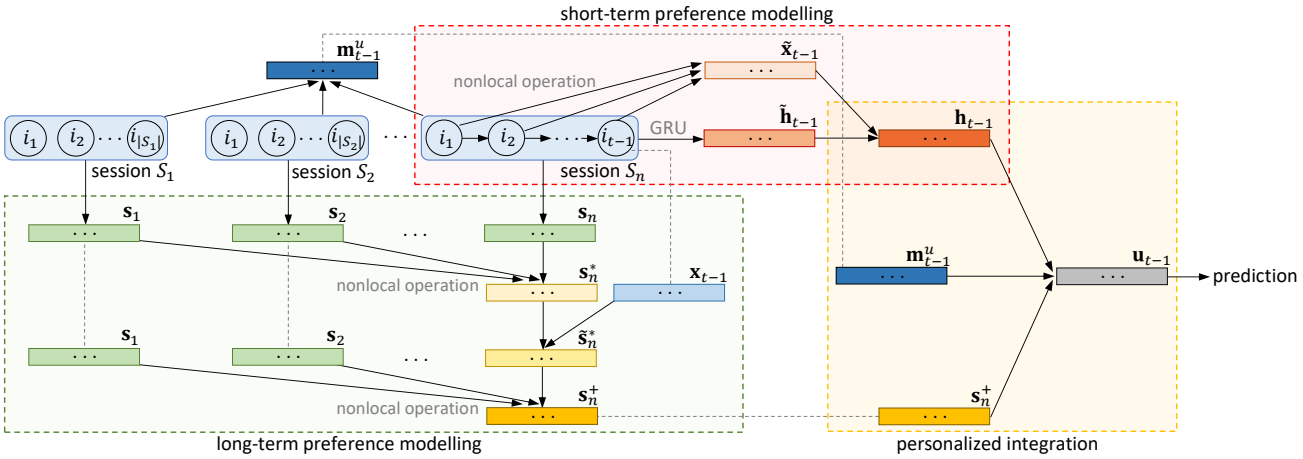


Figure 2: The architecture of our model.

First, to model the long-term preferences concealed in the session sequence, for each user, our two-layer nonlocal neural network takes a sequence of vectors $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n\}$ representing n different sessions as inputs. For session $S_m \in S$ containing $|S_m|$ items, we calculate its session-level representation $\mathbf{s}_m$ using the average pooling as follows:

$$\mathbf{s}_m = \frac{1}{|S_m|} \sum_{t=1}^{|S_m|} \mathbf{x}_t, \tag{2}$$

where $\mathbf{x}_t \in \mathbb{R}^{d\times 1}$ is the d-dimensional embedding vector for the t-th item in session S_m. All embeddings will be randomly initialized and then trained in the network. By applying average pooling on item representations, we can now represent the user's current session as $\mathbf{s}_n$ while the historical session sequence as $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{n-1}\}$. Note that we represent session $S_m \in S$ with one synergic vector $\mathbf{s}_m$ instead of all the items' vector representations in this session. This is because $\mathbf{s}_m$ encodes the second-order interactions between item features in the embedding space, thus being more representative of a user's session-level interest within a particular session.

Then, to derive the long-term user preference $\mathbf{s}_n^* \in \mathbb{R}^{d\times 1}$ from relevant historical sessions, we perform the nonlocal operation in Eq. 1 on all session representations. The intuition is that identifying the connection between a user's past session-level preferences and the current one can help the model better summarize the user's preference in the long run. In other words, the representation of the current session is a weighted average over all historical sessions, and we reformulate Eq. 1 as the following:

$$\mathbf{s}_n^* = \frac{1}{\sum_{j=1}^{n-1} f(\mathbf{s}_n, \mathbf{s}_j)} \sum_{j=1}^{n-1} f(\mathbf{s}_n, \mathbf{s}_j) g(\mathbf{s}_j), \tag{3}$$

where $\mathbf{s}_j \in \mathbb{R}^{d\times 1}$ denotes the representation of a historical session S_j, and $\frac{1}{\sum_{j=1}^{n-1} f(\mathbf{s}_n, \mathbf{s}_j)}$ is the normalization term.

The design of $g(\cdot)$ and $f(\cdot)$ is task-oriented. In our model, we respectively define the similarity function $f(\cdot)$ and the session feature generator $g(\cdot)$ as:

$$f(\mathbf{s}_n, \mathbf{s}_j) = e^{(\mathbf{W}_\theta \mathbf{s}_j)^T \mathbf{W}_\phi \mathbf{s}_n}, \tag{4}$$

$$g(\mathbf{s}_j) = \mathbf{W}_g \mathbf{s}_j, \tag{5}$$

where $\mathbf{W}_\theta \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_\phi \in \mathbb{R}^{d\times d}$ are weight matrices to be learned, while the learnable weight matrix $\mathbf{W}_g \in \mathbb{R}^{d\times d}$ projects $\mathbf{s}_j$ into a new embedding space. The rationale behind Eq. 4 is twofold. On one hand, for a given n, each $\frac{1}{\sum_{j=1}^{n-1} f(\mathbf{s}_n, \mathbf{s}_j)} f(\mathbf{s}_n, \mathbf{s}_j)$ is equal to the widely used $softmax$ function along dimension j which supports the efficient computation of derivatives. On the other hand, with $j \le n-1$, $softmax((\mathbf{W}_\theta \mathbf{s}_j)^T \mathbf{W}_\phi \mathbf{s}_n)$ serves the same purpose as the self-attention in [36], allowing the model to attend to information from different representation subspaces in different sessions, and assign more weights to the relevant sessions.

