Toward Better Interactions in Recommender Systems: Cycling and Serpentining Approaches for Top-N Item Lists

Qian Zhao[1], Gediminas Adomavicius[1], F. Maxwell Harper[1], Martijn Willemsen[2], Joseph A. Konstan[1]

[1]University of Minnesota
Minneapolis, MN, USA
{zhaox331, gedas, max, konstan}@umn.edu

[2]Eindhoven University of Technology
Eindhoven, The Netherlands
M.C.Willemsen@tue.nl

ABSTRACT

Current recommender systems often show the same most-highly recommended items again and again ignoring the feedback that users neither rate nor click on those items. We conduct an online field experiment to test two ways of manipulating top-N recommendations with the goal of improving user experience: ***cycling*** the top-N recommendation based on their past presentation and ***serpentining*** the top-N list mixing the best items into later recommendation requests. We find interesting tensions between opt-outs and activities, user perceived accuracy and freshness. *Cycling* within the same session might be a "*love it or hate it*" recommender property because users in it have a higher opt-out rate but engage in more activities. *Cycling across sessions* and *serpentining* increase user activities without significantly affecting opt-out rates. Users perceive more *change* and *freshness* but less *accuracy* and *familiarity*. Combining *cycling* and *serpentining* does not work as well as each individual manipulation separately. These two ways of manipulations on top-N list demonstrate some attractive properties but also call for innovative approaches to overcome their potential costs.

Categories and Subject Descriptors

H.1.2 [**User/machine systems**]: Human factors; H.3.3 [**Information storage and retrieval**]: Retrieval models

Keywords

recommender systems; user study; field experiment

INTRODUCTION

Recommender systems typically are optimized to produce a top-N list reflective of the most-highly recommended items a user has not yet rated. However, there are many reasons to believe that this order may not be the best order to present items to users, either within or across sessions. First, top-N does not consider whether a recommendation has already been displayed to the user before, that is, whether it is fresh vs. potentially stale. Second, presenting the standard top-N list may create an experience where continued exploration results in a sense of finding ever-worse alternatives recommended. In this paper, we explore two alternatives to the standard top-N approach designed to address these concerns. ***Cycling*** recommendations demotes recommended items after they have been viewed several times, while promoting fresher recommendations from the lower portions of the list. ***Serpentining*** displays a "*zig-zag*" order, in which the best recommendations (i.e., the top recommendations from a rating prediction model) are spread across several pages, offering high-quality items on each page as a user continues to explore. Cycling may happen within the same visit or across multiple visits, which we call ***intra-session*** or ***inter-session*** cycling. *Intra-session* cycling creates a more immediate and noticeable change but may cause confusion because potentially interesting recommendations may disappear when a user goes back to the previous page. *Inter-session* cycling is less likely to have this problem but may not be noticeable because users have forgotten what they saw previously.

The high-level research question in this work is whether *cycling* and *serpentining* – as two perspectives of re-examining top-N list – improve user experience. However, we are not trying to optimize a particular user experience. We recognize that different experiences may require different approaches. A situation where a site recommends a single item cannot benefit from *serpentining*. A user who treats the top-N list as a "*to-do*" list, taking the top item each time, would not be served well by *cycling*. Rather, we want to see how these manipulations relate to user experience in the hopes of guiding designers in adopting them, or offering them to users. Similarly to the finding from Ziegler's work [32] that users are willing to accept a certain loss of accuracy in order to have more diverse recommendations, we expect that the perceived accuracy of recommendations may get reduced because of the manipulation; however, we test whether the accuracy reduction may be preferred in exchange for the exposure to a broader and "fresher" set of items.

With this as the goal of our research, we look at multiple metrics and several dimensions of user experience. We recognize that users also have different goals, including

CSCW '17, February 25-March 01, 2017, Portland, OR, USA
 ISBN 978-1-4503-4335-0/17/03…$15.00
DOI: http://dx.doi.org/10.1145/2998181.2998211

those who want to explore deeply and ones who simply want to find an item quickly. For this reason, we look at (a) a variety of user activities, including *engagement measures* (levels of usage) and *success measures* (numbers of items selected) as well as (b) a variety of self-reported reactions, including assessments of *quality*, *freshness*, *usefulness*, etc. We follow the framework proposed by Knijnenburg et al. [12] for user-centric evaluation of recommender systems. Four components of the framework are involved: **OSA** (Objective System Aspects, e.g., recommender manipulations), **SSA** (Subjective System Aspects, i.e., user perceptions on different aspects of the recommender), **EXP** (experience, e.g., the overall perceived usefulness or satisfaction), and **INT** (interaction, i.e., user activities or behavioral data in the recommender). It leads to our research questions *RQ1-RQ3* listed below. We combine SSA, EXP, and INT measurements to better understand user experience. As pointed out by Velsen et al. [31], interpretation of user behavioral data is often troublesome, and they suggest *triangulating* objective behavior data with subjective experience data (which is collected through surveys in our experiment). For example, increased page views could be representative of better (more) user engagement, but it could also mean that users are forced to browse more in order to get useful recommendations. We are concerned that asking survey questions may have an effect on users' activities as well. Therefore, we also design another variation in addition to the above two manipulations: ***delayed*** asking vs. ***non-delayed*** asking. For users in the *delayed asking* condition, we only ask them survey questions after they joined our experiment for certain period of time (one month here) so that we can measure user activities in a setting that is closest to the production environment of an online recommender system for a while (which usually does not have surveys).

