The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study

Dominik Kowald[1(✉)], Markus Schedl[2], and Elisabeth Lex[3]

[1] Know-Center GmbH, Graz, Austria
`dkowald@know-center.at`
[2] Johannes Kepler University Linz, Linz, Austria
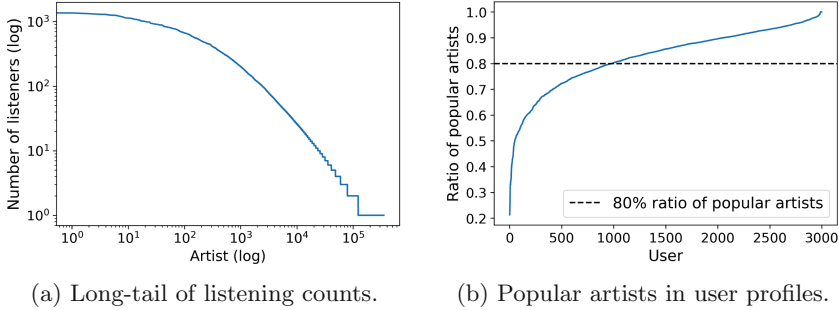`markus.schedl@jku.at`
[3] Graz University of Technology, Graz, Austria
`elisabeth.lex@tugraz.at`

Abstract. Research has shown that recommender systems are typically biased towards popular items, which leads to less popular items being underrepresented in recommendations. The recent work of Abdollahpouri et al. in the context of movie recommendations has shown that this popularity bias leads to unfair treatment of both long-tail items as well as users with little interest in popular items. In this paper, we reproduce the analyses of Abdollahpouri et al. in the context of music recommendation. Specifically, we investigate three user groups from the Last.fm music platform that are categorized based on how much their listening preferences deviate from the most popular music among all Last.fm users in the dataset: (i) low-mainstream users, (ii) medium-mainstream users, and (iii) high-mainstream users. In line with Abdollahpouri et al., we find that state-of-the-art recommendation algorithms favor popular items also in the music domain. However, their proposed Group Average Popularity metric yields different results for Last.fm than for the movie domain, presumably due to the larger number of available items (i.e., music artists) in the Last.fm dataset we use. Finally, we compare the accuracy results of the recommendation algorithms for the three user groups and find that the low-mainstreaminess group significantly receives the worst recommendations.

Keywords: Algorithmic fairness · Recommender systems · Popularity bias · Item popularity · Music recommendation · Reproducibility

1 Introduction

Recommender systems are quintessential tools to support users in finding relevant information in large information spaces [10]. However, one limitation of typical recommender systems is the so-called popularity bias, which leads to the underrepresentation of less popular (i.e., long-tail) items in the recommendation

(a) Long-tail of listening counts.   (b) Popular artists in user profiles.

Fig. 1. Listening distribution of music artists. We find that around 1/3 (i.e., 1,000) of our users actually listen to at least 20% of unpopular artists.

lists [1,4,5]. The recent work of Abdollahpouri et al. [2] has investigated this popularity bias from the user perspective in the movie domain. The authors have shown that state-of-the-art recommendation algorithms tend to underserve users, who like unpopular items.
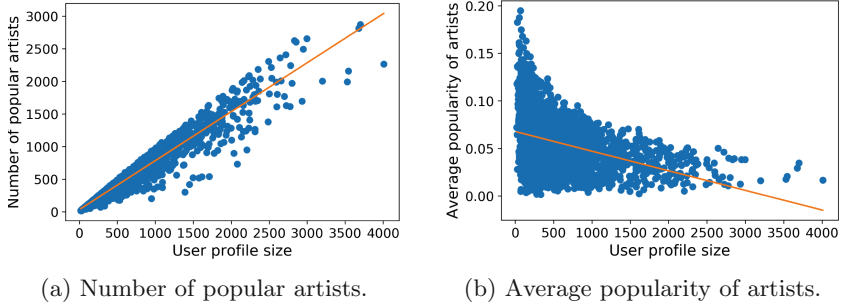
In this paper, we reproduce this study and conduct it in the music domain. As described in [16], there are several aspects of music recommendations that make them different to, e.g., movie recommendations such as the vast amount of available items. Therefore, we investigate music recommendations concerning popularity bias and, for reasons of comparability, raise the same two research questions as in [2]:

– *RQ1*: To what extent are users or groups of users interested in popular music artists?
– *RQ2*: To what extent does the popularity bias of recommendation algorithms affect users with different inclination to mainstream music?

