Exploiting Repeated Behavior Pattern and Long-term Item dependency for Session-based Recommendation

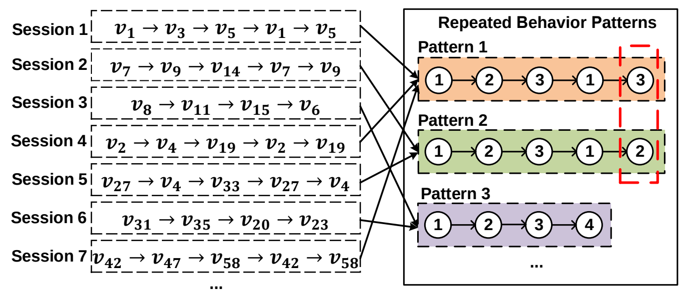
Abstract

Session-based recommendation (SBR) is a challenging task, which aims to predict users' future interests based on anonymous behavior sequences. Existing methods for SBR leverage powerful representation learning approaches to encode sessions into a low dimensional space. However, all the existing studies focus on the item transitions in the session, without modeling the behavior patterns, which are strong clues to capture the preference of users. Further, the long-term dependency within the session is neglected in most of the current methods. To this end, we propose a novel **R**epeat-aware **N**eural **M**echanism for **S**ession-based **R**ecommendation (RNMSR). Specifically, we introduce repeated behavior pattern into SBR, which contains the potential intent information and critical item frequency signal. Furthermore, we also built a similarity-based session graph based on long-term dependencies within a session. Extensive experiments conducted on two benchmark E-commerce datasets, Yoochoose and Diginetica demonstrate our proposed method outperforms the state-of-the-art methods consistently.

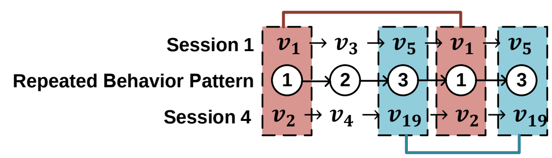
1 Introduction

Recommendation system is an important tool to alleviate the information overload in various applications domains, *e.g.,* e-commerce and social media. Conventional recommendation methods [Mnih and Salakhutdinov, 2008; Kabbur *et al.*, 2013; Hsieh *et al.*, 2017] usually heavily rely on user historical interactions, where information about users and items is essential. However, user identification is inaccessible in many recent real-world scenarios (*e.g.,* unlogged-in user) where only limited interaction of an ongoing session is available. Consequently, session-based recommendation (SBR) has attracted extensive attention recently, which predicts the next interested item based on a given anonymous behavior sequence in chronological order.

Due to its highly practical value, SBR has attracted increasing attention and many kinds of approaches have been developed. The early studies of SBR are based on Markov



(a) Example of sessions and the corresponding repeated behavior patterns.



(b) Toy example of repeated behavior pattern.

Figure 1: Example.

chain [Shani *et al.*, 2005; Rendle *et al.*, 2010], which infers all possible sequences of user choices over items and may suffer from intractable computation problem when the number of items is large. Recently, neural networks have been extensively investigated into SBR, most of which have highlighted the importance of using Recurrent Neural Networks (RNNs) [Hidasi *et al.*, 2016; Li *et al.*, 2017; Wang *et al.*, 2019e; Wang *et al.*, 2019c; Kang and McAuley, 2018; Wang *et al.*, 2019b] and Graph Neural Networks (GNNs)[Wu *et al.*, 2019; Xu *et al.*, 2019; Li *et al.*, 2016]. Although many existing methods have achieved exciting performance, none has focused on repeat consumptions, which not only exists but also accounts for a large proportion of the interactions in many recent real-world scenarios, such as regularly eat at a restaurant or frequently listen a list of songs.

To the best of our knowledge, RepeatNet [Ren *et al.*, 2019] is the only work that explicitly considers repeat consumption with neural networks for SBR, which predicts the users' repeat behaviors, but fails to capture item frequency (IF) information, which has been proven that is more important than repeated purchase pattern [Hu *et al.*, 2020]. Furthermore, RepeatNet [Ren *et al.*, 2019] cannot effectively to capture the long-range dependencies of items within a session, which has also been mentioned in [Liu *et al.*, 2018] that is of great importance for SBR. Without loss of generality, we take Figure 1 as an example for illustration. In Figure 1a, we show the toy example of converting session into repeated behavior pattern. And from Figure 1b, we can observe that, different sessions may belong to same repeated behavior pattern. As compared "Pattern 3", sessions with "Pattern 1" is more likely to lead to a re-consumption action, while sessions with "Pattern 3" is more likely to result in a new item purchase. From the observation, it is meaningful to construct the behavior relevance between different sessions, which can obtain a more accurate discriminate function of predicting the latent re-consumption intent for a given session. Furthermore, from Figure 1, the item frequency (TF) contained in the repeated behavior pattern is a critical signal for decision-making, in other words, the higher TF is associated with a higher probability of the corresponding item to appear in the next action within a same session. For instance, users with "Pattern 1" are more likely to re-consumed "item 1" and "item 3", while users with "Pattern 2" are more likely to re-consumed "item 1" and "item 2". In conclusion, repeated behavior pattern is a strong tool to connect the behavior relevance between different sessions, which reveals the potential intent of user.

