On the Value of Reminders within E-Commerce Recommendations

Lukas Lerche
TU Dortmund University
Dortmund, Germany
lukas.lerche@
tu-dortmund.de

Dietmar Jannach
TU Dortmund University
Dortmund, Germany
dietmar.jannach@
tu-dortmund.de

Malte Ludewig
TU Dortmund University
Dortmund, Germany
malte.ludewig@
tu-dortmund.de

ABSTRACT

Most research in recommender systems is focused on the problem of identifying and ranking items that are relevant for the individual users but *unknown* to them. The potential value of such systems is to help users discover new items, e.g., in e-commerce settings. Many real-world systems however also utilize recommendation lists for a different goal, namely to *remind* users of items that they have viewed or consumed in the past. In this work, we aim to quantify the value of such reminders in recommendation lists ("recominders"), which has to our knowledge not been done in the past. We first report the results of a live experiment in which we applied a naive reminding strategy on an online platform and compare them with results obtained through different offline analyses. We then propose more elaborate reminding techniques, which aim to avoid reminders of too obvious or of already outdated items. Overall, our results show that although reminders do not lead to new item discoveries, they can be valuable both for users and service providers.

Keywords

Recommender systems; e-commerce; reminders

1. INTRODUCTION

Recommender systems (RS) have become a common feature of modern e-commerce platforms. Many major web sites implement techniques to continuously suggest additional items to their customers when they navigate the website based on the customers' past profiles and their current navigation and item viewing behavior. The research literature often considers the main value of recommenders to be their ability to point users to potentially relevant items which the users are not aware of, thereby creating additional sales or activity on the site. In practice, however, a typical strategy of websites like Amazon.com is that some presented recommendation lists include almost exclusively items that the user has viewed in the past. In this case, the intended goal of the recommender is not item discovery or catalog exploration but to remind users of items that were of (recent) interest to them. Generally, recommending items that the user already knows can serve different purposes.

(1) Reminders as Shortlists: Before making a purchase decision, customers often browse the catalog to explore the set of alternatives. Reminders of recently visited items can serve as automated decision "shortlists" and thereby help to reduce the cognitive effort in online shopping [17, 22]. Similarly, customers might return to the site after having slept on a decision and reminders for items that were viewed in a previous session might be particularly valuable [12].

(2) Familiarity Aspects: Recommending familiar items can generally help to increase the customers' trust and willingness to purchase. Amazon.com, for example, relied on familiar items as recommendations for a long time [24].

(3) User Cold-Start: In cold-start situations, where little is known about the user, reminders can serve as a fallback before switching to more elaborate algorithms.

(4) Repeated Recommendations: Finally, in some domains like music or video recommendation, reminders help encourage users to revisit items that they have purchased or consumed in the past but may have forgotten [3, 16]. Such "repeated recommendations" can be based on long-term user profiles. In contrast, the focus of our work is on recommending items that the user knows (has viewed), but has not necessarily purchased so far. These reminders are in our view particularly helpful in scenarios where the user focuses on a smaller set of candidate items before making a decision.

In our previous work, we have explored – among other aspects – the usefulness of a naive "reversed history" reminding strategy using navigation logs of an online retailer [12]. The results showed that recommendations worked best in terms of information retrieval measures when they contain both reminders from the user's recent navigation history and collaborative filtering (CF) recommendations.

In this work, we first provide empirical evidence from deployed applications that show that (a) many online customers "accept" reminders in recommendation lists ("recominders") and (b) that reminders can help to generate business value. As our main contribution, we propose and empirically evaluate novel algorithmic approaches of reminding, which aim to avoid too obvious as well as outdated item recommendations and at the same time focus on reminders that are relevant for the user's current shopping context.

UMAP '16, July 13-17, 2016, Halifax, NS, Canada

DOI: http://dx.doi.org/10.1145/2930238.2930244

2. ANALYSIS OF RECOMINDERS IN DEPLOYED SYSTEMS

We evaluated the use and effectiveness of reminders in recommendations in the context of two real e-commerce sites.

2.1 User Acceptance of Recominders on an E-Commerce Site

First, we analyzed a dataset containing recommendation and user navigation logs provided to us by Zalando, a European online fashion retailer. From the comparably large dataset – more details in Section 4.3 – we sampled 3,000 "heavy users", for which we expect more reliable insights regarding their behavior than from one-time or infrequent visitors. The log file subset contains about 106,000 purchases and over 3.1 million *view actions* for over 188,000 different items[1]. An average shopping session of a user comprised about 9 item views and a purchase was made in about every third session.

The most important aspect for our analysis is that the item view actions in the log are usually accompanied by a three-element subset of recommendations that were displayed to the user when viewing the item. The click events on these recommended items were recorded in the log. Our log analysis revealed the following.

1. About every tenth of all *displayed* recommendations were reminders of already seen items, i.e., where a *view event* occurred before the recommendation was displayed. About one quarter of all recommendation lists contained at least one reminder.

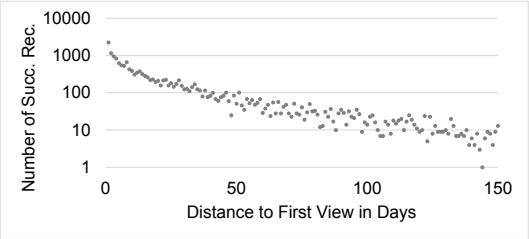
2. More than 40% of all *successful* recommendations, i.e., where the click on a recommended item led to a purchase in the same session, were reminders. Therefore, users often knew the recommended item they eventually purchase from a previous visit of its detail page.

3. For those items that were "purchased from a recommendation list", we observed that customers often had inspected this item several times (over the course of several days) before the purchase. This suggests that the recommendations served at least partially as navigation shortcuts. Figure 1 shows when and how often users viewed an item before they purchased it based on a suggestion in a recommendation list.