Besides the long-term preferences $\mathbf{s}_n^*$ learned from session-level representations, we argue that the target user's most recent check-in item tend to have the strongest impact on the user's next activity. For instance, in the scenario of POI recommendation, a user is likely to visit a POI in Hawaii given that her/his latest check-in is there. Inspired by the success of deep memory network [33], we design the second layer to look back and reconsider the historical sessions from both item and session perspectives in order to generate a more comprehensive representation $\mathbf{s}_n^+ \in \mathbb{R}^{d\times 1}$. So, similar to the computation in Eq. 3, we derive the fine-grained, long-term preference representation $\mathbf{s}_n^+$ using another nonlocal operation:

$$\mathbf{s}_n^+ = \frac{1}{\sum_{j=1}^{n-1} f'(\tilde{\mathbf{s}}_n^*, \mathbf{s}_j)} \sum_{j=1}^{n-1} f'(\tilde{\mathbf{s}}_n^*, \mathbf{s}_j) g'(\mathbf{s}_j), \tag{6}$$

where $\tilde{\mathbf{s}}_n^*$ is the element-wise sum of $\mathbf{s}_n^*$ and the embedding of the last check-in item $\mathbf{x}_{t-1}$:

$$\tilde{\mathbf{s}}_n^* = \mathbf{s}_n^* + \mathbf{x}_{t-1}, \tag{7}$$

and:

$$f'(\tilde{\mathbf{s}}_n^*, \mathbf{s}_j) = e^{(\mathbf{W}_{\theta'} \mathbf{s}_j)^T \mathbf{W}_{\phi'} \tilde{\mathbf{s}}_n^*}, \tag{8}$$

$$g'(\mathbf{s}_j) = \mathbf{W}_{g'} \mathbf{s}_j \tag{9}$$

where $\mathbf{W}_{g'}, \mathbf{W}_{\theta'}, \mathbf{W}_{\phi'} \in \mathbb{R}^{d\times d}$ are trainable weight matrices. Clearly, the final output $\mathbf{s}_n^+$ denotes the user's long-term preference representation which not only captures historical sessions' relation with the current session, but also uncovers how relevant each previous session is to the last visited item.

3.4 Short-Term Preference Modelling

Due to the power of modelling sequential patterns, RNN structures like GRU and LSTM have been widely applied to sequential recommendation systems. However, the drawback of most RNNs is that they can only model local sequential patterns from short and consecutive input segments. Furthermore, in session-based recommendation, users might buy relevant items in a non-consecutive manner. For example, although a user's main purpose in a session is making a cake, she/he might buy flour at the first place, and then turn to other groceries before purchasing butter and sugar.

As a result, to extract users' short-term preference from the current session, we propose a unified network which can learn both local and nonlocal patterns. The short-term preference modelling is carried out via a RNN and a nonlocal neural network in parallel. Given a sequence of item embeddings $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1}\}$ of the current session S_n, we firstly leverage the advantage of learning sequential representations from RNN. A gated recurrent unit (GRU) network architecture is adopted to locally model the sequential item features:

$$\tilde{\mathbf{h}}_{t-1} = GRU(\mathbf{x}_{t-1}, \tilde{\mathbf{h}}_{t-2}), \tag{10}$$

where $GRU(\cdot)$ is the GRU network, and $\tilde{\mathbf{h}}_{t-1} \in \mathbb{R}^{d\times 1}$ is the hidden state corresponding to the input at time $t-1$. Due to the tendency of RNNs to better represent recent inputs [1], the hidden state $\tilde{\mathbf{h}}_{t-1}$ will be focused on the inputs close to $\mathbf{x}_{t-1}$, thus being a local feature of user preference at time $t-1$. Note that the choice of RNNs can be diverse (e.g., LSTM is also applicable), but we choose GRU in our paper because a GRU network is easier to train as it has higher learning efficiency and less model parameters [3, 16] compared with LSTM.

At the same time, with the item embedding sequence of the current session $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1}\}$, we derive a nonlocal representation $\tilde{\mathbf{x}}_{t-1}$ of the last item in session S_n. This is achieved by aggregating information from relevant items in the same session via the following nonlocal neural network for short-term preference modelling:

$$\tilde{\mathbf{x}}_{t-1} = \frac{1}{\sum_{j=1}^{t-2} f_x(\mathbf{x}_{t-1}, \mathbf{x}_j)} \sum_{j=1}^{t-2} f_x(\mathbf{x}_{t-1}, \mathbf{x}_j) g_x(\mathbf{x}_j), \tag{11}$$

$$f_x(\mathbf{x}_{t-1}, \mathbf{x}_j) = e^{(\mathbf{W}_{\theta_x}\mathbf{x}_j)^T \mathbf{W}_{\phi_x}\mathbf{x}_{t-1}}, \tag{12}$$