- ***RQ1: How do cycling and serpentining recommendations affect user activities? (OSA → INT)***
- ***RQ2: How do cycling and serpentining affect user perceptions and their overall experience with the recommenders? (OSA → SSA and EXP)***
- ***RQ3: How does each user perception aspect contribute to user-perceived usefulness and overall satisfaction? (SSA → EXP)***

RELATED WORK

The cycling approach proposed above creates a distinct type of ***presentation-controlled*** dynamic, as it controls the presentation of a recommended item and cycles it out when it has certain exposure. We use recommender ***dynamic*** here to broadly refer to the change in recommendations. There are many kinds of dynamics in dynamical recommender systems [24]. The most classic ones model users' temporal preference drifting [13, 8, 3]. Rana and Jain [24] classified the dynamics of recommender systems into six categories: temporal changes, online processing, context, novelty, serendipity, and diversity. We review dynamics in recommender systems from a different perspective here. From the literature, dynamics can be achieved with two approaches: *model-based* and *algorithm-based*. Model-based approaches include context-aware recommenders [29, 1] and systems explicitly modeling user preference change [13, 3]. For example, in their work on Context-Aware Recommender Systems (CARS), Adomavicius et al. [1] examined how context can be defined and used in order to create more intelligent recommendations, such as using *pre-filtering* and *post-filtering* strategies with respect to contextual factors. In their classification, contexts can also be dynamic (vs. static), because designers may find previously relevant contexts no longer useful, such as shopping companion. Koren [13] proposed to track user preferences on products along the whole time period in history data sets by explicitly postulating parameters regarding temporal effects and successfully incorporated this idea into two popular recommender techniques: a factorization model [14] and an item-item neighborhood model [25] to improve preference prediction accuracy.

The second approach to achieve dynamics is through *algorithms*, i.e., how to find the optimal solution for the specified model and how frequently to update the model as new data set becomes available. As an example, matrix factorization [14] is a popular technique for recommendation, in which user preferences are modeled with latent factors and learned from user-item interaction matrix. Recognizing that factorizing those interaction data matrices in batch has significant delay compared with the time of receiving feedback from users, online learning or incremental techniques have been proposed and tested for real-time model updating [17, 16]. Most of machine learning based approaches to recommender systems have the incremental processing capability. For example, *learning-to-rank* [5] techniques directly learn the relative ranking of items to a specific user from data, whose dynamics critically depend on the updating latency of the ranking models, i.e., how quickly new available information is fed into the algorithm. Many recommender systems have both *model* and *algorithm* based dynamic perspectives, such as Markov Decision Processes (MDP) based recommenders [26] and contextual bandits [15] in computational advertising. In these techniques, recommendation problem is modeled as a dynamic decision making policy for an agent, and algorithms are designed to search the optimal solutions based on partially available and incrementally gained information such as "like" or "dislike" feedback from users.

However, there is a need for more systematic study of recommender models' and algorithms' *effects on user perceptions and experience*. Change in recommendations is a good thing when users perceive more freshness and less boredom, but also could be confusing when changes are highly unexpected or overly dramatic. In other words, several psychological factors (that may not be directly observable) may be involved in user's decision making and, therefore, a systematic user-centered approach is needed to evaluate their potential involvement. Users' exposure to

recommendations can also be studied by analyzing user actions, following the approaches and ideas from the science of *persuasion* and *marketing*. As Trellis's [30] work showed, advertising exposure has a nonlinear effect, in other words, repetitive exposure is necessary but has diminishing gain. Their and others' results [18] suggested that two to three ad exposures might be optimum. As discussed by Petty and Cacioppo [21] and also suggested by their results, repeating a persuasive communication tends to first increase and then decrease agreement. They proposed a two-stage attitude modification process: repetition enhances a person's ability to process a message in the first stage, and tedium and reactance are elicited by excessive exposures in the second stage. Similarly, this two-stage process might also apply in recommendations. Although CARS [1] adapt recommendations based on users' contextual state, i.e., based on time, mood, or companion(s), there might be contexts that are hard to measure and very sparse data about them is available for each individual user. Therefore, repetitive recommendations may increase the chances that users process the recommendations in relevant contexts. In addition, we study user-perceived boredom and freshness associated with the dynamics through surveys. There is not much research on changing recommendations based on users' past exposure to recommended items. One thread of related research is CTR (Click-Through Rate) estimation in information retrieval [2], where documents with many exposures but no positive feedback from users are downgraded because their estimated CTRs become lower. Recommender systems can also utilize indirect feedback, such as clicks, which would be treated as an implicit *preference* signal [9]. In other words, when focusing on implicit users' feedback in response to displayed recommendations, a recommender can be designed to achieve similar dynamics. We do not use CTR as the primary evaluation approach in our work, because the system studied is not targeted at generating click-throughs, but rather at helping users have better experience with in exploring and finding movies (as measured in a much more holistic, comprehensive manner). Moreover, in our system users can see and rate movies without clicking through to detail pages, so informativeness of clicks as a primary evaluation measure may be limited. However, we do keep track of clicks as one of several indicators of user activities and engagement with the system.

Recommender systems can be evaluated with offline metrics and online field experiments. Offline metrics sometimes make assumptions about online environments. One such important assumption is that the recommendation value decays going from the top to the bottom of a recommended item list. The *nDCG* [10] and *weighted recall* (or *Breese's score* [4]) evaluation metrics, for example, assume exponential decay. We propose to test this assumption, because it may not be optimal to display all best recommendations at the same time. *List-wise optimization* have been shown to improve recommendations [28], which suggests that an optimal list may not be the same as a collection of individually optimal items. Also, it has been shown that, in addition to accuracy, many other properties of a recommender are important aspects of user satisfaction [22, 19, 20, 32, 11], such as diversity, novelty, etc. Pu et al. [22] proposed a user-centric evaluation framework for recommender systems with state-of-art survey designs [23]. Knijnenburg et al. [12] proposed a comprehensive framework taking into account both objective system measurements and subjective user perceptions to explain user experience. We directly apply this framework in evaluating our manipulations. Particularly, they postulate six components and their causal relationships – objective system aspects (OSA), subjective system aspects (SSA), user experience (EXP), user interactions or activities (INT), situational characteristics (SC) and personal characteristics (PC) -- according to Theories of Reasoned Action (TRA) [7]. We use and model the former four components through methods of recording and analyzing user activities and survey responses here. This framework relies strongly on asking users their subjective experience through survey questions. In many examples of this type of studies, users typically interact one time with a system and then evaluate its performance. However, in our current study users can interact with a system over time, i.e., over several sessions. Because of this, we vary the moment of presenting the survey questions, to see if querying the user experience might affect how users interact with the system.

