Bias Disparity in Recommendation Systems*

Virginia Tsintzou
Department of Computer Science and Engineering
University of Ioannina
vtsintzou@cs.uoi.gr

Evaggelia Pitoura
Department of Computer Science and Engineering
University of Ioannina
pitoura@cs.uoi.gr

Panayiotis Tsaparas
Department of Computer Science and Engineering
University of Ioannina
tsap@cs.uoi.gr

ABSTRACT

Recommender systems have been applied successfully in a number of different domains, such as, entertainment, commerce, and employment. Their success lies in their ability to exploit the collective behavior of users in order to deliver highly targeted, personalized recommendations. Given that recommenders learn from user preferences, they incorporate different *biases* that users exhibit in the input data. More importantly, there are cases where recommenders may amplify such biases, leading to the phenomenon of *bias disparity*. In this short paper, we present a preliminary experimental study on synthetic data, where we investigate different conditions under which a recommender exhibits bias disparity, and the long-term effect of recommendations on data bias. We also consider a simple re-ranking algorithm for reducing bias disparity, and present some observations for data disparity on real data.

1 INTRODUCTION

Recommender systems have found applications in a wide range of domains, including e-commerce, entertainment and social media, news portals, and employment sites [12]. They have been proven to be extremely effective in predicting the preferences of the users, filtering the available content to provide a highly personalized and targeted experience.

One of the most popular classes of recommendation systems is collaborative filtering. Collaborative Filtering (CF) uses the collective behavior of all users over all items to infer the preferences of individual users for specific items [12]. However, given the reliance of CF algorithms on the user preferences, they are susceptible to *biases* that may appear in the input data. In this work we consider biases with respect to the preferences of specific groups of users (e.g., men and women) towards specific categories of items (e.g., different movie genres).

Bias in recommendations is not necessarily always problematic. For example, it is natural to expect gender bias when recommending clothes. However, gender bias is undesirable when recommending job postings, or information content. Furthermore, we want to avoid the case where the recommender system introduces bias in the data, by amplifying existing biases and reinforcing stereotypes. We refer to this phenomenon, where there is a difference between input and recommendation bias, as *bias disparity*.

In this paper, we consider the problem of bias disparity in recommendation systems, and we make the following contributions: (a) We define notions of bias and bias disparity for recommender systems; (b) Using synthetic data we study different conditions under which bias disparity may appear. We consider the effect of the iterative application of recommendation algorithms on the bias of the data; (c) We present some observations on bias disparity on real data, using the MovieLens[1] dataset; (d) We consider a simple re-ranking algorithm for correcting bias disparity, and we study it experimentally.

2 RELATED WORK

The problem of algorithmic bias, and its flip side, fairness in algorithms, has attracted considerable attention in the recent years [4, 5]. Most existing work focuses on classification systems, while there is limited work on recommendation systems.

One type of recommendation bias that has been considered in the literature is popularity bias [3, 6]. It has been observed that under some conditions popular items are more likely to be recommended leading to a rich get richer effect, and there are some attempts to correct this bias [6, 7]. Related to this is also the quest for diversity [8], where the goal is to include different types of items in the recommendations, or provide additional exposure for specific classes of items [1].

These notions of fairness do not take into account the presence of different (protected) groups of users, and different item categories that we consider in this work. This setting is considered in [2], where they define two types of bias, and they propose a modification of the recommendation algorithm in [9] to ensure a fair output. Their work focuses on fairness, rather than bias disparity, and works with a specific algorithm. The notion of bias disparity is examined in [14] but in a classification setting. Closely related to our work is the work in [11], where they consider a similar notion of bias disparity, and they propose *calibrated recommendations* for mitigating its effect. Their work assumes a single class of users, and they treat users individually, rather than as a group.

Fairness in terms of correcting rating errors for specific groups of users was studied in [13] for a matrix factorization CF recommender. A similar setting is considered in [10], where they provide a general framework for defining fairness (either individual or group fairness), and a methodology for enforcing fairness by inserting "antidote data" in the dataset. A notion of fairness for tensor-based recommendations that relies on statistical parity is explored in [15].

