



Popularity Bias in Collaborative Filtering-Based Multimedia Recommender Systems

Dominik Kowald^{1,2}(✉) and Emanuel Lacic¹

¹ Know-Center GmbH, Graz, Austria
{dkowald, elacic}@know-center.at

² Graz University of Technology, Graz, Austria

Abstract. Multimedia recommender systems suggest media items, e.g., songs, (digital) books and movies, to users by utilizing concepts of traditional recommender systems such as collaborative filtering. In this paper, we investigate a potential issue of such collaborative-filtering based multimedia recommender systems, namely popularity bias that leads to the underrepresentation of unpopular items in the recommendation lists. Therefore, we study four multimedia datasets, i.e., Last.fm, MovieLens, BookCrossing and MyAnimeList, that we each split into three user groups differing in their inclination to popularity, i.e., LowPop, MedPop and HighPop. Using these user groups, we evaluate four collaborative filtering-based algorithms with respect to popularity bias on the item and the user level. Our findings are three-fold: firstly, we show that users with little interest into popular items tend to have large user profiles and thus, are important data sources for multimedia recommender systems. Secondly, we find that popular items are recommended more frequently than unpopular ones. Thirdly, we find that users with little interest into popular items receive significantly worse recommendations than users with medium or high interest into popularity.

Keywords: multimedia recommender systems · collaborative filtering · popularity bias · algorithmic fairness

1 Introduction

Collaborative filtering (CF) is one of the most traditional but also most powerful concepts for calculating personalized recommendations [22] and is vastly used in the field of multimedia recommender systems (MMRS) [11]. However, one issue of CF-based approaches is that they are prone to popularity bias, which leads to the overrepresentation of popular items in the recommendation lists [2, 3]. Recent research has studied popularity bias in domains such as music [15, 16] or movies [3] by comparing the recommendation performance for different user groups that differ in their inclination to mainstream multimedia items. However, a comprehensive study of investigating popularity bias on the item and user level across several multimedia domains is still missing (see Sect. 2).

In the present paper, we therefore build upon these previous works and expand the study of popularity bias to four different domains of MMRS: music (Last.fm), movies (MovieLens), digital books (BookCrossing), and animes (MyAnimeList). Within these domains, we show that users with little interest into popular items tend to have large user profiles and thus, are important consumers and data sources for MMRS. Furthermore, we apply four different CF-based recommendation algorithms (see Sect. 3) on our four datasets that we each split into three user groups that differ in their inclination to popularity (i.e., LowPop, MedPop, and HighPop). With this, we address two research questions (RQ):

- **RQ1:** To what extent does an item’s popularity affect this item’s recommendation frequency in MMRS?
- **RQ2:** To what extent does a user’s inclination to popular items affect the quality of MMRS?

Regarding **RQ1**, we find that the probability of a multimedia item to be recommended strongly correlates with this items’ popularity. Regarding **RQ2**, we find that users with less inclination to popularity (LowPop) receive statistically significantly worse multimedia recommendations than users with medium (MedPop) and high (HighPop) inclination to popular items (see Sect. 4). Our results demonstrate that although users with little interest into popular items tend to have the largest user profiles, they receive the lowest recommendation accuracy. Hence, future research is needed to mitigate popularity bias in MMRS, both on the item and the user level.

2 Related Work

This section presents research on popularity bias that is related to our work. We split these research outcomes in two groups: (i) work related to recommender systems in general, and (ii) work that focuses on popularity bias mitigation techniques.

Popularity Bias in Recommender Systems. Within the domain of recommender systems, there is an increasing number of works that study the effect of popularity bias. For example, as reported in [8], bias towards popular items can affect the consumption of items that are not popular. This in turn prevents them to become popular in the future at all. That way, a recommender system is prone to ignoring novel items or the items liked by niche users that are typically hidden in the “long-tail” of the available item catalog. Tackling these long-tail items has been recognized by some earlier work, such as [10, 20]. This issue is further investigated by [1, 2] using the popular movie dataset MovieLens 1M. The authors show that more than 80% of all ratings actually belong to popular items, and based on this, focus on improving the trade-off between the ranking accuracy and coverage of long-tail items. Research conducted in [13] illustrates a comprehensive algorithmic comparison with respect to popularity bias. The authors analyze multimedia datasets such as MovieLens, Netflix, Yahoo!Movies and BookCrossing, and find that recommendation methods only consider a small fraction of

the available item spectrum. For instance, they find that KNN-based techniques focus mostly on high-rated items and factorization models lean towards recommending popular items. In our work, we analyze an even larger set of multimedia domains and study popularity bias not only on the item but also on the user level.