For our experiments, we use a publicly available Last.fm dataset and address *RQ1* in Sect. 2 by analyzing the popularity of music artists in the user profiles. Next, we address *RQ2* in Sect. 3 by comparing six state-of-the-art music recommendation algorithms concerning their popularity bias propagation.

2   Popularity Bias in Music Data

For our reproducibility study, we use the freely available LFM-1b dataset [14]. Since this dataset contains 1.1 billion listening events of more than 120,000 Last.fm users and thus is much larger than the MovieLens dataset used in [2], we focus on a subset of it. Precisely, we extract 3,000 users that reflect the three user groups investigated in [2]. To this end, we use the mainstreaminess score, which is available for the users in the LFM-1b dataset and which is defined as the overlap between a user's listening history and the aggregated listening history of all Last.fm users in the dataset [3]. It thus represents a proxy for a user's inclination to popular music.
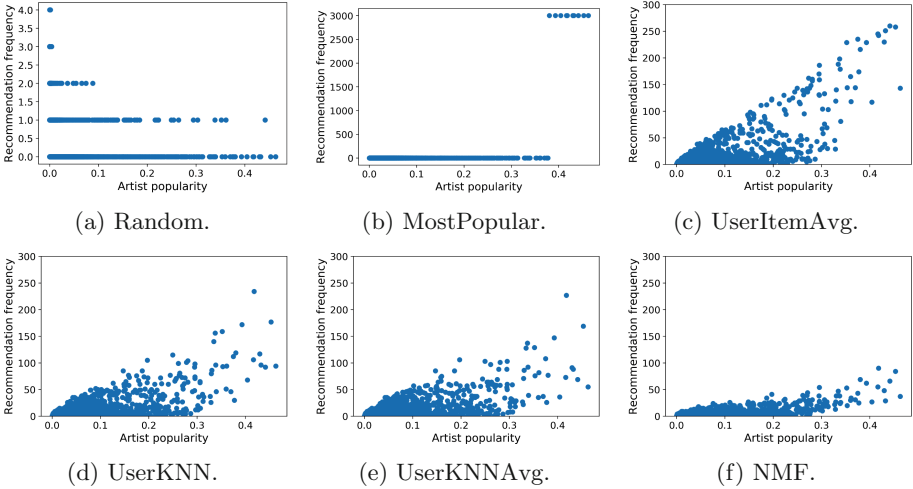
(a) Number of popular artists.   (b) Average popularity of artists.

Fig. 2. Correlation of user profile size and the popularity of artists in the user profile. While there is a positive correlation between profile size and number of popular artists, there is a negative correlation between profile size and the average artist popularity.

Our subset consists of the 1,000 users with lowest mainstreaminess scores (i.e., the LowMS group), the 1,000 users with a mainstreaminess score around the median (i.e., the MedMS group), and the 1,000 users with the highest mainstreaminess scores (i.e., the HighMS group). In total, we investigate 1,755,361 user-artist interactions between 3,000 users and 352,805 music artists. Compared to the MovieLens dataset with only 3,900 movies that Abdollahpouri et al. [2] have used in their study, our itemset is, consequently, much larger.

Listening Distribution of Music Artists. Fig. 1 depicts the listening distribution of music artists in our Last.fm dataset. As expected, in Fig. 1a, we observe a long-tail distribution of the listener counts of our items (i.e., artists). That is, only a few artists are listened to by many users, while most artists (i.e., the long-tail) are only listened to by a few users. Furthermore, in Fig. 1b, we plot the ratio of popular artists in the profiles of our 3,000 Last.fm users. As in [2], we define an artist as popular if the artist falls within the top 20% of artists with the highest number of listeners. We see that around 1,000 of our 3,000 users (i.e., around 1/3) have at least 20% of unpopular artists in their user profiles. This number also corresponds to the number of low-mainstream users we have in the LowMS user group.

User Profile Size and Popularity Bias in Music Data. Next, in Fig. 2, we investigate if there is a correlation between the user profile size (i.e., number of distinct items/artists) and the popularity of artists in the user profile. Therefore, in Fig. 2a, we plot the number of popular artists in the user profile over the profile size As expected, we find a positive correlation ($R = .965$) since the likelihood of having popular artists in the profile increases with the number of items in the profile. However, when plotting the average popularity of artists in the user profile over the profile size in Fig. 2b, we find a negative correlation ($R = -.372$), which means that users with a smaller profile size tend to listen to more popular artists. As in [2], we define the popularity of an artist as the ratio of users who have listened to this artist.

