Interpreting User Inaction in Recommender Systems

Qian Zhao
University of Minnesota
Minneapolis, United States
zhaox331@umn.edu

Martijn C. Willemsen
Eindhoven University of Technology
Eindhoven, The Netherlands
Jheronimus Academy of Data Science
’s-Hertogenbosch, The Netherlands
M.C.Willemsen@tue.nl

Gediminas Adomavicius
F. Maxwell Harper
Joseph A. Konstan
University of Minnesota
Minneapolis, United States
gedas,max,konstan@umn.edu

ABSTRACT

Temporally, users browse and interact with items in recommender systems. However, for most systems, the majority of the displayed items do not elicit any action from users. In other words, the user-system interaction process includes three aspects: browsing, action, and *inaction*. Prior recommender systems literature has focused more on actions than on browsing or inaction. In this work, we deployed a field survey in a live movie recommender system to interpret what inaction means from both the user’s and the system’s perspective, guided by psychological theories of human decision making. We further systematically study factors to infer the reasons of user inaction and demonstrate with offline data sets that this descriptive and predictive inaction model can provide benefits for recommender systems in terms of both action prediction and recommendation timing.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

user inaction; decision making; decision field theory

ACM Reference Format:
Qian Zhao, Martijn C. Willemsen, Gediminas Adomavicius, F. Maxwell Harper, and Joseph A. Konstan. 2018. Interpreting User Inaction in Recommender Systems. In *Twelfth ACM Conference on Recommender Systems (RecSys ’18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240323.3240366

1 INTRODUCTION

Imagine one of your friends asked for a restaurant recommendation. You told her about a sushi restaurant nearby and she did not end up going there in the next week. What can we say about her preferences for restaurants and the reason why she did not go? And, if she comes for more recommendations after a week, would you recommend the same restaurant again to her? In the beginning you might ask her for a reason, but as time goes by, you might be able to learn that if she crinkled her nose and it was the second time you recommended the restaurant, she probably did not like the recommendation and you’d better stop recommending that one. On the other hand, if she looked upwards trying to remember and it was the first time you recommended, you might want to recommend it again since it’s possible that she was interested but did not pay enough attention.

This type of scenarios happens similarly in online recommender systems, where users systematically browse items and decide to do something or not with the recommendations. In a typical online recommender interface, *e.g.*, the interface of Netflix, Amazon, or Youtube, grids or lists of recommendations are displayed to users once per page view. If users rate or consume (*e.g.*, watch or purchase) one of the items, the system learns from this explicit evaluative rating feedback or implicit behavioral action feedback to make better future recommendations. For example, this feedback can be incorporated into (contextual) preference models to estimate what users prefer in general or in that specific context [1, 17]. However, among all that have been displayed, the majority of recommendations do not elicit any actions from users. We refer to this case as user *inaction*.

Intuitively, displaying recommendations triggered by user browsing affects user perception and experience with the system, and this should include both action and inaction. Just as actions provide feedback about user preferences, so do inactions, and this should be accounted for in the (contextual) preference models. For instance, a recommender that keeps recommending the same item again and again while ignoring user inaction feedback might not engage the user. On the other hand, a recommender that forgets what has been shown and keeps changing its recommendations based on user (in)action feedback could be confusing when the user is not able to retrieve a previously displayed and interesting recommendation, which the user has not yet had a chance to further explore.

Prior work in recommender systems has mainly focused on studying action rather than inaction [14], partially because inaction data is more ambiguous than action: inaction can (just like action) represent a deliberate decision, but can also result from insufficient attention. Moreover, to understand the reasons for inaction, we need access to real users and their browsing activity data to better understand what reasons there are for inaction, and to build models to predict the type of inaction. With access to a live movie recommender system, we set out to answer the following research questions regarding user inaction by deploying a field survey about items not acted upon, analyzing and modeling the survey responses combined with a year of user browsing and interaction logs.

- **RQ1:** *What are the different categories of reasons for user inaction?*

RecSys ’18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240366

- **RQ2:** *How do different categories of user inaction affect user future recommendation preferences for items not acted upon?*
- **RQ3:** *How well can we infer or classify the categories of user inaction from user log data?*
- **RQ4:** *Can we improve recommender systems utilizing a user inaction model based on our earlier findings?*

To answer RQ1, we examined several behavioral decision making theories (notably, Decision Field Theory by Busemeyer *et al.* [3] and the ECHO model by Guo *et al.* [10]) to come up with seven major categories of reasons for user inaction to drive our field survey design. In answering RQ2, we found that users demonstrated significantly different preferences regarding the future recommendation of inaction items that belong to different inaction categories. We then moved to RQ3, for which we investigated factors from user log data that might be predictive for inferring the category or class of the inaction recommendation. Among the significantly predictive factors, we observe some interesting but intuitive effects on the inaction class probabilities that we can infer. Finally, we have some evidence for RQ4, showing that taking into account the best inaction model's output, we can improve user action prediction and potentially the timing of the recommendation. For example, if the model predicts, for a previously displayed inaction recommendation, that there is a high probability that the user is interested in it in the future but not now, we can delay this recommendation to the next session.

Together our results show that user inaction is important to understand before making decisions on the future recommendation strategies of previously displayed items. The categories of user inaction can to some extend be inferred from user log data which can be further used to improve recommendations. In what follows we will first discuss related work, before discussing in detail the methods we used to answer our four research questions.

2 RELATED WORK

Collaborative filtering algorithms *e.g.*, matrix factorization techniques [17] have been applied on user implicit feedback data, for instance, by treating the values of the observed acted-upon (*e.g.*, purchased, watched) items of a user as ones or positives while other unobserved items as zeros or negatives [14]. These values are, however, associated with uncertainty scores to represent that those feedbacks are not explicitly given by users and hence inherently uncertain.

