Towards Simulation-Based Evaluation of Recommender Systems with Carousel Interfaces

BEHNAM RAHDARI, University of Pittsburgh, USA
PETER BRUSILOVSKY, University of Pittsburgh, USA
BRANISLAV KVETON, Amazon, USA

Offline data-driven evaluation is considered a low-cost and more accessible alternative to the online empirical method of assessing the quality of recommender systems. Despite their popularity and effectiveness, most data-driven approaches are unsuitable for evaluating interactive recommender systems. In this paper, we attempt to address this issue by simulating the user interactions with the system as a part of the evaluation process. Particularly, we demonstrate that simulated users find their desired item more efficiently when recommendations are presented as a list of carousels compared to a simple ranked list.

CCS Concepts: • **Information systems** → **Recommender systems**; • **Human-centered computing** → **Interactive systems and tools**.

Additional Key Words and Phrases: Carousel-based interface; Human-AI collaboration, Navigability, Click Models

ACM Reference Format:
Behnam Rahdari, Peter Brusilovsky, and Branislav Kveton. 2024. Towards Simulation-Based Evaluation of Recommender Systems with Carousel Interfaces. In *Proceedings of ACM Transactions on Recommender Systems (Transactions on Recommender Systems).* ACM, New York, NY, USA, 25 pages.

1 INTRODUCTION

For many years, user studies have been the key approach to evaluating all types of user-adaptive systems, i.e., interactive systems that can adapt their behavior to individual users [11]. While user studies are the ultimate way to assess user-centered systems, these studies are very expensive. It is also a challenge to obtain user study data on a sufficient scale to reliably compare specific user modeling and personalization approaches. In response to these challenges, several research areas focused on user-adaptive and personalized systems established *data-driven* approaches for evaluating systems in these areas. For example, data-driven evaluation of learner modeling in personalized education systems is based on large collections of student problem-solving traces. The ability to better predict a learner's success in these traces is considered a sign of better quality learner modeling [17, 53]. Similarly, data-driven evaluation of recommender systems is based on the large volume of user past rating data. The ability to better approximate user ratings or place items positively rated higher on the ranked list is considered a sign of better quality recommendation [28, 43].

The establishment of data-driven evaluation approaches was very important for the field of recommender systems. Promoted by the Netflix Prize, these approaches helped to engage a large number of researchers in the work on recommender systems and stimulated rapid progress in the development and evaluation of recommendation algorithms. Data-driven evaluation quickly became the gold standard in the field, overshadowing empirical evaluation approaches. Numerous

Transactions on Recommender Systems,

articles discussed the comparative benefits of data-driven versus empirical evaluation and pointed out that these studies frequently deliver different results [28, 55, 63]. Proponents of data-driven evaluation stressed the opportunity to obtain large-scale data and to evaluate new ideas relatively quickly, especially given the increasing number of available datasets. Proponents of user studies stress that the end user is the ultimate judge and that evaluating many "beyond precision" aspects of recommendation is not possible without engaging users. It is currently accepted that data-driven evaluation is not a replacement for empirical evaluation, rather the two approaches, often referred to as offline and online evaluation, are complementary and together could offer a more complete picture in assessing and comparing recommender systems [28, 60]. In other words, it is important to have a choice between data-driven evaluation and user studies when evaluating recommender systems.

However, the choice between data-driven and empirical evaluation approaches is currently not available to researchers working on various *interactive recommender systems* [34], which present recommendation results in a more complex way than a ranked list and engage users in different forms of interaction with results. The key problem here is that the user behavior in these systems is more complex than in recommender systems that present recommendations as a ranked list. The user interaction with traditional ranked lists has been explored in a number of studies [24, 26], which revealed and measured the user's tendency to examine the list from the top down and favor the top-ranked items. These studies helped create commonly accepted metrics for offline evaluation, such as nDCG [39] or MRR [14]. However, traditional metrics do not apply to recommender systems with more complex interfaces. These systems might have multiple ranked lists or no ranked list at all, and their effectiveness is defined by the whole user interaction rather than a single-shot presentation of recommendation results. Does this mean that offline data-driven evaluation is not an option for this increasingly popular group of recommender systems? This paper argues that data-driven evaluation of recommender systems with advanced interfaces could be performed using a *simulation-based approach*. The idea of this approach is to perform a continuous simulation of user behavior in a target system while computing various performance metrics "on the go". The simulation-based approach could be applied to relatively complex interaction scenarios, as long as user behavior in these scenarios is sufficiently well understood and modeled. Although this approach enables the application of performance metrics that are typically used in empirical studies, it is based on simulated users rather than real users and can be performed offline. The simulation-based approach has been used to evaluate various types of interactive systems [8, 56], however, its application to the evaluation of recommender systems is still an exception [20, 32]. To make a case for simulation-based evaluation of recommender systems with more complex interfaces and to demonstrate the power of this approach, we apply it to comparative evaluations of *carousel-based recommender interfaces*. These increasingly popular interfaces (known also as *multilists*) organize the recommendations results in a set of topic-based "carousels" that allow users to choose their current topic of interest rather than leaving the recommender system to guess. Evaluation of carousel-based interfaces through offline studies is a known challenge [21]. In our paper, we demonstrate that this challenge could be addressed through simulation-based studies based on *models of user behavior* in carousel-based interfaces. To stress that user behavior could be modeled at different levels of complexity and to show the connection between the complexity of the model and the complexity of research questions that could be answered by performing a simulation-based study with this model, we demonstrate the application of the simulation-based evaluation approach through two case studies. The first case study demonstrates the use of a relatively simple *qualitative model* of user behavior to compare the navigability of a carousel-based interface and a ranked list. The second case study demonstrates how a more complex *quantitative model* of user behavior supports more elaborated simulation-based studies that could answer a wider range of research questions. To perform the latter study, we developed and validated *a carousel click model*, which generalizes the traditional *cascade models* [15] that

model user behavior in a ranked list. Using these click models, we demonstrate how simulation-based evaluation could be used to compare several ranking approaches within the carousel-based interface, as well as to compare the performance of a carousel-based interface and a traditional ranked list.

The paper is structured as follows. In Section 2, we review previous research on simulation-based evaluation, interactive recommender systems, and user behavior models. In Section 3 we present our first case study in which a simulation-based evaluation is performed using a simple qualitative model. Section 3 presents key components of our second case study, including the carousel click model (Section 4.2), its validation (Section 4.3 and Section 4.4), and two examples of simulation-based evaluation enabled by this model (Section 4.5 and Section 4.6). Finally, we discuss our results in Section 5.

2 RELATED WORK

In this section, we will review the related work in three areas of recommender systems: interactive recommender systems, simulation-based studies of recommender systems, and click models in information retrieval and recommender systems. Our goal is to provide a comprehensive overview of relevant research in these areas, highlighting key findings, approaches, and challenges.

2.1 Interactive Recommender Systems

For many years, a ranked list of items was a de facto standard for recommender systems to present recommendations and results to their users. With this approach, the power of a recommender system is fully based on the power of its Algorithm leaving the users almost no opportunities to affect the behavior of the system. However, an alternative stream of research on more complex recommender interfaces that enable humans and AI to work together to discover the most relevant items has been present in the field since its early days [6]. Now, when the need for human-AI collaboration is broadly accepted in a range of AI-based systems, research on *interactive recommender systems*, as this group or systems is often called [34], is rapidly expanding. Among the most popular (and not fully disjoint) groups of interactive recommender systems are critiquing-based systems [10], conversational recommenders [51], user-controlled recommenders [38], and visual recommenders that present recommendations results in more than one dimension, such as a grid [71] or a more complex visualization [45, 69]. Over the last 20 years, these groups of recommender systems have been explored and their effectiveness has been convincingly demonstrated [4, 54, 64, 70].

Today, the most noticeable group of interactive recommender interfaces is arguably carousel-based interfaces or multilists [3, 21–23, 48, 59]. While the interface with multiple carousels looks relatively complex – it presents several ranked lists, each marked with a category, in place of a single ranked list – it was embraced by the end-users and is now rapidly replacing the ranked list as a de facto standard to present recommendations in e-Commerce systems. From the prospect of recommender systems, the carousel-based interface provides an excellent example of human-AI collaboration in the recommendation context. While a single ranked list attempts to be "perfect", in reality, the intent of the user is often uncertain. Most importantly, in many real-life applications users might have multiple interests, and recommender systems rarely know which specific interest (for example, a movie genre) the users want to pursue at the given moment. A carousel-based interface leaves the task of choosing the most timely topic of interest (i.e. British documentaries) to the users. As a result, a user could quickly locate a ranked sub-list of the most relevant items while also indirectly informing the recommender system about the kind of items they prefer right now.

The popularity of carousel-based interfaces has not been ignored by researchers in recommender systems. A growing number of papers focused on carousel-based interfaces have been published in recent years [2, 21, 25, 37, 65, 71]. However, the evaluation of carousel-based interfaces and, more specifically, their comparison with other kinds of recommenders is still a bottleneck, since

it is typically based on expensive user studies. Our work attempts to bridge this gap. In this paper, we choose carousel-based recommender interfaces for the two case studies, which we present to demonstrate the application of simulation-based approach for the evaluation and comparison of recommender systems. To enable simulation-based evaluation of carousel-based interfaces, we also developed a novel carousel click model that can simulate how users interact with topic-labeled carousel interfaces. We hope that our new click model and the presented example of its use for simulation-based evaluation of carousel-based interfaces will facilitate future research in this area.

2.2 Simulation-Based Studies of Recommender Systems

A simulation-based approach has been used for exploration and evaluation in a number of fields where sufficiently detailed models of user behavior could be built. In particular, a simulation-based study is a recognized approach for evaluating various types of personalized interactive systems, from adaptive learning systems [8, 19] to personalized information access systems [50, 56]. The goals of simulation-based evaluation differ between application areas and often depend on the reliability of behavior models that support simulation. On one end of the spectrum are cognitively grounded behavior models that are supported by studies of human cognition and confirmed by empirical studies. A well-known example is SNIF-ACT model [56] that simulates user behavior in hypertext navigation. This model is based on Information Scent theory [7] and was used to assess the quality and navigability of Web sites without real users. Popular "simulated student" models [8, 49] used for evaluation of adaptive educational systems also belong to this group. On the other end, there are a range of simple behavior models [20, 72] that might not be able to reliably predict the details of user behavior but could be useful to explore a range of "what if" scenarios in assessing the impact of various interface enhancements.

