A Personalized Neighborhood-based Model for Within-basket Recommendation in Grocery Shopping

Mozhdeh Ariannezhad
AIRLab, University of Amsterdam
Amsterdam, The Netherlands
m.ariannezhad@uva.nl

Ming Li
University of Amsterdam
Amsterdam, The Netherlands
m.li@uva.nl

Sebastian Schelter
University of Amsterdam
Amsterdam, The Netherlands
s.schelter@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

ABSTRACT

Users of online shopping platforms typically purchase multiple items at a time in the form of a shopping *basket*. Personalized within-basket recommendation is the task of recommending items to complete an incomplete basket during a shopping session. In contrast to the related task of session-based recommendation, where the goal is to complete an ongoing anonymous session, we have access to the shopping history of the user in within-basket recommendation. Previous studies have shown the superiority of neighborhood-based models for session-based recommendation and the importance of personal history in the grocery shopping domain. But their applicability in within-basket recommendation remains unexplored.

We propose PerNIR, a neighborhood-based model that explicitly models the personal history of users for within-basket recommendation in grocery shopping. The main novelty of PerNIR is in modeling the short-term interests of users, which are represented by the current basket, as well as their long-term interest, which is reflected in their purchasing history. In addition to the personal history, user neighbors are used to capture the collaborative purchase behavior. We evaluate PerNIR on two public and proprietary datasets. The experimental results show that it outperforms 10 state-of-the-art competitors with a significant margin, i.e., with gains of more than 12% in terms of hit rate over the second best performing approach. Additionally, we showcase an optimized implementation of our method, which computes recommendations fast enough for real-world production scenarios.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

Within-basket recommendation; Nearest neighbors; Grocery shopping

WSDM '23, February 27–March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570417

ACM Reference Format:
Mozhdeh Ariannezhad, Ming Li, Sebastian Schelter, and Maarten de Rijke. 2023. A Personalized Neighborhood-based Model for Within-basket Recommendation in Grocery Shopping. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27–March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570417

1 INTRODUCTION

Recommender systems in retail help users to find the items that they need from large inventories in different domains [30]. In many shopping scenarios, such as grocery shopping, users purchase multiple items in a single transaction, which corresponds to a shopping *basket*. Within-basket recommendation is defined as recommending items for an incomplete shopping basket, which can reduce the burden on users to explore the inventory proactively, and result in shorter shopping times [16, 20]. This problem is relevant in both online and offline shopping. For an online platform, items can be recommended based on user activity and items that have previously been added to a basket. In brick-and-mortar grocery stores, RFID-tagged items and smart shopping carts allow real-time recommendation of items based on a user's smart cart [13].

The characteristics of recommendation in grocery shopping have been studied extensively in a closely related line of work, namely next basket recommendation [2, 7, 17]. These studies reveal the importance of the personal purchase history in grocery shopping. In particular, users shop for grocery items repeatedly and on a regular basis; grocery items have a short life-time and are repurchased frequently by the same user. Previous work on within-basket recommendation mostly focuses on learning a representation for the incomplete basket using factorization machines [13, 16], product embeddings [30], or graph neural networks [19, 20]. However, no prior work considers explicitly modeling personal preferences.

Recent studies show that nearest neighbor methods provide state-of-the-art performance in different recommendation settings, such as session-based recommendation [21, 23], next basket recommendation [7, 17], and session-aware recommendation [12]. Moreover, neighborhood-based recommendations are transparent and explainable [23]. Recent work on real-world recommender systems show that nearest neighborhood-based models scale to industry workloads [11]; the training time required for neural approaches has been found to be at least an order of magnitude larger than neighbor-based approaches [10].

We propose PerNIR, a *personalized nearest neighbor-based model for within-basket recommendation* that explicitly considers users' personal preferences (Section 4). PerNIR has two main components: (i) a personal component that explicitly models the personal preference of a user, and (ii) a collaborative component, which leverages the neighboring users for scoring candidate items. In both components, items in the current incomplete basket are used as signals to find candidate items for recommendation. The main novelty of PerNIR is that it models the long-term and short-term interests of users at the same time in the personal component. The personal component uses the historical purchases to assign scores to items that the user has previously purchased, and leverages the items in the current basket as signals for the short-term interests. The collaborative component calculates how neighboring users would score the candidate items, given the items in the user's current basket. The personal and collaborative item scores are aggregated to make up the final recommendation scores. Additionally, we provide an optimized vectorized implementation of our proposed method in Section 5, which precomputes static parts of our models on a per user-basis and thereby provides low-latency inference performance.

In Sections 6 and 7, we conduct experiments on two public and private grocery shopping datasets to evaluate PerNIR. We compare PerNIR against 10 state-of-the-art baselines and find that it significantly outperforms all, with gains of over 12% in terms of hit rate over the second best performing approach. We conduct an extensive ablation study with nine model variations as well as a hyper-parameter sensitivity experiment. We analyze the performance along different dimensions, i.e., characteristics of the users' historical purchase behavior and the current incomplete basket. Moreover, we study the inference efficiency of PerNIR to demonstrate its effectiveness under real-world latency requirements.

2 PROBLEM FORMULATION

The goal of the within-basket recommendation task is to recommend the next item that a user would add to their current incomplete basket, based on the items that are currently in the basket and the history of the items that they have purchased in the past [16, 19, 20]. One of the main use cases for the within-basket recommendation task is grocery shopping, which is the focus of this paper. In grocery shopping, users tend to purchase multiple items at a time, which corresponds to a shopping basket [2, 7].

A basket is a list of items defined as $\mathbf{b} = [x_0, x_1, \ldots, x_t]$, where $x_i \in I$ denotes an item from a set of items I, and x_i is the i-th item that the user has added to the basket $\mathbf{b}$. The sequence $\mathbf{B}^u = [\mathbf{b}_0^u, \mathbf{b}_1^u, \ldots, \mathbf{b}_n^u]$ denotes the history of baskets for a user $u \in U$. $\mathbf{b}_i^u$ is the i-th basket in the history of user u, and U is the set of all users. The goal is to predict the next item x_{t+1} to be added to the current incomplete basket $\mathbf{b}_{n+1}^u = [x_0, x_1, \ldots, x_t]$. For the basket history $\mathbf{B}^u$ and the new basket $\mathbf{b}_{n+1}^u$, the recommendation model assigns a score to all items $x_i \in X$, and the top-n items are returned as the candidates for the next item to be added to the basket.