Yang *et al.* [25] proposed the model of collaborative competitive filtering to reflect the fact that users make a decision of picking and acting on one item taking into account the competition of other context items displayed together. Lee *et al.* [18] studied how to estimate and utilize discounting functions of previous impressions (displays) to improve the conversion rate of recommendations, *i.e.*, based on how many times an item had been displayed and the last time the item was displayed to weight (downwards in the work because of the hypothesis that inaction tends to be negative feedback) the normal recommendation score of the item. This is similar to the cycling approach proposed by Zhao *et al.* [26] where the more exposed items are cycled to the bottom of the top-N list. They demonstrated an effect of increased user engagement, although they also observed negatively affected subjective user perception because of this manipulation.

In information retrieval (IR), Joachims *et al.* [15] examined the reliability of click-through feedback data using eye tracking and explicit relevance judgment. They concluded that clicks are informative but suffer from a position bias, because of the search results' presentation in a list layout. Following these findings, various user attention and browsing models are proposed to account for the bias in learning algorithms [5–7, 23] for CTR estimation in IR. Recommender systems that rely on implicit feedback [14] could suffer from position bias as well, as demonstrated by Hofmann *et al.* [13] in simulated experiments. The user browsing model could be substantially different for recommender systems compared with IR because modern recommender interfaces are typically grid-based [27].

Decision making researchers have developed normative and behavioral theories to explain human decision making processes [8]. Human behavioral decision making has two fundamental properties: determinism vs. probabilism (variability of preferences), statics versus dynamics (preference strength and deliberation time) [3]. Decision field theory (DFT) [2, 3] is a theory that takes into account variability of preferences using a dynamic computational model which has been shown to account for many prominent behavioral decision phenomena, such as context effects in which an additional third alternative influences the relative preference of two other alternatives. The theory postulates a temporal comparative mental decision process of people when faced with several options and the accumulative preference of each option (called *valence*) dynamically changes while the decision maker is paying attention to different aspects of the options that are important for the decision. Once the valence reaches certain threshold, a decision will be made, *i.e.*, it is observed that one of the options is chosen. DFT provides a framework for us to formalize the attentional and competitive factors that might affect user behaviors seeing a page of recommendations.

Other similar theories, such as the ECHO model [10], extend this work with a special contributing factor, called the external driver, representing the goal to make a decision. It suggests that context, tasks, and goals influence attentional processes in a dynamic way while users are browsing a page of recommendations. What users attend to (or not) reflects how they compare between options and part of the mental decision processes. In other words, action and inaction tells us more about the underlying preferences and might allow to improve recommendations when modeled and taken into account.

Recommender systems can be evaluated with offline metrics and online experiments. Widely used offline metrics [11] include Precision, Recall, Mean Average Precision (MAP), Area Under the ROC Curve (AUC), Mean Average Error (MAE), and Root Mean Squared Error (RMSE). Unfortunately, these offline metrics have to make assumptions about online environments, *e.g.*, assuming recommendation is a static ranking task to best recover or predict what users do in a held-out (future) part of the observed data sets. These metrics are limited because, *e.g.*, they cannot capture the dynamic interactive nature of how a recommender system is being perceived and used by people, like browsing page by page, going back and forth to compare or get more information to make decisions. As pointed out by McNee *et al.* [19], offline recommendation accuracy on its own often is not a sufficient indicator of recommendation quality, and further work by Knijnenburg *et al.* [16] and Pu *et al.* [20] proposed user-centric frameworks and evaluation metrics to answer a rich set of questions around user experience in recommender systems.

Inspired by the theories of behavioral decision making and following the user-centric approach, we set out to interpret user inaction in recommender systems from the perspective of understanding and improving user experience.

3 RESEARCH PLATFORM

We conducted our research on a live movie recommender system MovieLens (https://movielens.org). MovieLens is a movie information site in which users can browse movie information and get personalized movie recommendations. It has thousands of active users every month. In the *front* (or home) page of MovieLens, there are several sections displaying movies according to different criteria (*e.g.*, recent releases, most popular etc.). The top first section is *top-picks*, *i.e.*, according to the criteria of recommendation scores generated by recommendation algorithms. This section has eight movie cards horizontally displayed in standard PC screens (*i.e.*, one row with eight columns). Users can click a "see more" link besides the section to see more top picks in *explore* pages. Each *explore* page is organized in a three-by-eight grid (*i.e.*, 24 movie cards) and users can browse them page by page.

Each movie card enables three major features: rating, clicking, and adding into the wishlist. Users can enter ratings through a five-star rating widget (half-star increments) under the movie card to tell the system his or her preference for the movie (this can help the system make better future recommendations). Users can click a movie card and transition to a page with its detailed information (*e.g.*, the plot, cast, trailers). If a user wants to collect the movie for later watching, he or she can add the movie into a personal wishlist by clicking on a button in the movie card.

4 DATA COLLECTION

In order to collect survey data on user inaction, we summarized seven major categories of reasons for user inaction according to the postulated temporal decision making process from Decision Field Theory (DFT), [2], adapted to better fit the specific domain of the system.

People normally do not watch most movies multiple times, *i.e.*, the re-consumption of movies could greatly affect whether users interact with a movie recommendation or not. Some domains, however, see frequent re-consumption, *e.g.*, online grocery stores or music streaming services. From another perspective, whether a user has watched (or consumed) the movie before suggests that the user has a certain (highest especially right after the consumption) level of familiarity. A potentially important factor that contributes user inaction is lack of attention, as suggested by Zhao *et al.* 's eye tracking work [27] on grid-based interfaces (our survey in this work also supports this observation).

When a number of recommendations are displayed in one page view, users probably will not pay attention to all of them especially when the item is displayed in a non-prominent position, *e.g.*, the right bottom corner in a grid-based interface [27]. Even if a user does pay attention to an item, the user might prefer other alternatives that are displayed together or the user needs more information about the item to make a decision or just to find out whether it is actually less preferred compared with others. It could also be that the recommendation does not fit the user's movie taste, or that the user is looking for movies to watch with others so that additional constraints must be met. These scenarios reflect the effect of context described by the ECHO model [10].

