Letting Users Choose Recommender Algorithms: An Experimental Study

Michael D. Ekstrand[1], Daniel Kluver[2], F. Maxwell Harper[2], and Joseph A. Konstan[2]

[1]Dept. of Computer Science
Texas State University
San Marcos, TX, USA
ekstrand@txstate.edu

[2]GroupLens Research
Dept. of Computer Science
University of Minnesota
Minneapolis, MN, USA
{kluver,harper,konstan}@cs.umn.edu

ABSTRACT

Recommender systems are not one-size-fits-all; different algorithms and data sources have different strengths, making them a better or worse fit for different users and use cases. As one way of taking advantage of the relative merits of different algorithms, we gave users the ability to change the algorithm providing their movie recommendations and studied how they make use of this power. We conducted our study with the launch of a new version of the MovieLens movie recommender that supports multiple recommender algorithms and allows users to choose the algorithm they want to provide their recommendations. We examine log data from user interactions with this new feature to understand whether and how users switch among recommender algorithms, and select a final algorithm to use. We also look at the properties of the algorithms as they were experienced by users and examine their relationships to user behavior.

We found that a substantial portion of our user base (25%) used the recommender-switching feature. The majority of users who used the control only switched algorithms a few times, trying a few out and settling down on an algorithm that they would leave alone. The largest number of users prefer a matrix factorization algorithm, followed closely by item-item collaborative filtering; users selected both of these algorithms much more often than they chose a non-personalized mean recommender. The algorithms did produce measurably different recommender lists for the users in the study, but these differences were not directly predictive of user choice.

Categories and Subject Descriptors

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

General Terms

Human Factors, Algorithms

Keywords

Recommender systems, experiment, field study

1. INTRODUCTION

Recommender systems generally keep their algorithms "behind the scenes" and do not offer users a choice of algorithm. In this paper, we report on a field study where we deployed a recommender that offered users the choice of different algorithms. In this study, we investigate both how users explore and/or switch among algorithms and the preferences they reveal when given the chance to experience multiple algorithms.

This experiment takes place in the context of the release of a new version of the MovieLens movie recommendation service[1]. The new release adds support for multiple recommender algorithms and gives users the ability to switch their algorithm. Users are randomly assigned to one of the algorithms as their initial condition. This gives us the opportunity to study three things: the ways in which users take advantage of the ability to switch the algorithm providing their recommendations, the algorithm or algorithms users prefer, and the impact of the assigned algorithm on user behavior.

Within this setting, we seek to answer several research questions:

1. Do users take advantage of a means to switch recommender algorithms?
2. What algorithm(s) do users prefer to use? Is there a clear favorite, or do different users prefer different algorithms?
3. How do users explore the algorithm options?
4. How stable are user selections? Do they experiment for a little while, pick a recommender, and leave it alone, or do they change recommenders often throughout their use of the application?
5. How do the recommendations users receive from the algorithms differ?
6. Can we predict the user's choice of algorithm, or do we need to keep them in control in order to identify the algorithm with which they will be most satisfied?
7. Does the algorithm with which a user starts affect their use of the system, final choice of algorithm, or likelihood to continue using the system?

We use three primary algorithms in this study: a baseline using user and item mean ratings, an item-item collaborative filter, and a matrix factorization recommender. These were selected for broad usability (the baseline recommender), continuity with the system's previous behavior and widely-deployed algorithms (item-item CF) and to have an algorithm representative of the current state of the art in collaborative filtering (matrix factorization).

In the remainder of this paper, we describe our experimental setup, our key findings, and conclusions and lessons for the design and deployment of recommender applications.

[1] http://movielens.org

2. BACKGROUND AND RELATED WORK

It is well-understood that different recommender algorithms, even ones that have quantitatively similar behavior on common accuracy metrics, produce recommendations that differ in ways that users can perceive [7, 18] and that may impact their ability to meet different user needs, or the needs of different users.

One way of making use of the advantages of different algorithms to improve recommendation for particular users is through hybrid recommenders [2]. Hybrids can blend the strengths of different approaches, even tuning the blend on a per-user or per-item basis to use each algorithm the most where it has the most to offer [27], or attempting to detect the best algorithm for each user [9]. We can view the goal of such hybrids as that of building a *meta-recommender*, identifying the particular algorithm (or combination of algorithms) that will best meet the needs of a particular user in a particular context.

Another approach, which we explore in this paper, is to involve the users in the process of selecting their recommender. Many recommenders take explicit feedback from their users to form the basis of the recommendations [23], or to set up an initial context for new users [5, 17, 24]. Relevance feedback incorporates user responses — either explicit or implicit — into a feedback loop to improve future iterations of the recommender [1, 13]. In this work, we take a very basic approach to incorporating users' explicit feedback into the recommender selection process: we invite users to try different recommender algorithms and pick the one they want to use. This is similar to the work of Dooms, where users were given some control over their recommendations [6]; some results in that work parallel ours.