3 RELATED WORK

Our work is related to several lines of work.

Within-basket recommendation. Although less studied than other forms and definitions of the recommendation task, there are a few papers that address the exact same problem as ours in the literature. As some of the first authors active in this area, Le et al. [13] propose a basket-sensitive factorization machine, which models the recommendation as a function of four associations between a user, a target item, and the items in a basket. Focused on the grocery shopping domain, Wan et al. [30] learn product embeddings using the co-occurrence of items in baskets and further use the learned embeddings for in-basket and next basket recommendation. Graph-based neural networks have also been studied in work associated with Walmart Labs, again focused on grocery [19, 20]. More recently, Li et al. [16] have proposed a deep learning-based model (DBFM, deep basket-sensitive factorization machine) to address the task. However, the order of items in a basket is ignored in this work.

Our paper resembles the papers listed in terms of problem definition, domain, and datasets used. What we add is a highly effective, low-latency neighborhood-based model with a focus on personal history for within-basket recommendation in grocery shopping.

Session-based recommendation. A well-studied recommendation scenario is session-based recommendation (SBR), which refers to the task of recommending items for the next interactions in a given ongoing anonymous session. Various neural models have been proposed for the task [6, 15, 18, 24, 31, 32], as well as neighborhood-based models [5, 9]. Recent studies have revealed that neighborhood-based models, despite their sometimes simple nature, often perform equally well as, or even outperform, conceptually and computationally more complex deep neural models [21–23].

We build on successful ideas from neighborhood-based models in SBR in our work. However, in contrast to SBR, the identity of the user with an ongoing basket is assumed to be known for within-basket recommendation, which makes our problem space and solution different from existing work.

Personalized session-based recommendation. Analogous to within-basket recommendation, the goal in this scenario is to recommend items for an ongoing session, while the recommender system is aware of the user identifier (and the corresponding user history). Neural recommendation models are the dominant approach in this area, where recurrent neural networks [8, 26, 28, 29], attention networks [33], and graph neural networks [25, 35] have been proposed. A surprising recent study on personalized session-based recommender systems has revealed that such neural models are not better than approaches that do not use users' personal history, and extensions of neighborhood-based models proposed for SBR consistently outperform recent neural techniques [12].

None of these publications consider the grocery shopping domain for recommendation, which has been shown to have specific characteristics, such as the importance of a user's personal history [2, 7]. We focus on this domain, and propose a neighborhood-based model that is designed for within-basket recommendation.

Next basket recommendation. In this scenario, the goal is to recommend a list of items to a user, based on their history of previously purchased shopping baskets. This definition is similar to within-basket recommendation, as the history is in the form of baskets, and is mostly studied in the grocery shopping domain. Similar to SBR, neighborhood-based methods [4, 7] and neural methods [2, 14, 27, 34] have both been examined and show strong performance in next basket recommendation (NBR) [17]. In within-basket recommendation, however, the goal is to recommend items to *complete* an incomplete basket, which already contains some items that can be utilized for recommendation.

In this work, we make use of ideas that have been shown to be effective in NBR for grocery shopping and adapt them to our problem setting by using both the personal history and the current basket for computing the recommendation scores.

4 MODEL

Next, we introduce PerNIR, our personalized neighborhood-based model for within-basket recommendation in grocery shopping. We introduce the personal scoring function, the neighborhood-based scoring, as well as our final scoring function, which combines them.

4.1 Personal scoring function

Recent studies have shown the importance of users' personal history in grocery shopping [2, 7, 17]. However, they consider the next basket recommendation task as their main goal, where there is no information about a current basket $\mathbf{b}^u_{n+1}$. For simplicity, we use the notation $\mathbf{b}_c$ as the current basket that we aim to recommend items for as an alias for $\mathbf{b}^u_{n+1}$. We propose the following personal scoring function $PSF(x, u, \mathbf{b}_c)$ to compute the score of a candidate item x for the current incomplete basket $\mathbf{b}_c$ of user u:

$$PSF(x, u, \mathbf{b_c}) = \alpha \cdot HSF(x, u) + (1 - \alpha) \cdot BSF(x, u, \mathbf{b_c}), \quad (1)$$

where $HSF(x, u)$ is a history-based scoring function that scores an item x based on how the user u has consumed it in the past, regardless of the current basket. $BSF(x, u, \mathbf{b}_c)$ is a basket-based scoring function that considers both the history and the items in the current basket for assigning a score to x. Finally, α is a hyper-parameter to balance the two parts.

History-based scoring. $HSF(x, u)$, the history-based scoring function, accounts for the loyalty of a user to an item. In other words, it considers how frequently a user has purchased the item in the past. In addition to frequency, the recency of occurrences of an item is also important in predicting the next occurrence, as shown in [2, 4]. We define $HSF(x, u)$ as follows:

$$HSF(x, u) = \sum_{t=0}^{|\mathbf{B}_u|-1} \frac{1_{\{x \in \mathbf{b}_t\}}}{|\mathbf{B}_u| - t}, \quad (2)$$

where $1_{\{x \in \mathbf{b}_t\}}$ is equal to one if x occurs in $\mathbf{b}_t$ and zero otherwise. In this formulation, more recent occurrences of an item in the user history are given more weight.

