MBN: Towards Multi-Behavior Sequence Modeling for Next Basket Recommendation

YANYAN SHEN, BAOYUAN OU, and RANZHEN LI, Department of Computer Science and Engineering, Shanghai Jiao Tong University

Next basket recommendation aims at predicting the next set of items that a user would likely purchase together, which plays an important role in e-commerce platforms. Unlike conventional item recommendation, the next basket recommendation focuses on capturing item correlations among baskets and learning the user's temporal interest from the past purchasing basket sequence. In practice, most users interact with items in various kinds of behaviors. The multi-behavior data sheds light on user's potential purchasing intention and resolves noisy signals from accidentally purchased items. In this article, we conduct an empirical study on real datasets to exploit the characteristics of multi-behavior data and confirm its positive effects on next basket recommendation. We develop a novel Multi-Behavior Network (MBN) model that captures item correlations and acquires meta-knowledge from multi-behavior basket sequences effectively. MBN employs the meta multi-behavior sequence encoder to model temporal dependencies of each individual behavior and extract meta-knowledge across different behaviors. Furthermore, we design the recurring-item-aware predictor in MBN to realize the high degree of the repeated occurrences of items, leading to better recommendation performance. We conduct extensive experiments to evaluate the performance of our proposed MBN model using real-world multi-behavior data. The results demonstrate the superior recommendation performance of MBN compared with various state-of-the-art methods.

CCS Concepts: • **Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Recommendation, sequence modeling, multi-behavior data

ACM Reference format:
Yanyan Shen, Baoyuan Ou, and Ranzhen Li. 2022. MBN: Towards Multi-Behavior Sequence Modeling for Next Basket Recommendation. *ACM Trans. Knowl. Discov. Data.* 16, 5, Article 81 (March 2022), 23 pages.
https://doi.org/10.1145/3497748

1 INTRODUCTION

Recommender systems play an increasingly important role in filtering overwhelmed item information and providing personalized offers for users. In most real-world scenarios, users often purchase a basket of items during a visit to an e-commerce platform. **Next basket recommendation** (**NBR**),

This work is supported by the National Key Research and Development Program of China (No. 2018YFC0831604). Yanyan Shen is also supported by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), the Tencent Wechat Rhino-Bird Focused Research Program, and SJTU Global Strategic Partnership Fund (2021 SJTU-HKUST).
Authors' address: Y. Shen (corresponding author), B. Ou, and R. Li, Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang, Shanghai 200240; emails: {shenyy, gybuay}@sjtu.edu.cn, liranzhen@alumni.sjtu.edu.cn.

1556-4681/2022/03-ART81 $15.00
https://doi.org/10.1145/3497748

which aims at accurately predicting the set of items that a user would like to purchase in the next basket, is an essential task in recommender systems [39, 46]. The NBR problem distinguishes itself from the conventional item recommendation in two aspects. First, NBR requires capturing item correlations among baskets. For example, items in the same basket can be correlated to each other (e.g., bread and milk) or dependent on the items purchased in the previous baskets. Second, NBR has to learn a user's sequential purchasing interest from previously purchasing basket sequence. However, item recommendation has focused on modeling static user and item characteristics and exploring their interaction mode [18]. To this end, the widely adopted collaborative filtering methods such as latent factor models [1, 2, 7, 8, 27, 38] are thus inappropriate to recommend the next basket of items.

A variety of NBR researches have emerged with success, which mainly include **Markov Chain** (**MC**)-based methods and **Recurrent Neural Network** (**RNN**) based methods. MC-based models [39, 43], model user preference by the last K interaction(s) in a K-order MC, but ignore the other previous interactions. Recently, RNN has become a better choice in dealing with basket sequences, due to its impressive capability in modeling natural language sequences. RNN-based methods [4, 16, 29, 30, 34, 46] leverage all the interactions in a long sequence of an anonymous user, capturing the user's temporal interaction patterns. These approaches have been verified to be effective in predicting the next purchasing basket, but they do not consider different behaviors.

In practice, users often interact with items in a variety of behaviors. In addition to purchasing items, a user may interact with items by clicking, adding to cart, tagging with favor, and so on. In fact, the temporal item interactions of a user with multiple behaviors can be naturally organized into separate basket sequences, as illustrated in Figure 1. It is important to notice that exploring the information from multi-behavior sequences is beneficial to the NBR for two reasons. First, a user would likely purchase an item after showing interest in it via clicking the item link or adding it to the cart. Hence, non-purchasing sequences shed light on user's potential purchasing intention, especially for the new items that have not been purchased before. Second, learning collective knowledge from multi-behavior sequences improves the understanding of user's true latent interest, reducing noises from accidentally purchased items. In this article, we focus on the following problem: *given multiple basket sequences under distinct behaviors, how can we exploit multi-behavior knowledge to improve the accuracy of recommending next purchasing basket?*

Ideally, when modeling multi-behavior sequences, we would like to capture the latent and dynamic item correlations within and across sequences to disclose a user's purchasing habit, her general interest and the evolving pattern. While a line of researches [11–13, 23, 26, 35, 40, 42, 45, 47, 48] have studied the multi-behavior data, they focus on the typical recommendation task and treat multi-behavior data as auxiliary information to address the well-recognized data sparsity challenge. They are thus insufficient to solve the NBR problem due to the ignorance of temporal information and item correlations encoded in multi-behavior sequences. Another line of researches [15, 28, 32, 33, 37, 49] developed RNN-based models over multi-behavior sequential data for next item or session-based recommendation. For example, BINN [32] uses a single RNN to deal with different behaviors mixedly. These approaches do not establish a direct connection among the past items interacted with different behaviors or explicitly modeling their relationship to the item set to be purchased next.

To tackle the problem, we conduct a comprehensive empirical study on real datasets to disclose the characteristics of multi-behavior sequences and comprehend the strengths and challenges of exploiting multi-behavior data for NBR. Based on our observations, we propose a novel **Multi-Behavior Network** (**MBN**) model that learns collective knowledge from multi-behavior sequences towards better recommendation results. Our model consists of three major components. The basket encoder is developed to capture item correlations within any basket in a sequence. The

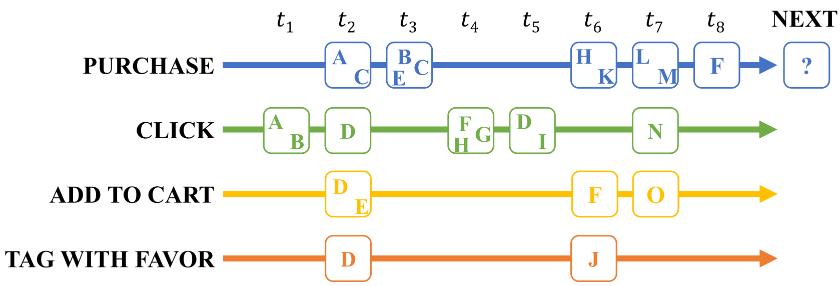


Fig. 1. NBR with multi-behavior basket sequences, where t_i represents a time span. The diagram involves four behaviors: purchasing, clicking, adding to cart, and tagging with favor.

meta multi-behavior sequence encoder is used to learn a user's general interest and its evolving trend by taking into consideration the collective meta-knowledge extracted from multi-behavior sequences. We introduce two schemes (i.e., gathering and gathering-scattering) that model the dependencies across sequences in different manners. Furthermore, to realize the repeated occurrences of items in historical user-item interactions, we present the recurring-item-aware predictor based on the copy mechanism [14] where a mixed probabilistic function is established to predict the probability of each item in the next basket given the multi-behavior knowledge.

The major contributions of this article are summarized as follows:

— We propose to exploit multi-behavior sequences for the NBR. An end-to-end neural approach is developed to learn collective behavioral knowledge from sequences towards better recommendation performance.
— We conduct an empirical study to examine the characteristics of multi-behavior sequence data and provide several insights on how to model multi-behavior sequences for NBR.
— We propose a novel neural network model named MBN following our insights. Our model consists of three components to compute basket representations, learn temporal meta-knowledge across multi-behavior sequences, and realize the recurring-item pattern, respectively.
— We conduct extensive experiments on three real-world datasets. The results demonstrate the effectiveness of exploiting multi-behavior knowledge for NBR and the superior recommendation performance of our proposed MBN compared with various existing methods.

The remainder of this article is organized as follows. Section 3 provides the definitions and problem statement. Section 4 analyzes the characteristics of multi-behavior sequences. Section 5 presents our MBN model for the NBR. Section 6 demonstrates the experimental results. We conclude this article in Section 7.1.

2 RELATED WORK

NBR can be viewed as a sub-area of sequential recommendation [6, 24, 44]. The general objective of sequential recommendation is to predict the next item according to the past sequence of a user's interactions. In addition to NBR, the sequential recommendation also includes two other sub-areas, i.e., next item recommendation [15, 28, 32, 49] and session-based recommendation [19, 34, 37]. We next focus on the related works proposed for the NBR and multi-behavior recommendation.

2.1 Next Basket Recommendation

NBR aims at predicting the next basket that a user will interact with, based on the previously purchased basket sequence. Recently, a substantial number of approaches have been proposed to the NBR problem, which can be generally divided into two categories: MC-based and RNN-based approaches. MC-based approaches [39, 43] mainly consider two consecutive baskets under the assumption the items to be purchased in the next basket is conditionally independent of the previous baskets given the current one. An important work is **Factorizing Personalized Markov Chain (FPMC)** [39], which organizes users' sequential basket data into a transition cube and performs tensor factorization to capture each user's sequential behavior and general taste. Instead of performing linear combination over sequential and general features, **Hierarchical Representation Model (HRM)** [43] employs a two-layer structure to facilitate more flexible feature interactions. With the advent of deep learning techniques, RNN-based methods such as DREAM [46] investigate historical purchasing sequences to capture long-term user interests and temporal purchasing habits. To enhance RNN approaches, recent attention [4, 29, 34] has been paid to exploiting item attributes and correlations via the attention mechanism [3], or capturing different factors (e.g., co-occurrence, periodicity) of the purchased items [16]. In addition, [22] proposed a kNN method that exploits item frequency information for NBR, achieving the state-of-the-art NBR performance. These researches have exploited different factors of the sequentially purchased items, but they overlook the latent multi-behavior knowledge.