Popularity Bias Mitigation Techniques. Typical research on mitigating popularity bias performs a re-ranking step on a larger set of recommended candidate items. The goal of such post-processing approaches is to better expose long-tail items in the recommendation list [2, 4, 6]. Here, for example, [7] proposes to improve the total number of distinct recommended items by defining a target distribution of item exposure and minimizing the discrepancy between exposure and recommendation frequency of each item. In order to find a fair ratio between popular and less popular items, [24] proposes to create a protected group of long-tail items and to ensure that their exposure remains statistically indistinguishable from a given minimum. Beside focusing on post-processing, there are some in-processing attempts in adapting existing recommendation algorithms in a way that the generated recommendations are less biased toward popular items. For example, [5] proposes to use a probabilistic neighborhood selection for KNN methods, or [23] suggests a blind-spot-aware matrix factorization approach that debiases interactions between the recommender system and the user. We believe that the findings of our paper can inform future research on choosing the right mitigation technique for a given setting.

3 Method

In this section, we describe (i) our definition of popularity, (ii) our four multimedia datasets, and (iii) our four recommendation algorithms based on collaborative filtering as well as our evaluation protocol.

3.1 Defining Popularity

Here, we describe how we define popularity (i) on the item level, and (ii) on the user level. We use the item popularity definition of [3], where the item popularity score Pop_i of an item i is given by the relative number of users who have rated i , i.e., $Pop_i = \frac{|U_i|}{|U|}$. Based on this, we can also define $Pop_{i,u}$ as the average item popularity in the user profile I_u , i.e., $Pop_{i,u} = \frac{1}{|I_u|} \sum_{i \in I_u} Pop_i$. Additionally, we can also define an item i as popular if it falls within the top-20% of item popularity scores. Thus, we define $I_{u,Pop}$ as the set of popular items in the user profile.

On the user level, we also follow the work of [3] and define a user u 's inclination to popularity Pop_u as the ratio of popular items in the user profile, i.e., $Pop_u = \frac{|I_{u,Pop}|}{|I_u|}$. As an example, $Pop_u = 0.8$ if 80% of the items in the user's item history are popular ones. We use this definition to create the LowPop, MedPop and HighPop user groups in case of MovieLens, BookCrossing and MyAnimeList.

Table 1. Statistics of our four datasets, where $|U|$ is the number of users, $|I|$ is the number of media items, $|R|$ is the number of ratings, sparsity is defined as the ratio of observed ratings $|R|$ to possible ratings $|U| \times |I|$, and R -range is the rating range.

Dataset	$ U $	$ I $	$ R $	$ R / U $	$ R / I $	Sparsity	R -range
Last.fm	3,000	352,805	1,755,361	585	5	0.998	[1–1,000]
MovieLens	3,000	3,667	675,610	225	184	0.938	[1–5]
BookCrossing	3,000	223,607	577,414	192	3	0.999	[1–10]
MyAnimeList	3,000	9,450	649,814	216	69	0.977	[1–10]

In case of Last.fm, we use a definition for Pop_u especially proposed for the music domain, which is termed the mainstreamness score [9]. Here, we use the $M_{R,APC}^{global}$ definition, which is already provided in the dataset¹ published in our previous work [16]. Formally, $M_{R,APC}^{global}(u) = \tau(\text{ranks}(APC), \text{ranks}(APC(u)))$, where APC and $APC(u)$ are the artist play counts averaged over all users and for a given user u , respectively. τ indicates the rank-order correlation according to Kendall’s τ . Thus, u ’s mainstreamness score is defined as the overlap between a user’s item history and the aggregated item history of all Last.fm users in the dataset. Thus, the higher the mainstreamness score, the higher a user’s inclination to popular music. Please note that we cannot calculate the mainstreamness score for the other datasets, since we do not have multiple interactions per item (i.e., play counts) in these cases (only one rating per user-item pair).

