

The Unfairness of Popularity Bias in Recommendation*

Himan Abdollahpouri
University of Colorado Boulder
Boulder, USA
himan.abdollahpouri@colorado.edu

Robin Burke
University of Colorado Boulder
Boulder, USA
robin.burke@colorado.edu

Masoud Mansoury
Eindhoven University of Technology
Eindhoven, Netherlands
m.mansoury@tue.nl

Bamshad Mobasher
DePaul University
Chicago, USA
mobasher@cs.depaul.edu

ABSTRACT

Recommender systems are known to suffer from the popularity bias problem: popular (i.e. frequently rated) items get a lot of exposure while less popular ones are under-represented in the recommendations. Research in this area has been mainly focusing on finding ways to tackle this issue by increasing the number of recommended long-tail items or otherwise the overall catalog coverage. In this paper, however, we look at this problem from the users' perspective: we want to see how popularity bias causes the recommendations to deviate from what the user expects to get from the recommender system. We define three different groups of users according to their interest in popular items (Niche, Diverse and Blockbuster-focused) and show the impact of popularity bias on the users in each group. Our experimental results on a movie dataset show that in many recommendation algorithms the recommendations the users get are extremely concentrated on popular items even if a user is interested in long-tail and non-popular items showing an extreme bias disparity.

KEYWORDS

Recommender systems; Popularity bias; Fairness; Long-tail recommendation

1 INTRODUCTION

Recommender systems have been widely used in a variety of different domains such as movies, music, online dating etc. Their goal is to help users find relevant items which are difficult or otherwise time-consuming to find in the absence of such systems.

Different types of algorithms are being used for recommendation depending on the domain or other constraints such as the availability of the data about users or items. One of the most widely-used classes of algorithms for recommendation is collaborative filtering. In these algorithms the recommendations for each user are being generated using the rating information from other users and items. Unlike other types of algorithms such as content-based recommendation, collaborative algorithms do not use the content information.

One of limitations of collaborative recommenders is the problem of popularity bias [1, 7]: popular items are being recommended too

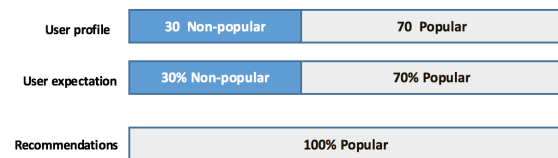


Figure 1: The inconsistency between user's expectation and the given recommendations

frequently while the majority of other items do not get the deserved attention.

Popularity bias could be problematic for a variety of different reasons: Long-tail (non-popular) items are important for generating a fuller understanding of users' preferences. Systems that use active learning to explore each user's profile will typically need to present more long tail items because these are the ones that the user is less likely to have rated, and where user's preferences are more likely to be diverse [16, 18].

In addition, long-tail recommendation can also be understood as a social good. A market that suffers from popularity bias will lack opportunities to discover more obscure products and will be, by definition, dominated by a few large brands or well-known artists [10]. Such a market will be more homogeneous and offer fewer opportunities for innovation and creativity.

In this paper, however, we look at the popularity bias from a different perspective: the users'. Look at figure 1 for example: the user has rated 30 long-tail (non-popular) items and 70 popular items. So it is reasonable to expect the recommendations to keep the same ratio of popular and non-popular items. However, most of the recommendation algorithms produce a list of recommendations that is over-concentrated on popular items (often it is even close to 100% popular items). In this paper, we show how different recommendation algorithms are propagating this bias differently for different users. In particular, we define three groups of users according to their degree of interest towards popular items and show the bias propagation for some groups is higher than the others. For instance, the niche users (users with the lowest degree of interest towards popular items) are affected the most by this bias. We also show these niche users and, in general users with lower interest in popular items, are more active (they have rated more items) in the system and therefore they should be considered as important stakeholders

*Copyright 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). Presented at the RMSE workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.

whose needs should be addressed properly by the recommender system. However, looking at several recommendation algorithms, we see these users are not getting the type of recommendations that they expect to get. In summary, here are the important research questions we would like to answer in this paper:

- **RQ1:** How much are different individuals or groups of users interested in popular items?
- **RQ2:** How do algorithms' popularity biases impact users with different degrees of interest in popular items?

