# また、Buy It Againの推奨のためのパーソナライズされたカテゴリ頻度予測

Amit Pande
amit.pande@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

Kunal Ghosh
kunal.ghosh@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

Rankyung Park
rankyung.park@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

## ABSTRACT

Buy It Again (BIA)の推奨は、小売業者にとって、顧客が自らのリピート購入パターンに基づいて再度購入する可能性の高い商品を提案することで、ユーザー体験とサイトエンゲージメントの向上に役立てるために極めて重要である。既存のBIA研究のほとんどは、ゲストのパーソナライズされた行動をアイテムの粒度で分析している。このような細かい粒度は、中小企業や検索目的の小さなデータセットに適しているかもしれない。しかし、この方法は、数億人のゲストと数千万の商品を持つ大手小売業者にとっては実現不可能である。このようなデータセットでは、顧客の行動をアイテムカテゴリーレベルで捉える粗視化モデルの方が実用的である。さらに、顧客は同じカテゴリー内のアイテムのバリエーションを一般的に探索する。例えば、ヨーグルトの異なるブランドやフレーバーを試す。このようなシナリオでは、カテゴリーベースのモデルがより適切かもしれない。我々は、パーソナライズされたカテゴリモデル(PCモデル)とカテゴリ内のパーソナライズされたアイテムモデル(ICモデル)から構成される階層的PCICモデルと呼ばれる推薦システムを提案する。PCモデルは、顧客が再び購入する可能性の高いカテゴリのパーソナライズされたリストを生成する。ICモデルは、ゲストがカテゴリー内で再集計する可能性が高いカテゴリー内のアイテムをランク付けする。階層的PCICモデルは、生存モデルを用いて製品の一般的な消費率を捉える。消費の傾向は、時系列モデルを用いて把握される。これらのモデルから得られた特徴量は、カテゴリー粒度のニューラルネットワークの学習に使用される。4つの標準的なオープンデータセットで、PCICを12の既存のベースラインと比較する。PCICはNDCGを最大16%改善し、リコールを約2%改善する。我々は、1億のゲストと3Mアイテムからなる大規模なデータセットで、PCICのスケールアップと学習(8時間以上)を行うことができた。PCICが導入され、A/Bが大手小売店の現場でテストされたことで、ゲストのエンゲージメントが大幅に向上した。

## KEYWORDS

パーソナライゼーション、レコメンダーシステム、Eコマース、リピート購入、再度購入、生存モデル、時系列モデル、ニューラルネットワーク

ACMリファレンス形式:
アミット・パンデ、クナル・ゴッシュ、ランキョン・パーク。2023. Buy It Againの推奨に対するパーソナライズされたカテゴリー頻度予測。第17回ACM推薦システム会議(RecSys '23)、9月18日～22日。2023年、シンガポール、シンガポール。ACM, ニューヨーク, 米国, 7ページ。https://doi.org/10.1145/3604915.3608822



*RecSys '23, September 18–22, 2023, Singapore, Singapore*

ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3608822

## 1 INTRODUCTION

電子商取引の出現により、レコメンデーションシステムは研究のホットトピックとなっている。Covid-19の登場により、ほとんどの買い物客がデジタルフルフィルメント、オーダーピックアップ、ドライブアップ、パーソナルショッパーをサポートするデジタルオーダーに切り替え、デジタル食料品の売上が急増した[6]。このような買い物客の行動の変化に伴い、顧客が次に購入したい、あるいは消費したい商品を提案する次バスケット推薦(NBR)[10, 13, 15-18]と、顧客の買い物体験を支援するパーソナライズされた仮想通路の構築の両方に多くの関心が向けられた。

顧客が過去に購入または消費したバスケットのシーケンスが与えられたとき、NBRシステムの目標は、顧客が次に購入または消費したいアイテムの次のバスケットを生成することである。NBRはさらに、似ているが異なる2つの問題に分けることができる。第一は、Buy It Again (BIA)問題と呼ばれるリピート購入推奨で、顧客がすでに購入した商品を推奨し、顧客がその商品から外れる可能性があるときにそれを行うことを目標とする。2つ目は、隣接するインスピレーションの推奨、つまり「あなたも好きなかもしれない」問題で、顧客が以前に購入したものや類似の顧客が購入したものを補完するような商品を購入するよう促すことを目的としている。

BIA勧告における既存の研究は、リカレントニューラルネットワーク[8-11, 14, 16-18]や統計モデル[1, 3-5, 7]の変種を用いた商品再購入確率のモデル化に焦点を当てている。大手小売業者は数億の商品とゲストを扱っているが、再購入取引の大半は、商品とゲストのごく一部である。これは、データが非常に高次元空間でまばらに表現されることになるため、項目粒度モデルのアンダーフィッティングにつながる可能性がある。最悪の場合、計算資源の制限により、学習自体が実行不可能になる可能性がある。

本研究では、BIA予測におけるパーソナライズドカテゴリ頻度モデリングの有効性を強調する。顧客は、異なるブランドを試したいという願望、顧客の家族内で様々な嗜好を満たす必要性、代替品の割引の有無などの理由で、カテゴリー内の品目や新しい品目のバリエーションを探索することが多い。カテゴリーベースの再購買モデリングは、これらの商品再購買ダイナミクスに関するより抽象度の高い情報を効果的に捉えることができる。図1に示すように、再購入件数の多い品目の割合は小さいが(図1a)、ほとんどのカテゴリーで再購入件数が多い(図1b)。この不一致は、カテゴリー再購入を目的としたモデルが、ゲストの選好を満たすのに効果的である可能性を意味している。さらに、前述のスパース性により

# Personalized Category Frequency prediction for Buy It Again recommendations

Amit Pande
amit.pande@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

Kunal Ghosh
kunal.ghosh@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

Rankyung Park
rankyung.park@target.com
Data Sciences, Target Corporation
Brooklyn Park, Minnesota, USA

## ABSTRACT

Buy It Again (BIA) recommendations are crucial to retailers to help improve user experience and site engagement by suggesting items that customers are likely to buy again based on their own repeat purchasing patterns. Most existing BIA studies analyze guests' personalized behaviour at item granularity. This finer level of granularity might be appropriate for small businesses or small datasets for search purposes. However, this approach can be infeasible for big retailers which have hundreds of millions of guests and tens of millions of items. For such data sets, it is more practical to have a coarse-grained model that captures customer behaviour at the item category level. In addition, customers commonly explore variants of items within the same categories, e.g., trying different brands or flavors of yogurt. A category-based model may be more appropriate in such scenarios. We propose a recommendation system called a *hierarchical PCIC model* that consists of a *personalized category model* (PC model) and a *personalized item model within categories* (IC model). PC model generates a personalized list of categories that customers are likely to purchase again. IC model ranks items within categories that guests are likely to reconsume within a category. The hierarchical PCIC model captures the general consumption rate of products using survival models. Trends in consumption are captured using time series models. Features derived from these models are used in training a category-grained neural network. We compare PCIC to twelve existing baselines on four standard open datasets. PCIC improves NDCG up to 16% while improving recall by around 2%. We were able to scale and train (over 8 hours) PCIC on a large dataset of 100M guests and 3M items where repeat categories of a guest outnumber repeat items. PCIC was deployed and A/B tested on the site of a major retailer, leading to significant gains in guest engagement.

## KEYWORDS

Personalization, Recommender Systems, E-commerce, Repeat purchases, Buy it again, Survival Models, Time-Series Models, Neural Network

**ACM Reference Format:**
Amit Pande, Kunal Ghosh, and Rankyung Park. 2023. Personalized Category Frequency prediction for Buy It Again recommendations. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3604915.3608822



*RecSys '23, September 18–22, 2023, Singapore, Singapore*

ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3608822

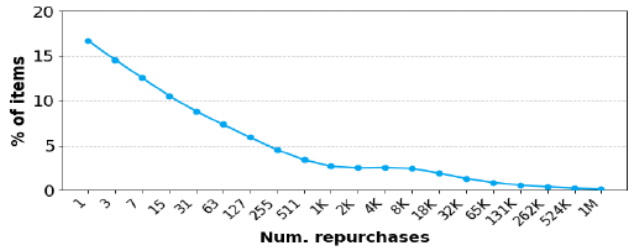
## 1 INTRODUCTION

With the advent of e-commerce, recommendation systems have become a hot topic for research. Digital grocery sales skyrocketed with the advent of Covid-19 as most shoppers switched to digital orders backed by digital fulfillment, order-pickup, drive-up, or personal shopper [6]. With this change in shoppers' behavior, a lot of attention went to both *next basket recommendation* (NBR) [10, 13, 15–18] that suggests items customers would like to purchase or consume next and to building personalized virtual aisles to aid the customer shopping experience.