Many previous studies have examined the impact of different recommendation techniques on users' subjective perceptions of recommendations [7, 14, 25], and it is common practice in industrial applications to measure the impact of recommendations on user behavior using A/B trials and similar experimental models [15]. In this work, we examine user response to different recommendation approaches when the action they can take is to switch recommender algorithms, rather than simply to accept or reject algorithms or discontinue use of the service. Our motivations are similar to those behind the idea of user studies: providing ratings and other actions in response to a single recommendation approach is an inadequate expression of the user's thoughts and preferences as to how their recommendations should be provided, and there is value in listening to what users say they want to do, not just optimizing the system for particular user responses that we the system's creators and analysts deem indicative of usefulness. In user studies, we solicit users' subjective impressions and stated preferences with regards to an algorithm; in this work, we examine the choices users make when they can select the recommender suggesting movies to them.

There has been previous work on providing users with insight into, and in some cases control over, the system's model of their preferences [4, 12]. Some commercial recommenders support some version of this, through explicit interest profiles often used in a new user onboarding process (exemplified by Twitter or Microsoft Cortana) or by allowing the user to view and correct profile information that will feed into their recommendations (a model adopted by Amazon.com). One of the goals of recommender explanation is also to provide some transparency to users into the source of their recommendations [28]. Ziegler et al. proposed a user-adjustable control for the amount of diversification applied to a recommendation list [30], and the retrieval models of Tapestry were entirely user-guided [10]. But there is comparatively little

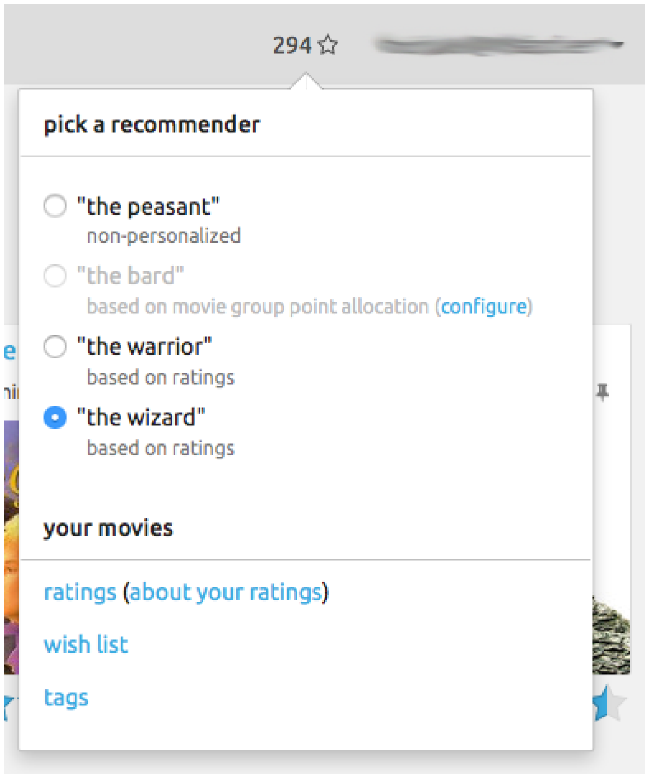

Figure 1: Recommender Switching Control

work on allowing users to control *what algorithm* provides their recommendations, as opposed to refining the algorithm's input data or re-tuning a structurally static algorithm.

3. EXPERIMENTAL SETUP

We studied the behavior of users of MovieLens, a non-commercial movie recommendation service, as they interacted with the recommender switching capabilities deployed as a part of an overhaul of the service. The new version of the service supports multiple recommender algorithms, selectable on a per-user basis, and gives users a control to pick the algorithm that they are using. When an existing user signed in to the new version of the service, or returned to it after the feature was deployed, they were randomly assigned one of the algorithms as their initial condition. They were also shown a brief interstitial informing them that the service now supports multiple algorithms, and pointing to the control for changing algorithms. In this study, we only consider previously-existing users who signed in to the new site.[2]

We consulted with our IRB on this study, and they determined that the research was exempt and did not require specific user consent beyond users' general agreement to use our service and acceptance of its terms of use.

The service supports four algorithms. Each algorithm was identified to users using a code name, derived from common role-playing character classes, so that they could be memorable but would not disclose to users exactly what algorithm they were using; a very brief description accompanied each algorithm name.

[2] New users who first signed up with the new site participated in a different experiment, so they could not be considered in the data analysis for this study.

- The *Baseline* algorithm predicts ratings using a user-item personalized mean. This algorithm was called the *Peasant*, and was described as "non-personalized".
- The *Pick-Groups* recommender is an item-item collaborative filter that uses synthetic item ratings derived from the user's choice of different movie groups [3]. It is intended to provide an improved user experience for new users of the system. No existing users started with this algorithm, but they could try it out if they wished. This algorithm was called the *Bard*, and was described as "based on movie group point allocation".
- The *Item-Item* recommender [26], called the *Warrior* and described as "based on ratings", is an item-item collaborative filter. It uses cosine similarity over item-mean-centered rating vectors, a maximum neighborhood size of 20, minimum neighborhood of 2, and a neighborhood similarity weight threshold of 0.1.[3] Item-item CF was the only recommender available in the service prior to deploying the recommender-switching feature.
- The *SVD* recommender, called the *Wizard* and also described as "based on ratings", is a matrix factorization recommender using the FunkSVD algorithm [19, 22] with 50 features, 125 training epochs per feature, and subtracting the user-item personalized mean prior to factorizing the matrix.