To address the above issues, we propose a novel GNN-based method called RNMSR, which explicitly model the repeated behavior pattern and long-term dependencies of items for SBR. Specifically, it learns item embeddings from two module: (i) *Repeat Module*, which computes the probability of each item in the session being re-clicked. (ii) *Explore Module*, which computes the probability of new items being clicked. The final prediction is decided by *Discriminate Module*, which leverages multi-layer perceptron (MLP) to compute the probability of executing *Repeat Module* and *Explore Module* based on repeated behavior patterns.

The major contribution of this work are summarized as follows,

- We propose a unified model to explicitly model the repeated behavior patterns and long-term dependencies of items for SBR.
- To the best of our knowledge, this is the first work to introduce repeated behavior pattern information into SBR, which contains the items frequency signal. To our best knowledge, we are the first to present and analyze this phenomenon for SBR.
- We propose a similarity-based session graphs which are built by the long-term dependencies among items.
- The proposed method is evaluated on two benchmark e-commerce datasets and its effectiveness and superiority are demonstrated by extensive experimental results.

2 Preliminaries

In this section, the SBR problem is first formally defined, and then introduce a behavior pattern, *i.e.,* repeated behavior pattern, which is used to measuring the probability of re-clicking/-consuming items of the given session in the next action.

2.1 Problem Definition

Let $\mathbf{V} = \{v_1, v_2, ..., v_m\}$ be the set of all distinct items over anonymous sessions, where each session is denoted by $\mathbf{S} = \{v_1^s, v_2^s, \cdots v_n^s\}$, consists of a sequence of actions (*e.g.,* an item bought by a user) in chronological order, and $v_i^s \in \mathbf{V}$ refers to the i-th interaction with session $\mathbf{S}$, our goal is to predict the next action v_{n+1}^s for session S.

2.2 Repeated Behavior Pattern

Recently, repeated behavior is theoretically and empirically studied in the filed of economics and psychology [Wang *et al.*, 2019a; Anderson *et al.*, 2014; Hu *et al.*, 2020], and has shown considerable improvement in sequential/session-based recommendation [Ren *et al.*, 2019; Hu *et al.*, 2020]. Intuitively, repeated behavior pattern mainly focuses on capturing *item-independent* relations within the repeated behavior pattern, rather than the item-related relations over original sessions. For example, given two sessions, *i.e.,* $v_1 \to v_2 \to v_3 \to v_1 \to v_3$ and $v_4 \to v_5 \to v_6 \to v_4 \to v_6$ correspond to the same repeated behavior pattern $1 \to 2 \to 3 \to 1 \to 3$. Hence, by following the principle as in [Jin *et al.*, 2020], we define it as follows,

Definition 1 ***(Repeated Behavior Pattern*** $(\mathcal{R})$***).*** *Given any session* $\mathbf{S} = \{v_1, v_2, \cdots, v_m\}$*, the* ***repeated behavior pattern*** *indicates an anonymous sequence, which is defined as follows,*

$$\mathcal{R}(\mathbf{S}) = (\mathcal{N}(\mathbf{S}, v_1), \mathcal{N}(\mathbf{S}, v_2), \cdots, \mathcal{N}(\mathbf{S}, v_m)), \quad (1)$$

where $\mathcal{N}(\mathbf{S}, v_i)$ *denotes the number of distinct items in session* $\mathbf{S}$ *before* v_i *that firstly appears in* $\mathbf{S}$*, which is defined as follows,*

$$\mathcal{N}(\mathbf{S}, v_i) = |\{v_1, v_2, \cdots, v_p\}|, p = min_j\{v_j = v_i\}. \quad (2)$$

3 The Proposed Method

In this section, we first present an overview of our proposed GNN-based Neural Mechanism method RNMSR (shown in Figure 2) in Section 3.1, which consists of four major components, *item representation learning*, *repeat module*, *explore module* and *discriminate module*, and then we will detail each component.

3.1 Overview

The goal of the session-based recommendation problem is to recommend the next item based on a sequence of items within a given session $\mathbf{S}$. Without loss of generality, the probability of the next action given a session $\mathbf{S}$ is defined according to the same principle as in RepeatNet [Ren *et al.*, 2019],

$$\Pr(v|\mathbf{S}) \leftarrow \Pr(r|\mathbf{S}) \Pr(v|r, \mathbf{S}) + \Pr(e|\mathbf{S}) P(v|e, \mathbf{S}), \quad (3)$$

where r and e denote *repeat* module and *explore* module, respectively. $\Pr(r|\mathbf{S})$ and $\Pr(e|\mathbf{S})$ refer to the probability of adopting repeat module and explore module, respectively. $\Pr(v|r, \mathbf{S})$ and $\Pr(v|e, s)$ indicate the probability of recommending v for session $\mathbf{S}$ in repeat module and in explore module, respectively. We will detail them in the following sections, respectively.