Early attempts to use simulations to explore information filtering and recommender systems were made in the first decade of 2000 [19, 50]. However, it took another 10 years for this approach to become truly noticed in this field [20, 33]. Although the volume of simulation-based research in the recommender system context is gradually increasing, simulations are most frequently used to explore the impact of recommender systems on various aspects of user behavior rather than to assess their performance and effectiveness in a comparative way. The most popular research direction enabled by simulation is to examine the impact of a recommender system on various aspects of user behavior *as a whole* [5, 19, 32, 72]. This work is typically enabled by the user choice models [33]. While research on click models reviewed in Section 2.2 offers solid ground for simulation-based studies, there are very few cases where models of user click behavior were used for comparative offline evaluation of recommender system design options. A notable exception is the work of Dzyabura and Tuzhilin [20] who used simulation to compare an interface based on a combination of search and recommendation to interfaces based on search or recommendation alone. However, this work used a relatively simple behavior model that was not based on empirical observations or theory. In our work, we perform simulation-based studies using more complex and empirically grounded models, which increase our chances of obtaining useful and reliable results.

Several other works used simulation-based evaluation for different purposes and in different contexts. Zhang and Balog [73] use simulations to evaluate the conversational recommender system. They take into account both individual preferences and the general flow of engagement with the system, to build a simulator, which produces replies that a real human would provide. Rohde et al. [62] utilize OpenAI Gym, a popular framework for simulating Reinforcement Learning agents, and create a recommendation environment that is based on a model of user visiting patterns on e-commerce sites and how people react to recommendations. In a different stream of work, Zou et al. [75] develops a customer simulator known as the World Model, which is made to imitate the environment and address the selection bias of logged data. Finally, Ie et al. [36] develops a

programmable authoring tool for simulation environments for recommender systems called RecSim, which supports sequential user interaction.

2.3 Click Models in Information Retrieval and Recommender Systems

The research on *click models* focuses on modeling and explaining user interaction with a ranked list of search or recommendation results. It started in the field of information retrieval and was originally motivated by the need to improve the performance of the Web search engine by applying user *click-through data* accumulated by search engine logs [66, 74]. While "old school" information retrieval considered item relevance as the only factor determining user decision to click on a specific result, it became evident that the position of items in a ranked list has to be considered as well [40]. Moreover, creative experiments demonstrated that a high item position in a ranked list could have a greater impact on click probability than item relevance [41]. A sequence of eye-tracking studies with users of search engines [24, 26, 27, 52] helped understand how users process a ranked list of results and measure the impact of item position in the list on the click probability.

This research provided a solid ground for developing click models for user interaction with ranked lists [12], which is now actively used in both information retrieval and recommender systems research. There are many click models [1, 9, 12, 16, 29, 30, 61]. Essentially, all of them try to explain the user behavior by a generative model which can be learned from data. As an example, the cascade model [16, 61] assumes that the user examines the list of recommended items from top to bottom until they find an attractive item. After that, they click on that item and leave satisfied. This seemingly simple model explains the observed position bias in recommender systems that lower-ranked items are less likely to be clicked than higher-ranked items. Click models can be used to debias click data, and in turn to learn better ranking policies either offline [12, 47] or online [13, 46]. In this work, we simulate click models to compare the utility of recommendations in carousels with more traditional approaches.

3 CASE STUDY 1: EXPLORING THE NAVIGABILITY OF CAROUSEL-BASED INTERFACES WITH SIMPLE INTERACTION MODELS

In this section, we demonstrate the importance of using user behavior models in simulation-based studies to gain valuable insights into the behavior of users in interactive recommendation settings. The study compares user interactions with two types of recommendation interfaces - a carousel-based multi-list and a traditional ranked list interface - from the perspective of navigability, which refers to the ease and efficiency with which users can explore and access information.

In this case study, we demonstrate that relatively simple user behavior models could empower simulation-based studies that could produce important and interesting results. Even with relatively simple models, we gain valuable insight into the behavior of users in interactive recommendation settings and identify important factors that influence their dynamics. As the complexity of these systems continues to increase, the use of user behavior models is likely to become an increasingly important tool for researchers and practitioners alike. In this study, we introduce a basic model of user interaction with a carousel-based recommender interface and use it along with a traditional ranked list interaction model (Section 2.3) to compare user work with two types of recommendation interfaces: a carousel-based multi-list and a traditional ranked-list interface. The comparison is performed from the prospect of *navigability* [18], a popular research stream in the broader area of information access, which explores navigation properties or various information access artifacts and compares different approaches to create these artifacts. In this context, navigability refers to the ease and efficiency with which users can explore and access information. By comparing different approaches to creating navigable information access artifacts, researchers aim to identify the most effective strategies for supporting user navigation and improving overall user experience. The choice of navigability was

important to introduce the idea of simulation-based studies to the hypertext research community, where an earlier version of this evaluation was presented [59]. However, navigability is relevant today in a broader set of information-rich environments, where users face an increasing need to efficiently locate and access relevant information amidst a sea of available options. For example, navigability studies have been performed in the past to compare the navigability of different approaches to generate tag clouds [68], to examine the effect of automatic linking on navigability [35], and to compare the navigability of regular and faceted tag clouds [67].

We perform this comparison in a typical modern recommendation context where items could be associated with multiple *interests* and users could favor several of these interests in parallel (although probably to a different extent and at a different time). Depending on the domain, these interests could have different semantic natures. For example, it could be a movie genre such as *action movies*, a topic of interest such as *context-aware recommendation*, or even a specific approach to select items such as *most popular* or *recently added*. In all these cases, each carousel represents a specific dimension of user interests. For uniformity, we refer to these generalized interests as *topics*. Note that some recommender systems could model interests as latent categories rather than explicitly presented, understandable topics. In this paper, we focus on domains with explicitly represented interests to separate the problem of latent interest discovery from the problems of user modeling and item ranking. As our data show, even relatively simple models of user behavior could clearly reveal the benefits of carousel-based interfaces in this modern multi-interest context, explaining the rapidly increased popularity of these interfaces.

This presentation of the first case study is structured as follows. We start by introducing basic user behavior models for carousel-based interfaces and 2D ranked lists in Section 3.1. In Section 3.2, we detail our experimental setup. In Section 3.3, we introduce different settings under which we perform comparisons of user navigation in carousel-based interfaces and 2D ranked lists. Finally, we present and discuss our results in Section 3.4.

3.1 A Basic Interaction Model for Carousels and 2D Ranked Lists

To quantify the benefit of carousels, we formalize the problem of carousel recommendation using a simple mathematical model, which we call a *carousel interaction model*. We have a matrix of $m \times K$ recommended items, where m is the number of rows (carousels) and K is the number of columns (items per carousel). Each carousel is associated with some topic, such as a movie genre. To simplify the exposition, we assume that each item belongs to a single topic. We refer to the item at row $i \in [m]$ and column $j \in [K]$ as (i, j).

The user preferences are defined by two sets of probabilities. The first are *topic preferences*. Specifically, $p_i \geq 0$ is the probability that the user is interested in topic i, for any $i \in [m]$. The second set are *topic-conditioned item preferences*. Specifically, $p_{j|i} \geq 0$ is the conditional probability that the user is interested in item j given that they desire topic i, for any $i \in [m]$ and $j \in [K]$. We assume that $\sum_{i=1}^{m} p_i = 1$, and that $\sum_{j=1}^{K} p_{j|i} = 1$ for any topic $i \in [m]$.

The user interacts in the carousel model as follows. First, the desired topic and the item in that topic are realized in the mind of the user, and then the user seeks them. In particular, the *desired topic* is sampled as $I \sim \mathrm{Cat}((p_i)_{i=1}^{m})$ and the *desired item* is sampled as $J \sim \mathrm{Cat}((p_{j|I})_{j=1}^{K})$, where $\mathrm{Cat}(\theta)$ is a categorical distribution with outcome probabilities θ. In plain English, exactly one topic is chosen with probability p_i, and exactly one item is chosen with probability $p_{j|I}$ conditioned on that topic. An equivalent way to think about this process is that exactly one (i, j) is chosen with probability $p_{i,j} = p_{j|i} p_i$. The user seeks the item (I, J) as follows. They start by examining the first carousel. If its topic does not match that of I, they proceed to the next carousel. The user examines

all carousels, from top to bottom, until they stop at carousel I. After that, the user examines the items in carousel I, from left to right, until they find the desired item in column J.

To make the comparison of a two-dimensional multi-list and a one-dimensional ranked list more clear and fair, we represent a traditional single ranked list in a comparable 2D format as the matrix of $m \times K$ recommended items introduced above, which is examined row by row. This approach to presenting a ranked list is becoming popular in modern recommenders due to its space-saving format [58, 71]. Applying traditional models of user work with a ranked list reviewed in Section 2.3 to this 2D presentation format, we obtain the following simple model of user behavior in a 2D ranked list. This user behavior model used in the study is a variant of the cascade model [15]. The user starts at position $(1, 1)$. If that item is not desired, the user proceeds to the next item $(1, 2)$. The user examines the row 1, from left to right, until the desired item is found or the end of the row is reached. If the end of the row is reached, the user moves to item $(2, 1)$, the first item in the next row. The user then examines this row, from left to right, and this process continues until the desired item is found.

In case study 1, we do not perform a separate validation of the suggested user behavior models against historical data, since these models are relatively simple and intuitive extensions of previous empirical research [15, 61]. However, as shown later in section 4.3 these models fit well with data obtained in recent studies of carousel-based interfaces [37]. Section 4.3 also shows an example of fitting more complex behavior models to real-world user data.

3.2 Experimental Setup: Navigability Simulation

We conduct a series of data-driven experiments to evaluate how our proposed *carousels* model performs against a standard baseline (*single ranked list*) model from the prospect of navigability. For our experiments, we choose the domain of movie recommendation. The choice of domain was motivated by two reasons. First, movie recommendation is a good example of a modern context where users can have multiple interests and favor different interests at different times. Second, it is the context where carousels are currently very popular, which makes it easier to simulate realistic carousel-based recommendations.

We use the MovieLens 1M Dataset [31] which consists of 1 million ratings applied to 4000 movies in 18 genres by 6000 users. In our experiments, we only utilize information about user ratings and movie genres. We apply a pre-processing step to remove movies with no genres or no ratings.