We designed our field survey to be dynamically adaptive with multiple steps of questions for users to answer. Since our inaction interpretation derives from decision making processes, we excluded scenarios where users are not obviously making decisions based on the recommendations, *e.g.*, rating a previously watched movie or just browsing movie information. In addition, depending on previous answers to some of our survey questions, certain questions may not make sense. For example, if a user did not notice a recommendation, it would not be a valid follow-up question to ask why the user did not interact with it.

When a user goes to the *explore* page with 24 top picks and then transitions away from that page, we randomly picked one movie that was displayed on that page but not acted upon by the user (*action* includes rating, clicking, and adding into the wishlist, but excludes mouse hovering). If conditions to survey the user were satisfied (the user was asked fewer than four times before, and the time of last asking was more than one week ago), a survey was popped up to ask the user the following questions organized according to the flow shown in Figure 1. The order of the options to each question was randomly flipped (excluding the free text box) to avoid position bias. The short names in the parentheses were not displayed but are included here for reference purposes. They also represent how multiple options are sometimes merged to make it easier for modeling, prediction and analysis (*i.e.*, NotNoticed in "notice", PastMonth/PastYear/YearsAgo in "when", Maybe in "future"). Users have the option of checking "don't ask me again" which has an associated message: "if you are under 18 years old, please check this option and dismiss the survey". This study was approved by the Institutional Review Board at the University of Minnesota.

- What was the primary reason you came to the MovieLens Top Picks today?(*reason*)
  - to find a movie to watch now or soon, probably by myself (FindForSelf)
  - to find a movie to watch now or soon, probably with someone else (FindWithOthers)
  - to build a list of movies to watch in the future (BuildList)
  - to browse movies without any specific plan to watch any of them in the near future (JustBrowse)
  - to find movies I've already seen to rate them (RateWatched)
  - other (free text) (FreeText)
- Did you happen to notice whether we displayed a movie recommendation XXX (1993) in the previous Top Picks page? (*notice*)
  - Yes, I noticed it (Noticed)
  - No, I didn't notice this movie being recommended (NotNoticed)

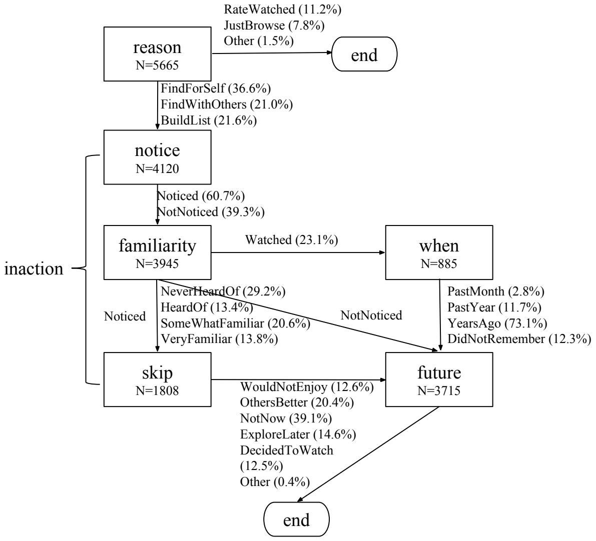

Figure 1: The flow of the inaction survey illustrating how the questions were asked. N is the number of responses to the corresponding question. The ratios are the proportions of options within those response.

  - I don't think it was displayed, but I would have noticed it (NotNoticed)
- How familiar are you with this movie XXX(1933)? (*familiarity*)
  - Never heard of it (NeverHeardOf)
  - Heard of its name but don't know what it's about (HeardOf)
  - Somewhat familiar with it but have not watched it (SomeWhatFamiliar)
  - Very familiar with it but have not watched it (VeryFamiliar)
  - I've watched it (Watched)
- When did you watch this movie XXX(1933) last time? (*when*)
  - Past week (PastMonth)
  - Past month (PastMonth)
  - 1-6 months (PastYear)
  - 6-12 months (PastYear)
  - 1-3 years (YearsAgo)
  - > 3 years ago (YearsAgo)
  - Don't remember (DidNotRemember)
- We noticed that you didn't interact with the card for movie XXXX (*i.e.*, you didn't wishlist, or click to see details)? Which best describes the reason why you didn't? (*skip*)
  - I already decided that I might watch it. (DecidedToWatch)
  - I was planning to click on this movie to explore it later; I just hadn't done so yet. (ExploreLater)
  - There were other movies recommended that seemed more interesting to me at the moment. (OthersBetter)
  - While I might be interested in this movie in the future, it isn't what I'm looking for right now. (NotNow)
  - I'm pretty sure I wouldn't enjoy this movie. (WouldNotEnjoy)
  - Other (free text) (FreeText)
- Should MovieLens continue recommending this movie to you in the future (until you rate it, of course)? (*future*)
  - Yes, definitely (Definitely)
  - Sometimes, but not always (Maybe)
  - Not now, but it would be nice to see it recommended again after some weeks or months (Maybe)
  - No, I'd rather get other recommendations (RatherNot)

From the survey question design, the seven possible categories of user *inaction* are labeled as *NotNoticed, WouldNotEnjoy, NotNow, OthersBetter, ExploreLater, DecidedToWatch, Watched*, which integrates the questions of "notice", "familiarity" and "skip" (note that Watched here only includes those inaction items that are Noticed). We launched the survey on July 28, 2017 and collected user responses until March 21, 2018. 3,206 users gave 3,923 responses for which the user inaction category can be determined.

Along with the survey data, we also have user interaction logs in the system from Jan. 1, 2017 to March 21, 2018. These include 53M movie displays browsed by 25K users with information regarding how they were displayed (*e.g.*, position in the interface and how long the page dwell time was etc.), 1.6M ratings, 369K clicks, 167K wishlist additions, 2.8M hovers (hovering is only logged when the accumulative hovering time on the movie card is longer than one second within the page view) by those users.