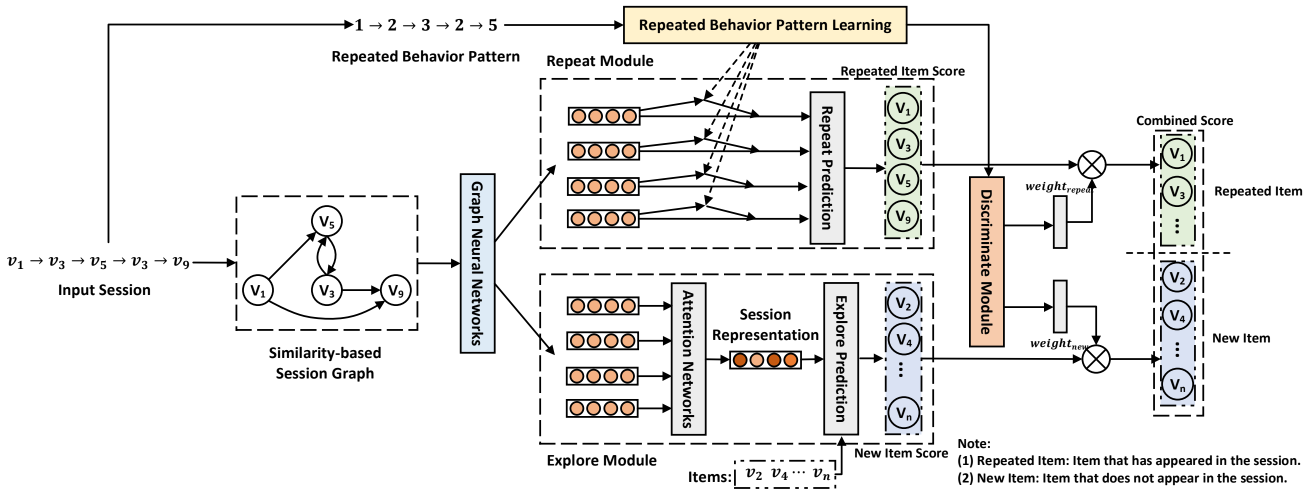


Figure 2: An overview of the proposed framework. We first obtain the repeated behavior pattern from current session and construct a similarity-based session graph. Then the item representations are learned by a GNNs layer. After that, the Repeat Module computes the scores for repeated items based on repeated behavior pattern and the Explore Module compute scores for new items based on Attention Networks. Finally, the Discriminate Module leverages repeated behavior pattern to compute weights for two kinds of scores and obtains the combines scores for all items.

3.2 Item Representation Learning

Aforementioned, traditional RNN-based [Hidasi *et al.*, 2016; Liu *et al.*, 2018; Ren *et al.*, 2019] or GNN-based methods [Wu *et al.*, 2019; Qiu *et al.*, 2019] usually make use of adjacent relations for modeling item relevance, which fail to consider the long-term dependencies among items, and thus easily involved irrelevant information due to the uncertainty of user intents in the session. Hence, here we propose a simple similarity-based session graph model to capture the long-term relations of items.

First, we use an embedding layer to project each item $v \in V$ into a low dimension latent space and the node vector $\mathbf{h} \in \mathbb{R}^d$ is the corresponding d-dimensional vector,

$$\mathbf{h}_i = Embed_{item}(v_i), \tag{4}$$

where v_i is the corresponding one-hot encoding and $Embed_{item}$ is the embedding layer for items.

Similarity-based Session Graph. For any session $\mathbf{S}$, let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ be the corresponding directed session graph, where $v_i \in V$ and $e_{ij} \in \mathcal{E}_s$ denote each node and each edge in $\mathcal{G}_s$, respectively. Here, we adopt *cosine similarity* (similar to [Wang *et al.*, 2020a]) to measure the relevance between two items within session $\mathbf{S}$, *i.e.,*

$$e_{ij} = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{|\mathbf{h}_i||\mathbf{h}_j|}, \tag{5}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ denotes the d-dimension embedding of node v_i. For filtering noise, here we only remain the edges when $e_{ij} > \eta$ (η is a hyper-parameter). And $\mathcal{G}_s$ is a directed graph as we use N_i^{left} and N_i^{right} denote the left neighbor set (*i.e.,* in-link) and right neighbor set (*i.e.,* out-link), respectively.

Item Representation Learning. Based on the built similarity-based session graph, a mean pooling based GNNs layer is proposed to handle two types of neighbors of each item v_i, the representations (*i.e.,* $\mathbf{N}_i^{left}$ and $\mathbf{N}_i^{left}$) of such two types of neighbors are generated by using *Mean Pooling*, *e.g.,* $\mathbf{N}_i^{in} = \text{MeanPooling}(\mathbf{h}_{v \in N_i^{in}})$. The new representation is learnt by a fully connected layer,

$$\mathbf{h}_i^N = \tanh\big(\mathbf{W}_s \mathbf{h}_i + \mathbf{W}_N [\mathbf{N}_i^{left} || \mathbf{N}_i^{right}] + \mathbf{b}_N\big), \tag{6}$$

where $\mathbf{W}_N \in \mathbb{R}^{d \times 2d}, \mathbf{W}_s \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_N \in \mathbb{R}^d$ are trainable parameters.