$$g_x(\mathbf{x}_j) = \mathbf{W}_{g_x}\mathbf{x}_j, \tag{13}$$

with weight matrices $\mathbf{W}_{g_x}, \mathbf{W}_{\theta_x}, \mathbf{W}_{\phi_x} \in \mathbb{R}^{d\times d}$ to learn. The representation of the last item $\tilde{\mathbf{x}}_{t-1}$ is a weighted average of all its previous items based on their relevance in the same session, hence it is a nonlocal summarization of a user's short-term preference in the current session S_n. At last, we formulate the final short-term preference representation $\mathbf{h}_{t-1}$ as the element-wise aggregation of both the nonlocal and local preference vectors, $\tilde{\mathbf{x}}_{t-1}$ and $\tilde{\mathbf{h}}_{t-1}$:

$$\mathbf{h}_{t-1} = \tilde{\mathbf{x}}_{t-1} + \tilde{\mathbf{h}}_{t-1}. \tag{14}$$

3.5 Personalized Integration and Prediction

3.5.1 Personalized Integration. Different users have varied interests on different items, and such diversity becomes even stronger when users tend to act differently at different stages (i.e., sessions). Hence, instead of treating every user's preference uniformly, we integrate the long-term and short-term preferences in a personalized way for the final prediction. First, for each user, we collect all the interacted items from her/his historical sessions $\{S_1, S_2, ..., S_{n-1}\}$, denoted by $I^h = \{\{i|i \in S_1\}, \{i|i \in S_2\}, ..., \{i|i \in S_{n-1}\}\}$. Then, we represent these items with their embedding vectors, i.e., $I^h = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{|I^h|}\}$. Similarly, all $t-1$ items in the current session are represented as $I^c = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1}\}$. Based on the features of interacted items I^h and I^c, we generate this user's historical representation $\mathbf{m}^h_{t-1} \in \mathbb{R}^{d\times 1}$ and current representation $\mathbf{m}^c_{t-1} \in \mathbb{R}^{d\times 1}$ at time $t-1$ via average pooling:

$$\mathbf{m}^h_{t-1} = \frac{1}{|I^h|} \sum_{j=1}^{|I^h|} \mathbf{x}_j, \tag{15}$$

$$\mathbf{m}^c_{t-1} = \frac{1}{|I^c|} \sum_{j=1}^{|I^c|} \mathbf{x}_j. \tag{16}$$

Next, we linearly combine the user's historical and current representation to obtain this user's overall representation $\mathbf{m}^u_{t-1} \in \mathbb{R}^{d\times 1}$:

$$\mathbf{m}^u_{t-1} = \lambda \mathbf{m}^h_{t-1} + (1-\lambda)\mathbf{m}^c_{t-1}, \tag{17}$$

where λ is a hyper-parameter used to balance the impacts of historical and current representations. Here, $\mathbf{m}^u_{t-1}$ carries the user's general item-level demand extracted from the items in both historical and current sessions. Finally, for user u, we calculate the personalized user preference $\mathbf{u}_{t-1} \in \mathbb{R}^{d\times 1}$ by fusing $\mathbf{m}^u_{t-1}$ with both long- and short-term preference representations:

$$\mathbf{u}_{t-1} = \frac{1}{2}\Big(\mathbf{W}_h(\mathbf{m}^u_{t-1} \oplus \mathbf{s}^+_n) + \mathbf{W}_c(\mathbf{m}^u_{t-1} \oplus \mathbf{h}_{t-1})\Big), \tag{18}$$

where $\oplus$ is the concatenation of two vectors, $\mathbf{W}_h \in \mathbb{R}^{d\times 2d}$ and $\mathbf{W}_c \in \mathbb{R}^{d\times 2d}$ are the trainable weight matrices. Intuitively, $\mathbf{u}_{t-1}$ represents the user's current personal interest by looking at her/his long- and short-term preferences as well as each individual item she/he has ever interacted with.

3.5.2 Prediction. In our paper, we treat the next-item recommendation as a classification task with the same problem setting as [23, 26]. With the user's representation $\mathbf{u}_{t-1}$, a $softmax$ layer is employed to calculate vector $\mathbf{p}$, a $|I|$-dimensional probability distribution over all items at time t:

$$\mathbf{p} = softmax(\mathbf{W}_i \mathbf{u}_{t-1}) \tag{19}$$

where $\mathbf{W}_i \in \mathbb{R}^{|I|\times d}$ is the weight for all items. A larger $p_k \in \mathbf{p}$ means user u is more likely to interact with item $i_k \in I$ at time t.

For model training, we adopt the cross-entropy loss over all training samples. Because the actual probability is 1 for ground truth item and 0 otherwise, the loss function can be simplified as:

$$\mathcal{L} = -\sum_{l}^{L} log(p_{g,l}), \tag{20}$$

where l and L are respectively the index and total number of training samples, g is the index of ground truth item for the l-th training sample. As an end-to-end model, stochastic Gradient Descent (SGD) based algorithms can be easily applied to optimize its parameters.

4 EXPERIMENT

In this section, we conduct extensive experiments on two real-world datasets to evaluate the performance of our model.

Table 1: Statistics of datasets in use.

Statistic	Tmall	Gowalla
#user	36,595	14,898
#item	26,576	15,291
avg. session length	2.903	3.267
#train session	105,560	166,683
#test session	35,613	14,892

4.1 Evaluation Datasets

We choose two public datasets, namely Tmall and Gowalla. Both of them are widely used in previous works [18, 34, 45].

