The Connection Between Popularity Bias, Calibration, and Fairness in Recommendation

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

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

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

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

ABSTRACT

Recently there has been a growing interest in fairness-aware recommender systems including fairness in providing consistent performance across different users or groups of users. A recommender system could be considered unfair if the recommendations do not fairly represent the tastes of a certain group of users while other groups receive recommendations that are consistent with their preferences. In this paper, we use a metric called miscalibration for measuring how a recommendation algorithm is responsive to users' true preferences and we consider how various algorithms may result in different degrees of miscalibration for different users. In particular, we conjecture that popularity bias which is a well-known phenomenon in recommendation is one important factor leading to miscalibration in recommendation. Our experimental results using two real-world datasets show that there is a connection between how different user groups are affected by algorithmic popularity bias and their level of interest in popular items. Moreover, we show that the more a group is affected by the algorithmic popularity bias, the more their recommendations are miscalibrated.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender systems; Calibration; Algorithmic bias; Popularity bias amplification

ACM Reference Format:
Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The Connection Between Popularity Bias, Calibration, and Fairness in Recommendation. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3383313.3418487

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09.
https://doi.org/10.1145/3383313.3418487

1 INTRODUCTION

Recommendations are typically evaluated using measures such as precision, diversity, and novelty [27]. Under such measures, depending on the situation, a recommended list of items may be considered good if it is relevant to the user, is diverse, and also helps the user discover products that s/he would have not been able to discover in the absence of the recommender system.

One of the metrics used to measure recommendation quality is calibration, which measures whether the recommendations delivered to a user are consistent with the spectrum of items the user has previously rated. For example, if a user has rated 70% action movies and 30% romance, the user might expect to see a similar pattern in the recommendations [29]. If this ratio differs from the one in the user's profile, we say the recommendations are miscalibrated.

In addition to the mentioned metrics, recently there has been a growing interest in other aspects of the recommendations such as bias and fairness [6, 9]. There have been numerous attempts to define fairness [22, 28, 31] and it is unlikely that there will be a universal definition that is appropriate across all applications. Recommendation fairness may have different meanings depending on the domain in which the recommender system is operating, the characteristics of different users or groups of users (e.g. protected vs unprotected), and the goals of the system designers. For instance, Mansoury et al. [20] and Eksrand et al. [10] defined fairness as consistent performance across different groups of users. In their experiments, they observed certain groups such as females get less accurate recommendations than males. In addition, authors in [30] define several fairness metrics which focus on having a consistent performance in terms of estimation error across different user groups. In this paper, we use the same definition to measure (un)fairness. In other words, we consider an algorithm to be unfair if it does not provide consistent performance for different groups of users.

In this paper we focus on (mis)calibration as a quality of the recommendations. Miscalibration by itself may not be considered unfair as it could simply mean the recommendations are not personalized enough. However, if different users or groups of users experience different levels of miscalibration in their recommendations, this may indicate an unfair treatment of a group of users. Inspired by the work in [10, 20] we are interested in investigating the potential factors that could lead to inconsistent performance of algorithms for different groups of users.

The class imbalance problem in machine learning typically is one of the main reasons for having unfair classification (inconsistent True Positive and Negative Rates across different classes). For instance, it is known that facial recognition systems have racial bias since different races are not equally represented in the training data [11]. The equivalent of class imbalance in machine learning is popularity bias in recommender systems [7]: popular items are rated frequently, while the majority of other items do not get much attention. Recommendation algorithms are biased towards these popular items [3]. We define algorithmic popularity bias as the tendency of an algorithm to amplify existing popularity differences across items. We measure this amplification through the metric of *popularity lift*, which quantifies the difference between average item popularity in input (user profile) and output (recommendation list) for an algorithm.