To answer these questions we look at the movieLens 1M dataset and analyze the users' profiles in terms of their rating behavior. We also compare several recommendation algorithms with respect to their general popularity bias propagation and the extent to which this bias propagation is affecting different users.

2 RELATED WORK

The problem of popularity bias and the challenges it creates for the recommender system has been well studied by other researchers [6, 8, 17]. Authors in the mentioned works have mainly explored the overall accuracy of the recommendations in the presence of long-tail distribution in rating data. Moreover, some other researchers have proposed algorithms that can control this bias and give more chance to long-tail items to be recommended [3–5, 14].

In this work, however, we focus on the fairness of recommendations with respect to users' expectations. That is, we want to see how popularity bias in the input data is causing the recommendations to deviate from the actual interests of users with respect to how many popular items they expect to see in the recommended list. We call this deviation unfair because it is caused by the existing bias in data and also the algorithmic bias in some of the recommendation algorithms. A similar work to ours is [19] where author proposed the idea of calibration: the recommendations should be consistent with the spectrum of items a user has rated. For example, if a user has rated 70% action movies and 30% romance, it is expected to see the same pattern in the recommendations. Our work is different as we do not use content information to see how different algorithms are calibrated. In fact, Our work could be considered as an explanation for the problem discussed relative to genre calibration, namely that genre distortion in recommendations could be caused by different levels of popularity across genres.

Moreover, the concept of fairness in recommendation has been also gaining a lot of attention recently [13, 20]. For example, finding solutions that removes algorithmic discrimination against users belong to a certain demographic information [21] or making sure items from different categories (e.g. long tail items or items belong to different providers)[9, 15] are getting a fair exposure in the recommendations. Our definition of fairness in this paper is more related to the *calibration fairness* [19] which means users should get the recommendations that are close to what they are expecting to get based on the type of items they have rated.

And finally, Jennach et al. [12] compared different recommendation algorithms in terms of accuracy and popularity bias. In that paper they observed some algorithms concentrate more on popular items than the others. In our work, we are mainly interested in seeing the popularity bias from the users' expectations perspective.

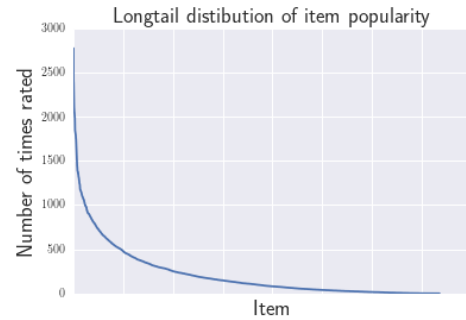


Figure 2: The long-tail of item popularity in rating data

3 POPULARITY BIAS IN DATA

For all of the following sections we use the MovieLens 1M dataset for our analysis which contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users [11].

Rating data is generally skewed towards more popular items—there are a few popular items with thousands of ratings and the other majority of the items combined have fewer ratings than those few popular ones. Figure 2 shows the long-tail distribution of item popularity in the famous MovieLens 1M dataset but the same distribution can be seen in other datasets as well.

Due to this imbalance property of the rating data, often algorithms inherit this bias and, in many cases, increase it by over-recommending the popular items and, therefore, giving them a higher opportunity of being rated by more users and, as a result, making them rated by more users and this goes on again and again—*the rich get richer and the poor get poorer*.

That by itself is very problematic and needs to be addressed using proper algorithms. However, what is often ignored is the fact that there are many users who are actually interested in non-popular items and they expect to get recommendations beyond some extremely popular items and blockbusters. What is worth noting is that users in a recommender system platform all should be considered as different stakeholders and their satisfaction should be taken care of by the algorithm [2]. An algorithm that is only capable of addressing the needs of a certain group of users (those whose interest is primarily concentrated on popular items) will be in the risk of losing the interest of the rest of the users if their needs are not taken care of.

In this paper we compare several recommendation algorithms and see how they perform in terms of keeping the ratio of popular and non-popular items according to what the users expect from the recommendations (i.e. the same ratio in their profile)

3.1 User Propensity for Popular Items

As we mentioned earlier, there might be users who are interested in non-popular items and therefore the recommender algorithm should be able to address the needs of those users as well. In this section, we analyze the degree of interest that different types of users have towards popular items. Figure 3 shows the ratio of popular items in the users' profiles.