- **Tmall:** This is an E-commerce dataset collected from the largest online shopping platform *www.tmall.com* in China. It provides more than 50 million interactions of 424,170 users on 1,090,390 items within six months. Additional information including action type and session id is also released. There are four kinds of activities: click, collect, add-to-cart and purchase. Following the settings in [18, 34, 45], we only use the purchase activities in our experiments.
- **Gowalla:** This is a check-in dataset [7] collected from the location-based social network Gowalla from February 2009 to October 2010. The total number of check-ins in this dataset is 6,442,890. Every record in the data consists of user id, timestamp, GPS location and POI id.

We preprocess the datasets following the criteria in [18, 45]. First, we filter out items which have been observed by less than 20 users. Then, we treat each user's transactions or check-ins in one day as a session and remove sessions having less than two interactions. Similar to [11, 19], each user's most recent session is held out for validation and test, while the historical ones are for training. In each most recent session, the last item is selected as the ground truth for test, and the second last item is used for validation. Table 1 lists the major statistics of both datasets after preprocessing.

4.2 Baseline Methods

We compare our method with the following baselines for evaluation:

- **POP:** This is a naive baseline that ranks items for recommendation according to their occurrence frequency.
- **Fossil [12]:** This method models users' long- and short-term preferences by fusing Markov chains with similarity methods to make personalized sequential recommendation.
- **GRU4Rec [16]:** This is the first session-based recommendation method. It views each session as a short sequence and employs GRU to capture users' short-term preferences.
- **NARM [23]:** This is a session-based recommendation method which introduces an attention mechanism to model the user's main purpose in the current session. Note that this method does not take user's long-term preference into consideration.
- **STAMP [26]:** This method proposes to make use of user's short-term and long-term preferences. Different from ours, STAMP extracts those two preferences from the same current session while we model them from the user's whole history.
- **HRNN [28]:** This method applies a hierarchical RNN for personalized cross-session recommendation based on GRU4Rec [16]. It designs an additional user-GRU to propagate information from the previous user session to the next one.
- **SHAN [45]:** This is the state-of-the-art method in session-based next-item recommendation. It combines users' long-term representation learned from past items with embeddings of current items to form a hybrid user representation.
- **BINN [24]:** This is the other state-of-the-art approach. It applies a Bi-LSTM structure to the whole historical interaction sequence and generates the unified long-term preference representation with the output hidden states.

Furthermore, in order to verify the performance gain from each key component of our proposed model, we further implement three variants of our model by removing one component each time:

- **Remove-LT:** We remove the two-layer nonlocal neural network module for long-term preference modelling.
- **Remove-ST:** We remove the parallel GRU and nonlocal neural network module for short-term preference modelling.
- **Remove-PI:** We remove the personalized integration module and simply average long- and short-term preference vectors to generate the final user preference representation.

4.3 Evaluation Metrics and Experimental Setup

4.3.1 Evaluation Metrics. To evaluate the recommendation performance of all models, we employ two widely-used evaluation metrics, namely *Recall@N* and *MRR*.

- **Recall@K:** This metric [24, 45] measures the proportion of cases where the ground truth items have been correctly ranked in top-K items in all test cases. It is defined as:

$$Recall@K = \frac{\#_{hit}}{N_{test}}, \tag{21}$$

  where N_{test} and $\#_{hit}$ respectively denotes the size of test set and the number of cases where the desired items appear in top-K ranking lists. It is worth mentioning that we adopt $K = 10$ and $K = 20$ which are the widely-used settings.
- **MRR:** This metric [24, 26] measures the mean reciprocal rank of the ground truth item. It is defined as:

$$MRR = \frac{1}{N_{test}} \sum_{n'=1}^{N_{test}} \frac{1}{Rank(i_{g,n'})}, \tag{22}$$

  where $Rank(i_{g,n'})$ is the rank of the ground truth item i_g in the n'-th test session.

Note that both metrics are in the range of $[0, 1]$, and a higher value indicates better performance.

4.3.2 Experimental Setup. For a fair comparison, we follow the reported optimal parameters and training settings to achieve the baselines' best performance. For our model, we adopt grid search to select the optimal hyperparameters. Specifically, the dimension d is set to 256 for all hidden states and embeddings, and λ is respectively set to 0.1 and 0.7 for Tmall and Gowalla dataset. To train our model, we use the learning rate of 0.001. Note that we iterate our model and all the comparison methods for 15 training epochs which ensures that all model losses can converge. Note that all the reported differences between our model and others are statistically significant ($p < 0.01$).

Table 2: Performance on session-based next-item recommendation. The best result in each column is marked in boldface wile the second best is underlined.