Fig. 3. Correlation of artist popularity and recommendation frequency. For all six algorithms, the recommendation frequency increases with the artist popularity.
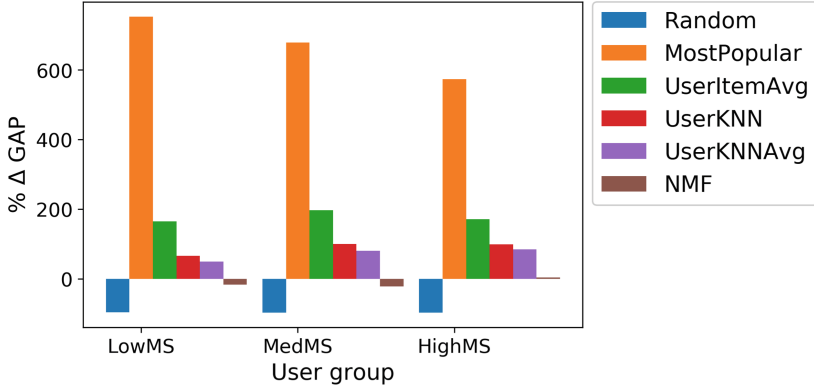
Concerning *RQ1*, we find that one-third of our Last.fm users have at least 20% of unpopular artists in their profiles and thus, are also interested in low-mainstream music. Furthermore, we find that users with a small profile size tend to have more popular artists in their profiles than users with a more extensive profile size. These findings are in line with what Abdollahpouri et al. have found [2].

3 Popularity Bias in Music Recommendation

In this section, we study popularity bias in state-of-the-art music recommendation algorithms. To foster the reproducibility of our study, we calculate and evaluate all recommendations with the Python-based open-source recommendation toolkit Surprise[1]. Using Surprise, we formulate our music recommendations as a rating prediction problem, where we predict the preference of a target user u for a target artist a. We define the preference of a for u by scaling the listening count of a by u to a range of [0, 1000] as also done in [15]. We then recommend the top-10 artists with the highest predicted preferences.

Recommendation of Popular Music Artists. We use the same evaluation protocol (i.e., 80/20 train/test split) and types of algorithms as in [2], which includes (i) baseline approaches, (ii) KNN-based approaches, and (iii) Matrix Factorization-based approaches. Specifically, we evaluate three baselines, i.e., Random, MostPopular, and UserItemAvg, which predicts the average listening count in the dataset by also accounting for deviations of u and a (e.g., if u tends

---

[1] http://surpriselib.com/.

Fig. 4. Group Average Popularity (Δ GAP) of recommendation algorithms for LowMS, MedMS and HighMS. Except for the Random and NMF algorithms, all approaches provide too popular artist recommendations for all three user groups.

to have in general more listening events than the average Last.fm user) [6]. We also evaluate the two KNN-based approaches [13] UserKNN and UserKNNAvg, which is a hybrid combination of UserKNN and UserItemAvg. Finally, we include NMF (Non-Negative Matrix Factorization) into our study [9]. To reduce the computational effort of our study, in our evaluation, we exclude ItemKNN [12] as well as SVD++ [11] in contrast to [2]. In Fig. 3, we plot the correlation of artist popularity and how often the six algorithms recommend these artists. For all algorithms except for Random, we find a positive correlation, which means that popular items are recommended more often than unpopular items. As expected, this effect is most evident for the MostPopular algorithm and not present at all for the Random algorithm. It also seems that this popularity bias is not as strong in the case of NMF, which we will investigate further in the next section of this paper.

Popularity Bias for Different User Groups. To investigate the popularity bias of music recommendations for different user groups (i.e., LowMS, MedMS, and HighMS), we use the Group Average Popularity (GAP) metric proposed in [2]. Here, $GAP(g)_p$ measures the average popularity of the artists in the user profiles p of a specific user group g. We also define $GAP(g)_r$, which measures the average popularity of the artists recommended by a recommendation algorithm r to the users of group g. For each algorithm and user group, we are interested in the change in GAP (i.e., ΔGAP), which shows how the popularity of the recommended artists differs from the expected popularity of the artists in the user profiles. Hence, $\Delta GAP = 0$ would indicate fair recommendations in terms of item popularity, where fair means that the average artist popularity of the recommendations a user receives matches the average artist popularity in the user's profile. It is given by: $\Delta GAP = \frac{GAP(g)_r - GAP(g)_p}{GAP(g)_p}$.

Table 1. MAE results (the lower, the better) for four personalized recommendation algorithms and our three user groups. The worst (i.e., highest) results are always given for the LowMS user group (statistically significant according to a t-test with $p < .005$ as indicated by ***). Across the algorithms, the best (i.e., lowest) results are provided by NMF (indicated by bold numbers).