To reduce the transmission loss, we also use residual connection [He *et al.*, 2016],

$$\mathbf{h}'_i = \mathbf{h}_i^N + \mathbf{h}_i^{all} + \mathbf{h}_i, \tag{7}$$

where $\mathbf{h}_i^{all}$ is an overall representation learnt by mean pooling (*i.e.,* $\mathbf{h}_i^{all} = \text{MeanPooling}(\mathbf{h}_{v \in N_i^{left} \vee N_i^{right} \vee v_i})$).

3.3 Repeat Module

The *repeat module* predicts the possibility of items in the session being re-clicked. As mentioned in [Hu *et al.*, 2020], item frequency (IF) is a critical signal for SBR to make correct recommendation and the higher IF is associated with a higher probability of the corresponding item to be the next clicked item. Hence, we consider to utilize the item frequency to strength the representation of items for improving the accuracy of calculating the re-clicked scores of items within the given session.

Repeated Behavior Pattern Learning. Here, we first convert each session $\mathbf{S}$ into the repeated behavior pattern according to Definition 1. Given a session $\mathbf{S}$, each repeated behavior patten $\mathcal{R}(\mathbf{S})$ is encoded into an unique one-hot encoding $u_{\mathcal{R}(\mathbf{S})}$, and then is projected into an unified low dimensional vector using a simple embedding layer, that is,

$$\mathbf{u}^{\mathcal{R}(\mathbf{S})} = Embed_{pattern}(u_{\mathcal{R}(\mathbf{S})}), \tag{8}$$

where $Embed_{pattern}$ denotes the embedding layer for encoding repeated behavior patten; and $\mathbf{u}^{\mathcal{R}(\mathbf{S})}$ is the representation of session $\mathbf{S}$'s repeated behavior pattern.

Next, we use a trainable reversed position matrix $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$ to encode the position of each item of session $\mathbf{S}$ into a vector according to [Wang *et al.*, 2020b], where $\mathbf{p}_1$ is the vector of the first position and corresponds to the last item h_n in the session sequence. Here we learn the impact of behavior pattern on different positions by a fully connected layer,

$$\mathbf{m}_i = \tanh(\mathbf{W}_m[\mathbf{p}_i \| \mathbf{u}^{\mathcal{R}(\mathbf{S})}] + \mathbf{b}_m), \tag{9}$$

where $\mathbf{W}_m \in \mathbb{R}^{d\times 2d}$ and $b \in \mathbb{R}^d$ are trainable parameters. Then we integrate the learnt impact vector with item features, the score of each item being re-clicked is computed as follows,

$$scores_i^r = \mathbf{q}_r^T \tanh(\mathbf{W}_r \mathbf{h}_i' + \mathbf{U}_r \mathbf{m}_{n-i+1} + \mathbf{b}_r), \tag{10}$$

where $\mathbf{W}_r, \mathbf{U}_r \in \mathbb{R}^{d\times d}$ and $\mathbf{q}_r, \mathbf{b}_r \in \mathbb{R}^d$ are trainable parameters. Finally, the probability of each item is obtained by normalizing through the softmax function:

$$P(v_i|r, \mathbf{S}) = \frac{\exp(scores_i^r)}{\sum_{v_k \in S} \exp(scores_k^r)}, \tag{11}$$

3.4 Explore Module

The *explore module* predicts the possibility of items that do not appear in session being clicked. In explore module, a session-level representation is learned to model the main preference of users. As each item in the sequence has different importance to the current session, we utilize attention mechanism to learn the importance weights for each item based on reversed position vectors,

$$\alpha_i = \mathbf{q}_e^T \tanh(\mathbf{W}_e \mathbf{h}_i' + \mathbf{U}_e \mathbf{p}_{n-i+1} + \mathbf{b}_e), \tag{12}$$

where $\mathbf{W}_e, \mathbf{U}_e \in \mathbb{R}^{d\times d}$ and $\mathbf{q}, \mathbf{b}_e \in \mathbb{R}^d$ are trainable parameters. Then we adopt softmax to normalize the importance weights and the session representation is computed by weighted sum of each item's features,

$$\begin{aligned} \alpha_i &= \text{softmax}(\alpha_i) \\ \mathbf{s}_e &= \sum_i \alpha_i \mathbf{h}_i^e \\ \mathbf{s}_e' &= \tanh(\mathbf{W}_s \mathbf{s}_\mathbf{e} + \mathbf{b}_s) + \mathbf{s}_\mathbf{e}, \end{aligned} \tag{13}$$

where $\mathbf{W}_s \in \mathbb{R}^{d\times d}$ is a trainable parameter and we employ a fully connected layer with residual connection to enhance the nonlinear ability of the model.

Based on the learnt session representation $\mathbf{s}_e'$, the score of each item is computed by inner product between $\mathbf{s}_e'$ and its own features. Then softmax function is used to normalize the scores,

$$\begin{aligned} scores_i^e &= \begin{cases} -\infty & v_i \in \mathbf{S} \\ {s_e'}^T v_i & v_i \in V - \mathbf{S} \end{cases} \\ P(v_i|e, \mathbf{S}) &= \frac{\exp(scores_i^e)}{\sum_{k=1}^m \exp(scores_k^e)}, \end{aligned} \tag{14}$$

where $-\infty$ denotes negative infinity.