Basket-based scoring. To compute the score of a candidate item x given the items that are already in the current basket, we define the basket-based scoring function $BSF(x, u, \mathbf{b}_c)$. It scores an item x based on its past co-occurrences with the items in the user's past baskets. It takes into account three factors while scoring an item: (i) recent baskets are more informative than older baskets, (ii) recent items in the current basket are more important than older items, and (iii) the distance between the candidate item x and the current item x_i in the past baskets is important. Formally:

$$\begin{aligned} BSF(x, u, \mathbf{b}_c) = &\sum_{i=0}^{|\mathbf{b}_c|-1} \frac{1}{|\mathbf{b}_c| - i} \cdot \\ &\sum_{t=0}^{|\mathbf{B}_u|-1} \frac{1_{\{x \in \mathbf{b}_t\}}}{|\mathbf{B}_u| - t} \cdot \frac{1_{\{x_i \in \mathbf{b}_t\}}}{|I(x, \mathbf{b}_t) - I(x_i, \mathbf{b}_t)|}, \end{aligned} \quad (3)$$

where $I(x, \mathbf{b}_t)$ indicates the index at which x has occurred in basket $\mathbf{b}_t$. In other words, more weight is given to a candidate item that has appeared in close proximity of current items in the recent baskets of the user. The first two factors have already been shown to be effective for session-based recommendation in [5], where they are used to compute the similarity of the current session with all existing sessions. Here, however, we compute the similarity with the personal baskets of the user for scoring candidate items.

4.2 Neighbor-based scoring function

In addition to the personal scoring function, our model has a collaborative component that assigns scores to candidate items according to the neighboring users of the target user u. Specifically, the neighbor-based scoring function $NSF(x, u, \mathbf{b}_c)$ is defined as follows:

$$NSF(x, u, \mathbf{b}_c) = \frac{1}{|N_u|} \sum_{v \in N_u} sim(u, v) PSF(x, v, \mathbf{b}_c), \quad (4)$$

where N_u is the set of neighbors of u of size k and $sim(u, v)$ is the similarity between users u and v. Note that the number of neighbors k is a hyper-parameter in our model. The neighbor-based scoring function $NSF(x, u, \mathbf{b}_c)$ calculates how an item x would be scored by the neighbors of user u, given the items already in the basket. The similarity between a neighbor and the target user is also considered.

User similarity. To compute the similarity $sim(u, v)$ between two users u and v, we consider the history-based scoring from Eq. 2. We represent each user u as a vector over the item space I, where each element in the vector, corresponding to an item $x \in I$, is the history-based score of the item for the user. The similarity is then defined as the cosine similarity between two user vectors. Formally:

$$sim(u, v) = \frac{\sum_{x \in I} HSF(x, u) HSF(x, v)}{\sqrt{\sum_{x \in I} HSF(x, u)^2} \sqrt{\sum_{x \in I} HSF(x, v)^2}}. \quad (5)$$

The set of k nearest neighbors of a user u, containing the most similar users in U to u, is then defined as N_u.

4.3 Final scoring function

The final score of a candidate item x for an incomplete basket $\mathbf{b}_c$ belonging to user u, is calculated as follows:

$$score(x, u, \mathbf{b}_c) = \beta \cdot PSF(x, u, \mathbf{b}_c) + (1 - \beta) \cdot NSF(x, u, \mathbf{b}_c), \quad (6)$$

where β is a hyper-parameter of our model, balancing the contributions of the personal history component and the collaborative component. The model assigns a score to all items in $I \setminus \mathbf{b}_c$, and the top n scored items are returned as the recommendation list.

5 VECTORIZED IMPLEMENTATION

Analogous to session-based recommendation, within-basket recommendation models are challenging to deploy in real-world scenarios, as they need to respond online to users filling their baskets. E-commerce applications typically need to respond with a latency of less than 50 milliseconds in at least 90 percent of requests [11], and it has been observed that low prediction latency contributes to the acceptance of recommendations by users [1, 10].

The design of PerNIR supports such low-latency prediction scenarios. Major parts of PerNIR depend on the static user history $\mathbf{B}_u$ (which only changes after purchases), while only the incomplete basket $\mathbf{b}_c$ changes dynamically during the shopping session. In the following, we describe how to exploit this structure to precompute the parts of the model that only depend on the static user history $\mathbf{B}_u$ and how to vectorize the underlying computations to score multiple candidate items at once. Note that we model a basket $\mathbf{b} \in \{0,1\}^{|I|}$ as a binary vector in item space for these purposes.

Offline precomputation. For each user u, we can precompute the outputs of HSF (Eq. 2) into a user-specific history vector $\mathbf{h}_u$:

$$\mathbf{h}_u = \sum_{t=0}^{|\mathbf{B}_u|-1} \frac{1}{|\mathbf{B}_u| - t}\, \mathbf{b}_t. \tag{7}$$

We analogously precompute the item co-occurrences (the second sum from Eq. 3) from their basket history matrix into a user-specific basket co-occurrence matrix $\mathbf{C}_u$. Each entry (i, j) of this matrix denotes the weight assigned to the co-occurrence of two items i and j in the basket history $\mathbf{B}_u$:

$$\mathbf{C}_u = \left[\sum_{t=0}^{|\mathbf{B}_u|-1} \frac{1_{\{x_i \in \mathbf{b}_t\}}\, 1_{\{x_j \in \mathbf{b}_t\}}}{(|\mathbf{B}_u| - t)\ |I(x_i, \mathbf{b}_t) - I(x_j, \mathbf{b}_t)|} \right]_{ij}. \tag{8}$$

Note that $\mathbf{C}_u \in \mathbb{R}^{|I| \times |I|}$ is high-dimensional but extremely sparse, as its number of non-zeros is at most the sum of the squares of the number of distinct items per basket in the user history. Next, we precompute the top-k similar users N_u for a user u (which only depend on the users' static history vectors). Based on the corresponding similarities, we precompute the combination $\mathbf{h}_{s_u}$ of the user's history vector $\mathbf{h}_u$ with the history vectors of their neighbors $v \in N_u$, and analogously precompute the combination $\mathbf{C}_{s_u}$ of their co-occurrence matrices:

$$\mathbf{h}_{s_u} = \beta\, \mathbf{h}_u + \frac{1-\beta}{|N_u|} \sum_{v \in N_u} sim(u,v)\, \mathbf{h}_v \tag{9}$$

$$\mathbf{C}_{s_u} = \beta\, \mathbf{C}_u + \frac{1-\beta}{|N_u|} \sum_{v \in N_u} sim(u,v)\, \mathbf{C}_v. \tag{10}$$

Online inference. At inference time, we compute a "summation and selection" vector $\phi(\mathbf{b}_c)$ from the dynamically changing incomplete basket $\mathbf{b}_c$, which contains the associated weights from the first sum of Eq. 3:

$$\phi(\mathbf{b}_c) = \left[\frac{1_{\{x_i \in \mathbf{b}_c\}}}{|\mathbf{b}_c| - I(x_i, \mathbf{b}_c)} \right]_i. \tag{11}$$

We can now efficiently compute item scores based on $\phi(\mathbf{b}_c)$ and the precomputed "personal" model $\mathbf{h}_{s_u}$ and $\mathbf{C}_{s_u}$ for user u:

$$score(u, \mathbf{b}_c) = \alpha\, \mathbf{h}_{s_u} + (1-\alpha)\, \mathbf{C}_{s_u}\, \phi(\mathbf{b}_c). \tag{12}$$

Note that the scoring computation only requires a single sparse matrix vector multiplication and a single sparse vector addition, for which we can leverage highly optimized implementations from SparseBLAS [3].