EXPERIMENT DESIGN

To answer our research questions, we conduct a field experiment in MovieLens[1], an online movie information and recommender system used by thousands of real users every month (41,125 movies as of July 2016). Typically, MovieLens users browse pages of movie cards organized in a grid layout, similarly to Netflix or Hulu. They can rate a movie in a five-star rating widget according to their preference for the movie and add a movie into their personal wishlist. They can also click a movie card to transition to another page to see the movie details.

We include users who have at least 15 ratings to make sure that we are testing on users who have a reasonable level of engagement with the system, since most of the active users have more than 15 ratings (as shown in the Results section). We also limit the study to include only users who have at least one session of usage in the system, excluding the current session for reasons explained below.

We invite qualified users to join the experiment through a link displayed on the home page: "*Would you like to experience a new movie recommender named Spirit in MovieLens?*" (*Spirit* is the recommender name we use for all conditions in this study). After clicking the link, users see an

[1] https://movielens.org

informed consent page, which briefly introduces the experiment including information about potential survey requests. If they consent, they are randomly assigned into one of the experimental conditions. Users can opt out of using the experimental recommender at any time by clicking a link at the top right corner which says "*Stop Using the Spirit*". Note that *opt-out* here specifically means stopping using the experimental recommender, not completely dropping out from the experiment. Users are informed that they can contact us through MovieLens to remove their data from the system if they wish to withdraw entirely. This experiment was approved by our institution's Institutional Review Board.

We employ a *between-subjects* 3x2x2 factorial design. The first design factor is *cycling*, which takes three levels -- *no cycling*, *inter-session cycling*, and *intra-session cycling* -- as mentioned in the previous section. What we want to accomplish through *cycling* is to control the amount of exposure a recommended item can have on a user, favoring those items that are least exposed but only after they have been presented certain number of times. This is achieved by re-ranking top-N items first based on the number of previous presentations and then based on the predicted preference from state-of-art algorithms. We use a *presentation* to specifically refer to a movie card display in the grid layout of MovieLens. Instead of directly using the number of presentations, we calculate how many times a movie has been presented to a user, divided by *three* (rounded to the smaller integer), which we call *presentation score*, based on a history of presentation data tracing backwards to one session before a user joins the experiment (this is enabled by our inclusion criterion of the participants, i.e., users who have at least one session before joining). The implication of this is that an item will first be exposed three times before the algorithm starts to downgrade the item's rank in the new list. The predicted preference (i.e., rating) of an item comes from the popular *item-item collaborative filtering algorithm* [25] built on the historical data of user ratings on items in the MovieLens. The top-N list is first sorted by the presentation score ascendingly and then sorted by the predicted ratings descendingly to get the new top-N list. Further, as mentioned before, two types of *cycling – intra-session* and *inter-session* – are designed to have different dynamical extent. For the intra-session type, we cycle the top-N recommendations once every time users go (or go back) to the home page even when it is within the same session. For *inter-session* cycling, we only cycle once when users come to the home page in a new session. We take *240* items as the (top-)N here. It spans *10* pages of movie cards (with each page displaying *24* movies) beyond which there is no manipulation on the recommendation list.

Another design factor, *serpentining*, takes two levels: *true* and *false*. What we want to accomplish here is to have a new list where users can see best items spread in multiple pages. When *serpentining=true*, we re-organize the top-N list based on the original rankings of the recommendations (i.e., 1 through *N*). Specifically, we pick movies intermittently with a constant ranking interval *M* (=4). This is achieved by first reshaping the *N-by-1* list into a *M-by-N/M* (i.e. 4-by-60 here) matrix in a column first order as illustrated in Table 1. Table 1 also gives the page index when users are requesting the *k-th* (k=1 to 10) page of their recommendations. Notice that the 9th and 10th pages span two rows because each row has only 12 movies left. The algorithm can be summarized as *Page-level Column First* and *Item-level Row First* (PCF-IRF) serpentining.

1st	1	5	.. .	93	***5th***	97	101	.. .	189	***9th***	193	197	.. .	237
2nd	2	6	.. .	94	***6th***	98	102	.. .	190	***9th***	194	198	.. .	238
3rd	3	7	.. .	95	***7th***	99	103	.. .	191	***10th***	195	199	.. .	239
4th	4	8	.. .	96	***8th***	100	104	.. .	192	***10th***	196	200	.. .	240

Table 1. *Page-level Column First* and *Item-level Row First* serpentining (PCF-IRF) algorithm illustrated with *(top-)N=240*, *M (number of rows)=4*. *1st, 2nd, … kth* are the page indices. *M* controls how scattered the new top-N list is in the original rankings and also how much change an item's ranking can have after *cycling*).

In the case where both *serpentining* and *cycling* algorithms are enabled (i.e. the interaction between the two), we first apply the serpentining algorithm, then apply a cycling algorithm to the results. However, we only allow the cycling algorithm to affect ordering within columns as shown in Table 1. The goal of this design is to control the freedom of an item's new presentation position after cycling. For example, items in the second column of Table 1 with rankings 1 through 4 can exchange presentation positions through cycling but not with other columns. It means the very best item may only go to the first of the 2nd, 3rd or 4th page. *Serpentining=false* actually is a special case where *M=N*, in which an item can be anywhere between 1 through N and may have dramatic position change, such as going from the 1st page to the 10th page.