It has been shown that popularity bias can lead to certain problems in recommendation such as shifting users' consumption towards more mainstream items over time and even causing homogenization of different groups of users [19]. In this paper, we conjecture that popularity bias can be also an important factor leading to miscalibration of the recommendation lists for different users. We also show that users with different levels of interest in popular items get different levels of miscalibration. Our contributions are as follows:

- **The disparate impact of popularity bias:** We show that different groups of users are affected differently by popularity bias based on how interested they are in popular items.
- **The connection between algorithmic popularity bias and miscalibration:** We show that users who are more affected by algorithmic popularity bias tend to also get less calibrated recommendations.

2 RELATED WORK

The problem of popularity bias and the challenges it creates for the recommender system has been well studied by other researchers [5, 8, 24]. Authors in the mentioned works have mainly explored the overall accuracy of the recommendations in the presence of long-tail distribution in rating data. In addition, some other researchers have proposed algorithms that can control this bias and give more chance to long-tail items to be recommended [2, 4, 15].

Moreover, the concept of fairness in recommendation has been also gaining a lot of attention recently [14, 30]. For example, finding solutions that remove algorithmic discrimination against users belong to a certain demographic information [32] or making sure items from different categories (e.g. long tail items or items belong to different providers) [18] are getting a fair exposure in the recommendations. Our definition of fairness in this paper is aligned with the fairness objectives introduced by Yao and Huang in [30] where they define unfairness as having inconsistent estimation error across different users. We can generalize the *estimation error* to simply be any kind of system performance such as the calibration of the recommendations as defined in [29]. In this paper, we use the same definition for fairness: a system is *unfair* if it delivers different degree of miscalibration to different users.

With regard to looking at the performance of the recommender system for different user groups, Ekstrand et al. [10] showed that some recommendation algorithms give significantly less accurate recommendations to groups from certain age or gender. In addition, Abdollahpouri et al. in [1] discuss the importance of recommendation evaluation with respect to the distribution of utilities given to different stakeholders. For instance, the degree of calibration of the recommendation for each user group (i.e. a stakeholder) could be considered as its utility and therefore, a balanced distribution of these utility values is important for having a fair recommender system.

3 POPULARITY BIAS AND MISCALIBRATION

Popularity bias and miscalibration are both aspects of an algorithm's performance that are computed by comparing the attributes of the input data with the properties of the recommendations that are produced for users. In this section, we define these terms more precisely.

3.1 Miscalibration

One of the interpretations of fairness in recommendation is in terms of whether the recommender provides consistent performance across different users or groups of users. A recommender system could be considered unfair if the recommendations do not fairly represent the tastes of a certain group of users while other groups receive recommendations that are consistent with their preferences. In this paper we use a metric called miscalibration [29] for measuring how a recommendation algorithm is responsive to users' true preferences and we consider how various algorithms may result in different degrees of miscalibration. As we mentioned earlier, miscalibration, if it exists across all users, could simply mean failure of the algorithm to provide accurate personalization. However, when different groups of users experience different levels of miscalibration, this could indicate unfair treatment of certain user groups. From this standpoint, we call a recommender system unfair if it has different levels of miscalibration for different user groups.

In machine learning, a classification algorithm is called calibrated if the predicted proportions of the various classes agree with the actual proportions of classes in the training data. Extending this notion to recommender systems, a calibrated recommender system is one that reflects the various interests of a user in the recommended list, and with their appropriate proportions.

For measuring the miscalibration of the recommendations we use the metric introduced in [29]. Assume u be a user and i be an item. Also, suppose for each item i there is a set of features C describing the item. For example, a song could be Pop or Jazz, or a movie could have genres Action, Romance, Comedy, etc. We use c for each of these individual categories. Also, we assume that each user has rated one or more items, showing interest in features c belonging to those items. We consider two distributions of categories $c \in C$ for each user: one for the items rated by u, P_u, and the other one for all recommended items to u, Q_u.