We assume that the user adopts two distinct browsing behaviors when searching for a movie (I, J), provided that the results are presented as a single ranked list or a set of carousels. To generate the recommendations, we consider two sets of probabilities. The *topic preferences* and the *topic-conditioned item preferences*. The preferences are computed as follows. The dataset of ratings is a set of tuples $\mathcal{D} = \{(U_t, j_t, r_t)\}_{t=1}^{K}$, where U_t is the index of the user in data point t, j_t is the index of the rated movie in data point t, and r_t is the corresponding rating. The topic-conditioned item preference reflects how representative the movie is of a genre. We computed it as the sum of all the ratings of the movie over the sum of all the ratings in its genre. Formally, let $\mathcal{G}_i$ be the set of all movies of genre i. Then for any movie $j \in \mathcal{G}_i$, the topic-conditioned item preference of movie j in genre i is

$$p_{j|i} = \frac{\sum_{t=1}^{n} \mathbb{1}\{j_t = j\}\, r_t}{\sum_{t=1}^{n} \mathbb{1}\{j_t \in \mathcal{G}_i\}\, r_t}\,.$$

We set $p_{j|i} = 0$ for any $j \notin \mathcal{G}_i$. For any user u, the topic preference reflects how much the user prefers a genre. We compute it as the sum of all ratings of a user in a given genre over all ratings by that user.

Formally, the topic preference of user u for genre i is

$$p_i = \frac{\sum_{t=1}^{n} \mathbb{1}\{U_t = u, j_t \in \mathcal{G}_i\}\, r_t}{\sum_{t=1}^{n} \mathbb{1}\{U_t = u\}\, r_t}\,.$$

Having a *User Profile* assigned to each user, we generate two sets of recommendations as follows: For the first set of recommendations for carousels, we use the *topic preferences* to sort them and then populate each one with movies using the topic-conditioned item preferences. This approach generates a set of carousels each representing a genre (18 carousels for 18 genres in the dataset). Each carousel contains all the movies within the representative genre. With an average of more than 335 movies in each genre, we assume that it is realistic for the user to scroll down or right, examine all items and find the desirable movie. Movies are sorted by their scores, where the movie score j is $\sum_{i=1}^{m} p_{i,j}$. Given the large number of movies in the dataset, we assume that users will be able to navigate through the list by scrolling down to find what they are looking for. This assumption is based on the expectation that users are familiar with browsing behavior and comfortable scrolling through long lists to find relevant items. In this evaluation, *User Profile* and the recommendations were not affected by further user interactions and remained unchanged throughout all sessions.

We define a session as a single instance of evaluation in which the user seeks a movie (I, J) from the set of recommended results, which can be displayed as a *single ranked list* or *carousels*. The process of simulation is as follows: For each setting, we first generate two sets of recommendations (one using *single ranked list* and another using *carousels*) for every user in the dataset. Next we run 100 independent sessions for every user. Each session includes selecting a genre, selecting a movie within that genre, and calculating the number of interactions required to reach that movie in both models. We consider the average value of these 100 sessions as the outcome of the experiment for the given user in the given setting.

To simulate user navigation in each session, we assume that the desired genre and a movie of that genre are "realized" in the mind of the user. The *desired genre* is sampled as $I \sim \mathrm{Cat}((p_i)_{i=1}^{m})$ and the *desired movie* is sampled as $J \sim \mathrm{Cat}((p_{j|I})_{j=1}^{k})$. In each session, the user is only interested in a single genre and a single movie within that genre.

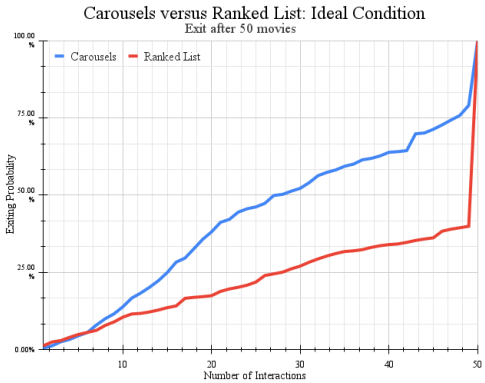
There are many ways to measure the complexity of the interaction with the recommended items in *single ranked list* and *carousels*. In this use case, We employ a custom metrics to evaluate our proposed approach. We define the *exiting probability* which determines on average what proportion of users left the session after a certain number of interactions.

3.3 Experimental Conditions

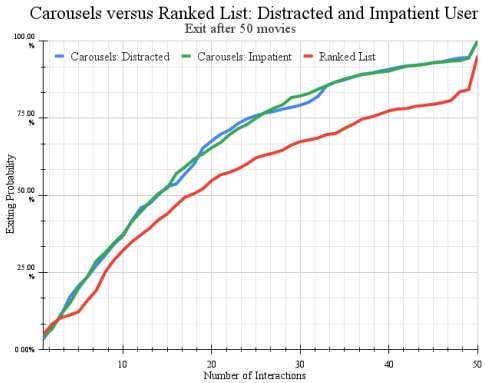
We compare carousel-based and 2D ranked-list interfaces in three increasingly more realistic settings reviewed below.

3.3.1 Ideal Setting. In the first setting, we assumed that the user continues to examine topics and items until the desired item was found. The behavior of such a user is described in Section 3.1. We are aware that this browsing behavior is unlikely to occur in a realistic situation due to the position bias effect [15]. However, we include this setting in our evaluation to highlight the difference between this and other more realistic behavioral patterns.

3.3.2 Impatient User. To better model the browsing behavior of a real user, we assume that the user has limited patience to find the desired item. We implemented this behavior as follows. The user starts by examining the first topic or item at position $(1, 1)$. The user exits with a probability of $p_q = 0.02$ after examining a carousel or item. This means that users abandon the session after 50 interactions on average, when no items or topics are desirable. This is the same as the ideal setting except for exiting with probability $p_q = 0.02$ upon each examination of a carousel or an item.



(a) Cumulative exiting probability under *ideal* condition.



(b) Cumulative exiting probability for *Distracted* and *Impatient* user.

Fig. 1. Comparing the cumulative *exiting probability* in *single ranked list* and *carousels* and in different experimental settings reveals the advantages of using a carousel-based representation compared to a ranked list-based representation. In all experimental settings users leave after 50 interactions regardless of success in finding the desirable item.

3.3.3 Distracted User. We initially assumed that the user always knew which carousel (with a genre as a topic) includes the desired movie. However, in reality, the user might get distracted and, as a result, begin browsing the wrong carousel or pass the correct carousel and miss out on finding the desired item. We consider this assumption to be an extension of the previous assumption described in Section 3.3.2.

In both ideal settings, when the user examines an undesirable carousel, they will move to the next carousel with probability 1. We define $p_d = 0.05$ as the *distraction probability*. Here user moves to the next carousel with probability $1 - p_d$ and starts examining items in the undesirable carousel with probability p_d. Similarly, when the user examines a desirable carousel, they move to the next carousel with probability p_d and begin examining items in the desirable carousel with probability $1 - p_d$. Considering a user *Distracted* only applies to *carousels*. Including this assumption in *carousels* allows us to capture the complexity that comes with providing additional information to the user in the form of carousel topics. Due to the lack of a large enough data set to accurately estimate the parameters of our proposed settings, we set the values of p_q and p_d intuitively based on how we presume the user would behave under those settings.

3.4 Results

To compare the behavior of our model in more realistic settings, we visualize the average *exiting probability* of users after a certain number of interactions with the recommendations in Figures 1a and 1b.

In Figure 1a we observe a significant difference (independent t-test, $p < 0.001$) between the *carousels* and the *single ranked list* under the ideal settings. In ideal settings, the user continues the examination until he reaches the desirable item. We limit the number of interactions to 50, meaning the user would exit unsatisfied if they could not find the desirable item in first 50 interactions. The higher *exiting probability* in *carousels* (blue line) shows that more users exit the system satisfied by finding their desirable item. A larger spike in *exiting probability* on *single ranked list* at the end indicates a higher number of users who left without finding their desired item. It should be noted that

based on the result of this experiment, a significantly larger portion of users (just under 80%) exit the system after finding their desired item. This number drops to close to 40% when recommendations are presented in the form of a ranked list.

The exiting behavior of the simulated *Impatient* and *Distracted* users is shown in Figure 1b. It is worth noting that in the ideal setting, the exiting probability can be considered a positive metric when the user finds the desired item after examination, but in distracted and impatient settings, the exiting probability could be an indication of either satisfactory or unsatisfactory results. In our experiments, we only compare the exiting probability under comparable settings. Although the gap between the probability of exiting the session in *carousels* and *single ranked list* models is less prominent, the former still performs better. The results of an independent t-test did not show a statistically significant difference between the models. Comparing the *Impatient* and *Distracted* exiting behavior indicates an insignificant difference between the two settings but shows a slight decrease in performance in *carousels*. Unlike in Figure 1a, where the *exiting probability* promotes a positive event (satisfaction of finding the desirable item), in Figure 1b there can be also adverse reasons for exiting a session, such as "impatience" and "distraction". Therefore, the improvement in this metric compared to the ideal setting is not necessarily a positive sign. Despite this, since we compare *carousels* and *single ranked list* in Figure 1b under the same setting where the probabilities of "distraction" and "impatience" are similar, an improvement in the metric likely signals a positive event.

4 CASE STUDY 2: EXPLORING CAROUSEL-BASED INTERFACES WITH ADVANCED CLICK MODELS OF USER BEHAVIOR

Our second case study demonstrates the application of click models – more advanced and precise models of user behavior – to simulation-based studies of carousel-based recommender systems. Extending our earlier work [57], the second study advances our first case study in three important directions. First, expanding the work on traditional click models, we develop a novel *Carousel Click Model* (CCM) that enables more advanced simulation-based studies of carousel interfaces. Second, we demonstrate how click models of user behavior could be validated using both existing empirical data and simulations. Third, we present several examples of simulation-based studies enabled by CCM. In particular, we demonstrate how the application of a more advanced behavior model could expand the range of research questions to be answered by a simulation-based study and enable the application of more precise (and traditional) evaluation metrics such as click probability.

The presentation of the second case study is structured as follows. First, in Section 4.1, we introduce two click models that simulate user behavior while browsing a ranked list: the standard *Cascade Model* [16, 61] and its extension, a *Terminating Cascade Model* (TCM), which introduces dependence on the order of items in the ranked list. Next, in Section 4.2, we introduce a Carousel click model, which allows us to simulate the user browsing behavior in a two-dimensional and topic-oriented presentation of recommendation results used by carousel-based interfaces. Then, we validate our model based on the fit to the real-world data and robustness in Sections 4.3 and 4.4, respectively. Finally, we demonstrate how the developed models of user behavior could be used to perform a simulation-based evaluation of specific recommender interfaces. In Section 4.5, we use CCM as a simulator to compare several ways to rank items in carousels. In Section 4.6 use both, CCM and TCM for a fine-grained comparison of the user behavior in the carousel and ranked list interfaces.