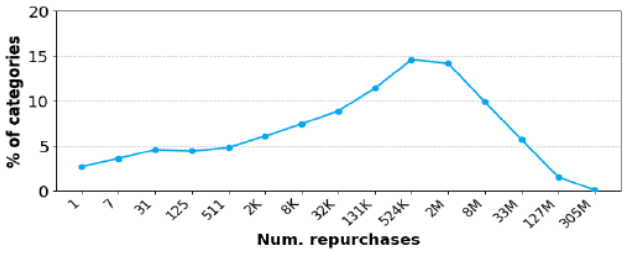
Given a sequence of baskets that a customer has purchased or consumed in the past, the goal of a NBR system is to generate the next basket of items that the customer would like to purchase or consume next. The NBR can be further divided into two similar but different problems. The first is repeat purchase recommendation, called the *Buy It Again* (BIA) problem, where the goal is to recommend items that customers have already purchased and do so at times when the customers might be running out of the item(s). The second is adjacent inspiration recommendation, or the *You might also like* problem, where the goal is to inspire customers to shop for items that may complement ones they have bought before or ones similar customers have purchased.

Existing work in BIA recommendations has focused on modeling item repurchase probabilities by using variants of recurrent neural networks [8–11, 14, 16–18] or statistical models [1, 3–5, 7]. Large retailers handle hundreds of millions of items and guests, but the majority of repurchase transactions are on a small subset of items and guests. This can lead to underfitting for item-grained models, as the data ends up being represented sparsely in a very high dimensional space. In the worst case, training itself may become infeasible due to computational resource limitations.

In this work, we emphasize the effectiveness of personalized category frequency modeling on BIA predictions. Customers will often explore variants of an item or new items within a category for reasons such as the desire to try different brands, the need to satisfy varying taste preferences in the customer's family, or the presence of discounts on alternative items. Category-based repurchase modeling can effectively capture higher abstraction information on these item repurchase dynamics. As shown in Figure 1, the percentage of items that have high numbers of repurchases is small (Figure 1a), but most categories demonstrate high levels of repurchases (Figure 1b). The discrepancy means that models geared toward category repurchases may be more effective at satisfying guest preferences. Furthermore, due to the aforementioned sparsity,




(a) 同じ商品をリパージしている顧客の割合




(b) 同じカテゴリーから再購入した顧客の割合

図1:1.5年間の再購入回数に対するアイテムとカテゴリの割合。(a) ほとんどの商品は再購入取引数が少ない。(b) ほとんどのカテゴリーで再購入取引が多い。カテゴリは項目よりもモデリングに十分なデータ量を持っている。

は、カテゴリー再購入よりもアイテム再購入の方が、性能の良いBIA推薦モデルを訓練することがはるかに困難である。

本研究の主な貢献は以下の通りである:

(1) 本論文では、BIA推薦のための2階層PCICモデルを提案する。パーソナライズド・カテゴリー・モデル(PCモデル)は、顧客が次回の訪問時に再びどのカテゴリーを購入するかを予測し、パーソナライズド・アイテム・イン・カテゴリー・モデル(ICモデル)は、カテゴリー内のアイテムのパーソナライズされたランクを提供する。個々の顧客に対する最終的なBIA推奨は、両方の予測を組み合わせることによって生成される。このモデルが、顧客はカテゴリー内の与えられたアイテムに似たブランド、サイズ、フレーバーなどを探索する傾向があるという我々の洞察をどのようにサポートするかを示す。

(2) 提案するPCICモデルが、既存の公開データセットのベースラインを凌駕することを実証する。また、PCICが大規模なデータセットに拡張可能であることを示す。

(3) PCICを商用環境に導入し、何百万人もの顧客にBIAの推奨を提供する。複数のA/Bテストによって証明された、現場でのゲスト体験の向上を実証する。PCICの導入と拡張の経験について述べる。

## 2 MODEL

### 2.1 カテゴリーレベルの再購入モデリング

カテゴリーレベルの特徴を用いて、顧客の再購入の可能性を予測する。各顧客は、購入履歴によって作成された独自の特徴を持っており、顧客の購入データの最後のm日間は、カテゴリレベルモデルを学習するためのラベルを生成するために使用される。このm日前の購入履歴をすべて特徴量を生成するために使用する。この期間に顧客が商品を買戻したカテゴリーはラベル1とみなされ、他のカテゴリーはラベル0と割り当てられる。モデルを学習するために考慮された主な特徴は、以降のサブセクションに列挙される。

2.1.1 生存分析。生存分析[12]は、関心のある事象が発生するまでの予想される期間に焦点を当てる。学習データの一部は部分的にしか観測できないという事実が従来の回帰とは異なり、これは打ち切りと表現される。これらの打ち切りオブザベーションでは,イベント時間が打ち切りのポイントでの時間よりも大きいことだけがわかる.小売シナリオでは、カテゴリー内の商品の購入をイベントとみなす。各カテゴリーについて、リピート購入データを使用して、各カテゴリーの顧客全体のライフテーブルを構築することができ、これにより、時間の関数としてリピート購入リスクを予測することができる。ライフテーブルは、時間経過に伴うイベントと打ち切りケースをまとめたものである。時刻0では、すべてのオブザベーション(参照購入)がまだリスクがあり、これは購入(イベント)をまだ繰り返していないか、打ち切られていないことを意味する。イベントや打ち切り事例が発生すると、オブザベーションはリスクセットから外れる。
リピート購入データは、いくつかの有用な機能を計算するために使用することができる:

1. ハザード(式1)は、k日目までにイベントが発生しなかったことを条件として、k日目にイベントが発生する確率である。あるイベント(再購入)が、あるユーザーがその時間までイベントフリーであり続ける(購入しない)という条件下で、ある時間間隔で発生するおおよその確率を示す。

$$\text{hazard}_k = \text{n\_event}_k / \text{n\_risk}_k \tag{1}$$

2. cum_hazard (eq. 2) は、ハザードの時間的な累積和である。

$$\text{cum\_hazard}_k = \sum_{kk=0}^{k} \text{hazard}_{kk} \tag{2}$$

3. 生存率(式3)は、k日目以降にイベントが発生する確率、または同等に、時刻tまでにまだイベントを経験していない割合である。

$$\text{survival}_k = \exp(-1 * \text{cum\_hazard}_k) \tag{3}$$

4. cum_survival (eq. 4)は、±3日以内にイベントが発生する確率として、今日まで。さらに、多くの食料品顧客が週に一度買い物をするため、この特徴を定義する。

$$\text{cum\_survival}_k = \text{生存}_{k+3} - \text{生存}_{k-3}. \tag{4}$$

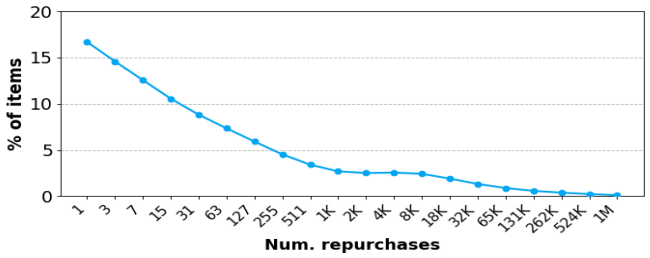
5. normalized_risk (eq. 5) は、今日のユーザーカテゴリーに関連するリスクを、購入当日のリスクの割合として定義する。

$$\text{norm\_risk}_k = \text{n\_risk}_k / \text{n\_risk}_0 \tag{5}$$

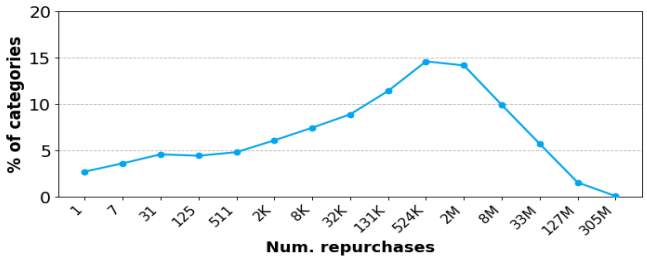
6. normalized_event (eq. 6) は、イベント+打ち切り母集団で正規化された、与えられた日のイベント確率として定義される。

$$\text{norm\_event}_k = \text{n\_event}_k / \text{n\_event\_\&\_censor}_k \tag{6}$$

このモデルを構築することで、商品の再購入率の母集団レベルの概要がわかる。




(a) Percentage of customers repurchasing the same item




(b) Percentage of customers repurchasing from the same category

**Figure 1: Percentage of items and categories against number of repurchases in 1.5 years. (a) Most items have small number of repurchasing transactions. (b) Most categories have large number of repurchasing transactions. Categories have more sufficient amount of data for modeling than items.**

it is far more difficult to train performant BIA recommendation models on item repurchases than it is on category repurchases.

