

Recommender Popularity Controls: An Observational Study

F. Maxwell Harper
University of Minnesota
max@umn.edu

ABSTRACT

We describe an observational study of a recommender system that provides users with direct control over their personalization. Specifically, we allow users to tune a movie recommender towards more or less popular content. We report on 14 months of usage, which includes 6,846 users who visited the interface at least once. We find, surprisingly, that the popularity of items a user has interacted with historically is a poor predictor of their use of this interface.

KEYWORDS

recommender systems; popularity; observational study.

1 INTRODUCTION

In prior work [3], we reported on a lab study of how users interact with unlabelled controls that modify the popularity or recency of a movie recommender system. Among several results, we found that users who adjusted popularity reported much higher satisfaction with their recommendations. While most users tuned their recommender towards more popular content, there was a wide range of preferences among users. Therefore, we built a *popularity tuner* interface into the live system to develop a better understanding of how this feature would be used in practice, and to collect more data to experiment with modeling users' preferred settings.

2 OBSERVATIONAL STUDY

Our platform, MovieLens (<http://movielens.org>), allows users to choose among multiple recommendation algorithms [1]; we enable the popularity interface on the most popular two: item-item collaborative filtering [4] and Funk SVD [2]. Our recommendation algorithm in each case (described in detail in [3]) is a linear blend of predicted star rating with popularity. We use number of ratings in the last year as our popularity metric. The default setting for both algorithms is a blend of 95% predicted rating with 5% popularity, which we chose based on the results of our lab study.

The popularity tuner interface is shown in Figure 1. On each more/less popular click action, we use a binary search to locate new weights that will replace 4 of the user's top 20 recommendations. For example, a user clicking "more popular" triggers a binary search that might yield a new blend of 92.1% predicted rating with 7.9% popularity. We emphasize the changes visually in the interface to help users understand the result of their actions. To encourage use of the feature, we added a "configure" button next to the user's list of top recommendations at the top of the home page.

We deployed the feature on April 11, 2016, and collected data through June 15, 2017. We consider two groups for analysis. First, for examining overall use of the popularity tuner, we look at all users who visited the interface one or more times (*visited_sample*,

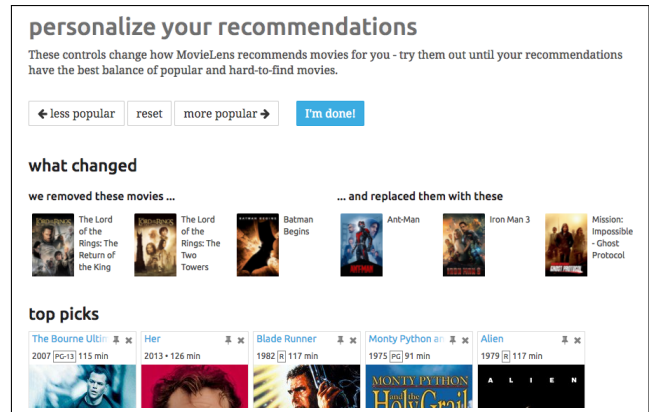


Figure 1: Screenshot of the popularity tuner interface. When a user clicks “more popular” or “less popular”, the list of 20 movies under “top picks” is immediately refreshed and his or her preferences are saved (and reflected site-wide).

$N=6846$, median logins=11, median ratings=168). Second, for examining users' preferred popularity settings, we look at users who took at least one action (*interacted_sample*, $N=4349$, median logins=14, median ratings=205). These samples are skewed toward power users as compared with all users that logged in during this period ($N=31371$, median logins=3, median ratings=39).

Over the span of this observational study (or even the span of a single session), users might change settings multiple times. In fact, users in the *visited_sample* opened the tuning interface 24,657 times (3.6 times per user); users in the *interacted_sample* visited the page 19,465 times (4.5 times per user). Unless otherwise noted, user-level analysis examines the user's *final configuration* at the time of analysis. For example, a user might try the tuner interface three times and explore the system in between each change; our user-level analysis considers only the most recent configuration.

2.1 Descriptive Statistics

The distribution of per-user activity in online systems is often modeled by a power law distribution [5]; use of the tuner interface is no exception. We find that a few users use the feature heavily, while the vast majority of users pay it little attention — see Table 1 for an overview. For instance, 82% of users take 5 or fewer more/less/reset actions and 78% of sessions are less than a minute in duration. On the other hand, power users have explored the feature thoroughly — one user has taken more than 5,500 actions.

Within the *interacted_sample*, the most common popularity setting is the default one (1,390 users, 32.0% of sample). That is, after exploring one or more new settings, many users return to the default, either by clicking the “reset” button, or by sending the

percentile	# actions	# sessions	session length (sec.)
25th	0	1	3
50th (median)	1	2	5
80th	5	5	67
95th	14	10	251
99th	35	20	1428

Table 1: Descriptive statistics of activity levels by percentile in the observational study across users in the *visited_sample*. # actions and # sessions are user-level variables, while session length is a session-level variable with possibly multiple observations per user.