4.1 Ranked List Click Models

In this section, we introduce two models that allow one to simulate user behavior in a ranked list. We start by explaining the traditional Cascade Model and suggest its extension into a Terminating Cascade Model, which enables a more realistic simulation.

4.1.1 Cascade Model. The *cascade model (CM)* is a popular model of user behavior in a ranked list. In this model, the user is recommended a list of K items. The user examines the list from the first item to the last, and clicks on the first attractive item in the list. The items before the first attractive item are not attractive, because the user examines them, but does not click on them. Items after the first attractive item are not observed because the user never examines them.

The user's preference for the item $a \in E$ in the cascade model is represented by its *attraction probability* $p_a \in [0, 1]$. The attraction of item a is a random variable defined as $Y_a \sim \mathrm{Ber}(p_a)$. Fix list A. The click on position k is denoted by C_k and defined as $C_k = E_k Y_{A_k}$, where E_k is an indicator that position k is examined. By definition, the position is examined only if none of the higher-ranked items is attractive. Thus $E_k = \prod_{\ell=1}^{k-1}(1 - Y_{A_\ell})$ and the probability of a click on position k is

$$P_{\mathrm{CM}}(A, k) = \mathbb{E}\left[E_k Y_{A_k}\right] = \left[\prod_{\ell=1}^{k-1}(1 - p_{A_\ell})\right] p_{A_k} .$$

In turn, the probability of a click on list A is

$$P_{\mathrm{CM}}(A) = \mathbb{E}\left[\sum_{k=1}^{K} E_k Y_{A_k}\right] = \sum_{k=1}^{K} P_{\mathrm{CM}}(A, k) . \tag{1}$$

This model has two notable properties. First, since (1) increases whenever a less attractive item in A is replaced with a more attractive item from E, the optimal list in the CM,

$$A_* = \arg\max_A P_{\mathrm{CM}}(A) ,$$

contains K most attractive items. Therefore, it can be easily computed. Second, the click probability $P_{\mathrm{CM}}(A, k)$ can be used to assess whether a model reflects the ground truth. In particular, let $\hat{\mu} \in [0, 1]^K$ be the frequency of observed clicks on all positions in list A. Then the click model is a good model of reality if $\hat{\mu}$ resembles the output of the model. This similarity can be measured in many ways, and we adopt the *total variation distance* of click probabilities, $\frac{1}{2}\|P_{\mathrm{CM}}(A, \cdot) - \hat{\mu}\|_1$, in this work.

A click model is a mapping from items in a ranked list to probabilities of interaction with them. For a click model M and list A, let $P_M(A)$ denote how engaging the list A under model M is, such as the click probability $P_{\mathrm{CM}}(A)$ in (1). Click models can be used to answer several types of queries. Computation of $P_M(A)$ is the *evaluation* of how engaging list A under model M is. A natural extension is a *comparison* of two lists under a fixed model. Specifically, if $P_M(A) > P_M(A')$ for lists A and A', list A is more engaging than list A' under model M. Finally, we can also compare the same list under two different models. In particular, if $P_M(A) > P_{M'}(A)$ for models M and M', list A is more engaging under model M than M'.

4.1.2 Terminating Cascade Model. One shortcoming of the cascade model is that the order of the items in the optimal list does not affect the click probability. This is why extending this model to structured problems is difficult, because the position of the item does not matter. To introduce dependence on the order of items, we modify the CM as follows. When the examined item is not attractive, the user leaves unsatisfied with *termination probability* $p_q \in [0, 1]$. It models a situation where the user gets tired after examining unsatisfactory items. We call this model a *terminating cascade model (TCM)*.

TCM is one of many extensions of the cascade model [12], such as the user browsing model. The closest related extension is the dependent click model [30], where the user may not leave satisfied after an item is clicked. This model explains multiple clicks. In comparison, we model a user that may leave unsatisfied without clicking. Our model can also be viewed as an instance of the dynamic Bayesian network model [9], where the click probability decreases with the number of examined items.

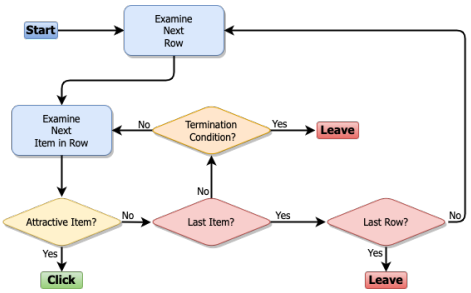


Fig. 2. Schematic view of TCM.

Fix list A. Let Q_k be an indicator that the user leaves unsatisfied at position k, which is defined as $Q_k \sim \text{Ber}(p_q)$. Then click on position k is defined as $C_k = E_k Y_{A_k}$, where E_k is an indicator that position k is examined. Since the position is examined only if none of the higher-ranked items is attractive, and the user does not leave unsatisfied upon examining these items, we have

$$E_k = \prod_{\ell=1}^{k-1} (1 - Q_\ell)(1 - Y_{A_\ell}) \,.$$

Thus the probability of a click on position k is

$$P_{\text{TCM}}(A, k) = \mathbb{E}\left[E_k Y_{A_k}\right] = (1 - p_q)^{k-1} \left[\prod_{\ell=1}^{k-1} (1 - p_{A_\ell})\right] p_{A_k} \tag{2}$$

and that on the list A is

$$P_{\text{TCM}}(A) = \mathbb{E}\left[\sum_{k=1}^{K} E_k Y_{A_k}\right] = \sum_{k=1}^{K} P_{\text{TCM}}(A, k) \,. \tag{3}$$

This model behaves similarly to the CM and has all of its desired properties. First, the optimal list in the TCM,

$$A_* = \arg\max_A P_{\text{TCM}}(A) \,,$$

contains K most attractive items in descending order of their attraction probabilities. Order matters because the position k in (3) is discounted by $(1 - p_q)^{k-1}$. Interestingly, this list is invariant to the value of the termination probability, as long as $p_q \in (0, 1)$. Second, since $P_{\text{TCM}}(A, k)$ can be easily computed for any list A and position k, the fit of the TCM to empirical click probabilities can be evaluated as in Section 4.1, using the total variation distance.

4.2 Carousel Click Model

In this section, we introduce a *carousel click model (CCM)*, which we developed to simulate user behavior in carousel-based interfaces. CCM is a natural extension of the single-list cascade models to the multi-list interfaces (i.e., carousels). For the purpose of modeling, a carousel-based interface could be represented as a matrix $A = (A_{i,j})_{i \in [m], j \in [K]}$, where m is the number of carousels (rows), K is the number of items per carousel (columns), and $A_{i,j}$ is the recommended item at position (i, j). To simplify notation, we denote carousel i in matrix A by $A_{i,:} = (A_{i,j})_{j=1}^{K}$. We assume that no item is in more than one carousel, that is $A_{i,j} \neq A_{i',j'}$ for any $(i, j) \neq (i', j')$.

4.2.1 User Behavior Assumptions. The assumptions of the overall user behavior in CCM extend the assumptions established for the simpler navigation-focused model in Section 3.1. The user examines the recommended matrix A from the first carousel until the last. When *carousel i is attractive*, at least one item in $A_{i,:}$ is attractive, the user starts to examine it and clicks on the first attractive item in it. To guarantee that the user can recognize an attractive carousel without examining it, we label the carousels with the topics of items in them. In this case, we can think of the user as having topics of interest on their mind and examining the first carousel with a matching topic. When *carousel i is not attractive*, no item in $A_{i,:}$ is attractive, the user proceeds to the next carousel $i+1$. When the user examines an unattractive carousel or item, they leave unsatisfied with probability $p_q \in [0,1]$, similar to the terminating cascade model in Section 4.1.2. Since each carousel is associated with a topic, the items in each carousel need to be semantically related. This amounts to a constraint on the items that can be in $A_{i,:}$.

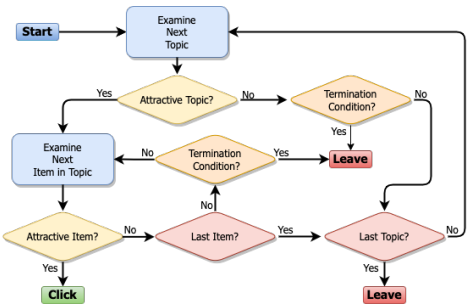


Fig. 3. Schematic view of CCM.

4.2.2 Click Probability. Fix matrix A. Let $Q_i \sim \mathrm{Ber}(p_q)$ be an indicator that the user leaves unsatisfied after examining the carousel i. Let $Q_{i,j} \sim \mathrm{Ber}(p_q)$ be an indicator that the user leaves unsatisfied after examining item at position (i,j). Then the indicator of a click on matrix A can be written as

$$\sum_{i=1}^{m} E_i \sum_{j=1}^{K} \left[\prod_{\ell=1}^{j-1} (1 - Q_{i,\ell})(1 - Y_{A_{i,\ell}}) \right] Y_{A_{i,j}} ,$$

where

$$E_i = \prod_{\ell=1}^{i-1} (1 - Q_\ell) \prod_{j=1}^{K} (1 - Y_{A_{\ell,j}})$$

is an indicator that carousel i is examined. The algebraic form of E_i follows from the fact that even E_i can occur only if all higher-ranked carousels are unattractive and the user does not leave unsatisfied after examining them. Thus, the probability of a click on matrix A is

$$P_{\mathrm{CCM}}(A) = \sum_{i=1}^{m} \mathcal{E}_i P_{\mathrm{TCM}}(A_{i,:}) , \tag{4}$$

where

$$\mathcal{E}_i = (1 - p_q)^{i-1} \prod_{\ell=1}^{i-1} \prod_{j=1}^{K} (1 - p_{A_{\ell,j}}) \tag{5}$$

is the probability that carousel i is examined.

