Recommender Response to Diversity and Popularity Bias in User Profiles

Sushma Channamsetty[1]
[1]Department of Computer Science
Texas State University
San Marcos, TX USA
sushmachannamsetty@gmail.com

Michael D. Ekstrand[1,2]
[2]People and Information Research Team
Department of Computer Science,
Boise State University, Boise, ID USA
michaelekstrand@boisestate.edu

Abstract

Recommender system evaluation usually focuses on the overall effectiveness of the algorithms, either in terms of measurable accuracy or ability to deliver user satisfaction or improve business metrics. When additional factors are considered, such as the diversity or novelty of the recommendations, the focus typically remains on the algorithm's overall performance. We examine the relationship of the recommender's output characteristics – accuracy, popularity (as an inverse of novelty), and diversity – to characteristics of the user's rating profile. The aims of this analysis are twofold: (1) to probe the conditions under which common algorithms produce more or less diverse or popular recommendations, and (2) to determine if these personalized recommender algorithms reflect a user's preference for diversity or novelty. We trained recommenders on the MovieLens data and looked for correlation between the user profile and the recommender's output for both diversity and popularity bias using different metrics. We find that the diversity and popularity of movies in users' profiles has little impact on the recommendations they receive.

Introduction

Recommender systems (Ekstrand, Riedl, and Konstan 2010; Adomavicius and Tuzhilin 2005) are widely deployed to assist users in selecting or purchasing items of interest in many domains such as music, movies, books, and news, and can be found in a wide array of predominantly online services. Many of these systems are personalized, suggesting items expected to be of particular interest to the specific user using the service based on their interests.

Recommender evaluation has historically focused on the accuracy or effectiveness of the recommender system: can it accurately predict missing ratings or identify items the user enjoys, increase sales or retention, or satisfy the user's desires. The hope with offline evaluations of accuracy is that they will predict online effects on user behavior and satisfaction (Gunawardana and Shani 2009).

However, accuracy does not tell the whole story of a recommender's impact. Tuning the recommender to produce most accurate recommendations might restrict the user from having useful recommendations (McNee, Riedl, and Konstan 2006). Two dimensions to consider are *diversity* and *novelty* (Hurley and Zhang 2011; Vargas and Castells 2011).

Most of this work has remained focused on the recommender's overall performance, aggregating across users. Different recommenders, however, do not treat all users equally (Ekstrand and Riedl 2012). In this work, we examine the recommender's behavior for individual users, looking at how the characteristics of the recommendations users receive are distributed and user profile characteristics that may influence the recommender's output. We seek to understand *when* a recommender is more or less diverse, for example, in its recommendations.

To that end, we raise the following research questions:

1. Does the users' input profile change the recommender response profile?
2. Do different recommender algorithms propagate the change in users' input profile differently?
3. How does the accuracy of the recommender correlate with diversity or popularity bias of the user?

We examine these questions across 5 recommender algorithms – three collaborative filters, a content-based filter, and a baseline – using the MovieLens 10M data set.

Methodology

We generate and examine recommendation lists for users in the MovieLens 10M data set (Harper and Konstan 2015), consisting of 10M ratings and 100K tag applications from

72K users on 10K movies on the MovieLens movie recommendation service. We combined this data with the Tag Genome (Vig, Sen, and Riedl 2012), a dense matrix of relevance scores for 1,100 tags over 10,000 movies.

Experimental Configuration

We used the LensKit recommender toolkit (Ekstrand et al. 2011), version 3.0-M2 for our experiment. The publicly-available source code to re-run the experiment in LensKit (and the subsequent analysis in R (R. Core Team 2012))[1] contains the full configuration details, but a summary of relevant experimental settings follows:

- We took 5 disjoint samples of 1000 users each. This strategy allows us to test on a large number of users, but not so many that expensive algorithms are intractable.
- For each user in each sample, we randomly selected 5 of their ratings as test items for measuring recommender accuracy; the remaining ratings from the in-sample users along with all ratings from out-of-sample users form the recommender's training data.
- We generated 100-item top-*N* lists from the set of all items not rated by the target user. In post-analysis we truncated these lists to 10 and 25 items.
- We measured prediction and recommendation accuracy with RMSE and Mean Average Precision (MAP), respectively. MAP was applied to the full 100-item lists.