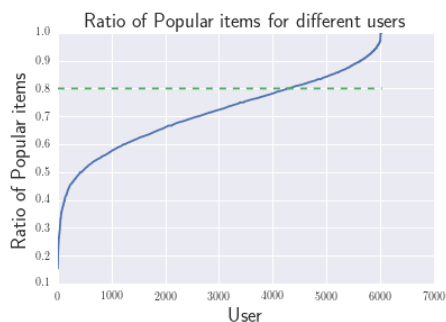


Figure 3: The ratio of popular items in users' profiles in MovieLens dataset

The users have been sorted first based on the ratio of popular items in their profile then the data has been plotted. We can see that more than 4000 users are actually interested in at least 20% non-popular items (from x-axis 0 to 4000). Even the other 2000 users also have some interest in non-popular items as the ratio of popular items, except for a dozen of users, is still between 80% and 100% meaning there is around 10-20% non-popular items in their profile.

3.2 Users Profile Size and Popularity Bias

Before going any further, we want to see if a user's profile size has any correlation with the number of popular items she has in her profile. In this paper, we define an item to be popular if its popularity value falls within the top 20% of item popularities. Figure 4-a shows a strong correlation between profile size and the number of popular items in users' profile which is kind of obvious due to the fact that having more items in a user's profile increases the likelihood of observing a popular item in it. So, we also show the correlation between the ratio of popular items in users' profiles and the profile size. Figure 4-b shows this correlation is negative meaning the larger the profile size, the smaller the ratio of popular items is in a user's profile. This effect is to be expected. There are a small number of popular items. As a user's profile gets larger, they will eventually run out of popular items to rate and therefore must include more and more non-popular items. Figure 4-c shows another perspective of popularity of the rated items in users' profile which is the average popularity of the rated items by each user. The popularity of each item is simply the ratio of users who have rated that item. Again in this figure we can see a negative correlation between profile size and the average popularity of rated items by a user indicating the more items a user has rated (larger profile size) the smaller the average popularity of her rated items is.

These results provide interesting insights about users with lower propensity to rate popular items: these users also tend to interact with the system more and provide more ratings. Therefore, they tend to have a more profound effect on the performance of the recommender system and their contributions should not be ignored.



(a) Correlation of profile size and the number of popular items in user profile



(b) Correlation of profile size and the ratio of popular items in user profile



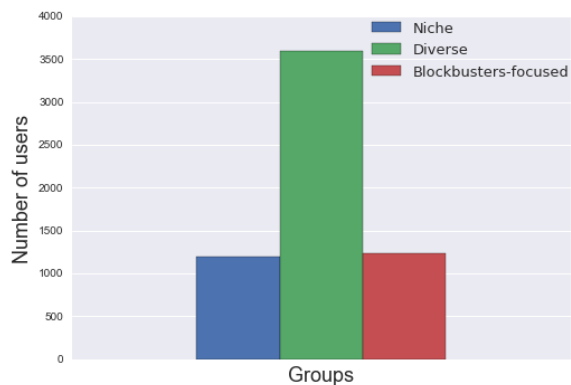
(c) Correlation of profile size and the average popularity of items in user profile

Figure 4: The correlation between the profile size and a) the number of popular items, b) the ratio of popular items and c) the average popularity of rated items in users' profiles

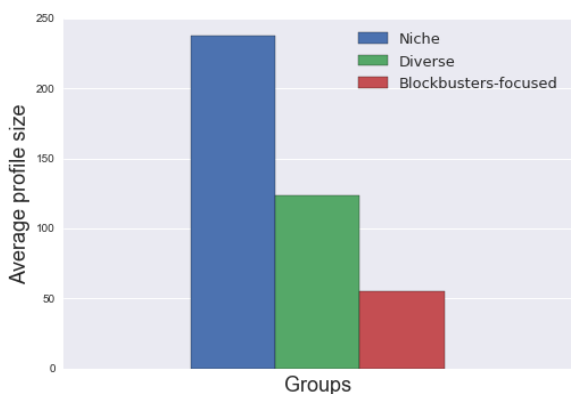
4 DIFFERENT GROUPS OF USERS IN TERMS OF PROPENSITY FOR POPULARITY

In this section, we want to define three groups of users based on the ratio of popular items in their profile. That is, how interested they are in popular items.

