ReCANet: A Repeat Consumption-Aware Neural Network for Next Basket Recommendation in Grocery Shopping

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

Sami Jullien
AIRLab, University of Amsterdam
Amsterdam, The Netherlands
s.jullien@uva.nl

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

Min Fang
Albert Heijn
Zaandam, The Netherlands
min.fang@ah.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

Retailers such as grocery stores or e-marketplaces often have vast selections of items for users to choose from. Predicting a user's next purchases has gained attention recently, in the form of next basket recommendation (NBR), as it facilitates navigating extensive assortments for users. Neural network-based models that focus on learning basket representations are the dominant approach in the recent literature. However, these methods do not consider the specific characteristics of the grocery shopping scenario, where users shop for grocery items on a regular basis, and grocery items are repurchased frequently by the same user.

In this paper, we first gain a data-driven understanding of users' repeat consumption behavior through an empirical study on six public and proprietary grocery shopping transaction datasets. We discover that, averaged over all datasets, over 54% of NBR performance in terms of recall comes from repeat items: items that users have already purchased in their history, which constitute only 1% of the total collection of items on average. A NBR model with a strong focus on previously purchased items can potentially achieve high performance. We introduce *ReCANet*, a repeat consumption-aware neural network that explicitly models the repeat consumption behavior of users in order to predict their next basket. ReCANet significantly outperforms state-of-the-art models for the NBR task, in terms of recall and nDCG. We perform an ablation study and show that all of the components of ReCANet contribute to its performance, and demonstrate that a user's repetition ratio has a direct influence on the treatment effect of ReCANet.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Personalization**.

KEYWORDS

Next basket recommendation, Repeat behavior, Grocery shopping

ACM Reference Format:
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 *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531708

1 INTRODUCTION

Recommendation systems in retail help users to find the items that they need from large inventories. Different retail industries such as fashion [7], general e-commerce [15], and grocery shopping [18] utilize recommendation systems to facilitate the shopping experience for their users. Different types of recommendation systems, such as top-n [8], sequential [25], and session-based [10] systems, are deployed for different scenarios. In grocery shopping in particular, the next basket recommendation (NBR) formulation is considered to be the most relevant and has gained increasing attention in the literature recently [9, 13, 20].

Next basket recommendation for grocery shopping. The NBR task is defined as recommending a group of items to a user based on their shopping history, where the history is a time-ordered sequence of baskets that they have purchased in the past. Each basket is a set of items with no particular order [30]. This formulation fits the grocery shopping setting well, where a user's purchase history occurs naturally in the form of such baskets. Two main characteristics of the grocery shopping scenario make it distinct from other retail domains: (1) users shop for grocery items repeatedly and on a regular basis, and (2) grocery items have a short life time and are repurchased frequently by the same user [17].

Previous work has studied the NBR problem within different algorithmic frameworks. Specifically, a number of neural network-based methods have been proposed to learn basket representations [12, 20, 26, 30], while nearest neighbor-based methods are dominant in modeling the personal history of users [9, 13]. In the grocery shopping domain, recent studies of NBR models has shown that nearest neighbor-based models that consider the characteristics of this domain and explicitly model the personal history of users significantly outperform neural network-based models that

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531708

are mostly focused on basket representation and consider the previously purchased items only implicitly [13, 16]. The superiority of nearest neighbor-based models is further shown to be caused by their ability to correctly recommend items that have previously been purchased by a user, so-called *repeat items* [16].

Understanding repeat-consumption behavior. Inspired by previous work highlighting the importance of repeat items in grocery shopping [2, 9, 13, 16], we study the repeat consumption behavior across six public and proprietary datasets in the grocery shopping domain (Section 4). We find that different users exhibit different levels of repeat consumption in all datasets. As time goes by and more baskets are added to a user's history, the repeat consumption increases. In all datasets, most of the items are subject to repurchasing with a high probability. We further demonstrate what these findings mean in terms of predictive performance for NBR. Specifically, we discover that averaged over all datasets, over 54% of the performance in terms of recall comes from such repeat items: items that the users have already purchased in their history, which only account for 1% of the total items on average. This high performance indicates that a NBR model that is focused on previously purchased items can potentially achieve a high performance, while considering only a small number of candidate items per user. Section 4.3 shows that most of the performance of recommender systems in NBR stems from the minority of repeat items in the dataset.

Existing nearest neighbor-based models [9, 13] consider three main factors in recommending repeat items for the next basket: (1) the frequency of items bought in a users' history, (2) the recency of items bought in the history, and (3) similarities between users. Repurchasing patterns of items are *not* reflected in these models. For example, one user might buy milk in every basket, and buy apples every two-to-three baskets only. Moreover, similarities between items are also not taken into account. Considering item similarities has the potential to help the generalization power of a recommendation model, in cases where the consumption data is scarce [3].

ReCANet. We propose ReCANet, a repeat consumption-aware neural network that addresses the shortcomings of existing NBR models (Section 5). In addition to frequency and recency of items purchased by a user, and user similarities, ReCANet further considers item similarities and personal repeat consumption patterns. The intuition behind ReCANet is that the consumption pattern of an item by a user indicates whether the item will appear in the next basket or not. Additionally, the consumption pattern of that particular item by other users can also help the prediction. To this end, we use embedding layers to represent items and users, and model the consumption patterns with LSTM [11] layers. To evaluate the performance of ReCANet for NBR, we compare it with several state-of-the-art models on six datasets in terms of multiple standard ranking metrics (Section 6). The experimental results confirm the effectiveness of ReCANet (Section 7); it significantly outperforms the state-of-the-art models in terms of recall and nDCG. The improvements over state-of-the-art baselines are consistent across all datasets.

To summarize, our contributions are as follows:

- We provide an empirical study on the characteristics of repeat consumption behavior across six grocery shopping datasets (Section 4). To the best of our knowledge, this is the first study of this kind.
- We propose ReCANet, a novel neural architecture for NBR in grocery shopping, focused on repeat items (Section 5).
- We demonstrate the effectiveness of ReCANet in comparison with state-of-the-art NBR models through experiments on three public and three proprietary datasets, where ReCANet is shown to significantly outperform the baselines (Section 7).
- We validate the importance of different components of ReCANet through an ablation study and conduct parameter and user sensitivity experiments to demonstrate its robustness for different hyper-parameters and user groups (Section 7).

2 PROBLEM FORMULATION

The goal of the NBR task is to recommend a full basket composed of a list of items to the user for their next basket, based on the history of the items that they have purchased in the past. Such recommendations reduce the burden on users to proactively find items of interest every time they need to shop.

Formally, a basket is a set of items defined as $B = \{x_1, x_2, \ldots, x_t\}$, where $x_i \in I$ denotes an item from a set of items I. Given the history of purchases for user $u \in U$ as the sequence $B^u = [B_1^u, B_2^u, \ldots, B_n^u]$, where B_i^u is the i-th basket in the purchase history of user u and U denotes the set of users, the goal is to predict the items in the next basket of the user, i.e., B_{n+1}^u. For the basket history B^u, the recommendation model assigns a score to all items $x_i \in X$, and the top-k items are returned as the candidate items for the next basket.

One of the main use cases for the NBR task is grocery shopping, which is also the focus of this paper. Users tend to purchase multiple items at a time, which corresponds to the concept of a shopping basket. Grocery shopping is repetitive; it is usually done on a regular basis. Moreover, many grocery items have a short life time, which leads to repeated purchases of these items [17].