Algorithms

We used five common algorithms for our analysis:

- **UserUser:** User-user collaborative filtering (Herlocker, Konstan, and Riedl 2002) with 30 neighbors and cosine similarity over user-mean-adjusted ratings (Ekstrand et al. 2011).
- **FunkSVD:** approximate matrix factorization (Funk 2006) with 40 factors and 125 iterations per feature.
- **ItemItem:** item-based collaborative filtering (Sarwar et al. 2001) with 20 neighbors and cosine similarity over item-mean-centered ratings (Ekstrand et al. 2011).
- **CBF:** An item-based content filter using movie tag data (Ekstrand and Riedl 2012), implemented with Apache Lucene and using 20 neighbors.
- **Popularity:** recommends the most frequently-rated items.

The configurations were derived from values found to work well on MovieLens data in previous experiments with LensKit (Ekstrand et al. 2011).

Metrics

For each test user in the experiment, we measured their training ratings (input profile) and the recommendation lists produced by each algorithm using the following metrics:

- **Intra-List Similarity** (ILS) (Ziegler et al. 2005) using Pearson correlation over tag genome vectors (Vig, Sen, and Riedl 2012) to measure diversity (lower ILS values are more diverse). We ignored movies not present in the tag genome.
- **Mean Popularity Rank** to measure popularity, which we use as a proxy for the inverse of novelty. The most-rated item has a popularity rank of 1.
- **Mean Average Precision** (recommendation lists only) to assess recommender list quality.

Results and Discussion

In this section, we walk through our findings for each of the profile characteristics we consider. The key correlations are summarized in

and described in more detail below. Throughout, we are comparing each algorithm's recommendations for each user with that user's *profile* of rated items.

Table 1: Correlation (Pearson's r) between characteristics of user profile and 25-item output lists for each algorithm.

Algorithm	Popularity	Diversity
Popularity	0.1653	-0.0590
ItemItem	0.0402	0.0565
CBF	0.0114	0.1324
FunkSVD	-0.1453	0.0885
UserUser	-0.2854	-0.0038

Popularity

Figure 1 shows the distribution of popularity bias in user profiles, and Figure 2 shows the output response of each recommender.

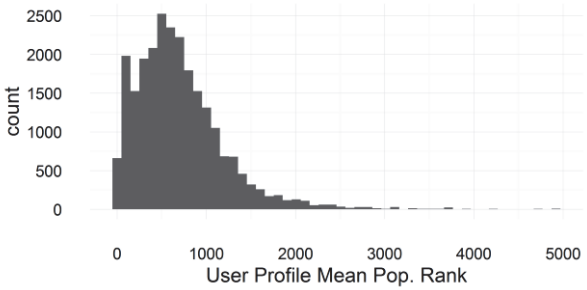

Figure 1:Popularity bias of user profiles.

The Popularity recommender naturally keeps the recommendation list popularity as high as possible, and does not respond to user input profile popularity.

The other algorithms all tend to recommend items less popular than those in the user's profile (evident as they fall almost entirely above the red dashed profile line), although to varying degrees. This novelty bias is expected if the recommenders are to effectively help users explore the long tail

[1] http://works.bepress.com/michael-ekstrand/18/

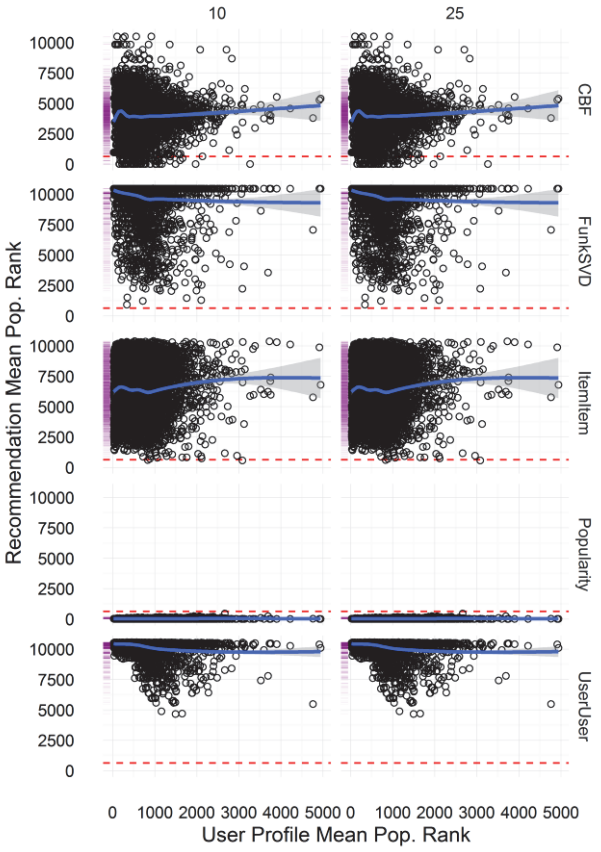

Figure 2: Average item popularity for recommendation lists plotted against user profiles. Rug plots on the left indicate marginal distribution of recommendation list popularity. Red dashed line indicates median user profile popularity for reference.

of the item space. ItemItem and CBF both have high variance in the popularity of their recommendation lists, while UserUser and FunkSVD consistently select unpopular items (as can be seen from their rug plots). The difference between user profile popularity and recommended item popularity is significant for all algorithms (paired t-test, $p < 10^{-6}$).