3.5 Discriminate Module

The *discriminate module* computes the probability of executing *repeat module* and *explore module*. Different from [Ren *et al.*, 2019] which entirely relies on item features, we incorporate behavior pattern information to enhance the model's ability to grasp the regular habits of users. First we apply self-attention [Vaswani *et al.*, 2017] to learn the importance of each item and obtain a fixed-length representation,

$$\begin{aligned} \beta_i &= \text{softmax}(\mathbf{q}_d^T \tanh(\mathbf{W}_d \mathbf{h}_i' + \mathbf{b}_d)) \\ \mathbf{s}_d &= \sum_i \beta_i \mathbf{h}_i, \end{aligned} \tag{15}$$

where $\mathbf{W}_d \in \mathbb{R}^{d\times d}$ and $\mathbf{q}_d, \mathbf{b}_d \in \mathbb{R}^d$ are trainable parameters. Then we use an L-layer Multi-layer Perceptron (MLP) to extract the latent condensed features from behavior pattern information and item features,

$$\begin{aligned} \mathbf{z} &= [\mathbf{u}^{\mathcal{R}(\mathbf{S})} || \mathbf{s}_d] \\ \mathbf{z}_L &= \mathcal{M}(\mathcal{M}(\cdots \mathcal{M}(\mathbf{z}))) = \mathcal{M}^L(\mathbf{z}), \end{aligned} \tag{16}$$

where $\mathcal{M}(\mathbf{z}) = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b})$ is a fully-connected layer. The probability distribution is learned by softmax regression,

$$[P(r|\mathbf{S}), P(e|\mathbf{S})] = \text{softmax}(\mathbf{W}_p \mathbf{z}_L), \tag{17}$$

where $\mathbf{W}_p \in \mathbb{R}^{2\times d}$ is a learnable transform weight, $P(r|\mathbf{S})$ and $P(e|\mathbf{S})$ are two scalars (*i.e.,* $weight_{repeat}$ and $weight_{new}$ in Figure 2) represent the probability of executing repeat module and explore module, respectively.

3.6 Optimization

The output prediction probability for each item can be computed as follow,

$$P(v_i|\mathbf{S}) = P(r|\mathbf{S})P(v_i|r, \mathbf{S}) + P(e|\mathbf{S})P(v_i|e, \mathbf{S}). \tag{18}$$

Our goal is to maximize the prediction probability of the ground truth item, the loss function is defined as the cross-entropy of the prediction results:

$$\mathcal{L} = -\sum_{i=1}^{|V|} \mathbf{y}_i \log(P(v_i|\mathbf{S})), \tag{19}$$

where $\mathbf{y}$ denotes the one-hot encoding vector of the ground truth item.

4 Experiments

In this section, we first introduce the experimental settings. Then we compare the proposed RNMSR with various baseline methods, and make detailed analysis on the experimental results.

4.1 Experimental Settings

Datasets

To evaluate the performance of our method, two representative benchmark datasets are employed, namely, **Diginetica**[1]

[1] http://cikm2016.cs.iupui.edu/cikm-cup/

Table 1: Statistics of the used datasets.

Dataset	Diginetica	Yoochoose 1/64	Yoochoose 1/4
# clicks	982,961	557,248	8,326,407
# items	43,097	16,766	29,618
# train sessions	719,470	369,859	5,917,745
# test sessions	60,858	55,898	55,898
avg. len.	5.12	6.16	7.42

and **Yoochoose**[2]. The Diginetica dataset comes from CIKM Cup 2016, containing anonymous transaction data. The Yoochoose dataset is obtained from the RecSys Challenge 2015, which consists of six mouths click-streams of an E-commerce website.

We conduct the same preprocessing steps as [Wu *et al.*, 2019] over two datasets. Since the training set of Yoochoose is extremely large, following [Wu *et al.*, 2019], we use the most recent portions $1/64$ and $1/4$ of the training sequences, denoted as "$Yoochoose1/64$" and "$Yoochoose1/4$" datasets, respectively. The statistics of preprocessed datasets are summarized in Table 1.

Evaluated Methods

To evaluate the performance for session based recommendation, we compare our proposed method with multiple baselines including several state-of-the-art models: (1) **POP**: A simple method which directly recommends the most popular items in the training set. (2) **Item-KNN** [Sarwar *et al.*, 2001]: A item-based collaborative filtering algorithm that recommends items similar to the historical items. (3) **FPMC** [Rendle *et al.*, 2010]: A personalized Markov chain model that utilizes matrix factorization for session based recommendation. (4) **GRU4Rec** [Hidasi *et al.*, 2016]: A RNN-based neural network mechanism which uses Gated Recurrent Unit to model the sequential behavior of users. (5) **NARM** [Li *et al.*, 2017]: A hybrid model which improves the GRU4Rec by incorporating an attention mechanism into RNN. (6) **STAMP** [Liu *et al.*, 2018]: An attention-based deep learning model which mainly uses the last item to capture the short-term interest of user. (7) **RepeatNet** [Ren *et al.*, 2019]: A state-of-the-art GRU-based method which proposes a repeat-explore mechanism to model the regular habits of users. (8) **SR-GNN** [Wu *et al.*, 2019]: It employs a gated GNN layer to learn item embeddings, followed by a self-attention of the last item to obtain the session level representation. (9) **GCE-GNN** [Wang *et al.*, 2020b]: A state-of-the-art GNN-based model that additionally employs global context information and reversed position vectors.