2.2 Multi-Behavior Recommendation

Multi-behavior recommendation [11–13, 23, 26, 35, 40, 42, 45, 47, 48] investigates various behaviors that user interacts with items to recommend precisely on target behaviors. However, these methods dive into solving a typical recommendation scenario, e.g., infer the unseen interact probability between user and item with the target behavior. Although these approaches have been proven effective in processing multi-behavior data, they can not hand out the NBR problem.

Other related works such as BINN [32], RLBL [33], and RIB [49] have modeled user's sequential behaviors. However, they focus on the next item recommendation problem, and more importantly, they mixedly model all the behaviors in the same RNN. RIB [49] exploits dwell time on items to model multi-behavior sequences by RNN. CBS [28] can only utilize one support behavior via a twin network. HUP [15] develops Pyramid Recurrent Neural Networks to capture hierarchical user interests. Different from the above models, MKM-SR [37] constructs knowledge graphs for item attributes and an interactive graph for each session. It employs a graph neural network for the recommendation. However, existing approaches fail to model the item correlations among different behavior sequences explicitly or acquire collective multi-behavior knowledge, which may result in suboptimal performance. Our experimental results have also confirmed the limitations of the existing multi-behavior recommendation methods.

3 DEFINITIONS AND PROBLEM

We consider a set U of users and a set V of items in an e-commerce platform. Let Φ denote the types of different behaviors such as clicking, adding to cart, tagging with favor, purchasing, and so on. For a user $u \in U$ and a behavior $\phi \in \Phi$, we sort by time his or her item records interacted with ϕ, and obtain a *sequence of baskets*.

Definition 1 (Basket Sequence $\mathcal{B}$ with Behavior ϕ). Given a user $u \in U$, the sequence of baskets under behavior ϕ sorted by time is denoted as $\mathcal{B}_u^\phi = \{B_{u,1}^\phi, \ldots, B_{u,T}^\phi\}$, where $B_{u,t}^\phi \subseteq V$ is the basket of items interacted by u under behavior ϕ at time t, and T is the size of the time window.

The conventional NBR [39, 43, 46] only leverages the previous use-item interaction records with the single purchasing behavior (i.e., $\mathcal{B}_u^\phi$), and tries to predict the next basket of items a user would likely buy (i.e., $B_{u,T+1}^\phi$). Since the ground-truth number of items in the next basket is unknown,

Table 1. Notation Table

Notation	Description
Φ	a set of behaviors
$u \in U$	a user in the user set U
$v \in V$	an item in the item set V
$\phi^* \in \Phi$	purchasing (or target) behavior
$\phi \in \Phi$	any behavior in the behavior set Φ
T	time window size of the basket sequences
$B_{u,t}^{\phi}$	a basket of items from u with ϕ at time t
$\mathcal{B}^{\phi}$	a basket sequence for behavior ϕ
$\mathbf{e}_v$	vector of item v
$\mathbf{E}_{u,t}^{\phi}$	vector of basket $B_{u,t}^{\phi}$

the NBR is typically formulated as a prediction problem that infers the probability of each item i that user u would purchase in the next basket. By ranking all the items in V according to their probabilities, we recommend top K items as the next basket to the user.

Definition 2 (NBR with Behavior ϕ). Given a sequence $\mathcal{B}_u^{\phi}$ of baskets with behavior ϕ, the NBR problem aims at estimating the probability of an item v that u would like to interact in the next basket, i.e.,

$$Pr_{\phi}\left(v \in V \mid \mathcal{B}_u^{\phi}\right). \tag{1}$$

The predicted next basket with behavior ϕ is defined as

$$\mathcal{B}_{u,T+1}^{\phi} = \text{TopK}\left(Pr_{\phi}\left(v \in V \mid \mathcal{B}_u^{\phi}\right)\right). \tag{2}$$

In this article, we aim at modeling multiple basket sequences with different behaviors in Φ for NBR, i.e., K items a user would most likely purchase in the next basket. Henceforth, we treat *purchasing* in Φ as the **target behavior** ϕ^*, while the other behaviors $\phi \in \Phi \backslash \{\phi^*\}$ are referred to as **context behaviors**.

Definition 3 (NBR with Multiple Behaviors Φ). Given the basket sequences $\{\mathcal{B}_u^{\phi}\}_{\phi \in \Phi}$ for user u over the previous T time units, the NBR with multiple behaviors is to predict the probability of an item v that u would like to interact with the target behavior ϕ^* in the next basket, i.e.,

$$Pr_{\phi^*}\left(v \in V \mid \mathcal{B}_u^{1}, \ldots, \mathcal{B}_u^{|\Phi|}\right). \tag{3}$$

The predicted next basket with multiple behaviors Φ is computed as

$$\mathcal{B}_{u,T+1}^{\phi^*} = \text{TopK}\left(Pr_{\phi^*}\left(v \in V \mid \mathcal{B}_u^{1}, \ldots, \mathcal{B}_u^{|\Phi|}\right)\right). \tag{4}$$

Without loss of generality, we assume that different basket sequences are time-aligned. That is, each sequence involves T baskets across the same T time units. And $B_{u,t}^{\phi}$ can be empty if no item is interacted by user u with ϕ at time t.

While purchasing is treated as the target behavior in this article, it is worth mentioning that due to the symmetry of our proposed model, we are able to predict the next basket for any behavior without model-level modification. Table 1 summarizes all the symbols and their descriptions used throughout this article. Moreover, our model can predict next baskets for all the behaviors by training with a simple aggregation of the objective functions for different behaviors.

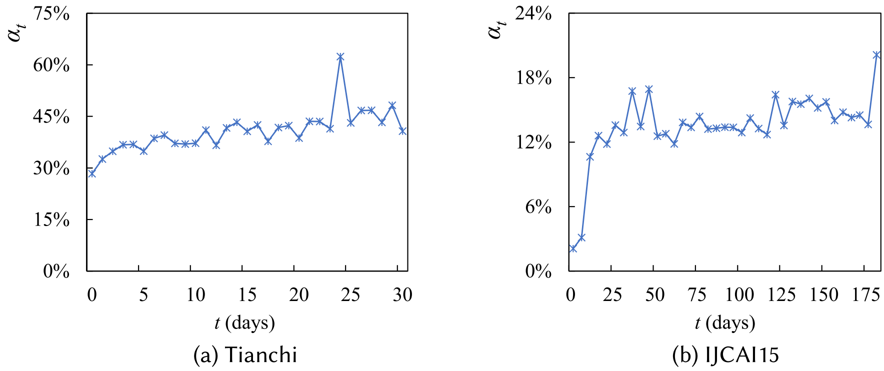


Fig. 2. The proportion of new items (bought for the first time) that were previously interacted (Q1).

4 MULTI-BEHAVIOR SEQUENCES ANALYSIS

In this section, we examine the real-world multi-behavior sequences used for the NBR. The **Tianchi** dataset contains the user-commodity behavior data on Alibaba M-Commerce platforms from 2014-11-18 to 2014-12-18. The **IJCAI15** dataset includes users' shopping logs from 2014-05-11 to 2014-11-11, which are sampled from the actual records collected by *Tmall.com*. Both of the datasets are publicly available involving four types of behaviors: clicking, adding to cart, tagging with favor, and purchasing. We consider an hour and one day (24 hours) as the time units for Tianchi and IJCAI15, respectively. The records are then organized into multi-behavior basket sequences accordingly. We also removed inactive users and unpopular items. The statistics of the datasets after preprocessing will be presented in the experiments.

We conduct the empirical analysis to illustrate the strengths of multi-behavior sequences and the insights of our proposal. Specifically, we answer the following three questions:

Q1. Is utilizing multi-behavior basket sequences beneficial to the NBR?
Q2. Is there any user-item interaction pattern involved in the multi-behavior basket sequences?
Q3. Are multi-behavior basket sequences correlated with each other?

4.1 Q1: Purchasing New Items After Interaction

A rational intuition is that a user is very likely to interact with an item of interest before purchasing it. That is, *the basket sequences with context behaviors provide hints on the items a user would likely buy in the future, especially the "new items" that are never bought by the user before.* Let $N_{u,t}^{\phi^*}$ denote the set of "new items" that are purchased by user u at time t for the first time. Let $S_{u,<t}^{\Phi}$ be the set of items u have interacted before time t. We compute the proportion of new items (bought for the first time) that are contained in the previous basket sequences with context behaviors as

$$\alpha_{u,t} = \frac{\left|N_{u,t}^{\phi^*} \cap S_{u,<t}^{\Phi}\right|}{\left|N_{u,t}^{\phi^*}\right|}. \tag{5}$$

Intuitively, the larger α, the more contributions it has to examine the items interacted by the context behaviors for next basket prediction. Figure 2 provides the results of $\alpha_t = \frac{1}{|U|}\sum_{u \in U} \alpha_{u,t}$ by varying the value of t on the two datasets. On average, about 40% and 12% of the first-time-bought items have been previously interacted by context behaviors on Tianchi and IJCAI15, respectively. Compared with IJCAI15, Tianchi users are more likely to buy new items after interacting them with context behaviors. We also observe that the values of α_t are slightly increasing with t on both

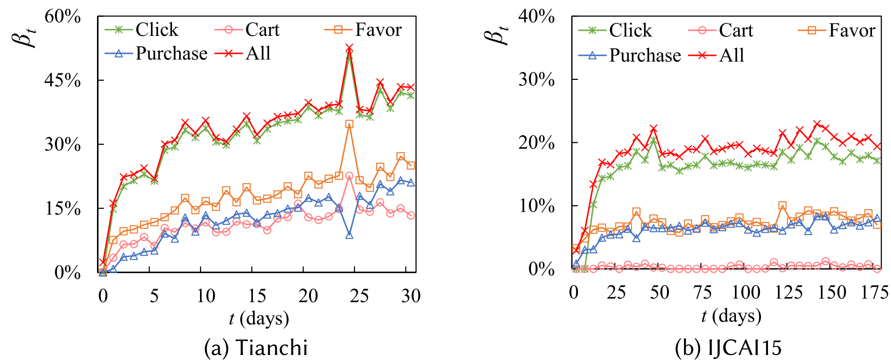


Fig. 3. The mean proportion of purchased items that are included in other behavior basket sequences (Q2).

datasets. This implies that context behaviors over a longer time window would be informative to disclose user's intention to purchase new items.