Table 1 shows the correlation between user and recommendation profile popularity for 25-item lists produced by each algorithm. FunkSVD and UserUser are noticeably *negatively* correlated: if a user likes popular items, these algorithms are more likely to recommend unpopular items. (Popularity's relatively high correlation can be disregarded because there is so little variance to actually explain).

We did not find user profile popularity to be a useful predictor of top-N MAP; while linear models achieve statistical significance, they fit poorly (the best model, for *Popularity*, achieves $R^2 = 0.07$).

Diversity

Figure 3 shows the distribution of the intra-list similarity of user profiles. We can see that there is not a large spread

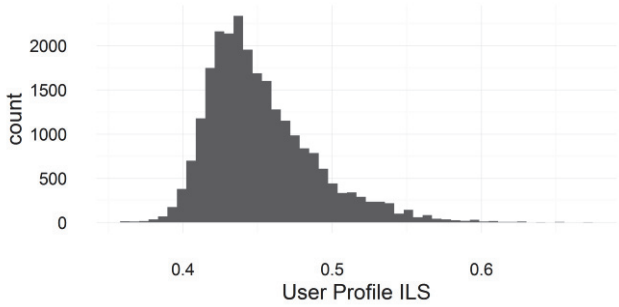

Figure 3: Distribution of user profile diversity.

in the input profiles. Figure 4 shows the diversity of 10- and 25-item recommendation lists. The increased sparsity for 10-item lists, particularly for UserUser and FunkSVD, is due to these algorithms' propensity to recommend obscure items that are less likely to be included in the tag genome.

Overall, all algorithms except content-based filtering tended to produce recommendation lists with lower diversity than users' input profiles; these differences are all highly significant (paired t-test, $p < 10^{-6}$). However, while the overall trend was towards less (or for content-based fil-

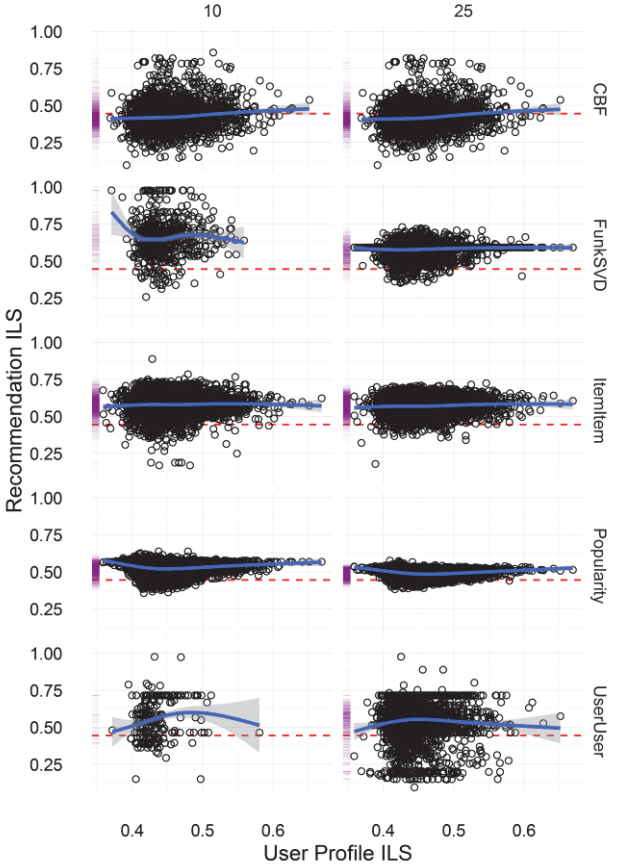

Figure 4: Diversity of recommendation lists against user profile diversity; lower is more diverse

tering, slightly more) diversity, there was very little correlation between individual users' input profile diversity and the diversity of recommendations they received. As seen in , the algorithms with that produced the highest correlation in their recommendations were CBF (Pearson's $r = 0.1324$) and FunkSVD ($r = 0.0885$). Linear models did not fit well; the best one, for CBF, achieved an R^2 of 0.0175.