For each feature c, we measure the ratio of items that have feature c and rated by the user u, $p(c|u)$, and the ratio of items that have feature c and are recommended to the user u, $q(c|u)$, as follows:

$$p(c|u) = \frac{\sum_{i \in \Gamma_u} \mathbb{1}(i \in c)}{|\Gamma_u|}, \qquad q(c|u) = \frac{\sum_{i \in \Lambda_u} \mathbb{1}(i \in c)}{|\Lambda_u|} \tag{1}$$

where $\mathbb{1}(.)$ is the indicator function returning zero when its argument is False and 1 otherwise. Γ_u is the set of items rated by user u and Λ_u is the set of recommended items to user u.

In order to determine if a recommendation list is calibrated to a given user, we need to measure the distance between the two probability distributions P_u and Q_u. We use the Hellinger distance, H, for measuring the statistical distance between these two distributions. We measure miscalibration for user u, $MC(P_u, Q_u)$, as follows:

$$MC(P_u, Q_u) = H(P_u, Q_u) = \frac{\left\|\sqrt{P_u} - \sqrt{Q_u}\right\|_2}{\sqrt{2}} \tag{2}$$

The overall miscalibration for each group G is obtained by averaging $MC(P_u, Q_u)$ across all users u in group G.

3.2 Popularity Bias

Generally, rating data is skewed towards more popular items–there are a few popular items with the majority of the ratings while the rest of the items have far fewer ratings. Although it is true that popular items are popular for a reason, not every user has the same degree of interest towards these items [3, 23]. There might be users who are interested in less popular, niche items. The recommender algorithm should be able to address the needs of those users as well.

Due to this common imbalance in the original rating data, often algorithms propagate and, in many cases, amplify the bias by over-recommending the popular items. In the next sections, we will define a metric for measuring the degree to which popularity bias is propagated / amplified by the recommendation algorithm. We will empirically evaluate how different recommendation algorithms propagate the popularity bias for different groups of users and to what extent it causes their recommendations to be miscalibrated.

3.2.1 Algorithmic Popularity Lift. In order to measure how each algorithm amplifies the popularity bias in its generated recommendations for different user groups, we define *popularity lift*. First, we measure the average item popularity of a group G (i.e. Group Average Popularity) as follows:

$$GAP_p(G) = \frac{\sum_{u \in G} \frac{\sum_{i \in \Gamma_u} \theta(i)}{|\Gamma_u|}}{|G|}, \qquad GAP_q(G) = \frac{\sum_{u \in G} \frac{\sum_{i \in \Lambda} \theta(i)}{|\Lambda_u|}}{|G|} \tag{3}$$

where $\theta(i)$ is the popularity value for item i (i.e. the ratio of users who rated that item) and subscript p and q refer to the profile of user and recommendations given to the user, respectively.

Therefore popularity lift (PL) for group G is defined as:

$$PL(G) = \frac{GAP_q(G) - GAP_p(G)}{GAP_p(G)} \tag{4}$$

Positive values for PL indicate amplification of popularity bias by the algorithm. A negative value for PL happens when, on average, the recommendations are less concentrated on popular items than the users' profile. Moreover, the PL value of 0 means there is no popularity bias amplification.

4 METHODOLOGY

We conducted our experiments on two publicly available datasets. The first one is MovieLens 1M dataset which contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 users [13]. Each movie is associated with at least one genre in this dataset with a total of 18 unique genres in the entire dataset. The second dataset we used is a core-10 Yahoo Movies[1] which contains 173,676 ratings on 2,131 movies provided by 7,012 users. Similarly, in this dataset each movie is associated with at least one genre with a total of 24 genres in the entire datatset.