4.2.3 Empirical Fit. Similarly to the CM and TCM, we also have a closed form for the click probability on position (i, j),

$$P_{\mathrm{CCM}}(A, (i,j)) = (1-p_q)^{i+j-2} \left[\sum_{\ell,s:(\ell<i)\vee(s<j)} (1-p_{A_{\ell,s}}) \right] p_{A_{i,j}} \,. \tag{6}$$

This can be used to assess if a model reflects the ground truth. In particular, let $\hat{\mu} \in [0,1]^{m \times K}$ be the frequency of observed clicks on all entries of matrix A. Then, if we treat $P_{\mathrm{CCM}}(A, \cdot)$ and $\hat{\mu}$ as vectors, the total variation distance of the click probabilities $\frac{1}{2}\|P_{\mathrm{CCM}}(A, \cdot) - \hat{\mu}\|_1$ measures whether the CCM is a good model of reality.

4.2.4 Optimal Solution. The optimal solution in the CCM does not have a closed form anymore. Nevertheless, we can still characterize some of its properties. Specifically, in the optimal matrix $A_* = \arg\max_A P_{\mathrm{CCM}}(A)$, the items in each carousel must be ordered from the highest attraction probability to the lowest. This can be seen as follows. For any matrix A and carousel i in it, $P_{\mathrm{TCM}}(A_{i,:})$ in (4) has the highest value when the attraction probabilities in $A_{i,:}$ are in a descending order. This argument is analogous to that in the TCM (Section 4.1.2). This change has no impact on $\mathcal{E}_i$ in (5). Regarding the order of carousels, we approximate A_* by presenting the carousels in the descending order of their total attraction probabilities, $\sum_{j=1}^{K} p_{A_{i,j}}$. This guarantees that carousels with more attractive items are presented first, which minimizes the probability of users leaving unsatisfied.

4.2.5 CCM vs. TCM. To stress the difference between CCM and TCM, it is useful to show that carousel click model can lead to higher click probabilities than the TCM, under the assumption that the attraction and termination probabilities in both models are comparable. Because the parameters of the models are comparable, this shows that the structure can be beneficial in recommendations.

We compare the TCM and CCM under the assumption that all attraction probabilities are identical and small. Specifically, let $p_a = p$ for all items a and $p = O(1/Km)$. Then $(1-p)^k = O(1)$ for any $k \in [Km]$. In the TCM, we view A as a single ranked list of Km items. Then

$$P_{\mathrm{TCM}}(A) \approx p \sum_{k=1}^{Km} (1-p_q)^{k-1}, \quad P_{\mathrm{CCM}}(A) \approx p \sum_{i=1}^{m} \sum_{j=1}^{K} (1-p_q)^{i+j-2} \,.$$

Now when we bound $j-1$ in $P_{\mathrm{CCM}}(A)$ as $j - 1 \le K(j-1)$, the two objectives become equal. Since $1 - p_q \le 1$ and $j - 1 \ge 0$, we get $P_{\mathrm{TCM}}(A) \le P_{\mathrm{CCM}}(A)$. The improvement is due to the fact that the user is much less likely to leave unsatisfied with the CCM.

4.3 Validation of CCM: Fit to Real-World Data

To study how well CCM and TCM model user behavior, we fit them to an existing dataset of real user interactions. The dataset was collected by Jannach et al. [37] to assess the effect of different design choices on human decision-making. It contains $n = 776$ instances of clicks on recommendations presented in two settings, ranked list and carousels, with a comparable number of clicks in each setting. Despite its small size, we found that this dataset is the only publicly available data source of human interaction with a carousel-based interface, which can be used to validate CCM.

The dataset has two parts. In both parts, the recommended items are presented in $m = 5$ rows and $K = 4$ items per row. In the first part (conditions 1 to 4), the items are presented as a single ranked list, row by row. We hypothesize that the user scans these items row by row, from left to right, and clicks on the first attractive item. We call these data *ranked list*. The second part of the data (conditions 5 to 8) is similar to the first except that each row is labeled with the topic of items in that row, such as "Action Movies". This can be viewed as a list of carousels. We hypothesize that the user scans

the carousels from top to bottom and stops at the first attractive topic. After that, they examine the items within that carousel from left to right and click on the first attractive item. We call these data *carousel*.

In both the ranked list and the carousel data, we compute empirical click probabilities for all positions and plot their logarithm in Figure 4. A closer look at these probabilities reveals different user interaction patterns in the two settings. In the carousel data, we observe more clicks in the first column, which represents the first items in all carousels. This is in contrast to ranked list data, where clicks are more concentrated in the first row, which represents the highest positions in the ranked list. The difference between the interactions is consistent with our proposed mathematical models, and we provide more quantitative evidence below.

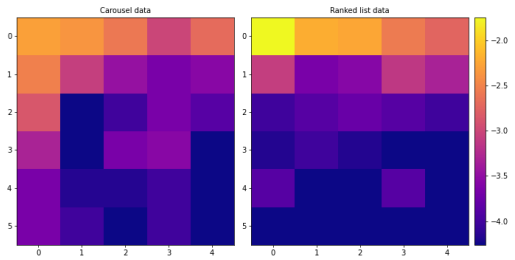


Fig. 4. Empirical click probabilities for carousel (left) and ranked list (right) data.

One challenge in evaluating our mathematical models is that most items in our dataset appear only once. Therefore, it is impossible to accurately estimate their attraction probabilities. However, we know that the items are recommended in decreasing relevance order. Therefore, we parameterize the attraction probabilities of the items as follows. In the TCM, the click probability at position (i, j) is computed as in (2), where $k = K(i-1)+j$ and $p_{A_k} = p_0\gamma^{k-1}$. Here $p_0 \in [0, 1]$ is the highest attraction probability and $\gamma \in [0, 1]$ is its discount factor. Note that the attraction probability decreases with the rank of the item in the list, which is presented as a matrix. We denote the click probability at position (i, j) by $\mu_{\text{TCM}}(i, j; p_0, \gamma, p_q)$, with parameters p_0, γ, and p_q. In the CCM, the click probability at position (i, j) is calculated as in (6), where $p_{A_{i,j}} = p_0\gamma^{i+j-2}$ and all parameters are defined as in the TCM. Again, the attraction probability decreases with the number of rows and columns, and we denote the click probability at position (i, j) by $\mu_{\text{CCM}}(i, j; p_0, \gamma, p_q)$.

Let $\hat{\mu}_{\text{LIST}} \in [0, 1]^{m\times K}$ and $\hat{\mu}_{\text{CAROUSELS}} \in [0, 1]^{m\times K}$ denote the matrices of empirical click probabilities estimated from the ranked list and carousel data (Figure 4), respectively. For each model $M \in \{\text{TCM}, \text{CCM}\}$ and dataset $\mathcal{D} \in \{\text{LIST}, \text{CAROUSELS}\}$, we compute

$$\delta_{M,\mathcal{D}} = \frac{1}{2} \min_{p_0,\gamma,p_q\in[0,1]^3} \|\mu_M(\cdot, \cdot; p_0, \gamma, p_q) - \hat{\mu}_{\mathcal{D}}\| .$$

This quantity measures the fit between the hypothesized model, represented by $\mu_M(\cdot, \cdot; p_0, \gamma, p_q)$ and optimized over $p_0, \gamma, p_q \in [0, 1]^3$, and the empirical evidence, $\hat{\mu}_{\mathcal{D}}$. We approximate the exact minimization over $[0, 1]^3$ by grid search, where the grid resolution is 0.01. We report all total variation distances in Table 1.

Our results in Table 1 show that TCM fits the ranked list data better (smaller total variation distance 0.086) than CCM (larger total variation distance 0.095). They also show that CCM fits the carousel data better (smaller total variation distance 0.128) than the TCM (larger total variation distance 0.133). In summary, our mathematical models of click probabilities in TCM and CCM match the observed

Model M	Dataset $\mathcal{D}$	p_0	γ	p_q	$\delta_{M,\mathcal{D}}$
TCM	Ranked list	0.17	0.92	0.02	**0.086**
CCM	Ranked list	0.17	0.9	0.02	0.095
TCM	Carousel	0.099	0.96	0.01	0.133
CCM	Carousel	0.11	0.84	0.01	**0.128**

Table 1. Comparing the total variation distance between TCM and CCM models on ranked list and carousel data.

clicks. We also use the results of this experiment to set the value of the termination probability p_q in the remaining experiments. This value is $p_q = 0.01$.

4.4 Validation of CCM: Robustness

The purpose of our second validation experiment is to show that CCM generalizes to an unseen test set. Specifically, we show that the optimal recommendation under CCM in the training set also has a high value in the test set.

This experiment is carried out on the MovieLens 1M dataset [31], which consists of 1 million ratings applied to 4000 movies (items) in 18 genres (topics) by 6000 users. For simplicity, we assume that each movie is associated only with one genre. For a movie with more than one genre, we assign the genre with the highest popularity among all users. The recommended movies in CCM are organized in 18 carousels. Each carousel represents a genre and has a label that shows the topic of the carousel, such as "Action Movies". We denote by n_u the number of users and by n_a the number of items.

We randomly split the dataset into two equal sets, which we call the training set $\hat{\mathcal{D}}$ and the test set $\mathcal{D}$. Then, for all users and items, we estimate the ratings using matrix factorization, which is a standard approach for rating estimation in recommender systems [44]. The approach involves decomposing a sparse rating matrix into a low-rank matrix that captures latent factors representing user and movie preferences. This approach has been shown to achieve high prediction accuracy and has been extensively studied by researchers. In our work, we used non-negative matrix factorization with $d = 5$ latent factors to estimate the ratings for movies that the user has not seen or ranked. We denote the estimated rating of item $a \in [n_a]$ by user $u \in [n_u]$ in the training (test) set by $\hat{r}_{u,a}$ ($r_{u,a}$). Next, we apply a softmax transformation to the estimated ratings and convert them into attraction probabilities in both the training and test sets,

$$\hat{p}_{u,a} = \frac{\exp[\hat{r}_{u,a}]}{\sum_{a=1}^{n_a} \exp[\hat{r}_{u,a}]}, \quad p_{u,a} = \frac{\exp[r_{u,a}]}{\sum_{a=1}^{n_a} \exp[r_{u,a}]}.$$

This is just a monotone transformation that transforms the estimated ratings of each user into a probability vector.