User profile diversity was also not useful for predicting recommender accuracy.

Conclusions and Future Work

We examined the diversity and popularity of the movies that users have rated, and examined how recommender outputs respond based on those characteristics. We found that the algorithms we considered, particularly collaborative filtering approaches do not propagate very much — if any — of the users' observable preference for popularity or diversity into their recommendations. This suggests that common individual recommender algorithms may be missing a component of personalization. This is not entirely surprising, as the algorithms in question consider individual items and do not have a concept of the entire list or set being recommended. However, it suggests a weakness in personalization using common algorithms. The ideal personalized recommender should capture and respond to many aspects of the user's personalization.

This may or may not be a problem in practice, and indeed it may make recommenders resilient to other failure conditions. For example, this obliviousness to the user's breadth of taste may mitigate filter bubble effects.

There are several opportunities to build on this work:

- Resample data sets to produce user profiles with different biases.
- Consider set-, list-, or rank-aware algorithms such as Bayesian Personalized Ranking (Rendle et al. 2009).
- Develop algorithms to explicitly measure and respond to user profile characteristics.
- Repeat analysis on additional domains.

We hope this work provides researchers and practitioners with useful insight into the behavior of their algorithms.

References

Adomavicius, G., and A. Tuzhilin. 2005. "Toward the next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering* 17 (6): 734–49. doi:10.1109/TKDE.2005.99.

Ekstrand, Michael D., Michael Ludwig, Joseph A. Konstan, and John T. Riedl. 2011. "Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit." In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 133–140. RecSys '11. New York, NY, USA: ACM. doi:10.1145/2043932.2043958.

Ekstrand, Michael, and John Riedl. 2012. "When Recommenders Fail: Predicting Recommender Failure for Algorithm Selection and Combination." In *Proceedings of the 6th ACM Conference on Recommender Systems*, 233–236. ACM. doi:10.1145/2365952.2366002.

Ekstrand, Michael, John Riedl, and Joseph A. Konstan. 2010. "Collaborative Filtering Recommender Systems." *Foundations and Trends® in Human-Computer Interaction* 4 (2): 81–173. doi:10.1561/1100000009.

Funk, Simon. 2006. "Netflix Update: Try This at Home." Blog. *The Evolution of Cybernetics*. December 11. http://sifter.org/~simon/journal/20061211.html.

Gunawardana, Asela, and Guy Shani. 2009. "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks." *J. Mach. Learn. Res.* 10: 2935–62. http://jmlr.org/papers/v10/gunawardana09a.html.

Harper, F. Maxwell, and Joseph A. Konstan. 2015. "The MovieLens Datasets: History and Context." *ACM Trans. Interact. Intell. Syst.* 5 (4): 19:1–19:19. doi:10.1145/2827872.

Herlocker, Jonathan, Joseph A. Konstan, and John Riedl. 2002. "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms." *Inf. Retr.* 5 (4): 287–310. doi:10.1023/A:1020443909834.

Hurley, Neil, and Mi Zhang. 2011. "Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation." *ACM Trans. Internet Technol.* 10 (4): 14:1–14:30. doi:10.1145/1944339.1944341.

McNee, Sean M., 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*, 1097–1101. ACM. doi:10.1145/1125451.1125659.

R. Core Team. 2012. *A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria*. http://www.R-project.org/.

Rendle, Steffen, 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*, 452–461. UAI '09. Arlington, Virginia, United States: AUAI Press. http://dl.acm.org/citation.cfm?id=1795114.1795167.

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Reidl. 2001. "Item-Based Collaborative Filtering Recommendation Algorithms." In *ACM WWW '01*, 285–95. ACM. doi:10.1145/371920.372071.

Vargas, Saúl, and Pablo Castells. 2011. "Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems." In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 109–116. ACM. doi:10.1145/2043932.2043955.

Vig, Jesse, Shilad Sen, and John Riedl. 2012. "The Tag Genome: Encoding Community Knowledge to Support Novel Interaction." *ACM Trans. Interact. Intell. Syst.* 2 (3): 13:1–13:44. doi:10.1145/2362394.2362395.

Ziegler, Cai-Nicolas, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. "Improving Recommendation Lists Through Topic Diversification." In *Proceedings of the 14th International Conference on World Wide Web*, 22–32. WWW '05. New York, NY, USA: ACM. doi:10.1145/1060745.1060754.