The next basket of a user consists of both *repeat items*, i.e., the items that they have already purchased in the past, and completely new items that they have not purchased previously, i.e., *explore items*. A recent study on state-of-the-art methods in NBR on grocery shopping datasets [16] has shown that NBR methods that are biased towards repetition, i.e., that generate a predicted basket mostly from repeat items, significantly outperform explore-biased approaches. Explore-biased approaches methods are even repeatedly outperformed by the simplest model for NBR, which recommends the most frequently purchased items in a user's history. This calls for further studies aimed at understanding repeat consumption behavior in grocery shopping, as well as designing personalized NBR models that are able to effectively model this behavior.

In this paper, we focus on the following two research questions as a step towards this direction:

RQ1: What are the characteristics of repeat consumption behavior in the grocery shopping domain?

RQ2: How can we design a NBR model that takes the personal repeat behavior of users into account?

In Section 4, we address RQ1 through an empirical study of six grocery shopping datasets. We then propose ReCANet in Section 5, a personalized next basket recommender that models the repeat consumption behavior effectively.

3 RELATED WORK

Next basket recommendation. There is growing interest in research into e-commerce in the information retrieval (IR) community [27]. As one of the core recommendation tasks in e-commerce, the NBR task has gained considerable attention in the recent years, where many studies focus on learning representations for sequence of baskets with neural networks. Yu et al. [30] use recurrent layers and inspired by word2vec in natural language processing, Wan et al. [28] introduce triple2vec for NBR. Correlations between items in baskets are used to recommend more coherent baskets in [14]. An encoder-decoder architecture using recurrent layers is proposed in [12]. Inspired by the transformer architecture, Sun et al. [26] leverage multi-head attention to learn a representation for a sequence of sets. Yu et al. [31] build a co-occurrence graph for items and use graph convolutions to learn item relationships in baskets. A contrastive learning framework is introduced in [20] to denoise basket generation by considering only the relevant items in the history. The focus in these works is on modeling item relations in baskets; personal preferences of users are considered only implicitly.

In another line of work, Hu et al. [13] propose TIFU-KNN, a nearest neighbor-based model for NBR that directly models the personal history of users. The model is shown to perform superior to strong neural network-based baselines designed for the task, demonstrating the importance of repeat behavior, i.e., recommending items that a user has purchased before. Similarly, Faggioli et al. [9] propose UP-CF@r, a simple model that combines the personal popularity with collaborative filtering, leading to strong performance for NBR. A recent survey on NBR models demonstrates the effectiveness of TIFU-KNN and UP-CF@r compared to more complex neural models [16]. While these models consider the frequency and the recency of items bought in the history, repurchasing patterns of items are not reflected in these models.

In this paper, we propose a neural architecture that learns from the personal consumption patterns of users and recommends from the previously purchased items.

Understanding repeat behavior. People are creatures of habit [1, 19]. The tendency to repeat behavior has been uncovered in many different domains, from revisiting places to re-listening to the same songs [1, 4]. Reiter-Haas et al. [22] adopt a psychological theory of human cognition that models the operations of human memory for music re-listening prediction. Chen et al. [6] derive four generic features that influence people's short-term reconsumption behaviors independent of domain and predict whether or not a user will perform a reconsumption at a specific time.

Repeat behavior has also been studied in the context of recommendation. Rappaz et al. [21] show that carefully modeling repeat consumption plays a significant role in achieving state-of-the-art recommendation performance on a video live-streaming platform. Repeat purchase recommendations on the Amazon.com website lead to an over 7% increase in product click through rate on its personalized recommendations page [5]. Ren et al. [23] propose a model with an encoder-decoder architecture for session-based recommendation, where the encoder has a separate component for modeling repeat consumption. Finally, Wang et al. [29] combine collaborative filtering with temporal point processes to recommend novel items and consumed items in a sequential recommendation setting.

What we add to this literature is a next basket recommendation model that focuses on recommending items from the users' past consumption history in the grocery shopping domain.

4 EMPIRICAL STUDY

In this section, we study the characteristics of repeat consumption behavior in grocery shopping. We describe six datasets that contain transaction data from online and physical grocery stores. We then analyze the repeat behavior across users, baskets, and items. We conclude this section by establishing an upper bound on the predictive performance for repeat-focused recommendation.

4.1 Dataset description

We use six grocery shopping datasets in this paper. The datasets are described below and their statistics are summarized in Table 1. We leverage three types of datasets: "online" and "offline" refer to online shops and physical stores, respectively. "Multi-channel" is the case where users purchase goods from both online and offline shopping channels. Among them, three are publicly available: Dunnhumby,[1] ValuedShopper,[2] and Instacart.[3] We further use three proprietary datasets from a large food retailer in Europe, namely X-online, X-offline, and X-multi-channel.

All datasets contain transactions: which items are bought by which user at which time. All items bought in the same transaction are treated as a basket. In all datasets, we remove users who have less than three baskets and items that occur in less than five baskets in total. For every user, we sort the baskets by time. The last basket of every user is reserved for testing or validation, while the rest are treated as the training data. Table 1 shows the statistics for the training data after preprocessing.

"Avg. item per user" shows the number of unique items that a user has interacted with, averaged over all users. "Personal item ratio (%)" is the average number of items per user divided by the total number of items in the dataset, in percentage. "Avg. repeat ratio" shows the fraction of items in the last training basket of the users that have already appeared in their past, averaged over users. "Avg. explore ratio" shows the opposite, the fraction of new items.

The datasets vary in different aspects. In terms of history per user, Dunnhumby and ValuedShopper are very rich, with more than a hundred baskets per user on average. Instacart, X-offline and X-multi-channel have much less history for users, ranging from 13 to 16 basket per user on average, and X-online is scarce with only 6 basket per user on average. The number of users varies across datasets as well, with Instacart being the largest with 197K users and Dunnhumby being the smallest with 2K users. The basket size is similar for all datasets, except X-online and X-multi-channel which have larger basket sizes than the rest. The total number of items does not vary a lot across datasets, ranging from 13K to 47K.

[1] https://www.dunnhumby.com/careers/engineering/sourcefiles
[2] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/overview
[3] https://www.kaggle.com/c/instacart-market-basket-analysis

Table 1: Dataset statistics after preprocessing.

Type	Dataset	Users	Items	Baskets	Avg. item per basket	Avg. basket per user	Avg. item per user	Personal item ratio (%)	Avg. repeat ratio	Avg. explore ratio
Online	Instacart	197,523	47,389	3,122,453	10.08	16	65	0.14	0.60	0.40
	X-online	55,528	18,570	360,672	41.84	6	161	0.87	0.58	0.42
Offline	Dunnhumby	2,483	36,963	270,416	9.15	109	524	1.42	0.49	0.51
	ValuedShopper	9,601	13,271	1,068,285	8.87	111	371	2.80	0.71	0.29
	X-offline	62,789	23,379	791,971	11.15	13	96	0.41	0.36	0.64
Multi-channel	X-multi-channel	94,912	24,693	1,286,689	23.13	14	197	0.80	0.46	0.54

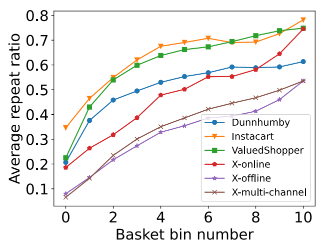

Figure 1: Distribution of repeat ratio in the baskets of users for all datasets.