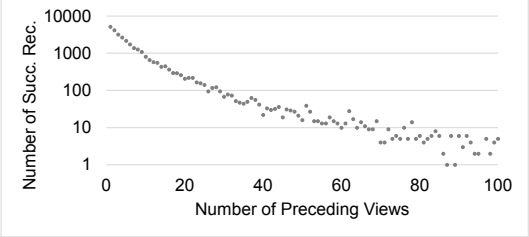
4. In general, in each session customers on average only look at items from about 2.7 different product categories (of 334 existing ones). This indicates that they have often very specific shopping goals when arriving at the site. In such situations, reminders could be particularly helpful as *shortlists* [17, 22] during the decision making process.

Overall, our analysis reveals that reminders occur in the recommendation lists of the examined transaction logs and that customers in practice use these reminders as a starting point for subsequent purchase actions. Clearly, we cannot infer with certainty whether these purchases would not have been made if there were no reminders. At least, however, we see that customers seem to rely on recommendation lists even if they contain items which they already know.

[1]The data was preprocessed by Zalando in a way that no conclusions about the real proportions in Zalando's live traffic can be drawn.



(a) Distance in days between first view event for an item and its successful recommendation.



(b) Number of preceding view events for an item before its successful recommendation.

Figure 1: Distribution statistics for successful recommendations (Zalando *heavy* subset; Y-axes in base-10 log scale.)

2.2 Effectiveness Analysis on an Electronics and Gadgets Website

Our second analysis is based on an A/B test performed on the website of China-Gadgets, a German platform for consumer goods and electronics from Chinese wholesale sites. On China-Gadgets, items are presented as blog articles, sometimes with a textual review. We implemented different strategies to generate four-item recommendation lists that were shown both below the first (most recent) article on the site's landing page and on each item's detail page. Besides a popularity-based baseline, the techniques included Bayesian Personalized Ranking (*BPR*) [20], two content-based strategies, as well as a "Recently Viewed" [12] strategy.

The user interactions after implementing the different recommenders were recorded over the course of three months. On average, there were about 1.8 sessions per user and 80% of the users only visited the shop once. In our subsequent analyses we focus only on the remaining 20% of the users who have visited the shop at least twice during the monitoring period. The resulting subset contains 287,000 item view events by 49,000 users for 4,100 products. About 2.6% of the views were induced by a recommendation.

To estimate the business value of the recommendations, we measure their *success rate*. For each recommended item that was displayed to the user we call the recommendation "successful" if the item was clicked and the user subsequently visited the external Chinese site. Similar to the analysis in the previous section, a substantial fraction (38%) of the successful recommendations were not new to the users. Including reminders in recommendations can therefore represent a promising strategy.

Figure 2 shows the average success rates of the different strategies. The reminding strategy actually worked best in this domain. The success rate is at 0.34% which is signifi-

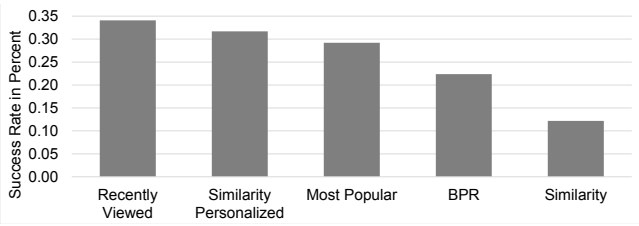


Figure 2: Success rate of the different recommendation strategies in the China-Gadgets live experiment.

cantly better[2] than the next best strategy *Similarity Personalized*, which recommends items similar to the user's history based on the TF-IDF representations of the textual item descriptions. Recommending the most popular items generally seems to be a reasonable strategy in this domain since we observe a comparably skewed popularity distribution of the items, as will be discussed later on in Section 4.3. The *BPR* method – whose model was re-trained every 30 minutes and whose parameters were optimized in an offline process – in contrast is less effective, presumably because there are only very few implicit ratings per user that can be used for learning. Finally, a *Similarity* recommender, which suggests items similar to the currently viewed item, led to the weakest performance.

All tested algorithms were "allowed" to recommend items that the user had viewed in the past. The *Similarity Personalized* method – without being designed for reminders – in fact produced lists in which about 75% of the items were already known by the users. The results therefore not only suggest that reminding can be beneficial also in this domain, but that other techniques of reminding can be successful.

3. ADAPTIVE REMINDERS

Our analyses so far show that placing (some) reminders in the recommendation lists can be an effective strategy. However, the simple "Recently Viewed" technique can easily lead to suggestions that are too obvious or already outdated. In the subsequent sections we will therefore design and evaluate different "adaptive" reminding strategies that can help to avoid some of these problems.

The general approach of all proposed strategies is to first determine a set of candidate items that the user has interacted with in the past and which can be used as reminders. The specific task is then to filter and rank these items with the goal to improve the recommendation accuracy and, if desired, to avoid too obvious suggestions. As the candidate set is usually only a small part of the whole item catalog, the reminders can be computed efficiently at runtime.

This set of candidate items H_u^I is taken from the user's recent history H_u which consists of a limited window of the user's previous sessions[3]. The choice of H_u will be discussed in more detail in Section 4.1. Each session is a set of interaction tuples of the form $\langle item, action, timestamp \rangle$. Possible actions for example include an item view, a purchase, or a shopping cart event. In this work, we will focus on past item *view* actions and use H_u^V to denote the set of all tuples describing the view actions in all sessions of H_u.

[2]According to a four field χ-squared test ($p < 0.05$) with Bonferroni correction to compare the number of successful and unsuccessful recommendations between two strategies.

[3]A session represents a sequence of user actions within a particular time period identified by a session ID.