3 MODEL

3.1 Definitions

We consider a set of n users $\mathcal{U}$ and a set of m items $\mathcal{I}$. We are given a binary $n \times m$ matrix S, where $S(u, i) = 1$ if user u has selected

Presented at the RMSE workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.

[1] MovieLens 1M: https://grouplens.org/datasets/movielens/1m/

item i, and zero otherwise. Selection may mean that user u liked post i, or that u purchased product i, or that u watched video i.

We assume that users are associated with an attribute A_U, that partitions them into *groups* depending on their value for the attribute. For example, the attribute A_U may be the gender of the user, partitioning the users into men and women. We will typically assume that we have two groups, and one of the groups is the *protected group*. Similarly, we assume that items are associated with an attribute A_I, e.g., the genre of a movie, which partitions them into *categories*, e.g., action and romance movies.

Given the association matrix S, we define the input *preference ratio* $PR_S(G, C)$ of group G for category C as the fraction of selections from group G that are in category C. Formally:

$$PR_S(G, C) = \frac{\sum_{u \in G} \sum_{i \in C} S(u, i)}{\sum_{u \in G} \sum_{i \in \mathcal{I}} S(u, i)} \quad (1)$$

This is essentially the conditional probability that a selection is in category C given that it comes from a user in group G.

To assess the importance of this probability we compare it against the probability $P(C) = |C|/m$ of selecting from category C when selecting uniformly at random. We define the *bias* $B_S(G, C)$ of group G for category C as:

$$B_S(G, C) = \frac{PR_S(G, C)}{P(C)} \quad (2)$$

Bias values less than 1 denote *negative bias*, that is, the group G on average tends to select less often from category C, while bias values greater than 1 denote *positive bias*, that is, that group G favors category C disproportionately to its size.

We assume that the recommendation algorithm outputs for each user u a ranked list of r items R_u. The collection of all recommendations can be represented as a binary matrix R, where $R(u, i) = 1$ if item i is recommended for user u and zero otherwise. Given matrix R, we can compute the output preference ratio of the recommendation algorithm, $PR_R(G, C)$, of group G for category C using Eq. (1), and the output bias $B_R(G, C)$ of group G for category C.

To compare the bias of a group G for a category C in the input data S and the recommendations R, we define the *bias disparity*, that is, the relative change of the bias value.

$$BD(G, C) = \frac{B_R(G, C) - B_S(G, C)}{B_S(G, C)} \quad (3)$$

Our definitions of preference ratios and bias are motivated by concepts of group proportionality, and group fairness considered in the literature [4, 5].

3.2 The Recommendation Algorithm

For the recommendations, in our experiments, we use a user-based K-Nearest-Neighbors (UserKNN) algorithm. The UserKNN algorithm first computes for each user, u, the set $N_K(u)$ of the K most similar users to u. For similarity, it uses the Jaccard similarity, $\mathcal{J}Sim$, computed using the matrix S. For user u and item i not selected by u, the algorithm computes a *utility value*

$$V(u, i) = \frac{\sum_{n \in N_K(u)} \mathcal{J}Sim(u, n) S(n, i)}{\sum_{n \in N_K(u)} \mathcal{J}Sim(u, n)} \quad (4)$$

The utility value $V(u, i)$ is the fraction of the similarity scores of the top-K most similar users to u that have selected item i. To recommend r items to a user, the r items with the highest utility values are selected.

4 BIAS DISPARITY ON SYNTHETIC DATA

In this section, we present experiments with synthetic data. Our goal is to study the conditions under which the UserKNN exhibits bias disparity.