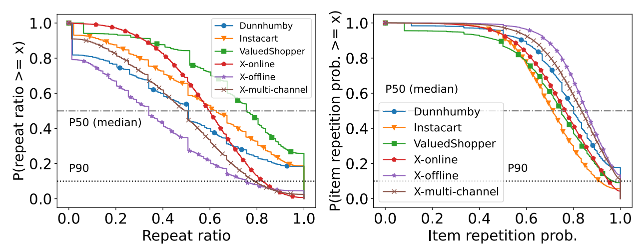

Figure 2: Cumulative distribution of repeat ratio in the last basket of users (Left) and repetition probability of items (Right) in all datasets.

The average repeat ratio differs between datasets, with ValuedShopper having the highest and X-offline having the lowest values. However, even the lowest repeat ratio is over 36%. Personal item ratio is less than 3% for all datasets, which shows that from all of the items that are present in the inventory, every individual user is only concerned with a very small fraction of them. Interestingly, this is also the case for the datasets with a rich history for users, such as Dunnhumby and ValuedShopper: even when a user's purchase history covers a long period, the number of unique items they interact with does not grow that much.

4.2 Characteristics of repeat consumption

In this section, we study the repeat consumption behavior in the grocery shopping datasets introduced above. We aim to gain insights that show what a recommendation model should focus on when using user and item data. In this section, we only consider the training baskets.

Repeat ratio across baskets. Repeat ratio in a basket is defined as the fraction of items in that basket that have already been purchased by the same user in the previous baskets. We study the repeat ratio in users' baskets to understand the stability of repeat consumption across time. Specifically, we are interested in analyzing how the repeat ratio changes as more baskets are added to the history of a user. For each user, we distribute their baskets uniformly in 10 bins, where the smaller bin numbers correspond to the earliest baskets in the user's history. Given B_t^u as a basket in user u's history, where t indicates the index in the history sequence and n is the total number of u's baskets, the bin number for B_t^u is defined as $\lceil \frac{t}{bl_u} \rceil$, where bl_u corresponds to the bin length for u, defined as $\lceil \frac{n}{10} \rceil$.

Figure 1 shows the distribution of the averaged repeat ratio for each bin across datasets. As expected, the repeat ratio grows as more baskets are added to the history. For the Instacart, Dunnhumby and ValuedShopper datasets, we see a jump at the start of the history. The jump at the start shows that after the first baskets of the user, when sufficient history has been gathered, the repeat ratio stabilizes to some extent. That is, when users have purchased goods for some time, their history is rich enough for a repeat-based recommendation. On the X-datasets, the repeat ratio is still increasing and has not reached the plateau point yet. This might be the result of the minimum preprocessing and preselection of users that we performed when creating these datasets, in contrast to the ones that are made public by others.

The repeat ratio reaches its highest value in the final baskets of the users. This means that the last baskets of the users in the history contain more repeat items than the earlier ones. This shows the importance of considering the recency: the next basket we aim to predict is closer to the last baskets of the users, in terms of repeat ratio, than to the early baskets.

Repeat ratio across users. To analyze the repeat ratio across users, we consider the repeat ratio in their last basket in the training data, which is the temporally closest one to the future basket we aim to predict. The cumulative distribution of repeat ratio in the last basket of training data is shown in Figure 2 (Left). The distributions differ across datasets. On the Instacart, Dunnhumby and ValuedShopper datasets, 20 to 25 percent of users have a repeat ratio of 1: their last basket is entirely composed of items that they have already purchased before. On the ValuedShopper dataset, half of the users have a repeat ratio of more than 0.75, and for only 20 percent of the users the repeat ratio is less than 0.50. Instacart follows the same trend, with slightly lower values for the repeat ratio. On the other hand, in the X-offline and Dunnhumby datasets, around 20 percent of the users have a repeat ratio of zero: these are the users who will not benefit from repeat-based recommendations. This percentage is less than 10 for the other datasets. X-online has the smoothest distribution, with no long tail on either side of the spectrum.

Overall, we observe that not all users benefit to the same degree from a personalized recommendation, which is expected: users differ in terms of amount of history that we have for them, and they differ in their tendency for repeat consumption. However, even for the X-offline dataset as the most challenging dataset, over 30% of the last basket can be predicted based on the history, for more than 50% of the users.

Repeat ratio across items. So far, we have analyzed the repeat behavior from the users' perspective and across baskets. Another aspect to consider for repeat behavior is the item dimension. Items differ in their tendency to be repurchased. We define the *item repetition probability* as the number of users who have repurchased the item (a.k.a. purchased the item more than once) divided by the number of users who have purchased the item only once.

Figure 2 (Right) shows the cumulative distribution of the probability of an item being repurchased for all items across datasets. Unlike the analysis on baskets and users, where the repeat ratio varied a lot across datasets, we observe very similar distributions for items. On all datasets, less than 10% of the items have a repetition probability of less than 0.5. The median ranges between 0.7 to 0.9, which means that most of the items in the grocery shopping datasets will be repurchased with a high probability.

4.3 Upper bound on predictive performance for personalized next basket recommendation

Our study in Section 4.2 shows that the users' repeat behavior becomes more dominant as time passes. Furthermore, while being different for different users, repeat items account for a considerable fraction of items in baskets even when the average repeat ratio is low. Moreover, most of the items in grocery shopping datasets are repurchased frequently. In this section, we analyze how the repeat consumption behavior relates to the next basket recommendation problem. To this end, we consider a personal NBR model that only focuses on items that a user has purchased before. That means that instead of ranking all of the items in the dataset, the model only assigns scores to the users' previously purchased items.

What is the maximum performance of a personal NBR model? We design an oracle personal NBR model that achieves this performance as follows: given the items purchased in the history $B^u = [B^u_1, B^u_2, \ldots, B^u_n]$ of user u, the oracle recommends the ones that occur in B^u_{n+1}. Table 2 shows the performance of the oracle model on different datasets, in terms of Recall@$|B|$ and nDCG@$|B|$, where $|B|$ is the size of the target basket B^u_{n+1}. Recall@$|B|$ measures the fraction of items in the target basket that are present in the top $|B|$ items, and nDCG measures how high in the top $|B|$ list the relevant items are ranked. Repeat items, which account for less than 3% of all items in all datasets, are responsible for 38 to 62% of the oracle performance in NBR. The remaining 97% of items contribute to the oracle performance in NBR similarly: they account for the remaining 38 to 62% of performance. Averaged over all datasets, 54% of the oracle performance in NBR in terms of Recall@$|B|$ comes from only 1% of items! This indicates the importance of the repeat items, and the potential of a personalized recommendation model that is focused on those. Next, we introduce our proposed recommendation model that builds on these insights.

Table 2: Performance of the oracle personal NBR model.

Type	Dataset	Recall@$\|B\|$	nDCG@$\|B\|$
Online	Instacart	0.6087	0.6945
	X-online	0.6214	0.7236
Offline	Dunnhumby	0.4488	0.5331
	ValuedShopper	0.7168	0.7815
	X-offline	0.3811	0.4843
Multi-channel	X-multi-channel	0.4959	0.6055

5 A PERSONALIZED NEXT BASKET RECOMMENDATION MODEL

Based on insights from our empirical study, we design a network that is focused on repeat items and that models the personal consumption patterns in a user's history. We introduce ReCANet, a repeat consumption-aware neural network.

5.1 Architecture overview

The goal of the personalized NBR task is to decide which of the items that a user consumed in the past will appear in the next basket. We model it as a binary classification task, where the labels "positive" and "negative" correspond to an item being present in the target basket or not, respectively.