3.1 Ranking Strategies

All proposed ranking strategies rely on assigning personalized scores to the recommendation candidates H_u^I based on heuristics. Items with a score of zero are not recommended[4].

Interaction Recency (IRec): This baseline recommends items depending on the time point of the last interaction of the user with an item, e.g., an item view event, a purchase or a shopping cart event. *IRec* therefore corresponds to a generalized "Recently Viewed" strategy as used in the previous section and in [12]. To make our results comparable with previous works, we only use the timestamps of the view events to compute a ranking score_{ui}^{IRec} for a user u and an item i. Let $H_u^V(i)$ be the set of interaction tuples in H_u^V concerning item i and $time()$ be a function that returns the timestamp of a past interaction tuple t, then

$$\text{score}_{ui}^{IRec} = \max_{t \in H_u^V(i)} (time(t)) \tag{1}$$

As we only include reminders in our recommendations, the "distance" between timestamps of the recommendations is not relevant. If reminders are mixed with novel recommendations, more weight could be given to the more recent interactions by applying an exponential decay function [3].

Interaction Intensity (IInt): If a user interacts longer or more frequently with an item, this *interaction intensity* indicates increased interest in the item. Here, we use the frequency of item view events in the interaction history, i.e.,

$$\text{score}_{ui}^{IInt} = |H_u^V(i)|. \tag{2}$$

If two items have the same score, the *IRec* strategy can be used as a fallback to determine the ranking order.

Item Similarity (ISim): Assume a user was searching for shoes two weeks ago, but did not purchase a pair in the end. One week ago, the user browsed for scarves and today, she searches again for shoes. The idea of the proposed *ISim* strategy is to find elements in the past interaction history that match the current shopping goal.

This proposed approach corresponds to a content-based hybrid technique. We first determine the similarity of the items in H_u^I to the items viewed in the current session of user u. Let C_u be the set of items appearing in the current session of u, then

$$\text{sim}_{item}(u,i) = \frac{\sum_{j \in C_u} \text{sim}_{cos}(i,j)}{|C_u|}. \tag{3}$$

In our experiments we use the cosine similarity sim_{cos} as a distance measure. The used feature vectors are TF-IDF representations of item descriptions or are created from additional item information like category and brand identifiers.

We then pick the k most similar items from H_u^I according to Equation 3, denoted as $\text{top}_k^I(u)$. To determine the final score score_{ui}^{ISim}, we re-sort these items by their view frequency according to the *IInt* heuristic, i.e.,

$$\text{score}_{ui}^{ISim} = \text{score}_{ui}^{IInt} \cdot \mathbf{1}_{\text{top}_k^I(u)}(i). \tag{4}$$

[4]In our work we focus on determining relevance scores and the ranking of *already known items*. Combining the reminding strategies with other algorithms is possible and might lead to mixed recommendations of known and new items. In these cases, the reminding techniques can provide (additional) relevance scores for the already known items.

The indicator function in Equation 4 leads to a score of zero for items not appearing in the top list. Other scoring techniques are possible that, e.g., use the similarity score in a weighted approach.

Session Similarity (SSim): Continuing the example from above, assume that two weeks ago the user also searched for belts that go well with the shoes. Once we observe that the user is again browsing for shoes, the previous *ISim* approach however would not remind the user of the belts.

To find good complementary items, like belts, from the user's past, we propose the *SSim* strategy that aims to find past sessions of a given user that had the same shopping intent as the current session. In contrast to the *ISim* technique, we rather look for similar sessions than for similar individual items. Let s be a past session of u and V_s the set of the viewed items of session s, then

$$\text{sim}_{session}(u, s) = \frac{\sum_{i \in V_s} \sum_{j \in C_u} \text{sim}_{cos}(i, j)}{|V_s| \cdot |C_u|}. \quad (5)$$

Let $\text{top}_k^S(u)$ be the k most similar sessions when compared with the current session according to Equation 5. Also, let $Items(\text{top}_k^S(u))$ be the set of items appearing in these sessions. Like in the *IInt* method, we re-rank the items that appear in the top k sessions based on their view frequency.

$$\text{score}_{ui}^{SSim} = \text{score}_{ui}^{IInt} \cdot \mathbf{1}_{Items(\text{top}_k^S(u))}(i) \quad (6)$$

Note that both the *SSim* and *ISim* strategies are still useful when multiple users share an account, because the items are selected based on the most recent user actions and therefore fit the interests of the currently active user.

3.2 Feature-based Category Filtering

In certain e-commerce domains repeated purchases for some item categories can be uncommon. Therefore, at least for some time, no items should be recommended that belong to a category in which a user has recently made a purchase. The proposed ***Feature Filter (FF)*** technique removes items from the list of reminder candidates that are not relevant anymore, i.e., belong to the *same category* as a recently purchased item. We therefore propose to maintain a *blacklist* of "outdated" categories based on the purchases made in the user's previous sessions. If a user however continues to view items from such a presumably outdated category, we remove the category from the *blacklist* again.

Generally, the *FF* technique can be used in combination with any of the previously discussed ranking strategies that provide a score_{ui}^{*}. If the category of an item i is on the blacklist, the filter sets its score to zero. In case the filter leads to too few remaining items, a popularity-based baseline technique can be used to fill up the recommendation list.

Note that in principle various combinations (hybrids) of the ranking strategies like *ISim* and *SSim* themselves are possible as well. In this work, we however focus on analyzing the effects of the individual ranking techniques in combination with the feature filtering method.

4. EXPERIMENTAL EVALUATION

The goal of our experimental analysis is to explore the effectiveness of different reminding strategies in terms of their capability to find relevant items. Other potential quality factors like diversity are not in the focus of our current work.