Method	Tmall			Gowalla		
	Rec@10	Rec@20	MRR	Rec@10	Rec@20	MRR
POP	0.0213	0.0339	0.0085	0.0424	0.0638	0.0155
Fossil	0.1251	0.1523	0.0647	0.0894	0.1189	0.0342
HRNN	0.5200	0.5520	0.3973	0.1615	0.2015	0.0853
GRU4Rec	0.5736	0.6014	0.4601	0.3183	0.3820	0.1889
NARM	0.6117	0.6450	0.4683	0.3607	0.4441	0.2003
STAMP	0.6112	0.6442	0.4643	0.3395	0.4189	0.1805
SHAN	0.5834	0.6123	0.4409	0.3421	0.4214	0.1942
BINN	0.5312	0.5775	0.3896	0.3679	0.4549	0.2146
Ours	**0.6228**	**0.6530**	**0.4757**	**0.3954**	**0.4844**	**0.2301**

4.4 Performance Comparison

We compare the full version of our model with all baseline methods, and the results on two datasets are illustrated in Table 2. From the recommendation results, we draw the following observations:

(1) The performance of our model shows obvious superiority on Tmall dataset, especially in *Recall*@10, which indicates our model can accurately rank the ground truth item in the top 10 candidates. On Tmall dataset, it is also worth noting that NARM and STAMP outperform SHAN and BINN while only taking the short-term preference into account. Different from Gowalla dataset, Tmall is an E-commerce dataset collected from an online shopping platform, which means that a user's next decision can be largely influenced by her/his last several check-in items within the same session. Thus, the user's short-term preference is more important than long-term preference on the Tmall dataset. On the contrary, NARM and STAMP perform worse than BINN on Gowalla dataset due to the lack of long-term preferences. Meanwhile, our model still outperforms all the baselines, showing the benefit of leveraging all useful information from both long- and short-term perspectives.

(2) On Gowalla dataset, our model outperforms all the baselines in terms of *Recall*@10, *Recall*@20 and *MRR* by a significant margin. Our model achieves a 7.4% and 7.2% improvement against the best competitor BINN regarding *Recall*@10 and *MRR* scores, respectively. Although both SHAN and BINN take the users' long-term preferences into account, our model's consistent advantage against SHAN and BINN clearly demonstrates the benefits from modelling long-term preferences by focusing on the most relevant historical sessions.

(3) The second best performance on Gowalla is produced by BINN which takes the user's long-term preference into consideration. One main reason is that users' check-ins within successive sessions tend to be continuous due to the geographical restriction. However, BINN experiences an obvious performance drop on Tmall. This phenomenon is probably due to the fact that the correlations between successive sessions are weak on Tmall dataset. In such circumstances, it is no longer appropriate for BINN to consider the whole historical purchase actions as a continuous sequence. In contrast, our model uncovers the subtle correlations among different sessions by modelling users' preferences in a nonlocal manner, thus being able to suit different recommendation scenarios.

(4) Though HRNN can be viewed as an extension of GRU4Rec, it shows worse recommendation performance than GRU4Rec on both datasets. Compared with HRNN, the performance of GRU4Rec is slightly more promising on two datasets, and there are mainly two reasons. On one hand, a user's short-term preferences can exert a strong influence on her/his next decisions, and GRU has the powerful ability to model short-term sequential patterns. At the same time, as HRNN simply combines the current item with personal information, it may introduce noise to the GRU's input, which weakens its capability of modelling short-term user preferences.

4.5 Impact of Different User History Lengths

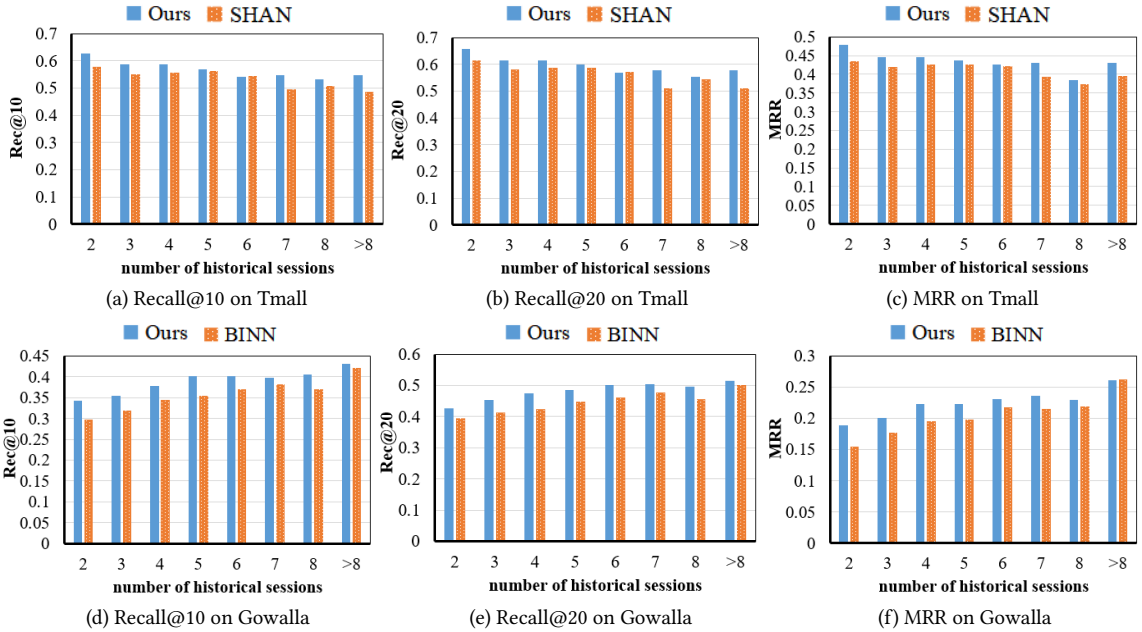
In this section, we investigate the impact of different user history lengths on the recommendation outcomes. We split the users' sessions into 8 groups based on their numbers of historical sessions. The first group contains users with history length of 2, while users having 8 or more historical sessions are in the last group. We then evaluate the performance separately with each data group as the model input. To benchmark our model's sensitivity to history lengths, we also show the results of SHAN on Tmall and BINN on Gowalla. We choose BINN on Gowalla because it is the best baseline on this dataset. Note that on Tmall, we choose the third-best SHAN instead of the best two (i.e., NARM and STAMP) because NARM and STAMP only takes the items in the current session rather than the full history as input, and these two methods do not fit the setting of this case study.