4.1 Synthetic data generation

Users are split into two groups G_1 and G_2 of size n_1 and n_2 respectively, and items are partitioned into two categories C_1 and C_2 of size m_1 and m_2 respectively. We assume that users in G_1 tend to favor items in category C_1, while users in group G_2 tend to favor items in category C_2. To quantify this preference, we give as input to the data generator two parameters ρ_1, ρ_2, where parameter ρ_i determines the preference ratio $PR_S(G_i, C_i)$ of group G_i for category C_i. For example, $\rho_1 = 0.7$ means that 70% of the ratings of group G_1 are in category C_1. The datasets we create consist of 1,000 users and 1,000 items. We assume that each user selects 5% of the items in expectation, and we recommend $r = 10$ items per user.

We perform two different sets of experiments. In the first set, we examine the effect of the preference ratios, and in the second set, the effect of group and category sizes. All reported values are averages over 10 experiments.

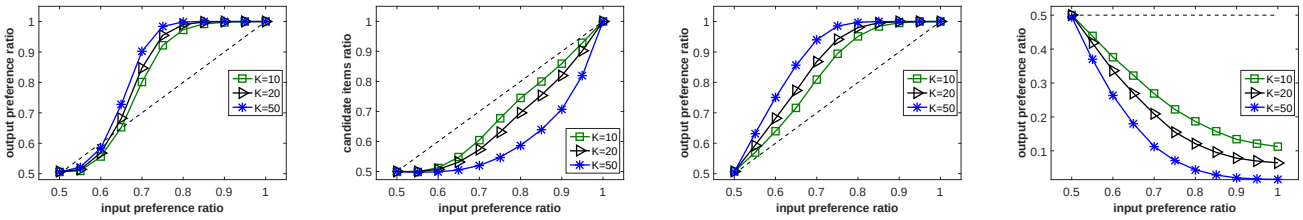
4.2 Varying the preference ratios

In these experiments, we create datasets with equal-size groups G_1 and G_2, and equal-size item categories C_1 and C_2, and we vary the preference ratios of the groups.

4.2.1 Symmetric Preferences: In the first experiment, we assume that the two groups G_1 and G_2 have the same preference ratios by setting $\rho_1 = \rho_2 = \rho$, where ρ takes values from 0.5 to 1, in increments of 0.05. In Figure 1(a), we plot the output preference ratio $PR_R(G_1, C_1)$ (eq. $PR_R(G_2, C_2)$) as a function of ρ. Note that in this experiment, bias is the preference ratio scaled by a factor of two. We report preference ratios to be more interpretable. The dashed line shows when the output ratio is equal to the input ratio and thus there is no bias disparity. We consider different values for K, the number of neighbors. A first observation is that when the input bias is small ($PR_S \leq 0.6$), the output bias decreases or stays the same. In this case, users have neighbors from both groups. For higher input bias ($PR_S > 0.6$), we have a sharp increase of the output bias, which reaches its peak for $PR_S = 0.8$. In these cases, the recommender polarizes the two groups, recommending items only from their favored category.

In Figure 1(b), we report the preference ratio for all candidate items for recommendation for each user (i.e., if the system recommended all items with non zero utility). Surprisingly, the candidate items are less biased even for high values of the input bias. This shows that: (a) Utility proportional to user-similarity increases bias, since the top-r recommendations with the highest utility are significantly more biased, (b) It is possible to reduce bias by re-ranking the candidate items.

Increasing the value of K increases the output bias. Adding neighbors increases the strength of the signal, and the algorithm discriminates better between the items in the different categories, causing



(a) PR_R, symmetric case (b) Ratio of candidate items (c) $PR_R(G_1, C_1)$, asymmetric case (d) $PR_R(G_2, C_2)$, asymmetric case

Figure 1: Experiment with different preference ratios.

it to favor the preferred category. Understanding the role of K is a subject for future study.

4.2.2 Asymmetric Preferences: In this experiment, group G_1 has preference ratio ρ_1 ranging from 0.5 to 1, while G_2 is unbiased with fixed preference ratio $\rho_2 = 0.5$. In Figure 1, we show the recommendation preference ratio for groups G_1 (Figure 1(c)) and G_2 (Figure 1(d)) as a function of ρ_1.