We design a neural model for this problem, whose architecture is shown in Fig. 3. The intuition is that the repurchase of an item is predictable from the past consumption pattern of the item by this user, other users, and other items. To model this, the network takes three inputs: item id x, user id u, and history vector with the length of window size w, corresponding to the consumption pattern of x by u. Formally, given user u, for every $x \in X_u$ where X_u is the set of all items user u has purchased in their history, ReCANet produces p^u_x, which is the probability of item x being in the next basket of u. At the first layer, item id and user id are embedded separately, concatenated, and fed to a feed-forward layer to create a user-item representation. The combination of the user-item representation and the history vector is then fed to a stack of LSTM layers, which models the temporal consumption pattern of item x by user u. The output of the last LSTM layer is the input for a stack of feed-forward layers, with a sigmoid activation at the last layer which generates the probability of item x appearing in the target basket.

5.2 Architecture details

The goal of ReCANet is to predict the probability of an item appearing in the next basket of a user. As explained above, to model this, ReCANet utilizes different components, which we will now discuss in detail.

Inputs. The network takes three inputs: (1) An item id x, which is a unique identifier for each item $x \in X$; (2) A user id u, similarly a unique identifier for each user $u \in U$; (3) A history vector $\vec{h} \in \mathbb{R}^w$, where window size w is a hyper-parameter of our model.

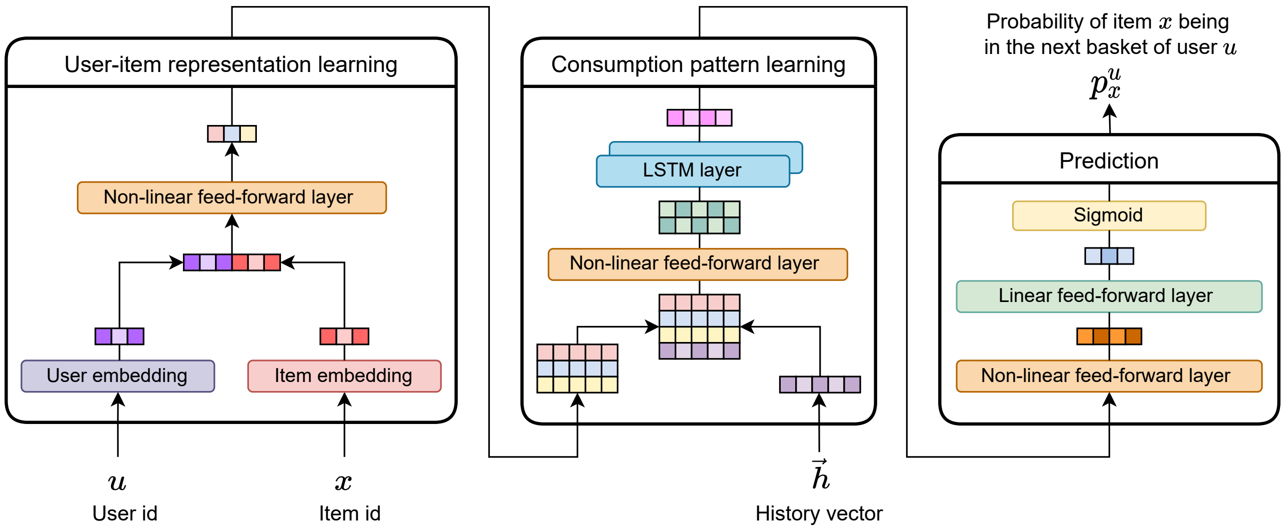

Figure 3: Overview of ReCANet's architecture.

The history vector $\vec{h}$ is defined in a way to reflect the consumption history of item x by user u. As we aim to predict the occurrence of x in basket B^u_{n+1}, we consider the last w baskets in $[B_1, \ldots, B_n]$ that contain x. $[b_1, b_2, \ldots, b_w]$ denote the indices in the basket history sequence, that is $1 \le b_i \le n$. For $0 \le i \le w-1$, the i-th element in $\vec{h}$ is equal to $b_{i+1} - b_i$. For $i = w$, the i-th element in $\vec{h}$ is equal to $(n+1) - b_w$. If the number of baskets that contain x are less than w, the corresponding elements in $\vec{h}$ are filled with zero.

As an example, assume that user u has 99 baskets in their history, and we want to predict whether item x appears in the 100th basket. We know that x has appeared in baskets $\{15, 29, 45, 78, 95, 97, 99\}$. Given a window size of $w = 5$, vector $\vec{h}$ is equal to $[78-45, 95-78, 97-95, 99-97, 100-99] = [33, 17, 2, 2, 1]$.

The history vector $\vec{h}$ has three important properties: (1) it contains the frequency of consuming the item, as the number of non-zero elements in the vector are the representative of the purchase frequency; (2) it contains the recency information, since it is a sequence and the index in the sequence indicates how far in the history the item has been consumed; and (3) it contains information about the consumption pattern, as the values in the vector demonstrate the gaps between consecutive purchases of the item.

User item representation learning. Our goal is to predict the appearance of item x in the next basket of user u. As a result, we aim to model the user-item relationship. We separately embed users and items, and then combine the embeddings. Assume $W_i \in \mathbb{R}^{|I| \times d_i}$ is the trainable embedding matrix for items and $W_u \in \mathbb{R}^{|U| \times d_u}$ is the trainable embedding matrix for users, where d_i and d_u are the embedding dimensions. Given x and u as item and user identifiers, the corresponding rows in the embedding matrices denote the item embedding $e_x \in \mathbb{R}^{d_i}$ and user embedding $e_u \in \mathbb{R}^{d_u}$, respectively. These embeddings allow the network to learn item and user representations that keep similar items and similar users close in the embedding space. The final user-item representation $e_{ui} \in \mathbb{R}^m$ is obtained as follows:

$$e_{ui} = \mathrm{ReLU}((e_u \oplus e_i) \cdot W_1 + b_1),$$

where $W_1 \in \mathbb{R}^{(d_i+d_u) \times m}$ and $b_1 \in \mathbb{R}^m$ are trainable parameters, ReLU is the activation function, and m is a hyper-parameter of our model that denotes the hidden layer size, i.e., the model size. $\oplus$ denotes concatenation.

Consumption pattern learning. The history vector $\vec{h} \in \mathbb{R}^w$ contains the consumption pattern of item x by user u. In order to predict the repurchase probability, we first need to combine the user-item representation with the history vector. To this end, we replicate e_{ui} w times to obtain $e^w_{ui} \in \mathbb{R}^{w \times m}$. The combination representation $c_{ui} \in \mathbb{R}^{w \times m}$ is the computed as follows:

$$c_{ui} = \mathrm{ReLU}((e^w_{ui} \oplus_r \vec{h}) \cdot W_2 + b_2), \tag{1}$$

where $\oplus_r$ denotes row-wise concatenation; that is, each element of $\vec{h}$ is concatenated with the corresponding row in e^w_{ui} (i.e., $(e^w_{ui} \oplus_r \vec{h}) \in \mathbb{R}^{w \times (m+1)}$). $W_2 \in \mathbb{R}^{(m+1) \times m}$ and $b_2 \in \mathbb{R}^m$ are trainable parameters. c_{ui} is a sequence of length w that contains information from both the user-item representation and the consumption history vector. To model sequential information, we use a stack of two LSTM layers. Formally:

$$\begin{aligned}
f_{j_t} &= \mathrm{Sigmoid}(W_{f_j} x_t + U_{f_j} h_{t-1_j} + b_{f_j}) \\
i_{j_t} &= \mathrm{Sigmoid}(W_{i_j} x_t + U_{i_j} h_{t-1_j} + b_{i_j}) \\
o_{j_t} &= \mathrm{Sigmoid}(W_{o_j} x_t + U_{o_j} h_{t-1_j} + b_{o_j}) \\
\hat{c}_{j_t} &= \mathrm{Tanh}(W_{c_j} x_t + U_{c_j} h_{t-1_j} + b_{c_j}) \\
c_{j_t} &= f_{j_t} \cdot c_{j_{t-1}} + i_{j_t} \cdot \hat{c}_{j_t} \\
h_{j_t} &= o_{j_t} \cdot \mathrm{Tanh}(c_{j_t}),
\end{aligned} \tag{2}$$

where $j \in \{1, 2\}$ denotes the first and the second LSTM layer, respectively; $1 \le t \le w$ indexes the time step; $x_t \in \mathbb{R}^m$ denotes the input of the LSTM, which is equal to the corresponding time stamp in c_{ui} for the first layer, and the corresponding time stamp in cell state vector of the first layer (i.e., c_{1_t}) for the second layer.

$W_{f_j}, W_{u_j}, W_{o_j}, U_{f_j}, U_{u_j}, U_{o_j}, b_{f_j}, b_{u_j}, b_{o_j}$ are weight matrices and bias vector parameters that will be learned during training. The hidden state of the second layer, i.e., $h_{w_2} \in \mathbb{R}^m$, is used as the input for the prediction layers.

Prediction. We apply a two-layer feed-forward network with ReLU activation in between and a sigmoid activation at the end to produce the probability of item x appearing in the next basket of user u, with trainable parameters $W_{o_1}, W_{o_2} \in \mathbb{R}^{m \times m}$ and $b_{o_1}, b_{o_2} \in \mathbb{R}^m$:

$$p_x^u = \text{Sigmoid}(\text{ReLU}(h_{w_2} \cdot W_{o_1} + b_{o_1}) \cdot W_{o_2} + b_{o_2}). \qquad (3)$$

5.3 Learning process

Training. We use the training baskets for creating training samples. For each user, we consider their last L baskets, where L is a parameter than can be set per user or per dataset. We set it per dataset. The intuition behind truncating the baskets is two-fold: (1) as shown in our empirical study, the last baskets of users are more similar to their upcoming basket in terms of repeat consumption behavior than their earlier baskets; and (2) the number of baskets per user varies, and this may result in unfair presence of users in training data. Limiting the number of baskets per user helps to mitigate this problem. For each basket B_t in the last L baskets of a user, the items that have been purchased in the previous $t-1$ baskets are the candidates for recommendation. Each item along with the user and its consumption history make up one training sample. To create labels for the binary classification task, the items that occur in B_t are the positive samples; the rest are negative ones. We use the binary cross-entropy loss for training the network.

Testing. The last basket of each user is used for testing or validation. The inputs of the network are generated with the same procedure as the training phase, where all items in the training baskets of the user are recommendation candidates. For each user, the items are then ranked based on the output of the network, and the top-k items are returned as the predictions for the next basket.

6 EXPERIMENTAL SETUP

We introduce our baselines, data split scheme, evaluation metrics, parameter setting and implementation details.

Baselines. In addition to two simple baselines (P-POP, GP-POP), we compare ReCANet with two neighborhood-based models (TIFU-KNN, UP-CF@r), and one neural network (DNNTSP). These models have been shown to outperform other methods in NBR for grocery shopping and achieve state-of-the-art performance [16]:[4]

- P-POP: Recommends the most popular items in the user history, sorted by frequency of purchases. In NBR, P-POP is considered one of the strongest baselines.
- GP-POP: First uses P-POP to fill the predicted basket. If there are remaining empty slots (a.k.a., the basket size is larger than the number of items in the user's history), they will be filled with the most popular items in total.
- TIFU-KNN [13]: A nearest neighbor-based model that has been shown to outperform deep recurrent neural networks. The model relies on the similarity of the target user with other users and the history of the target user.
- UP-CF@r [9]: A collaborative filtering-based approach based on user-wise item popularity which also considers the recency of purchases.
- DNNTSP [31]: A deep neural network model that learns item relationships by constructing a co-occurrence graph and performs graph convolutions on the dynamic relationship graphs.

Data split. We follow the same procedure as Faggioli et al. [9]. The training data is composed of all baskets but the last one of all users. We randomly select 50% of the last baskets for the validation set, and the remaining ones for the test set. The validation set is used to select the best hyper-parameters of the methods, and the final results are reported on the test set.

Evaluation metrics. Following [13, 31], we use Recall@k and nDCG@k for evaluation. Recall is widely used in NBR, measuring how many of the items in the target basket are present in the predicted items for the next basket. nDCG is a ranking-based measure that takes into account the order of elements. Both measures are calculated across the predicted next basket for all test users. We report the metrics for $k \in \{10, |B|\}$, where $|B|$ stands for the length of the test basket B. We use 10 because there are limits for showing recommendations to users. We use $|B|$ to be able to compare performance across varying basket sizes and different datasets.

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. We set the model size m to 64, sweep the embedding sizes d_u and d_i in $\{16, 32, 64, 128\}$ and the window size w in $\{5, 15, 25, 35, 45\}$. The hyper-parameters of the baselines are either tuned or set according to instructions in the papers if available. For TIFU-KNN we use the best parameters reported in the original paper [13] for the Instacart and ValuedShopper datasets. For UP-CF@r, we use the reported parameters on the Instacart and Dunnhumby dataset [9]. We tune these models on the rest of the datasets. DNNTSP is used with the parameter setting stated in [31] for all datasets, as there are no instructions for tuning.

Implementation details. ReCANet is implemented using the Keras framework with TensorFlow as backend. We use mini-batch stochastic gradient descent (SGD) together with the Adam optimizer to train the models. We set the batch size to 2048, and stop training when the loss on the validation set converges. To remove the random initialization effect, we run our model 10 times and report the average results. It is worth mentioning that the standard deviation is less than 0.001 in all cases. For users who have more than 100 baskets in the training data, we only keep their last 100 baskets for creating the training samples ($L = 100$), except on the ValuedShopper dataset, where we set the threshold to 50. We make our code public to facilitate reproducibility and follow up research.[5]

7 RESULTS AND DISCUSSION

7.1 Overall performance

Table 3 (left-hand side) shows the results of ReCANet alongside state-of-the-art baselines on all datasets. We observe that the performance of P-Pop and GP-Pop is identical in most cases, and in cases where a difference is observed (all metrics on Instacart and

[4] While a large number of NBR methods have been proposed in recent years, these are all outperformed by the methods that we have selected as baselines; see [13, 16].

[5] https://github.com/mzhariann/recanet

Table 3: Results of ReCANet compared against the baselines (left-hand side), as well as ablation study results (right-hand side). Boldface and underline indicate the best and the second best performing model among NBR models, respectively. Significant improvements of ReCANet over the best baseline results are marked with † (paired t-test, p < 0.05). ▲% shows the improvements of ReCANet against the best baseline. ▼% shows the drop in performance compared to ReCANet.