We use Figure 3 to show the results in terms of *Recall*@10 and *MRR*. Clearly, our model consistently and significantly outperforms the compared methods on both Tmall and Gowalla datasets in most cases. On Tmall dataset, both *Recall*@10 and *MRR* decrease when the historical session length is increasing. A reasonable explanation is that, for Tmall dataset, the short-term preferences have larger impacts on users' next purchase decisions than the long-term preferences. Nevertheless, the advantages of our method against SHAN confirm that extracting users' long-term preferences related to current interests has positive effects on the recommendation results.

Meanwhile, different from the Tmall dataset, both *Recall*@10 and *MRR* on Gowalla gradually increase while the historical session length is growing. This can be caused by the fact that people are more likely to visit POIs near her/his familiar activity areas, which are reflected by the long-term preferences in our case. It is also notable that on either dataset, our model shows more significant improvements when tackling short history lengths. This further demonstrates our model's effectiveness of learning the fine-grained user preferences, especially when facing the lack of sufficient transaction data for user preference modelling.

4.6 Analysis on Different Model Components

As mentioned in Section 4.2, we also verify the contribution of each proposed component for session-based recommendation. To quantify the performance gain achieved from different components, we implement three degraded versions of our model, namely Remove-LT, Remove-ST and Remove-PI for ablation tests. We show the results of ablation tests on both Tmall and Gowalla in Table 3.



(a) Recall@10 on Tmall (b) Recall@20 on Tmall (c) MRR on Tmall

(d) Recall@10 on Gowalla (e) Recall@20 on Gowalla (f) MRR on Gowalla

Figure 3: Results w.r.t. different history lengths.

Table 3: Results of ablation tests. We use 'Default' to denote the full version of our proposed model.

Dataset	Method	Rec@10	Rec@20	MRR
Tmall	Default	0.6228	0.6530	0.4757
	Remove-LT	0.5827↓	0.6099↓	0.4704↓
	Remove-ST	0.5858↓	0.6135↓	0.4711↓
	Remove-PI	0.5768↓	0.6116↓	0.4362↓
Gowalla	Default	0.3954	0.4844	0.2301
	Remove-LT	0.3618↓	0.4470↓	0.2178↓
	Remove-ST	0.3813↓	0.4666↓	0.2252↓
	Remove-PI	0.3404↓	0.4255↓	0.1802↓

4.6.1 The Long-Term Preference Modelling Module. After removing the two-layer nonlocal neural network for long-term user preference modelling, the Remove-LT model suffers from an obvious decrease on the recommendation performance on both Tmall and Gowalla. Therefore, modelling users' long-term preferences by mining the relationship between historical sessions and the current one is actually beneficial to the recommendation. It is worth mentioning that the performance decrease of Remove-LT on Gowalla (e.g., 6.4% on *Recall*@10) is more server than that on Tmall (e.g., 8.5% on *Recall*@10). This is because there are substantially more long-term interest patterns on the Gowalla dataset, which we have pointed out in Section 4.5. Hence, the long-term preference modelling module plays a pivotal role in our model.

4.6.2 The Short-Term Preference Modelling Module. Without the parallel GRU and nonlocal neural network module for short-term user preference modelling, the Remove-ST model experiences a performance drop which is similar to Remove-LT. Though Remove-ST does not affect the performance as greatly as Remove-LT, the extraction of short-term user preferences is apparently helpful for generating accurate recommendation results. We can conclude that the short-term preference modelling module offers positive contributions to the session-based recommendation.

Table 4: Results of different layers. We use 'Default' to denote the full version of our proposed model.

Dataset	Method	Rec@10	Rec@20	MRR
Tmall	Default	0.6228	0.6530	0.4757
	One-layer	0.6229↑	0.6530	0.4760↑
Gowalla	Default	0.3954	0.4844	0.2301
	One-layer	0.3901↓	0.4792↓	0.2290↓

4.6.3 The Personalized Integration Module. In particular, compared with Remove-LT and Remove-ST, removing the personalized integration module (i.e., Remove-PI) incurs the largest loss in performance. This suggests that each user has her/his own behavior features. Apart from extracting long- and short-term user preferences from the past sessions, the item-level user demand representation computed in this module can greatly help the model to lay more emphasis on specific feature spaces in Eq. 18. As a result, the personalized integration module is indispensable for our model.

4.7 Analysis on Two-Layer Nonlocal Architecture

In this section, we remove the second nonlocal operation layer from our model to investigate the impact of the last check-in item. Results on two datasets are shown in Table 4.