For all experiments, we set aside a random selection of 80% of the rating data as training set and the remaining 20% as the test set. We used several recommendation algorithms including user-based collaborative filtering ($UserKNN$) [25], item-based collaborative filtering ($ItemKNN$) [26], singular value decomposition (*SVD++*) [16], and biased matrix factorization (*BMF*) [17] to cover both neighborhood based and latent factor models. We also included the *Most-popular* method (a non-personalized algorithm recommending the most popular items to every user) as an algorithm with extreme popularity bias. We tuned each algorithm to achieve its best performance in terms of precision. The precision values for *UserKNN*, *ItemKNN*, *SVD++*, *BMF*, and *Most-popular* on MovieLens, are 0.214, 0.223, 0.122, 0.107, and 0.182, respectively. On Yahoo Movies, these values are 0.13, 0.127, 0.2, 0.047, and 0.1, respectively. We set the size of the generated recommendation list for each user to 10. We used the open source recommendation library *librec-auto* [21] and LibRec 2.0 [12] for running our experiments.

In this paper, we are interested in seeing how different groups of users with varying degree of interest towards popular items are treated by the recommender system. Therefore, we grouped users in both datasets into an arbitrary number of groups (10 in this paper) based on their degree of interest in popular items. That is, we first measure the average popularity of the rated items in each user's profile and then organize them into 10 groups with the first group having the lowest average item popularity (extremely niche-focused users) and the last group with the highest average item popularity (heavily blockbuster-focused users). We denoted these groups as G_1 through G_{10}.

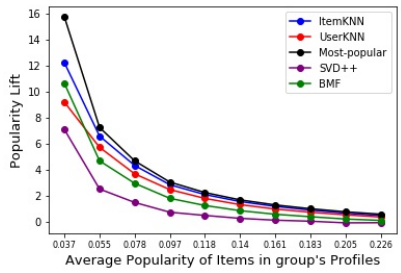
5 RESULTS

5.1 The Connection Between Popularity Bias and Miscalibration

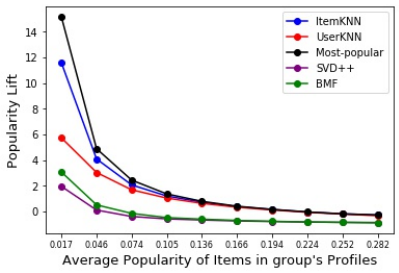
In this section we show the connection between popularity bias and unfairness (disparate miscalibration) in recommendation. Making this connection would be helpful in many fairness-aware recommendation scenarios because fixing the popularity bias could be used as an approach to tackle this type of unfairness.

An illustration of the effect of the algorithmic popularity bias on different user groups is shown in Figure 1. Each dot represents a group with certain average popularity of the users' profiles in that group which is shown on the x-axis. On the y-axis, the popularity lift of different algorithms on each user group is depicted. It can be seen that groups with the lowest average popularity (niche tastes) are being affected the most by the algorithmic popularity bias and the higher the average popularity of the group, the lesser the group is affected by the popularity bias. This shows how, unfairly, popularity bias is affecting different user groups.

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=r



(a) MovieLens



(b) Yahoo Movies

Figure 1: Average item popularity of user groups (G1 through G10 from left to right) and their observed popularity lift in two datasets.

Next, we want to show whether popularity lift has a connection with the miscalibration of the recommendations. Table 1 shows the popularity lift and miscalibration experienced by two extreme groups: G_1 (the most niche-focused group) and G_{10} (the most blockbuster-focused group). It can be seen that, for all of the algorithms, the popularity lift experienced by G_1 is significantly higher than G_{10}. Also, it can be seen that, for all algorithms, the miscalibration value for G_1 is significantly higher than the one for G_{10}. This shows the group with the lowest average item popularity has experienced the highest popularity lift for their recommendations. Moreover, we also saw that this group experienced the highest miscalibration as well. This shows again how popularity lift might be associated with miscalibration.