The third design factor ***asking*** takes two levels: *delayed asking* and *non-delayed asking*. For *non-delayed* asking condition, we survey users as soon as they have enough interactions (see the measurements for more details) with the experimental recommender. For *delayed* asking condition, we only survey users after they have joined the experiment for at least one month and also have enough interactions.

Measurements

Based on Knijnenburg et al. [12], we measure users' interactions with the system (INT), user perceived subjective system aspects (SSA), and user experience with the recommender (EXP). For INT, the following list of metrics are computed in a fixed period (*half a month* here) of time for *each user*.

- ***optOutRate***: What percentage of users opt out from the experimental recommender?

- ***numSession***: How many times users come to use the recommender?
- ***totalLength***: How long do users stay in the recommender (in seconds), which equals to the sum of all their session lengths?
- ***numPageViews***: How many recommendation pages do users browse? Note that we specifically use ***page views*** to refer to recommendation page (with list of movie cards) browsing, not including movie details page view.
- ***numActions***: How many actions do users do in the recommender? ***Action*** refers to either rating, clicking, or wishlisting.
- ***numRatings***: How many items do users rate?
- ***numInterested***: How many of the actions are clicks or wishlistings, which indicate that users are interested in the specific movies that were displayed?
- ***numInterestedPerPage***: How effectively is each recommendation page capturing user interest so that users click the shown items or add them into the wishlist?

Metric	Survey Question
accuracy	My top-picks match my tastes in movies.
familiarity	I am familiar with many of the movies in my top-picks.
diversity	My top-picks have a diverse selection of movies.
novelty	My top-picks help me discover new movies.
change	I have noticed my top-picks changing.
freshness	I like my top-picks for having new recommendations.
confusion	I get disoriented sometimes by the change of my top-picks.
boredom	I am bored by my top-picks for recommending the same movies.
usefulness	My top-picks help me find interesting movies.
satisfaction	Overall, I am satisfied by my recent top-picks.

Table 2: SSA and EXP metrics and their corresponding survey questions

We measure SSA and EXP by embedding a survey into the recommendation page. To have informative responses from users, we set the minimum amount of experience users are required to have in the experimental recommender; specifically, we survey them after they browse ***more than three recommendation pages*** with lists of movie cards. We invite users to respond to the survey by displaying a survey link together with the prominent *recommendation section* title in the page. If users click the link (which is optional), a survey with 10 Likert-scale questions is expanded as listed in Table 2. The first eight are measuring SSA, and the remaining two are measuring EXP (*usefulness* and *satisfaction*). Metrics for SSA include four classic metrics used in recommender systems literature: perceived *accuracy*, *familiarity*, *diversity*, and *novelty*. The questions are designed referencing Pu et al.'s work on evaluating recommenders through surveys [22, 23]. Because the survey was given while the user interacted with the recommender and to reduce the opt-outs due to a long survey, we chose to implement a short survey that measures each aspect with only one item, rather than using multiple items per question as proposed by Knijnenburg et al. [12]. We also design specific questions pertaining to our manipulation, measuring perceived *change*, *freshness*, *confusion*, and *boredom*. Here the *freshness* question is about the positive aspect of the change, *confusion* asks about the negative aspect of too much change, and *boredom* is about the negative aspect of too little change. After displaying the first survey, we ask a user the second time with the exact same survey one week later (if they come back to the system and browse for another three or more recommendation pages), in case users do not perceive the recommender dynamics when responding initially.

Cycling → Serpentining	no	intra-session	inter-session
true	*serp.*	*serp-intra.*	*serp-inter.*
false	*ctrl.*	*intra.*	*inter.*

Table 3. Condition naming for the interaction between cycling and serpentining factors.

RESULTS

We ran the experiment and collected data from March 22, 2016 until May 14, 2016. During this period of time, 6249 users were active in the site. 5158 users were presented with the invitation link to join the experiment and 1218 clicked the link. Overall, 987 users joined the experiment, with each of the 12 (3x2x2) conditions having around 80 users. In subsequent analysis, we also consider the subsample of 802 users who joined the experiment at least half a month before analysis, with each of the conditions having around 66 users. 103 users responded to the surveys, providing 121 responses, 92 of which are complete across all the survey questions.

To verify the randomization, we conducted an analysis on the participants' activity history before they joined the experiment to make sure users across different conditions were comparable. Specifically, we looked at each user's INT metrics during the half month before joining and did not find significant differences.

