Recency Aware Collaborative Filtering for Next Basket Recommendation

Guglielmo Faggioli
faggioli@dei.unipd.it
University of Padova
Padova, Italy

Mirko Polato
mpolato@math.unipd.it
University of Padova
Padova, Italy

Fabio Aiolli
aiolli@math.unipd.it
University of Padova
Padova, Italy

ABSTRACT

E-commerce and online services are getting more and more ubiquitous day by day. Like many other e-commerce paradigms, online grocery services can highly benefit from recommender systems, especially when it comes to predicting users' shopping behavior. This specific scenario owns peculiar characteristics, such as repetitiveness and loyalty, which makes the task very different from the standard recommendations. In this work, we present an efficient solution to compute the next basket recommendation, under a more general top-n recommendation framework. We propose a set of collaborative filtering based techniques able to capture users' shopping patterns. Furthermore, we analyzed how recency plays a key role in this particular task. We finally compare our method with state-of-the-art algorithms on two online grocery service datasets.

KEYWORDS

next basket analysis, grocery recommendation, collaborative filtering, popularity, recency

ACM Reference Format:
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 '20), July 14–17, 2020, Genoa, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340631.3394850

1 INTRODUCTION

Among the plethora of online services that have arisen in recent years, online grocery shopping services are getting more and more popular day by day. Like many other online services, grocery shopping services can highly benefit from the presence of a recommender system for several reasons. First, a recommender system allows users to save time, suggesting them products they usually buy, by learning users' needs, that in this setting tend to be highly linked to essential human needs.

A second aspect that makes a recommender system useful in grocery recommendation is related to diversity. Often, when it comes to doing grocery shopping, users tend to be repetitive, buying almost always the same articles out of habit, or for fear of not being satisfied with the alternatives. If users learn to trust the recommender system, it gets easier to propose new products allowing stakeholders to follow precise marketing strategies or to direct customers toward the maximum possible satisfaction.

Thanks to the amount of available information, nowadays, it is possible to highly personalize users' recommended baskets. A big challenge is those old techniques, based for example on association rules [11, 19], are not efficient enough to deal with the huge amount of data and the combinatorial nature of the problem [2]. It is necessary to develop a system that both takes into account the intrinsic nature of the problem, and that allows fast and easy computation.

In this work, we propose a highly efficient and parallelizable recommendation framework that exploits the natural human tendency to be repetitive in this scenario. Our framework is built on collaborative filtering (CF) [3] for top-N item recommendation. The proposed method relies on two main aspects: (i) popularity, especially user-wise, which is intrinsically a required feature in grocery recommendation, and (ii) recency, since both users and stores are subject to drift. Specifically, these two aspects are merged in a single predictor, called *recency aware user-wise popularity*, which is then used as "feedback" inside a collaborative filtering approach. Empirical analysis shows how our proposed framework can reach state-of-the-art performance and how important to consider a recency window is in building our model.

This paper is organized as follows: Section 2 discusses the state-of-the-art in the next basket recommendation task, with particular attention toward the specific setting of grocery shopping. Section 3 describes our recommendation framework with a thorough discussion about the importance of both popularity and recency in this setting. Section 4 is dedicated to the experiments on two publicly available online grocery service datasets. We describe in detail our empirical results, the datasets and the carried out analysis. Finally, results are compared against state-of-the-art algorithms, showing that we can achieve more accurate recommendations in terms of nDCG.

2 RELATED WORK

Even though Market Basket Analysis represents a quite old research field, it has gained a new vitality in recent years, thanks to the increase in online grocery shopping services. Most of the seminal works in this field, like [6] and [7], relied on the use of association rules, finding items or groups of items that give evidence on the presence of other items in the same basket.

One of the main recommendation techniques in this field is presented in [18]. The model, dubbed FPMC, learns a transition cube, which can be seen as a set of slices of shape $m \times m$, where m is

UMAP '20, July 14–17, 2020, Genoa, Italy

ACM ISBN 978-1-4503-6861-2/20/07…$15.00
https://doi.org/10.1145/3340631.3394850

the number of items in the catalogue, for each user. The cell (u, i, j) of this transition cube corresponds to the maximum likelihood estimation of finding item i in a basket for the users u given that the item j was in her previous basket. The algorithm learns a low rank factorization of the transition cube, using SBPR (sequence bayesian personalized ranking) as optimization criterion to find the factorization. SBPR is a variation to the well-known BPR general approach [17]. This criterion optimizes AUC, which is known of being a controversial metric in this context as it gives no biases towards the top of the ranking.