On the Tmall dataset, the one-layer model performs slightly better in terms of *Recall*@10 and *MRR* scores, but the results are almost identical due to the marginal difference. This phenomenon indicates that on an E-commerce dataset like Tmall, users' check-in records in the same session only carry very weak sequential dependencies. Due to the randomness and variety in online shopping scenarios, a user's next decision depends not only on the previous record but also earlier records within the same session. This observation further proves that the nonlocal structure is capable of modelling non-consecutive dependencies for recommendation. As for Gowalla dataset, the performance of the one-layer variant decreases in all three metrics. Different from the Tmall dataset, for

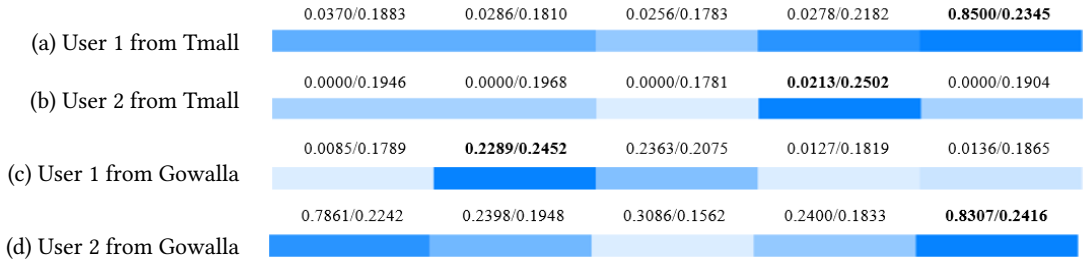
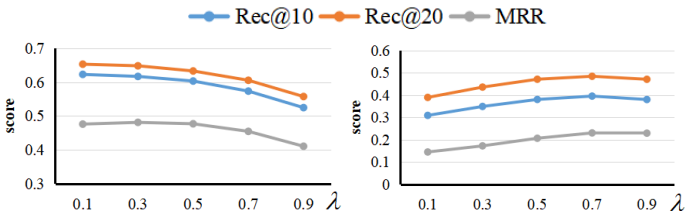


Figure 4: Session relevance visualization. The scores above each historical session denote the Jaccard score/attention weights. Note that darker colors represent higher weights learned by the model.



(a) Tmall (b) Gowalla

Figure 5: The impact of λ on two datasets.

POI platform users, people are more likely to choose POIs that are close to their current check-in POIs as the next spots to visit. Hence, with the second nonlocal layer, the last check-in items can play an important role in our model.

4.8 Analysis on Hyperparameter λ

In this section, we study the model's sensitivity on hyper-parameter λ which is used to balance the target user's two representations for historical and current context in our model. We tune λ in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ on both Tmall and Gowalla datasets, and corresponding results are shown in Figure 5.

On Tmall dataset, the recommendation performance drops when a larger λ is selected. However, the model performance gradually increases when the value of λ grows. This observation is also consistent with our conclusions in Section 4.5 that the short-term preferences of Tmall users are the dominant factor while long-term preferences of Gowalla users play a major role. Specifically, our model achieves the best performance with $\lambda = 0.1$ on Tmall and $\lambda = 0.7$ on Gowalla. This infers that on Tmall dataset, the current context plays a more important role than that for historical ones. At the same time, on Gowalla dataset, a large λ is required to obtain the optimal performance, which is also determined by the characteristics of the dataset where the long-term preferences often drive users' choices.

4.9 Analysis on Importance of Different Sessions

In our model, the two-layer nonlocal neural network can identify relevant historical sessions based on their strength of relationship with the current session. To illustrate the importance of different sessions, we randomly select two test users from each dataset, and visualize the learned weights of their last five sessions before the test session. Since it is hard to directly evaluate the association between the historical session and the test session due to the absence of detailed item/session information such as item name or category, we use the Jaccard coefficient to measure the similarity between the past session S_{hist} and test session S_{test}, i.e.,

$$Score(S_{hist}, S_{test}) = \frac{|U_{S_{hist}} \cap U_{S_{test}}|}{|U_{S_{hist}} \cup U_{S_{test}}|} \tag{23}$$

where U_{S_m} denotes the set of users who have visited at least one item in session S_m. The intuition of computing the session-wise similarity with Jaccard coefficient is that items purchased by the same user are more similar to each other.

In Figure 4, we visualize the empirical Jaccard session similarity as well as these sessions' weights assigned by our two-layer nonlocal neural network. From these four case studies, we can find some insights listed below:

(1) The nearest sessions might not be the most relevant ones (e.g., Figure 4.(b) and Figure 4.(c)), which supports our hypothesis that local sequential patterns may not be useful for modelling users' current interests. Instead, it is more reasonable to extract users' long-term preferences in a nonlocal manner based on session-wise relevance, and the benefits from the nonlocal neural network are also demonstrated via the effectiveness evaluation.

(2) Our method is capable of highlighting the relevant sessions (shown in deep colors in Figure 4), regardless of their distances from the most recent session. For example, the second session in Figure 4.(c) has the highest similarity with respect to the test session in terms of the empirical Jacarrd score. At the same time, our method also assigns the largest weight to this session, which demonstrates our model's capability of distinguishing important sessions for recommendation.