5 INTERPRETING USER INACTION

RQ1: What are the different categories of reasons for user inaction? Figure 1 illustrates the distribution of user responses to the survey questions. If we only consider responses in the seven inaction categories from the figure, we found that 38.6% of inaction recommendations were because of lacking attention (NotNoticed). 18.2% were because of lacking the right context (NotNow). 14.6% had already been consumed by the users (Watched), which were still recommended by the system because of lacking consumption records of the users. 9.5% were because of the effects of competition (OthersBetter). 5.8% did not match the user's taste (WouldNotEnjoy). 6.9% needed exploration later for more information to make a decision (ExploreLater). 5.8% had already reached the user's acceptance decision (DecidedToWatch) after that page view although it is an inaction recommendation. Lastly, outside of the options we provided for the "skip" question, we had 0.25% free-text responses. These numbers suggest that simply treating inaction as a signal of negative feedback (or simply ignoring the inaction feedback) could be problematic. Particularly, it points out that the effects of the two most important inaction factors – attention and context – need to be incorporated into the design of recommender models or the presentation of top-N recommendations.

6 FUTURE RECOMMENDATION

In this section, we answer *"RQ2: How do different categories of user inaction affect user future recommendation preference?"* to demonstrate the significance of distinguishing different categories of user inaction. Figure 2 illustrates the distribution of future recommendation preferences for different user inaction categories. We conducted pairwise comparisons through six ordinal regression (specifically, mixed-effects cumulative link) models, assuming that "future" question has three ordinal levels: RatherNot, Maybe, Definitely (as mentioned previously, MaybeLater and Sometimes are merged into one level: Maybe), treating "inaction" as the fixed effect and user ID as the random intercept, varying the baseline condition of "inaction". To control false discovery, we employed Bonferroni correction [12] (effective significance p-value threshold is 0.0083; note the number of models built is six). The overall conclusion is that future recommendation preference can be statistically and substantially different for different inaction categories. Specifically, there is a preferred order of future recommendation for the seven inaction categories. *From the least to the most preferred in future recommendation, the order of inaction categories is WouldNotEnjoy < Watched < NotNotice < NotNow or Others Better < ExploreLater or DecidedToWatch.* Note that movies with an inaction reason of DecidedToWatch are similarly preferred as ExploreLater, and users prefer being recommended inaction movies that they did not notice over ones that they noticed but had already watched.

As described in the survey design section, we also asked questions regarding "reason" (specific contexts), "familiarity", and "when" to watch. Figure 1 shows that the majority of user visits to the *explore* page top-picks are for finding movies to watch, although the context of finding the movie to watch may not be only for the user (*i.e.*, with others) or for immediate consumption.

We analyzed how "reason", "familiarity", "when" to watch might affect users' future recommendation preference by building similar ordinal regression models. We found that FindWithOthers has a significant negative effect on future recommendation preference compared with FindForSelf (coef.=-0.181, std.=0.090, p=0.044). It suggests that when users come to the recommender to find a movie to watch with others, the movies that they browse generally do not reflect their own preference and hence users prefer the system not to recommend these movies in future. For "familiarity", we found that Watched has a significant negative effect on future recommendation preference compared with not Watched yet (p<0.001) but we did not see significant differences among the cases from NeverHeardOf to VeryFamiliar. For different "when" options, we did not see significant differences either. However, we observe a trend that suggests users may be more likely to want to see a watched movie recommended in future when it was watched in the past year compared with the one watched either very recently or very far away in time.

7 CLASSIFYING USER INACTION

The previous section shows that different categories of user inaction significantly affect the future recommendation preference of users. However, we are not able to gather data about user inaction on each of the displayed recommendations in the system. One possible alternative is to build classification models to predict, which is what our RQ3 is about: *How well can we infer or classify the categories of user inaction?* In order to answer this research question, we temporally split the survey data and system logs into two subsets. We used the subset of survey data and system logs before Feb. 1, 2018 (*i.e.*, Jan. 1, 2017 to Jan. 31, 2018) as the *training* data (around 90% of system logs, 80% of survey data). We used the remaining

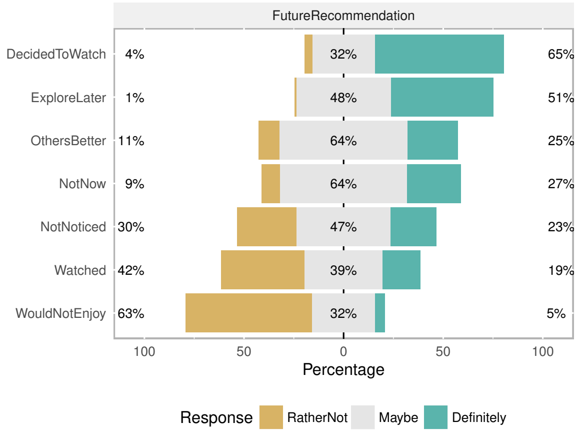

Figure 2: The distribution of future recommendation preference for different user inaction categories. The order of preference (significant after Bonferroni correction, p<0083) is *WouldNotEnjoy* < *Watched* < *NotNotice* < *NotNow or Others Better* < *ExploreLater or DecidedToWatch.*

subset of survey data and system logs (*i.e.*, Feb. 1, 2018 to Mar. 21, 2018) as the *testing* or *evaluation* data.

The problem of inferring the category of user inaction can be formalized as a 7-class classification problem. Note that this is not predicting user inaction before a page of recommendations are displayed, but is inferring user inaction reason after observing how a page of recommendations are displayed and interacted with by the user. This classification model can be used for recommendation when we want to know whether we should re-recommend an item that has been previously displayed to the user before, which will be described in the last section. For now, we focus on answering RQ3.