We evaluate CCM as follows. Let $\hat{P}_{\mathrm{CCM}}(A, u)$ ($P_{\mathrm{CCM}}(A, u)$) be the click probability on recommendation A by user u on the training (test) set, calculated using (4) and attraction probabilities $\hat{p}_{u,a}$ ($p_{u,a}$). First, we compute the best recommendation for user u on the training set $\hat{A}_u = \arg\max_A \hat{P}_{\mathrm{CCM}}(A, u)$, where maximization is performed as described in Section 4.2.4. Second, we evaluate $\hat{A}_u$ on the test set, by computing the test click probability $P_{\mathrm{CCM}}(\hat{A}_u, u)$. Third, we calculate the best recommendation for user u on the test set $A_{u,*} = \arg\max_A P_{\mathrm{CCM}}(A, u)$, where the maximization is done as described in Section 4.2.4. Finally, we compare the average test click probabilities, for all users u, of the best

training and test recommendations, formally given by

$$\frac{1}{n_u}\sum_{u=1}^{n_u} P_{\text{CCM}}(\hat{A}_u, u)\,, \quad \frac{1}{n_u}\sum_{u=1}^{n_u} P_{\text{CCM}}(A_{u,*}, u)\,.$$

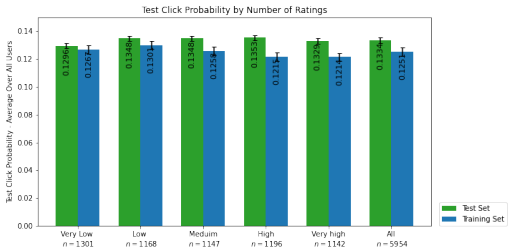


Fig. 5. Average test click probabilities of the best recommendations on the training and test sets.

Our results are shown in Figure 5. In addition to reporting the average click probability over all users, we break it into five groups based on the number of ratings per user: "very low" to "very high" which provides an intuitive understanding of the number of ratings associated with each bin. This categorization allows us to analyze and compare different subsets of users based on the number of ratings they provided. We choose the number of ratings because it represents the size of the user profile. The results of this experiment confirm that CCM can generate recommendations comparable to the best recommendations on the test set, both overall (3.96% difference) and in the five user groups (2.9-4.3% difference). Our independent t-test did not yield a statistically significant difference between the two groups compared. This testifies to the generalizability of our proposed model. In particular, we show that our model does not overfit and performs well on the test set.

4.5 Using CCM Simulator: Comparing Different Levels of Personalization

This experiment investigates the effect of personalization on the click probability of recommendations. We compare the click probability of recommendations generated by CCM using personalized and two non-personalized attraction probabilities.

The setup of this experiment is the same as in Section 4.4. The only difference is in the definitions of ratings in the training set. This definition affects how the optimal recommendation $\hat{A}_u$ is chosen, but not how it is evaluated. That is, $P_{\text{CCM}}(\hat{A}_u, u)$ is the value of $\hat{A}_u$ under the personalized ratings on the test, as defined in Section 4.4. The first approach, PERSONALIZED, uses the definition of training set ratings in Section 4.4. The second approach, POPULAR, calculates the rating of item a for user u as

$$\hat{r}_a = \frac{\sum_{u=1}^{n_u} \hat{r}_{u,a}}{n_u}\,.$$

This is not personalized because the rating of item a is the same for all users. The last approach, RANDOM, assigns random ratings $\hat{r}_a \in [1, 5]$ to all items.

Our results are reported in Figure 6. In addition to reporting the average click probability over all users, we break it down into the same five user groups as in Figure 5. As explained before, this segmentation helps to better visualize the effect of user profile size on click probability. We observe that the personalized model outperforms the non-personalized baselines by a large margin. Specifically, the average click probability over all users in POPULAR decreases by 23.8% from that in PERSONALIZED; and the average click probability over all users in RANDOM decreases by 32.3% from

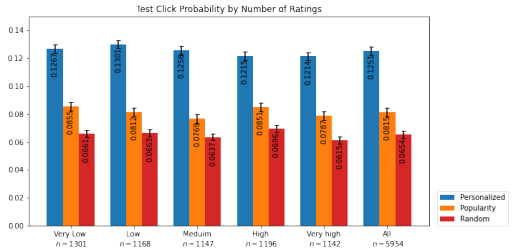


Fig. 6. Comparing the click probability of CCM in different settings.

that in PERSONALIZED. Our independent t-test results indicate that the improvement gained from the use of personalized recommendations compared to the nonpersonalized baselines is statistically significant with a p-value of less than 0.05. These improvements are consistent across all five user groups, indicating stability with respect to profile size. We conduct this experiment because the non-personalized baselines depend less on the training data of a given user, and thus may generalize better to the test set. This experiment shows that this is not the case.

4.6 Using CCM and TCM Simulators: Comparing User Behavior in Carousel and Ranked List

Earlier in this section, we demonstrated that our carousel click model is robust and fits the real-world data. Additionally, we showed that a personalized model outperforms the random and popularity-based models of user behavior in carousels. In this part, we compare the user behavior in a carousel and a ranked list under the standard user behavior assumption introduced Section 4.2.1. We also repeat the same set of simulations under a more realistic user behavior assumption that considers the initial visibility of items in carousels. Finally, we utilize the log click probability to demonstrate how users browse a carousel compared to a ranked list.

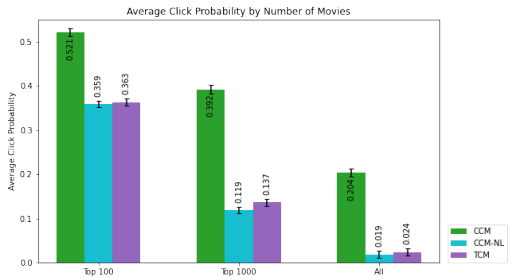


Fig. 7. Comparing the average click probability in CCM, CCM-NL and TCM.

4.6.1 Comparing Carousels and Ranked List under Standard User Behavior Assumption.

This experiment compares the click probabilities under CCM with two other click models: TCM and a variant of CCM called CCM-NL. The goal of this experiment is to show the gain in click probabilities when using CCM.

TCM (Section 4.1.2) is a cascade click model [16] that models a user that may terminate unsatisfied. CCM-NL, which stands for a *carousel click model with no labels*, is a variant of CCM that models the scenario when topic labels are removed. Specifically, we calculate the optimal list using CCM

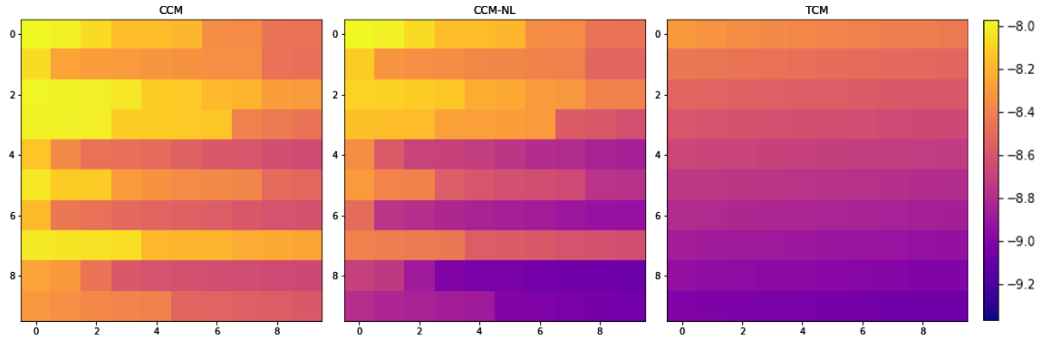


Fig. 8. Sample distribution of log click probability in CMM, CMM-NL and TCM for all users. Darker colors mean less click probability.

and then remove the labels of the carousels. This means that the user browses the recommendations as if they were a single ranked list, and we adopt TCM for this simulation. We do this to study the importance of labels for carousels and to see to what extent they affect the average click probability in CCM. We use the same evaluation protocol as in Section 4.4.

Our results are reported in Figure 7. Click probabilities are reported for three different sets of items: top 100, top 1 000, and all; where items are ranked by the sum of their ratings. This experiment shows that the click probability in CCM is significantly (independent t-test, p-value < 0.01) higher than in TCM and CCM-NL. Specifically, when recommending the top 100 items, the click probability in TCM decreases by 27.9% from that in CCM; and the click probability in CCM-NL decreases by 33.5.0% from that in CCM. The improvement increases as the number of items increases. Specifically, when recommending all items, the click probability in TCM decreases by 90.02% from that in CCM; and the click probability in CCM-NL decreases by 91.9% from that in CCM. In summary, CCM attracts about 10 times more clicks than both TCM and CCM-NL when recommending all items. This improvement trend indicates that CCM is a good candidate for practice, where a large number of recommended items is common.

4.6.2 A More Realistic User Behavior Assumption. CCM assigns the same termination probability to all items in the carousel. However, in practice, items that are not initially displayed are less likely to be examined because the user needs to scroll to see them. To study this behavior, we assign different termination probabilities to different parts of the carousel: $p_q = 0.01$ to the first 10 columns and $p_q = 0.1$ to the rest. Selecting the first 10 items as the visible part of the carousel is an intuitive choice based on real-life recommender systems. In other aspects, the setting is the same as in Section 4.6.

Figure 9 compares the CCM, CCM-NL, and TCM in the new setting. We observe that the performance of CCM worsens. In absolute terms, the click probability for top 100 items is comparable to that in Figure 7. However, for all items, it is about 5 times lower. Relatively to the baselines, when recommending top 100 items, the click probability in TCM decreases by 23.9% from that in CCM; and the click probability in CCM-NL decreases by 27.81% from that in CCM. When recommending all items, the click probability in TCM decreases by 40.11% from that in CCM; and the click probability in CCM-NL decreases by 51.4% from that in CCM. Although the improvement for all items is not as impressive as in Figure 7, CCM still outperforms both baselines by a healthy margin.

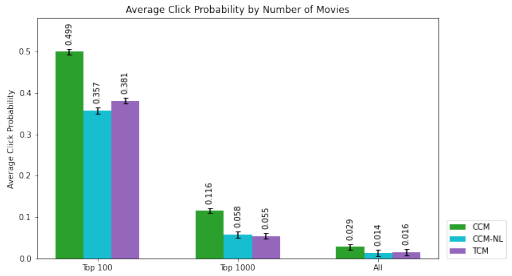


Fig. 9. Comparing the average click probability in a more realistic CCM, CCM-NL and TCM.

4.6.3 Visualizing the Log Click Probability. To visualize the reason behind the considerably better performance in CCM, we plot the average log click probability for all users using (6) in the first ten rows and columns of the optimal recommendation in CCM and the two baselines in Figure 8. The light color (yellow) in the plots corresponds to high average click probabilities, whereas the dark color (blue) corresponds to low average click probabilities. In CCM, we observe that the items at the beginning of each row (the first few items in each carousel) are more likely to be clicked by the user. We can also observe the effect of removing labels from the carousels in CCM-NL. This is manifested by overall darker colors because the user struggles to find the carousel with attractive items and leaves unsatisfied. The last plot shows that the average click probability in TCM decreases uniformly. This is expected when the recommended items are ranked in the order of decreasing relevance.