6 EXPERIMENTAL SETUP

We address the following research questions (RQs): (RQ1) How does PerNIR perform compared with existing state-of-the-art models for within-basket recommendation in grocery shopping? (RQ2) What are the effects of the different components *HSF*, *BFS*, and *NFS* of PerNIR and hyper-parameters α, β, and the number of neighbors k on the overall performance? (RQ3) How sensitive is PerNIR's performance to characteristics such as the number of baskets per user or the number of items in the current basket? (RQ4) How efficient is PerNIR in terms of inference latency? To answer our research questions we consider two experimental setups.

Table 1: Dataset statistics after preprocessing.

Dataset	Users	Items	Baskets	Avg. item per basket	Avg. basket per user
Instacart	30,000	43,936	413,860	11.9	13.8
X-online	10,000	22,486	141,342	38.2	14.1

Setup for effectiveness experiments. To answer RQ1–RQ3 we design a set of contrastive experiments with a diverse set of state-of-the-art baselines.

Datasets. We use two datasets in our experiments. While other grocery shopping datasets do exist [17], they do not contain the add-to-basket order data which is crucial for within-basket recommendation [19, 20]. The datasets are described below and their statistics are summarized in Table 1. Instacart[1] is publicly available, and X-online comes from a large food retailer in Europe. We randomly sample 30,000 users from Instacart and 10,000 users from X-online and retrieve all baskets of the users. In both datasets, we remove users with less than three baskets, items occuring in less than five baskets in total, and baskets with less than four items.

Baselines. In addition to the simple baseline P-POP, we compare PerNIR with within-basket recommendation models (BasConv, MITGNN), session-based recommendation models (VSKNN, STAN), a personalized session-based recommendation model (HG-GNN), and next basket recommendation models (TIFU-KNN, ReCANet):

- P-POP: Recommends the most popular items in the user history, sorted by their frequency of purchases. P-POP is considered one of the strongest baselines in NBR for grocery shopping [2], and we consider it as a baseline for our task as well.
- BasConv [20]: Defines a basket entity to represent the basket intent and models the recommendation task as a basket-item link prediction task in the user-basket-item graph. It utilizes graph convolutional networks to learn representations.
- MITGNN [19]: An approach based on graph neural networks to model multiple intents in the baskets.
- VSKNN [21]: A neighborhood-based model proposed that puts a strong emphasis on more recent events of a session when computing session similarities.
- STAN [5]: A neighborhood-based model that considers the position of an item in the current session, the recency of a past session w.r.t. to the current session, and the position of a recommendable item in a neighboring session when computing session similarities.
- HG-GNN [25]: A graph-augmented hybrid encoder that consists of a heterogeneous graph neural network and a personalized session encoder to generate a session preference embedding for personalized session-based recommendation.
- TIFU-KNN [7]: A nearest neighbor-based model that outperforms deep recurrent neural networks in NBR. The model relies on the similarity of the target user with other users and the purchase history of the target user.

[1] https://www.kaggle.com/c/instacart-market-basket-analysis

Table 2: Results of PerNIR compared against the baselines. Boldface and underline indicate the best and second best performing model, respectively. Significant improvements of PerNIR over the best baseline are marked with † (paired t-test, $p < 0.05$).

		Instacart				X-Online			
Recommendation model type	**Model**	HR@10	MRR@10	HR@20	MRR@20	HR@10	MRR@10	HR@20	MRR@20
Within-basket	BasConv	0.0699	0.0268	0.1070	0.0293	0.0385	0.0151	0.0602	0.0166
	MITGNN	0.0672	0.0258	0.1026	0.0283	0.0379	0.0151	0.0606	0.0166
Personalized session-based	HG-GNN	0.0353	0.0125	0.0551	0.0139	0.0499	0.0222	0.0726	0.0237
Session-based	VSKNN	0.1105	0.0426	0.1637	0.0462	0.0829	0.0312	0.1271	0.0342
	STAN	0.0914	0.0385	0.1280	0.0410	0.0975	0.0477	0.1282	0.0498
	PVSKNN	0.2372	0.0915	0.3354	0.0983	0.1839	0.0742	0.2621	0.0796
	PSTAN	0.2142	0.0871	0.3115	0.0938	0.2009	0.1049	0.2607	0.1090
Next basket	PPOP	0.2224	0.0846	0.3210	0.0914	0.1765	0.0695	0.2557	0.0750
	TIFUKNN	0.2069	0.0799	0.2843	0.0853	0.1555	0.0600	0.2267	0.0649
	ReCANet	0.2550	0.0977	0.3585	0.1049	0.1835	0.0734	0.2665	0.0791
	PerNIR	**0.2592**†	**0.1044**†	**0.3625**†	**0.1115**†	**0.2258**†	**0.1069**†	**0.3006**†	**0.1120**†

- ReCANet [2]: A repeat consumption-aware neural network that explicitly models the repeat consumption behavior of users in order to predict their next basket.

In the case of NBR baselines, we considered two setups: discarding the current basket and treating it as the final basket in the history. The former gave the best performance results and we report those. In case of session-based and personalized session-based baselines, each session is considered as a basket. We further use two modified session-based models, namely Personalized SKNN (PSKNN) and Personalized STAN (PSTAN). In these two models, the neighboring baskets are the personal baskets of the user.