We hypothesize that predicting potential actions users might perform on recommendations might be useful in inferring user inaction. For instance, estimated probability of displaying (*predDisplay*) a movie by the system might signify the probability of ExploreLater. Estimated user preference (*predRating*) might signify the probability of WouldNotEnjoy. Estimated probability of rating a movie might (*predRate*) signify the probability of Watched. Estimated probability of clicking a movie (*predClick*) might signify the probability of ExploreLater. Lastly, estimated probability of adding into the wishlist (*predWishlist*) might signify the probability of DecidedToWatch. Therefore, we first built five models (referred to as *sub-models*) to generate these predictions through the classical matrix factorization technique [17] (latent factor dimension is 32). These action models (except rating value prediction which is a regression problem) only have positive items observed (*i.e.*, what movies were displayed, rated, clicked or added) for which we need to sample negative items. For each item that was acted upon, we randomly sampled 2K items from the whole item space (≈45K) excluding the acted-upon item after which the classical matrix factorization technique can be applied (rating prediction uses L2-norm loss while action prediction uses binary logistic loss). The accuracy of these models is MAE=1.24, RMSE=1.51 for rating value prediction, Precision@1=0.512, 0.166, 0.018, 0.013 for predicting

Table 1: The confusion matrix of the inaction model (rows are the predicted classes and columns are the actual classes) and the accuracy in terms of AUC for each class (binary classification of one vs. others using the probabilistic output of the 7-class classification model).

Class	Watched	OthersBetter	DecidedToWatch	NotNoticed	NotNow	WouldNotEnjoy	ExploreLater	**AUC**
Watched	**96**	8	4	36	8	7	4	0.799
OthersBetter	0	**0**	0	0	0	0	0	0.720
DecidedToWatch	0	0	**0**	0	0	0	0	0.702
NotNoticed	59	29	21	**217**	80	22	27	0.696
NotNow	4	14	9	10	**16**	3	4	0.676
WouldNotEnjoy	0	0	0	0	0	**0**	0	0.621
ExploreLater	0	0	0	0	0	0	**0**	0.605

Table 2: Coefficients (with standard errors) of predictors from a multinomial regression model predicting the category of user inaction. They represent the log odd-ratio change with respect to the baseline category NotNoticed. "closest" denotes the closest item that has an action if there is any action on the page. "row" or "col" denotes the row or column index of the position in the grid. "Sim" denotes similarity score. "numShow" denotes the total number of displays while "numFrontShow" denotes the total number of *front-page* displays. "[action]Ratio" denotes the ratio of items having the corresponding [action]. *closetInvDist* represents the inverse of the Euclidean distance between the acted-upon item (if there is any) and the inaction item. Significance levels: *p<0.05, **p<0.01, ***p<0.001.