4.2 Q2: Recurrent Items in Basket Sequences

Prior works [16, 22] have confirmed the repeated occurrences of items in the purchasing basket sequence $\mathcal{B}_u^{\phi^*}$. We observe that *recurrent items also exist between* $\mathcal{B}_u^{\phi^*}$ *and* $\mathcal{B}_u^{\phi}$ *for any* $\phi \in \Phi \backslash \{\phi^*\}$. For example, users often purchase the items that were previously interacted via context behaviors, or buy the same items (e.g., the articles for daily use) repeatedly. Consider a user $u \in U$. Let $S_{u,t}^{\phi^*}$ denote the set of items purchased by u at time t, and $S_{u,<t}^{\phi}$ be the set of items interacted by u with the behavior $\phi \in \Phi$ before t. To measure the degree of the repeated occurrences of items, we compute the proportion of items in $S_{u,t}^{\phi^*}$ that have been interacted by u via ϕ before, which is defined as

$$\beta_{u,t}^{\phi} = \frac{\left|S_{u,t}^{\phi^*} \cap S_{u,<t}^{\phi}\right|}{\left|S_{u,t}^{\phi^*}\right|}, \forall \phi \in \Phi. \tag{6}$$

By computing the mean average of $\beta_{u,t}^{\phi}$ over all the users, we obtain the mean proportion of repeating items at time t, i.e., $\beta_t^{\phi} = \frac{1}{|U|} \sum_u \beta_{u,t}^{\phi}$. As shown in Figure 3, on both datasets, the most important driving force of the repeated occurrences of items comes from the clicking behavior, followed by tagging with favor. This is reasonable as the clicked or tagged items directly reflect user interests, and would be purchased later with higher possibilities. It is also noticeable that users would likely purchase the same item repeatedly. This phenomenon is more significant on Tianchi dataset than IJCAI15. On average, about 35% and 5% of items are bought more than once by Tianchi and IJCAI15 users, respectively. It is interesting to see that the items being added to the cart may not be purchased on IJCAI15. This is because IJCAI15 contains a very small number of carting records, which makes the carting behavior negligible.

4.3 Q3: Item Correlation Among Behaviors

Since utilizing multi-behavior basket sequences is beneficial to next basket prediction, we would like to further examine the correlation between different behavior sequences. As our ultimate goal is to predict the next purchasing basket, we mainly focus on the item overlapping between the target behavior against the context behaviors. For each user u, we compute the Jaccard similarity over two item sets from the purchasing sequence $\mathcal{B}_u^{\phi^*}$ and any of the context sequence $\mathcal{B}_u^{\phi}$,

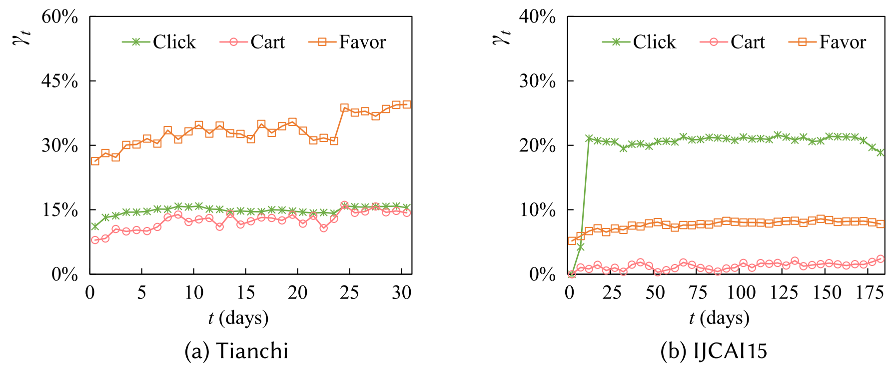


Fig. 4. Item correlation among behavior sequences (Q3).

respectively. Formally, we define the *item correlation* between $\mathcal{B}_u^{\phi^*}$ and $\mathcal{B}_u^{\phi}$ as

$$\gamma_{u,t}^{\phi} = \frac{\left|S_{u,\leq t}^{\phi^*} \cap S_{u,\leq t}^{\phi}\right|}{\left|S_{u,\leq t}^{\phi^*} \cup S_{u,\leq t}^{\phi}\right|}, \forall \phi \in \Phi. \tag{7}$$

Note that $\gamma_{u,t}^{\phi^*} = 1$ holds for all u and t. The larger γ, the higher degree of overlapping between two sequences in terms of the involved items. For illustration purposes, we compute the mean average of Jaccard similarities over all the users, i.e., $\gamma_t^{\phi} = \frac{1}{|U|} \sum_{u \in U} \gamma_{u,t}^{\phi}$. Figure 4 shows the averaged item correlation results by varying t. The results confirm the item correlations between target and context behavior sequences. The overall item correlations on Tianchi are higher than those on IJCAI15. We can see that on Tianchi, the Jaccard similarity between purchased and favored items is about 0.3, which is two times larger than clicking and adding to the cart. As for IJCAI15, the most significant item correlation is observed between purchasing and clicking sequences, with a Jaccard similarity of 0.2. This is because the ratio between the numbers of clicking and purchasing records in IJCAI15 is much smaller than that in Tianchi, i.e., 2.54 versus 6.34, which results in a smaller value of the denominator in γ.

4.4 Summary of Observations

We summarize our key observations as follows:

O1. Exploring historical purchasing records for the NBR is insufficient in terms of identifying items that are never bought before. Particularly, the basket sequences with context behaviors are beneficial to disclose new items of user interest.

O2. The degree of the repeated occurrences of items across multi-behavior sequences is high. A large proportion of items to be purchased were interacted before, especially with the clicking behavior. This implies a *biased item distribution* in the next basket.

O3. Context behavior sequences are correlated with the target sequence in terms of item overlapping. As there is no strict order among context behaviors in terms of their item correlations against the target sequence, it is desirable to learn multi-behavior knowledge collectively without considering the order among behaviors.

5 RECOMMENDING NEXT BASKET WITH MULTI-BEHAVIOR SEQUENCES

The NBR is inherently a multi-label classification problem given multiple sequences as input, where the key challenges lie in (i) capturing temporal dependencies of each individual basket

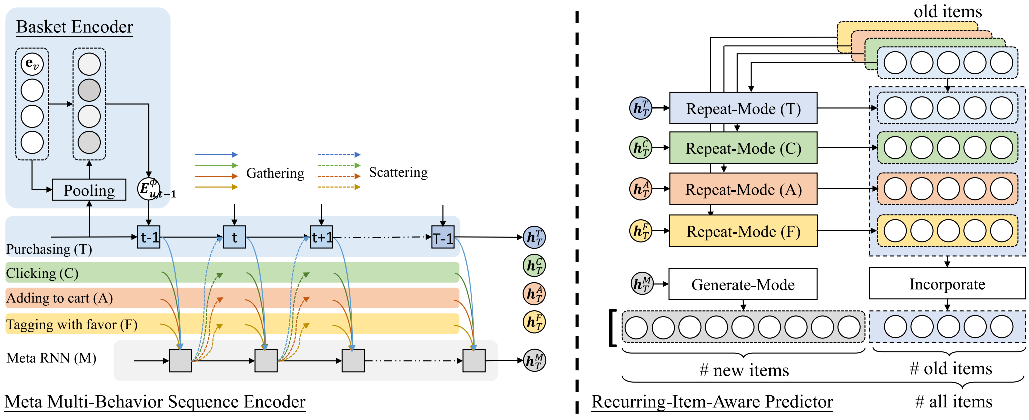


Fig. 5. The Architecture of MBN (with three context behaviors for example): (1) The left top is Basket Encoder with pooling. (2) The left bottom is Meta Multi-Behavior Sequence Encoder including one Meta-RNN and four Behavior-RNNs. (3) The right is Recurring-Item-Aware Predictor where the generate-mode in gray is to score the new items, while the repeat-modes are to score previously interacted items.

sequence and the correlations among sequences, (ii) predicting the next basket based on multi-sequence information. In this section, we introduce the MBN model following our observations. As depicted in Figure 5, our model consists of three major components: The *basket encoder* learns the latent representation for each basket of items. The *meta multi-behavior sequence encoder* captures temporal meta-knowledge and item correlations of multi-behavior sequences. The *recurring-item-aware predictor* generates the next basket by exploiting the fact of the repeated occurrences of items among sequences. In what follows, we provide the details of the three components.

5.1 Basket Encoder

The basket encoder includes an item embedding layer, followed by a basket pooling layer, to obtain the representation of a basket of items. Each item in the set V is initially encoded as a $|V|$-dimensional vector with the one-hot encoding. We apply a shared embedding layer to project the items in V onto a latent space. Let $\mathbf{v}$ denote the one-hot vector for item $v \in V$ and $\mathbf{e}_v$ be the corresponding latent vector. The item embedding layer is defined as a projection matrix $\mathbf{P} \in \mathbb{R}^{|V| \times d_e}$ such that

$$\mathbf{e}_v = \mathbf{P}^\top \mathbf{v}, \quad \forall v \in V, \tag{8}$$

where d_e is the size of the embedding space.

Given a basket $B^{\phi}_{u,t}$ of items, we compute the basket representation by aggregating the embeddings of the involved items, denoted as $\mathbf{E}^{\phi}_{u,t}$. Since the basket size changes dynamically, we apply max pooling to keep the most significant features among the items in the basket. An empty basket is a zero vector. Formally, we have

$$\mathbf{E}^{\phi}_{u,t}[i] = \max_{v \in B^{\phi}_{u,t}} \mathbf{e}_v[i], \quad i \in [1, d_e]. \tag{9}$$

Another alternative is to apply attentive pooling by assigning an importance score to each item in the basket, whereas it may introduce additional model parameters and higher computational cost.