Dataset	Metric	Baselines					Ours				
		P-Pop	GP-Pop	TIFU-KNN	UP-CF@r	DNNTSP	ReCANet (▲%)	w/o user (▼%)	w/o item (▼%)	w/o user and item (▼%)	w/o history (▼%)
Instacart	Recall@10	0.3222	0.3230	0.3434	0.3400	0.3427	**0.3592**† (4.6)	0.3585 (0.2)	0.3507 (2.4)	0.3506 (2.4)	0.3135 (12.7)
	nDCG@10	0.3131	0.3134	0.3332	0.3319	0.3336	**0.3502**† (5.0)	0.3496 (0.2)	0.3406 (2.7)	0.3407 (2.7)	0.2976 (15.0)
	Recall@$\lvert B\rvert$	0.2901	0.2904	0.3115	0.3077	0.3079	**0.3295**† (5.8)	0.3291 (0.1)	0.3203 (2.8)	0.3202 (2.8)	0.2807 (14.8)
	nDCG@$\lvert B\rvert$	0.3299	0.3301	0.3521	0.3479	0.3480	**0.3717**† (5.6)	0.3712 (0.1)	0.3613 (2.8)	0.3612 (2.8)	0.3172 (14.7)
X-online	Recall@10	0.1603	0.1603	0.1443	0.1619	0.1601	**0.1645**† (1.6)	0.1632 (0.8)	0.1609 (2.2)	0.1598 (2.9)	0.1155 (29.8)
	nDCG@10	0.6283	0.6283	0.5771	0.6361	0.6334	**0.6462**† (1.6)	0.6409 (0.8)	0.6285 (2.7)	0.6238 (3.5)	0.4604 (28.8)
	Recall@$\lvert B\rvert$	0.3613	0.3613	0.3009	0.3680	0.3679	**0.3735**† (1.5)	0.3712 (0.6)	0.3637 (2.6)	0.3626 (2.9)	0.2997 (19.8)
	nDCG@$\lvert B\rvert$	0.4323	0.4323	0.3731	0.4396	0.4391	**0.4467**† (1.6)	0.4436 (0.7)	0.4342 (2.8)	0.4323 (3.2)	0.3441 (23.0)
Dunnhumby	Recall@10	0.1315	0.1315	0.1485	0.1421	0.0892	**0.1621**† (9.2)	0.1615 (0.4)	0.1544 (4.8)	0.1526 (5.9)	0.1330 (18.0)
	nDCG@10	0.1267	0.1267	0.1379	0.1364	0.0825	**0.1489**† (8.0)	0.1488 (0.1)	0.1447 (2.8)	0.1417 (4.8)	0.1206 (19.0)
	Recall@$\lvert B\rvert$	0.1048	0.1048	0.1244	0.1173	0.0576	**0.1310**† (5.3)	0.1337 (-2.1)	0.1226 (6.4)	0.1250 (4.6)	0.1067 (18.5)
	nDCG@$\lvert B\rvert$	0.1216	0.1216	0.1423	0.1346	0.0685	**0.1503**† (5.6)	0.1529 (-1.7)	0.1412 (6.1)	0.1428 (5.0)	0.1230 (18.2)
Valued-Shopper	Recall@10	0.2030	0.2030	0.2194	0.2161	0.1927	**0.2311**† (5.3)	0.2309 (0.1)	0.2264 (2.0)	0.2257 (2.3)	0.2007 (13.2)
	nDCG@10	0.2006	0.2006	0.2142	0.2096	0.1954	**0.2228**† (4.0)	0.2221 (0.3)	0.2209 (0.9)	0.2193 (1.6)	0.1949 (12.5)
	Recall@$\lvert B\rvert$	0.1735	0.1735	0.1853	0.1822	0.1647	**0.1958**† (5.7)	0.1966 (-0.4)	0.1933 (1.3)	0.1925 (1.7)	0.1650 (15.7)
	nDCG@$\lvert B\rvert$	0.2032	0.2032	0.2174	0.2134	0.1929	**0.2282**† (5.0)	0.2287 (-0.2)	0.2264 (0.8)	0.2244 (1.7)	0.1958 (14.2)
X-offline	Recall@10	0.1585	0.1587	0.1136	0.1644	0.1298	**0.1704**† (3.6)	0.1692 (0.7)	0.1597 (6.3)	0.1598 (6.2)	0.1342 (21.2)
	nDCG@10	0.2006	0.2007	0.1345	0.2055	0.1964	**0.2120**† (3.2)	0.2113 (0.3)	0.2009 (5.2)	0.2006 (5.4)	0.1575 (25.7)
	Recall@$\lvert B\rvert$	0.1569	0.1570	0.1090	0.1619	0.1593	**0.1675**† (3.5)	0.1669 (0.4)	0.1579 (5.7)	0.1582 (5.6)	0.1280 (23.6)
	nDCG@$\lvert B\rvert$	0.1882	0.1883	0.1282	0.1933	0.1899	**0.1998**† (3.4)	0.1990 (0.4)	0.1895 (5.2)	0.1893 (5.3)	0.1496 (25.1)
X-multi-channel	Recall@10	0.1170	0.1170	0.0797	0.1181	0.1064	**0.1203**† (1.9)	0.1203 (0.0)	0.1166 (3.1)	0.1174 (2.4)	0.0826 (31.3)
	nDCG@10	0.3628	0.3628	0.2375	**0.3673**	0.3607	0.3648 (-0.7)	0.3683 (-1.0)	0.3509 (3.8)	0.3548 (2.7)	0.2587 (29.1)
	Recall@$\lvert B\rvert$	0.2132	0.2133	0.1634	0.2172	0.2152	**0.2214**† (1.9)	0.2218 (-0.2)	0.2128 (3.9)	0.2144 (3.2)	0.1685 (23.9)
	nDCG@$\lvert B\rvert$	0.2574	0.2575	0.1848	0.2616	0.2582	**0.2642**† (1.0)	0.2656 (-0.5)	0.2541 (3.8)	0.2564 (3.0)	0.1954 (26.0)

X-offline, Recall@$|B|$ and ndCG@$|B|$ on X-multi-channel) GP-Pop outperforms P-Pop with a very small margin that is not significant. This means that given 10 and $|B|$ as cut-off points for the recommendation list, users have enough items in their history that will surpass the threshold. However, the differences will likely be more substantial if larger thresholds are used, as suggested in [16].

The performance of P-Pop is the lowest in most cases, but it is in the same range as state-of-the-art methods. This result once again confirms the importance of personal history in next basket recommendation. P-Pop as the simplest method for NBR should always serve as a baseline when a new model is proposed. This has not always been the case; in [20], for example, the authors do not compare their proposed method with P-Pop, and it has later been shown that P-Pop outperforms it in most cases [16].

DNNTSP is not the best baseline for any of the datasets and metrics, except for nDCG@10 on Instacart. The performance in all cases is in the same range with other methods, except on Dunnhumby, where there is a large gap between DNNTSP's performance and the others. We suspect that this is a result of the extremely low number of users in Dunnhumby, compared to the other datasets. DNNTSP has been tested on datasets with 10k to 100k users in the original paper [31], and additional measures such as hyper-parameter tuning might be required for a dataset with small number of users.

TIFU-KNN and UP-CF@r are the best performing baselines on all datasets and metrics, except for one case (nDCG@10 on Instacart). Both methods are nearest neighbor-based, and show superior performance to the state-of-the-art neural approach DNNTSP. These results are in line with findings in previous work [13, 16]; modeling the personal history is at least as effective as basket representation learning, if not more. The performance of UP-CF@r is more stable across datasets than TIFU-KNN; in cases where it is not the best baseline (Instacart, Dunnhumby, ValuedShopper) its performance does not differ significantly from TIFU-KNN. On the other hand, TIFU-KNN's performance drops on X datasets and even falls short of P-Pop. We suspect that this a result of non-optimal hyperparameters on these datasets. The ranges for tuning the parameters given in the original paper are for Dunnhumby (but another version of the dataset), Instacart and ValuedShopper [13].