To get a better feeling of the relationship between average item popularity scores in the user profiles (i.e., $Pop_{u,i}$) and the user profile size (i.e., $|I_u|$), we plot these correlations for our four datasets and per user group in Fig. 1. Across all datasets, we see a negative correlation between average item popularity and user profile size, which means that users with little interest in popular items tend to have large user profiles. This suggests that these users are important consumers and data sources in MMRS, and thus, should also be treated in a fair way (i.e., should receive similar accuracy scores as users with medium or high interest in popular items).

3.2 Multimedia Datasets

For our study, we use four datasets containing rating data of users for media items. The statistics of our datasets can be found in Table 1, and we provide the datasets via Zenodo². The users in each of our four datasets are split into three equally-sized user groups: (i) LowPop, i.e., the 1,000 users with the least inclination to popular items, (ii) MedPop, i.e., 1,000 users with medium inclination to popular media items, and (iii) HighPop, i.e., the 1,000 users with the highest inclination to popular media items. This sums up to $|U| = 3,000$ users per

¹ <https://zenodo.org/record/3475975>.

² <https://zenodo.org/record/6123879>.

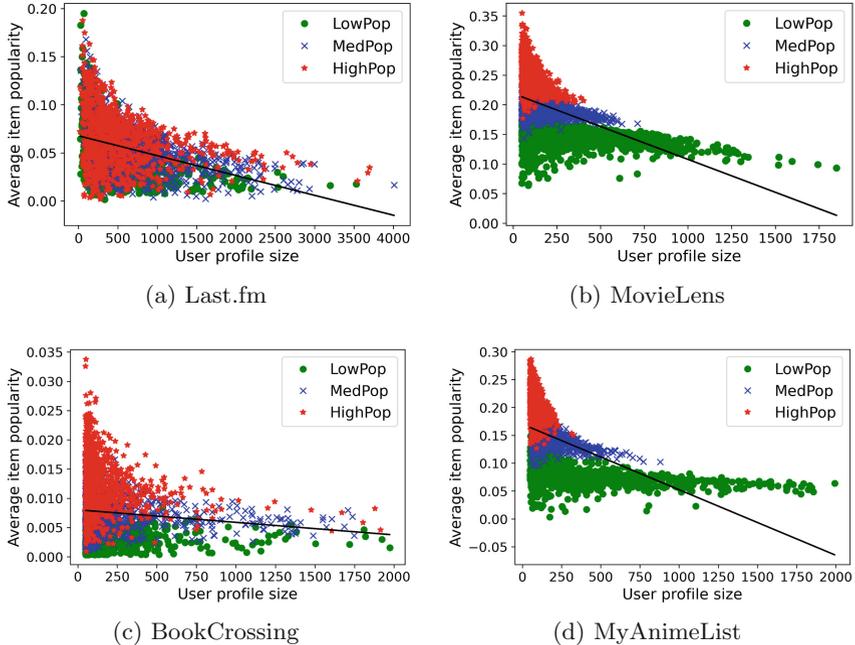


Fig. 1. Relationship between average item popularity scores in the user profiles (i.e., $Pop_{u,i}$) and user profile size (i.e., $|I_u|$). We see that users with little interest in popular items tend to have large user profiles.

dataset. Next, we describe our four datasets and how we split the user groups based on the popularity definitions given before:

Last.fm. For the music streaming platform Last.fm, we use the dataset published in our previous work [16], which is based on the LFM-1b dataset³. Here, a user is assigned to one of the three groups LowPop, MedPop and HighPop based on the user’s mainstreamness score [9], which we defined earlier (i.e., $M_{R,APC}^{global}$). Additionally, in this Last.fm dataset, the listening counts of users for music artists are scaled to a rating range of [1–1,000]. When looking at Table 1, Last.fm has the largest number of items $|I| = 352,805$ and the largest number of ratings $|R| = 1,755,361$ across our four datasets.