Evaluation Metrics

We adopt two widely used ranking based metrics for session-based recommendations: **P@N** and **MRR@N** by following previous work [Wu *et al.*, 2019]. The P@N score indicates the precision of the top-N recommended items. The MRR@N score is the average of reciprocal rank of the correctly-recommended items in the top-N recommendation items. The MRR score is set to 0 when the rank of ground-truth item exceeds N. In this paper, we set $N = 20$ for both P@N and MRR@N.

[2]https://competitions.codalab.org/competitions/11161

Table 2: The performance of evaluated methods on three datasets.

Method	Diginetica		Yoochoose 1/64		Yoochoose 1/4	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
POP	1.18	0.28	7.31	1.69	1.37	0.31
Item-KNN	35.75	11.57	51.60	21.81	52.31	21.70
FPMC	22.14	6.66	45.62	15.01	51.86	17.50
GRU4Rec	30.79	8.22	60.64	22.89	59.53	22.60
NARM	48.32	16.00	68.37	28.87	69.73	29.23
STAMP	46.62	15.13	68.74	28.67	70.44	30.00
RepeatNet	48.49	17.13	70.06	30.55	70.71	31.03
SR-GNN	50.73	17.59	70.57	30.94	71.36	31.89
GCE-GNN	54.22	19.04	70.90	31.26	71.40	31.49
RNMSR	**54.66**	**20.00**	**72.11**	**33.01**	**72.22**	**33.43**

Implementation Details

Since the amount of distinct repeated behavior patterns grows factorially with the length of session, we obtain the repeated behavior patterns from the last 6 items of each session. Following previous methods [Liu *et al.*, 2018; Wu *et al.*, 2019], the dimension of the latent vectors is fixed to 100, and the size for mini-batch is set to 100. And we keep the hyper-parameters of all evaluated methods consistent for a fair comparison. For our model, all parameters are initialized with a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. We use the Adam optimizer [Kingma and Ba, 2014] with the initial learning rate 0.001, which will decay by 0.1 after every 3 epochs, and the L2 penalty is set to 10^{-5}. To avoid overfitting, we adopt dropout layer [Srivastava *et al.*, 2014] after the embedding layer of items (*i.e.,* Equation 4). The dropout ratio is searched in $\{0, 0.25, 0.5\}$ and threshold η is searched in $\{0, 0.1, 0.2, \cdots, 0.9\}$ on a validation set, which is a random 10% subset of the training set.

4.2 Experimental Results

The experimental results of all baselines and our proposed method are reported in Table 2, where the best result of each column is highlighted in boldface.

From Table 2, we observe that RNMSR *consistently* outperforms both traditional methods and neural network methods on three datasets in terms of the two metrics. The experimental results demonstrate the effectiveness of our proposed method. To better understand the performance of different models, we present thorough discussions as follows.

Among the traditional methods, the performance of POP is relatively poor, as it ignores the preference of users and simply recommends top-N popular items. FPMC performs better than POP over three datasets, which shows the effectiveness of using first-order Markov Chain to model session sequences. Comparing with POP and FPMC, Item-KNN achieves better performance by computing the similarity between items, which indicates the importance of co-occurrence information. However, it cannot capture the sequential transitions between items due to the fact that it fails to capture chronological orders in the sessions.

Different from traditional methods, deep learning-based baselines obtain better performance over all datasets. GRU4Rec is a RNN-based method for SBR, which is able to achieve similar or better results than traditional methods. This result demonstrates the strength of RNN in modeling sequential data. However, GRU4Rec is incapable of capturing the user's preference as it merely regards SBR as a sequence modeling task. The subsequent methods, NARM and STAMP significantly outperforms GRU4Rec over three datasets. NARM explicitly captures the main preferences of users and STAMP utilizes attention mechanism to consider user's short-term preference, which lead them to perform better than GRU4Rec. By considering the repeat consumption patterns of users, RepeatNet outperforms traditional methods and other RNN-based methods, which shows the importance of modeling users' regular habits. However, the improvement is still marginal as shown in Table 2, which may be caused by two reasons: it is insufficient to model the repeat consumption behaviors only based on item features, and RNN-based architecture can not capture the collective dependencies within the session.

By converting every session sequence into a subgraph and ecoding items within the session via GNNs, SR-GNN and GCE-GNN achieve better results than RNN models. Specifically, SR-GNN employs a gated GNN layer to learn the collective dependencies within the session, and GCE-GNN explores the global context of each item from the transitions in all training sessions. However, these two methods neglect the regular habits of user and the underlying frequency signal in repeated behavior patterns. Moreover, the constructed session graphs are unable to capture the long-range dependencies within the sessions.

Our proposed RNMSR model outperforms all the baselines. Specifically, RNMSR outperforms the best result of baselines by $0.8\%, 1.7\%, 1.1\%$ in terms of P@20 and $5.0\%, 5.6\%, 6.1\%$ in terms of MRR@20 on three datasets respectively. RNMSR explicitly models the regular habits of users through repeated behavior patterns, which highly improves the performance of the method. Moreover, the proposed similarity-based session graphs enable the RNMSR to capture the long-range dependencies in the sessions without introducing noise.