		Conditions combining *cycling* and *serpentining*					
		ctrl. (n=148)	**inter. (n=134)**	**intra. (n=129)**	**serp. (n=145)**	**serp-inter. (n=128)**	**serp-intra. (n=118)**
optOutRate		0.094 (0.024)	0.149 (0.030)	**0.217 (0.036) >ctrl. ****	0.117 (0.026)	0.132 (0.029)	0.093 (0.026)
totalLength	***au***	2148 (333)	3107 (506)	**3542 (588) >ctrl. ***	**3147 (493) >ctrl. +**	2688 (448)	2153 (374)
	su	2148 (344)	**3369 (585) >ctrl. +**	**3442 (635) >ctrl. +**	**3314 (543) >ctrl. +**	2647 (466)	2127 (387)
numPageViews	***au***	21.6 (2.09)	**27.3 (2.76) >ctrl. +**	**28.1 (2.90) >ctrl. +**	24.9 (2.43)	19.6 (2.04)	21.3 (2.31)
	su	20.6 (2.06)	**29.5 (3.18) >ctrl. ***	**27.8 (3.19) >ctrl. ***	**26.5 (2.70) >ctrl. +**	18.9 (2.08)	20.5 (2.30)
numActions	***au***	11.9 (1.87)	15.2 (2.49)	**21.7 (3.62) >ctrl. ****	14.5 (2.28)	10.3 (1.74)	15.4 (2.70)
	su	10.5 (1.70)	**16.6 (2.91) >ctrl. +**	**19.5 (3.63) >ctrl. ***	**15.7 (2.60) >ctrl. +**	8.45 (1.51)	**15.9 (2.87) >ctrl. +**
numRatings	***au***	8.28 (1.77)	9.61 (2.16)	**15.8 (3.63) >ctrl. ***	8.68 (1.88)	6.04 (1.39)	11.1 (2.68)
	su	7.35 (1.64)	10.5 (2.55)	**13.6 (3.50) >ctrl. +**	9.70 (2.21)	4.43 (1.09)	11.6 (2.91)
numInterested	***au***	3.70 (0.588)	**5.77 (0.952) >ctrl. +**	**6.08 (1.02) >ctrl. ***	**6.00 (0.951) >ctrl. ***	4.35 (0.740)	4.35 (0.771)
	su	3.21 (0.538)	**6.26 (1.11) >ctrl. ****	**6.15 (1.16) >ctrl. ***	**6.25 (1.04) >ctrl. ****	4.07 (0.742)	4.32 (0.801)
numInterestedPerPage	***au***	0.137 (0.022)	0.148 (0.025)	0.204 (0.036)	0.160 (0.026)	0.168 (0.029)	0.193 (0.035)
	su	0.132 (0.022)	0.153 (0.028)	**0.214 (0.042) >ctrl. +**	0.148 (0.026)	0.156 (0.029)	0.188 (0.036)

Table 4. Results of different conditions for INT metrics. *au* indicates the measurement and effect across all users in that treatment group, including users who opt-out, returning to their default recommender. *su* indicates the measurement and effect for users who retain the experimental recommender in the measured first half month. We include both to estimate the effects both on those who retain the treatment and on the population of users offered the treatment overall. We analyzed users who opt-out separately, and in no measurement did they differ significantly from the control group. *numSessions* (overall mean is 4.59) is not shown because there were no statistically significant differences. See Table 3 for the definition of condition names that combine *cycling* and *serpentining*. The numbers are means with standard errors in the parentheses and only significant comparisons (through *negative binomial regressions*) are marked with significance codes: + (p<0.1), * (p<0.05), ** (p<0.01).

RQ1: OSA → INT. We only consider users who joined the experiment for at least half a month and calculate the metrics for each user's first half month. For users who choose to opt out of the experiment, we exclude activities after the opt-out time. For each metric, we build a *negative binomial regression* model with the recommender factor as shown in Table 3 (six levels) and *asking* factor (two levels) as the predictors. All models are significantly better than their Poisson regression counterparts (i.e., the data is more overdispersed than what a Poisson model assumes). The results are summarized in Table 4. We report results with p-values less than 0.1 here. First, we find that users (including both users who stay and those who opt out) in *intra.* condition have a higher probability of opting out than those in *ctrl.* condition (0.217 vs. 0.094, p=0.005). We do not find significant difference in *numSession* (overall mean is 4.59). On contrary, users in *intra.* and *serp.* condition have higher or marginally higher *totalLength* than users in *ctrl.* condition (3542 vs. 2148, p=0.027; 3147 vs. 2148, p=0.083). The following two metrics *numPageViews* and *numActions* explain why users in *intra.* condition spend more time in the recommender. It shows that *intra.*

	inter.	intra.	serp.	serp-inter.	serp-intra.
satisfaction	-0.698 (0.707)	-0.849 (0.662)	-0.863 (0.768)	-0.482 (0.740)	-0.440 (0.757)
usefulness	-0.793 (0.774)	-1.19 (0.719) +	-1.99 (0.866) *	-1.72 (0.799) *	-1.26 (0.804)
accuracy	-1.72 (0.751) *	-1.94 (0.715) **	-1.51 (0.828) +	-0.477 (0.822)	-1.08 (0.785)
familiarity	-0.731 (0.782)	-1.47 (0.748) *	-2.19 (0.911) *	-0.453 (0.811)	-0.940 (0.850)
diversity	-0.037 (0.737)	0.999 (0.705)	-0.823 (0.843)	0.874 (0.793)	0.413 (0.786)
novelty	-0.449 (0.698)	-0.201 (0.659)	-1.22 (0.779)	-0.777 (0.761)	-0.673 (0.772)
change	2.78 (1.00) **	2.37 (0.868) **	0.668 (0.912)	1.16 (0.869)	2.02 (0.954) *
freshness	0.903 (0.741)	1.63 (0.720) *	0.437 (0.826)	1.60 (0.816) +	1.08 (0.806)
boredom	-0.454 (0.803)	0.319 (0.763)	1.89 (0.966) +	1.08 (0.891)	1.11 (0.871)
confusion	1.87 (0.895) *	1.10 (0.792)	-0.827 (0.963)	0.163 (0.865)	1.28 (0.888)

Table 5. Results of different conditions for SSA and EXP metrics. See Table 3 for the definition of condition names that combine *cycling* and *serpentining*. *ctrl.* condition is the base to compared with in the ordinal regressions. The numbers are coefficients (in log odd-ratio scale) with standard errors in the parentheses. Significance codes: + (p<0.1), * (p<0.05), ** (p<0.01).