The main contributions of this work as summarized below:

(1) In this paper, we propose a 2-tier *PCIC model* for BIA recommendations. The *personalized category model* (PC model) predicts which categories customers will buy again on their next visit, and the *personalized item within categories model* (IC model) provides personalized ranks of items in categories. Final BIA recommendations for individual customers are generated by combining both predictions. We show how the model supports our insights that customers tend to explore brands, sizes, flavors, etc. similar to a given item within a cateogy.
(2) We demonstrate that the proposed PCIC model outperforms existing baselines of public datasets. We also show that PCIC scales to large datasets.
(3) We deploy PCIC in a commercial setting to provide BIA recommendations for millions of customers. We demonstrate improved guest experience on the site as evidenced by multiple A/B tests. We discuss our experiences deploying and scaling PCIC.

## 2 MODEL

### 2.1 Category level repurchase modeling

We use category level features to predict the customers' likelihood to repurchase items. Each customer has their own features crafted by their purchase history, and the last $m$ days of customer purchase data is used to generate labels to train a category level model. All purchase history before this $m$ days is used to generate the features. Any category in which customers repurchased an item in this time period is considered label 1 while the other categories are assigned label 0. The main features considered to train the model are enumerated in subsequent subsections.

*2.1.1 Survival Analysis.* Survival analysis [12] focuses on the expected duration of time until occurrence of an event of interest. It differs from traditional regression by the fact that parts of the training data can only be partially observed, which is stated as being censored. For these censored observations, we only know that the event time is greater than the time at the point of censoring. In the retail scenario, we consider the purchase of an item within a category as an event. For each category, repeat purchase data can then be used to construct a life table across customers for each category, which will allow us to predict repeat purchase risk as a function of time. A life table summarizes the events and censored cases across time. At time 0, all observations (reference purchases) are still at risk, which meants that they have not yet repeated the purchase (event) or been censored. As events and censored cases occur, observations fall out of the risk set.

Repeat purchase data can be used to compute a few useful features:

1. `hazard` (eq. 1) is the probability of event occurring at kth day, conditional on the event not occurring before day k. It denotes an approximate probability that an event (repurchase) occurs in a given time interval, under the condition that an user would remain event-free up to that time (no purchase).

$$\texttt{hazard}_k = \texttt{n\_event}_k / \texttt{n\_risk}_k \tag{1}$$

2. `cum_hazard` (eq. 2) is cumulative sum of hazard over time.

$$\texttt{cum\_hazard}_k = \sum_{kk=0}^{k} \texttt{hazard}_{kk} \tag{2}$$

3. `survival` (eq. 3) is probability of the event occurring after day k or equivalently, the proportion that have not yet experienced the event by time t.

$$\texttt{survival}_k = \exp(-1 * \texttt{cum\_hazard}_k) \tag{3}$$

4. `cum_survival` (eq. 4) as probability of event occuring in ±3 days to today. We additionally define this feature since many grocery customers shop once a week.

$$\texttt{cum\_survival}_k = \texttt{survival}_{k+3} - \texttt{survival}_{k-3} \tag{4}$$

5. `normalized_risk` (eq. 5) is defined as risk associated with the user category today as a fraction of risk on the day of purchase.

$$\texttt{norm\_risk}_k = \texttt{n\_risk}_k / \texttt{n\_risk}_0 \tag{5}$$

6. `normalized_event` (eq. 6) is defined as the event probability on the given day normalized by event plus censor population.

$$\texttt{norm\_event}_k = \texttt{n\_event}_k / \texttt{n\_event\_\&\_censor}_k \tag{6}$$

Building this model gives a population level overview of the item repurchase rate.

| | Num Items | Num Users | Basket Size | Baskets/ User | Items/ user |
|---|---|---|---|---|---|
| tafeng | 12062 | 13949 | 6.27 | 5.69 | 6.397 |
| dunhumby | 4997 | 36241 | 7.33 | 7.99 | 22.56 |
| shoppers | 7907 | 10000 | 8.71 | 56.85 | 24.934 |
| instacart | 8000 | 19935 | 8.97 | 7.97 | 33.271 |
| Internal | ~3M | ~100M | ~10 | ~25 | ~200 |

表1:評価のために考慮したデータセットのいくつかの特徴

2.1.2 ARIMA モデル。自己回帰積分移動平均(ARIMA)モデルは、非定常時系列問題の短期予測に有用である。各顧客とカテゴリーについて、ARIMAを用いて購買パターンを特徴付け、翌日の購買を予測することを試みる。ARIMAモデルは3つのパラメータ(p, d, q)を持ち、pは自己回帰モデルの次数、dは差分化の度合い、qは移動平均モデルの次数である。あるカテゴリー内で過去に購入した日付を観測して次のカテゴリーを予測するARIMAモデルと、購入した品目の数量を検討し、顧客の現在の消費率を予測するモデルを構築する(例えばXは毎日2オンスシャンプーを使用する)。そして、これを用いて、顧客が品目から外れる可能性が高い日付を予測する。各顧客とカテゴリのペアについて、これらのモデルを訓練し、それらの予測ARIMA(date)とARIMA(rate)を特徴として使用する。

2.1.3 その他の特徴。我々は、さらに3つの行動カテゴリーレベルの特徴を考慮する: NumPurchases ある顧客がそのカテゴリーから購入した回数、tripsSinceLastPurchased そのカテゴリーで購入してから顧客が行った他のカテゴリーでの購入回数、daysSinceLastPurchased そのカテゴリーで購入した今日と最後の日付の時間差。

2.1.4 モデルの学習。過去1.5年間のユーザー・ショッピング・データを用いてモデルを学習し、年ごとのケイデンスを確実に把握する。最後のm日間のデータはラベルを生成するために保留される。例えば、2021年1月24日2022年7月24日データセットを用いて、すべてのゲストの特徴量を生成することができる。7月25日31日に買い物をしたゲスト(m=7)については、買い物をしなかったカテゴリーと買い物をしたカテゴリーについて、それぞれラベル0と1を生成する。生存モデルからの6つの特徴、2つのARIMAモデルからの2つの予測、および前述の他の3つの特徴が、各ユーザーとカテゴリのペアについて生成される。
カテゴリレベルのゲスト購入データセットに対して、2層のニューラルネットワークを学習させた。入力特徴量の数が少ないので、軽いので、使いたい(l1)、ユーザー数が多いので、うまくスケールするようにしたい。最も性能の良いニューラルネットは、シグモイド活性を持つ2つの完全連結層(10と5ニューロン)で構成されていた。出力層はソフトマックスにかけられ、ロジスティック損失関数が最適化に使われる。

## 2.2 カテゴリ間積ランキング

一般に、顧客は最も頻繁に、あるいは最も最近購入した商品を購入する可能性が高いことが観察された。カテゴリー内の商品をランク付けするために使用される主な特徴は、購入頻度(Freq)と購入の新しさ(Rec)の2つである。我々は、最適なランクに到達するために、両者を組み合わせたいと考えたが、再帰性は日数で測定され、頻度はカウントである。共通の土台に到達するために、両者をランクに変換する。

項目頻度ランク(IFR)と項目再帰性ランク(IRR)は、頻度カウントと、ある項目の最終購入からの日数(それぞれDaysSincePurchase)をランク付けすることで得られる。IFR = Rk (Freq)、IRR = Rk (購入からの日数)。ランクを加重平均で結合し、再度ランクを付け、そのランクを購入回数で割る(N IB)。この洞察はユーザーからのフィードバックに基づいており、後のセクションで説明する。式7は、最終的な項目ランク(IR)の計算方法を示している。

$$\mathrm{IR} = ceil(\frac{1}{NIB} \times Rk(\alpha \times IRR + \beta \times IFR)) \tag{7}$$

ここで、パラメータ$\alpha$と$\beta$は[0,1]の範囲で網羅的グリッド探索を用いて求めた。

## 2.3 Model output

PCモデルとICモデルの出力を組み合わせて、推薦のためのアイテムの集約された単一リストを得る。$Rk_{PC}$と$Rk_{IC}$はそれぞれアイテムのカテゴリのPCランクとアイテムのICランクを表すとする。PCICモデルはラウンドロビン方式で出力する、すなわちRk = Rk (sortByAscending($Rk_{PC}$, $Rk_{IC}$ ))

# 3 EXPERIMENTS

本節では、以下の問いに答えるための実験を行う。Q1: 提案手法の有効性はどの程度か?最先端のNBR/BIA手法を凌駕するか?Q2:この方法は、何百万人ものユーザーにレコメンデーションを生成するために、どの程度スケールアップできますか?Q3:モデルの性能は入力特徴量によってどのような影響を受けるか?Q4:トレーニング日とテスト日の範囲は、モデルの性能をどのように変化させるか?