The approaches presented in [23] and [24] are based on representational learning, but they highly differ each other on how the representation is learnt and used. In [23] authors highlighted three main patterns in grocery purchases: *complementary* (items bought together), *compatibility* (items compatible to user's tastes) and *loyalty* (items bought because the user is loyal to the brand). In that work, the proposed method, called *triple2vec*, tries to learn a representation for users and items: it considers triples with the form (i, j, u) where i and j are items appearing in the same basket of the user u. Then, one of the three elements is removed and the remaining two are used as context (similarly to word2vec [14]) to predict the removed one. Finally, for a user-item pair, given its representation and the items of the previous basket, it is computed the probability that such item belongs to a basket. Additionally, probabilities learnt in the previous phase are weighted by the user's loyalty toward the item (the empirical user-wise frequency of purchase). In order to do so another algorithm, called *adaLoyal*, is used. In [24] a hybrid representation of users and items is proposed. In particular, given a vector representation for each user and item, a basket is represented by a pooling combination of all vectors representing items contained in the basket. Finally, the probability of a product being in the current basket for a specific user, given her previous basket, is obtained with the scalar product of the item's representation and the hybrid representation of the user at the previous basket, normalized over all possible items.

A traditional dichotomy in literature considers two possible paradigms for the next basket prediction problem: sequential recommender and general recommender. Sequential recommender systems aim to predict which will be the user's behavior based on her recent interactions with the system and their temporal ordering. General recommenders, on the other hand, like [3, 18], do not take into account sequential information to produce recommendations, but rather focus on users' general tastes.

A large body of research in this field comes also from industry. For example, [25] aims at studying shopping behavior and a multilevel basket recommendation system is proposed. Authors point out that customers value greatly the possibility of saving time, thus grocery websites should have basket autocompletion powered by recommender systems. They also noted that grocery shopping is both *multitasking* and *multi-people*. "Multitasking" means that the user has different goals in mind when doing shopping: she might want to buy food for the dinner, cleaning products or, for example, products for a party. Often these tasks are not related to each other. "Multi-people" means that users often do not buy only according to their tastes, but rather considering also, for example, their partner or family members: we can imagine a family where someone is vegan, someone else celiac, someone prefers a specific kind of cookies and so on. Moreover, authors highlight how online grocery is a frequent and repetitive activity with repurchases of the same group of items over time, and thus, it makes sense to track the item's inter-purchase interval and recommend them accordingly. The proposed recommender system first identifies categories of products that are often sold together, then it considers the partial basket of the user and ranks items accordingly to their complementarity with items already in the basket. Finally, timing is taken into account, suggesting products that are more likely to be bought by the user, due to the interval of time passed since the previous purchase of the product. Other relevant works in this context are: food recommendation [20], automatic promotion generation [15] and modelling price sensitivity [22].

In the past, the time has been taken into account by several recommendation algorithms. It seems a key aspect when it comes to making grocery recommendations since the set of available products might be subject to seasonality (concept drifts), as well as, users' tastes may change over time. Recency-based rating prediction [8, 9] is one of the oldest approaches in the literature. The proposed technique in both [8] and [9] deal with recency by means of weights proportional to the timestamp of the last rating of the target item (or related items). The importance of recency is also underlined in the study [16]. Here, experiments, on different CF algorithms on purchases of a French store, show, on average, that using only recent information leads to better performance w.r.t. using the whole history. In the context of One-Class CF, [21] proposes to "guess" negative items on the basis of their recency: the main hypothesis is that less recent items are more likely of being true negatives.

3 PROPOSED FRAMEWORK

Let $\mathcal{U}$ be the set of users of cardinality n, and let $\mathcal{I}$ be the set of items of cardinality m. For each user u, we consider an ordered set of transactions b_u^t where t indicates the ordinal position of the grocery shopping, with b_u^1 being the first basket and $b_u^{B_u}$ the last one for u. We can define $\mathcal{B}_u = \{b_u^t | t \in 1, \ldots, B_u\}$ as the set of transactions of a specific user u. We also define the set of baskets of a user u containing a specific item i as $\mathcal{B}_u^i = \{b_u^t | b_u^t \in \mathcal{B}_u \wedge i \in b_u^t\} \subseteq \mathcal{B}_u$, $|\mathcal{B}_u^i| = B_u^i$. In the remainder, the terms *transaction* and basket will be used interchangeably, due to the fact that, in both cases, we are referring to an unordered set of items.

3.1 On the effect of popularity

Popularity, as already pointed out in [5, 13, 23], represents one of the strongest strategies in grocery recommendation.

Grocery, as noted by [25] is both "multi-people" and "multitask". In practical terms, being multi-people means that we cannot assume purchases are results of the tastes of a single person but can be rather a complex compound of different tastes of many people (e.g., a family). The challenge related to multitasking is that tasks (e.g., satisfying the hunger, thirst or personal hygiene needs) are often not correlated and thus the complete grocery cannot always be guessed by few products.

Another aspect that needs to be considered about grocery tasks is unfrequent purchases that are not directly linked to a specific human need but depend on specific situations. For example, assume a young man desires to surprise his newly met partner with a homemade romantic dinner. Being a young person who lives alone, generally speaking, we can imagine him not being used to spending much time cooking, probably he is used to buying frozen food. But since he needs to please someone, probably he will buy fresh, and possibly expensive, products. If we would consider directly only the grocery done for the romantic dinner to predict the next one, we would ignore the user's behavioral pattern, for example suggesting her fresh products and ingredients, while, he generally prefers to buy frozen food or pre-cooked meals. User-wise popularity can overcome these limitations, the problem of multi-people groceries, in this case, can be addressed by not focussing on specific user's aspects, but rather considering the frequency of her purchases without posing any bias on user's tastes (like, e.g. modeling if the user is vegan, or her peculiar features). Secondly, multi-tasking limitation is addressed thanks to the fact that recommended products do not need to be related to each other. Finally, we implicitly ignore those baskets that represent a variation from the user's standard habits (since they will end up being irrelevant in the final model).