condition users have marginally higher *numPageViews* (28.1 vs. 21.6, p=0.065), i.e., they browse more; also these users have higher *numActions* (21.7 vs. 11.9, p=0.008), i.e. they do more actions than *ctrl.* condition users. We find that users in *inter.* condition have marginally higher *numPageViews* than those in *ctrl.* condition as well (27.3 vs. 21.6, p=0.098). The following two metrics *numRatings* and *numInterested* further explain which actions users perform more. Consistently with the overall increase of *numActions* for *intra.* condition users, they not only have higher *numRatings* (15.8 vs. 8.28, p=0.037, i.e. they rate more) but also higher *numInterested* (6.08 vs. 3.70, p=0.031; i.e., they are interested in more recommendations) compared with those in *ctrl.* condition. Users in *serp.* and *inter.* conditions also have higher *numInterested* than *ctrl.* (6.00 vs. 3.70, p=0.031; 5.77 vs. 3.70, p=0.052), which explains why users in *serp.* condition spend more time in the recommender. We also separately analyze the metrics for users who stay. They are consistent with the above results and become more statistically significant. Users in either *inter., intra., serp.* condition have higher *totalLength*, *numPageViews*, *numActions* and *numInterested*. *Intra.* conditions users also have higher *numInterestedPerPage* which means the probability of those users being interested to click or wishlist is higher compared with users in *ctrl.* We do not compare across conditions for dropped out users, because it is highly likely that the user population is different.

We also find some interesting effects on *numActions* and *numRatings* for *ask=non-dela*y (vs. *delay*) condition that are not included in the table due to space limitations. Specifically, users surveyed in a *non-delayed* way have higher *numRatings* (12.2 vs. 7.48, p=0.008, i.e., they rate more) and hence have marginally higher *numActions* (16.6 vs. 12.9, p=0.061) than those surveyed in a delayed way.

RQ2: OSA → SSA and EXP. For simplicity of analysis, we only use the 92 *complete* responses for all survey questions to explore this research question. For each metric, we build an *ordinal regression* (cumulative link mixed effects) model with user identification as the random intercept (as we have more than one survey response for some users).[2] The fixed effects part of the model has the interaction between *cycling* and *serpentining*, and also *asking* factor as the input. All the results are summarized in Table 5. We do not find any significant effects for *asking* and, therefore, it is not included in the table due to space considerations.

First of all, we do not find statistically significant differences between conditions for overall *satisfaction*. However, users in *intra.*, *serp.* and *serp-inter.* conditions perceive the recommendations to have less *usefulness* compared with those in *ctrl.* condition. This EXP level feedback from users can be explained by comparing the individual SSA metrics. For classic metrics, we find users in *inter.*, *intra.* and *serp.* conditions perceive the recommendations to be less accurate than those in *ctrl.* condition. Similarly, users in *intra.* and *serp.* conditions report that they are less familiar with the recommended items.

[2] The number of responses was too low to build a complete structural equation model relating all concepts to each other, as typically used in the Knijnenburg et al. framework [12].

We also analyze the specially designed metrics for our manipulations. First, we notice that users in *inter., intra.* and *serp-intra.* conditions report more perceived *change* compared with *ctrl.* condition. This is by design but reassures us that indeed our manipulations are perceived by the users. Given that users perceive the change, a further question regarding the change is whether users like it. In terms of the the positive aspect of the change, users in *intra.* and *serp-inter.* perceive significantly more *freshness* than users in *ctrl.* condition. Regarding the negative aspects of too much change and too little change, we find users in *serp.* condition perceive more *boredom* than those in *ctrl.* condition. Interestingly, we find users in *inter.* condition report more *confusion* than those in *ctrl.* while users in *intra.* condition do not perceive significantly more *confusion* than *ctrl.* condition.

SSA	EXP	
	usefulness	***satisfaction***
accuracy	**1.10 (0.29) *****	**1.54 (0.31) *****
familiarity	-0.160 (0.24)	**0.807 (0.26) ****
novelty	**1.22 (0.29) *****	**0.661 (0.28) ***
diversity	**0.613 (0.27) ***	**0.569 (0.26) ***
change	-0.204 (0.31)	0.018 (0.28)
confusion	-0.093 (0.28)	**-0.479 (0.28) +**
freshness	**0.498 (0.30) +**	**0.702 (0.29) ***
boredom	**-0.466 (0.26) +**	**-0.778 (0.27)****

Table 6. The coefficients and standard errors (in parentheses) of the ordinal regressions using individual SSA to predict EXP (*usefulness* and *satisfaction*). Significance codes: + (p<0.1), * (p<0.05), ** (p<0.01), * (p<0.001).**

RQ3: SSA → EXP. We are interested in how users perceived SSA on recommendations (particularly the ones we specially designed for our manipulations, i.e., *change*, *freshness*, *boredom*, and *confusion*) affect user EXP (see [12] for the postulated causal relationship between SSA and EXP). To answer this question, we only use all complete survey responses for the 92 users and build two ordinal regression models to predict *usefulness* and *satisfaction* with the individual SSA. Table 6 shows the results with regression coefficients and p-values. From the table, we find that *novelty*, *accuracy*, *diversity*, and *freshness* (marginally significant effect for *freshness*) contribute positively to user perceived *usefulness*, in the descending order of effect sizes. *Boredom* contributes negatively to usefulness although it is marginally significant. While all factors matter for user overall *satisfaction* except perceived *change* (which is reasonable, because it measures perception, not preference), the order of the effect sizes is: *accuracy*, *familiarity*, *boredom*, *freshness*, *novelty*, *diversity* and then *confusion*, where only *confusion* and *boredom* contribute negatively and others contribute positively.