All algorithms are built with the LensKit toolkit [8]. To compute top-N lists, the algorithm's predicted rating is the primary ranking factor, but it is blended with the popularity of the item:

$$\widehat{s_{ui}} = 0.9 \cdot rank(s_{ui}) + 0.1 \cdot rank(p_i)$$

s_{ui} is the score; p_i is the number of ratings the item received in the last 365 days; and $rank(x)$ normalizes its input to a rank, returning 1 for the largest value (across all items) and 0 for the smallest value. This blending is the result of empirical evidence that balancing popularity with prediction rank leads to greater satisfaction with top-N recommendation lists.

Full LensKit configurations for each algorithm are available as a supplement in the ACM Digital Library.

Once in the system, users could change their algorithm by clicking a widget in the site's top bar, as show in Figure 1. This took effect immediately; once the user selected a different recommender, the system would reload the list of movies on the current page (if they were viewing a movie recommendation page) and show the results from the freshly-selected recommender. The system's interface is organized around three major functions:

1. Movie recommendations: the default view shows a set of movie recommendations with relevant images, usually a movie poster, and the predicted rating.
2. Movie details: clicking on a movie image brings up more details such as production metadata, tags, average rating statistics, etc.
3. Search: a keyword-based search feature, with additional options to search by various movie attributes. Search results are displayed in the same fashion as recommendations, and the recommender's predicted rating is a component of the search result ranking function.

The user's choice of recommender algorithm persisted throughout the application, affecting all use of predictions for display or movie ranking.

[3] This means that it would only recommend items for whom it could find a neighborhood with a total similarity of at least 0.1.

Table 1: Summary of Experiment Data

Users	3005
Users switching at least once	748 (24.9%)
Recommender change events	11,423
Median changes per user w/ at least 1 change	3
Median account age	1653 days
Median ratings per user (at end of study)	354

Our reporting here is based on logs of the feature's use from its deployment in November 4, 2014 through March 31, 2015. Table 1 summarizes the data we collected. 3005 users used the system, of which 748 changed recommenders at least once; the median account age upon signing in to the new system was 1653 days (min 0, max 6086). Our logs have 11,423 change events.

In addition to using the logs of user interactions, we also used the ratings database to run an offline evaluation to measure the accuracy of the algorithms on each user's recent history prior to entering the experiment. Starting with a current dump of the ratings database, we put all ratings from all users not in our experiment into a training set. For each user in our experiment, we did the following:

- Discarded all ratings after entering the experiment
- Put the 5 most recent ratings prior to entering the experiment into a set of test ratings
- Put the user's remaining ratings into the training set

We then ran each algorithm (with the exception of *Pick-Groups*; it is difficult to do historical recreations of that algorithm outside of a running instance of the recommendation service) and measured the following:

- Prediction accuracy (RMSE) for the test ratings.
- Top-*N* accuracy (Recall) for finding the user's recent ratings among the top 24 recommendations.[4]
- Diversity (intra-list similarity [30], with cosine similarity over descriptive attribute data [29] as the movie similarity metric) of a 24-item recommendation list. Not all movies have sufficient data for a similarity computation, so we normalized the intra-list similarity by dividing it by the ILS that would be achieved if all pairs of genome movies recommended had a similarity of 1.
- Popularity (mean popularity rank, where 1 is the most popular movie) of movies in a 24-item list.

We used 24 items for each recommendation list because that is the length of a single page of recommendations in the recommender application's web interface.

4. RESULTS

In this section, we describe our findings from this study. We have organized this section in order of our research questions, though more than one question may be addressed in a section.

[4] We also measured reciprocal rank and average precision, but the recommenders found users' rated items so infrequently that these did not provide very meaningful comparisons. They effectively compare how well less than a third of the algorithms/user pairs do at ranking items for the user. Whether the algorithm was able to find one (or more) of the user's items is a question that can be meaningfully answered for all algorithm/user pairs.

4.1 RQ1: Users Switch Algorithms

Of the 3005 users in our study, 748 (24.9%) changed recommenders at least once. This does not count the 31 users whose only interaction with the control was to select their current recommender. These users' activity was likely the result of an interface artifact; the control indicated the user's current algorithm (with a filled radio button) but did not disable that algorithm, and some users may have thought that clicking the algorithm would provide more information on it.

While the feature was certainly not used universally, it was tried by a significant fraction of users. 539 users (72.1% of the users who tried the control) settled on a different algorithm than they had been assigned. This answers RQ1 in the affirmative: users did make use of the ability to change recommenders.

Of the users who entered the experiment before March 1, 2015 (so we have data on them for at least a month since they entered), users who switched recommenders were more likely to sign in again at least a week later (83.4% vs. 54.5%; p<0.001). This is not necessarily a causal relationship, but may indicate that more active users in the system are more particularly interested in the recommender-switching feature. This finding is consistent with those of Dooms [6].