Level	**Predictor**	**WouldNotEnjoy**	**NotNow**	**OthersBetter**	**ExploreLater**	**DecidedToWatch**	**Watched**
item	*predWishlist*	**10.912 (3.455)** **	1.249 (11.793)	-0.813 (3.121)	-0.052 (1.243)	**4.668 (1.751)** **	-7.124 (13.426)
	predDisplay	3.963 (2.785)	0.058 (7.334)	-3.441 (5.578)	**-5.332 (1.709)** **	-1.402 (1.421)	5.773 (7.508)
	predRating	-0.244 (0.282)	0.069 (0.190)	**0.699 (0.267)** **	0.160 (0.290)	-0.018 (0.324)	0.238 (0.224)
	predClick	0.372 (2.754)	1.974 (18.183)	3.065 (9.895)	-2.920 (2.545)	-0.945 (2.099)	-0.238 (18.831)
	predRate	5.197 (2.657)	-1.875 (6.307)	-5.322 (4.063)	**-5.501 (1.382)** ***	-1.602 (1.238)	**16.935 (7.122)** *
page	*row*	-0.004 (0.133)	**-0.209 (0.088)** *	-0.136 (0.110)	-0.139 (0.132)	**-0.320 (0.151)** *	**-0.243 (0.089)** **
	col	**-0.174 (0.044)** ***	**-0.154 (0.028)** ***	**-0.081 (0.037)** *	-0.077 (0.043)	-0.051 (0.048)	-0.047 (0.031)
	dwell	**0.180 (0.072)** *	0.075 (0.052)	0.099 (0.067)	**0.146 (0.074)** *	**0.217 (0.075)** **	**0.206 (0.055)** ***
	closestCol	**0.102 (0.052)** *	**0.100 (0.035)** **	0.051 (0.044)	0.000 (0.051)	0.064 (0.059)	-0.033 (0.037)
	closestInvDist	0.002 (0.227)	0.134 (0.148)	**0.395 (0.194)** *	**0.435 (0.220)** *	-0.035 (0.251)	**0.385 (0.156)** *
	minSim	-1.245 (1.117)	**-1.930 (0.702)** **	**-1.947 (0.881)** *	-1.002 (1.091)	-2.178 (1.234)	0.663 (0.775)
	medianSim	-4.272 (3.216)	-3.221 (1.976)	-4.939 (2.671)	**-7.648 (3.199)** *	-4.333 (3.410)	-4.025 (2.208)
	meanSim	4.425 (4.465)	**6.532 (2.692)** *	**9.196 (3.556)** **	**9.030 (4.494)** *	6.169 (4.824)	2.532 (3.140)
	maxSim	**-3.073 (1.212)** *	-1.425 (0.762)	-1.923 (0.983)	-1.980 (1.158)	-1.090 (1.299)	-1.002 (0.810)
	closestSim	**0.621 (0.308)** *	0.419 (0.214)	0.182 (0.261)	-0.148 (0.321)	0.247 (0.363)	0.015 (0.221)
	clickRatio	-1.645 (7.072)	**-12.134 (4.887)** *	-4.474 (5.853)	1.977 (6.448)	-6.812 (8.460)	**-24.583 (6.879)** ***
	ratingRatio	-0.661 (1.191)	0.418 (0.831)	0.420 (0.984)	0.984 (1.176)	-0.508 (1.387)	**-1.726 (0.800)** *
	lowRateRatio	**6.840 (2.256)** **	3.584 (1.852)	**4.497 (2.187)** *	1.620 (3.055)	**7.191 (3.011)** *	**4.603 (1.795)** *
	predRateMin	-0.065 (0.408)	-1.257 (0.767)	-0.533 (1.003)	**-1.321 (0.369)** ***	**-1.175 (0.241)** ***	**3.486 (0.795)** ***
	predRateMean	1.231 (1.495)	-1.045 (2.689)	-1.831 (2.741)	**-2.797 (1.072)** **	**-2.330 (1.017)** *	**7.225 (2.413)** **
	predRateMedian	0.714 (1.280)	-1.959 (2.379)	-1.800 (2.564)	**-2.668 (0.953)** **	**-2.436 (0.820)** **	**8.059 (2.235)** ***
	predRatingMin	0.041 (0.214)	-0.134 (0.122)	0.124 (0.165)	0.298 (0.230)	-0.155 (0.199)	0.463 (0.274)
	predRatingMean	-0.536 (2.146)	1.168 (1.302)	-2.417 (1.612)	-2.719 (2.034)	1.620 (2.338)	-1.041 (1.951)
	predRatingMedian	0.600 (1.788)	-0.064 (1.077)	**2.860 (1.407)** *	2.825 (1.726)	-0.138 (1.981)	0.335 (1.511)
	predRatingMax	-0.368 (0.570)	0.183 (0.317)	-0.159 (0.450)	-0.057 (0.556)	0.132 (0.551)	0.164 (0.439)
	predClickMin	-0.055 (0.039)	**-0.243 (0.100)** *	0.049 (0.118)	**0.160 (0.043)** ***	0.009 (0.034)	**0.274 (0.108)** *
	predClickMean	1.272 (1.049)	1.969 (2.030)	0.688 (1.841)	-1.051 (0.743)	0.056 (0.809)	0.377 (1.724)
	predClickMedian	0.681 (0.493)	-0.006 (1.038)	0.805 (0.894)	-0.575 (0.364)	-0.140 (0.408)	0.460 (0.912)
	predWishlistMin	**-0.249 (0.060)** ***	**0.267 (0.129)** *	0.006 (0.174)	0.022 (0.058)	**-0.298 (0.035)** ***	-0.081 (0.154)
	predWishlistMax	4.412 (11.852)	13.995 (7.845)	1.154 (12.143)	-5.502 (16.207)	-4.041 (18.608)	1.706 (9.528)
	predDisplayMin	-0.102 (0.096)	0.327 (0.218)	**-0.692 (0.251)** **	-0.066 (0.098)	-0.066 (0.101)	**1.049 (0.149)** ***
	predDisplayMean	-0.197 (1.200)	-0.450 (2.288)	-1.284 (2.826)	-1.554 (1.067)	-1.181 (0.839)	**5.458 (2.218)** *
	predDisplayMedian	-0.030 (0.744)	-0.145 (1.390)	-0.760 (1.675)	-1.065 (0.632)	**-1.519 (0.536)** **	**3.016 (1.235)** *
	predDisplayMax	-5.349 (7.038)	-6.996 (15.833)	-5.690 (18.309)	-5.104 (6.797)	-2.694 (4.402)	**36.050 (16.626)** *
session	*numFrontShow*	**0.818 (0.252)** **	**0.646 (0.168)** ***	**0.659 (0.209)** **	**0.832 (0.237)** ***	**0.750 (0.253)** **	**0.600 (0.191)** **
	lowRateRatio	**14.226 (6.922)** *	-6.013 (4.337)	-5.836 (5.772)	**-24.322 (5.887)** ***	-11.919 (7.467)	-2.251 (5.417)
user	*length*	0.002 (0.023)	0.022 (0.016)	0.001 (0.020)	-0.007 (0.023)	-0.022 (0.028)	**-0.047 (0.017)** **
	numShow	0.203 (0.106)	**0.299 (0.066)** ***	**0.253 (0.088)** **	**0.251 (0.100)** *	**0.567 (0.111)** ***	-0.079 (0.095)
	ratingRatio	**-17.097 (8.493)** *	11.615 (9.645)	2.015 (10.951)	7.201 (5.728)	2.984 (3.826)	2.160 (8.145)
	highRateRatio	16.444 (8.744)	-12.647 (9.837)	-1.533 (11.134)	-6.516 (6.038)	1.408 (4.318)	-0.257 (8.222)
	lowRateRatio	-0.556 (9.815)	-5.772 (9.532)	-0.386 (11.086)	10.145 (6.072)	2.154 (5.632)	-2.212 (8.952)
	wishlistRatio	-5.888 (12.539)	3.325 (6.203)	**14.435 (4.925)** **	**13.525 (5.520)** *	0.213 (11.142)	**14.693 (4.442)** ***

the probabilities of being displayed, rated, clicked, wishlisted respectively. We see that predicting rating, whether to display or rate are easier tasks than predicting whether to click or add into the wishlist in the system.

With the pre-built models, after systematically examining the factors that are potentially predictive, we summarized the following list of predictors.