Note that, rare baskets can be ignored because our task is to recommend the next basket. If we would like to build an intra-basket recommender system, ignoring the specific items the user is buying would hardly be a good strategy.

The user-wise popularity for a given item i is defined as:

$$\pi_i^u = \frac{B_u^i}{B_u}.$$

Note that, the concept of popularity here expressed is very similar to the concept of *reminder* presented in [12]. In the following we will also refer to the global popularity of an item, which is, by definition, independent from a single user, and it can be defined as

$$\pi_i = \frac{\sum_{u \in \mathcal{U}} B_u^i}{\sum_{u \in \mathcal{U}} B_u}.$$

3.2 Recency

As previous literature (e.g. [24]) pointed out, a common pattern in grocery shopping is seasonality and drifts in items' popularity. We can expect specific products to have spikes in their popularity. For example, Christmas decorations will suddenly become popular during Christmas time, while almost being ignored during the rest of the year. Other products' popularity, on the other hand, changes more slowly. For example, seasonal vegetables can gain popularity when their season begins and lose it when the season ends.

A second aspect to be considered is the items' availability. We cannot expect all items to be always available in the service's catalog. Consider for example Figure 1. The precise day of the purchase is not available in `Instacart` (the dataset used to build said plots), yet it is possible to compute the difference in days between the first and the last purchase. Therefore, to build the plots presented in 1, we considered only users that have a difference between the first and the last basket of at least 335 days (maximum one year). This way, we can align baskets belonging to different users with, at worst, an error of one month. These plots show how the cumulative purchase frequency of specific item changes throughout the year.

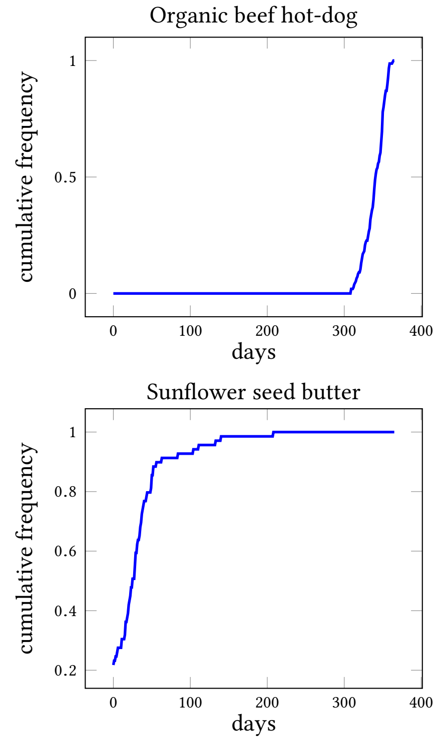

Figure 1: Examples of an (top) item entered lately in the catalogue, and (bottom) an item which went out of the catalogue. These data are taken from `Instacart`.

It is possible to observe that the item "*Organic Beef hot-dog*" (left-hand side of Fig. 1) hasn't ever been bought before approximatively around the 9th-11th month, and thus we can expect the item to not being available prior to that date. On the other hand, all purchases involving "*Sunflower seed butter*" happened in the first 6-8 months (right-hand side of Fig. 1). Because of that, it is possible to suppose that the item was removed from the catalog.

With changes in popularity for items (either because of trends or because of changes in their availability), we can expect to observe drift in user-wise items' popularity. These drifts might indicate to not recommend again the item since the user has lost interest in it, or, conversely, to start to recommend it more often since the user has suddenly gained interest. To take into consideration the drifts in items' popularity and time-related aspects of this setting, we included in our model the concept of *recency aware user-wise popularity*.

The recency aware user-wise popularity is defined as

$$\pi_i^u@r = \frac{\sum_{t=[B_u - r]_+}^{B_u} [\![i \in b_u^t]\!]}{\min(r, B_u)},$$

where r is the size of the recency window, $[\![x]\!]$ is the indicator function which returns 1 if the predicate x is true, 0 otherwise, and $[x]_+$ indicates the maximum between x and 0. Clearly, if $r \geq B_u$ then $\pi_i^u@r = \pi_i^u$. In a similar way it is also possible to define a *recency aware global popularity*.