MovieLens. In case of the movie rating portal MovieLens, we use the well-known MovieLens-1M dataset⁴. We extract all users with a minimum of 50 ratings and a maximum of 2,000 ratings. We assign these users to one of the three user groups LowPop, MedPop and HighPop based on the ratio of popular items in the user profiles [3] as described earlier (i.e., Pop_u). Table 1 shows that MovieLens is the least sparse (i.e., most dense) dataset in our study and also has the highest number of ratings per items ($|R|/|I|$).

³ <http://www.cp.jku.at/datasets/LFM-1b/>.

⁴ <https://grouplens.org/datasets/movielens/1m/>.

BookCrossing. The dataset of the (digital) book sharing platform BookCrossing was provided by Uni Freiburg⁵. We use the same popularity definitions, group assignment method as well as rating thresholds as in case of MovieLens. However, in contrast to MovieLens, BookCrossing contains not only explicit feedback in the form of ratings but also implicit feedback when a user bookmarks a book. In this case, we set the implicit feedback to a rating of 5, which is the middle value in BookCrossing’s rating range of [1–10]. Across all datasets, BookCrossing is the dataset with the highest sparsity.

MyAnimeList. We apply the same processing methods as used in case of BookCrossing to the MyAnimeList dataset, which is provided via Kaggle⁶. Similar to BookCrossing, MyAnimeList also contains implicit feedback when a user bookmarks an Anime, and again we convert this feedback to an explicit rating of 5, which is the middle value in the rating range.

3.3 Recommendation Algorithms and Evaluation Protocol

We use the same set of personalized recommendation algorithms as used in our previous work [16] but since we focus on CF-based methods, we replace the UserItemAvg algorithm with a scalable co-clustering-based approach [12] provided by the Python-based Surprise framework⁷. Thus, we evaluate two KNN-based algorithms without and with incorporating the average rating of the target user and item (UserKNN and UserKNNAvg), one non-negative matrix factorization variant [19] (NMF) as well as the aforementioned CoClustering algorithm. In most cases, we stick to the default parameter settings as suggested by the Surprise framework and provide the detailed settings in our GitHub repository⁸.

We also follow the same evaluation protocol as used in our previous work [16] and formulate the recommendation task as a rating prediction problem, which we measure using the mean absolute error (MAE). However, instead of using only one 80/20 train-set split, we use a more sophisticated 5-fold cross-validation evaluation protocol. To test for statistical significance, we perform pairwise t-tests between LowPop and MedPop as well as between LowPop and HighPop since we are interested if LowPop is treated in an unfair way by the MMRS. We report statistical significance for LowPop only in cases in which there is a significant difference between LowPop and MedPop as well as between LowPop and HighPop for all five folds.

4 Results

We structure our results based on our two research questions. Thus, we first investigate popularity bias on the item level by investigating the relationship

⁵ <http://www2.informatik.uni-freiburg.de/~chiegler/BX/>.

⁶ <https://www.kaggle.com/CooperUnion/anime-recommendations-database>.

⁷ <http://surpriselib.com/>.

⁸ <https://github.com/domkowald/FairRecSys>.