We observe that the output bias of group G_1 is amplified at a rate much higher than in Figure 1(a), while group G_2 becomes biased towards category C_1. Surprisingly, the presence of the unbiased group G_2 has an amplifying effect on the bias of G_1, rather than a moderating one, more so than an oppositely-biased group. This is due to the fact that the users in the unbiased group G_2 provide a stronger signal in favor of category C_1, compared to the symmetric case where group G_2 is biased over C_2. This reinforces the bias in favor of category C_1. As expected, the unbiased group adopts the biases of the biased group

4.3 Varying group and category sizes

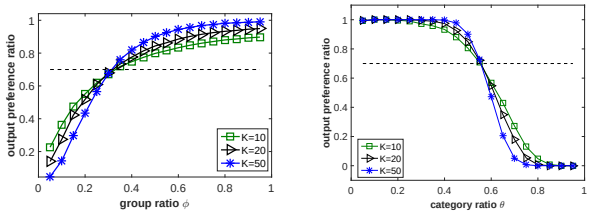
In this experiment we examine bias disparity with unbalanced groups and categories.

4.3.1 Varying Group Sizes: We first consider groups of uneven size. We set the size n_1 of G_1 to be a fraction ϕ of the number of all users n, ranging from 5% to 95%. Both groups have fixed preference ratio $\rho_1 = \rho_2 = 0.7$. Figure 2(a) shows the output recommendation preference ratio $PR_R(G_1, C_1)$ as a function of ϕ. The plot of $PR_R(G_2, C_2)$ is the mirror image of this one, so we do not report it.

We observe that for $\phi \leq 0.3$ group G_1 has negative bias disparity ($PR_R(G_1, C_1) < 0.7$). That is, the small group is drawn by the larger group. For medium values of ϕ in $[0.35, 0.5]$, the bias of both groups is amplified, despite the fact that G_1 is smaller than G_2. The increase is larger for the larger group, but there is increase for the smaller group as well.

We also experimented with the case where G_2 is unbiased. In this case G_2 becomes biased towards C_1 even for $\phi = 0.05$, while the point at which the bias disparity for G_1 becomes positive is much earlier ($\phi \approx 0.2$). This indicates that a small biased group can have a stronger impact than a large unbiased one.

4.3.2 Varying Category Sizes: We now consider categories of uneven size. We set the size m_1 of C_1 to be a fraction θ of the number items m, ranging from 10% to 90%. We assume that both groups have fixed preference ratio $\rho_1 = \rho_2 = 0.7$. Figure 2(b) shows the recommendation preference ratio $PR_R(G_1, C_1)$ as a function of θ. The plot of $PR_R(G_2, C_2)$ is again the mirror image of this one.



(a) Group Size (b) Category Size

Figure 2: (a) Unbalanced group sizes, (b) Unbalanced category sizes; input preference ratio $PR_S(G_i, C_i) = 0.7$.

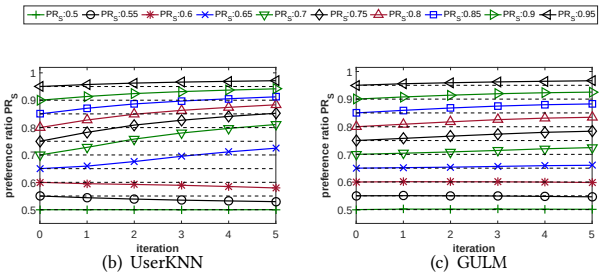
Note that as long as $\theta \leq 0.7$, group G_1 has positive bias (greater than 1) for category C_1 since bias is equal to ρ_1/θ. However, it decreases as the size of the category increases. When the category size is not very large ($\theta \leq 0.5$), the output bias is amplified regardless of the category size. For $\theta > 0.7$, G_1 is actually biased in favor of C_2, and this is reflected in the output. There is an interesting range $[0.6, 0.7]$ where G_1 is positively biased towards C_1 but its bias is weak, and thus the recommendation output is drawn to category C_2 by the more biased group.