- **Niche Users (N):** These are the bottom 20% of users in terms of the ratio of popular items in their profile. For these users, More than half of their profile consists of non-popular (long-tail) items.



(a) The number of users in each group



(b) The average profile size in different user groups

Figure 5: a) The number of users in three different groups N, D and B and b) The average profile size of the users in these groups

- **Blockbuster-focused Users (B):** These are the top 20% of users in terms of the ratio of popular items in their profile. These users, on average, have more than 85% popular items in their profile.
- **Diverse Users (D):** These are the users that are not *Niche* neither are they *Blockbuster-focused*. So the rest of the users fall within this category.

Figure 5-a shows the population of each of these different groups in MovieLens 1M dataset. As you can see the majority of the users are in group *D* (i.e. diverse) with more than 3500 users. In addition, group *N* and *D* combined take up more than 4500 of the user population (around 75% of the users).

Figure 5-b shows the average profile size of different user groups. Consistent with our previous analysis, niche users have larger profiles followed by diverse users. The Blockbuster-focused group has the shortest average profile size.

5 ALGORITHMIC PROPAGATION OF POPULARITY BIAS

As we observed in previous section, there is a bias in the rating data: certain items are rated very frequently while many others are rated by fewer users. In this section, we show how different recommendation algorithms propagate this bias into their recommendations. We first look at their general performance without paying attention to how they perform for different users or groups of users. We tested several algorithms including *User KNN*, *Item KNN*, *SVD++* and *Biased Matrix factorization* for our experiments. For all these algorithms we set aside 80% of the rating data as training set and the remaining 20% as the test set. For each user in test set, a list of 10 items are recommended. Moreover, we tuned all these algorithms so they can achieve a similar precision (around 0.1) for a fair comparison of their popularity bias propagation. In addition, we also used two simple *Most popular* and *Random* algorithms. Figure 6 shows the correlation between the number of times an item is rated and the number of times it is recommended by different algorithms.

It can be seen that in all algorithms except for the *random* there are many items that are almost never recommended (the items fall on the horizontal tail of the scatter plot). In addition, the *Most popular* algorithm seems to have the strongest correlation between the number of times an item is being rated and the number of times it is recommended, as it was expected, followed by *User KNN* and *Item KNN* which also have a strong correlation. Looking at the plot for the *Biased MF* algorithm, it seems there is not a significant correlation between how often an item is rated and how often it is recommended. Nevertheless, its very sparse scatter plot shows the number of recommended items is still low. In this algorithm there are a lot of items that are recommended very rarely (even some of the ones that are rated more frequently) while some few items are being recommended more frequently. *SVD++* also shows a positive correlation between the number of times an item is rated and is recommended. The random algorithm obviously does not show any correlation between the number of times an item is rated and the number of times it is recommended. For each point on the scatter plot, you can also see the average rating for the corresponding item to illustrate the quality of these recommended items using each algorithm.

5.1 Popularity Bias of Recommendations in Different User Groups

In this part, we want to look at how these different algorithms perform in terms of keeping the right ratio of popular and non-popular items recommended to different user groups according to their expected value of such ratio.

Figure 7 shows the ratio of popular items in the profiles of users in different groups versus the ratio of popular items in their recommendations. As you can see, for all algorithms except for random, the ratio of popular items in the recommendations is significantly increased for all user groups confirming how these algorithms impose popularity on users. Specifically, the Niche users are seeing the largest discrepancy of such ratio imposed by the algorithms. However, the recommendations generated by *SVD++* seem to be more consistent with what the users expect to get in terms of the right ratio of popular and non-popular items.