3.3 Popularity-based CF

A standard collaborative approach, regardless of whether it is user-based or item-based, recommends items on the basis of a weighted combination of ratings (explicit or implicit). For instance, a user-based CF algorithm recommends to a user u items that have been appreciated by similar users (for some notion of similarity). In this section, we propose a different collaborative approach in which ratings are replaced by the user-wise popularity (either recency aware or not). In the context of grocery recommendation, the popularity, especially user-wise, is the most reliable indicator of how much a user likes (or needs) an item. For this reason, it seems reasonable to consider it as a surrogate of an explicit rating. In the following sections, we instantiate the Popularity-based CF for both the user-based and item-based approaches.

3.4 Item popularity-based CF (IP-CF)

Often products, although very different, share features that make them really similar. For example, we might have a low budget hand-soap and low budget frozen pizza: products themselves are highly different, but still share a common characteristic. This holds true for other features, like biological, cruelty-free products and so on. To recommend the user with her ideal basket, we might want to determine which products are similar to those bought by the user herself and recommend her with those.

We call $\mathcal{B}_i$ the set of transactions (i.e., baskets) in which the item i appears. More so, we define the similarity between two items as the asymmetric cosine similarity [4]:

$$s(i,j) = \frac{|\mathcal{B}_i \cap \mathcal{B}_j|}{|\mathcal{B}_i|^{\alpha}|\mathcal{B}_j|^{1-\alpha}} = p(i|j)^{\alpha} p(j|i)^{1-\alpha},$$

where $0 \leq \alpha \leq 1$ is a trade-off parameter which balances the importance of the probability $p(i|j)$ and $p(j|i)$. Note that, if $\alpha = 0.5$ the similarity corresponds to the cosine similarity of the items represented in the space of the basket.

3.4.1 Locality. The *locality*, as already presented in [4], is an hyper-parameter which represents the strength we want to use to enforce similarity. It is applied as an exponent to the similarity: $s(i,j)^q$, where $q \in \mathbb{R}_{\geq 0}$. Due to the fact that $s(i,j) \in [0,1]$ we have that, the higher the locality, the lower the similarity. Higher value of q means that in the neighbourhood of an item we want to keep only highly similar items, i.e., $s(i,j) \approx 1$. In the limit case of $q \to \infty$, $s(i,j) > 0$ if and only if $i = j$ or item i and j ($i \neq j$) are always bought together.

3.4.2 Prediction. Given the similarity $s(i,j)$, and the locality hyper-parameter q, we can define the score for an item i and a user u as:

$$\hat{r}_i^u = \sum_{j \in \mathcal{I}} s(i,j)^q \pi_j^u.$$

Using this scoring function, we will give high relevance to those items that are very similar to those appreciated (i.e., popular for u) by the user. Note that, having $s(i,i) = 1$ and thus $\hat{r}_i^u \geq \pi_i^u$, by default we are giving a bias toward user-wise popular items: this is positive since popularity is a good baseline and gives very good results in practice.

This method differs from the standard user-wise popularity because it is able to sort less popular items or items that have never been bought before. We expect this method to perform much better in a real environment w.r.t. the popularity because (i) for a regular user who always buys the same set of items, such a group of items will be recommended, and (ii) users that tend to vary their purchases will be satisfied as well since the recommender is able to suggest products similar to what the user usually buys even though they have never been purchased before. In the remainder with IP-CF@r, we will refer to IP-CF based on the recency aware user-wise popularity.

3.5 User Popularity-based CF (UP-CF)

The Item popularity-based CF method aims at recommending baskets that contain popular items similar to the ones of the target user, but it might lack in capturing potential novel items that fit well a particular user appreciated by other similar users. This can be addressed by considering a user-based collaborative approach that relies on similar users to find new items that can be of interest to the target user.

Akin IP-CF, as similarity function we employ the asymmetric cosine similarity:

$$w(u,v) = \frac{|\mathcal{I}_u \cap \mathcal{I}_v|}{|\mathcal{I}_u|^{\alpha}|\mathcal{I}_v|^{1-\alpha}},$$

where $0 \leq \alpha \leq 1$.

3.5.1 Locality. In this case, the locality hyper-parameter q acts on the similarity between users, i.e., $w(u,v)^q$. With high locality, we are allowing only users that are really similar to the target one, to contribute to the final score. In our setting, this corresponds to select only those users that have bought at least once all products bought by the target user. Note that when $q = \infty$, unless there are users v for which $\mathcal{I}_u \subseteq \mathcal{I}_v$, that is a rare event, the similarity will be non-zero only between the user and herself.

3.5.2 Prediction. We already know the importance of user-wise popularity to define a user's shopping patterns, thus, we want to combine that information with the one gained through users' segmentation. The score for item i to user u is:

$$\hat{r}_i^u = \sum_{v \in \mathcal{U}} w(u,v)^q \pi_i^v.$$

In the remainder with UP-CF@r we will refer to UP-CF based on the recency aware user-wise popularity.

4 EXPERIMENTS

In this section we present the evaluation of our framework on two public datasets. We will describe the used datasets, the pre-processing and the splitting in the training, validation and test set. Subsequently, we discuss the effect of the recency window, and how it affects the performance of the user-wise popularity. Then, the achieved results are presented and compared to the ones achieved by the baseline algorithms. Finally, we briefly discuss the computational complexity of different methods.