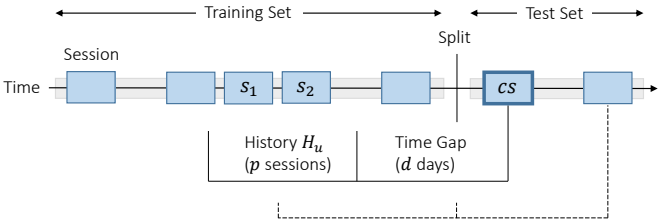


Figure 3: Evaluation scheme. The *training* and *test set* are determined per user. For each purchase that has to be predicted, p previous sessions before a time gap of d days represent the considered *history* H_u of a user, e.g., if *cs* is the *current session* for which purchases have to be predicted, the sessions $H_u = \{s_1, s_2\}$ are the user's relevant history before the time gap.

4.1 Evaluation protocol

An extension of the protocol proposed in [12] was used for the comparative evaluation of the proposed techniques and is shown in Figure 3.

Creating Training and Test Sets.

For each user the log entries are organized in sessions and split into a *training* set and hidden *test set*. In the experiments the splitting was done in a way that the test sessions comprise the last 20% of the purchases of a user. The recommendation task consists of predicting each purchase action in the sessions of the *test set* individually. To account for random effects, we use a 5-fold user-wise cross-validation by randomly assigning the users into five bins and executing the experiments five times without including the users of one bin each time.

Defining The Short-Term History.

Size of the history: Similar to [12] we assume that it is important to consider the most recent user behavior – the history H_u – to adapt the recommendations to the user's current shopping intents. The history H_u consists of p sessions that precede the currently evaluated test session plus those item view events of the currently evaluated test session that happened before the purchase that we want to predict. The choice of p depends on the domain. Too large values might lead to the inclusion of already outdated items; too small values might make the set of reminder candidates too small. We report the results for $p = 6$ in Section 5 as this value led to good results for all the used datasets.

Adding an "Obviousness Gap": As stated above, reminders should not be too obvious. To be able to vary and analyze the *degree of obviousness*, we introduce a time gap of d days between the user's history and the purchase that has to be predicted. Items that the user has interacted with in this time gap should not be used as reminder candidates. This also means that no actions from the current session are used as candidate items for reminders if $d > 0$. For example, for $d = 1$, items that were viewed within the last day of the current session would be considered too obvious and of too little value for the user as a reminder, see Figure 3. By varying d, we can therefore measure the performance of our reminding strategies when avoiding recommendations that are too obvious. Alternatively, the time gap could also be defined in terms of sessions instead of days.

Evaluation Measures.

As evaluation measures we use the *Hitrate@N* (Recall) and Mean Reciprocal Rank *MRR@N* for the computed *top-N* recommendation lists. Since we predict each purchase individually, there can only be one relevant item per recommendation list. Precision is therefore proportional to Recall.

4.2 Baselines Techniques

Although the focus of this work is on the adaptive reminding strategies, we are also interested in the effects of recommending known items using (a) standard CF techniques and (b) special context-aware schemes from the literature. We selected three implicit feedback techniques that have shown good predictive performance in the past.

(1) Bayesian Personalized Ranking (*BPR*): A learning-to-rank technique for implicit feedback [20]. To create a personalized ranking of all items for each user, *BPR* optimizes a generic optimization criterion that is the maximum posterior estimator for a Bayesian analysis of the ranking task. The model is trained with a bootstrapped, stochastic gradient descent method. *BPR* was the best-performing individual CF method in the analysis in [12]. In contrast to the usual setup, we allow *BPR* to include items in the recommendation lists that the user already knows as done in our A/B test on the China-Gadgets platform[5].

(2) Context-KNN (*C-KNN*): A k-nearest-neighbor recommendation technique using those sessions of all users in the *training set* that are most similar to the items in the history H_u^I of the current user u. This technique has been successfully used for contextualized playlist generation in [10] or [13]. Let V_s again be the set of items viewed in a session s. To calculate the similarity $\text{sim}_{binary}(u, s)$ between a user u and a session s from the *training set*, we use the cosine similarity $\text{sim}_{cos}(\vec{H_u^I}, \vec{V_s})$ between bit vectors that indicate for each item whether it is included in each set or not.

$$\text{sim}_{binary}(u, s) = \text{sim}_{cos}(\vec{H_u^I}, \vec{V_s}) \tag{7}$$

Let $\text{top}_k^T(u)$ be the k most similar sessions to u from all sessions in the *training set* T according to $\text{sim}_{binary}(u, s)$. The recommendation score for item i and user u is the sum of similarities for sessions in $\text{top}_k^T(u)$ where item i was viewed.

$$\text{score}_{ui}^{C\text{–}KNN} = \sum_{s \in \text{top}_k^T(u)} \text{sim}_{binary}(u, s) \cdot \mathbf{1}_{V_s}(i) \tag{8}$$

(3) Co-occurrence patterns (*CO*): A technique based on association rule mining used in [12] for e-commerce data and for the contextualized recommendation of workflow elements in [11]. This method learns size-two association rules of the form "Users who viewed A, also viewed B" from the training data. The recommendation score of an item i is the sum of the *confidence* values for the associations $j \rightarrow i$ by applying the rules to the items in the users current history H_u^I.

$$\text{score}_{ui}^{CO} = \sum_{j \in H_u^I} \text{confidence}(j \rightarrow i) \tag{9}$$

As we are interested in how many of the recommendations made by the CF baselines are actually known items, we furthermore report a metric *RepeatRate@N* for them which calculates the percentage of items in the *top-N* lists that the user has already viewed in a previous session. The *RepeatRate@N* for the proposed reminding strategies is of course 100%.