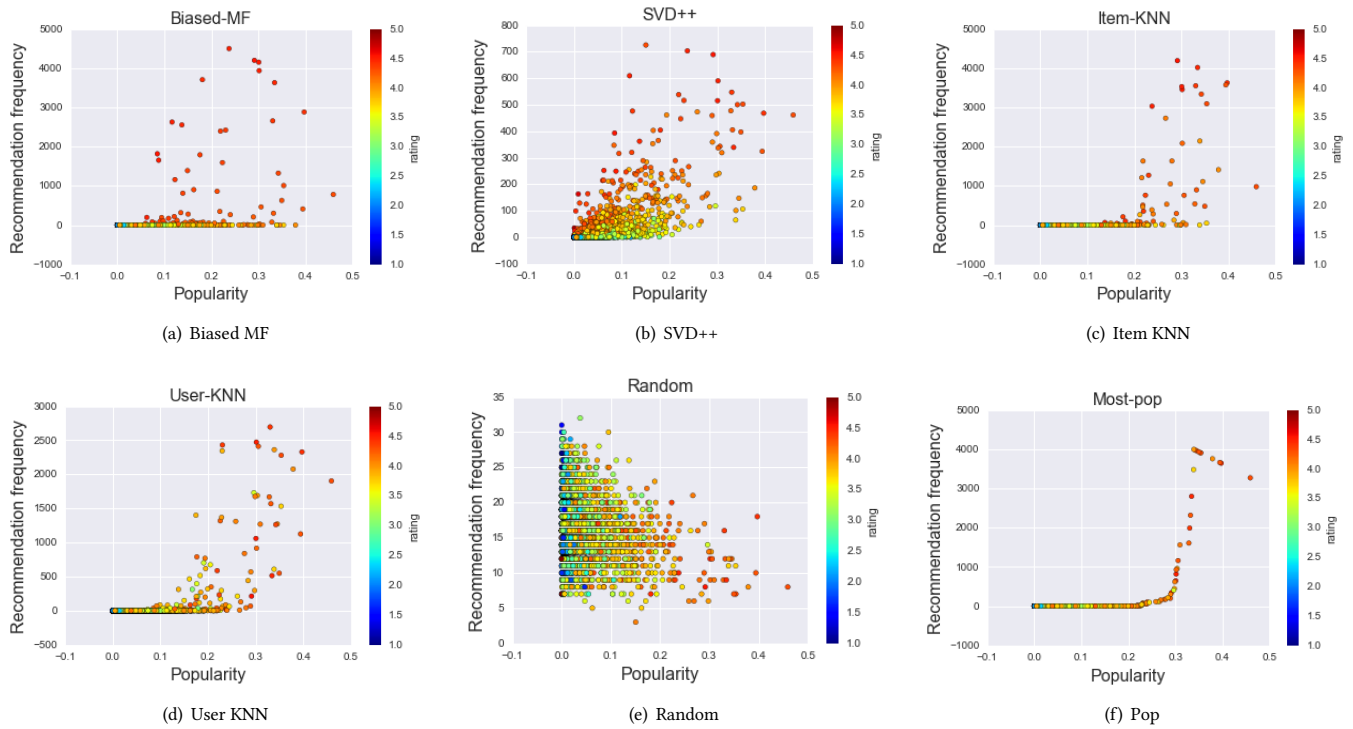


Figure 6: The correlation between the popularity of items and the number of times they are being recommended using different algorithms