DISCUSSION, LIMITATION AND FUTURE WORK

Here we discuss the main findings about different top-N list manipulation approaches explored in this paper. We find that *intra-session* cycling has an effect of "*scaring*" some users away, while at the same time increasing activity levels for other users, such as browsing more recommendation pages, rating more items, clicking or wishlisting more items, and hence spending more time in the recommender (at least in the first half month we measured). It suggests that this type of manipulation may be a "*love it or hate it*" recommender property. The results obtained via survey questions reveal some potential reasons. In particular, users with this recommender report less perceived *accuracy*, *familiarity*, and marginally less *usefulness*, although they also perceive more *freshness* because of more *change*. Thus, changing recommendations in the same session attracts more user activities but may increase the risk of churning. We hypothesize that the following aspects might be relevant with respect to observed effects and future extensions. First, the platform we use does not have actual item consumption capabilities built in, i.e., users use this movie recommender mainly as a tool to find interesting movies but do not actually watch movies on the site. The dynamics may be quite different in platforms with consumption, because users can proceed with item consumption directly after a recommendation instead of speculating or processing the recommendation as a piece of information to be used later. The increasing effect of user activities should be interesting to system designers, but further study is needed to explore to what extent this effect generalizes to platforms with built-in consumption. Second, cycling 240 items (i.e., the value of N in our top-N recommendations) in our study may represent too big of a range for some users. They may experience dramatic accuracy degradation after cycling for a while, which could contribute to their opt-out. Thus, testing the cycling approach with smaller values of N in different platforms also constitutes an interesting topic for future research.

Inter-session cycling and *serpentining* are the two best-performing conditions in our experiment, considering both opt-out rate and user activities. They do not have a significant effect of "*scaring*" users away, especially the serpentining approach. At the same time, both of them increase user activities such as clicking or wishlisting, especially for users who stay (i.e., do not opt out). The results also show a trend that, in these conditions, users who stay are more active, while users who stay in the control condition are less active compared with those who opt out. This suggests that we are able to retain more active users through our top-N list manipulations. However, we also want to point out that users with *inter-session-cycled* recommendations report less

accuracy, more *change*, and also more *confusion*. Users in a *serpentining* recommender also report less *accuracy*, *familiarity*, and more *boredom*. Interestingly, users in *inter-session cycling* instead of *intra-session cycling* perceive more *confusion* than those in the *control* condition. This might result from the fact that users perceive the change of recommendations but cannot connect the change with their own previous activities when they come back to the system in the next visit. This suggests that users demand at least certain extent of control (or sense of control) in using a recommender system. They expect the recommendations to change based on their taste or at least what they tell about their taste to the recommender.

We see interesting interactions between *cycling* and *serpentining*. *Serpentining* mitigates the negative effects of *cycling*, such as opt-out rate for *intra-session*, reduction in *accuracy*, and increased *confusion* for *inter-session*. However, *serpentining* also reduces the positive effects of cycling, such as increased user activities and improved *freshness* for *intra-session*. One exception is that the effect of *inter-session* cycling with *serpentining* on perceived *freshness* is positive. According to these results, it seems that combining the two manipulations makes things too complicated for users to build a mental model on how the recommender is working.

Although the interface of MovieLens has a grid-view layout, we believe that these approaches are generalizable to other layouts, such as lists. Even if a list does not have pagination, our algorithms can be adapted by using the top-N as the length of the list, which can be *serpentined* and *cycled*, although such manipulations might be more apparent from the user's perspective (e.g., if the list is short).

We would like to note to two limitations in our study: self selection bias and uncertainty about longer term effects. First, our analysis shows that users who were qualified for the experiment but chose not join were significantly less active than users who joined the experiment. Second, the duration of this experiment does not permit us to draw conclusions about longer-term usage patterns, either for those to retain *serpentined* and/or *cycled* recommendations or from those who experience them but opt out. Studies of longer-term effects represent an interesting direction for future work.

The results overall suggest that *cycling* and *serpentining* of the top-N list have certain attractive properties, but future work is needed to design and test approaches that can increase *freshness* and even *novelty* without compromising *accuracy*, because we show that all of these aspects contribute positively to user-perceived *usefulness* and overall *satisfaction*. The use of online machine learning algorithms that can combine different types of user actions as real-time feedback [15] could be a promising direction. However, we demonstrate that objective user activity measures are not enough to comprehensively evaluate whether a recommender is better. For example, if we only focus on optimizing for user actions, there is a good chance that users might not be satisfied by what they have to do and ultimately opt out or churn from the system ("*the time spent is not worth the effort*"), even though activity metrics may seem to be positive. We also consider it necessary to conduct more detailed studies by inviting users into the lab, recording them using the system, and interviewing them in order to fully understand the effects. In addition to the presented results, we also find users using the recommender less often but reporting higher perceived accuracy. It suggests that an ideal recommender should be there to *assist* but not *stand in the way* to *consumption*, because most recommenders only provide access to the actual service being consumed. As suggested by Knijnenburg et al. [12] and McNee et al. [19], more future research on evaluating recommender systems from both system and user perspectives is desired.

CONCLUSION

We conduct an online field experiment to test two perspectives of rethinking top-N recommendations: *cycling*, i.e., the reranking of items in the top-N recommendation list based on user's past exposure to these items, and *serpentining*, i.e., the reranking of top-N item list by mixing the best-predicted items into later recommendation requests. We find interesting tensions between opt-outs and activities, user perceived accuracy and freshness. *Intra-session cycling* might be a "*love it or hate it*" recommender property, because users in it have a higher opt-out rate, but also engage in more activities such as page views, ratings, clicks and wishlistings, especially for those who stay. *Inter-session cycling* and *serpentining* increase activity without significantly increasing opt-out rate. Users perceive more *change* and *freshness* on *cycled* recommendations and less *accuracy*, *familiarity* on both *cycled* and *serpentined* recommendations. Combining *cycling* and *serpentining* does not work as well as each individual manipulation. These two manipulations on top-N list demonstrate some attractive properties in various dimensions but also call for new innovative approaches to further overcome their potential costs.