[5]We made additional experiments with other techniques including in particular Factorization Machines [19] or Funk's SVD [1]. As these methods perform worse than *BPR*, we do not report the results here.

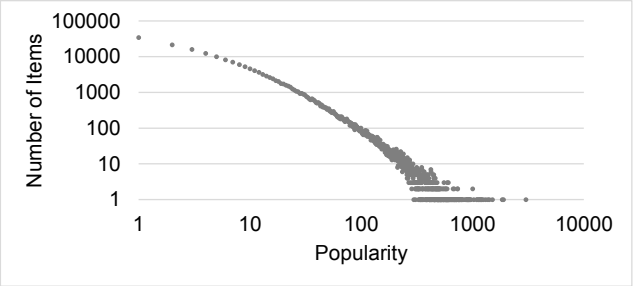


Figure 4: Popularity distribution for the Zalando *heavy* subset. X-axis: Item popularity (view and purchase actions per item). Y-axis: Number of items. Both in base-10 log scale.

4.3 Datasets

We use log data from Zalando, China-Gadgets and a public dataset from TMall [12]. For Zalando, we created subsets of *heavy* and *random* users. For the *heavy* users, we observe at least 80 site visits (sessions) during one year. The average number of sessions for *random* users is 28. For the smaller logs of China-Gadgets, we selected 10,000 users that are not one-time visitors. For TMall we selected 5,000 random users.

The Zalando and China-Gadgets log data files contain a few user events that can be attributed to the already deployed recommendation components on the sites. We removed those user actions from the logs that were induced by the recommenders, i.e., we removed item views that resulted from a click on recommendation lists. A small bias can still remain as users might have noticed the recommendations without having clicked on them. The detailed statistics of the datasets are shown in Table 1. In addition, Figure 4 shows the popularity distribution for the Zalando *heavy* subset. As is typical for RS datasets, the distribution is skewed with few highly popular and many unpopular items from the long-tail. The Zalando *random* and TMall datasets have similar characteristics, but the Zalando *heavy* subset contains more items with high popularity. The China-Gadgets subset has an overall smaller item catalog.

5. RESULTS

We first report the results of the CF baselines and then discuss the performance of the reminding strategies.

5.1 Effects of Recommending Known Items in Baseline Algorithms

In our first measurement, we compared the best-performing "standard" CF method from [12], *BPR*, with the context-aware baselines *C-KNN* and *CO*. We manually tuned the algorithm parameters for *BPR* and *C-KNN* ($k = 5$) to optimize the hit rate and tested two configurations: (a) without repeated recommendations, (b) with repeated recommendations[6]. Table 2 shows the results for the Zalando *heavy*

[6]Remember that we do not call these repeated recommendations "reminders" as the selection of items is not limited to items in the recent user history.

Table 1: Characteristics of the Zalando, China-Gadgets and TMall datasets.

	Zalando			China-Gadgets		TMall	
	complete	*heavy* subset	*random* subset	complete	subset	complete	subset
Users	3.5M	3k	3k	466k	10k	424k	5k
Items	460k	188k	121k	4.4k	3.7k	1M	195k
Views	200M	3.1M	906k	1.1M	192k	49M	660k
Purchases	3.9M	106k	47k	804k	142k	3.2M	50k
Sessions	3.5M	338k	89k	859k	123k	7.1M	96k

subset with the configuration parameters $d = 0$ and $p = 6$, i.e., the contextualized strategies use six previous sessions before the user's current session *cs* as the history H_u.

The results show that considering recent navigation actions from previous sessions – as done by the *C-KNN* and *CO* methods – leads to significant (Student's t-test, $p < 0.01$) accuracy improvements when compared to a "context-agnostic" learning-to-rank method like *BPR* as was discussed in [12][7]. Also, *BPR* tends to recommend many already known items and *C-KNN* is significantly more accurate than *CO*. Second, including repeated recommendations in all settings proved to be important and strongly increased the accuracy. Similar observations were also made for the other datasets and are omitted due to space constraints.

In the live experiment for China-Gadgets (Section 2.2), repeated recommendations were allowed for *BPR*. Preliminary online tests also indicated that the success rate of *BPR* is significantly lower when repeated recommendations are forbidden, which corroborates our offline results.

5.2 Results of the Reminding Strategies

In the next series of experiments, we benchmarked the simple "reverse history" strategy *IRec*, which was also used in [12], against the recominder strategies proposed in Section 3. The choice of p depends on the indented goal of the reminders and used techniques, e.g. increasing p can improve the accuracy of *ISim* due to more candidates but decrease the accuracy of *IInt* due to noise from older items. We take $p = 6$ sessions for the history H_u (from which the reminders will be selected) as an overall acceptable setting to show the general characteristics of all techniques. Fine-tuning of p is not in the scope of this work. To show the influence of reminding obvious items on the accuracy metrics, we also report the results for different time gap values $d = 0$ and $d = 3$ and a special case $d = 0^*$. In that last case we exclude the events from the user's current sessions in the history that were views of the item that should be predicted. This way, we can show the impact of not recommending the most obvious item. The results are given in Table 3 for the Zalando data and in Table 4 for the China-Gadgets and TMall data.

Including Obvious Reminders.

First, consider the situation $d = 0$, i.e., where users can be reminded of items they have just looked at. Both in the Zalando *heavy* and *random* user subset, the basic reminding strategy *IRec* works comparably well as users naturally often view items right before they purchase them. The hit rate is however not 1, since the purchased item does not necessarily need to be among the last 10 viewed items. Putting more weight on frequently viewed items (*IInt*) or reminding the user of similar items (*ISim*) leads to worse results.