Data split. We follow the same procedure as Latifi et al. [12]. For every user, we sort the baskets by purchase time and use the last basket of each user as test data. The second-to-last basket is used as validation data to tune the parameters. The remaining baskets are considered as training data. Table 1 shows the statistics for the training data after preprocessing. Each test or validation basket produces several test samples. Specifically, the first three items in the basket are used as the seed for the current basket, and the rest of the items are added iteratively to the current basket and the next item is used as the ground truth item. Evaluation is performed for each non-seed item in the basket. For example, given a test or validation basket $b = [x_1, x_2, \ldots, x_6]$, the evaluation is performed on the following samples: $\{X = [x_1, x_2, x_3], Y = [x_4]\}$, $\{X = [x_1, x_2, x_3, x_4], Y = [x_5]\}$,$\{X = [x_1, x_2, x_3, x_4, x_5], Y = [x_6]\}$.

Evaluation metrics. We use Hit Rate (HR) @n and Mean Reciprocal Rank (MRR) @n as evaluation metrics. HR measures if the relevant item appears in the list of recommendations, and MRR measure how high the relevant item is ranked. Both measures are calculated across the predicted next item for all test users and all test samples. We report the metrics for $n \in \{10, 20\}$.

Parameter settings. We perform a grid search to find the hyper parameters that result in the best performance on the validation set, and use those for testing. The hyper-parameters of the baselines are either tuned or set according to instructions in the original papers if available.

Setup for efficiency experiments. To answer RQ4, we use an alternative experimental design. Our goal will be to showcase that PerNIR is able to handle inference workloads with a latency low enough for production workloads, where users browse a site and fill their baskets in response to online recommendations. We choose the Instacart dataset for this experiment as it is larger than the X-Online dataset in terms of users, items and baskets, and therefore more challenging for inference workloads.

We run this experiment in a single thread on a `StandardD8v4` instance in the Microsoft Azure cloud, with an Intel Xeon Platinum 8272CL CPU@2.60GHz and 32gb of RAM, using Ubuntu 20.04, Python 3.9 and scipy 1.9.0.

We pick 1,000 users at random from the Instacart dataset, and ask PerNIR to score items for five randomly chosen incomplete test baskets per user. We base the prediction on the precomputed "personal" model for each user (as discussed in Section 5). We measure the response time in milliseconds, and repeat this experiment for increasing numbers of neighbors k, ranging from 20 to 500.

Reproducibility. To facilitate reproducibility and follow-up research, we share our code.[2]

7 RESULTS AND ANALYSIS

We conduct extensive experiments to answer our research questions. In this section, we describe the results of our experiments.

Performance comparison (RQ1). We compare the performance of PerNIR with several state-of-the-art baselines, and list the results in Table 2. First, we observe that HG-GNN is the baseline with the weakest performance. This model is originally proposed for personalized session-based recommendation, which is identical to within-basket recommendation in terms of problem formulation. However, the grocery shopping domain has special characteristics, such as repeat behavior and larger baskets compared to online sessions, which are not considered by HG-GNN and other personalized session-based recommendation models.

[2] https://github.com/mzhariann/pernir

Table 3: Results of the ablation study. Different variations of PerNIR are compared with each other and the final model.

		Instacart				X-Online			
#	**Model variation**	HR@10	MRR@10	HR@20	MRR@20	HR@10	MRR@10	HR@20	MRR@20
1	w/o history-based scoring ($\alpha = 0$)	0.2525	0.1011	0.3544	0.1081	0.2138	0.1021	0.2867	0.1072
2	w/o basket-based scoring ($\alpha = 1$)	0.2465	0.0955	0.3513	0.1028	0.1764	0.0693	0.2513	0.0745
3	w/o personal history ($\beta = 0$)	0.1363	0.0594	0.1859	0.0628	0.0880	0.0389	0.1249	0.0414
4	w/o neighbor scoring ($\beta = 1$)	0.2548	0.1020	0.3542	0.1089	0.2145	0.1031	0.2841	0.1079
5	w/o user similarity weighting	0.2564	0.1032	0.3569	0.1101	0.2047	0.0936	0.2778	0.0987
6	w/ binary user similarity scoring	0.2586	0.1035	0.3628	0.1107	0.2166	0.1042	0.2873	0.1091
7	w/o basket recency	0.2481	0.0986	0.3496	0.1056	0.2120	0.1011	0.2801	0.0989
8	w/o item recency	0.2519	0.0969	0.3587	0.1043	0.1851	0.0751	0.2643	0.0805
9	w/o item distance	0.2527	0.0996	0.3569	0.1068	0.1835	0.0744	0.2576	0.0795
	PerNIR	0.2592	0.1044	0.3625	0.1115	0.2258	0.1069	0.3006	0.1120

The GNN-based within-basket recommendation models BasConv and MITGNN perform poorly. One possible reason for this is that these models are trained on large baskets only (more than 40 items) in the original papers, which might limit their ability to generalize to our setting, where we set the lower bound of the basket size to four items. Moreover, these models are trained for predicting the rest of the basket where 80% of the items in the current basket are already given, which is again different from our setting. We add items to the baskets one by one and use each item as a label during evaluation, which is closer to the real-world setting where users receive recommendations after each item is added to the basket.

The session-based recommendation models VSKNN and STAN in their original formulation perform better than the previous models, but fall short of their modified personalized versions. This shows that the user personal history is an important indicator in within-basket recommendation. VSKNN and STAN focus on finding items from baskets that are similar to the current incomplete basket, and cannot utilize the personal preferences of users. By limiting the search for similar baskets to the personal baskets, PVSKNN and PSTAN get a boost in performance. Specifically, PSTAN is the best performing baseline on X-Online based on three out of four metrics, and PVSKNN is the second-best performing one on Instacart. This shows that the components in VSKNN and PSTAN are able to utilize the items in the current basket for scoring candidate items.

Next basket recommendation models provide the best baseline performance on both datasets. Among them, ReCANet is the strongest competitor, followed by PPOP and TIFUKNN. This is in line with the results observed for NBR [2]. NBR models focus on utilizing personal purchasing history, but are unable to use the signals in the current incomplete basket. These models demonstrate higher performance compared to the performance of session-based recommendation models that use only the incomplete basket but not the personal history. This result once again confirms the importance of the personal history in grocery shopping.