4.2 RQ2: User Algorithm Preferences

Among users who tried different algorithms, SVD was the most favored algorithm, followed by Item-Item and finally the Pick-Groups and Baseline recommenders. Figure 2 shows how many users who switched recommenders at least once selected each of the algorithms as their final choice (their active recommender as of close of data collection).

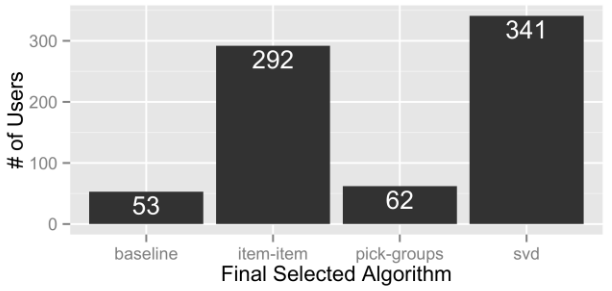

Figure 2: User Choice of Algorithm

We can see that users clearly favor the personalized algorithms over the baseline, and over the group-based recommender. Since users knew that the baseline was non-personalized, and that the pick-groups recommender was based on movie groups, we cannot infer from this data that users preferred the recommendations provided by the personalized algorithms (as opposed to the idea of personalization). We can, however, observe that, given the choice, most users will choose to use recommendations they know (or at least believe) to be personalized. This result is not surprising, but does provide more data to support the idea that users believe the work of recommender systems to be useful.

4.2.1 Response to Initial Algorithm

Users who were assigned to the non-personalized condition were more likely to try other recommenders. Item-item users were the next most likely to use the switcher. Figure 3 shows the fraction of users in each initial algorithm condition who tried a different recommender; all differences are significant ($p < 0.05$, χ^2 with Bonferroni correction for multiple tests). Analyzing this result shares the confound discussed previously: users knew that the baseline algorithm was non-personalized. They did have to access (but not necessarily use) the control in order to obtain this information, but all logging is server-side, so we do not have log data on how many users viewed the switcher control but did not use it.

Integrating users' final choices and whether they switched at all indicates that more users find the SVD-based recommendations to be most satisfactory; item-item the next most satisfactory; and finally the baseline algorithm.

We did not find any effect of the user's initial algorithm on their final choice if the user tried different algorithms; users settled on each algorithm at similar rates irrespective of the algorithm they started with. The only measurable impact of initial algorithm on final choice was via its impact on whether users experimented with different algorithms in the first place.

In addition to refining our answer to RQ2 (users are most satisfied with SVD, followed by item-item, we make two observations with respect to RQ7: the initial algorithm affects the likelihood that a user will try other recommenders, but once they decide to experiment, it has little bearing on their final choice.

4.3 RQ3&4: Switching Behavior

The vast majority (97.3%) of users switched recommenders no more than 20 times (although one user recorded 7296 transitions). Most users switched just a few times; only 21.4% switched more than 5 times. Figure 4 shows the distribution of user transition counts (for users switching no more than 20 times).

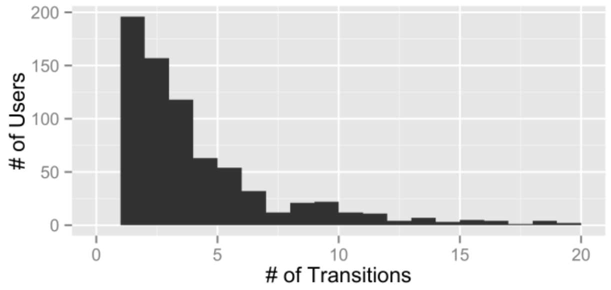

Figure 4: Transition Count Distribution

The most common switching patterns were for a user to switch away to a different algorithm, or try two or three algorithms and then settle down. The median number of transitions is 3, after which there is a marked drop-off; this is enough transitions for a user to try each of the personalized algorithms and return to their favorite.

4.3.1 Transition Sequences

We examine the transitions in more detail, looking to see the relative likelihood of different state transition sequences. Table 2 shows the common transition patterns among users who switched at least once. Percentages are the percentage of users who started

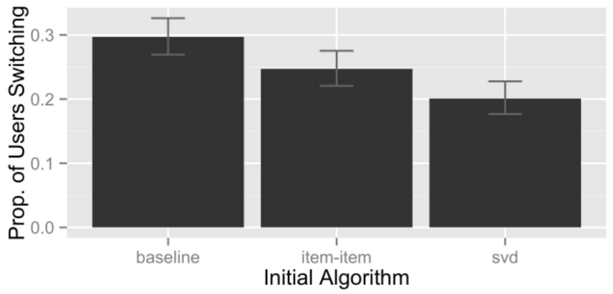

Figure 3: Likelihood to Switch Recommenders

with that algorithm and transitioned at least once. Patterns not shown appeared less than 5% of the time. Transition chains are complete: II → SVD means that the user started with item-item, transitioned to SVD, and did not switch recommenders again.

Users generally experimented with the personalized algorithms, trying each at most once and sometimes returning to an earlier good algorithm. Behavior of users starting with the baseline was somewhat more diffuse than users starting with one of the personalized algorithms; the most users switched from the baseline to one of the personalized recommenders (29.6%, evenly split between the two of them), with an additional 7.9% trying both and staying with SVD, and 6.5% trying both and selecting item-item.