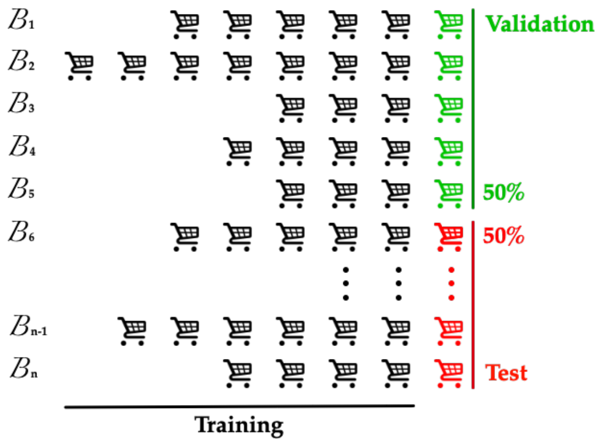

Figure 2: Training, validation and test set division of the datasets. Black baskets are the one kept for the training, green ones (50% of last baskets) are used to select the best hyper-parameters (validation), and the red baskets (the remaining 50% of last baskets) are used as test set.

4.1 Datasets

Both datasets used are publicly available and they can be considered as benchmarks on the next basket recommendation task:

- Dunnhumby[1] - The Complete Journey: this dataset contains information on transactions at household level from 2500 frequent shoppers;
- Instacart [1]: this dataset was initially used in the Instacart challenge. It contains information regarding over 3 millions online purchases on the Instacart grocery service.

Details of both datasets are reported in Table 1.

The two datasets differ under many aspects: Dunnhumby's users are biased toward frequent shoppers, with a high average of transactions per user, while Instacart's users are more uniformly distributed, with some loyal users (with a couple of transactions per week), and occasional users with few purchases over the entire year of data.

To use the Dunnhumby dataset, given the number of items, we first applied a light preprocessing removing all items that appear in less than 10 baskets. Users with less than 2 baskets after the previous removal have been excluded from further analysis.

In order to build and validate the proposed methods we split the dataset as in the following:

Training set is composed by all basket but the last one of all users (black baskets in Figure 2);

Validation set is used to select the best hyper-parameters of the methods, and it composed by the last baskets of 50% of all users randomly selected (green baskets in Figure 2);

Test set is the set of remaining 50% of last baskets (red baskets in Figure 2).

For our methods, the validation has been performed considering the following set of hyper-parameters' values:

Recency window $r \in \{1, 5, 25, 100, \infty\}$;

Locality $q \in \{1, 5, 10, 50, 100, 1000\}$;

Asymmetry $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$.

The reported results are the averages over fifty repetitions of the 50%-50% random splits as previously described.

4.2 Baseline methods

We compare our methods against four baselines:

Item-based CF (IB-CF) : standard item-based collaborative filtering approach [3]. We considered a positive (implicit) feedback each item that appears in at least one basket of a user. As for similarity function, the cosine similarity has been used;

User-based CF (UB-CF) : standard user-based collaborative filtering approach [3]. The setting is the same as of IB-CF;

Global popularity (GPop) : (see Section 3.1) this baseline represents the simplest approach to many recommendation tasks and consists in recommending items that appear often throughout the dataset;

User-wise popularity (UWPop) : (see Section 3.1) users are recommended items they usually buy, sorted by frequency of purchases. In this task, user-wise popularity is one of the strongest baselines. When the recency is considered the method will be called UWPop@r;

FPMC : this method represents a state-of-the-art approach for the next basket prediction. FPMC is based on the optimization of the AUC and takes into account both the user's tastes and sequential behavior (see Section 2). The tests have been performed using the implementation provided at https://github.com/khesui/FPMC;

Triple2vec+AdaLoyal (T2V+Ada) : state-of-the-art approach described in Section 2.

T2V+Ada's implementation used in our experiments is available at https://github.com/MengtingWan/grocery. To compute results for T2V+Ada on Instacart, we used the best parameterization suggested on the related GitHub page. On Dunnhumby, for technical reasons, we reported results obtained by the best parametrization over the test set, hence the achieved nDCG is optimistic and a bit unfair w.r.t. to the other methods. In particular, we evaluated 4 possible values for the Triple2vec embedding dimension (the same suggested in the original paper), i.e., $\{16, 32, 64, 128\}$, and four possible values for the loyalty initial value, i.e., $\{0.2, 0.4, 0.6, 0.8\}$.

4.3 Evaluation metric

To evaluate our methods, we used *normalized Discounted Cumulative Gain* (nDCG):

nDCG@k : it is an evaluation measure based on ranking, that favours those techniques that place highly relevant items. nDCG@k is defined as:

$$nDCG@k(R_u) = \frac{1}{IDCG@k} \sum_{i=1}^{k} \frac{\mathrm{rel}(R_{iu})}{\log(i+1)},$$

where

$$IDCG@k = \sum_{i=1}^{k} \frac{1}{log(i+1)}.$$

[1] https://www.dunnhumby.com/careers/engineering/sourcefiles

Dataset	users	items	items selected	# baskets	min baskets	max baskets	avg # baskets per user
Dunnhumby	2500	92353	25610	269502	2	514	88.25
Instacart	206209	49685	49685	3346083	3	100	16.22

Table 1: Details of the datasets used in the experiments.