In contrast, our content-based *SSim* strategy, which looks for similar shopping sessions in the past, works significantly ($p < 0.01$) better than the simple *IRec* strategy on the *heavy* user subset both in terms of the hit rate and the MRR. This indicates that it is not unusual that users abandon an unsuccessful shopping or browsing session, look for different items, and eventually continue the previous search to finally make a purchase. As a result, the *SSim* strategy shows that *adapting reminders* to the current user goals can be beneficial. However, for the *random* user dataset the *IRec* baseline cannot be beaten by *SSim*, since there are often not enough similar past sessions available for new users that can be used to predict the continuation of the current session.

If the time gap is zero ($d = 0$), actions from the current session are part of the history, including view actions for the item that was eventually purchased and that we want to predict. In the additional experiment labeled with $d = 0^*$ we did not include these actions in the history. As can be seen from Table 3, excluding this item results in a strong accuracy decrease, especially for the *IRec* strategy. Under this configuration, reminding frequently viewed items (*IInt*) performs best. Still, the hit rate of *IRec* is comparably high which indicates that users often browse items in previous sessions and sleep on their decisions before the purchase.

Avoiding the Obvious: Long-Term Reminders.

To avoid obvious recommendations, we do not remind users of what they have looked at during the last few days by increasing the time gap; here we report $d = 3$ days. Generally, when increasing d, the absolute values for the hit rate and the MRR decrease, which is however expected, given that some of these less obvious reminders may already be outdated. How to set d for "long-term reminders" depends on the application domain. In e-commerce, a few days might be appropriate [3]. In the domain of music streaming, it might in contrast be plausible to remind users of what they liked and listened to a few weeks or months ago.

The results in Table 3 show that the simple *IRec* strategy seems to include too many already outdated reminders under the $d = 3$ configuration. All other techniques proposed in our work mostly lead to better results than *IRec* in terms of the rate and MRR. By far the best results are obtained when using the content-based *ISim* method in combination with the feature filter (*FF*), i.e., when we remove reminders that relate to item categories in which the user has already purchased in the meantime. The *SSim* and *ISim* methods show a similar performance when the feature filter is not applied. The results however suggest that *SSim* includes too many item view events for products that the user eventually lost interest in and did not purchase in the end.

[7]As the recommendations are generated from a fairly large pool of items, the resulting values for the hit rate and the MRR are expected to be generally small [14].

Table 2: *Hitrate@10* (HR), *MRR@10* and *RepeatRate@10* (RR) results for the CF techniques for the Zalando *heavy* subset with/without the ability to recommend known items. History $p = 6$, time gap $d = 0$. The best values are highlighted in grey.

Dataset	Zalando – *heavy* subset								
Technique	*C-KNN*			*CO*			*BPR*		
Metric@10	HR	MRR	RR	HR	MRR	RR	HR	MRR	RR
No repeated recomm.	0.051	0.032	0	0.013	0.005	0	0.001	0.001	0
Repeated recomm.	0.156	0.072	0.375	0.123	0.046	0.343	0.066	0.024	0.830

Table 3: *Hitrate@10* (HR) and *MRR@10* results for the Zalando *heavy* subset and the Zalando *random* subset. History $p = 6$, time gap $d = 0$, $d = 0^*$, $d = 3$. The best values are highlighted in grey.

Dataset	Zalando – *heavy* subset						Zalando – *random* subset					
Configuration	$p = 6$, $d = 0$		$p = 6$, $d = 0^*$		$p = 6$, $d = 3$		$p = 6$, $d = 0$		$p = 6$, $d = 0^*$		$p = 6$, $d = 3$	
Metric@10	HR	MRR	HR	MRR	HR	MRR	HR	MRR	HR	MRR	HR	MRR
FF (*SSim*)	0.681	0.296	0.293	0.139	0.193	0.110	0.642	0.268	0.237	0.109	0.136	0.077
FF (*ISim*)	0.561	0.219	0.353	0.146	0.521	0.200	0.561	0.217	0.266	0.112	0.504	0.191
SSim ($k = 2$)	0.697	0.327	0.210	0.111	0.167	0.093	0.678	0.294	0.176	0.088	0.117	0.062
ISim ($k = 20$)	0.588	0.241	0.319	0.137	0.170	0.077	0.600	0.236	0.240	0.105	0.135	0.060
IInt	0.561	0.217	0.363	0.147	0.162	0.071	0.545	0.205	0.271	0.111	0.129	0.054
IRec	0.653	0.309	0.230	0.069	0.136	0.058	0.697	0.295	0.174	0.052	0.120	0.054

Results for the China-Gadgets and TMall Datasets.

Table 4 shows the results obtained with the same protocol configurations for the China-Gadgets and TMall datasets. The results are mostly in line with those from the previous experiment, which shows that the proposed reminder strategies may generalize to other datasets.

Specifically, when applying a time gap of $d = 3$, the feature filter method in combination with the content-based *ISim* method works consistently better than any other tested strategy. For the TMall subset with $d = 0$, the *SSim* method is again better than *IRec* in terms of hit rate and MRR, similar to Zalando *heavy*. However, this is not the case for China-Gadgets where – like Zalando *random* – there are not enough similar past sessions available for some users. A possible explanation that the *IRec* method works particularly well for the blog-style China-Gadgets platform can be that the entries on this site are time-ordered and the most recent additions appear on the top of the page. The order of the items may therefore often be directly related to their general attractiveness. Thus, the recommendations made by the *IRec* strategy can be used as navigation shortcuts for users to repeatedly inspect these most recent items.

Lastly, when removing the item that should be predicted from the history ($d = 0^*$), the *IInt* strategy works best for TMall, as was the case for the Zalando subsets. However, for China-Gadgets none of the reminding strategies stands out in particular regarding the hit rate and MRR.

Discussion.