PerNIR outperforms all the baselines on both datasets, and the improvements over the best performing baseline are statistically significant on all metrics, using a paired t-test. The improvements on Instacart are 1.6% and 1.1% for HR@10 and HR@20, respectively. Larger improvements are obtained on MRR@10 and MRR@20, namely 6.8% and 6.2%. Improvements on X-Online are even more substantial, especially in terms of HR. We observe 1.9% and 2.7% increases in terms of MRR@10 and MRR@20 with respect to PSTAN, as well as 12.3% and 12.7% improvements over PSTAN and ReCANet in terms of HR@10 and HR@20. Overall, the results show that PerNIR is effective for within-basket recommendation.

Ablation study (RQ2). We conduct an extensive ablation study with nine different variations of PerNIR to analyze the effect of different components in the model. Table 3 shows the results. First, we observe that the final model is superior to all variations on both datasets for all metrics, with the exception of one case (HR@20 on variation 6, where there is a 0.1% performance degradation). This indicates that all of the components in PerNIR contribute to the final performance and are necessary to achieve the best performance.

The first two variations concern the effect of HSF and BSF in the personal scoring function. By setting α to zero, we remove the HSF and with setting it to one, we remove BSF. Both cases result in a drop in performance for all metrics on both datasets. However, removing BSF has a more substantial effect on performance ranging from 3.0% to 8.5% on Instacart and 16.4% to 35.1% on X-Online. The effect is considerably higher on X-Online; one reason for this could be the larger basket sizes on X-Online, which translates to more signals being lost when the current basket is ignored.

In variations 3 and 4, we remove the personal scoring function PSF and neighboring scoring function NSF by setting β to zero and one, respectively. In both cases the performance degrades for all metrics and datasets. The effect of removing PSF is substantially larger than of removing NSF, where the performance drop ranges from 43.1% to 48.7% on Instacart and 58.4% to 63.6% on X-Online. This indicates the importance of modeling personal history in PerNIR. Removing NSF results in 1.6% to 2.3% and 3.5% to 5.4% drops in performance on Instacart and X-Online, respectively. Therefore, the neighbor-based scoring function is essential and contributes to the final performance.

Another component in PerNIR is the user similarity computation, which is analyzed by variations 5 and 6. In variation 5, we remove the user similarity scores altogether and set $sim(u, v) = 1$ in Eq. 4, which means that all neighbors have the same weight in the neighbor-based scoring function NSF. This results in a small drop in performance on Instacart, ranging from 1% to 1.5%. The effect is considerable on X-Online, where the degradation ranges from 7.5%

to 12.4%. One explanation is the larger number of users on Instacart, which could lead to more similar users in the neighborhood, which lowers the need for the similarity scores. In variation 6, we use another similarity scoring function to compute $sim(u, v)$, which is essentially removing *HSF* from Eq. 5, and representing each user by a binary vector with the length of the item space, where each element in the vector indicates the occurrence of the corresponding item in the purchasing history of the user. This results in the lowest performance drop on both datasets for all metrics compared to the other variations, ranging from −0.1% to 4.4%. Hence, using *HSF* in computing user similarity is effective, but the effect is not critical.

In variations 7, 8, and 9, we study the effect of different components in computing the basket-based scoring function *BSF*. In variation 7, we remove the effect of basket recency by using the same weight for all baskets in the user history in Eq. 3. We remove the effect of item recency in the current basket in variation 8 by setting the weights of all items in the current basket to one. We further remove the effect of the distance between a candidate item and the item in the current basket in historical baskets in variation 9, by using the same weight of one for all candidate items. In all cases, the performance degrades; all of these components contribute to the performance. Interestingly, for variations 8 and 9, the performance drop for X-Online is more than for Instacart, ranging from 12.0% to 30.4%. This could be a result of larger basket sizes in X-Online; since baskets are larger on X-Online, it becomes more important to consider the position of items in the baskets, in both incomplete baskets and in the historical baskets.

We further study the effect of the three hyper-parameters α, β, and k on the performance. We perform a grid search where α and β are swiped in $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$ and k in $\{50, 250, 500\}$. We find that on Instacart, $k = 250$, $\alpha = 0.1$, and $\beta = 0.2$ result in the best performance on the validation set in terms of HR@20. A similar observation is made on X-Online, with the difference of $k = 50$ giving the best performance.

Figure 1 (left) shows the performance of PerNIR for different values of α and k. Increasing α leads to lower performance for all k values. Hence, in the personal scoring function *PSF*, items in the current basket should be considered more effectively; relying heavily on *HSF*, which ignores the current basket hurts, performance.

Figure 1 (right) shows the performance of PerNIR for different values of β and k. For all values of k, setting β to 0.2 results in the best performance. β balances the contribution of personal scoring function and neighbor-based scoring function to the final score of items. A high value for β results in a lower effect for *NSF*, which degrades the performance. The optimal value of k is 250 on Instacart, however, it has a lower effect on the performance compared to α and β. Increasing the number of neighbors from 250 to 500 has a minimal effect on performance, while the difference is more apparent between $k = 50$ and $k = 250$, which indicates that a too low value for k hurts the performance.

Sensitivity analysis (RQ3). In this section, we study the sensitivity of PerNIR's performance to characteristics of users and baskets. Specifically, we analyze the recommendation performance for incomplete baskets of varying lengths and users with varying numbers of baskets in their history. We also consider the performance for cases where the target item is a previously purchased item vs. cases where it is a new item.

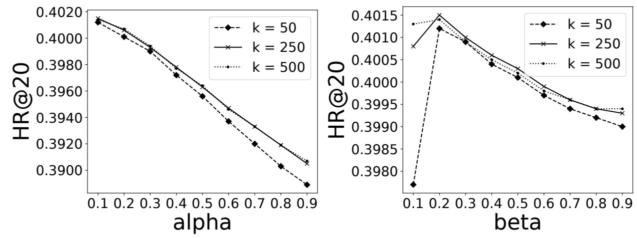

Figure 1: Sensitivity of the performance in terms of HR@20 to hyper-parameters α and β for different values of k on Instacart.