It is usually well suited to evaluate recommender systems, especially when applied to real tasks since only a handful of items can be shown to the final user and thus it is important to rank highly relevant items.

In the experiments, we fixed $k = 5, 10, B$, where B stands for the length of the test basket.

We also evaluated the methods in terms of AUC (Area Under the ROC Curve) as done in [18]. However, we argue that this metric is not suitable for this task since it does not give any bias towards the top of the ranking. For this reason, we do not report the AUC results although they confirm the reported ones.

4.4 The effect of the recency window

Figure 3 shows how nDCG@B changes when we take into account the recency to build a recommendation based on user-wise popularity.

Both plots exhibit the same pattern, with an initial increase in performance and a peak. For Dunnhumby this peak is between 20 and 30 baskets considered, while for Instacart the peak is when 5-10 baskets are taken into account. We can expect this to be associated with the different dimensions of the datasets: Instacart has a smaller maximum and an average number of baskets per user, while both the measures are greater in Dunnhumby (see Table 1).

Note that, even though this result confirms what was theoretically supposed in subsection 3.2, we can compute only an approximation of the results obtained by real recency driven approach. Instacart does not provide exact purchase dates, and hence we deliberately decided not to use this information on Dunnhumby, also. By applying a recency window over the number of baskets, we are deliberately assuming that the period covered by the most recent r baskets for each user is approximatively the same. This represents a strong hypothesis that is likely to be violated by many users' histories. Users do not buy their groceries after the same number of days, with possibly months passed between two consecutive baskets. Yet, even with such a loose hypothesis, plots suggest a correlation between the period and quality of the predictions. In a setting where purchase dates are available, it might be possible to compute much more significant results using a recency window.

4.5 Experimental results

4.5.1 Comparison against baselines. Tables 2 and 3 show how the proposed approach, compared against several baselines, it is able to achieve state-of-the-art performance.

Three different possible metrics have been reported: nDCG@5, nDCG@10, and nDCG@B. We selected nDCG@5 and nDCG@10 because they represent typical recommendation settings. A 10 elements long list, can fit decently enough on a modern laptop screen thus we aim to show, in general, how much satisfied will be the user when presented with that list. A 5 elements list, conversely, might

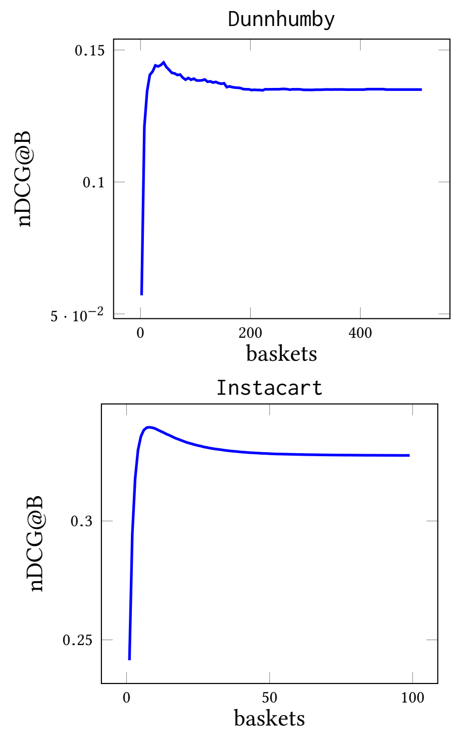

Figure 3: nDCG@B of UWPop@r varying the recency window r on Dunnhumby (top) and on Instacart.
(bottom).

algorithm	nDCG@B	nDCG@10	nDCG@5
GPop	0.052	0.095	0.105
UWPop	0.136	0.187	0.197
IB-CF	0.071	0.094	0.106
UB-CF	0.053	0.073	0.085
FPMC	0.087	0.106	0.110
T2V+Ada	0.146	0.190	0.199
UWPop@r	0.142	0.193	0.202
IP-CF@r	0.144	0.196	0.206
UP-CF@r	**0.149**	**0.202**	**0.212**

Table 2: Results on Dunnhumby's dataset. Highest results on each metric are highlighted in bold.

represent the available space on a mobile phone: using nDCG@5 we can approximate the satisfaction of a user, browsing the grocery service's catalog with her mobile phone. Finally, nDCG@B

algorithm	nDCG@B	nDCG@10	nDCG@5
GPop	0.081	0.109	0.098
UWPop	0.327	0.387	0.406
IB-CF	0.206	0.256	0.256
UB-CF	0.123	0.147	0.166
FPMC*	0.089	0.108	0.115
T2V+Ada	0.302	0.358	0.367
UWPop@r	0.336	0.395	0.415
IP-CF@r	0.346	0.408	0.427
UP-CF@r	**0.349**	**0.411**	**0.429**

Table 3: Results on Instacart's dataset. *FPMC has been trained on 5% of the whole dataset, since the full training did not finish in 24 hours of computation. Highest results on each metric are highlighted in bold.

represents the hardest case possible: the iDCG is maximum, and thus algorithms that are unable to guess most of the basket will be highly penalized. This also explains the consistently lower results in nDCG@B w.r.t. other metrics.