A comparison of the absolute values obtained for the reminding strategies and the baseline techniques in Table 2 shows that reminders lead to substantially higher accuracy values. The results of our offline experiments therefore confirm the results obtained from our analyses of real-world systems in Section 2 and emphasize the potential value of including known items into recommendation lists. Furthermore, the experiments indicate that more elaborate reminding strategies can be more effective than a "reverse history" approach. To which extent combinations of these reminding strategies with long-term CF models can help to further increase the accuracy is part of our ongoing work.

Finally, all our reminding strategies needed less than 10 ms to compute one reminder list using a current generation Intel Xeon CPU. On the same hardware, the baseline techniques (*BPR*, *C-KNN*, *CO*) needed significantly more time (up to one second) to compute one list. Combining personalization methods with adaptive reminders will therefore not significantly increase the run-time complexity.

6. RELATED WORK

The systematic inclusion of reminders in recommendation lists has, to our knowledge, not been discussed or analyzed to a large extent in the RS research literature. Some exceptions exist, which, for example, mention the use of reminders in "check-out" situations in online shops ("Don't forget to buy") [18] or report of a mobile shopping assistant that displays similar known items while browsing for new products ("Similar to what you liked in the past") [17]. Both mentioned works however do not propose algorithmic solutions.

Recommendations for repeated consumption were investigated, e.g., in [3] and [15]. In contrast to repeated recommendations of already purchased or consumed items, our work focuses on reminders, i.e., items that the user knows, but that were not necessarily consumed so far. The analysis in [3] showed that the recency of interactions is the strongest predictor of repeated consumption and that there is an exponential decay factor. The authors propose a model to recommend known items with a probability that is proportional to the time of its last consumption. Such an exponential decay function can be integrated into our *IRec* reminding strategy, in particular when it should be combined with a CF model. In [15] the listening behavior of users when browsing for music tracks was investigated and the authors propose an approach to automatically derive whether the user is currently looking for something new or not. Their approach could be combined with our work to decide if reminders should be generally included in the current user session or not.

In the online shopping scenario, this question of "When to remind?" translates to determining whether the user currently is in a catalog exploration phase or in the phase of re-inspecting the choice set before making a decision. Based

Table 4: *Hitrate@10* (HR) and *MRR@10* results for the China-Gadgets and TMall subsets. History $p = 6$, time gap $d = 0$, $d = 0^*$, $d = 3$. The best values are again highlighted in grey.

Dataset	China-Gadgets – subset						TMall – subset					
Configuration	$p = 6$, $d = 0$		$p = 6$, $d = 0^*$		$p = 6$, $d = 3$		$p = 6$, $d = 0$		$p = 6$, $d = 0^*$		$p = 6$, $d = 3$	
Metric@10	HR	MRR	HR	MRR	HR	MRR	HR	MRR	HR	MRR	HR	MRR
FF (*SSim*)	0.577	0.344	0.219	0.128	0.206	0.121	0.482	0.220	0.228	0.101	0.142	0.078
FF (*ISim*)	0.532	0.276	0.172	0.116	0.481	0.225	0.415	0.185	0.239	0.096	0.405	0.176
SSim ($k = 2$)	0.616	0.376	0.223	0.136	0.212	0.121	0.483	0.214	0.173	0.082	0.135	0.067
ISim ($k = 20$)	0.541	0.254	0.223	0.134	0.189	0.109	0.415	0.185	0.195	0.082	0.140	0.067
IInt	0.539	0.254	0.223	0.134	0.188	0.106	0.429	0.181	0.241	0.094	0.138	0.063
IRec	0.615	0.586	0.220	0.126	0.186	0.099	0.463	0.191	0.125	0.032	0.117	0.049

on the context of the user, there might be situations where reminding is particularly useful as discussed in [18] and the problem of finding the best time for repeated recommendations has also been mentioned for commercial systems like eBay [23, 26] or Netflix [8]. In our work, we focus on the problem of selecting the right items and we consider the problem of determining whether we should present reminders at all as a related but complementary problem.

Our reminding strategies can also be seen as context-aware recommendation approaches, where the context to which the recommendations are adapted is the user's short-term shopping goal. In particular the *ISim* and *SSim* techniques try to rank those items from the user's navigation history higher that are presumably a good match for the items that the user has most recently inspected.

In practice, one would probably not implement a "pure" reminding strategy – which only includes items that the user already knows – but rather create one or multiple lists that contain a mix of items that match the long-term user model and a number of reminders. Some of these strategies that combine assumed short-term shopping intents with long-term collaborative filtering models were discussed in our previous work [12]. The results showed that the combined models which implemented a "contextual post-filtering" strategy [2] worked best in the tested configurations, but already the most simple reminders had a significant impact on the recommendation accuracy. Our current work continues this research and shows that there are more effective reminding strategies than the "reverse history" approach from [12].

Research on short-term adaptation for collaborative filtering approaches is comparably scarce in the recommender systems literature[8]. Recent exceptions include [10], [21], or [25]. These works propose different strategies of understanding the user's shopping intents, but do not explicitly include reminders in their recommendations. From an algorithmic perspective, our *Session Similarity* reminding method is related to nearest-neighbor approaches used in music playlist generation techniques as described in [9], where the goal is to find past listening sessions or playlists that are similar to the user's most recent listening behavior. Our *Feature Filter* method on the other hand implements a similar idea to the post-purchase product recommendation technique described in [26] where the goal was to avoid the recommendation of items on eBay that are too similar to recent purchases.

[8]In conversational and critiquing based systems [6] usually only short-term goals are relevant and the item filtering process is based on explicit user constraints. In our work we are however interested in learning-based, adaptive recommendation approaches.