Table 2: Common Transition Sequences

Sequence	Count	%	Cum %
II → SVD	52	21.1%	21.1%
II → SVD → II	23	9.3%	30.4%
SVD → II	30	15.2%	15.2%
SVD → II → SVD	21	10.0%	25.2%
BL → II	45	14.8%	14.8%
BL → SVD	45	14.8%	29.6%
BL → II → SVD	24	7.9%	37.5%
BL → II → SVD → II	12	3.9%	41.4%
BL → SVD → II	8	2.6%	44.0%

4.3.2 *Transition Time and Other Events*

26.2% of users who switched recommenders only did so within their first hour of using the new system. The median user performed their first transition in the first 10 minutes or 16 logged actions[5] (viewing a page, changing recommenders, or taking an action on a movie or a tag), and their last transition within the first 18 hours.

We also broke user actions down into *sessions*, considering a session to end when the user is inactive for at least 60 minutes [11]. As implied by the timing data in the previous paragraph, the median user first switched recommenders in the first session. 44.1% of users only switched recommenders during their first session; these users had a median of 2 and mean of 9 further sessions after the one in which they changed recommenders. 63% of users only switched recommenders in a single session, even if it was not their first, and 80.7% of users only switched recommenders in 2 different sessions.

Users also interacted with the recommender changer without very many intervening events. They would change recommenders, see the results, and if they switched recommenders again, they usually did so quickly. The median user viewed one page between recommender changes within a single session.

4.3.3 *Summary*

To answer RQs 3 and 4, users tended to experiment with the recommender switching control early in their use. Most users flipped the switch a small number of times, usually in one or two distinct sessions and without much other intervening activity, and then left the recommender setting alone.

[5] These 16 actions are not counting the 2 actions at the beginning of every user's logged data: viewing the interstitial and the application's front page.

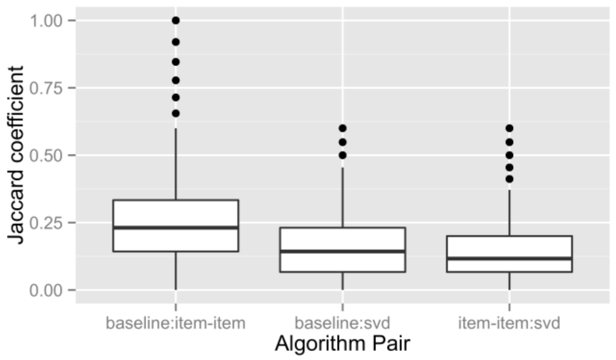

Figure 5: Overlap in Recommendations

4.4 RQ5: Recommendation Properties

Our offline simulation of the recommenders the users interacted with allow us to examine how different the output of the different recommender algorithms are in terms of various objectively measurable characteristics. In addition to providing insight for interpreting user behavior, these results also provide some measurement of the recommendations that are seen by actual active users in the system, rather than the mix of inactive and active users seen in typical offline evaluations over entire data sets. This does not take into account changes resulting from users adding ratings to the system after joining the experiment, but captures the behavior of the recommender immediately before they entered the study.

374 of the users participating in our study did not have sufficient data for the offline analysis (they had 5 or fewer ratings prior to joining the experiment), so they are excluded from the analysis in this section.

First, the algorithms did produce different lists of items. The three recommenders (baseline, item-item, and SVD) produced an average of 53.8 unique items per user (out of a maximum of 72 for three 24-item lists). Figure 5 shows the Jaccard similarity of each pair of recommendation lists for each user. Baseline and item-item were the most similar, with SVD producing results that were more divergent from them.

The algorithms also had differing prediction and recommendation accuracy (measured against the user's 5 most recent ratings prior to entering the experiment), shown in the left side of Figure 6. The patterns in objective accuracy track with those in user preference, with SVD being the most accurate, followed by item-item and finally baseline.

The second shows Boolean recall, the fraction of users for which each algorithm produced at least one of the user's test ratings in the first 24 recommendation results. This metric tracks differently from user's choice of algorithms: even though item-item found the user's rated item for more items, more users preferred the

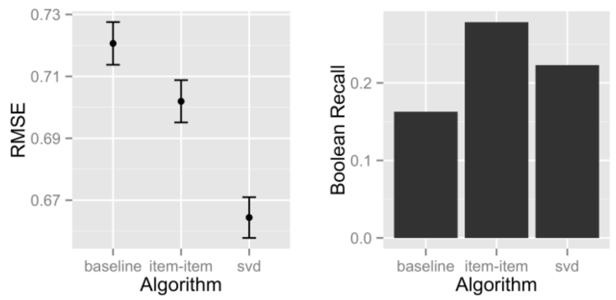

Figure 6: Algorithm Accuracy

SVD recommender. We recognize that the effects of recall can be difficult to interpret; low recall may suggest that an algorithm that is not successfully finding the user's most preferred items, while high-recall may suggest that an algorithm is not recommending enough new and unfamiliar content.