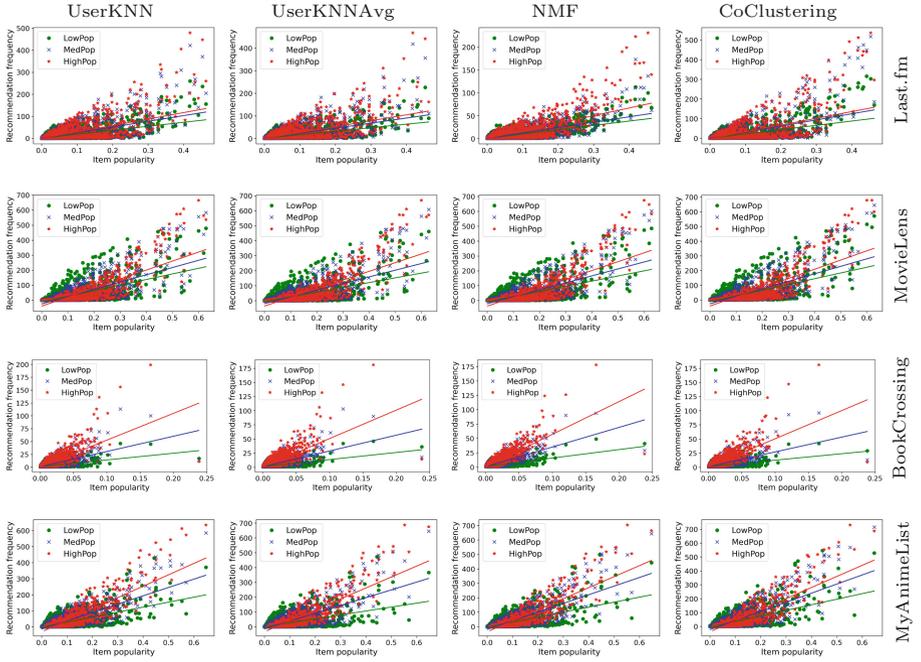


Fig. 2. RQ1: Relationship between item popularity and recommendation frequency of four CF-based algorithms for Last.fm, MovieLens, BookCrossing and MyAnimeList. In all 16 cases, we see that popular media items have a higher probability of being recommended than unpopular ones.

between item popularity and recommendation frequency (**RQ1**). Next, we investigate popularity bias on the user level by comparing the recommendation performance for our three user groups (**RQ2**).

4.1 RQ1: Relationship Between Item Popularity and Recommendation Frequency

Figure 2 shows the relationship between item popularity and recommendation frequency for the four CF-based algorithms UserKNN, UserKNNAvg, NMF and CoClustering on all five folds of our four multimedia datasets Last.fm, MovieLens, BookCrossing and MyAnimeList. The solid lines indicate the linear regression between the two variables for the three user groups.

In all 16 plots, and all three user groups, we observe a positive relationship between an item’s popularity and how often this item gets recommended (**RQ1**). However, for NMF applied to Last.fm, the maximum recommendation frequency is much lower as in case of the other algorithms. Thus, only in case of NMF applied to Last.fm, we see a weak relationship between popularity and recommendation frequency, while in all other cases, we see a strong relationship

Table 2. RQ2: Mean absolute error (MAE) results (the lower, the better) of our study. The lowest accuracy is always given for the LowPop user group (statistically significant according to a t-test with $p < 0.001$ as indicated by ******* and $p < 0.05$ as indicated by ******). Across the algorithms, the best results are indicated by **bold numbers** and across the user groups, the best results are indicated by *italic numbers*.

Dataset	User group	UserKNN	UserKNNAvg	NMF	CoClustering
Last.fm	LowPop	49.489 ^{***}	46.483 ^{***}	39.641^{**}	47.304 ^{***}
	MedPop	<i>42.899</i>	<i>37.940</i>	32.405	<i>37.918</i>
	HighPop	45.805	43.070	38.580	42.982
MovieLens	LowPop	0.801 ^{***}	0.763 ^{***}	0.753 ^{***}	0.738^{***}
	MedPop	0.748	0.727	0.722	0.705
	HighPop	<i>0.716</i>	<i>0.697</i>	<i>0.701</i>	0.683
BookCrossing	LowPop	1.403 ^{***}	1.372^{***}	1.424 ^{***}	1.392 ^{***}
	MedPop	<i>1.154</i>	1.122	<i>1.214</i>	<i>1.134</i>
	HighPop	1.206	1.155	1.274	1.162
MyAnimeList	LowPop	1.373 ^{***}	1.001^{***}	1.010 ^{***}	1.001 ^{***}
	MedPop	1.341	0.952	0.968	<i>0.956</i>
	HighPop	<i>1.311</i>	0.948	<i>0.951</i>	0.975

between these variables. This is in line with our previous related work investigating popularity bias in Last.fm [16]. When comparing the three user groups, we see the weakest relationship between the variables for LowPop and the strongest relationship for HighPop. We will refer to this finding when investigating **RQ2**.