5.2 Meta Multi-Behavior Sequence Encoder

This component aims at capturing the temporal dependencies of each individual behavior sequence and the collective knowledge of multi-behavior sequences. Inspired by the multi-task learning paradigm [5], we devise two networks for encoding multi-behavior sequences. At a high level, a *basic network* is associated with each basket sequence to extract behavior-specific information, and a *meta network* is shared among all the behavior sequences for learning multi-behavior knowledge. We employ RNN to instantiate both networks. RNN variants such as **Gated Recurrent Unit (GRU)** [9] and LSTM [20] can also be applied. Another novel attempt is the two schemes to correlate the basic and meta networks for our encoder, namely, *gathering* and *gathering-scattering* schemes. Without otherwise specified, this article refers to the *basic networks* and *meta network* as **Behavior-RNNs** and **Meta-RNN**, respectively.

5.2.1 *Gathering Scheme.* As illustrated in Figure 5, in this scheme, we supply the sequence of basket representations $\mathbf{E}^{\phi}_{u,1}, \ldots, \mathbf{E}^{\phi}_{u,T}$ produced by Basket Encoder into a Behavior-RNN that converts the basket representations into a series of hidden states. Specifically, for the behavior sequence $\mathcal{B}^{\phi}_{u}$, the hidden state $\mathbf{h}^{\phi}_{t}$ at time t is computed using the following equation:

$$\mathbf{h}^{\phi}_{t} = f\left(\mathbf{W}^{\phi}_{1}\mathbf{E}_{B^{\phi}_{u,t}} + \mathbf{W}^{\phi}_{2}\mathbf{h}^{\phi}_{t-1} + \mathbf{b}^{\phi}\right), \tag{10}$$

where $f(\cdot)$ is a nonlinear activation function, e.g., $\tanh(\cdot)$. $\mathbf{W}^{\phi}$ and $\mathbf{b}^{\phi}$ are the parameters for the behavior $\phi \in \Phi$. In order to capture the behavior-specific sequential information, Behavior-RNN is not shared (in parameters) across behaviors. That is, we can obtain $|\Phi|$ hidden states at time t.

We then construct Meta-RNN by referring to all the hidden states from $|\Phi|$ Behavior-RNNs. At time t, the meta hidden state $\mathbf{h}^{M}_{t}$ of Meta-RNN is computed via the following equation:

$$\mathbf{h}^{M}_{t} = f\left(\mathbf{W}^{M}\left[\mathbf{h}^{1}_{t-1}, \ldots, \mathbf{h}^{|\Phi|}_{t-1}\right] + \mathbf{b}^{M}\right), \tag{11}$$

where $\mathbf{W}^{M}$ and $\mathbf{b}^{M}$ are the weight matrices and bias vector, respectively. The purpose of Meta-RNN is to learn multi-behavior knowledge by *gathering* the information from each individual behavior sequence. Since the number of behaviors is known in advance, we aggregate the hidden states $\{\mathbf{h}^{\phi}_{t}\}_{\phi\in\Phi}$ via the simple concatenation. Finally, the last hidden state $\mathbf{h}^{M}_{T}$ encodes the meta-knowledge from the multi-behavior sequences.

5.2.2 *Gathering-Scattering Scheme.* In practice, the behaviors at time t should be aware of all the item interactions with different behaviors before t. Unfortunately, in the gathering scheme, the item correlations across different behavior sequences are not explicitly modeled. To address this issue, we propose the gathering-scattering scheme to *scatter* meta-knowledge from Meta-RNN to Behavior-RNNs for calibrating the modeling of individual behaviors. The construction of Behavior-RNN in this scheme is shown in Figure 5. Specifically, the hidden state $\mathbf{h}^{\phi}_{t}$ at time t of the behavior sequence $\mathcal{B}^{\phi}_{u}$ depends on the current basket embedding $\mathbf{E}^{\phi}_{u,t}$, its previous hidden state $\mathbf{h}^{\phi}_{t-1}$ and the meta hidden state $\mathbf{h}^{M}_{t}$ of Meta-RNN

$$\mathbf{h}^{\phi}_{t} = f\left(\mathbf{W}^{\phi}_{1}\mathbf{E}_{B^{\phi}_{u,t}} + \mathbf{W}^{\phi}_{2}\mathbf{h}^{\phi}_{t-1} + \mathbf{W}^{\phi}_{3}\mathbf{h}^{M}_{t} + \mathbf{b}^{\phi}\right), \tag{12}$$

where the $\mathbf{W}, \mathbf{b}$ terms are the parameters to be learned. Since $\mathbf{h}^{M}_{t}$ encodes multi-behavior knowledge of the previous baskets from all the behaviors, Equation (12) essentially allows each behavior sequence to absorb the information from other sequences as early as possible. The updating equation for Meta-RNN is the same as specified in Equation (11).

Note that our proposed scheme is different from the functional sharing scheme developed in Meta-LSTMs [5] in two aspects. First, they aim at developing a multi-task learning framework where the meta network and all the basic networks absorb the same input and deliver the outputs for all the tasks simultaneously. In contrast, we focus on modeling different behavior basket sequences separately and extracting the collective knowledge from multiple sequences for one target task. Second, they use a shared meta network to generate the parameters of the task-specific networks, whereas our Behavior-RNNs are parameter-independent among each other. To be more specific, our schemes are used for knowledge transfer, rather than for function-level sharing used in Meta-LSTMs.

5.3 Recurring-Item-Aware Predictor

Let $\mathcal{B}_u^{\Phi}$ denote the set of all the behavior sequences for user u, i.e., $\mathcal{B}_u^{\Phi} = \{\mathcal{B}_u^{\phi}\}_{\phi \in \Phi}$. A simple prediction model for the next basket is to perform the softmax operation over all the items by referring to the multi-behavior knowledge. For instance, let $Pr(v \in V \mid \mathcal{B}_u^{\Phi}) = \text{softmax}(\mathbf{e}_v^{\top} \mathbf{W} \mathbf{h}_T^M)$, where $\mathbf{W}$ denotes a parameter matrix and $\mathbf{h}_T^M$ is the last hidden state of the Meta-RNN. However, according to our observation O2, there are repeated occurrences of items in the multi-behavior sequences. This implies that an item that was previously interacted with any of behaviors is very likely to be purchased in the future. To establish such a biased item distribution for the next basket, we borrow some ideas from the copy mechanism [14, 17] used for sequence generation. The copying mechanism is proposed to copy appropriate input segments during the sequence decoding stage. It realizes the prediction of a word via two modes: *copy* and *generate*, and constructs a mixed probabilistic model for the two modes during prediction. In our problem, we consider $|\Phi| + 1$ modes, namely one **generate-mode** Gen and $|\Phi|$ **repeat-modes** Rep_{ϕ}:

- Gen represents the scenario that v is not included in any previous baskets of the sequences, i.e., $v \notin \bigcup_{\phi \in \Phi} \bigcup_{t=1}^{T} B_{u,t}^{\phi}$.
- Rep_{ϕ} denotes the scenario that v is included in the previous baskets of the sequence $\mathcal{B}_u^{\phi}$, i.e., $v \in \bigcup_{t=1}^{T} B_{u,t}^{\phi}$.

We use separate repeat-modes because *the degree of the repeated occurrences of items is variant to different kinds of behaviors (O2)*. By taking all the modes into account, we define the probability of an item v in the next basket using the following mixed probabilistic function:

$$Pr\left(v \mid \mathcal{B}_u^{\Phi}\right) = p\left(v, \text{Gen} \mid \mathcal{B}_u^{\Phi}\right) + \sum_{\phi \in \Phi} p\left(v, \text{Rep}_{\phi} \mid \mathcal{B}_u^{\Phi}\right). \tag{13}$$

Since the scenarios under different modes are not mutually exclusive, the functions $p(\cdot)$ in Equation (13) are given by

$$p\left(v, \text{Gen} \mid \mathcal{B}_u^{\Phi}\right) = \begin{cases} \frac{1}{Z} e^{\eta_{\text{Gen}}(v)}, & v \notin \bigcup_{\phi \in \Phi} \bigcup_{t=1}^{T} B_{u,t}^{\phi} \\ 0, & \text{otherwise} \end{cases}, \tag{14}$$

$$p\left(v, \text{Rep}_{\phi} \mid \mathcal{B}_u^{\Phi}\right) = \begin{cases} \frac{1}{Z} e^{\eta_{\text{Rep}_{\phi}}(v)}, & v \in \bigcup_{t=1}^{T} B_{u,t}^{\phi} \\ 0, & \text{otherwise} \end{cases}, \tag{15}$$

where $\eta_{\text{Gen}}(\cdot)$ and $\eta_{\text{Rep}_{\phi}}(\cdot)$ are the *score functions* for the respective modes. Let $V' = V - \bigcup_{\phi \in \Phi} \bigcup_{t=1}^{T} B_{u,t}^{\phi}$. It is important to note that $Z = \sum_{v \in V'} e^{\eta_{\text{Gen}}(v)} + \sum_{\phi \in \Phi} \sum_{v \in \bigcup_{t=1}^{T} B_{u,t}^{\phi}} e^{\eta_{\text{Rep}_{\phi}}(v)}$ is a shared normalization term that ensures the validity of the probability function $Pr(v \mid \mathcal{B}_u^{\Phi})$ in Equation (13).

We define the score functions as follows:

$$\eta_{\text{Gen}}(v) = \mathbf{w}_{v,g}^{\top}\mathbf{h}_T^M, \tag{16}$$

$$\eta_{\text{Rep}_\phi}(v) = \mathbf{w}_{v,\phi}^{\top}\mathbf{h}_T^{\phi}, \quad \forall\phi \in \Phi, \tag{17}$$

where $\mathbf{w}_{v,g}$ and $\mathbf{w}_{v,\phi}$ are the item-specific weight vectors for different modes to be learned. The rationality behind the score functions is that the probability of a new item (i.e., never been interacted before) in the next basket is dependent on the encoded multi-behavior knowledge $\mathbf{h}_T^M$, whereas that of a repeatedly interacted item should be conditioned on the behavior-specific information $\mathbf{h}_T^{\phi}$.