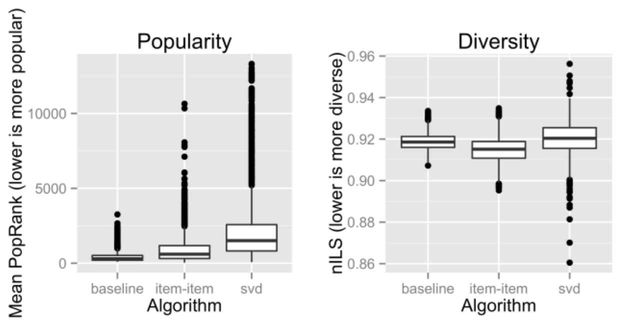

Figure 7: Popularity and Diversity of Recommendation Lists

Figure 7 shows the popularity and diversity of recommendations achieved by each recommender. SVD produced the most novel (least popular) recommendations. Item-item produced the most diverse recommendations, with SVD having the greatest variance in diversity, although the differences in diversity are small.

The charts so far have all considered the performance of the algorithms without accounting for their relative performance within the user experience. For any particular user, it is unlikely to matter whether one algorithm tends produce more diverse recommendation lists than another; rather, when comparing algorithms, they will notice if one algorithm produces more or less diverse results than another *for them*. To address this question, Figure 8 shows the relative performance of the two personalized algorithms, normalized to the performance of the baseline algorithm on each user. For example, an RMSE of 0.03 for a user means that the algorithm in question had an RMSE that was 0.03 better than that of the baseline algorithm for that particular user. We averaged the relative performance across all users, and show the results with standard error bars. Recall in this chart is the ordinary recall, the fraction of relevant items retrieved, rather than Boolean recall.

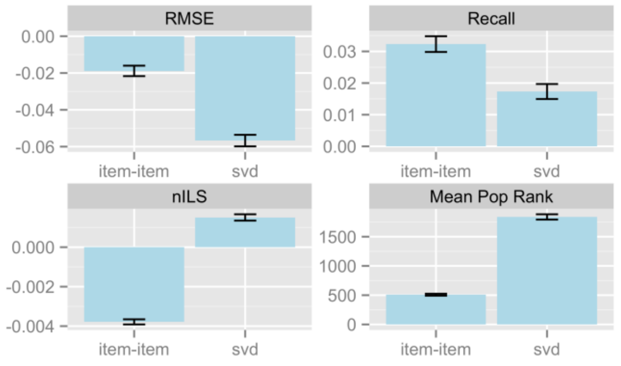

Figure 8: Relative Algorithm Performance

We note significant differences in algorithm performance on all metrics. These differences are trend-consistent with the overall differences previously, but allow us to see how much difference in e.g. diversity each algorithm is likely to make for an individual user. The differences in diversity are quite small, while the differences in popularity and accuracy are bigger and more likely to be noticeable (particularly the popularity difference).

Our offline metrics do not predict which algorithm users will prefer, or if they will switch algorithms in the first place, beyond the predictive power of the algorithm's identity. That is, the popularity of a recommender's results is no more powerful of a predictor of user switching and selection behavior than 'is the algorithm SVD?'. The most-preferred recommender – SVD – produces less popular (and therefore more likely to be novel) results than the other recommenders, but do not have direct evidence that this difference in novelty is the reason that users prefer SVD. The fact that the relative popularity of recommendations from the three algorithms is rank-consistent with users' tendency to choose those algorithms, whereas their relative diversity is not, suggests that popularity or novelty may play a larger role in user preference than diversity, at least as we have measured it here, but is not conclusive evidence.

4.5 RQ6: Predicting User Behavior

We built logistic regressions to attempt to predict switching behavior. The most significant predictor of whether a user would try a different algorithm was the algorithm they started with. The significant differences in metrics between algorithms induced multicolinearities that made a single regression incorporating both identity and metrics infeasible.

To work around this problem, we built separate logistic regression models for users starting with each algorithm. With these models, we found that for users starting with the baseline algorithm, increased diversity in the list of recommendations increased the likelihood that they would try another algorithm ($p < 0.01$), and users starting with item-item were more likely to try another if the list of recommendations was more novel ($p < 0.01$).

We have similarly been unable to predict the user's final choice of recommender beyond the fact that it is a particular recommender. Comparing the popularity of the items recommended by user's initial and final recommenders, we saw that users tended to choose final recommenders with less popular items, but that seems to be linked to the fact that they tended to prefer SVD.

We also examined several properties of users to look for predictors of either behavior or of final preferred algorithm, but did not find any significant predictors. We considered the user's account age, the number of ratings they had provided, and the diversity of the movies they had rated, but none were effective for predicting the user's activity.

4.6 RQ7: Initial Algorithm and Retention

As noted in section 4.2.1, the user's initial recommender influenced whether or not a they changed algorithms, but did not directly affect their final choice of algorithm.

In section 4.1, we noted that users who make use of the recommender switching control were more likely to come back to the site. We also examined whether, for users who did not switch recommenders, the initial recommender condition affected retention, but found no effect. The initial recommender the user encountered in the new site did not, for our experiment, affect the user's likelihood to continue using the service.

To summarize our answer to RQ7, the primary effect of initial algorithm that we could observe was that it influenced whether the user would use the recommender selection control.