4.2 RQ2: Relationship Between Users' Inclination to Popular Items and Recommendation Accuracy

Table 2 shows the MAE estimates for the aforementioned CF-based recommendation algorithms (UserKNN, UserKNNAvg, NMF, and CoClustering) on the four multimedia datasets (Last.fm, MovieLens, BookCrossing, and MyAnimeList) split in three user groups that differ in their inclination to popularity (LowPop, MedPop, and HighPop). Additionally, we indicate statistically significant differences between both LowPop and MedPop, and LowPop and HighPop according to a t-test with $p < 0.001$ using ******* and with $p < 0.05$ using ****** in the LowPop lines.

Across all datasets, we observe the highest MAE estimates, and thus lowest recommendation accuracy, for the LowPop user groups. The best results, indicated by *italic numbers*, are reached for the MedPop group in case of Last.fm and BookCrossing, and for the HighPop group in case of MovieLens and MyAnimeList. For Last.fm this is in line with our previous work [16]. Across the algorithms, we see varying results: for Last.fm, and again in line with our previous work [16], the best results are reached for NMF. For MovieLens, we get the best results for

the CoClustering approach, and for BookCrossing and MyAnimeList the highest accuracy is reached for the UserKNN variant UserKNNAvg. We plan to investigate these differences across the user groups and the algorithms in our future research, as outlined in the next section.

Taken together, users with little inclination to popular multimedia items receive statistically significantly worse recommendations by CF-based algorithms than users with medium and high inclination to popularity (**RQ2**). When referring back to our results of **RQ1** in Fig. 2, this is interesting since LowPop is the group with the weakest relationship between item popularity and recommendation frequency. However, this suggests that recommendations are still too popular for this user group and an adequate mitigation strategy is needed.

5 Conclusion

In this paper, we have studied popularity bias in CF-based MMRS. Therefore, we investigated four recommendation algorithms (UserKNN, UserKNNAvg, NMF, and CoClustering) for three user groups (LowPop, MedPop, and HighPop) on four multimedia datasets (Last.fm, MovieLens, BookCrossing, and MyAnimeList). Specifically, we investigated popularity bias from the item (**RQ1**) and user (**RQ2**) perspective. Additionally, we have shown that users with little interest into popular items tend to have large profile sizes, and therefore are important data sources for MMRS.

With respect to **RQ1**, we find that the popularity of a multimedia item strongly correlates with the probability that this item is recommended by CF-based approaches. With respect to **RQ2**, we find that users with little interest in popular multimedia items (i.e., LowPop) receive significantly worse recommendations than users with medium (i.e., MedPop) or high (i.e., HighPop) interest in popular items. This is especially problematic since users with little interest into popularity tend to have large profile sizes, and thus, should be treated in a fair way by MMRS.

Future Work. Our results demonstrate that future work should further focus on studying this underserved user group in order to mitigate popularity bias in CF-based recommendation algorithms. We believe that our findings are a first step to inform the research on popularity bias mitigation techniques (see Sect. 2) to choose the right mitigation strategy for a given setting.

Additionally, as mentioned earlier, we plan to further study the differences we found with respect to algorithmic performance for the different user groups and multimedia domains. Here, we also want to study popularity bias in top- n settings using ranking-aware metrics such as nDCG (e.g., as used in [18]). Finally, we plan to work on further bias mitigation strategies based on cognitive-inspired user modeling and recommendation techniques (e.g., [14, 17, 21]).

Acknowledgements. This research was funded by the H2020 project TRUSTS (GA: 871481) and the “DDAI” COMET Module within the COMET - Competence Centers for Excellent Technologies Programme, funded by the Austrian Federal Ministry for

Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry for Digital and Economic Affairs (bmdw), the Austrian Research Promotion Agency (FFG), the province of Styria (SFG) and partners from industry and academia. The COMET Programme is managed by FFG.