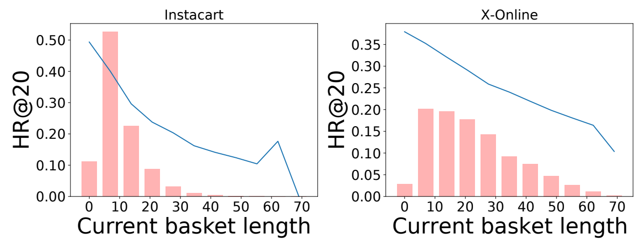

Figure 2: Performance of PerNIR in terms of HR@20 based on the number of items in the current basket. Bars show the distribution of basket lengths.

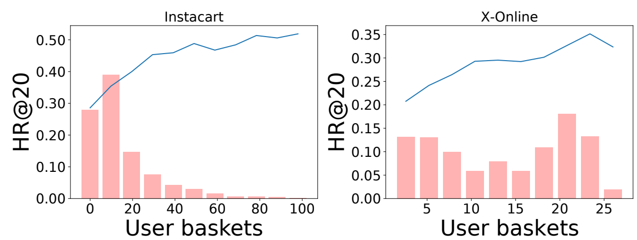

Figure 3: Performance of PerNIR in terms of HR@20 based on the number of baskets in users' history. Bars show the distribution of number of baskets per user.

Figure 2 shows the distribution of basket lengths, i.e., number of items in the current incomplete basket, in the test samples for Instacart and X-Online (bars) as well as the performance in terms of HR@20 (line). We observe that in both datasets the distribution is right-skewed; most of the baskets contain a small number of items. The performance correlates negatively with the basket size in both cases: the smaller the size of the current basket, the higher the performance of PerNIR. This means that recommending the last items in the basket is more difficult for PerNIR. One reason for this could be that users add more obvious, regular items earlier to their basket, which are easier for the model to predict.

Figure 3 shows the distribution of the number of baskets in the users' history (bars) as well as the performance in terms of HR@20 (line). In Instacart, the distribution is right-skewed; most users have a small number of baskets in their history. In X-Online, the distribution is closer to uniform, with users with a small, medium, or large number of baskets in their history. In both datasets, the

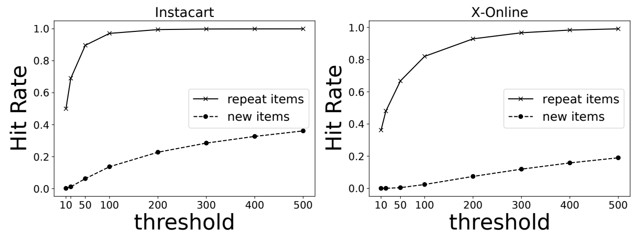

Figure 4: Performance of PerNIR in terms of hit rate at different thresholds, for two cases: when the target item is a previously purchased item and when it is a new item.

performance has a positive correlation with the number of baskets in the users' history. As users purchase more, PerNIR gets better at learning their shopping behavior and recommends more accurately.

We further analyze the performance of PerNIR for cases where the target item has previously been purchased by the user (i.e., a repeat item) vs. the cases where it is a new item. Figure 4 shows the performance in terms of hit rate at different thresholds, for repeat and new items on both datasets. There is a significant difference in performance: predicting a repeat item is much easier than predicting a new item. While the performance reaches it maximum for repeat items at a threshold of 100 and 500 items for Instacart and X-Online, respectively, recommending accurate new items remains challenging even at high thresholds. The new items are recommended based on the items that neighboring users have purchased before. This makes the search space larger and finding the correct target item more challenging. Additionally, not all new items can be found in the neighbors' item space, which further limits the achievable performance. We conclude that repeat item recommendation is an easier task for PerNIR in the within-basket recommendation scenario, in line with previous findings in NBR for grocery shopping [17].

Prediction latency (RQ4). Next, to answer RQ4, we evaluate the prediction latency of our precomputation approach from Section 5.

We plot the median and 90th percentile (p90) of the response times of PerNIR in Figure 5. Both the median and p90 prediction latency scale linearly with the number of neighbors k. Our precomputation approach is able to conduct inference with low prediction latency even for higher numbers of k, e.g., with a p90 latency of less 1.4 milliseconds for $k = 250$, which scored high in our evaluation. These results give a strong indication that PerNIR handles real-world deployments where low-latency online responses for users browsing a site and filling their baskets are required.

In addition, we list the average model sizes (in terms of the number of non-zero entries in the sparse history vectors $\mathbf{h}_{s_u}$ (nnz-h) and cooccurrence matrices $\mathbf{C}_{s_u}$ (nnz-c)) of the precomputed personal models for various values of k in Table 4. The model size also scales linearly with k and the personal models only require a small number of megabytes in terms of storage, e.g., the average model size for $k = 250$ would be less than ten megabytes with 64bit floating point numbers. This small size would, for example, make it feasible to ship these models to user devices such as mobile phones.

8 CONCLUSION

In this paper, we have proposed PerNIR, a personalized neighborhood-based model for within-basket recommendation in grocery shopping. Our proposed method has different components for explicitly modeling personal preferences of users and utilizing the purchasing behavior of neighboring users.

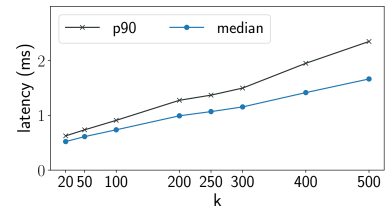

Figure 5: Median and 90-th percentile of the prediction latency of PerNIR for an increasing number of neighbors on the Instacart dataset (in milliseconds). PerNIR is able to respond with a p90 latency of less than 1.4 milliseconds for $k = 250$, which scored high in our evaluation.

Table 4: Average number of non-zero entries in the sparse history vectors (nnz-h) and cooccurrence matrices (nnz-c) of the precomputed personal models for various values of k.

k	100	200	250	300	400	500
nnz-h	3,173	5,027	5,789	6,468	7,654	8,700
nnz-c	153,236	307,630	384,850	463,615	614,432	761,671

Through extensive experiments, we have demonstrated the effectiveness of PerNIR compared to state-of-the-art baselines in terms of different performance metrics. We have further studied the contribution of different components in our model in an ablation study, which has revealed the necessity of all of them for achieving the best performance. Moreover, we have provided a vectorized implementation of PerNIR which allows for fast inference. As a result, PerNIR can provide low-latency predictions that are required in real-world recommendation systems.