## 3.1 実験設定

3.1.1 データセットと指標。表1に示す4つの公開データセットを用いて、提案手法と文献にある既存手法の性能を比較する: ValuedShopper[1], Instacart[2], Dunnhumby[3], TaFeng[4].また、大規模小売店におけるユーザーの販売履歴からなる内部データセットを用いて評価する。このデータセットには約100Mのユーザーと3Mの商品がある。我々は、想起(@K)とNDCG(@K)のメトリクスを用いて、我々の手法を評価し、比較する。

3.1.2 ベースライン TopSell: ユーザから購入された最も頻度の高い商品を全ユーザへの推薦文として使用する。FBought: ユーザーによって購入された最も頻度の高い商品を推薦文として使用する。userKNN [13]: 購入バスケットにkNNに基づく古典的な協調フィルタリングを使用する。リピートネット[15]

[1]https://www.kaggle.com/c/acquire-valued-shoppers-challenge/overview
[2]https://www.kaggle.com/c/instacart-market-basket-analysis
[3]https://www.dunnhumby.com/careers/engineering/sourcefiles
[4]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset

| | Num Items | Num Users | Basket Size | Baskets/ User | Items/ user |
|---|---|---|---|---|---|
| tafeng | 12062 | 13949 | 6.27 | 5.69 | 6.397 |
| dunhumby | 4997 | 36241 | 7.33 | 7.99 | 22.56 |
| shoppers | 7907 | 10000 | 8.71 | 56.85 | 24.934 |
| instacart | 8000 | 19935 | 8.97 | 7.97 | 33.271 |
| Internal | ~3M | ~100M | ~10 | ~25 | ~200 |

**Table 1: Some characteristics of datasets considered for evaluation**

*2.1.2 ARIMA models.* Autoregressive Integrated Moving Average or ARIMA models are useful for short term forecasts on non-stationary time series problem. For each customer and category, we try to characterize their purchase pattern using ARIMA and predict the next day of purchase. ARIMA models have three parameters ($p$, $d$, $q$) where $p$ is the order of the autoregressive model, $d$ is the degree of differencing, and $q$ is the order of the moving-average model. We build one ARIMA model that observes the past dates of purchases within a category to predict the next one and a second model to consider the quantity of item purchased and predict the current rate of consumption by the customer (say X uses 2 oz of shampoo daily). This is then used to predict the date when the customer will likely run out of the item. For each customer-category pair, we train these models and use their forecasts `ARIMA(date)` and `ARIMA(rate)` as features.

*2.1.3 Other features.* We consider three more behavioral category level features: `NumPurchases` - Number of times a given customer has purchased from the category, `tripsSinceLastPurchased` - the number of purchases in other categories customer has made since purchasing in this category, `daysSincelastPurchased` - the time difference between today and last date the customer made a purchase in this category.

*2.1.4 Model training.* We take the past 1.5 years of user shopping data to train the model to ensure we capture a yearly cadence. The last $m$ days of data is held out to generate labels. For example - we may take Jan 2021- July 24 2022 dataset to generate features for all guests. For those guests who shopped during July 25 - 31 (m = 7), we generate labels 0 and 1 for categories not shopped and shopped respectively. The 6 features from survival model, 2 predictions from two ARIMA models and the 3 other features mentioned earlier are generated for each user and category pair.

We trained a 2 layer neural network on the category level guest purchase dataset. We wanted to keep it light because the number of input features is small (11), and we wanted it to scale well for the large number of users. The most performant neural net was composed of 2 fully connected layers (10 and 5 neurons) with sigmoid activations. The output layer is run through a softmax and the logistic loss function is used for optimization.

## 2.2 Inter-category Product Ranking

In general, we observed that a customer is most likely to repurchase their most frequently or most recently bought items. The two main features used to rank products within a category are frequency (Freq) and recency (Rec) of purchase. We wanted to combine them both to arrive at optimal ranks, however, recency is measured in days and frequency is a count. To come to a common ground, we convert both into ranks. Item Frequency Rank (IFR) and Item Recency Rank (IRR) are obtained by ranking the frequency counts and days (respectively) since the last purchase of an item (DaysSincePurchase). $\texttt{IFR} = Rk(Freq)$, $\texttt{IRR} = Rk(DaysSincePurchase)$. We combine the ranks using a weighted average, rank again, then divide the rank by number of times the item is bought ($NIB$). This insight was based on user feedback and will be discussed in later sections. The equation 7 shows how final Item Rank (IR) is calculated.

$$\texttt{IR} = ceil(\frac{1}{NIB} \times Rk(\alpha \times IRR + \beta \times IFR)) \tag{7}$$

where the parameters $\alpha$ and $\beta$ were obtained using exhaustive grid search in the range [0,1].

## 2.3 Model output

We combine the outputs of PC and IC models to get an aggregated single list of items for recommendations. Let $Rk_{PC}$ and $Rk_{IC}$ represent the PC rank for an item's category and IC rank of the item respectively. The PCIC model outputs in a round robin manner i.e. $Rk = Rk(sortByAscending(Rk_{PC}, Rk_{IC}))$

# 3 EXPERIMENTS

In this section, we conduct experiments to answer the following questions: Q1: What is the effectiveness of the proposed method? Does it outperform state-of-the-art NBR/ BIA methods? Q2: How well does this method scale up to generate recommendations for millions of users? Q3: How is model performance impacted by the input features? Q4: How do training and testing date ranges change the performance of the model?

## 3.1 Experimental Settings

*3.1.1 Datasets & Metrics.* We use four publicly available datasets shown in Table 1 to compare the performance of the proposed method with existing methods in literature: ValuedShopper[1], Instacart[2], Dunnhumby[3], and TaFeng[4]. We also evaluate using an internal dataset consisting of the sales history of users at a large retailer. There are around 100M users and 3M products in this dataset. We use recall (@K) and NDCG (@K) metrics to evaluate and compare our methods.

*3.1.2 Baselines.* **TopSell**: It uses the most frequent items that are purchased by users as the recommendations to all users. **FBought**: It uses the most frequent items that are purchased by a user as the recommendationto him. **userKNN** [13]: It uses classical collaborative filtering based on kNN on purchase baskets. **RepeatNet** [15]:

[1]https://www.kaggle.com/c/acquire-valued-shoppers-challenge/overview
[2]https://www.kaggle.com/c/instacart-market-basket-analysis
[3]https://www.dunnhumby.com/careers/engineering/sourcefiles
[4]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset

| | Recall @10 | | | | NDCG @10 | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | V Shopper | Instacart | Dunhumby | TaFeng | V Shopper | Instacart | Dunhumby | TaFeng |
| TopSell | 0.0982 | 0.0724 | 0.0819 | 0.0773 | 0.0779 | 0.0641 | 0.0601 | 0.0519 |
| FBought | *0.2109* | *0.3426* | *0.1853* | 0.0704 | *0.2128* | *0.3618* | 0.1771 | 0.0766 |
| userKNN | 0.0988 | 0.0720 | 0.1135 | 0.1089 | 0.1415 | 0.1020 | 0.1707 | 0.0832 |
| RepeatNet | 0.1031 | 0.2107 | 0.1324 | 0.0645 | 0.1439 | 0.2285 | 0.1545 | 0.0592 |
| FPMC | 0.0951 | 0.0763 | 0.0919 | 0.0868 | 0.1188 | 0.0946 | 0.1025 | 0.0667 |
| DREAM | 0.0991 | 0.0866 | 0.0915 | 0.0902 | 0.1231 | 0.1063 | 0.1009 | 0.0763 |
| SHAN | 0.0847 | 0.0902 | 0.1007 | 0.0878 | 0.1032 | 0.1152 | 0.1149 | 0.0813 |
| Sets2Sets | 0.1259 | *0.3021* | 0.2068 | 0.1190 | *0.1626* | 0.3487 | *0.2134* | 0.0844 |
| TIFUKNN (NBR) | **0.3578** | **0.3952** | **0.2087** | 0.1301 | 0.3060 | *0.3825* | *0.1983* | 0.1011 |
| TIFUKNN(BIA) | *0.3500* | *0.3700* | *0.1940* | 0.0990 | *0.3000* | *0.3800* | *0.1860* | 0.0860 |
| RCP | 0.0416 | 0.1090 | 0.0635 | **0.3860** | 0.0591 | 0.1175 | 0.0634 | **0.2363** |
| ATD | 0.0350 | 0.1600 | 0.0468 | *0.3100* | 0.0605 | 0.1264 | 0.0350 | *0.2310* |
| PG | 0.1694 | 0.2375 | *0.1332* | *0.3100* | 0.0684 | 0.1331 | 0.0351 | *0.2336* |
| MPG | 0.1762 | 0.2183 | 0.0820 | *0.3200* | 0.0680 | 0.1240 | 0.0450 | *0.1600* |
| PCIC model | *0.3528* | 0.2548 | 0.1540 | 0.1427 | **0.3531** | **0.5700** | **0.2321** | 0.1180 |