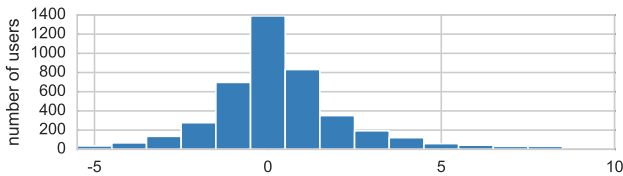


Figure 2: Histogram of users' final configurations, represented as number of clicks from the default (0). E.g., an value of two means two *net* clicks on “more popular”.

variable	estimate	
intercept	-1.160000	***
# logins	0.000012	
# ratings	0.000062	***
predictor==SVD	0.068710	***
mean pp1y, rated	0.818100	**
mean pp1y, wishlisted	0.064480	
mean pp1y, clicked	0.401600	***

Table 2: Summary of our regression model to predict users' choices, $R^2 = 0.04$. Statistically significant p-values indicated as * ($p < 0.001$) or ** ($p < 0.01$).**

same number of “more popular” and “less popular” actions. More users (1,714, 39.4%) end with a setting more popular than default, as compared with the number that end less popular than default (1,245, 28.6%). See Figure 2 for a visualization of this data, represented as number of steps from the default.

2.2 Analysis: Predicting Settings

It is possible that we can predict users' preferred popularity settings from their behavioral data. To explore this idea, we construct regression models using popularity-related predictor variables. In particular, we include a metric — “one-year popularity percentile” (*pp1y*) — that exactly mirrors the variable used in the popularity blending algorithm. We measure *pp1y* of rated, wishlisted, and clicked items to capture a user's propensity to interact with more or less popular content. In addition, we use variables that capture the user's number of logins, number of ratings, and chosen prediction algorithm.

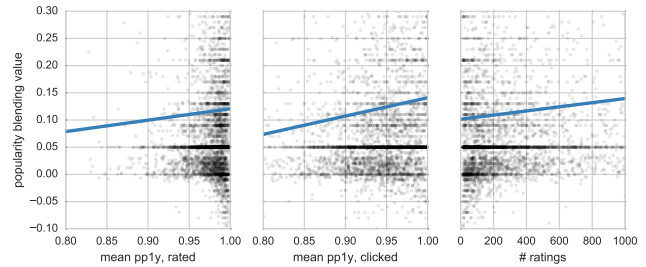


Figure 3: Scatterplots with regression lines showing the relationship between user-level variables and popularity blending values.

Our regression model has several statistically significant variables, but overall, it is a poor fit with users' popularity blending choices (adjusted R-squared=0.04); see Table 2 for a summary and Figure 3 for a visualization of several variables. The model indicates that the popularity of a user's rated and clicked-on movies is positively correlated with the user's final configuration value. Also, the model shows a relationship between a user's choice of prediction algorithm and their chosen popularity blending weight (means: item-item=0.09, funkSVD=0.15). The model shows that number of ratings is positively correlated with the popularity setting, but with a tiny effect size; number of logins and popularity of wishlisted content are not statistically significant.

3 CONCLUSION

Thousands of users have tried the popularity tuner feature to configure their movie recommender. They have used the feature in very different ways, and it proves difficult to predict their choices based on behavioral data. The poor overall fit of our regression model underscores the potential helpfulness of giving control to users. While it is possible that our model is too preliminary, or that our interface does not sufficiently elicit users' ideal preferences, it is also possible that users have fickle preferences for their recommenders that are difficult to get right through modeling alone.

REFERENCES

- [1] Michael D. Ekstrand, Daniel Kluver, F. Maxwell Harper, and Joseph A. Konstan. 2015. Letting Users Choose Recommender Algorithms: An Experimental Study. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 11–18. DOI: <http://dx.doi.org/10.1145/2792838.2800195>
- [2] Simon Funk. 2006. Netflix update: Try this at home. (Dec. 2006). <http://sifter.org/~simon/journal/20061211.html>
- [3] F. Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen. 2015. Putting Users in Control of Their Recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 3–10. DOI: <http://dx.doi.org/10.1145/2792838.2800179>
- [4] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295. DOI: <http://dx.doi.org/10.1145/371920.372071>
- [5] Dennis M. Wilkinson. 2008. Strong Regularities in Online Peer Production. In *Proceedings of the 9th ACM Conference on Electronic Commerce (EC '08)*. ACM, New York, NY, USA, 302–309. DOI: <http://dx.doi.org/10.1145/1386790.1386837>