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 (2017)
2. Abdollahpouri, H., Burke, R., Mobasher, B.: Managing popularity bias in recommender systems with personalized re-ranking. In: The Thirty-second International Flairs Conference (2019)
3. Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B.: The unfairness of popularity bias in recommendation. In: RecSys Workshop on Recommendation in Multistakeholder Environments (RMSE) (2019)
4. Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B., Malthouse, E.: User-centered evaluation of popularity bias in recommender systems. In: Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, pp. 119–129 (2021)
5. Adamopoulos, P., Tuzhilin, A.: On over-specialization and concentration bias of recommendations: probabilistic neighborhood selection in collaborative filtering systems. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 153–160 (2014)
6. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* **24**(5), 896–911 (2011)
7. Antikacioglu, A., Ravi, R.: Post processing recommender systems for diversity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 707–716 (2017)
8. Baeza-Yates, R.: Bias in search and recommender systems. In: Fourteenth ACM Conference on Recommender Systems, p. 2 (2020)
9. 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)
10. Brynjolfsson, E., Hu, Y.J., Smith, M.D.: From niches to riches: anatomy of the long tail. *Sloan Manage. Rev.* **47**(4), 67–71 (2006)
11. Deldjoo, Y., Schedl, M., Cremonesi, P., Pasi, G.: Recommender systems leveraging multimedia content. *ACM Comput. Surv. (CSUR)* **53**(5), 1–38 (2020)
12. George, T., Merugu, S.: A scalable collaborative filtering framework based on co-clustering. In: Fifth IEEE International Conference on Data Mining (ICDM 2005), p. 4. IEEE (2005)
13. Jannach, D., Lerche, L., Kamehkhosh, I., Jugovac, M.: What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Model. User Adapt. Interact.* **25**(5), 427–491 (2015). <https://doi.org/10.1007/s11257-015-9165-3>
14. Kowald, D., Lex, E.: The influence of frequency, recency and semantic context on the reuse of tags in social tagging systems. In: Proceedings of Hypertext 2016, pp. 237–242. ACM, New York, NY, USA (2016)

15. Kowald, D., Muellner, P., Zangerle, E., Bauer, C., Schedl, M., Lex, E.: Support the underground: characteristics of beyond-mainstream music listeners. *EPJ Data Sci.* **10**(1), 1–26 (2021). <https://doi.org/10.1140/epjds/s13688-021-00268-9>
16. Kowald, D., Schedl, M., Lex, E.: The unfairness of popularity bias in music recommendation: a reproducibility study. In: Jose, J.M., et al. (eds.) *ECIR 2020. LNCS*, vol. 12036, pp. 35–42. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45442-5_5
17. Lacic, E., Kowald, D., Seitlinger, P.C., Trattner, C., Parra, D.: Recommending items in social tagging systems using tag and time information. In: *Proceedings of the 1st Social Personalization Workshop co-located with the 25th ACM Conference on Hypertext and Social Media*, pp. 4–9. ACM (2014)
18. Lacic, E., Kowald, D., Traub, M., Luzhnica, G., Simon, J.P., Lex, E.: Tackling cold-start users in recommender systems with indoor positioning systems. In: *Poster Proceedings of the 9th ACM Conference on Recommender Systems. Association of Computing Machinery* (2015)
19. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Trans. Indust. Inform.* **10**(2), 1273–1284 (2014)
20. Park, Y.J., Tuzhilin, A.: The long tail of recommender systems and how to leverage it. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*, pp. 11–18 (2008)
21. Seitlinger, P., Kowald, D., Kopeinik, S., Hasani-Mavriqi, I., Lex, E., Ley, T.: Attention please! a hybrid resource recommender mimicking attention-interpretation dynamics. In: *Proceedings of WWW 2015 companion*, pp. 339–345. ACM (2015)
22. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. *ACM Comput. Surv.* **47**(1), 3:1–3:45 (2014)
23. Sun, W., Khenissi, S., Nasraoui, O., Shafto, P.: Debiasing the human-recommender system feedback loop in collaborative filtering. In: *Companion Proceedings of The 2019 World Wide Web Conference*, pp. 645–651 (2019)
24. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fair: A fair top-k ranking algorithm. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1569–1578 (2017)