The ranking of the reminders in our approaches is often based on a time-based criterion. Our work is therefore related to "time-aware recommender systems" as a special case of context-aware systems. Campos et al. review time-aware systems with a focus on evaluation aspects in [5]. According to their classification, our protocol represents a time-dependent cross-validation procedure with user resampling. In addition, our protocol implements a session-wise evaluation procedure as done in similar form in [4]. We however use the hit rate and the MRR as standard evaluation measures.

As discussed in the introduction, one possible value of automatically generated reminders as recommendations is that they can be used as navigation shortcuts by users. In [22], Schnabel et al. conduct a user study to assess the value of providing online customers with a web shop feature to create their own shortlists. Their study shows that many participants used these shortlists to organize their shopping sessions and remind themselves of recently inspected items. In addition, shortlists were shown to be helpful by reducing the cognitive effort for the users. An earlier study from the business literature [7] revealed that some online users – in absence of a shortlist functionality – use their shopping baskets to keep track of their candidate items, i.e., they misuse the cart as a shortlist with a reminder functionality. In contrast to these works, the discussed reminding strategies populate such shortlists automatically and in particular our results obtained from the logs of the deployed systems indicate that customers actually use these shortlists.

7. SUMMARY AND FUTURE WORKS

In this paper, we have analyzed the value of including reminders in recommendations, which is a common strategy in real-world systems, through offline and online experiments. We have furthermore proposed and successfully evaluated novel strategies which avoid the recommendation of unrelated or too obvious items. In practice, recommending only reminders might in many cases not be the most effective strategy. While reminders might increase familiarity and trust, they are by design unsuited to help users discover new items. Measuring the true value or obviousness of recommendations in offline settings is however challenging [18].

In our future work we will thus explore strategies that combine novel recommendations and reminders in a balanced way to increase diversity and plan to investigate techniques to better assess the *right time* for reminding, e.g., taking product consumption cycles into consideration. Furthermore, we will try to estimate their perceived value through user studies and measure their effect in real world scenarios.

8. REFERENCES

[1] http://sifter.org/~simon/journal/20061211.html (last accessed: 02/2016), 2006.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, pages 217–253. Springer, 2011.

[3] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. The dynamics of repeat consumption. In *Prof. WWW '14*, pages 419–430, 2014.

[4] David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. RecSys challenge 2015 and the YOOCHOOSE dataset. In *Proc. RecSys '15*, pages 357–358, 2015.

[5] Pedro G. Campos, Fernando Díez, and Iván Cantador. Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols. *UMUAI*, 24(1-2):67–119, 2014.

[6] Li Chen and Pearl Pu. Critiquing-based recommenders: Survey and emerging trends. *UMUAI*, 22(1):125–150, 2011.

[7] Angeline G. Close and Monika Kukar-Kinney. Beyond buying: Motivations behind consumers' online shopping cart use. *Journal of Business Research: Advances in Internet Consumer Behavior & Marketing Strategy*, 63(9–10):986–992, 2010.

[8] Carlos A. Gomez-Uribe and Neil Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM TMIS*, 6(4):13, 2015.

[9] Negar Hariri, Bamshad Mobasher, and Robin Burke. Context-aware music recommendation based on latenttopic sequential patterns. In *Proc. RecSys '12*, pages 131–138, 2012.

[10] Negar Hariri, Bamshad Mobasher, and Robin Burke. Adapting to user preference changes in interactive recommendation. In *Proc. IJCAI '15*, pages 4268–4274, 2015.

[11] Dietmar Jannach, Michael Jugovac, and Lukas Lerche. Adaptive recommendation-based modeling support for data analysis workflows. In *Proc. IUI '15*, pages 252–262, 2015.

[12] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. Adaptation and evaluation of recommendations for short-term shopping goals. In *Proc. RecSys '15*, pages 211–218, 2015.

[13] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosch. Beyond "hitting the hits" – generating coherent music playlist continuations with the right tracks. In *Proc. RecSys '15*, pages 187–194, 2015.

[14] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. What recommenders recommend: An analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491, 2015.

[15] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A. Konstan, and Paul Schrater. "I like to explore sometimes": Adapting to dynamic user novelty preferences. In *Proc. RecSys '15*, pages 19–26, 2015.

[16] Elizabeth Hellmuth Margulis. *On Repeat: How Music Plays the Mind*. Oxford University Press, 2014.

[17] Carolin Plate, Nathalie Basselin, Alexander Kröner, Michael Schneider, Stephan Baldes, Vania Dimitrova, and Anthony Jameson. Recomindation: New functions for augmented memories. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 141–150, 2006.

[18] George Prassas, Katherine C. Pramataris, Olga Papaemmanouil, and Georgios J. Doukidis. A recommender system for online shopping based on past customer behaviour. In *Proc. 14th Bled eConference*, pages 766–782, 2001.

[19] Steffen Rendle. Factorization machines with libFM. *ACM TIST*, 3(3):57:1–57:22, 2012.

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. UAI '09*, pages 452–461, 2009.

[21] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proc. WWW '10*, pages 811–820, 2010.

[22] Tobias Schnabel, Paul N. Bennett, Susan T. Dumais, and Thorsten Joachims. Using shortlists to support decision making and improve recommender system performance. *CoRR*, abs/1510.07545, 2015.

[23] Neel Sundaresan. Recommender systems at the long tail. In *Proc. RecSys '11*, pages 1–6, 2011.

[24] Kirsten Swearingen and Rashmi Sinha. Interaction design for recommender systems. In *Proc. DIS' 02*, pages 312–334, 2002.

[25] Maryam Tavakol and Ulf Brefeld. Factored MDPs for detecting topics of user sessions. In *Proc. RecSys '14*, pages 33–40, 2014.

[26] Jian Wang, Badrul Sarwar, and Neel Sundaresan. Utilizing related products for post-purchase recommendation in e-commerce. In *Proc. RecSys '11*, pages 329–332, 2011.