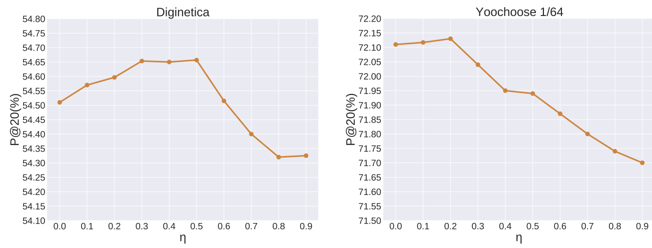
4.3 Ablation Study

To investigate the effectiveness of repeated behavior pattern and similarity-based session graphs, we conduct the following ablation studies as stated in table 3. In (1), we remove the GNN layer from RNMSR and use the original item features. In (2), we construct the session graph as previous studies [Wu *et al.*, 2019; Xu *et al.*, 2019]. In (3), we remove the repeated behavior pattern from repeat module. In (4), we remove the repeated behavior pattern form discriminate module. In (5), we remove the repeated behavior pattern learning layer from RNMSR. In (6), the overall RNMSR model is presented.

From the results presented in table 3, we have the following observations. First, from (1) and (4), the results show that the proposed GNNs layer is powerful to extract features from sessions and can improve the performance of model. Second, the comparison between (2) and (4) indicates that using our similarity-based session graphs can slightly improves the model performance, which demonstrates the superiority of our similarity-bsed session graphs. Lastly, by comparing (3) - (5) and (4), we can observe that incorporating repeated behavior pattern can highly improve the performance, especially in terms of MRR@20. It confirms the strength of repeated behavior pattern to model the habits of users.

Table 3: The ablation study. SSG denotes the similarity-based session graph, and RBP denotes the repeated behavior pattern.

Method	Diginetica		Yoochoose 1/64	
	P@20	MRR@20	P@20	MRR@20
(1) w/o GNN	53.06	19.82	71.08	32.37
(2) w/o SSG	54.26	19.94	71.78	32.73
(3) w/o RBP-r	54.34	19.40	71.94	32.48
(4) w/o RBP-d	54.18	18.86	71.91	32.36
(5) w/o RBP	54.14	18.38	71.90	31.72
(6) RNMSR	**54.66**	**20.00**	**72.11**	**33.01**



(a) (b)

Figure 3: The impact of threshold η.

4.4 Impact of Hyper-parameters

The hyper-parameter η is important when constructing the similarity-based session graphs in the feature space. Thus, we conduct experiments to evaluate the impact of η on the proposed method. [3]

The MRR@20 results are shown in Figure 3. We can observe that when the η is close to 1, the performance of RNMSR becomes worse on both datasets, as there are few neighbors for each item. And the model does not perform well when η is set close to 0 on Diginetica dataset because there may be too much connection noise. The model achieves best the MRR@20 score when η is set to 0.2 on Yoochoose 1/64 and 0.5 on Diginetica, respectively.

5 Related Work

Markov Chain-based SBR. Most traditional recommendation methods are designed for explicit user-item interactions, and some have been employed for session-based recommendation. Shani *et al.* [Shani *et al.*, 2005] employ markov de-

[3] The range of similarity weight $\mathbf{E}_{ij}$ in Equation 5 is $(-1, 1)$, and we evaluate the impact of η from 0 to 1.

cision processes (MDPs) for SBR. Rendle *et al.* [Rendle *et al.*, 2010] apply Markov chain to model the transitions over user-item interactions and the prediction of next action based on the recent interactions of a user. However, Markov chain-based models suffer from one inherent limitation as they can only capture the short-term dependencies while neglecting the long-term dependencies in the sequence.

Deep-learning based SBR. Recently, deep neural networks have dominated SBR and show significant improvements over traditional recommendation approaches. Among these architectures, RNNs achieves promising results for its natural strength to model sequential data. By considering the session sequence as the input of RNNs, Hidasi *et al.* [Hidasi *et al.*, 2016] apply RNN with Gated Recurrent Unit (GRU) for SBR, which is then extended by introducing data augmentation technology [Tan *et al.*, 2016]. Li *et al.* [Li *et al.*, 2017] take the user's main purpose into account and propose NARM to explore a hybrid GRU encoder with attention mechanism to model the sequential behavior of user. To emphasize the importance of the last-click in the session, Liu *et al.* [Liu *et al.*, 2018] propose an attention-based short-term memory networks (named STAMP), which uses external memory to capture user's interests in general and his/her current interests. Further, Song *et al.* [Song *et al.*, 2019] introduce variational autoencoder (VAE) into RNN at each timestamp to model the interest shift of user. MCPRNs [Wang *et al.*, 2019e] proposes GRU-based routing networks to model multi-purpose user sequential behavior. RepeatNet [Ren *et al.*, 2019] proposes an encoder-decoder structure to model the regular habits of user. However, RNN-based methods focus on modeling the sequential transitions of adjacent items [Wang *et al.*, 2019d], which only captures the pair-wise dependencies while ignoring the collective dependencies.