4.4 Iterative Application of Recommendations

We observed bias disparity in the output of the recommendation algorithm. However, how does this affect the bias in the data? To study this we consider a scenario where the users accept (some of) the recommendations of the algorithm, and we study the long-term effect of the iterative application of the algorithm on the bias of the data. More precisely, at each iteration, we consider the top-r recommendations of the algorithm ($r = 10$) to a user u, and we normalize their utility values, by the utility value of the top recommendation. We then assume that the user accepts a recommendation with probability equal to the normalized score. The accepted recommendations are added to the data, and they are fed as input to the next iteration of the recommendation algorithm.

We apply this iterative algorithm on a dataset with two equally but oppositely biased groups, as described in Section 4.2.1. The results of this iterative experiment are shown in Figure 3(a), where we plot the average preference ratio for each iteration. Iteration 0 corresponds to the input data. In our experiment a user accepts on average 7 recommendations. For this experiment we set the number of neighbors K to 50.

We observe that even with the probabilistic acceptance of recommendations, there is a clear long-term effect of the recommendation bias. For small values of input bias, we observe a decrease, in line



(b) UserKNN (c) GULM

Figure 3: The evolution of the preference ratio in the data for different input preference ratios (PR_S), after 5 iterations of (a) UserKNN and (b) GULM. Iteration 0 shows the original preference ratio of each experiment.

with the observations in Figure 1(a). For these values of bias, the recommender will result in reducing bias and smoothing out differences. The value of preference ratio 0.6 remains more or less constant, while for larger values the bias in the data increases. Therefore, for large values of bias the recommender has a reinforcing effect, which in the long term will lead to polarized groups of users.

5 BIAS DISPARITY ON REAL DATA

In this experiment, we use the Movielens 1M dataset[2]. We consider as categories the genres Action and Romance, with 468 and 463 movies. We extract a subset of users $\mathcal{U}$ that have at least 90 ratings in these categories, resulting in 1,259 users. Users in $\mathcal{U}$ consist of 981 males and 278 females.

In Table 1, we show the input/output bias and in parentheses the bias disparity for each group-category combination. The right part of the table reports these numbers when the user groups are balanced, by selecting a random sample of 278 males. We observe that males are biased in favor of Action movies while females prefer Romance movies. The application of UserKNN increases the strong input bias for males in the output. Females are moderately biased in favor of Romance movies. Hence, their output bias is drawn to Action items. We observe a very similar picture for balanced data, indicating that the changes in bias are not due to the group imbalance.

Table 1: Gender bias on action and romance

	Unbalanced Groups		Balanced Groups	
	Action	Romance	Action	Romance
M	1.39/1.67 (0.2)	0.58/0.28 (-0.51)	1.40/1.66 (0.18)	0.57/0.29 (-0.49)
F	0.97/1.14 (0.17)	1.03/0.85 (-0.17)	0.97/1.08 (0.11)	1.03/0.92 (-0.10)

6 CORRECTING BIAS DISPARITY

To address the problem of bias disparity, we consider an algorithm that performs post-processing of the recommendations. Our goal is to adjust the set of recommended items, so as to ensure that there is no bias disparity. In addition, we would like the new recommendation set to have the maximum possible utility.

Abusing the notation, let R^* denote the set of user-item pairs produced by our recommendation algorithm, where $(u, i) \in R^*$ denotes that u was recommended item i. We will refer to the pair (u, i) as a recommendation. The set R^* contains r recommendations for each user, thus, rn recommendations in total. Let $V(R^*) = \sum_{(u,i)\in R^*} V(u, i)$ denote the total utility of the recommendations in set R^*. Since R^* contains for each user u the top-r items with the highest utility, R^* has maximum possible utility among all sets with r recommendations per user.