5.2 Popular Item or Popular Feature?

In Table 1 we observed a connection between the degree of popularity lift experienced by a group of users and how miscalibrated their recommendations are. This connection can be justified as follows: when a list is miscalibrated, it means some genres (or features) are either over-represented or under-represented in the recommendation list compared to the users' interactions. Therefore, theoretically, it is possible for a list to be miscalibrated even when non-popular genres are over-represented in the recommendations. However, what happens in practice is that algorithmic popularity lift increases the recommendation frequency of popular movies and the genres associated with them. As a result, these popular genres become over-represented and thus causing overall miscalibration.

Figure 2 and 3 show the popularity (i.e. frequency) of different genres in rating data and in different recommendation algorithms in MovieLens and Yahoo Movies datasets, respectively. On both datasets, Comedy is the most popular genre. However, interestingly, the recommendations have not amplified this genre as one might expect. The reason is, the popularity of a genre does not necessarily mean every movies associated with that genre are also popular as it could simply be due to the fact that there are many movies with that particular genre. In fact, looking at the plot for MovieLens, we can see that genres Action, Thriller, Sci-Fi, and Adventure are amplified even though they are not the most popular genres. The reason is the most popular movies in this dataset are actually the ones that fall within those genres and since the recommendation algorithms are biased towards popular items (not genres) these genres are amplified. For instance, "Star Wars (episodes IV, V, and VI)", "Jurassic Park (1993)", "Terminator 2 : Judgment Day (1991)", and "The Matrix (1999)" are among the most popular movies and their associated genres are Action, Thriller, Adventure, and Sci-Fi. Similarly, on Yahoo Movies, Action, Adventure, Sci-Fi, Fantasy, Crime, and Gangster are amplified even though some of these genres are not the most popular ones in the rating data. The reason is, most popular movies in this data are "Pirates of the Caribbeans: The curse of the black pearl (2003)", "Terminators 3: Rise of the Machines (2003)", and "The Matrix reloaded (2003)" which are associated with Action, Adventure, Crime, Gangster, and Sci-Fi. This further supports our hypothesis that the popularity bias in recommendation could be a leading factor to miscalibration.

6 CONCLUSION AND FUTURE WORK

In this paper, we looked at the popularity bias problem from the user's perspective and we observed different groups of users can be affected differently by this bias depending on their degree of interest in popular items. In particular, we showed that users who are more affected by popularity bias tend to also receive less calibrated recommendations. For future work, we intend to further study the causality of popularity bias on miscalibration. In particular, we will design experiments such as sampling methods to control popularity bias in data and see the effect of that on miscalibration and fairness. We will also investigate how mitigating algorithmic popularity bias can help to lower miscalibration and unfairness.

ACKNOWLEDGMENTS

Authors Abdollahpouri and Burke were supported in part by the National Science Foundation under Grant Number: 1911025.

REFERENCES

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. 2020. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* 30 (2020), 127–158. Issue 1.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. In *The Thirty-Second International Flairs Conference*.

[3] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided Exposure Bias in Recommendation. In *KDD Workshop on Industrial Recommendation Systems*.

[4] 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

Table 1: The popularity lift (*PL*) and Miscalibration (*MC*) of different recommendation algorithms on two groups G_1 and G_{10}. Bold values show significance difference with $p < 0.05$.

algorithms	MovieLens				Yahoo Movies			
	G_{10}		G_1		G_{10}		G_1	
	PL	*MC*	*PL*	*MC*	*PL*	*MC*	*PL*	*MC*
ItemKNN	0.458	(0.250)	**12.19**	**0.418**	-0.26	0.345	**11.57**	**0.466**
UserKNN	0.348	0.248	**9.17**	**0.446**	-0.31	0.345	**5.738**	**0.395**
Most-popular	0.563	0.277	**15.7**	**0.501**	-0.25	0.342	**15.13**	**0.471**
SVD++	-0.09	0.380	**7.063**	**0.556**	-0.84	0.272	**1.96**	**0.315**
BMF	0.086	0.396	**10.60**	**0.635**	-0.871	0.290	**3.079**	**0.345**