表2:既存のベースラインとの性能比較。データセットで上位のalgoは太字で表示されている。3つのランナーアップはイタリック体で示されている。

セッションベースの推薦のためのRNNベースのモデルで、ユーザーの繰り返しの購買行動を捉える。FPMC [16]: 行列因数分解は、マルコフ連鎖が時間的にシーケンス効果を捉えることができるのに対して、ユーザーの一般的な嗜好を学習するためにすべてのデータを使用する。DREAM [18]: Dynamic Recurrent bAsket Model (DREAM)は、ユーザの動的な表現を学習するが、バスケット間の大域的な逐次特徴も捉える。SHAN [17]: 階層的注意ネットワークに基づく深層モデル。Sets2Sets [10]: Sets2Sets[10]:RNNに基づく複数のバスケット予測に従うための最先端のエンドツーエンド手法。RCP [2]: 繰り返し顧客確率(RCP)は、あるアイテムの繰り返し確率とそれに基づいて、繰り返しアイテムを見つける。ATD [2]: 時間分布の集計モデルは、繰り返し項目の確率分布と時間特性をモデル化するために時間分布に適合する。PG [2]: 購買行動を予測するために適合したポアソンガンマ分布。MPG [2]: 結果を時間依存にし、顧客確率を繰り返すために修正されたPG分布。

比較した手法では、グリッドサーチを用いてハイパーパラメータを調整する。userKNNでは、最近傍の数をrange(100, 1300)から探索する。FPMCでは、値の集合から因子の次元を探索する[16, 32, 64, 128]。RepeatNet、DREAM、SHAN、Sets2Setsについては、値の集合[16, 32, 64, 128]から埋め込みサイズを探索する。PCICモデルでは、ARIMAモデルは(3, 3, 0)の範囲で自動適合された。

## 3.2 性能比較(Q1)

表2は、PCICモデルと既存のベースラインとの性能比較である。表からいくつかの観察ができる。まず、PCICモデルは、Valued Shopper、instacard、Dunhumbyデータセットにおいて、ほとんどの場合、最も高い(またはそれに近い)再現率とNDCG値を持つことが観察される。意外なことに、RCPモデルはtafengデータセットで良好な性能を示した。我々のモデルがパーソナライズされたカテゴリー頻度を捉えるように、TIFUKNNモデルはパーソナライズされたアイテム頻度を明示的に捉えようとする。TIFUKNNモデルは、協調フィルタリングに最近傍アプローチを使用し、他のユーザーからの再購入パターンを学習する。BIA 3.3タスクScaling only TIFUKNN(BIA)で実行するためにコードを修正し、実行した。

(Q2)PCICモデルでは、生存分析機能はカテゴリーレベルでのユーザーの再購入パターンを使用する。Sets2Setsはパーソナライズされたアイテムの頻度を明示的に捉えるが、その後RNNの係数を学習する。RCP、ATD、PG、MPGモデルは、ポアソンガンマ分布または修正ポアソンガンマ分布を使用して、リピート購入パターンをモデル化しようとする。したがって、これらの方法は、RepeatNet、userKNN、FPMC、DREAMのような、アイテムやカテゴリの頻度を捕捉しない既存のどの方法よりも優れた性能を発揮することがわかる。FBoughtは、ユーザーの最も頻繁に購入されるアイテムをその順番でランク付けするだけで、かなり単純なベースラインである。この結果は、多くのベースラインよりも驚くほど良い結果を示している。ベースラインを実装するのは簡単で、かなり良い結果が得られる。

## 3.3 Scaling up (Q2)

我々は、より大規模な(100Mユーザー)データセットで、上記の上位の性能を持つモデルの訓練を試みた。TIFUKNNは商品カタログ全体のサイズを埋め込むユーザーを使用しているため、このデータセットへのスケールアップは不可能である。その結果、より大きなデータセットをサブサンプリングし、1Mのユーザーからなる代表的なサンプルを作成した。このサブサンプルデータを用いて、TIFUKNNとSets2SetsをPCICと比較した。TIFUKNNとSets2Setsでは、PCICに対してNDCGとリコールの指標が30～35%減少することが確認された。その結果、どちらのアルゴリズムもスケーリングに労力を費やさなかった。

PCICはApache Sparkを使用した分散ハドープクラスタで実装され、1億人のユーザーを対象にモデルの学習とテストに約6～8時間かかる。また、論文に記載されている数学を用いて、分散クラスタにMPGモデルを実装した。表3は、FBought、MPG、PCICモデルの性能比較である。PCICはNDCGの点では良好であるが、想起はMPGよりわずかに低い。次に、元の項目レベルではなく、カテゴリレベルでMPGパラメータを計算し、特徴の一部としてPCICに入力した。統合PCIC(+MPG)の性能はPCICとMPGの両方を上回る。

## 3.4 特徴の重要度(Q3)

特徴の重要度を得るために、元のニューラル層を勾配ブースティング木分類器に置き換えた。

| | Recall @10 | | | | NDCG @10 | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | V Shopper | Instacart | Dunhumby | TaFeng | V Shopper | Instacart | Dunhumby | TaFeng |
| TopSell | 0.0982 | 0.0724 | 0.0819 | 0.0773 | 0.0779 | 0.0641 | 0.0601 | 0.0519 |
| FBought | *0.2109* | *0.3426* | *0.1853* | 0.0704 | *0.2128* | *0.3618* | 0.1771 | 0.0766 |
| userKNN | 0.0988 | 0.0720 | 0.1135 | 0.1089 | 0.1415 | 0.1020 | 0.1707 | 0.0832 |
| RepeatNet | 0.1031 | 0.2107 | 0.1324 | 0.0645 | 0.1439 | 0.2285 | 0.1545 | 0.0592 |
| FPMC | 0.0951 | 0.0763 | 0.0919 | 0.0868 | 0.1188 | 0.0946 | 0.1025 | 0.0667 |
| DREAM | 0.0991 | 0.0866 | 0.0915 | 0.0902 | 0.1231 | 0.1063 | 0.1009 | 0.0763 |
| SHAN | 0.0847 | 0.0902 | 0.1007 | 0.0878 | 0.1032 | 0.1152 | 0.1149 | 0.0813 |
| Sets2Sets | 0.1259 | *0.3021* | 0.2068 | 0.1190 | *0.1626* | 0.3487 | *0.2134* | 0.0844 |
| TIFUKNN (NBR) | **0.3578** | **0.3952** | **0.2087** | 0.1301 | 0.3060 | *0.3825* | *0.1983* | 0.1011 |
| TIFUKNN(BIA) | *0.3500* | *0.3700* | *0.1940* | 0.0990 | *0.3000* | *0.3800* | *0.1860* | 0.0860 |
| RCP | 0.0416 | 0.1090 | 0.0635 | **0.3860** | 0.0591 | 0.1175 | 0.0634 | **0.2363** |
| ATD | 0.0350 | 0.1600 | 0.0468 | *0.3100* | 0.0605 | 0.1264 | 0.0350 | *0.2310* |
| PG | 0.1694 | 0.2375 | *0.1332* | *0.3100* | 0.0684 | 0.1331 | 0.0351 | *0.2336* |
| MPG | 0.1762 | 0.2183 | 0.0820 | *0.3200* | 0.0680 | 0.1240 | 0.0450 | *0.1600* |
| PCIC model | *0.3528* | 0.2548 | 0.1540 | 0.1427 | **0.3531** | **0.5700** | **0.2321** | 0.1180 |

**Table 2: Performance comparison with existing baselines. The top performing algo in a dataset are in bold. The three runner ups are in italics.**

RNN-based model for session-based recommendation which captures the repeated purchase behavior of users. **FPMC** [16]: Matrix Factorization uses all data to learn the general taste of the user whereas Markov Chains can capture sequence effects in time. **DREAM** [18]: Dynamic REcurrent bAsket Model (DREAM) learns a dynamic representation of a user but also captures global sequential features among baskets. **SHAN** [17]: A deep model based on hierarchical attention networks. It partitions the historical baskets into longterm and short-term parts.**Sets2Sets** [10]: The state-of-the-art end-to-end method for following multiple baskets prediction based on RNN. **RCP** [2]: Repeat Customer Probability (RCP) finds repeat probably of an item & repeat items based on that. **ATD** [2]: Aggregate Time Distribution Model fits a time distribution to model probablity distribution and time characteristics of repeat items. **PG** [2]: Poisson Gamma distribution fitted to predictaggregate purchasing behavior. **MPG** [2]: A modified PG distribution to make the results time dependent and intergate repeat customer probability.