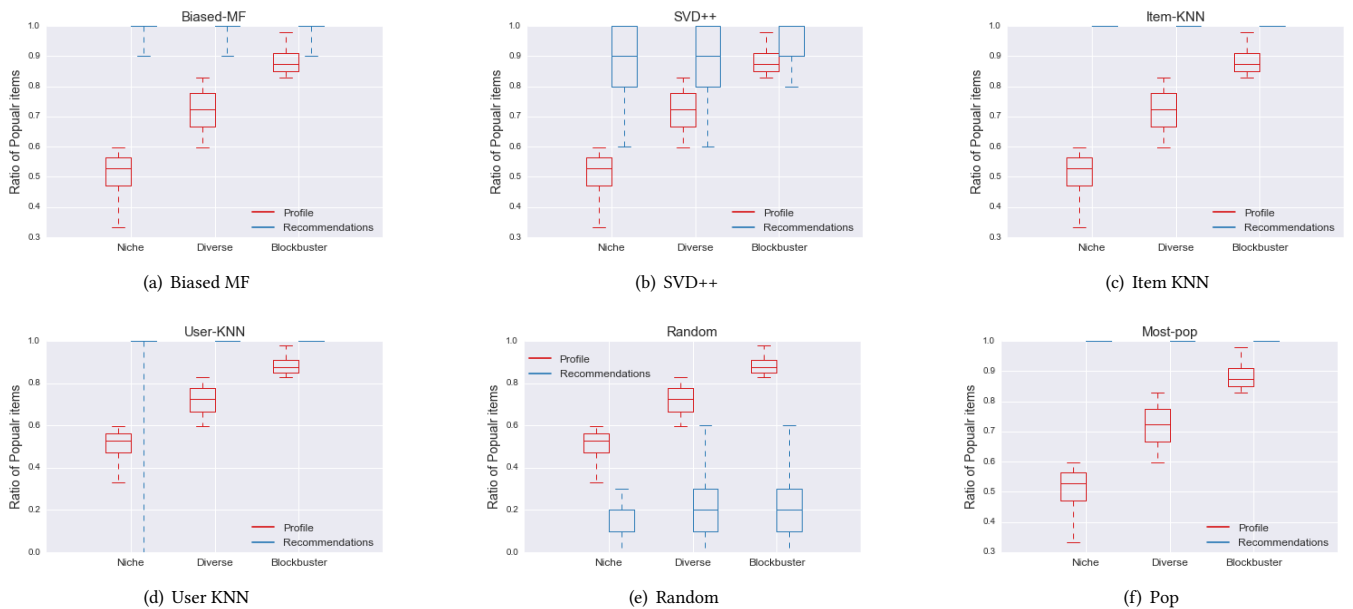


Figure 7: The ratio of popular items in the profiles of users in different groups versus the ratio of popular items in their recommendations

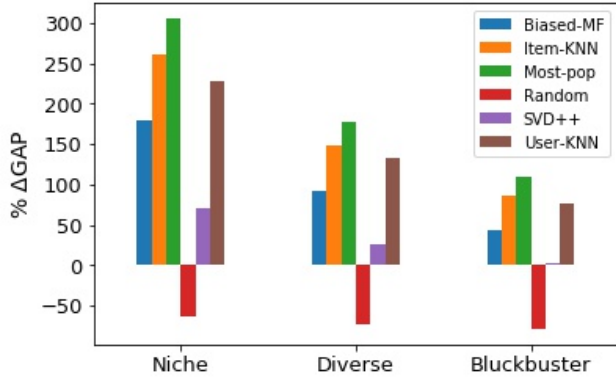


Figure 8: The Δ GAP of different algorithms in user groups

In addition, we define the *Group Average Popularity* ($GAP(g)$) metric that measures the average popularity of items in the profiles of users in a certain group g or their recommendation lists. If we look at the recommendations given to a group of users then GAP measures the average popularity of the recommended items to the users in that group. On the other hand, if we look at the users' profiles, then GAP measures the average popularity of the items rated by the users in that group.

$$GAP(g) = \frac{\sum_{u \in g} \frac{\sum_{i \in p_u} \phi(i)}{|p_u|}}{|g|} \quad (1)$$

were g is the group of users (in our case it is either N , D or B). $\phi(\cdot)$ is the popularity of a certain item (i.e. the number of times it is rated divided by the total number of users) and p_u is the list of items in the profile of user u . We represent the GAP value in user profiles as GAP_p and the GAP for recommendations by GAP_r . For each recommendation algorithm, We measure the change in GAP which is the amount of unwanted popularity in the recommendations imposed by the algorithms to each group. The value of $\Delta GAP = 0$ means a fair representation of users' preferences towards item popularity in the recommendations.

$$\Delta GAP(g) = \frac{GAP(g)_r - GAP(g)_p}{GAP(g)_p} \quad (2)$$

Figure 8 shows the change in the group average popularity (ΔGAP) in different groups using different algorithms. *Most popular* algorithm imposes the highest positive change in GAP followed by item KNN. Random has negative ΔGAP meaning the recommendations on average are less popular compared to the average popularity of users profiles in each group. We can also see that the niche users have the largest ΔGAP for all algorithms.

These results show the unfair nature of popularity bias in the recommendations and how its effect on different users varies based on how interested the users are in popular items.