- *item level*

  - popularity of the movie, *i.e.*, the number of ratings the movie has in the system
  - predicted rating, likeliness of displaying, rating, clicking and adding into the wishlist from the *sub-models*.
  - position where the item was displayed
- *page level*
  - user dwell time on the page
  - the ratio of *action* (used to refer to either clicking, rating or adding into the wishlist except hovering) and hovering on the 24 movies (*i.e.*, the number of acted movies divided by 24)
  - the min, max, median and mean of rating value, action and displaying probability predictions for the page of movies
  - the closest acted item's position in the grid, its Euclidean distance from the inaction movie (inversed), rating value and action predictions, if there is any action on the page
  - the min, max, median and mean of the similarities of the other movies displayed together with the inaction movie (these are cosine similarities computed on the movie latent factor representation from the rating value matrix factorization model)
- *session level*
  - the length of the session in seconds and how many movies were displayed before the page view
  - the ratio of hovering, action on the displayed movies before the page view
  - how many times the inaction movie has been shown in the session before the page view (this is further separated into two types of displays: displays on the *front* page and displays on the *explore* page).
- *user level*
  - the tenure of the user in the system in seconds and how many movies were displayed to the user before the page view
  - the ratio of hovering, action which includes either clicking, rating or adding into the wishlist except hovering on the displayed movies before the page view. Rating is further divided into lowRating(<4.0) vs. highRating(>=4.0).
  - how many times the inaction movie has been shown across sessions in the user history before the page view

The inaction model serves two purposes in this work: 1) understanding what predictors and how these predictors help infer the inaction categories and 2) achieving usable accuracy so that a recommendation algorithm can utilize its output to improve future predictions. Therefore, we employed two types of techniques respectively for these two purposes. First, for the purpose of interpretation, we employed multinomial regression to both test the significance of the predictors and their effects' signs and sizes. Second, to achieve the best classification accuracy, we used Gradient Boosted Decision Tree (GBDT, boosted ensemble of decision trees) model [9] and the popular implementation: xgboost [4]. This implementation supports sparsity regularization (similar to the technique of LASSO [24]) which enables automatic feature selection by tuning the regularization strength parameter α. After tuning, we found the best parameter set to be: $max\_depth = 1$ (the maximum depth of each tree), $num\_trees = 51$ (total number of boosted trees), $\alpha = 5.0$ (L1-norm based sparsity regularization), $\lambda = 1.0$ (L2-norm based regularization). The best model accuracy (7-class classification accuracy) is 48.5%, which is significantly better (8.6% accuracy boost) than the naive baseline of always predicting the majority class (p=0.0001 based on exact binomial test; the number of testing survey responses is 678; the majority class NotNoticed occupies 39.9%).

Table 1 shows the confusion matrix and the accuracy in terms of AUC for each class (binary classification of one vs. others using the output of the best 7-class classification model) of the inaction model. It shows that the model is struggling in differentiating other classes from the majority class NotNoticed. However, the probability scores predicted by the model still have certain capabilities to different each class from others as suggested by the AUC metric, specifically the model performs best in predicting for the classes of Watched, OthersBetter, DecidedToWatch but performs worst for the classes of ExploreLater, WouldNotEnjoy. Generally, inferring inaction categories is a hard task.

Because of the use of the sparsity regularization, GBDT model can provide non-zero-importance predictors after regularization. The effects of non-zero-importance predictors on inaction inference (from one multinomial regression model for interpretability) are listed in Table 2. Note that the coefficients in Table 2 are the log odd-ratio change of the corresponding category compared with NotNoticed. To illustrate, we present two examples:

- The higher predicted probability of a user to rate a movie, the more likely that the user has watched the movie before. It suggests that predicting whether an item will be rated by a user in the system can approximate the user's familiarity on the item. (the cell in *predRate* & Watched)
- The higher the *predWishlist* is, the more likely the reason for inaction is DecidedToWatch. It suggests that predicting whether a movie will be added into the wishlist by a user in the system can approximate the likeliness of a user deciding to watch the movie, which is consistent with the design goal of the wishlist feature of the system.

As shown in Table 2, the predicted action probabilities can be useful signals in inferring the reasons of inaction for displayed recommendations. Although the possible actions that users can do in different systems vary, we see many interfaces in modern recommender systems such as Netflix, Youtube support similar actions as rating, clicking or adding into a wishlist. Therefore, these findings could potentially generalize across multiple platforms. However, future research is necessary to further validate this.

8 IMPROVE RECOMMENDATION

In this section, we answer *RQ4: Can we improve recommender systems utilizing the user inaction model?* There are three possible ways of utilizing the inaction model to improve recommender systems.

- Preference estimation. Can we improve rating prediction accuracy by utilizing the inaction model output?
- Action prediction. Can we improve action prediction accuracy by utilizing the inaction model output?
- Recommendation timing. Can we do better in terms of when to recommend by utilizing the predicted probabilities of NotNow from the inaction model?

Table 3: Three possible ways to utilize the user inaction model in recommender systems. MF denotes Matrix Factorization and FM denotes Factorization Machine.

Goal	Model	Metric

Rating prediction (regression, L2-norm loss)	MF: user ID + item ID	MAE=0.912
		RMSE=1.11

	FM: user ID + item ID + (7-class predicted probabilities)	MAE=0.950
		RMSE=1.17

Action prediction (whether action or not when displayed; binary classification, logistic loss)	MF: user ID + item ID	AUC=0.774
	FM: user ID + item ID + (7-class predicted probabilities)	AUC=0.787
Recommendation timing	predicted NotNow probability vs. time taken for action	Pearson=0.0264 p<0.001***
	predicted NotNow probability vs. whether acted in a different session	AUC=0.562

We use similar training and testing data sets as previous sections to answer these questions. However, we only took part of the system logs because it is expensive to extract predicted inaction category probabilities for all the 53M movie displays. To put these models into real systems requires amortizing the computational costs of running the additional models. For each of the 25K users, we take ten page views and their user interactions before Feb. 01, 2018 as the training set (176K ratings, 4.4M movie displays) and take one page view and their interactions on and after Feb. 01, 2018 as the testing set (7K ratings, 135K movie displays). Table 3 shows the results of testing the three possible ways of utilizing user inaction model output. The rationale of these approaches is that if we want to know whether or when we should recommend an item to a user, we first check whether we have displayed this item to the user before and generate predicted inaction probabilities as input to guide our decision. We used the technique of Factorization Machine [21] (FM) to incorporate the additional inputs of predicted inaction category probabilities. When an item was never displayed before, we use a default values of zeros as the additional input to FM. For both Matrix Factorization (MF) [17] and FM, we used 32 latent dimensions. To answer the recommendation timing question, we select inaction items that were later acted upon by users and analyze the Pearson correlation between the predicted NotNow probability of each inaction item and the time it takes for the user to act on it later. We also use this predicted NotNow probability to predict whether the action was in a different session measured in terms of AUC.