UP-CF@r can achieve constantly the highest results in both datasets, followed by the other proposed algorithms (IP-CF@r and UWPop@r) The quality of the proposed algorithms is partly due to the high performances the user-wise popularity itself can achieve. In both datasets, considering the user-wise popularity alone grants satisfying results.

Validating on r, the size of the recency window, seemed a performing strategy: the recency aware user-wise popularity approach can achieve consistently about 1% more than its counterpart without the recency.

It is also remarkable that, as previously suggested, in this task classical collaborative filtering algorithms are unable to achieve good results, and thus they require strong modifications to become competitive.

On Dunnhumby, even though UP-CF@r achieves consistently higher performance than T2V+Ada, both algorithms perform very well compared to other baselines. We justify the higher results of T2V+Ada by considering the nature of the dataset itself. The dataset is smaller with roughly 2500 users and 25000 items in the filtered version, so it is easier for the Triple2Vec component to better represent items. Secondly, much more information is available for each user, given the much higher average number of baskets per user. This boosts the Adaloyal algorithm, granting general higher performances. Note that, the dataset seems to be in general a harder task w.r.t. Instacart all methods achieve worse results.

4.5.2 Hyper-parameters validation. As mentioned at the beginning of this section, for our methods we validated three hyper-parameters: the recency window r, the locality q, and the asymmetry α. Different methods have shown different sensitivity on these hyper-parameters, as well as, different datasets led to very different best parametrizations. On Dunnhumby, for both IP-CF@r and UP-CF@r the best performing recency window during validation has been $r = 25/50$, which is on average less than half the full history of the user. Regarding the locality, UP-CF@r worked best with $q = 5$ or $q = 10$, while for IP-CF@r higher values of q (e.g., 1000) has been usually selected during validation. For the asymmetry α, for both methods, it is difficult to draw a real trend. On Instacart the validation can be considered more stable. For both methods, $r = 5$ seems, on average, the best recency window setting, which corresponds to roughly a third of the average complete user history. Even on this dataset, UP-CF@r worked best with relatively low values of q (i.e., 10), like IP-CF@r which, differently from Dunnhumby, seemed to prefer low q (i.e., 5) as well. Finally, the asymmetry α had very different behaviors. For IP-CF@r the best asymmetry is the one which only considers $p(j|i)$ (i.e., $\alpha = 0$), that is the probability of seeing an item given that the target one is in the basket. While for UP-CF, the best performing asymmetry parameter has been $\alpha = 0.75$.

4.5.3 Complexity analysis. It is possible to make some observations on the complexity of the different methods, i.e., FPMC, T2V+Ada, and the proposed framework. FPMC is the oldest method and thus suffers the most from a complexity point of view: even though the method is highly elegant and theoretically well-founded, it requires a large number of computational resources, both from memory and time perspectives. For example, on Instacart we weren't able to obtain a complete computation over the entire dataset due to the required time. T2V+Ada is again an elegant method that can combine representational learning and popularity based information. Its main limitation is linked to the required computational time, in fact, by default, the implementation provided by authors samples the possible number of triples used to train the Triple2vec model. The most interesting aspect of this method is the capability of obtaining high results even with a subsampling, and a dimensionality reduction of the items' representation. Thus it represents a great resource when it is necessary to compute recommendation in a limited computational resource setting.

Finally, the proposed framework has its disadvantages. Its main limitation is linked to the memory consumption: computing the entire user-user similarity matrix used in UP-CF@r is highly memory expensive and might be unfeasible in specific settings. Yet, the framework is highly parallelizable [2]. It is possible to compute only the needed matrix row or computing bands of the matrix (this can be done in parallel), it is possible to obtain the results in an acceptable time (order of minutes). Moreover, the matrix is highly sparse and thus it can be efficiently handled using sparse matrices properties. We also argue that using only the k most similar users/items would be enough to get good performance. If so, then the memory consumption would be largely reduced.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented a collaborative filtering framework for the next basket recommendation. We have shown that the proposed set of approaches can achieve state-of-the-art performance. The two main characteristics that our framework leverages to succeed in this specific task are: (i) the popularity, which is known of being a really strong baseline, and (ii) the recency window, which helps in dealing with issues related to concept drift. These two aspects are combined in an elegant and versatile collaborative filtering

[2]The source code will be soon released.

framework which has empirically shown of being able to achieve state-of-the-art performance. One of the main advantages of our proposal is the high efficiency and capability of parallel execution, that are both very important features in an online setting. In the future, we plan to study more in-depth the correlation between products and time: we want to understand the correlation existing between a product and the time that passes between two consecutive purchases of the said product. Finally, we wish to understand better consecutiveness among products' purchases: as some literature [10, 18, 24] suggests. It is possible to gain information on the content of the next basket, thanks to the previous one; we aim to study this relation and possibly add it to our model.