6 DISCUSSION

In this section we would like to summarize the answers we found to the research questions we listed in section 1.

- **Answer to RQ1:** Our analysis showed that indeed not everyone has the same level of interest in popular items. We found that there are large number of users who are also interested in non-popular items and expecting to get some of those as recommendations. We showed that users with larger profiles are typically less likely to have exclusively rated popular items and they do have some interest in long-tail or non-popular items.
- **Answer to RQ2:** Our results showed that the recommendations generated by all the algorithms were extremely unfair to users with lesser interest in popular items as almost all the recommendations they received were popular items. That showed not every user is affected by the popularity bias the same as other users.

7 CONCLUSION AND FUTURE WORK

Due to the popularity bias in rating data, many recommendation algorithms propagate this bias even further by recommending popular items more frequently while not giving enough exposure to less popular ones. In this paper, we investigated the popularity bias from a different point of view: the user's perspective. We showed that even though there are many users who are interested in getting some non-popular items, they almost do any get any of those in their recommendations. We further defined three groups of users (niche, diverse and Blockbuster-focused) based on the level of their interest towards popular items. We observed that all algorithms recommend items that are much more popular than what the users in those groups have rated. Especially the niche users get really inappropriate recommendations when the level of popularity of the recommendations is compared to what they are interested in. For future work, we will use more datasets for our analysis. We also want to investigate other measures of satisfaction for different user groups such as how relevant their recommendations are, how diverse they are etc.

REFERENCES

- [1] Himan Abdollahpouri. 2019. Popularity Bias in Ranking and Recommendation. In *In AAAI/ACM Conference on AI, Ethics, and Society (AIES'19) January 27–28, 2019, Honolulu, HI, USA*. ACM.
- [2] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. 2019. Beyond Personalization: Research Directions in Multistakeholder Recommendation. *arXiv preprint arXiv:1905.01986* (2019).
- [3] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning to Rank Recommendation. In *Proceedings of the 11th ACM conference on Recommender systems*. ACM, 42–46.
- [4] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-ranking. In *Florida AI Research Symposium (FLAIRS)*. ACM, To appear.
- [5] Gediminas Adomavicius and YoungOk Kwon. 2012. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 896–911. <https://doi.org/10.1109/TKDE.2011.15>
- [6] Chris Anderson. 2006. *The long tail: Why the future of business is selling more for less*. Hyperion.
- [7] Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. 2006. From niches to riches: Anatomy of the long tail. *Sloan Management Review* 47, 4 (2006), 67–71.
- [8] Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. 2006. From niches to riches: Anatomy of the long tail. *Sloan Management Review* (2006), 67–71.
- [9] Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordoñez-Gauger. 2017. Balanced Neighborhoods for Fairness-aware Collaborative Recommendation. In *Workshop on Responsible Recommendation (FATRec)*.
- [10] Oscar Celma and Pedro Cano. 2008. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD*

Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. ACM, 5.

- [11] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2015), 19.
- [12] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* 25, 5 (2015), 427–491.
- [13] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware Classifier with Prejudice Remover Regularizer. In *Proc. of the ECML PKDD 2012, Part II*. 35–50.
- [14] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2014. Correcting Popularity Bias by Enhancing Recommendation Neutrality. In *Poster Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014, Foster City, Silicon Valley, CA, USA, October 6–10, 2014*. http://ceur-ws.org/Vol-1247/recsys14_poster10.pdf
- [15] Weiwen Liu and Robin Burke. 2018. Personalizing Fairness-aware Re-ranking. *arXiv preprint arXiv:1809.02921* (2018). Presented at the 2nd FATRec Workshop held at RecSys 2018, Vancouver, CA.
- [16] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. ACM, 677–686.
- [17] Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 11–18.
- [18] Paul Resnick, R Kelly Garrett, Travis Kriplean, Sean A Munson, and Natalie Jomini Stroud. 2013. Bursting your (filter) bubble: strategies for promoting diverse exposure. In *Proceedings of the 2013 conference on Computer supported cooperative work companion*. ACM, 95–100.
- [19] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 154–162.
- [20] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. *CoRR* abs/1705.08804 (2017). <http://arxiv.org/abs/1705.08804>
- [21] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-Aware Tensor-Based Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1153–1162.