As illustrated in Table 3, we did not see improvement for rating prediction in terms of MAE or RMSE, *i.e.*, estimating user preference, but saw improvement in action prediction in terms AUC (predicting whether there will be any action on an item if it is recommended). It suggests that the inaction model can potentially improve recommender systems that targets maximizing user action (user engagement) in the system. We also observe that the predicted NotNow probability of the inaction model may help the system make the decision of delaying the recommendation of an inaction item to the next session or later in time.

9 DISCUSSION AND CONCLUSION

In online recommender systems, there are many possibilities to explain user inaction on recommendations. In this work, we summarized and collected data on seven major categories of them inspired by the psychology literature of behavioral decision making and found they significantly affect user future recommendation preferences on inaction items. In recommender systems literature, inaction or missing observations are assumed to reflect lack of interest and thus are usually treated as negative feedback [14, 18, 22]. Our research suggests that inaction is more complex than this assumption. For example, there is a high chance that the cases of ExploreLater and DecidedToWatch inaction are positive user feedback. We found attention plays a significant role in user inaction, which implies that new ways of presenting top recommendations might be necessary to better utilize user attention.

We designed and tested models to infer user inaction so that systems can avoid always asking users for inaction reasons. We achieved significantly better-than-random classification accuracy especially for certain inaction categories, *e.g.*, Watched, OthersBetter or DecidedToWatch. We found interesting predictors that signify how we might better infer the inaction category probabilities, *e.g.*, predicted wishlist probability signifies a higher chance of DecidedToWatch and predicted being-rated probability signifies a higher chance of Watched. Generally, we found that user inaction inference is a hard task. However, with the advent of more accessible sensors like portable eye-tracking equipments, we consider it promising further work to explore how these new measurements can help better infer user inaction.

We demonstrated that the user inaction classification model we built can improve action prediction tasks which can be used by recommender systems to maximize user action engagement (*i.e.*, recommending items that have the highest predicted action probabilities). We also showed that the predicted probability of NotNow from the inaction model could potentially improve recommendation timing, *e.g.*, delaying a previously displayed recommendation to the next session if the model predicts that this item can only be interesting to the user in future but not now.

Our future work is to test the effects of this inaction model on user experience through designing and deploying field experiments. This can help answer the question of whether this model may improve recommendation freshness without hurting accuracy or reduce the confusion of recommender systems that are based on dynamic algorithms learning from user action feedback neglecting user inaction.

10 ACKNOWLEDGMENT

This work was supported by the National Science Foundation under grant IIS-1319382. The first author was also supported by the Doctoral Dissertation Fellowship, 2016-17, by the Graduate School at the University of Minnesota. We thank all the MovieLens users who participated in our study.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2015. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 191–226.

[2] Jerome R Busemeyer and Joseph G Johnson. 2004. Computational models of decision making. *Blackwell handbook of judgment and decision making* (2004), 133–154.

[3] Jerome R Busemeyer and James T Townsend. 1993. Decision field theory: A dynamic-cognitive approach to decision making in an uncertain environment. *Psychological review* 100, 3 (1993), 432.

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

[5] Ye Chen and Tak W Yan. 2012. Position-normalized click prediction in search advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 795–803.

[6] 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*. ACM, 87–94.

[7] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 331–338.

[8] Hillel J Einhorn and Robin M Hogarth. 1981. Behavioral decision theory: Processes of judgement and choice. *Annual review of psychology* 32, 1 (1981), 53–88.

[9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[10] Frank Y Guo and Keith J Holyoak. 2002. Understanding similarity in choice behavior: A connectionist model. In *Proceedings of the twenty-fourth annual conference of the cognitive science society*. 393–398.

[11] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[12] Yosef Hochberg and Yoav Benjamini. 1990. More powerful procedures for multiple significance testing. *Statistics in medicine* 9, 7 (1990), 811–818.

[13] Katja Hofmann, Anne Schuth, Alejandro Bellogin, and Maarten De Rijke. 2014. Effects of position bias on click-based recommender evaluation. In *European Conference on Information Retrieval*. Springer, 624–630.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[15] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. Acm, 154–161.

[16] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

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

[18] Pei Lee, Laks VS Lakshmanan, Mitul Tiwari, and Sam Shah. 2014. Modeling impression discounting in large-scale recommender systems. In *Proceedings of SIGKDD'14*. ACM, 1837–1846.

[19] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 1097–1101.

[20] Pearl Pu, Li Chen, and Rong Hu. 2011. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 157–164.

[21] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[23] Ramakrishnan Srikant, Sugato Basu, Ni Wang, and Daryl Pregibon. 2010. User browsing models: relevance versus examination. In *Proceedings of SIGKDD'10*. ACM, 223–232.

[24] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.

[25] Shuang-Hong Yang, Bo Long, Alexander J Smola, Hongyuan Zha, and Zhaohui Zheng. 2011. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of SIGIR'11*. ACM, 295–304.

[26] Qian Zhao, Gediminas Adomavicius, F Maxwell Harper, Martijn Willemsen, and Joseph A Konstan. 2017. Toward Better Interactions in Recommender Systems: Cycling and Serpentining Approaches for Top-N Item Lists. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 1444–1453.

[27] Qian Zhao, Shuo Chang, F Maxwell Harper, and Joseph A Konstan. 2016. Gaze Prediction for Recommender Systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 131–138.