REFERENCES

[1] Accessed on 2018. The Instacart Online Grocery Shopping Dataset 2017. https://www.instacart.com/datasets/grocery-shopping-2017.

[2] Andres G. Abad and Luis I. Reyes Castro. 2017. Collaborative Filtering using Denoising Auto-Encoders for Market Basket Data. *CoRR* abs/1708.04312 (2017).

[3] G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (June 2005), 734–749. https://doi.org/10.1109/TKDE.2005.99

[4] Fabio Aiolli. 2013. Efficient Top-n Recommendation for Very Large Scale Binary Rated Datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 273–280. https://doi.org/10.1145/2507157.2507189

[5] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The Dynamics of Repeat Consumption. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. ACM, New York, NY, USA, 419–430. https://doi.org/10.1145/2566486.2568018

[6] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. 1997. Dynamic Itemset Counting and Implication Rules for Market Basket Data. *SIGMOD Rec.* 26, 2 (June 1997), 255–264. https://doi.org/10.1145/253262.253325

[7] Yen-Liang Chen, Kwei Tang, Ren-Jie Shen, and Ya-Han Hu. 2005. Market basket analysis in a multiple store environment. *Decision Support Systems* 40, 2 (2005), 339 – 354. https://doi.org/10.1016/j.dss.2004.04.009

[8] Yi Ding and Xue Li. 2005. Time Weight Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05)*. ACM, 485–492. https://doi.org/10.1145/1099554.1099689

[9] Yi Ding, Xue Li, and Maria E. Orlowska. 2006. Recency-based Collaborative Filtering. In *Proceedings of the 17th Australasian Database Conference - Volume 49 (ADC '06)*. Australian Computer Society, Inc., 99–107.

[10] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction* 27, 3 (01 Dec 2017), 351–392. https://doi.org/10.1007/s11257-017-9194-1

[11] Choonho Kim and Juntae Kim. 2003. A recommendation algorithm using multi-level association rules. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*. 524–527. https://doi.org/10.1109/WI.2003.1241257

[12] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. 2016. On the Value of Reminders Within E-Commerce Recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization (UMAP '16)*. ACM, New York, NY, USA, 27–35. https://doi.org/10.1145/2930238.2930244

[13] Ming Li, Malcolm Dias, Ian Jarman, Wael El-Deredy, and Paulo Lisboa. 2009. Grocery shopping recommendations based on basket-sensitive random walk. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1215–1224. https://doi.org/10.1145/1557019.1557150

[14] Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. 1–12.

[15] Petteri Nurmi, Antti Salovaara, Andreas Forsblom, Fabian Bohnert, and Patrik Floréen. 2014. PromotionRank: Ranking and Recommending Grocery Product Promotions Using Personal Shopping Lists. *ACM Transactions on Interactive Intelligent Systems* (01 2014).

[16] Bruno Pradel, Savaneary Sean, Julien Delporte, Sébastien Guérif, Céline Rouveirol, Nicolas Usunier, Françoise Fogelman-Soulié, and Frédéric Dufau-Joel. 2011. A Case Study in a Recommender System Based on Purchase Data. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, 377–385. https://doi.org/10.1145/2020408.2020470

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

[18] 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 (WWW '10)*. ACM, New York, NY, USA, 811–820. https://doi.org/10.1145/1772690.1772773

[19] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. Analysis of Recommendation Algorithms for e-Commerce. In *Proceedings of the 2Nd ACM Conference on Electronic Commerce (EC '00)*. ACM, 158–167.

[20] Christoph Trattner and David Elsweiler. 2017. Food Recommender Systems: Important Contributions, Challenges and Future Research Directions. (11 2017).

[21] João Vinagre, Alípio Mário Jorge, and João Gama. 2015. Collaborative Filtering with Recency-based Negative Feedback. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15)*. ACM, 963–965. https://doi.org/10.1145/2695664.2695998

[22] Mengting Wan, Di Wang, Matt Goldman, Matt Taddy, Justin Rao, Jie Liu, Dimitrios Lymberopoulos, and Julian McAuley. 2017. Modeling Consumer Preferences and Price Sensitivities from Large-Scale Grocery Shopping Transaction Logs. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1103–1112. https://doi.org/10.1145/3038912.3052568

[23] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian 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 '18)*. ACM, New York, NY, USA, 1133–1142. https://doi.org/10.1145/3269206.3271786

[24] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. ACM, New York, NY, USA, 403–412. https://doi.org/10.1145/2766462.2767694

[25] Mindi Yuan, Yannis Pavlidis, Mukesh Jain, and Kristy Caster. 2016. Walmart Online Grocery Personalization: Behavioral Insights and Basket Recommendations. In *Advances in Conceptual Modeling*, Sebastian Link and Juan C. Trujillo (Eds.). Springer International Publishing, Cham, 49–64.