We use grid search to tune the hyper-parameters in compared methods. For userKNN, the number of nearest neighbors is searched from range(100, 1300). For FPMC, the dimension of factor is searched from the set of values [16, 32, 64, 128]. For RepeatNet, DREAM, SHAN, and Sets2Sets, the embedding size is searched from the set of values [16, 32, 64, 128]. For PCIC model, ARIMA model was autofitted in range (3, 3, 0).

## 3.2 Performance Comparson (Q1)

Table 2 gives the performance comparison of PCIC model with existing baselines. Several observations can be made from the table. First, we observe that the PCIC model has highest (or near highest) recall and NDCG values in most cases on Valued Shopper, instacard and Dunhumby datasets. Surprisingly, RCP model performs well on tafeng dataset. Just like our model captures personalized category frequency, TIFUKNN model tries to explicitly capture personalized item frequency. TIFUKNN model uses nearest neighbor approach to collaborative filtering to learn repurchasing pattern from other users. We modified the code and ran it to run on the BIA task only TIFUKNN(BIA). In PCIC model, the survival analysis features use user repurchasing pattern at category level. Sets2Sets captures personalized item frequency explicitly but subsequently learns coeffients for RNN. RCP, ATD, PG and MPG models try to model repeat purchase pattern using a Poisson Gamma or modified Poisson Gamma distribution. Hence, we can see that these methods perform better than any existing methods which do not capture item or category frequency such as RepeatNet, userKNN,FPMC and DREAM. FBought is a pretty simple baseline in that it simply ranks the most frequently bought items of a user in that order. It surprisingly performs better than many baselines here. It is a simple to implement baseline and performs pretty well.

## 3.3 Scaling up (Q2)

We attempted to train the top performing models above on a much larger (100M user) data set. TIFUKNN uses a user embedding the size of the entire product catalog, which made it impossible to scale up to this data set. As a result, we subsampled the larger data set, creating a representative sample with 1M users. We compared TIFUKNN and Sets2Sets to PCIC using this subsampled data. We observed a 30-35% reduction in NDCG and recall metrics in TIFUKNN and Sets2Sets against PCIC. As a result, we did not put effort into scaling either algorithm.

PCIC was implemented in a distributed hadoop cluster using Apache Spark and takes around 6-8 hours of time to train and test the model for 100M users. We also implemented MPG model in distributed cluster using the maths described in the paper. Table 3 shows the performance comparison of FBought, MPG and PCIC models. Although PCIC performs well in terms of NDCG, the recall is slightly lower than MPG . Next, we calculated MPG parameters at category level instead of original item level and input it as part of features to PC. The performance of integrated PCIC(+MPG) outperforms both PCIC and MPG.

## 3.4 Feature Importance (Q3)

To obtain the feature importance, we replaced the original neural layer with a Gradient Boosting Tree clasifier. The values are plotted

| | Recall@3 | NDCG@3 | Recall@5 | NDCG@5 |
|---|---|---|---|---|
| FBought | 0.2020 | 0.0832 | 0.0305 | 0.1212 |
| MPG | **0.0307** | 0.1036 | **0.0433** | 0.1328 |
| PCIC | 0.0267 | **0.1071** | 0.0377 | **0.1368** |
| PCIC(+MPG) | 0.0317 | **0.1091** | 0.0447 | **0.1408** |

表3：内部データセットでの性能比較

その値を図2にプロットする。ARIMA予測は、モデルの出力、特にユーザーによる個々のアイテムの消費率に基づいて次の購入を予測しようとするモデルに非常に大きな影響を与えることが観察できる。生存特徴は予測品質への影響が小さく、他のユーザーの購入は、自分の特徴よりもユーザーの再購入において小さな役割を果たすことを意味する。これは、itemKNNやTIFUKNNのように、協調的なユーザー行動に焦点を当てたアプローチがPCICほどうまく機能しない理由の1つである可能性がある。MPGは統計モデルで消費率を捉えており、PCICに近い。過去の購入からの日数や明示的なカテゴリー頻度(num purchage)などの特徴も、特徴の重要度が高い。上位3つの特徴を収集した場合、ユーザーが過去に何回購入したか、前回一緒に購入してから何日経過したか、最後に何回購入したか、どのくらい続くか、に基づいて、今日その商品を購入するかどうかを予測できると言える。

特徴の重要度を得るために、元のニューラル層を勾配ブースティング木分類器に置き換えた。その値を図2にプロットする。ARIMA予測は、モデルの出力、特にユーザーによる個々のアイテムの消費率に基づいて次の購入を予測しようとするモデルに非常に大きな影響を与えることが観察できる。生存特徴は予測品質への影響が小さく、他のユーザーの購入は、自分の特徴よりもユーザーの再購入において小さな役割を果たすことを意味する。これは、itemKNNやTIFUKNNのように、協調的なユーザー行動に焦点を当てたアプローチがPCICほどうまく機能しない理由の1つである可能性がある。

### 3.5 訓練データとテストデータの選択による影響(Q4)

このデータセットから、直近の顧客の購入を1週間テスト用に除外し、その1週間以前に行われた購入を1年間トレーニング用として使用した。顧客とその製品購入は、顧客がトレーニング期間(テスト期間のy年前、y =1.5)に製品を購入し、テスト期間中に同じ製品を購入した場合にのみ、テスト期間中に再購入とみなされた。この期間に購入した(ユーザー、カテゴリー)ペアは1とラベル付けされ、この期間に購入しなかったカテゴリーは0とラベル付けされた。

パンデミックによりアプリやウェブサイトの普及が進む中、特にオンラインショッピングの頻度が高くなった。最初のフィードバックによると、BIAリストは特にエンゲージメントの高いユーザーにとって更新されていないことが確認された。これは、以下の理由によると考えられる:(1)すべてのユーザーに対して学習されたモデルは、高関与度ユーザーのシグナルや行動を正確に捉えることができない可能性がある。(2) ラベルは、購入した最後の1週間をもとに取得した。しかし、エンゲージメントの高いユーザーは買い物をする頻度が高いので、ラベルはあまり正確ではない。ユーザー購入の1日について、毎日モデルを採点する実験を行った。また、25以上のカテゴリで購入したユーザーと定義される、最もエンゲージメントの高いユーザーのみでモデルを学習させる実験も行った。

表4(a)は、テスト時間帯の変更と、最も熱心なユーザーのみを対象としたトレーニングによる、PCモデルのNDCGメトリックの改善を示している。テスト時間枠を小さくすることで、モデルの性能が大幅に向上した。テスト日が7日の場合、最も熱心なユーザーは、すべてのユーザーよりもNDCGのパフォーマンスが低かった。また、最も熱心なユーザーのみでモデルを訓練すると、訓練時間の節約にはつながるものの、すべてのユーザーのNDCGも改善されることが観察された。特徴量の生成とモデルの学習にかかる時間は、全ユーザに対して2.5倍であり、高関与ユーザに対して2.5倍である。

## 4 展開ジャーニー

このセクションでは、我々が取り組んだいくつかのユーザー向けの質問と、PCICの導入経験について述べる。

## 4.1 展開とオンライン経験

PCICを、Apache Sparkエコシステム上のコンピュートクラスタで毎日レコメンデーションを生成し、クラウドにエクスポートしてリアルタイムに配信するプロダクション環境に展開した。ユーザーがサイトを訪問すると、これらのレコメンデーションは、ユーザーが選択した在庫と利用可能な出荷オプションに基づいて、アイテムの可用性に基づいてフィルタリングされ、ユーザーに提供される。

### 4.2 ヒューマンインザループフィードバック

まず、この結果を社内のチームメンバーのプールに展開し、テストを行った。これにより、ユーザーが(友人や家族などと)アプリであまり見慣れないかもしれないいくつかのカテゴリの除外リストがあることについて、いくつかのフィードバックが得られた。フィードバックに基づき、レコメンデーションの上にフィルタとして適用されるカテゴリの除外リストを構築した。

ユーザーが最近購入した商品(例えば、新しいフレーバーのヨーグルト)を、再購入したカテゴリーから勧められることがあるが、再購入したいカテゴリーからは勧められないことがわかった。このような項目は、推奨事項からフィルタリングする2段階のアプローチで行った。カテゴリーが再購入カテゴリーであることとは別に、過去nヶ月間に少なくとも2回、ゲストに購入された商品であることを確認しようとした(n=6)。これは、顧客が購入した商品を再度購入した商品としてリストアップするのに役立つ。次に、再購入率が低い商品(RCP[2]の再購入率の閾値と同様)を特定し、削除した。

最初のテストでは、テストユーザーは通常、特定のカテゴリー(例えば、2種類以上のヨーグルト)から1回の旅行で複数の商品を購入すると指摘した。これを解決するために、旅行ごとにユーザーが購入した回数を示す変数N I Bを計算する。項目ランクをNIBで割り、ceil関数を用いて新しい項目ランクを作成することで、2つのリストを結合するために使用する数学を微調整した。