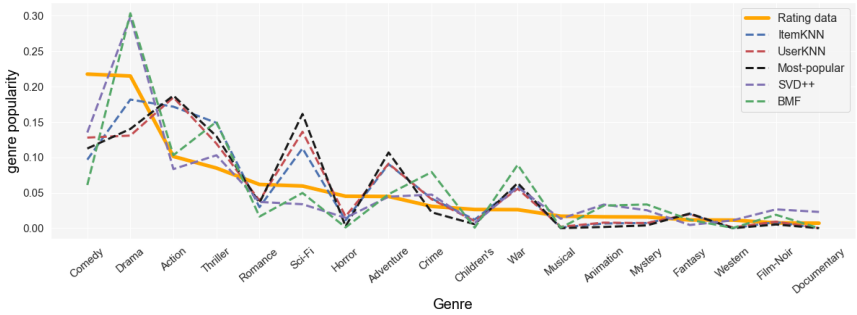


Figure 2: Genre popularity in rating data and in the recommendations (MovieLens)

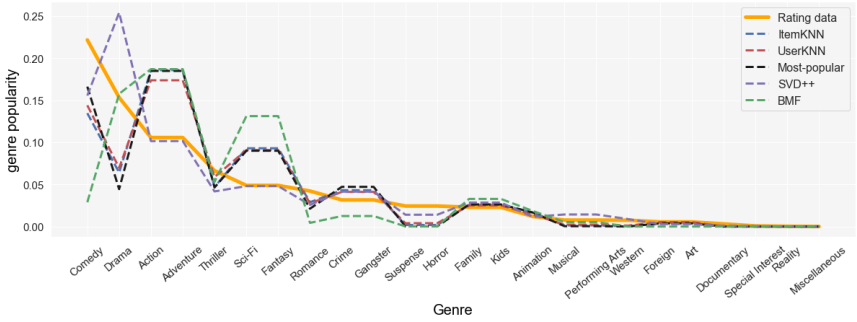


Figure 3: Genre popularity in rating data and in the recommendations (Yahoo Movies)

[5] Chris Anderson. 2006. *The long tail: Why the future of business is selling more for less.* Hyperion.

[6] Engin Bozdag. 2013. Bias in algorithmic filtering and personalization. *Ethics and information technology* 15, 3 (2013), 209–227.

[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] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 214–226.

[10] Michael D Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. In *Conference on Fairness, Accountability and Transparency*. 172–186.

[11] Clare Garvie and Jonathan Frankle. 2016. Facial-recognition software might have a racial bias problem. *The Atlantic* 7 (2016).

[12] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems.. In *UMAP Workshops*.

[13] 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.

[14] 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.

[15] 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

[16] Y. Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[17] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[18] 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.

[19] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback Loop and Bias Amplification in

Recommender Systems. In *arxiv*.
[20] Masoud Mansoury, Himan Abdollahpouri, Jessie Smith, Arman Dehpanah, Mykola Pechenizkiy, and Bamshad Mobasher. 2020. Investigating Potential Factors Associated with Gender Discrimination in Collaborative Recommender Systems. In *The Thirty-Third International Flairs Conference*.
[21] Masoud Mansoury, Robin Burke, Aldo Ordonez-Gauger, and Xavier Sepulveda. 2018. Automating recommender systems experimentation with librec-auto. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 500–501.
[22] Arvind Narayanan. 2018. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, Vol. 1170.
[23] Jinoh Oh, Sun Park, Hwanjo Yu, Min Song, and Seung-Taek Park. 2011. Novel recommendation based on personal popularity tendency. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 507–516.
[24] 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.
[25] P. Resnick and H.R. Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 58.
[26] 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*. ACM, 285–295.
[27] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 257–297.
[28] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.
[29] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 154–162.
[30] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. *CoRR* abs/1705.08804 (2017). http://arxiv.org/abs/1705.08804
[31] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. arXivpreprintarXiv:1507.05259
[32] 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.