User group	UserItemAvg	UserKNN	UserKNNAvg	NMF
LowMS	42.991***	49.813***	46.631***	**38.515*****
MedMS	33.934	42.527	37.623	**30.555**
HighMS	40.727	46.036	43.284	**37.305**
All	38.599	45.678	41.927	**34.895**

In Fig. 4, we plot the ΔGAP for our six algorithms and three user groups. In contrast to the results presented in [2], where the LowMS group (i.e., the niche users) receives the highest values, we do not observe a clear difference between the groups except for MostPopular. We think that this is the case because of the large number of items we have in our Last.fm dataset (i.e., 352,805 artists compared to 3,900 movies in MovieLens). However, in line with Fig. 3, we again find that Random and NMF provide the fairest recommendations.

To further investigate $RQ2$, we analyze the Mean Average Error (MAE) [17] results of the four personalized algorithms for our user groups. As shown in Table 1, the LowMS group receives significantly worse (according to a t-test) recommendations than MedMS and HighMS for all algorithms. Interestingly, the MedMS group gets the best recommendations, probably since the users in this group have the largest profiles (i.e., on average 715 artists per user as compared to around 500 for the other two groups). Across the algorithms, NMF provides the best results. This is especially of interest since NMF also provided the fairest results in terms of artist popularity across the personalized algorithms.

4   Conclusion and Future Work

In this paper, we reproduced the study of [2] on the unfairness of popularity bias in movie recommender systems, which we adopted to the music domain. Similar to the original paper, we find (i) that users only have a limited interest in popular items ($RQ1$) and (ii) that users interested in unpopular items (i.e., LowMS) receive worse recommendations than users interested in popular items (i.e., HighMS). However, we also find that the proposed GAP metric does not provide the same results for Last.fm as it does for MovieLens, probably due to the high number of available items.

For future work, we plan to adapt this GAP metric in order to make it more robust for various domains. Furthermore, we want to study the characteristics of the LowMS users in order to better understand why they receive the worst recommendations and to potentially overcome this with novel algorithms (e.g., [7]).

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 42–46. ACM (2017)
2. Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B.: The unfairness of popularity bias in recommendation. In: Workshop on Recommendation in Multi-stakeholder Environments (RMSE 2019), in conjunction with the 13th ACM Conference on Recommender Systems, RecSys 2019 (2019)
3. Bauer, C., Schedl, M.: Global and country-specific mainstreaminess measures: definitions, analysis, and usage for improving personalized music recommendation systems. PloS one **14**(6), e0217389 (2019)
4. Brynjolfsson, E., Hu, Y.J., Smith, M.D.: From niches to riches: anatomy of the long tail. Sloan Manag. Rev. **47**(4), 67–71 (2006)
5. Jannach, D., Lerche, L., Kamehkhosh, I., Jugovac, M.: What recommenders recommend: an analysis of recommendation biases and possible countermeasures. User Model. User-Adap. Interact. **25**(5), 427–491 (2015). https://doi.org/10.1007/s11257-015-9165-3
6. Koren, Y.: Factor in the neighbors: scalable and accurate collaborative filtering. ACM Trans. Knowl. Discov. Data (TKDD) **4**(1), 1 (2010)
7. Kowald, D., Lex, E., Schedl, M.: Modeling artist preferences for personalized music recommendations. In: Proceedings of the Late-Breaking-Results Track of the 20th Annual Conference of the International Society for Music Information Retrieval (ISMIR 2019) (2019)
8. Kowald, D., Schedl, M., Lex, E.: LFM user groups (2019). https://doi.org/10.5281/zenodo.3475975
9. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans. Ind. Inform. **10**(2), 1273–1284 (2014)
10. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Boston (2011). https://doi.org/10.1007/978-0-387-85820-3_1
11. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Proceedings of the Fifth International Conference on Computer and Information Science, vol. 27, p. 28 (2002)
12. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J., et al.: Item-based collaborative filtering recommendation algorithms. In: WWW 1, 285–295 (2001)

---

13. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9

14. Schedl, M.: The LFM-1B dataset for music retrieval and recommendation. In: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ICMR 2016), pp. 103–110. ACM, New York (2016)

15. Schedl, M., Bauer, C.: Distance-and rank-based music mainstreaminess measurement. In: Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization, pp. 364–367. ACM (2017)

16. Schedl, M., Zamani, H., Chen, C.-W., Deldjoo, Y., Elahi, M.: Current challenges and visions in music recommender systems research. Int. J. Multimedia Inf. Retr. **7**(2), 95–116 (2018). https://doi.org/10.1007/s13735-018-0154-2

17. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim. Res. **30**(1), 79–82 (2005)