5.4 Training and Optimization

Consider a training set $\mathcal{T} = \{(\mathbf{x}_i = \mathcal{B}_u^{\Phi}, \mathbf{y}_i = B_{u,T+1}^{\phi^*})\}_{i=1}^{|\mathcal{T}|}$, where $\mathbf{y}_i = B_{u,T+1}^{\phi^*}$ is the ground-truth next purchased basket given $\mathbf{x}_i$. Recall that the NBR is essentially a multi-label classification problem. Hence, for each training sample $\mathbf{s}_i = (\mathbf{x}_i = \mathcal{B}_u^{\Phi}, \mathbf{y}_i = B_{u,T+1}^{\phi^*})$, we compute the cross-entropy loss

$$\mathcal{L}(\mathbf{s}_i) = -\frac{1}{|\mathbf{y}_i|}\sum_{v \in \mathbf{y}_i} \log Pr(v \mid \mathbf{x}_i), \tag{18}$$

where $|\mathbf{y}_i|$ is the size of the ground-truth next basket and $Pr(\cdot)$ is the predicted probability as defined in Equation (13). To learn from the training set, the overall objective is to minimize the following loss function:

$$\mathcal{L}(\mathcal{T}) = -\sum_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{T}} \frac{1}{|\mathbf{y}_i|}\sum_{v \in \mathbf{y}_i} \log Pr(v \mid \mathbf{x}_i). \tag{19}$$

As the above loss function is fully differentiable and the parameters in our model can be optimized in an end-to-end manner using backpropagation. During online inference, we can compute the probability of each item in the next basket given a user's historical multi-behavior sequences. The final recommendation results are top-K items ranked with the highest probabilities.

6 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of MBN and answer the following research questions:

RQ1. How does MBN perform compared with state-of-the-art recommendation methods?
RQ2. How do the generate- and repeat- modes in the proposed predictor benefit the recommendation performance?
RQ3. What are the effects of context behaviors and Meta-RNN in our proposed methods?
RQ4. How does the time window size affect the performance?
RQ5. How does the model explain its recommendation results?

6.1 Experimental Settings

6.1.1 Datasets. We used two real-world datasets: Tianchi[1] and IJCAI15[2] as described in Section 4. In addition, another multi-behavior dataset JD[3] was adopted to evaluate the performance of our proposed methods. These datasets contain basket sequences with four behaviors: clicking

[1] https://tianchi.aliyun.com/competition/entrance/231522/information.
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=47.
[3] https://jdata.jd.com/html/detail.html?id=8.

Table 2. Statistics of the Two Datasets

Dataset	#users	#items	#baskets			
			C	A	F	Purchase
Tianchi	9,197	17,611	250,452	28,766	14,249	39,513
IJCAI15	30,696	15,084	758,653	1,685	143,620	298,208
JD	8,691	3,552	313,079	8,739	6,174	67,595

(**C**), adding to cart (**A**), tagging with favor (**F**) and purchasing ϕ^*, where purchasing is our target behavior. Similar to the k-core subset requirement in [39, 46], we only consider the users that purchased at least n items and the items that were bought no less than n times. The values of n are set to 2 , 5, and 10 for Tianchi, JD and IJCAI15, respectively. Table 2 provides the details of these datasets after preprocessing.

Following the previous work [32], (i) we split each dataset into the training and test sets according to the cut time. On Tianchi, we use the first 27 days of data for training and the last 3 days of data for the test. On JD, we use the first 67 days of data for training and the last 7 days of data for test. On IJCAI15, we use the first 169 days of data for training and the last 11 days of data for test. (ii) By default, we allow varying-size time windows, i.e., using all the previous baskets for predicting the next purchasing basket. (iii) During testing, we treat each purchasing basket after the cut time as the *next basket* to be predicted, based on all the historical baskets after the cut time of the user. To reproduce our experiment results, We release the codes at https://github.com/gybuay/MBN.

6.1.2 Metrics. Following [32, 39, 46], we choose the top K (i.e., K$\in$ $\{20, 40, 60\}$) items in the ranking list as the recommended basket. We compare the performance of all the approaches using three popular metrics Recall@K, F1-Score@K and NDCG@K [36].

6.1.3 Comparison Methods. We compared our proposed models **MBN-g** and **MBN-gs** using the gathering and gathering-scattering schemes, respectively, with three categories of methods:

Simple baselines:

- **TopK** reports the K most frequent items that appear in the training sequences of the target behavior.
- **PTopK** recommends the K most frequent items in the past purchasing sequence of a particular user. If the total number of items in user's past purchasing sequences is less than K, we supplement with the most frequent items in the training sequences.
- **MTopK** produces the most frequent items in the past multi-behavior sequences of each user. We perform the same supplement as PTopK.

NBR methods:

- **FPMC** [39] method considers the impact of user-item, last item-item and last item-user interactions within the MC framework.
- **DREAM** [46] is a dynamic recurrent model for the NBR based on max pooling and GRU.
- **Beacon** [29] models the pairwise correlations among items. Each basket is encoded by taking into account the relative importance of items and correlations among item pairs.
- **Sets2sets** [21] is the state-of-the-art RNN-based framework that outputs sequential sets based on a sequence of previous sets. It treats baskets as sets and predicts the next basket only. A repeat purchase module is adopted in this method.

— **TIFUKNN** [22] is a state-of-the-art kNN-based method that exploits personalized item frequency information for NBR.

Multi-behavior methods:

— **RLBL** [33] integrates RNN and the Log-BiLinear model over multi-behavior sequential data for prediction.
— **CBS** [28] is a twin network that only utilizes one support behavior. We use clicking as the support behavior, and use purchasing as the target behavior.
— **HUP** [15] proposes a Pyramid RNN and Behavior-LSTM, which utilizes user's historical micro-behaviors and dwell time of behaviors to form user's real-time hierarchical interests. We adapt this method to our datasets without dwell time information for NBR.
— **MKM-SR** [37] is a state-of-the-art session-based recommendation method that constructs knowledge graphs for item attributes and applies GGNN [31] on multi-behavior graphs. We exclude the auxiliary knowledge learning task since our datasets do not have item attributes.
— **Sets2sets-Mul** is an extension of Sets2sets where the number of RNN encoders is set to $|\Phi|$ for different behaviors. The hidden states of $|\Phi|$ RNN encoders are concatenated to predict the next basket.
— **Sets2sets-Mix** is another extension of Set2sets, where all historical multi-behavior interactions are fed to one RNN encoder.

Note that RLBL, CBS, HUP, and MKM-SR were proposed for the next item or session-based recommendation using multi-behavior data, and we adapt them to NBR. Furthermore, we extend the state-of-the-art RNN-based NBR method Sets2sets to multi-behavior sequences and obtain Sets2sets-Mul and Sets2sets-Mix.

6.1.4 Model Configurations. For our proposed model, the dimension of item embeddings is set to 100. By default, we use the gather-scattering scheme in MBN. We implement the RNNs in the multi-sequence encoder by GRUs [9] due to the computational efficiency. The model is trained with the Adam optimizer [25] and we set the learning rate to 0.001 with step decay by 0.99 and the batch size to 32. For the comparison methods, we followed the original articles and used their published codes to tune the hyperparameters with grid search. For FPMC, the factorization dimension is searched from {8, 16, 32, 64, 128}; for DREAM, the dimension of item vector is searched from {10, 20, 40, 80, 160}; for Beacon, the dimensions of embedding and RNN hidden state are both searched from {16, 32, 64, 96, 128}; for Sets2sets, the embedding size is searched from {8, 16, 32, 64, 128}; for TIFUKNN, we searched k from {100, 300, 500, 700}, rb from {0.7, 0.8, 0.9, 1.0}, rg from {0.5, 0.6, 0.7, 0.8, 0.9}, α from {0.5, 0.6, 0.7, 0.8, 0.9}, and m from {5, 6, 7, 8, 9}; for RLBL, we searched the latent dimension from {8, 16, 32, 64, 128}; for CBS, the number of dense units and RNN hidden units are searched from {8, 16, 32, 64, 128} and {32, 64, 128, 192} respectively; for HUP, we searched item embedding size from {30, 60, 120} and behavior embedding size from {5, 10, 20}; for MKM-SR, we searched the graph node embedding size and behavior embedding size from {50, 100, 150}.

6.2 Performance Comparison (RQ1)

Table 3 shows the performance comparison results of various approaches on three datasets. The performances of all the methods are consistent over different metrics. Several observations are obtained. **First**, TopK always reports the worst results due to its ignorance of personalized historical purchasing information. All the other methods utilize the items previously purchased or interacted by the user and achieve better recommendation performance. **Second**, PTopK and MTopK surprisingly outperform FPMC and DREAM over all the metrics and their advantages are more significant on Tianchi than on IJCAI15. Our empirical analysis has shown that the proportion of the

Table 3. Performance Comparison Results for NBR (RQ1)