ACKNOWLEDGEMENTS

This work was supported by NSF IIS-1319382. We thank all the MovieLens users who participated in our study.

REFERENCES

1. G. Adomavicius, and Tuzhilin, A., 2011. Context-aware recommender systems. In Recommender systems handbook (pp. 217-253). *Springer* US.
2. E. Agichtein, Brill, E. and Dumais, S., 2006, August. Improving web search ranking by incorporating user behavior information. In *SIGIR'06* (pp. 19-26). ACM.
3. L. Baltrunas, and Amatriain, X., 2009, October. Towards time-dependant recommendation based on implicit feedback. In *CARS'09*.
4. J.S. Breese, Heckerman, D. and Kadie, C., 1998, July. Empirical analysis of predictive algorithms for

collaborative filtering. In *UAI'98* (pp. 43-52). Morgan Kaufmann Publishers Inc..

5. C. Burges, Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. and Hullender, G., 2005, August. Learning to rank using gradient descent. In *ICML'05* (pp. 89-96). ACM.
6. L. Chen, Wu, W. and He, L., 2013, April. How personality influences users' needs for recommendation diversity?. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems* (pp. 829-834). ACM.
7. M. Fishbein, and Ajzen, I., 1977. Belief, attitude, intention, and behavior: An introduction to theory and research.
8. W. Hong, Li, L. and Li, T., 2012. Product recommendation with temporal dynamics. *Expert Systems with Applications*, 39(16), pp.12398-12406.
9. Y. Hu, Koren, Y. and Volinsky, C., 2008, December. Collaborative filtering for implicit feedback datasets. In Data Mining, 2008. *ICDM'08*. Eighth IEEE International Conference on (pp. 263-272). IEEE.
10. K. Järvelin, and Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. *TOIS'02*, 20(4), pp.422-446.
11. K. Kapoor, Subbian, K., Srivastava, J. and Schrater, P., 2015, February. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *ICWSDM'15* (pp. 233-242). ACM.
12. B.P. Knijnenburg, Willemsen, M.C., Gantner, Z., Soncu, H. and Newell, C., 2012. Explaining the user experience of recommender systems. *UMUAI'12*, 22(4-5), pp.441-504.
13. Y. Koren, 2010. Collaborative filtering with temporal dynamics. *Communications of the ACM*,53(4), pp.89-97.
14. Y. Koren, Bell, R. and Volinsky, C., 2009. Matrix factorization techniques for recommender systems. *Computer*, (8), pp.30-37.
15. T. Lu, Pál, D. and Pál, M., 2010. Contextual multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics* (pp. 485-492).
16. X. Luo, Xia, Y. and Zhu, Q., 2012. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, 27, pp.271-280.
17. J. Mairal, Bach, F., Ponce, J. and Sapiro, G., 2009, June. Online dictionary learning for sparse coding. In *ICML'09* (pp. 689-696). ACM.
18. Colin McDonal, 1971. "What Is the Short-Term Effect of Advertising?" *Special Report* No. 71-142 (February). Marketing Science Institute.
19. S.M. McNee, Riedl, J. and Konstan, J.A., 2006, April. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI EA'06* (pp. 1097-1101). ACM.
20. S.M. McNee, Riedl, J. and Konstan, J.A., 2006, April. Making recommendations better: an analytic model for human-recommender interaction. In *CHI EA'06* (pp. 1103-1108). ACM.
21. R.E. Petty, and Cacioppo, J.T., 1986. The elaboration likelihood model of persuasion (pp. 1-24). *Springer* New York.
22. P. Pu, Chen, L. and Hu, R., 2011, October. A user-centric evaluation framework for recommender systems. In *RecSys'11* (pp. 157-164). ACM.
23. P. Pu, Chen, L. and Hu, R., 2012. Evaluating recommender systems from the user's perspective: survey of the state of the art. *UMUAI'12*, 22(4-5), pp.317-355.
24. C. Rana, and Jain, S.K., 2015. A study of the dynamic features of recommender systems. *Artificial Intelligence Review*, 43(1), pp.141-153.
25. B. Sarwar, Karypis, G., Konstan, J. and Riedl, J., 2001, April. Item-based collaborative filtering recommendation algorithms. In *WWW'01* (pp. 285-295). ACM.
26. G. Shani, Brafman, R.I. and Heckerman, D., 2002, August. An MDP-based recommender system. In *UAI'02* (pp. 453-460). Morgan Kaufmann Publishers Inc.
27. S. Shen, Hu, B., Chen, W. and Yang, Q., 2012, February. Personalized click model through collaborative filtering. In *WSDM'12* (pp. 323-332). ACM.
28. Y. Shi, Larson, M. and Hanjalic, A., 2010, September. List-wise learning to rank with matrix factorization for collaborative filtering. In *RecSys'10* (pp. 269-272). ACM.
29. N. Srivastava, and Schrater, P., 2015. Learning What to Want: Context-Sensitive Preference Learning. *PloS one*, 10(10), p.e0141129.
30. G.J. Tellis, 1988. Advertising exposure, loyalty, and brand purchase: A two-stage model of choice. *Journal of marketing research*, pp.134-144.
31. Van Velsen, L., Van Der Geest, T., Klaassen, R. and Steehouder, M., 2008. User-centered evaluation of adaptive and adaptable systems: a literature review. *The knowledge engineering review*, 23(03), pp.261-281.
32. C.N. Ziegler, McNee, S.M., Konstan, J.A. and Lausen, G., 2005, May. Improving recommendation lists through topic diversification. In *WWW'05* (pp. 22-32). ACM.