### 4.3 Metrics

既存のオンラインベースラインに対してA/Bテストを実施した。各検定は2週間以上実施し、検体が統計的に有意であることを確認した上で中止した。テストのために考慮されるメトリクスは以下のように定義される:

| | Recall@3 | NDCG@3 | Recall@5 | NDCG@5 |
|---|---|---|---|---|
| FBought | 0.2020 | 0.0832 | 0.0305 | 0.1212 |
| MPG | **0.0307** | 0.1036 | **0.0433** | 0.1328 |
| PCIC | 0.0267 | **0.1071** | 0.0377 | **0.1368** |
| PCIC(+MPG) | 0.0317 | **0.1091** | 0.0447 | **0.1408** |

**Table 3: Performance comparison on internal dataset**

in figure 2. We can observe that the ARIMA forecasts have a very high impact on the output of the model, particularly the model that tries to predict the next purchase based on rate of individual consumption of item by the user. The survival features have smaller impact on the prediction quality meaning other user's purchases play a small role in user's repurchase than his own characteristics. This can be one of the reason why approaches like itemKNN or TIFUKNN which focus on collaborative user behavior don't perform as well as PCIC. MPG does capture rate of consumption with a statistical model and it comes close to PCIC. The features such as number of days since past purchase and explicit category frequency (num purchase) also have high feature importance. if we were to collect the top 3 features, we can say that we can predict whether a user will purchase an item today based on how many times he has purchased before, how many days since his last purchase with us, how much did he purchase last time and how long will it last.

To obtain the feature importance, we replaced the original neural layer with a Gradient Boosting Tree clasifier. The values are plotted in figure 2. We can observe that the ARIMA forecasts have a very high impact on the output of the model, particularly the model that tries to predict the next purchase based on rate of individual consumption of item by the user. The survival features have smaller impact on the prediction quality meaning other user's purchases play a small role in user's repurchase than his own characteristics. This can be one of the reason why approaches like itemKNN or TIFUKNN which focus on collaborative user behavior don't perform as well as PCIC.

## 3.5 Impact of train and test data selection (Q4)

We held out one week of the most recent customer purchases from this dataset for testing and used one year of purchases made prior to that week for training. A customer and their product purchase were considered as a repeat purchase in the test period only if the customer purchased a product in the training period (y years before the test period, y =1.5) and also purchased the same product sometime in the test period. The (user, category) pairs purchased in this duration are labeled 1 and the categories the user did not purchase in this duration was labeled as 0.

As the pandemic caused increased adoption of the app and website, users started shopping online more frequently particularly. Based on the initial feedback, we observed that the BIA list was not updating particularly for the highly engaged users. We hypothesized that this can be because of the following reasons: (1) the model being trained on all users may not be able to exactly capture the signals and behavior of highly engaged user. (2) The labels are captured based on last 1 week of purchases. But highly engaged users shop much more often, hence their labels are not very accurate. We experimented with scoring the model daily on 1 day of user purchases. We also experimented on training the model only on the most engaged users, defined as users who have made purchases in more than 25 categories.

Table 4(a) shows the improvement in NDCG metric for the PC model with the changes in test time frame and with training on only the most engaged users. Reducing the test time frame significantly improved the performance of the model. The most engaged users had a lower NDCG performance than all users when the test dates were 7 days. We also observed that training the model only on the most engaged users improves NDCG for all users too although it leads in savings on training time. The time taken to train the generate the features and train the model on all users is 2.5x the time taken for highly engaged users

# 4 DEPLOYMENT JOURNEY

In this section, we discuss several user-facing questions we addressed as well as our experience in deploying PCIC.

## 4.1 Deployment and Online Experience

We deployed PCIC to a production environment where recommendations are generated daily in our compute cluster on an Apache Spark ecosystem and exported to the cloud for real-time serving. When a user visits the site, these recommendations are then served to them, filtered on the item availability based on inventory and available shipment options selected by the user.

## 4.2 Human-in-the-loop feedback

We first rolled out the results to a pool of internal team members for testing. This gave us some feedback as to having an exclusion list of some categories which users may not be very comfortable looking at, in their App (with friends and family or otherwise). Based on the feedback, we built an exclusion list of categories which are applied on top of recommendations as filters.

We found that users were sometimes recommended an item they'd recently purchased (e.g., a new flavor of yogurt) from a category where they repurchase, but not one they'd like to repurchase. We used a two step approach filter out such items from recommendations. Apart from the category being a repurchase category, we tried to ensure that the item was bought by the guest at least twice in the past n months (n=6). This helps the customer to identify the items in buy it again list as an item they have repeat purchased. Second, we identified items with low repurchase rates (similar to repurchase rate threshold in RCP [2]) and removed them.

In initial testing, test users noted they typically buy more than one item from a specfic category (e.g., two or more flavors of yogurt) in a single trip. To resolve this, we calculate a variable *NIB* which denotes the number of times the item was purchased by the user per trip. We tweaked the math used to combine the two lists by dividing item rank by NIB and then taking a ceil function to create new item ranks.

## 4.3 Metrics

We performed A/B tests against existing online baselines. Each test was run for more than two weeks and stopped after ensuring that the samples are statistically significant. The metrics considered for tests are defined as follows:

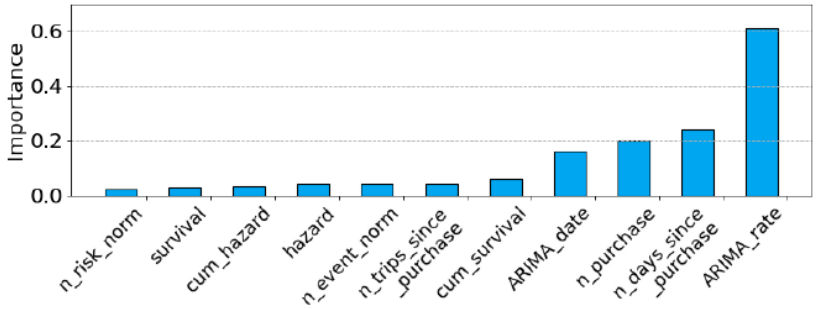



図2:PCモデルに対する入力特徴量の相対的重要度

| Trained on | Test Timeframe | NDCG (Test) | |
|---|---|---|---|
| | | Most Engaged | All |
| All | 7 days | 0.2009 | 0.2325 |
| All | 1 day | 0.3501 | 0.3583 |
| Most Engaged | 1 day | 0.3602 | 0.3589 |

| | Lift (%) |
|---|---|
| CTR | 6 |
| Conversion | 8.5 |
| Units | 27.5 |

表4:テスト結果（a）学習データ選択とテスト時間帯の変更に伴うPCモデルの性能の変化。(b) オンラインA/BテストにおけるFBoughtに対するBIAの影響の測定。

- CTR またはクリック率 ： ゲストがクリックしたレコメンデーションディスプレイの割合。
- コンバージョン率: クリックされたレコメンデーションのうち、ゲストが同日に購入した割合。
- ユニット: ユニット:治療に参加したユーザーが購入したユニットの総数。

## 4.4 A/B試験結果

ゲストのショッピング体験にBuy It Againの推薦リストを導入した際、FBoughtのベースラインに対してPCICをA/Bテストした。結果は表4(b)に示すとおりである。CTRでは6%、Conversionでは9%、購入台数では27%の3つの指標すべてにおいて、大きな上昇が見られることがわかる。

また、全ユーザーの検索結果に、Buy It Again推薦リストを追加するテストも行った。このため、検索クエリコンテキストを使用して、Buy It Againの結果をフィルタリングした。この推薦リストとのユーザーインタラクションは、既存の検索結果よりも有意に高い(20%以上)ことがわかった。カートの足し算、注文の平均値、注文ごとの単位が0～2%増加したことが観察された(ゲストが新しい商品を探すすべてのゲスト訪問を含む)。

## 4.5 仮想通路の構築

次に、カテゴリ(Milk、Yogurt、Beautyなど)ごとにレコメンデーションをフィルタリングすることで、ゲストにBIAを展開し、App/siteの専用スペースでオンラインユーザーのための仮想通路体験を作成した。PCモデルを用いて、各ゲストのパーソナライズされたカテゴリのリストを使用する。各カテゴリについて、ICモデルから仮想通路を形成するために推奨される項目のリストを示す。各通路で、まずゲストのBIA項目を示し、次に他の関連項目を示した。これらのレコメンデーションと対話したユーザーは、1オーダーあたりのユニット数(25～50%)、平均オーダー数(7～35%)が大幅に増加した。また、必要不可欠な購入はチケットの安いアイテムであるため、注文額に対するドルの影響は1注文あたりの単位よりも小さい。アプリでは、サイトよりもバーチャル通路へのゲストのエンゲージメントが高いことがわかった。