However, as we have seen in our experiments, the set R^* may have high bias disparity. We will adjust the recommendations in the set R^* to produce a new set of recommendations R, with r recommendations per user, with zero bias disparity. Clearly this will come at the expense of utility. Our goal is to find the set R with the *minimum utility loss*.

Since we have two categories, to achieve zero bias disparity, it suffices to have $B_R(G_i, C_i) = B_S(G_i, C_i)$. Without loss of generality assume that $B_{R^*}(G_i, C_i) > B_S(G_i, C_i)$. Let $\overline{C_i}$ denote the category other than C_i. We decrease the output bias B_R by swapping recommendations (u, i) of category C_i with recommendations (u, j) of category $\overline{C_i}$. Given a target bias value, we can compute the number of swaps for achieving zero bias disparity. The utility loss incurred by swapping (u, i) with (u, j) is $V(u, i) - V(u, j)$. The goal is to find the swaps with the minimum utility loss.

We present a simple and efficient greedy algorithm for this task. Let N_S denote the desired number of swaps. The algorithm starts with the set $R = R^*$, and performs N_S steps. At each step it computes a set of candidate swaps by pairing for each user u the lowest-ranked recommendation (u, i) in R from category C_i, with the highest ranked recommendation (u, j) not in R from category $\overline{C_i}$, and performs the swap with the minimum utility loss. It is easy to show that the algorithm is optimal, that is, it achieves the minimum utility loss. We refer to this algorithm as the GULM (Group Utility Loss Minimization) algorithm.

By design, when we apply the GULM algorithm on the output of the recommendation algorithm, we eliminate bias disparity (modulo rounding errors) in the recommendations. We consider the iterative application of the recommendation algorithm, in the setting described in Section 4.4, assuming again that the probability of a recommendation being accepted depends on its utility. The results are shown in Figure 3(b). For values of preference ratio up to 0.65, we observe that bias remains more or less constant after re-ranking. For larger values, there is some noticeable increase in the bias, albeit significantly smaller than before re-ranking. The increase is due to the fact that the recommendations introduced by GULM have low probability to be accepted.

7 CONCLUSIONS

In this short paper, we performed a preliminary study of bias disparity in recommender systems, and the conditions under which it may appear. Using synthetic data, we observed that recommendation algorithms can introduce bias disparity even for moderately biased groups. We view this analysis as a first step towards a systematic analysis of the factors that cause bias disparity. We intend to investigate more recommendation algorithms, and more datasets, including the case of numerical, rather than binary, ratings. It is also interesting to examine this work in the context of other definitions for bias and fairness.

[2]MovieLens 1M: https://grouplens.org/datasets/movielens/1m/

REFERENCES

[1] Alex Beutel, Ed H. Chi, Zhiyuan Cheng, Hubert Pham, and John Anderson. 2017. Beyond Globally Optimal: Focused Learning for Improved Recommendations. In *Proceedings of the 26th International Conference on World Wide Web*.

[2] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Conference on Fairness, Accountability and Transparency, FAT*.

[3] Òscar 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 (NETFLIX)*. ACM.

[4] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science*.

[5] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic Bias: From Discrimination Discovery to Fairness-aware Data Mining. In *ACM SIGKDD*.

[6] 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 (Dec. 2015), 427–491.

[7] 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*.

[8] Matev Kunaver and Toma Porl. 2017. Diversity in Recommender Systems A Survey. *Know.-Based Syst.* (2017).

[9] X. Ning and G. Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *2011 IEEE 11th International Conference on Data Mining*.

[10] Bashir Rastegarpanah, Krishna P. Gummadi, and Mark Crovella. 2019. Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*. 231–239.

[11] Harald Steck. 2018. Calibrated Recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 154–162.

[12] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. in Artif. Intell.* (2009).

[13] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *NIPS*.

[14] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *EMNLP*.

[15] 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 (CIKM '18)*. ACM, New York, NY, USA, 1153–1162.