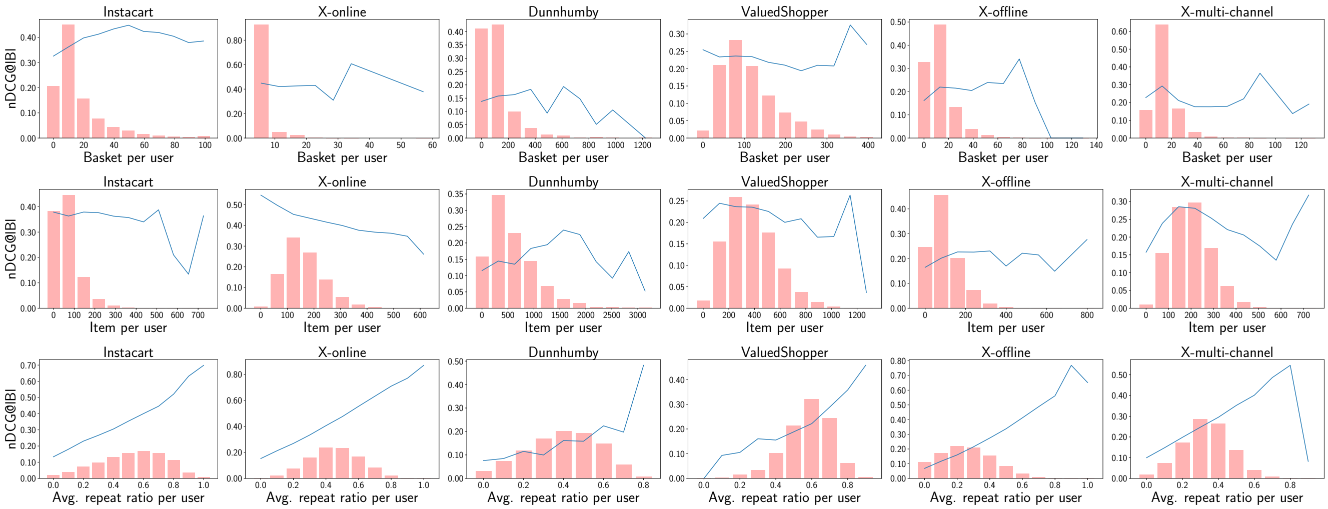

Figure 4: Performance of ReCANet in terms of nDCG@$|B|$ based on (top) the number of baskets per user, (center) the number of unique items per user, and (bottom) the average repeat ratio in users' history. Bars show the distribution of users. Performance correlates with the average repeat ratio, but is mostly robust for different basket and unique item sizes.

ReCANet significantly outperforms the best baseline in all datasets and across all metrics, except for one case (nDCG@10 on X-multi-channel), where it performs the same as UP-CF@r. The improvements range from 1.0% to 9.2%. X-online and X-multi-channel are the datasets with the lowest performance gains compared to the best baseline. These are the datasets with a large average basket size (42 and 23, compared to around 10 on other datasets). This indicates that predicting larger baskets is more challenging for ReCANet. Improving the ranking component of the network could mitigate this problem, which is left as future work.

7.2 Ablation study

We design an ablation study to investigate the effect of different components of ReCANet and compare it with four variations: (1) w/o user: the user embedding layer is omitted, and the item embedding alone is combined with history vector; (2) w/o item: the item embedding layer is omitted, and the user embedding alone is combined with history vector; (3) w/o user and item: the user-item representation learning module is omitted; and (4) w/o history: the consumption pattern learning module is omitted.

Table 3 (right-hand side) shows the results. ReCANet w/o history has the lowest performance compared to the other versions, across all metrics and datasets. Omitting the consumption pattern learning module results in drops in performance ranging from 12.7% to 31.3%. This indicates that the history vector contains valuable information for next basket predication, and the designed module is effective in utilizing this information. The smallest effect on performance comes from the user embedding layer, as illustrated in the results of ReCANet w/o user. The drops in performance are small, ranging from 0% to 0.8%. In some cases ReCANet w/o user even outperforms ReCANet, albeit with small margins. The user information is implicitly encoded in the model, in the way that we create the training and test data. That is, the positive and negative samples come from the personal history of users in the training data, and the personal items are ranked during testing. This results in a reduced need for explicitly modeling users via the user embedding layer. Item embeddings have a positive effect on performance across all metrics and dataset; ReCANet w/o item results in 0.9% to 6.4% drops in performance. Hence, modeling item information is important for the NBR task. Omitting the user-item representation learning module all together has a similar effect on performance. As expected, in cases where user information is not helpful, the effect is lower than omitting the item embedding alone. In other cases where user information helps the performance, the effect of the user-item representation learning is more apparent.

In summary, all components of ReCANet contribute to its performance, and are essential to achieve the best overall performance.

7.3 User-level analysis

Table 3 shows the performance of ReCANet averaged over all users. In this section, we analyze the treatment effect for different groups of users [24]. We group users in three ways: (1) by the number of baskets per user, (2) by the number of unique items per user, and (3) by the average repeat ratio per user. Do different groups of users benefit from ReCANet in different degrees?

Fig. 4 (top) shows the performance of ReCANet in terms of nDCG@$|B|$ w.r.t. the number of baskets per user for each dataset. The distribution of users based on their number of baskets is also shown with bars. All datasets are right-skewed; most of the users have relatively small number of baskets in their history. The performance shows no correlation with the number of baskets, and is robust across users. Sudden drops in performance on the Dunnhumby and X-offline datasets only appear at the far right side of the distributions, where the number of users is very low.

Fig. 4 (center) shows the performance of ReCANet in terms of nDCG@$|B|$ w.r.t. to the number of unique items per user for each dataset. Similar to the number of baskets, the distribution of users w.r.t. to the number of unique items has a right-skewed shape, but it is closer to the normal distribution on X-online, ValuedShopper and X-multi-channel. The performance is robust w.r.t. the number of items on Instacart and X-offline, but there is a drop on Instacart for a small number of users in the tail. On Dunnhumby, ValuedShopper and X-multi-channel, the performance increases with the increase in the number of items and decreases after a pick around 1500, 200, and 200 items, respectively. When the number of unique items per user is very small, the performance is low due to the fact that the user is still exploring; their history is not rich enough for personalized recommendation. When the number of items passes the peak point, the performance decreases; for users with many unique items, the final ranking is more subject to noise. ReCANet performs better for users with a higher number of unique items, as it results in a more accurate estimation of the users' taste.

Fig. 4 (bottom) shows the performance of ReCANet in terms of nDCG@$|B|$ w.r.t. to the average repeat ratio in the training baskets per user for each dataset. The distribution of users is rather close to the normal distribution in all datasets; most of the users have a medium repeat ratio, with smaller numbers of users with a very high or very low repeat ratio. In all datasets, performance of ReCANet has a clear correlation with the repeat ratio; the more a user tends to repurchase, the bigger benefit they will get from the personalized recommendations. This also means that the average repeat ratio can serve as predictor for the expected performance of ReCANet. In that case, one can decide to explore non-personal items for recommendation to users with low repeat ratios.