(3) The Jacarrd scores for less relevant sessions might be 0 (e.g., the first three sessions in Figure 4.(b)) as it only measures the first-order similarity without considering the high-order similarity between two sessions. In contrast, our method will still assign a small attention weight. This is much more reasonable since in recommendation systems, most of users can only access a very small fraction of items within each session. Hence, the session with a 0 first-order Jacarrd score should not be simply identified as the irrelevant one.

To have a deeper look, we take two examples whose 1-order Jaccard scores are both 0 from Tmall dataset. We show their first- and second-order Jaccard scores as well as the attention weights in Figure 6. From Figure 6, it is clear that even if the first-order and second-order Jaccard scores between and investigated session and the test one are 0 (defined in Eq. 23), these sessions can still possibly contain the items that are eventually purchased.

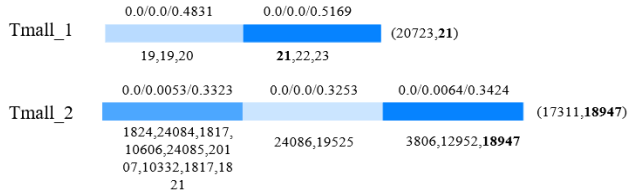


Figure 6: Session relevance w.r.t. first/second order Jaccard and attention weights. The scores above each historical session denote the first-order Jaccard score/the second-order Jaccard score/attention weights. The numbers under each historical session denote the items in this session, and those in brackets are the items purchased in the next session.

5 CONCLUSION

In this paper, we propose a novel model for session-based next-item recommendation by considering users' long- and short-term preferences. We design a two-layer nonlocal neural network to precisely capture user's long-term preferences based on the relationship between the historical sessions and the current ongoing session. We further deploy the GRU network coupled with a nonlocal structure to model short-term preferences. We finally present a personalized strategy to adaptively combine the learned long- and short-term preferences. Empirical results on two real-world datasets demonstrate the effectiveness of our proposed model via the superior performance over the state-of-the-art methods.

ACKNOWLEDGMENTS

This work is supported by Australian Research Council Discovery Project (Grant No. DP190101985 and DP170103954). It is also partially supported by National Natural Science Foundation of China (Grant No. 61572376 and 91646206).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR* (2015).
[2] Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, Yutao Zhu, Pan Du, and Ji-Rong Wen. 2018. An Attribute-aware Neural Attentive Model for Next Basket Recommendation. In *SIGIR*. 1201–1204.
[3] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating dwell time in session-based recommendations with recurrent Neural networks. In *CEUR Workshop*. 57–59.
[4] Tong Chen, Hongzhi Yin, Hongxu Chen, Rui Yan, Quoc Viet Hung Nguyen, and Xue Li. 2019. AIR: Attentional Intention-Aware Recommender Systems. In *ICDE*.
[5] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM*. 108–116.
[6] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. DELF: A Dual-Embedding based Deep Latent Factor Model for Recommendation.. In *IJCAI*. 3329–3335.
[7] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *KDD*. 1082–1090.
[8] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*. 39–46.
[9] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. *arXiv preprint arXiv:1804.10862* (2018).
[10] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming Session-based Recommendation. In *KDD*.
[11] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *RecSys*. 161–169.
[12] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM*. 191–200.
[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
[14] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*.
[15] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR*. 230–237.
[16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
[17] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *RecSys*. 241–248.
[18] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying personalized recommendation with user-session context. In *AAAI*. 1858–1864.
[19] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *SIGIR*. 505–514.
[20] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *RecSys*. 306–310.
[21] Yehuda Koren and Robert Bell. 2015. Advances in collaborative filtering. In *Recommender systems handbook*. 77–118.
[22] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.
[23] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *CIKM*. 1419–1428.
[24] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In *KDD*. 1734–1743.
[25] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. *arXiv preprint arXiv:1609.05787* (2016).
[26] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *KDD*. 1831–1839.
[27] Shuzi Niu and Rongzhi Zhang. 2017. Collaborative Sequence Prediction for Sequential Recommender. In *CIKM*. 2239–2242.
[28] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys*. 130–137.
[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.
[30] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*.
[31] Vikas Sindhwani, Serhat S Bucak, Jianying Hu, and Aleksandra Mojsilovic. 2010. One-class matrix completion with low-density factorizations. In *ICDM*.
[32] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *RecSys*. 17–22.
[33] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900* (2016).
[34] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.
[35] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *WWW*. 729–739.
[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.
[37] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*. 403–412.
[38] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based transactional context embedding for next-item recommendation. In *AAAI*.
[39] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. 2015. Geo-SAGE: A Geographical Sparse Additive Generative Model for Spatial Item Recommendation. In *KDD*.
[40] Weiqing Wang, Hongzhi Yin, Shazia Sadiq, Ling Chen, Min Xie, and Xiaofang Zhou. 2016. SPORE: A sequential personalized spatial item recommender system. In *ICDE*.
[41] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2017. Non-local neural networks. *arXiv preprint arXiv:1711.07971* 10 (2017).
[42] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. Tem: Tree-enhanced embedding model for explainable recommendation. In *WWW*.
[43] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based poi embedding for location-based recommendation. In *CIKM*.
[44] Hongzhi Yin, Bin Cui, Yizhou Sun, Zhiting Hu, and Ling Chen. 2014. LCARS: A Spatial Item Recommender System. *TOIS* (2014).
[45] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *IJCAI*.
[46] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *SIGIR*. 729–732.