Method	Recall@20	Recall@40	Recall@60	F1@20	F1@40	F1@60	NDCG@20	NDCG@40	NDCG@60
Tianchi									
TopK	0.712	1.415	1.852	0.080	0.081	0.074	0.242	0.389	0.472
PTopK	19.410	20.420	20.862	2.099	1.155	0.797	11.999	12.220	12.302
MTopK	33.219	37.903	39.673	3.606	2.153	1.540	18.496	19.509	19.848
FPMC	6.028	6.792	7.370	0.354	0.269	0.218	3.359	3.500	3.594
DREAM	2.914	3.278	3.687	0.159	0.120	0.104	1.515	1.579	1.651
Beacon	9.537	11.464	13.110	1.604	1.187	0.549	6.116	7.643	9.024
Sets2sets	22.506	23.532	23.707	2.428	1.320	0.901	13.479	13.702	13.738
TIFUKNN	24.377	24.727	24.766	2.652	1.400	0.946	16.707	16.790	16.798
RLBL	15.774	17.266	18.131	1.934	0.897	0.749	11.417	11.985	12.112
CBS	29.913	32.609	36.392	2.853	1.675	1.249	16.718	17.209	19.728
HUP	26.005	30.167	31.599	2.648	1.629	1.118	13.573	15.385	15.874
MKM-SR	30.027	33.553	35.657	3.243	2.101	1.651	18.336	19.141	19.534
Sets2sets-Mul	22.812	28.904	31.965	2.403	1.511	1.168	12.103	14.868	15.942
Sets2sets-Mix	24.310	40.593	48.831	2.499	2.251	1.715	12.779	16.948	18.489
MBN-g	38.416	50.393	55.056	4.178	2.905	2.161	16.652	19.254	20.129
MBN-gs	**39.956**	**51.687**	**55.847**	**4.351**	**2.962**	**2.192**	**18.822**	**21.356**	**22.149**
IJCAI15									
TopK	0.505	1.238	1.780	0.096	0.135	0.135	0.193	0.383	0.503
PTopK	3.708	4.482	4.933	0.667	0.441	0.338	2.118	2.311	2.412
MTopK	8.064	11.831	13.973	1.675	1.355	1.102	4.264	5.255	5.738
FPMC	1.748	2.237	2.647	0.304	0.211	0.171	1.071	1.191	1.277
DREAM	1.158	1.874	2.393	0.202	0.174	0.154	0.613	0.784	0.894
Beacon	2.514	4.288	5.935	0.602	0.572	0.539	1.096	1.584	1.966
Sets2sets	2.828	3.811	4.620	0.510	0.368	0.306	1.745	1.983	2.156
TIFUKNN	4.741	5.128	5.494	0.815	0.476	0.350	3.515	3.610	3.689
RLBL	7.315	9.345	10.720	1.154	0.803	0.541	4.715	5.203	5.496
CBS	8.152	12.176	15.017	0.776	0.594	0.492	3.656	4.475	4.974
HUP	12.072	14.223	15.569	1.860	1.002	0.614	6.021	6.550	6.839
MKM-SR	13.274	16.114	18.751	3.117	1.495	0.828	6.511	7.229	7.593
Sets2sets-Mul	3.915	7.506	10.363	1.277	0.893	0.530	3.624	5.352	6.487
Sets2sets-Mix	4.390	14.907	21.369	1.298	0.952	0.758	3.727	6.381	8.279
MBN-g	**16.193**	**25.360**	**30.055**	3.726	3.234	**2.639**	**7.939**	**10.436**	**11.531**
MBN-gs	16.125	25.288	29.881	**3.738**	**3.238**	2.634	7.816	10.310	11.381
JD									
TopK	5.451	7.202	9.704	0.684	0.499	0.504	4.234	4.630	5.163
PTopK	25.636	26.892	28.527	3.304	1.826	1.327	20.878	21.157	21.484
MTopK	50.905	58.867	60.492	7.648	4.414	3.072	32.571	33.732	34.044
FPMC	27.218	30.787	32.968	3.881	2.432	1.816	20.757	21.665	22.145
DREAM	25.265	31.040	33.247	3.602	2.001	1.859	11.087	12.796	13.952
Beacon	13.864	24.007	32.622	1.894	2.052	1.806	8.779	11.269	12.923
Sets2sets	32.634	34.038	35.321	4.448	2.469	1.810	24.712	25.051	25.374
TIFUKNN	35.103	40.372	42.011	5.052	3.232	2.320	26.168	27.522	27.887
RLBL	29.752	33.572	36.734	4.784	2.319	1.770	21.982	22.883	23.657
CBS	44.083	50.193	54.601	6.198	2.448	1.790	24.732	25.991	26.773
HUP	42.138	47.329	52.839	5.799	3.264	1.123	25.057	26.408	27.480
MKM-SR	47.342	52.904	56.278	7.421	4.213	2.071	27.207	29.076	30.623
Sets2sets-Mul	49.722	57.032	60.079	7.976	4.119	2.082	27.222	30.542	31.413
Sets2sets-Mix	50.851	59.644	64.394	7.759	4.592	2.116	30.272	32.848	33.910
MBN-g	52.249	61.756	64.980	8.031	5.204	3.680	34.479	37.004	37.6498
MBN-gs	**52.404**	**62.514**	**65.966**	**8.106**	**5.242**	**3.712**	**34.751**	**37.056**	**37.661**

repeatedly interacted items on Tianchi is larger than that on IJCAI. As users are very likely to purchase an item that was interacted before, it is beneficial to explicitly model the frequently interacted items for prediction. Sets2sets realizes the frequently appearing items in the past and works well among the non-statistical methods. TIFUKNN explicitly utilizes the personalized

Table 4. Performance Comparison Results on IJCAI15 Using Personalized Length $K = k^*$ (RQ1)

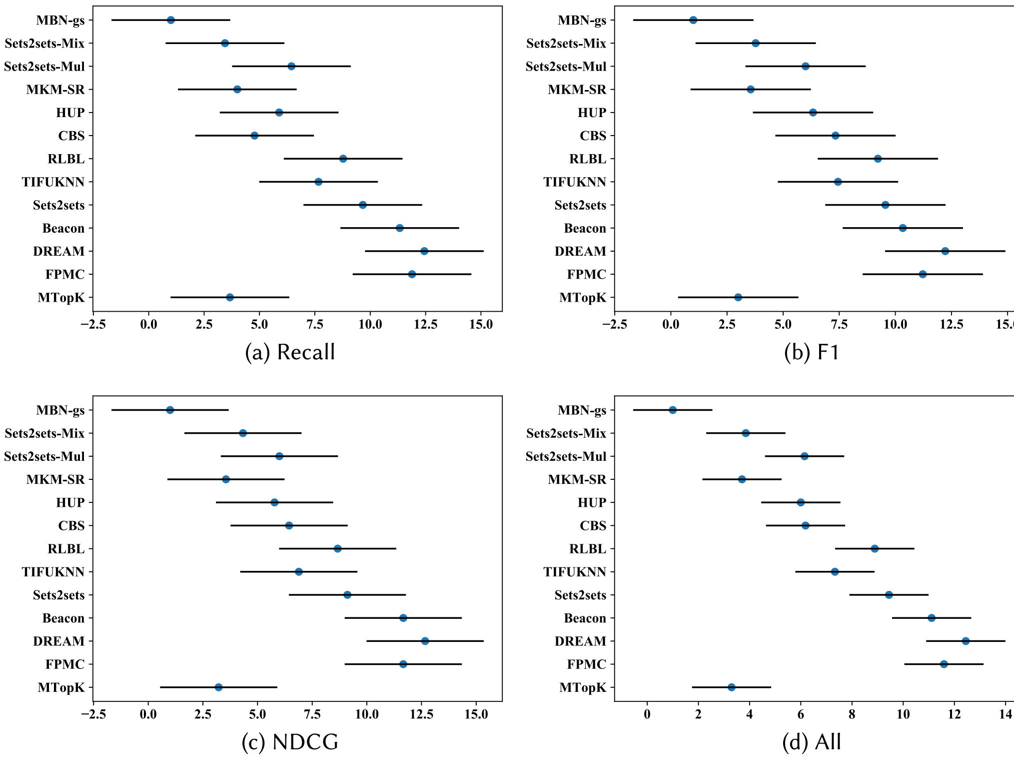
	MTopK	DREAM	Beacon	S2S	TIFUKNN	CBS	HUP	MKM-SR	S2S-Mul	S2S-Mix	MBN-gs
Recall	2.598	0.872	1.343	1.578	2.043	2.326	2.441	2.758	1.923	1.936	**3.468**
F1	1.767	0.334	0.735	1.044	1.261	1.409	1.421	1.602	1.142	1.198	**2.706**
NDCG	2.196	0.516	1.128	1.571	1.661	1.838	1.908	2.093	1.558	1.576	**2.996**

item frequency information with a repeated purchase component to recommend frequently interacted items and performs better than FPMC, DREAM, Beacon, and Sets2sets. For multi-behavior methods, MKM-SR achieves the best performance considering different transition patterns between items and behaviors. Interestingly, Set2sets-Multi outperforms Sets2sets-Mix on all the cases. This indicates that the ignorance of inter-behavior correlations would hurt the final performance. **Third**, our proposed MBN-g and MBN-gs outperform all the baselines over these datasets. On Tianchi, MBN-gs achieves about 7.0%–11.1%, 0.5%–1.1%, and 0.5%–2.6% absolute improvements on Recall, F1-score, and NDCG, respectively, compared with the second best method MKM-SR/Sets2sets-Mix. On JD, MBN-gs achieves about 1.6%–2.9%, 0.2%–1.6%, and 3.7%–4.5% absolute improvements on Recall, F1-score and NDCG ,respectively, compared with the second best method Sets2sets-Mix. The performance improvements on IJCAI15 of our proposed methods are more significant. Furthermore, when K is 40 or 60, MBN-gs achieves over 51% and 25% Recall values on Tianchi and IJCAI15, respectively. Both values are respectively larger than the average proportions of the first-time-bought items in the context-behavior basket sequences, i.e., 40% and 12% in Figure 2. This implies our proposed methods are able to recommend new items that have not been purchased by the user before. **Fourth**, the gathering-scattering scheme consistently performs better than the gathering scheme on Tianchi and JD. However, MBN-gs performs slightly worse than MBN-g in some cases on IJCAI15. This is mainly because IJCAI15 involves a large number of purchasing baskets to well disclose users' purchasing interests and the advantages of meta-knowledge from context behaviors are thus limited. Without abundant historical purchasing records, it is still beneficial to transfer knowledge among multi-behavior sequences as early as possible.

Considering each user may have a typical basket length, we have used personalized basket length $K = k^*$ r predicting the next basket for different users, where k^* is set to the average basket length of a particular user. Considering each user has his own preferred basket length, we give the comparison results of personalized basket length of the predicted next basket using the average basket length of each user. Table 4 shows the performance comparison results on IJCAI15 and the same conclusion can be made on the other datasets. We can see that MBN-gs outperforms the baselines under this personalized basket length setting in Recall, F1-score, and NDCG.