## 5 今後の方向性

Buy It ユーザーがショッピングミッションを迅速に完了できるように、再度推奨する。従来のアプローチは、ゲストのパーソナライズされた行動をアイテムの粒度でモデル化する傾向がある。本稿では、顧客の行動をアイテムカテゴリーレベルで捉えることができる粗視化モデルのケースを紹介する。提案するパーソナライズドカテゴリ(PC)モデルとカテゴリ内アイテム(IC)モデルの組み合わせは、標準的な公開データセットにおいて、既存のBIAモデルやNBRモデルを凌駕する。PCICモデルは、数百万サイズの商品カタログと数百万のアクティブなゲストを持つ大手小売業者にもうまくスケールする。サイトでのA/Bテストでは、ゲストのショッピング体験とゲストの消費量が大幅に改善された。

今後、小売業者は、パーソナライズド・カテゴリー特徴とパーソナライズド・アイテム特徴からの洞察を組み合わせたモデルを検討することを推奨する。さらに、同時消費はリピート消費と何らかの固有の関係を持つため、アイテムやカテゴリー間の相互興奮を考慮することを推奨する。

## REFERENCES

[1] 1959. The Pattern of Consumer Purchases. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 8, 1 (1959), 26–41. http://www.jstor.org/stable/2985810
[2] 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*. 62–70.
[3] Chris Chatfield and Gerald Goodhardt. 1973. A Consumer Purchasing Model with Erlang Inter-Purchase Times. *J. Amer. Statist. Assoc.* 68 (1973), 828–835.
[4] Suvodip Dey, Pabitra Mitra, and Kratika Gupta. 2016. Recommending Repeat Purchases Using Product Segment Statistics. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/2959100.2959145
[5] P S Fader, B G Hardie, and K Lee. 2009. Probability Models for Customer-Base Analysis. *Journal of Interactive Marketing* 23 (2009).
[6] Sofia Gomes and João M Lopes. 2022. Evolution of the online grocery shopping experience during the COVID-19 Pandemic: Empiric study from Portugal. *Journal of Theoretical and Applied Electronic Commerce Research* 17, 3 (2022), 909–923.

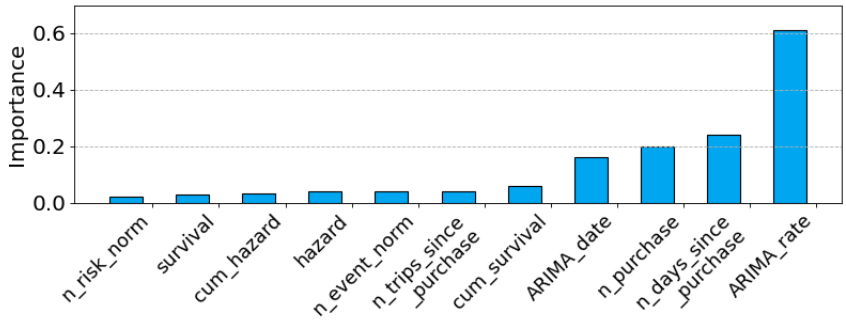



**Figure 2: Relative importance of input features to PC model**

| Trained on | Test Timeframe | NDCG (Test) | |
|---|---|---|---|
| | | Most Engaged | All |
| All | 7 days | 0.2009 | 0.2325 |
| All | 1 day | 0.3501 | 0.3583 |
| Most Engaged | 1 day | 0.3602 | 0.3589 |

| | Lift (%) |
|---|---|
| CTR | 6 |
| Conversion | 8.5 |
| Units | 27.5 |

**Table 4: Test results (a) Modifications in performance of PC model with changes in training data selection and testing timeframe. (b) Measuring impact of BIA against FBought in online A/B test.**

- CTR or Click Through Rate : Percentage of recommendation displays which were clicked by the guest.
- Conversion Rate: Percentage of clicked recommendations which were purchased by the guest same day.
- Units: The total number of units purchased by the users who were part of the treatment.

### 4.4 A/B testing results

When we introduced Buy It Again recommendation lists to the guest shopping experience, we A/B tested PCIC against a baseline of FBought. The results are given in Table 4(b). We can see that there is significant lift across all three metrics - 6% in CTR, 9% in Conversion and 27% in units purchased.

We also tested adding a Buy It Again recommendation list to the search results of all users. For this, we filtered the Buy It Again results using the search query context.We found that the user interaction with this recommendation list was significantly higher than existing search results (by over 20%). It was observed that the add-to-carts, average order values, and units per order went up by 0-2% (including all guest visits where guests looked for new items).

### 4.5 Building virtual aisles

We then rolled out BIA to guests by filtering recommendations by categories (Milk, Yogurt, Beauty, etc) to create a virtual aisles experience for online users in a dedicated space in App/site. We use the personalized list of categories for each guest using PC model. For each category, we present a list of recommended items from IC model to form a virtual aisle. In each aisle, we first showed the BIA items of the guest followed by other relevant items. Users who interacted with these recommendations had a significant increase in units per order (25-50%), and average order value (7-35%). Since the buy it again essentials are lower ticket items, they have a smaller dollar impact in order value than units per order. We saw higher guest engagement with virtual aisles experience in the App than in the site.

## 5 FUTURE DIRECTIONS

Buy It Again recommendations help users to quickly complete their shopping missions. Traditional approaches tend to model guest personalized behavior at item granularity. In this paper, we present the case for a coarse grained model which can capture the customer behavior at item category level. The proposed Personalized Category (PC) model combined with Items-within-Category (IC) model outperform existing BIA and NBR models on standard public datasets. The PCIC model also scales well for large retailers with millions sized product catalogs and millions of active guests. The A/B tests on the site show a significant improvement in guest shopping experience and guest spends.

In the future, we would recommend that retailers explore models that combine the insights from Personalized Category features with Personalized Item features. Moreover, we would recommend considering mutual excitation among items and categories as simultaneous consumption has some inherent relationship with repeat consumption.

## REFERENCES

[1] 1959. The Pattern of Consumer Purchases. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 8, 1 (1959), 26–41. http://www.jstor.org/stable/2985810
[2] 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*. 62–70.
[3] Chris Chatfield and Gerald Goodhardt. 1973. A Consumer Purchasing Model with Erlang Inter-Purchase Times. *J. Amer. Statist. Assoc.* 68 (1973), 828–835.
[4] Suvodip Dey, Pabitra Mitra, and Kratika Gupta. 2016. Recommending Repeat Purchases Using Product Segment Statistics. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/2959100.2959145
[5] P S Fader, B G Hardie, and K Lee. 2009. Probability Models for Customer-Base Analysis. *Journal of Interactive Marketing* 23 (2009).
[6] Sofia Gomes and João M Lopes. 2022. Evolution of the online grocery shopping experience during the COVID-19 Pandemic: Empiric study from Portugal. *Journal of Theoretical and Applied Electronic Commerce Research* 17, 3 (2022), 909–923.

[7] Gary L. Grahn. 1969. NBD Model of Repeat-Purchase Loyalty: An Empirical Investigation. *Journal of Marketing Research* 6 (1969), 72 – 78.
[8] Ruining He, Wang-Cheng Kang, Julian J McAuley, et al. 2018. Translation-based Recommendation: A Scalable Method for Modeling Sequential Behavior.. In *IJCAI*. 5264–5268.
[9] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.
[10] 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*. 1491–1499.
[11] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1071–1080.
[12] Komal Kapoor, Mingxuan Sun, Jaideep Srivastava, and Tao Ye. 2014. A hazard based approach to user return time prediction. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (08 2014). https://doi.org/10.1145/2623330.2623348
[13] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM* 40, 3 (1997), 77–87.
[14] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.
[15] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.
[16] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
[17] 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 network. In *IJCAI International Joint Conference on Artificial Intelligence*.
[18] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 729–732.

[7] Gary L. Grahn. 1969. NBD Model of Repeat-Purchase Loyalty: An Empirical Investigation. *Journal of Marketing Research* 6 (1969), 72 – 78.
[8] Ruining He, Wang-Cheng Kang, Julian J McAuley, et al. 2018. Translation-based Recommendation: A Scalable Method for Modeling Sequential Behavior.. In *IJCAI*. 5264–5268.
[9] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.
[10] 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*. 1491–1499.
[11] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1071–1080.
[12] Komal Kapoor, Mingxuan Sun, Jaideep Srivastava, and Tao Ye. 2014. A hazard based approach to user return time prediction. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (08 2014). https://doi.org/10.1145/2623330.2623348
[13] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM* 40, 3 (1997), 77–87.
[14] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.
[15] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.
[16] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
[17] 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 network. In *IJCAI International Joint Conference on Artificial Intelligence*.
[18] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 729–732.