5 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

In this paper, we advocate the use of a simulation-based approach for the evaluation of recommender interfaces. To demonstrate the power of this approach, we presented two case studies in which a comparative evaluation of carousel-based interfaces was performed using an offline simulation. The two case studies attempt to answer research questions of different complexity and use considerably different models of user behavior to answer these research questions using simulation-based study with these models. The first case study demonstrates the use of a relatively simple *qualitative model* of user behavior to compare the navigability of a carousel-based interface and a ranked list. The second case study demonstrates how a more complex *quantitative model* of user behavior supports more elaborated simulation-based studies that could answer more comples research questions, which require a higher precision of similation. To perform the latter study, we developed and validated *a carousel click model*, which generalizes the traditional *cascade models* [15] that model user behavior in a ranked list. Taken together, the two case studies demonstrate that user behavior could be modeled at different levels of complexity and that the complexity of the simulation model should be selected by taking into account the complexity of research questions to be answered through a simulation-based study with this model.

Although we consider this demonstration to be the main contribution of the paper, we stress two additional contributions that carry a separate value for the field. First, to enable simulation-based studies of carousel-based interfaces, we developed and validated a carousel click model (CCM). To quantify user work with multiple carousels, the model proposes that the user examines carousels proportionally to the attraction of items in them and then clicks on the items within the examined carousel proportionally to their individual attraction. This model is motivated by the cascade model developed for the traditional "single" ranked list. We consider the carousel click model as a valuable theoretical and practical contribution, which is important well beyond the scope of this paper. In

particular, this model enables further simulation-based quantitative evaluation of carousel-based interfaces.

Second, as part of our second case study, we were able to compare a carousel-based interface and a traditional ranked list on equal ground. In this simulation-based comparison, we used the traditional cascade model to simulate user behavior in a ranked list and a carousel click model to simulate user behavior in a carousel-based interface. Our results demonstrated that a structured examination of a large item space supported by carousels is more efficient than scanning a single ranked list. These findings help to explain a rapid rise of carousel-based recommendation interfaces, which have become a de-facto standard approach to recommending items to end users in many real-life recommenders.

We consider this work as the first attempt to explore the application of a simulation-based approach to study recommender system interfaces "beyond the ranked list", such as carousel-based interfaces. As the first attempt, the work has a number of limitations that point to several directions for future work. First, the goal of this paper was to make a case for the simulation-based evaluation approach and demonstrate its applicability to answer valuable research questions (for example, to compare a carousel-based and a ranked list presentation of results). While we provided two cases to demonstrate that similation-based approach could be use to answer different research questions and that the user behavior could be modeled at different levels of complexity, we used only one recommender algorithm in each case for the demonstration. In real life, the same behavior model could be used for simulation-based studies of different algorithms, moreover, it could be used to compare different algorithms "on an equal ground". While using the simulation-based approach to compare a set of advanced recommendation algorithms was not the goal of the current paper, we plan to explore this opportunity in our future work. In particular, we want to use simulation-based evaluation to assess the value of multi-armed bandits-based methods in a carousel context.

Second, in the process of validating our models, we observed that there is no large-scale public dataset of user interactions with carousel interfaces. To our knowledge, the dataset used in Section 4.3 is the largest such data set, yet it is too small to fully validate our model, which is a limitation of our study. We believe that it is important to release a large-scale public carousel interaction dataset to encourage more research in this area, similar to what the Yandex dataset [42] did for regular click models [12]. We plan to collect and publish such a dataset in the future.

Third, every behavior simulation model has its limits, even if it is developed on the basis of past empirical data. To better understand the limits of simulation-based evaluation, it is important to periodically compare the results obtained by offline simulation studies with the results obtained in empirical studies with end users. This comparison has been made for the traditional ranked list interfaces leading to valuable insights [28, 55, 63]. In our future work, we plan to combine simulation-based evaluation of carousel interfaces with empirical studies of these interfaces.

Finally, to obtain more reliable results, it is important to perform studies in multiple domains. While the study presented in this paper has been carried out in the popular domain of movie recommendations, we plan to expand it to other domains, such as food recommendations. Specifically, we want to understand how carousel-based interfaces influence user preferences in culinary choices and how this might promote a healthier lifestyle. Furthermore, we plan experiments in the domain of health recommendations, particularly focusing on health-related document recommendations for patients and their caregivers. By expanding our research into these domains, we aim to gain a deeper understanding of the advantages and challenges of incorporating carousels in presenting recommendation results.

Going beyond our own future work, two important future directions should be acknowledged. First, our carousel click model is only one potential model for carousels, motivated by the cascade model in ranked lists. An alternative model could be based on the position-based model, where the

user would examine the item at position (i, j) proportionally to the probability of examining the row i and the column j. We believe that many of these models could be developed in the future, similar to the countless click models developed for ranked lists [9].

Finally, the essence of our advocated approach is to use reliable models of user behavior to simulate user work with recommender interfaces and perform various types of evaluation *offline* without engaging real users. In this paper, we made the case for simulation-based evaluation of carousel-based recommender interfaces. However, with the advancement of our knowledge of user behavior in other types of recommender interfaces, reliable behavior models could be built for more sophisticated interfaces. With that, the application scope of the simulation-based approach will be considerably increased.

REFERENCES

[1] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. 2006. Learning User Interaction Models for Predicting Web Search Result Preferences. In *Proceedings of the 29th Annual International ACM SIGIR Conference*. ACM, New York, NY, 3–10.

[2] Walid Bendada, Guillaume Salha, and Bontempelli. 2020. Carousel Personalization in Music Streaming Apps with Contextual Bandits. In *Fourteenth ACM Conference on Recommender Systems*. ACM, New York, NY, 420–425.

[3] Walid Bendada, Guillaume Salha, and Théo Bontempelli. 2020. Carousel personalization in music streaming apps with contextual bandits. In *Proceedings of the 14th ACM Conference on Recommender Systems*. ACM, New York, NY, 420–425.

[4] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: A Visual Interactive Hybrid Recommender System. In *6th ACM Conference on Recommender System*. ACM, New York, NY, 35–42.

[5] Dimitrios Bountouridis, Jaron Harambam, Mykola Makhortykh, Mónica Marrero, Nava Tintarev, and Claudia Hauff. 2019. SIREN. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, New York, NY, 150–159.

[6] Robin Burke, Kristian Hammond, and Benjamin C. Young. 1997. The FindMe Approach to Assisted Browsing. *IEEE Intelligent Systems* 12, 4 (1997), 32—40.

[7] Stuart K. Card, Peter Pirolli, M. Van De Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, and J. Boshart. 2001. Information scent as a driver of web behavior graphs: Results of a protocol analysis method for web usability. In *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'2001)*. ACM Press, New York, NY, 498–505.

[8] John Champaign and Robin Cohen. 2013. Ecological Content Sequencing: From Simulated Students to an Effective User Study. *International Journal of Learning Technology* 8, 4 (2013), 337–361.

[9] Olivier Chapelle and Ya Zhang. 2009. A Dynamic Bayesian Network Click Model for Web Search Ranking. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, New York, NY, 1–10.

[10] Li Chen and Pearl Pu. 2012. Critiquing-Based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 125–150.

[11] David Chin. 2001. Empirical Evaluations of User Models and User-Adapted Systems. *User Modeling and User-Adapted Interaction* 11, 1-2 (2001), 181–194.

[12] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers, New York, NY.

[13] Richard Combes, Stefan Magureanu, Alexandre Proutiere, and Cyrille Laroche. 2015. Learning to Rank: Regret Lower Bounds and Efficient Algorithms. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (Portland, Oregon, USA) *(SIGMETRICS '15)*. Association for Computing Machinery, New York, NY, USA, 231–244. https://doi.org/10.1145/2745844.2745852

[14] Nick Craswell. 2009. Mean Reciprocal Rank. In *Encyclopedia of Database Systems*, Ling Liu and M. Tamer Özsu (Eds.). Springer US, Boston, MA, 1703–1703.

[15] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-Bias Models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (Palo Alto, California, USA) *(WSDM '08)*. Association for Computing Machinery, New York, NY, USA, 87–94.

[16] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-Bias Models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, 87–94.

[17] Michel C Desmarais and Ryan SJ d Baker. 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction* 22 (2012), 9–38.

[18] Andrew Dillon, John Richardson, and Cliff McKnight. 1990. Navigation in hypertext: A Critical review of the concept. http://hdl.handle.net/10150/106184 This item is not the definitive copy. Please use the following citation when referencing this material: Dillon, A., Richardson, J. and McKnight, C. (1990) Navigation in Hypertext: a critical review of the concept. In D.Diaper, D.Gilmore, G.Cockton and B.Shackel (eds.) Human-Computer Interaction-INTERACT'90. North Holland: Amsterdam, 587-592. Abstract: With the advent of hypertext it has become widely accepted that the departure from the so-called "linear" structure of paper increases the likelihood of readers or users becoming lost. In this paper we will discuss this aspect of hypertext in terms of its validity, the lessons to be learned from the psychology of navigation and the applicability of the navigation metaphor to the hypertext domain.

[19] Hendrik Drachsler, Hans Hummel, and Rob Koper. 2008. Using Simulations to Evaluate the Effects of Recommender Systems for Learners in Informal Learning Networks. In *2nd SIRTEL'08 Workshop on Social Information Retrieval for Technology Enhanced Learning (CEUR Workshop Proceedings, Vol. 382)*, Riina Vuorikari, Barbara Kieslinger, Ralf Klamma, and Erik Duval (Eds.). CEUR, New York, NY.

[20] Daria Dzyabura and Alex Tuzhilin. 2013. Not by search alone. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, New York, NY, 371–374.

[21] Nicolò Felicioni, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2021. A Methodology for the Offline Evaluation of Recommender Systems in a User Interface with Multiple Carousels. In *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, New York, NY, 10–15.

[22] Maurizio Ferrari Dacrema, Nicolò Felicioni, and Paolo Cremonesi. 2021. Optimizing the selection of recommendation carousels with quantum computing. In *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, New York, NY, 691–696.

[23] Maurizio Ferrari Dacrema, Nicolò Felicioni, Paolo Cremonesi, et al. 2022. Evaluating Recommendations in a User Interface With Multiple Carousels. In *CEUR workshop proceedings*, Vol. 3177. CEUR, CEUR, Online.