A broader implication of our work is that PerNIR is applicable to the personalized session-based recommendation scenario, where sessions can be treated as baskets.

In terms of limitations, we have found that the performance of PerNIR degrades as the number of items in an incomplete basket increases. In future work, we aim to further study and address this phenomenon. One possible solution for this could be learning better weights for items in a given basket, which is currently calculated solely based on the recency of the addition of the item to the basket. Our experimental results have revealed a substantial difference in performance of PerNIR between the cases when the target item is a previously purchased item or a new item. In future work, we aim to improve the collaborative component in PerNIR to better capture signals for recommending items unseen in users' personal history.

ACKNOWLEDGMENTS

This research was supported by Ahold Delhaize, the China Scholarship Council (grant number 20190607154), and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Ioannis Arapakis, Xiao Bai, and Berkant Barla Cambazoglu. 2014. Impact of response latency on user behavior in web search. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 103–112.

[2] Mozhdeh Ariannezhad, Sami Jullien, Ming Li, Min Fang, Sebastian Schelter, and Maarten de Rijke. 2022. ReCANet: A Repeat Consumption-Aware Neural Network for Next Basket Recommendation in Grocery Shopping. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1240–1250.

[3] Iain S Duff, Michael A Heroux, and Roldan Pozo. 2002. An Overview of the Sparse Basic Linear Algebra Subprograms: The New Standard from the BLAS Technical Forum. *ACM Transactions on Mathematical Software (TOMS)* 28, 2 (2002), 239–267.

[4] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2020. Recency Aware Collaborative Filtering for Next Basket Recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization.* ACM, 80–87.

[5] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and Time Aware Neighborhood for Session-based Recommendations: STAN. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1069–1072.

[6] Jiayan Guo, Yaming Yang, Xiangchen Song, Yuan Zhang, Yujing Wang, Jing Bai, and Yan Zhang. 2022. Learning Multi-granularity Consecutive User Intent Unit for Session-based Recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining.* ACM, 343–352.

[7] 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.* ACM, 1071–1080.

[8] Liang Hu, Qingkui Chen, Haiyan Zhao, Songlei Jian, Longbing Cao, and Jian Cao. 2018. Neural Cross-Session Filtering: Next-Item Prediction Under Intra- and Inter-Session Context. *IEEE Intell. Syst.* 33, 6 (2018), 57–67.

[9] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems.* ACM, 306–310.

[10] Barrie Kersbergen and Sebastian Schelter. 2021. Learnings from a Retail Recommendation System on Billions of Interactions at bol.com. In *37th IEEE International Conference on Data Engineering.* IEEE, 2447–2452.

[11] Barrie Kersbergen, Olivier Sprangers, and Sebastian Schelter. 2022. Serenade - Low-Latency Session-Based Recommendation in e-Commerce at Scale. In *SIGMOD '22: International Conference on Management of Data.* ACM, 150–159.

[12] Sara Latifi, Noemi Mauro, and Dietmar Jannach. 2021. Session-aware Recommendation: A Surprising Quest for the State-of-the-art. *Inf. Sci.* 573 (2021), 291–315.

[13] Duc-Trong Le, Hady Wirawan Lauw, and Yuan Fang. 2017. Basket-Sensitive Personalized Item Recommendation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence.* ijcai.org, 2060–2066.

[14] Duc-Trong Le, Hady W. Lauw, and Yuan Fang. 2019. Correlation-Sensitive Next-Basket Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.* ijcai.org, 2808–2814.

[15] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* ACM, 1419–1428.

[16] Miao Li, Xuguang Bao, Liang Chang, and Tianlong Gu. 2022. Modeling Personalized Representation for Within-basket Recommendation based on Deep Learning. *Expert Syst. Appl.* 192 (2022), 116383.

[17] Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. 2021. A Next Basket Recommendation Reality Check. *CoRR* abs/2109.14233 (2021). arXiv:2109.14233

[18] 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 & Data Mining.* ACM, 1831–1839.

[19] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and Philip S. Yu. 2020. Basket Recommendation with Multi-Intent Translation Graph Neural Network. In *2020 IEEE International Conference on Big Data (IEEE BigData 2020).* IEEE, 728–737.

[20] Zhiwei Liu, Mengting Wan, Stephen Guo, Kannan Achan, and Philip S. Yu. 2020. BasConv: Aggregating Heterogeneous Interactions for Basket Recommendation with Graph Convolutional Neural Network. In *Proceedings of the 2020 SIAM International Conference on Data Mining.* SIAM, 64–72.

[21] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User Model. User Adapt. Interact.* 28, 4-5 (2018), 331–390.

[22] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance Comparison of Neural and Non-neural Approaches to Session-based Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems.* ACM, 462–466.

[23] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2021. Empirical Analysis of Session-based Recommendation Algorithms. *User Model. User Adapt. Interact.* 31, 1 (2021), 149–181.

[24] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten de Rijke. 2020. Star Graph Neural Networks for Session-based Recommendation. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management.* ACM, 1195–1204.

[25] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous Global Graph Neural Networks for Personalized Session-based Recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining.* ACM, 775–783.

[26] Tu Minh Phuong, Tran Cong Thanh, and Ngo Xuan Bach. 2019. Neural Session-Aware Recommendation. *IEEE Access* 7 (2019), 86884–86896.

[27] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 859–868.

[28] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems.* ACM, 130–137.

[29] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. 2017. Inter-Session Modeling for Session-Based Recommendation. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems.* ACM, 24–31.

[30] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian J. McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* ACM, 1133–1142.

[31] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, The Thirty-First Innovative Applications of Artificial Intelligence Conference, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence.* AAAI Press, 346–353.

[32] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, The Eleventh Symposium on Educational Advances in Artificial Intelligence.* AAAI Press, 4503–4511.

[33] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.* ijcai.org, 3926–3932.

[34] 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.* ACM, 729–732.

[35] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. 2020. Personalized Graph Neural Networks with Attention Mechanism for Session-Aware Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1.