In addition, we conducted the non-parametric Friedman test and the post-hoc Nemenyi test with Critical Difference [10], to statistically validate our evaluation results on these datasets. The Friedman test was rejected for all three metrics with a p-value of 0.05, indicating that all comparison methods got different performance on three datasets. Taking post-hoc Nemenyi test of Recall results as an example, we consider the results of Recall@K with different values of K over all the three datasets to conduct the test. The post-hoc Nemenyi test results across three datasets are shown in Figure 6, "All" means that results on all the metrics are used to carry out the post-hoc Nemenyi test. The post-hoc test results show that our MBN-gs method outperforms other comparison methods under the same confidence interval.

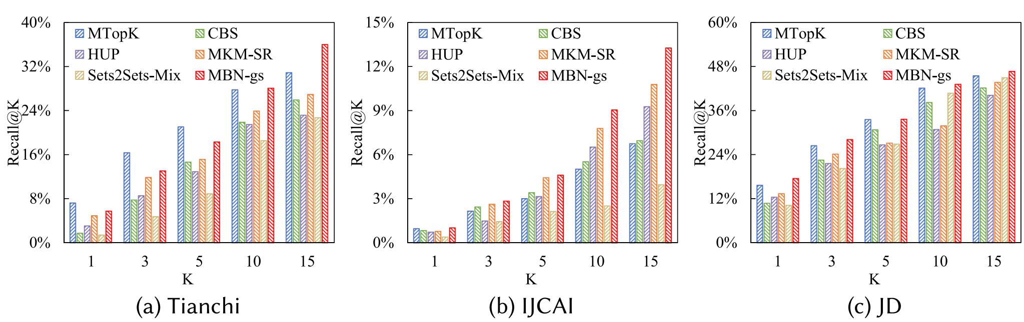
Finally, we compare the performance of different approaches for K in {1,3,5,10,15}. We reported the results of five baselines that achieves the highest Recall performance, i.e., MTopK, MKM-SR, Sets2sets-Mix, HUP, and CBS. The trends of F1-score and NDCG results are similar, and we eliminate the results due to redundancy. As shown in Figure 7,our method MBN-gs outperforms the baselines on all the three datasets, except for MTopK with $K \in \{1, 3, 5\}$ on the Tianchi dataset. Our



(a) Recall (b) F1

(c) NDCG (d) All

Fig. 6. Results of the post-hoc Nemenyi test with Critical Difference plots (RQ1).



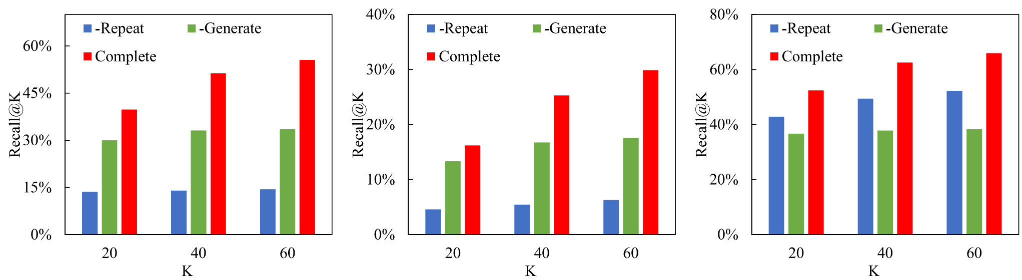
(a) Tianchi (b) IJCAI (c) JD

Fig. 7. Performance comparison results of Recall with smaller K (RQ1).

observations in Section 4 imply that there is a large proportion of the repeatedly interacted items on Tianchi. MTopK leverages this strong frequency information and hence achieves good performance for small values of K on Tianchi. When K becomes larger, simply producing the most frequent items is insufficient to recommend the satisfying items. In more general cases, our method shows its superiority by modeling collective knowledge from multi-behavior basket sequences effectively.

6.3 Effects of Repeat-/Generate- Modes (RQ2)

We investigate the effects of the repeat- and generate- modes in the predictor of our model. As shown in Figure 8, "-Repeat" and "-Generate" represent the removal of the $|\Phi|$ repeat-modes and



(a) Tianchi (b) IJCAI15 (c) JD

Fig. 8. Effects of different modes in the predictor (RQ2).

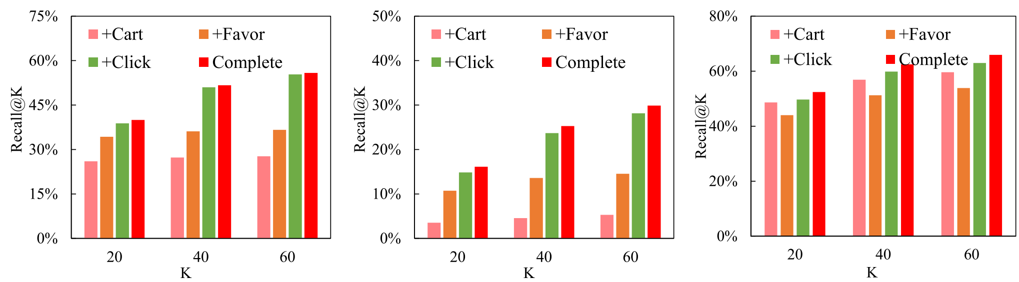
Table 5. Effects of All the Context Behaviors (RQ3)

Dataset	Tianchi			IJCAI15			JD		
Recall@K	**20**	**40**	**60**	**20**	**40**	**60**	**20**	**40**	**60**
MBN-P	17.110	18.503	19.051	3.540	4.579	5.358	41.623	49.852	51.422
MBN-gs	**39.956**	**51.687**	**55.847**	**16.125**	**25.288**	**29.881**	**52.404**	**62.514**	**65.966**
F1@K	**20**	**40**	**60**	**20**	**40**	**60**	**20**	**40**	**60**
MBN-P	2.433	1.283	0.872	0.789	0.480	0.341	6.390	4.234	2.960
MBN-gs	**4.333**	**2.941**	**2.179**	**3.788**	**3.271**	**2.652**	**8.106**	**5.242**	**3.712**
NDCG@K	**20**	**40**	**60**	**20**	**40**	**60**	**20**	**40**	**60**
MBN-P	9.582	9.883	9.984	1.848	2.105	2.273	30.294	32.372	32.675
MBN-gs	**18.822**	**21.356**	**22.149**	**7.816**	**10.310**	**11.381**	**34.751**	**37.056**	**37.661**

the generate-mode in predicting the item probabilities in the next basket (see the two terms in Equation (13)), respectively. We observe that our complete model MBN-gs achieves the highest Recall values in all the cases. Removing the score functions for the repeat-modes (i.e., -Repeat) degrades the model performance by 65% in Recall, demonstrating the significance of explicitly modeling previous interacted items with context behaviors for NBR. Removing the score function for the generate-mode (i.e., -Generate) also hurts the performance, and the gap between -Generate and MBN-gs becomes larger as K increases. This indicates the importance of recommending new items that have not been interacted by the user for larger K, which is consistent with the evolving pattern in user interests. We notice that the performance of MBN-gs is not a simple addition over that of -Repeat and -Generate. The combination of the two kinds of modes makes an extra contribution to the overall recommendation performance. Similar observations are obtained in F1-score and NDCG, which are omitted due to redundancy.

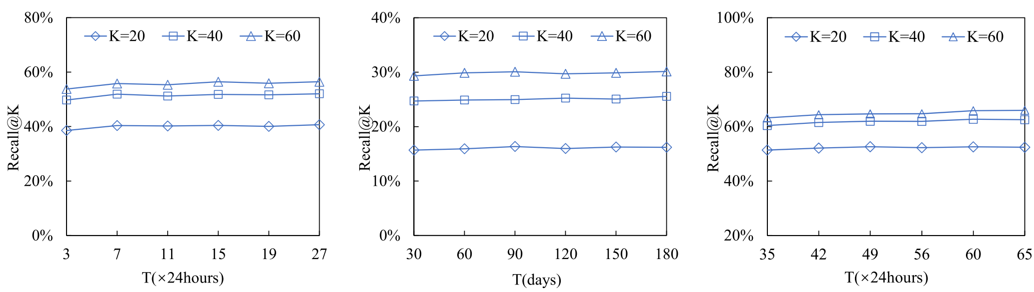
6.4 Effects of Context Behaviors and Meta-RNN (RQ3)

We first evaluate the effects of context behaviors on the recommendation performance. First, we use "MBN-P" to represent the case that removes all the context behavior sequences from the input. Table 5 provides the Recall, NDCG, and F1-score results of MBN-P and MBN-gs. We can see that MBN-gs outperforms MBN-P significantly over all the cases. Specifically, MBN-gs achieves 22.8%–36.8%, 12.6%–24.5%, and 10.8%–14.5% higher Recall values than MBN-P on Tianchi, IJCAI15 and JD, respectively. This verifies the significance of utilizing context behavior data for the NBR. As K increases, the performance gap between the two methods becomes even larger on both datasets. Second, we use "+Cart", "+Click", and "+Favor" to denote the cases that include individual context behavior data. Figure 9 provides the Recall results, whereas the relative performance trends in



(a) Tianchi (b) IJCAI15 (c) JD

Fig. 9. Effects of individual context behavior (RQ3).



(a) Tianchi (b) IJCAI15 (c) JD

Fig. 10. Effects of time window size T (RQ4).

F1-score and NDCG are the same. It is easy to see that different context behaviors have different influence on the recommendation performance. Among all the context behaviors, clicking always has the highest contribution to the recommendation performance, followed by favoring, and finally adding to the cart. This is consistent with our observation O2 that there is a large overlap between the items to be purchased in the next basket and those being clicked before. Recall in Table 2, the clicking baskets account for over 80% of total baskets on both datasets, and hence the performance of using clicking data is close to that of our complete model. At last, the best results on both datasets can always be achieved by taking all the behavior sequences into consideration.