7.4 Parameter sensitivity

We analyze the performance of ReCANet w.r.t. to three hyper-parameters: window size w, user embedding size d_u and item embedding size d_i. On all datasets, the performance is most sensitive to the window size. Small and large window sizes both degrade the performance; 15 and 25 result in the highest performance. This means that too little history makes it hard for the model the learn the consumption patterns, and too much history adds noise to the data that matters most. In contrast, the performance is not sensitive to the user and embedding size; the performance only changes marginally (less than 0.1%) when increasing the dimension size from 16 to 128. This indicates that we can use a small embedding size, which results in a smaller model size and less training time without loss of performance.

8 CONCLUSION

In this paper, we have analyzed the repeat consumption behavior in grocery shopping. We have found that repeat items, i.e., items that a user has previously purchased, have a high chance of reappearing in users' future baskets. We have focused on the next basket recommendation (NBR) problem, and proposed ReCANet, a neural model for the task that learns from users' personal consumption patterns of items. Our experiments show that ReCANet, while focused on the repeat items that make up a small percentage of the total items in the inventory, is able to outperform state-of-the-art NBR models. This means that the repeat consumption behavior in grocery shopping is a strong indicator for future purchases, and explicitly modeling it leads to improvements in the recommendation performance.

Our experiments further show that not all users benefit to the same degree from the repurchase recommendations; the recommendation performance correlates with the average repeat ratio in the previous baskets of the users. In future work, we aim to extend ReCANet to help such users in discovering new items, while modeling the consumption behavior of repeat items at the same time. Moreover, while ReCANet significantly outperforms the best baselines, there is still a gap between ReCANet's performance and the performance of an oracle personalized NBR model. This indicates that the repeat-focused recommendation task is far from solved and there is still room for improvement. Currently, item relations in baskets are only considered implicitly and through the training sample generation procedure in ReCANet. In future work, we aim to add a basket representation learning component, in the history baskets modeling phase as well as the final basket generation phase.

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] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The Dynamics of Repeat Consumption. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014.* ACM, 419–430. https://doi.org/10.1145/2566486.2568018

[2] Mozhdeh Ariannezhad, Sami Jullien, Pim Nauts, Min Fang, Sebastian Schelter, and Maarten de Rijke. 2021. Understanding Multi-channel Customer Behavior in Retail. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021.* ACM, 2867–2871. https://doi.org/10.1145/3459637.3482208

[3] Oren Barkan and Noam Koenigstein. 2016. Item2vec: Neural Item Embedding for Collaborative Filtering. In *Proceedings of the Poster Track of the 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, USA, September 17, 2016 (CEUR Workshop Proceedings, Vol. 1688).* CEUR-WS.org. http://ceur-ws.org/Vol-1688/paper-13.pdf

[4] Austin R. Benson, Ravi Kumar, and Andrew Tomkins. 2016. Modeling User Consumption Sequences. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016.* ACM, 519–529. https://doi.org/10.1145/2872427.2883024

[5] Rahul Bhagat, Srevatsan Muralidharan, Alex Lobzhanidze, and Shankar Vishwanath. 2018. Buy It Again: Modeling Repeat Purchase Recommendations. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018.* ACM, 62–70. https://doi.org/10.1145/3219819.3219891

[6] Jun Chen, Chaokun Wang, and Jianmin Wang. 2015. Will You "Reconsume" the Near Past? Fast Prediction on Short-Term Reconsumption Behaviors. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.* AAAI Press, 23–29. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9491

[7] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.* ACM, 765–774. https://doi.org/10.1145/3331184.3331254

[8] Yifan Chen and Maarten de Rijke. 2018. A Collective Variational Autoencoder for Top-N Recommendation with Side Information. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2018, Vancouver, BC, Canada, October 6, 2018.* ACM, 3–9. https://doi.org/10.1145/3270323.3270326

[9] 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, UMAP 2020, Genoa, Italy, July 12-18, 2020. ACM, 80–87. https://doi.org/10.1145/3340631.3394850

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.* http://arxiv.org/abs/1511.06939

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[12] 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 & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* ACM, 1491–1499. https://doi.org/10.1145/3292500.3330979

[13] 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1071–1080. https://doi.org/10.1145/3397271.3401066

[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 2019, Macao, China, August 10-16, 2019.* ijcai.org, 2808–2814. https://doi.org/10.24963/ijcai.2019/389

[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, CIKM 2017, Singapore, November 06 - 10, 2017.* ACM, 1419–1428. https://doi.org/10.1145/3132847.3132926

[16] 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 https://arxiv.org/abs/2109.14233

[17] Yue Liu, Helena Lee, Palakorn Achananuparp, Ee-Peng Lim, Tzu-Ling Cheng, and Shou-De Lin. 2019. Characterizing and Predicting Repeat Food Consumption Behavior for Just-in-Time Interventions. In *Proceedings of the 9th International Conference on Digital Public Health, PDH 2019, Marseille, France, November 20-23, 2019.* ACM, 11–20. https://doi.org/10.1145/3357729.3357736

[18] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2021. Variational Bayesian Representation Learning for Grocery Recommendation. *Inf. Retr. J.* 24, 4-5 (2021), 347–369. https://doi.org/10.1007/s10791-021-09397-1

[19] Thomas Neifer, Dennis Lawo, Gunnar Stevens, Alexander Boden, and Andreas Gadatsch. 2021. Recommender Systems in Food Retail: Modeling Repeat Purchase Decisions on Transaction Data of a Stationary Food Retailer. In *Proceedings of the 18th International Conference on e-Business, ICE-B 2021, Online Streaming, July 7-9, 2021.* SciTePress, 25–36. https://doi.org/10.5220/0010553600250036

[20] 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, Virtual Event, Canada, July 11-15, 2021.* ACM, 859–868. https://doi.org/10.1145/3404835.3462836

[21] Jérémie Rappaz, Julian J. McAuley, and Karl Aberer. 2021. Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021.* ACM, 390–399. https://doi.org/10.1145/3460231.3474267

[22] Markus Reiter-Haas, Emilia Parada-Cabaleiro, Markus Schedl, Elham Motamedi, Marko Tkalcic, and Elisabeth Lex. 2021. Predicting Music Relistening Behavior Using the ACT-R Framework. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021.* ACM, 702–707. https://doi.org/10.1145/3460231.3478846

[23] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* AAAI Press, 4806–4813. https://doi.org/10.1609/aaai.v33i01.33014806

[24] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48).* JMLR.org, 1670–1679. http://proceedings.mlr.press/v48/schnabel16.html

[25] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.* ACM, 1441–1450. https://doi.org/10.1145/3357384.3357895

[26] Leilei Sun, Yansong Bai, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. 2020. Dual Sequential Network for Temporal Sets Prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020.* ACM, 1439–1448. https://doi.org/10.1145/3397271.3401124

[27] Manos Tsagkias, Tracy Holloway King, Surya Kallumadi, Vanessa Murdock, and Maarten de Rijke. 2020. Challenges and Research Opportunities in eCommerce Search and Recommendations. *SIGIR Forum* 54, 1 (2020).

[28] 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, CIKM 2018, Torino, Italy, October 22-26, 2018.* ACM, 1133–1142. https://doi.org/10.1145/3269206.3271786

[29] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019.* ACM, 1977–1987. https://doi.org/10.1145/3308558.3313594

[30] 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, SIGIR 2016, Pisa, Italy, July 17-21, 2016.* ACM, 729–732. https://doi.org/10.1145/2911451.2914683

[31] Le Yu, Leilei Sun, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. 2020. Predicting Temporal Sets with Deep Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.* ACM, 1083–1091. https://doi.org/10.1145/3394486.3403152