Recently, with the development of graph neural networks, GNNs-based methods attract increasing attention in SBR. SR-GNN [Wu *et al.*, 2019] employs Gated GNN [Li *et al.*, 2016] to learn the item embedding from session graph and use attentions to integrate each learnt item embedding. Following the success of SR-GNN, GC-SAN [Xu *et al.*, 2019] proposes to combine GNNs with self attention networks (SANs) to further improve the performance. Qiu *et al.* [Qiu *et al.*, 2019] propose FGNN which apply graph attention networks [Veličković *et al.*, 2018] to learn item representation. Wang *et al.* [Wang *et al.*, 2020b] propose GCE-GNN which introduces global context information and reversed position vectors into SBR. However, these methods ignore users' regular habits and the GNNs they apply is hard to capture the long-range dependencies.

To the best of our knowledge, RepeatNet [Ren *et al.*, 2019] is the only work explicitly considers repeat consumption with neural networks for SBR, which predicts the users' repeat behaviors, but fails to capture item frequency (IF) information (which has been proven that is more important than repeated purchase pattern [Hu *et al.*, 2020]), and cannot effectively to capture the long-range dependencies of items within a session (which has also been mentioned in [Liu *et al.*, 2018] that is of great importance for SBR). In contrast, our proposed model converts long-term dependencies of items into similarity-based session for learning more accurately item representation, and exploring repeated behavior patterns with item frequency for modeling re-consumption actions for SBR.

6 Conclusion

In this paper, we study the problem of session-based recommendation, which is a challenging but practical task. We propose a novel architecture for session-based recommendation via leveraging repeated behavior patterns and long-term dependencies of items over sessions. Specifically, we convert sessions into repeated behavior pattern with item frequency for predicting re-consumption actions, and build similarity-based session graphs based on long-term item dependencies for learning item representations more accurately. Extensive experiments on two large-scale real world datasets, which demonstrate that the repeated behavior pattern is effective to improve the accuracy of predicting next re-consumption action, and the proposed method significantly outperforms baseline methods, indicating it can be effectively used to solve real-world session-based recommendation problems.

References

[Anderson *et al.*, 2014] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. The dynamics of repeat consumption. In *WWW*, pages 419–430, 2014.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2016.

[Hsieh *et al.*, 2017] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *WWW*, pages 193–201, 2017.

[Hu *et al.*, 2020] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. Modeling personalized item frequency information for next-basket recommendation. 2020.

[Jin *et al.*, 2020] Yilun Jin, Guojie Song, and Chuan Shi. Gralsp: Graph neural networks with local structural patterns. In *AAAI*, pages 4361–4368, 2020.

[Kabbur *et al.*, 2013] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *SIGKDD*, pages 659–667, 2013.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *ICDM*, pages 197–206, 2018.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2016] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *ICLR*, 2016.

[Li *et al.*, 2017] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *CIKM*, pages 1419–1428, 2017.

[Liu *et al.*, 2018] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: short-term attention/memory priority model for session-based recommendation. In *SIGKDD*, pages 1831–1839, 2018.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2008.

[Qiu *et al.*, 2019] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. Rethinking the item order in session-based recommendation with graph neural networks. In *CIKM*, pages 579–588, 2019.

[Ren *et al.*, 2019] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *AAAI*, 2019.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.

[Sarwar *et al.*, 2001] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[Shani *et al.*, 2005] Guy Shani, David Heckerman, and Ronen I Brafman. An mdp-based recommender system. In *JMLR*, pages 1265–1295, 2005.

[Song *et al.*, 2019] Jing Song, Hong Shen, Zijing Ou, Junyi Zhang, Teng Xiao, and Shangsong Liang. Islf: Interest shift and latent factors combination model for session-based recommendation. In *IJCAI*, pages 5765–5771, 2019.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, pages 1929–1958, 2014.

[Tan *et al.*, 2016] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 17–22, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang *et al.*, 2019a] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *WWW*, pages 1977–1987, 2019.

[Wang *et al.*, 2019b] Huizhao Wang, Guanfeng Liu, An Liu, Zhixu Li, and Kai Zheng. Dmran:a hierarchical fine-grained attention-based network for recommendation. In *IJCAI*, pages 3698–3704, 2019.

[Wang *et al.*, 2019c] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. A collaborative session-based recommendation approach with parallel memory modules. In *SIGIR*, 2019.

[Wang *et al.*, 2019d] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. Sequential recommender systems: Challenges, progress and prospects. In *IJCAI*, pages 6332–6338, 2019.

[Wang *et al.*, 2019e] Shoujin Wang, Liang Hu, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Longbing Cao. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *IJCAI*, pages 3771–3777, 2019.

[Wang *et al.*, 2020a] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. Am-gcn: Adaptive multi-channel graph convolutional networks. In *SIGKDD*, pages 1243–1253, 2020.

[Wang *et al.*, 2020b] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. Global context enhanced graph neural networks for session-based recommendation. In *SIGIR*, pages 169–178, 2020.

[Wu *et al.*, 2019] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *AAAI*, pages 346–353, 2019.

[Xu *et al.*, 2019] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*, pages 3940–3946, 2019.