We next investigate the effectiveness of Meta-RNN in our model. We ignore all the repeat-modes in the predictor and examine the pure performance gain of using Meta-RNN. The resultant "MBN-gs-Repeat" model predicts the next basket based on the concatenation of $\mathbf{h}_T^M$ and $\mathbf{h}_T^{\phi^*}$. We also construct two basic models involving only $|\Phi|$ Behavior-RNNs: (i) "Concat" predicts the next basket based on the concatenation of the last hidden states of all the Behavior-RNNs; (ii) "SelfAttn" gathers the last hidden states of $|\Phi|$ Behavior-RNNs using the self-attention mechanism [41]. According to Table 6, we can see that "SelfAttn" performs slightly better than "Concat" on Tianchi and IJCAI, but the performance on JD is the opposite. Overall, "MBN-gs-Repeat" achieves 2.7%–63%, 2.3%–110%, and 3.3%–57% relative improvements against the two basic models in Recall, F1-score and NDCG, respectively. This confirms the positive effects of Meta-RNN in capturing meta-knowledge from multi-behavior sequences.

6.5 Effects of Time Window Size (RQ4)

In this part, we investigate the effects of different time window sizes T. Figure 10 provides the Recall results by varying T. The results on F1-score and NDCG have similar trends. It is interesting to

Table 6. Effects of Meta-RNN (RQ3)

Dataset	Tianchi			IJCAI15			JD		
Recall@K	**20**	**40**	**60**	**20**	**40**	**60**	**20**	**40**	**60**
Concat	8.103	9.501	9.824	2.228	3.023	3.719	40.598	47.849	50.832
SelfAttn	8.310	9.275	10.004	2.863	3.471	4.159	39.451	47.303	49.875
MBN-gs-Repeat	**13.617**	**14.021**	**14.416**	**4.594**	**5.461**	**6.231**	**42.854**	**49.394**	**52.227**
F1@K	**20**	**40**	**60**	**20**	**40**	**60**	**20**	**40**	**60**
Concat	0.507	0.369	0.295	0.371	0.283	0.240	6.429	4.192	3.032
SelfAttn	0.523	0.348	0.313	0.401	0.299	0.268	6.173	4.081	2.929
MBN-gs-Repeat	**1.102**	**0.629**	**0.445**	**0.636**	**0.482**	**0.400**	**6.728**	**4.293**	**3.109**
NDCG@K	**20**	**40**	**60**	**20**	**40**	**60**	**20**	**40**	**60**
Concat	5.891	6.025	6.122	1.447	1.647	1.795	28.560	30.455	31.028
SelfAttn	5.934	6.058	6.239	1.586	1.720	1.902	27.026	28.998	29.521
MBN-gs-Repeat	**8.162**	**8.236**	**8.304**	**2.501**	**2.696**	**2.856**	**29.776**	**31.459**	**32.436**

Table 7. A Case Study of the Probability Proportion of Different Modes in MBN-gs while Predicting the Next Basket (RQ5)

Items in $\mathcal{B}_t^C$	Items in $\mathcal{B}_t^A$	Items in $\mathcal{B}_t^F$	Items in $\mathcal{B}_t^P$	Items in $\mathcal{B}_{t+1}^P$	$v \in \mathcal{B}_{t+1}^P$	Gen	Rep$_C$	Rep$_A$	Rep$_F$	Rep$_P$
{119, 689, 2117, 174, 134, 1436, 100, 1680, 2119, 3277, 2798, 1437}	{2117, 134}	{119, 1436, 1680}	{119, 3243, 2412, 851, 3277, 2269}	{119, 103, 3277, 2361, 1787}	119	7.6%	23.8%	2.6%	36.4%	29.6%
					103	75.3%	7.5%	3.8%	2.7%	10.7%
					3277	16.5%	35.3%	3.1%	4.2%	40.9%
					2361	86.8%	6.3%	1.7%	2.8%	2.4%
					1787	71.4%	12.1%	4.8%	4.5%	7.2%

see that the performance of MBN-gs is quite stable with different values of T. While examining longer historical basket sequences can benefit the recommendation accuracy, the improvement gain is insignificant. There are two possible reasons: (i) the multi-behavior sequences within a small time window encode sufficient insights on the next basket; (ii) larger time window sizes make the sparsity issue of multi-behavior sequences more severe, which may compromise the performance. While we considered all the historical interactions in the experiments, our proposed method can perform consistently well using a small T, for the efficiency concern.

6.6 Interpretability (RQ5)

Lastly, in order to better explain the next basket predicted by our method, Table 7 reports a case of one user's next basket predicted by MBN-gs. We provide the proportion of the probability given by generate-mode (Gen) $p(v, \text{Gen} \mid \mathcal{B}_u^{\Phi})$ and repeat-modes (Rep) $\sum_{\phi \in \Phi} p(v, \text{Rep}_{\phi} \mid \mathcal{B}_u^{\Phi})$ mentioned in Equation (13). $\mathcal{B}_t^{\phi}$ in Table 7 means behavior-specific interaction basket (C for Clicking, F for Favor, A for Adding to cart, and P for Purchasing) in time step t, Rep$_{\phi}$ is behavior-specific repeat-mode, and the items in each basket are associated with items ID. $\mathcal{B}_{t+1}^P$ is the next purchase basket we need to predict, and the proportion of the probability for item v contributed by Gen and Rep are reported in Table 7. From Table 7, we can see that in this case, the Gen part contributed most of the scores in the final probability when the item has no occurrence before, such as {103, 2361, 1781} in this basket. However, the Rep part plays an important role if the item is a recurrent item that previously interacted via context behaviors, or purchase repeatedly, just like {119, 3277} in this case. This suggests that our method computes the significance of each behavior for the user by the votes from different behaviors for a particular item. Meanwhile, for those new items that have no interaction before, Gen accounted for most of the final probability.

7 CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this article, we proposed to exploit collective knowledge from multi-behavior basket sequences for NBR. We conducted a comprehensive empirical study to reveal the characteristics of multi-behavior data and offer useful insights on the item correlations among sequences and the recurring item pattern. We introduced a novel neural approach named MBN to learn basket representation, capture meta-knowledge from multi-behavior sequences with two schemes, and predict the next basket by taking into consideration the repeated occurrences of items. Extensive experiments conducted over three real-world multi-behavior datasets confirmed the superiority of our proposed MBN on the NBR, compared with various state-of-the-art methods.

7.2 Future Work

Our work can be extended in multiple directions. In this article, we recommend the next basket mainly based on the representations of the meta-knowledge and the last baskets in different behavioral sequences. How to leverage the information from all the previous baskets is a kind of future work. Moreover, the items to be purchased in the next basket may be correlated with each other. How to model personalized item correlations within baskets is another challenge.

ACKNOWLEDGMENTS

We thank all the reviewers for their detailed comments that improve our paper.

REFERENCES

[1] Charu C. Aggarwal. 2016. *Recommender Systems - The Textbook*. Springer. 1–498.

[2] Charu C. Aggarwal and Srinivasan Parthasarathy. 2001. Mining massively incomplete data sets by conceptual reconstruction. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 227–232.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations.*

[4] 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 *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1201–1204.

[5] Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang. 2018. Meta multi-task learning for sequence modeling. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence.* 5070–5077.

[6] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 108–116.

[7] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu. 2021. Dual-embedding based deep latent factor models for recommendation. *ACM Transactions on Knowledge Discovery and Data* 15, 5 (2021), 85:1–85:24.

[8] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. DELF: A dual-embedding based deep latent factor model for recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence.* 3329–3335.

[9] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, 8th Workshop on Syntax, Semantics and Structure in Statistical Translation.* 103–111.

[10] Janez Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

[11] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An improved sampler for bayesian personalized ranking by leveraging view data. In *Companion Proceedings of the The Web Conference 2018.* 13–14.

[12] Jingtao Ding, Guanghui Yu, Xiangnan He, Yuhan Quan, Yong Li, Tat-Seng Chua, Depeng Jin, and Jiajie Yu. 2018. Improving implicit recommender systems with view data. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence.* 3343–3349.

[13] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural multi-task recommendation from multi-behavior data. In *Proceedings of the 35th IEEE International Conference on Data Engineering*. 1554–1557.

[14] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 1631–1640.

[15] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 223–231.

[16] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. 2018. Personalized market basket prediction with temporal annotated recurring sequences. *IEEE Transactions on Knowledge and Data Engineering* 31, 11 (2018), 2151–2163.

[17] Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 199–208.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations*.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[21] Haoji Hu and Xiangnan He. 2019. Sets2Sets: Learning from sequential sets with neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1491–1499.

[22] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1071–1080.

[23] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–668.

[24] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining*. 197–206.

[25] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

[26] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*. 173–182.

[27] Duc-Trong Le, Hady Wirawan Lauw, and Yuan Fang. 2017. Basket-sensitive personalized item recommendation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2060–2066.

[28] Duc-Trong Le, Hady W. Lauw, and Yuan Fang. 2018. Modeling contemporaneous basket sequences with twin networks for next-item recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3414–3420.

[29] Duc-Trong Le, Hady W. Lauw, and Yuan Fang. 2019. Correlation-sensitive next-basket recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2808–2814.

[30] Ranzhen Li, Yanyan Shen, and Yanmin Zhu. 2018. Next point-of-interest recommendation with temporal and multi-level context attention. In *Proceedings of the International Conference on Data Mining*. 1110–1115.

[31] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated graph sequence neural Networks. In *Proceedings of the 4th International Conference on Learning Representations*.

[32] 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 *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1734–1743.

[33] Qiang Liu, Shu Wu, and Liang Wang. 2017. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1254–1267.

[34] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1831–1839.

[35] Babak Loni, Roberto Pagano, Martha Larson, and Alan Hanjalic. 2016. Bayesian personalized ranking with multi-channel user feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 361–364.

[36] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[37] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1091–1100.

[38] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*, Vol. 2007. 5–8.

[39] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 811–820.

[40] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 650–658.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30, 5998–6008.

[42] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 86–94.

[43] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 403–412.

[44] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 6332–6338.

[45] Lianghao Xia, Y. Xu, C. Huang, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

[46] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 729–732.

[47] Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex graph neural networks for multi-behavior recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2313–2316.

[48] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H. Chi. 2015. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*. 1406–1416.

[49] Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. 2018. Micro behaviors: A new perspective in e-commerce recommender systems. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 727–735.

Received April 2021; revised September 2021; accepted November 2021