5. DISCUSSION

Users used the recommender-switching feature, and often changed to a different recommender than their initially-assigned one, even if they were already using a personalized recommender. While we have not seen any conclusive effect on user retention in our system, and do not have significant qualitative feedback on from users on the recommender-switching feature, it seems to have enough use to warrant continued inclusion and development. Other systems may also want to consider allowing the user to have more control over the way in which their recommendations are computed.

Users in the user forums did express interest in knowing what algorithms were used, or understanding the difference between the *Wizard* and *Warrior* recommenders; this information was withheld for experimental purposes, but we will be considering how best to present more information about the meaning of the user's choice.

Users who made use of the recommender selection control usually experimented with it a little bit early in their use of the system, and then left it set for the duration of their usage. They also came back for multiple sessions after leaving it set, suggesting some degree of satisfaction with their choice.

It is instructive to compare our results with those in our previous user study [7]. In that study, users did not have a measurable preference between item-item and SVD, in the context of reviewing a single 10-item recommendation list. Our results here suggest that more users do prefer SVD when they have the opportunity to interact with the algorithms in a longer-term context. However, many users still preferred the item-item recommender; there was no clear (near-unanimous) winner. Further, while we have noted a strong correlation between novelty and the user's preference (stronger than diversity), as was found in the aforementioned user study, the directionality of this correlation is reversed: users were more likely to stick with the algorithm that was also more novel (recommended less popular items). This provides additional evidence that negative impact of novelty is primarily concentrated in users' initial reactions, and that after they gain experience it may even be a positive influence in the user's satisfaction.

We tried several different ways to predict whether the user would switch from properties of the recommendations produced by their initial algorithm, and properties of the user (user account age, rating count, diversity of their rated items), but found very few significant predictors. If there are identifiable characteristics of users that predict the usefulness of different algorithms for providing their recommendations, we have yet to find them (or at least to demonstrate conclusively that identifiable differences are, indeed, the reason for particular user preferences).

6. CONCLUSIONS AND FUTURE WORK

We have reported on an experiment in which we gave users the ability to select the algorithm that would be providing their recommendations in a movie recommender application. We found that users made use of the control, typically experimenting with the different options to find a satisfactory recommender early on and keeping their choice for the remainder of their use of the system. We also found that SVD is the most preferred algorithm, followed somewhat closely by item-item, and finally the baseline and group-based recommenders. This shows that users do have a preference for personal recommendations, and while the preference between item-item and SVD was close, SVD was preferred by more users. We also observe a correlation between users taking advantage of the feature and long-term user retention. Giving users choice may promote their long-term use of the system.

This is an initial investigation of the dynamics of giving users control of the means by which their recommendations are computed. Previous systems have given users some control of their recommendations, such as Amazon's feature whereby users can indicate that they want certain items excluded from the data Amazon uses to provide recommendations. However, we are not aware of other services that allow end users to switch the entire recommender algorithm, or published research on how users interact with such a feature, aside from the implied ability to re-shop recommendation techniques in decentralized architectures such as the original GroupLens architecture [23].

There are a number of things needed to build on this work and carry it forward. First, it should be examined in other domains: do users benefit from the ability to select recommenders in other types of applications, such as book recommenders or tools for finding health information?

Second, we would like to consider more effective tools for allowing users to preview the recommendations. In our application, users could switch recommenders, but could not view recommendations side-by-side or 'try out' a recommender other than by selecting it and switching back. Users may use the feature in different ways if they can directly compare the output of different recommenders.

Thirdly, our switching mechanism was very course, allowing the user to swap out one recommender for another. But choice of recommender algorithm does not need to be an either-or decision, and there are many more nuanced decisions that could be made. There could be great potential in an interface that allows users to adjust the blend between different algorithms, and tweak the behavior of algorithms in other ways (e.g. adjusting the minimum neighbor count for a k-NN collaborative filtering, making it more or less conservative in its recommendations). Ziegler et al. [30] suggested a knob to allow users to adjust the amount of diversity they wanted added to their recommendations; we envision a panel of options that allow users to fine-tune their recommendations, ideally viewing the impact of these changes in real time. In order to make this feasible, we will not only need to develop useful interfaces, but identify the user-visible impact of different recommender tweaks so that the controls can be labeled in a manner that is understandable to end users, rather than 'Number of Latent Features'.

Providing users with control over the recommendation has the potential to significantly improve the user experience, and the sense of investment that users feel in the system, leading to better user retention and engagement. The control, and sense of transparency that comes with it, may also make users more comfortable with the system. A current popular complaint about recommender systems is that they are opaque algorithms with limited transparency and accountability for their outputs. User push-back against such algorithms changing and affecting their experience has been particularly pronounced in reactions to the Facebook emotional contagion study [16], or in concerns about the potential isolating effects of personalized information filtering [21]. Even if such concerns are not well-founded or even contrary to available evidence [20], users still need to feel that they can trust the system, and that trust may be damaged. Providing opportunities for the user to inspect and/or control the means by which their recommendations are computed may be valuable tool for the system to build and maintain their trust.

This paper represents our initial investigation into what happens when we give users the ability to control their recommendations. We look forward to further results, from our own work and that of others, on how to provide compelling, customizable recommendation experiences.

7. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grants IIS 0808692 and IIS 1017697.

8. REFERENCES

[1] Balabanović, M. and Shoham, Y. 1997. Fab: content-based, collaborative recommendation. *Commun. ACM*. 40, 3 (1997), 66–72.

[2] Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*. 12, 4 (Nov. 2002), 331–370.

[3] Chang, S., Harper, F.M. and Terveen, L. 2015. Using Groups of Items for Preference Elicitation in Recommender Systems. In *Proc. ACM CSCW '15* (2015), 1258–1269.

[4] Cook, R. and Kay, J. 1994. The justified user model: a viewable, explained user model. *Proceedings of the Fourth International Conference on User Modelling* (1994).

[5] Cremonesi, P., Garzottto, F. and Turrin, R. 2012. User Effort vs. Accuracy in Rating-based Elicitation. In *Proc. RecSys 2012* (2012), 27–34.

[6] Dooms, S. 2014. *Dynamic Generation of Personalized Hybrid Recommender Systems*. Ph.D thesis, Universiteit Gent.

[7] Ekstrand, M.D., Harper, F.M., Willemsen, M.C. and Konstan, J.A. 2014. User Perception of Differences in Recommender Algorithms. In *Proc. RecSys '14* (2014).

[8] Ekstrand, M., Ludwig, M., Konstan, J.A. and Riedl, J. 2011. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. In *Proc. RecSys '11* (2011), 133–140.

[9] Ekstrand, M. and Riedl, J. 2012. When recommenders fail: predicting recommender failure for algorithm selection and combination. In *Proc. RecSys '12* (2012), 233–236.

[10] Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM*. 35, 12 (1992), 61–70.

[11] Halfaker, A., Keyes, O., Kluver, D., Thebault-Spieker, J., Nguyen, T., Shores, K., Uduwage, A. and Warncke-Wang, M. 2014. User Session Identification Based on Strong Regularities in Inter-activity Time. *arXiv:1411.2878 [cs]*. (Nov. 2014).

[12] Kay, J. 2006. Scrutable Adaptation: Because We Can and Must. *Adaptive Hypermedia and Adaptive Web-Based Systems*. V.P. Wade, H. Ashman, and B. Smyth, eds. Springer Berlin Heidelberg. 11–19.

[13] Kelly, D. and Belkin, N.J. 2001. Reading Time, Scrolling and Interaction: Exploring Implicit Sources of User Preferences for Relevance Feedback. In *Proc. SIGIR '01* (2001), 408–409.

[14] Knijnenburg, B., Willemsen, M., Gantner, Z., Soncu, H. and Newell, C. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*. 22, 4-5 (Oct. 2012), 441–504.

[15] Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R.M. 2008. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*. 18, 1 (Jul. 2008), 140–181.

[16] Kramer, A.D.I., Guillory, J.E. and Hancock, J.T. 2014. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*. 111, 24 (Jun. 2014), 8788–8790.

[17] Levi, A., Mokryn, O., Diot, C. and Taft, N. 2012. Finding a Needle in a Haystack of Reviews: Cold Start Context-based Hotel Recommender System. In *Proc. RecSys '12* (2012), 115–122.

[18] McNee, S., Kapoor, N. and Konstan, J.A. 2006. Don't Look Stupid: Avoiding Pitfalls When Recommending Research Papers. In *Proc. CSCW '06* (2006), 171.

[19] Netflix Update: Try This at Home: 2006. *http://sifter.org/~simon/journal/20061211.html*. Accessed: 2010-04-08.

[20] Nguyen, T.T., Hui, P.-M., Harper, F.M., Terveen, L. and Konstan, J.A. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proc. WWW '14* (2014), 677–686.

[21] Pariser, E. 2011. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. Penguin.

[22] Paterek, A. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup and Workshop 2007* (Aug. 2007).

[23] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In *Proc. ACM CSCW '94* (1994), 175–186.

[24] Rich, E. 1979. User modeling via stereotypes. *Cognitive Science*. 3, 4 (Oct. 1979), 329–354.

[25] Said, A., Fields, B., Jain, B.J. and Albayrak, S. 2013. User-centric Evaluation of a K-furthest Neighbor Collaborative Filtering Recommender Algorithm. In *Proc. ACM CSCW '13* (2013), 1399–1408.

[26] Sarwar, B., Karypis, G., Konstan, J. and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. WWW '01* (2001), 285–295.

[27] Sill, J., Takacs, G., Mackey, L. and Lin, D. 2009. Feature-Weighted Linear Stacking. *arXiv:0911.0460*. (Nov. 2009).

[28] Tintarev, N. 2007. Explanations of recommendations. In *Proc. RecSys '07* (2007), 203–206.

[29] Vig, J., Sen, S. and Riedl, J. 2012. The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM Trans. Interact. Intell. Syst.* 2, 3 (Sep. 2012), 13:1–13:44.

[30] Ziegler, C.-N., McNee, S., Konstan, J.A. and Lausen, G. 2005. Improving Recommendation Lists through Topic Diversification. In *Proc. WWW '05* (2005), 22–32.