[24] Laura Granka, Thorsten Joachims, and Geri Gay. 2004. Eye-tracking analysis of user behavior in WWW search. In *SIGIR '04: the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, 478–479.

[25] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About Playlist Recommendation Algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, 420–428.

[26] Zhiwei Guan and Edward Cutrell. 2007. An eye tracking study of the effect of target rank on web search. In *CHI '07: ACM SIGCHI conference on human factors in computing systems*. ACM Press, New York, NY, 417–420.

[27] Zhiwei Guan and Edward Cutrell. 2007. What are you looking for?: an eye-tracking study of information usage in web search. In *CHI '07: ACM SIGCHI conference on human factors in computing systems*. ACM Press, New York, NY, 407–416.

[28] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, Boston, MA.

[29] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi Min Wang, and Christos Faloutsos. 2009. Click Chain Model in Web Search. In *Proceedings of the 18th International Conference on World Wide Web*. WWW, New York, NY, 11–20.

[30] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient Multiple-Click Models in Web Search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, 124–131.

[31] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2015), 19 pages.

[32] Naieme Hazrati, Mehdi Elahi, and Francesco Ricci. 2020. Simulating the Impact of Recommender Systems on the Evolution of Collective Users' Choices. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*. ACM, New York, NY, 207–212.

[33] Naieme Hazrati and Francesco Ricci. 2022. Simulating Users' Interactions with Recommender Systems. In *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, New York, NY, 95–98.

[34] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56, C (2016), 59–27.

[35] Denis Helic, Sebastian Wilhelm, Ilire Hasani-Mavriqi, and Markus Strohmaier. 2011. The Effects of Navigation Tools on the Navigability of Web-based Information Systems. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. ACM, New York, NY, 16:1–16:8. https://doi.org/10.1145/2024288.2024308

[36] Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. 2019. RecSim: A Configurable Simulation Platform for Recommender Systems. https://doi.org/10.48550/ARXIV.1909.04847

[37] Dietmar Jannach, Mathias Jesse, Michael Jugovac, and Christoph Trattner. 2021. Exploring Multi-List User Interfaces for Similar-Item Recommendations. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and*

Personalization. ACM, New York, NY, 224–228.

[38] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. 2017. User Control in Recommender Systems: Overview and Interaction Challenges. In *International Conference on Electronic Commerce and Web Technologies (Lecture Notes in Business Information Processing, Vol. 278)*. Springer, New York, NY, 21–33.

[39] Kalervo Jarvelin and Jaana Kekalainen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[40] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. ACM, New York, NY, 133–142.

[41] Mark Keane, Maeve O'Brien, and Barry Smyth. 2008. Are people biased in their use of search engines? *Commun. ACM* 51, 2 (2008), 49–52.

[42] Eugene Kharitonov and PS Yandex. 2013. Yandex Personalized Web Search Challenge. https://www.kaggle.com/c/yandex-personalized-web-search-challenge.

[43] Daniel Kluver, Michael D. Ekstrand, and Joseph A. Konstan. 2018. *Rating-Based Collaborative Filtering: Algorithms and Evaluation*. Springer International Publishing, Cham, 344–390. https://doi.org/10.1007/978-3-319-90092-6_10

[44] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[45] Johannes Kunkel, Benedikt Loepp, and Jürgen Ziegler. 2017. A 3D Item Space Visualization for Presenting and Manipulating User Preferences in Collaborative Filtering. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, New York, NY, 3–15. https://doi.org/10.1145/3025171.3025189

[46] Branislav Kveton, Csaba Szepesvári, Zheng Wen, and Azin Ashkan. 2015. Cascading bandits: learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, New York, NY, 767–776.

[47] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S. Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline Evaluation of Ranking Policies with Click Models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1685–1694.

[48] Luyi Ma, Nimesh Sinha, Parth Vajge, Jason HD Cho, Sushant Kumar, and Kannan Achan. 2021. Event-based Product Carousel Recommendation with Query-Click Graph. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, IEEE, New York, NY, 4119–4125.

[49] Noboru Matsuda, William W. Cohen, Jonathan Sewall, Gustavo Lacerda, and Kenneth R. Koedinger. 2007. Predicting Students Performance with SimStudent: Learning Cognitive Skills from Observation. In *13th International Conference on Artificial Intelligent in Education, AI-ED 2007*, Rosemary Luckin, Kenneth R. Koedinger, and Jim Greer (Eds.). IOS, Amsterdam, Netherlands, 467–476.

[50] Javed Mostafa, Snehasis Mukhopadhyay, and Mathew Palakal. 2003. Simulation Studies of Different Dimensions of Users' Interests and their Impact on User Modeling and Information Filtering. *Information Retrieval* 6, 2 (01 April 2003), 199–223.

[51] Fedelucio Narducci, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2018. Improving the User Experience with a Conversational Recommender System. In *International Conference of the Italian Association for Artificial Intelligence*. Springer, Italy, 528–538. https://doi.org/10.1007/978-3-030-03840-3_39

[52] Bing Pan, Helene Hembrooke, Geri Gay, Laura Granka, Matthew Feusner, and Jill Newman. 2004. The determinants of web page viewing behavior: an eye-tracking study. In *ETRA'2004: Proceedings of the symposium on Eye tracking research and applications*. ACM, New York, NY, 147–154. http://portal.acm.org/citation.cfm?id=968391

[53] Zachary A. Pardos and Neil Heffernan. 2011. KT-IDEM: introducing item difficulty to the knowledge tracing model. In *19th International Conference on User Modeling, Adaptation, and Personalization, UMAP 2011*, Joseph Konstan, Ricardo Conejo, Josep Marzo, and Nuria Oliver (Eds.), Vol. 6787. Springer-Verlag, New York, NY, 243–254.

[54] Denis Parra and Peter Brusilovsky. 2015. User-controllable personalization: A case study with SetFusion. *International Journal of Human-Computer Studies* 78 (2015), 43–67.

[55] Ladislav Peska and Peter Vojtas. 2020. Off-line vs. On-line Evaluation of Recommender Systems in Small E-commerce. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*. ACM, New York, NY, 291–300.

[56] Peter Pirolli and Wai-Tat Fu. 2003. SNIF-ACT: A model of information foraging on the World Wide Web. In *9th International User Modeling Conference (Lecture Notes in Artificial Intelligence, Vol. 2702)*, Peter Brusilovsky, Albert Corbett, and Fiorella de Rosis (Eds.). Springer Verlag, New York, NY, 45–54.

[57] Behnam Rahdari and Peter Brusilovsky. 2022. Simulation-Based Evaluation of Interactive Recommender Systems. In *9th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with the 16th ACM Conference on Recommender Systems (RecSys 2022) (CEUR Workshop Proceedings, Vol. 3222)*. CEUR, online, 122–136. http://ceur-ws.org/Vol-3222/paper8.pdf

[58] Behnam Rahdari, Peter Brusilovsky, and Dmitriy Babichenko. 2020. Personalizing Information Exploration with an Open User Model. In *31st ACM Conference on Hypertext and Social Media*. ACM, New York, NY, 167–176.

[59] Behnam Rahdari, Branislav Kveton, and Peter Brusilovsky. 2022. The magic of carousels: Single vs. multi-list recommender systems. In *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*. ACM, New York, NY, 166–174.

[60] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*. Springer, Boston, MA, 1–34.

[61] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 521–530. https://doi.org/10.1145/1242572.1242643

[62] David Rohde, Stephen Bonner, Travis Dunlop, Flavian Vasile, and Alexandros Karatzoglou. 2018. RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. https://doi.org/10.48550/ARXIV.1808.00720

[63] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, New York, NY, 31–34.

[64] Barry Smyth, Lorraine McGinty, James Reilly, and Kevin McCarthy. 2004. Compound Critiques for Conversational Recommender Systems. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI '04)*. IEEE Computer Society, USA, 145–151.

[65] Alain Starke, Edis Asotic, and Christoph Trattner. 2021. "Serving Each User": Supporting Different Eating Goals Through a Multi-List Recommender Interface. In *Fifteenth ACM Conference on Recommender Systems*. ACM, New York, NY, 124–132.

[66] Danny Sullivan. 2001. How Direct Hit Works. *Search Engine Watch* (2001). https://searchenginewatch.com/sew/news/2047514/how-direct-hit-works

[67] Christoph Trattner, Yi-Ling Lin, Denis Parra, Zhen Yue, William Real, and Peter Brusilovsky. 2012. Evaluating Tag-Based Information Access in Image Collections. In *Proceedings of the 23rd ACM conference on Hypertext and hypermedia*. ACM, New York, NY, USA, 113–122.

[68] Petros Venetis, Georgia Koutrika, and Hector Garcia-Molina. 2011. On the selection of tags for tag cloudsx. In *Fourth ACM International Conference on Web Search and Data Mining, WSDM '11*, Vol. 29. ACM, New York, NY, 835–844.

[69] Katrien Verbert, Denis Parra-Santander, and Peter Brusilovsky. 2016. Agents Vs. Users: Visual Recommendation of Research Talks with Multiple Dimension of Relevance. *ACM Transactions on Interactive Intelligent Systems* 6, 2 (2016), Article No. 11. https://doi.org/10.1145/2946794

[70] Jesse Vig, Shilad Sen, and John Riedl. 2012. The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM Transactions on Interactive Intelligent Systems* 2, 3 (2012), Article 13.

[71] Liang Wu, Mihajlo Grbovic, and Jundong Li. 2021. Toward User Engagement Optimization in 2D Presentation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, 1047–1055.

[72] Sirui Yao, Yoni Halpern, Nithum Thain, Xuezhi Wang, Kang Lee, Flavien Prost, Ed H. Chi, Jilin Chen, and Alex Beutel. 2020. Measuring Recommender System Effects with Simulated Users. In *FATES 2020 : 2nd Workshop on Fairness, Accountability, Transparency, Ethics and Society on the Web at The Web Conference 2020*. WWW, Online.

[73] Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1512–1520. https://doi.org/10.1145/3394486.3403202

[74] Jianhan Zhu, Jun Hong, and John Hughes. 2001. PageRate: counting Web users' votes. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, Hypertext '01*. ACM, New York, NY, 131–132. http://portal.acm.org/citation.cfm?id=504251

[75] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo Dyna-Q: A Reinforcement Learning Framework for Interactive Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 816–824. https://doi.org/10.1145/3336191.3371801