Popularity-Opportunity Bias in Collaborative Filtering

Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, James Caverlee
Department of Computer Science and Engineering, Texas A&M University
zhuziwei,yunhe,xingzhao,zhan13679,jlwang,caverlee@tamu.edu

ABSTRACT

This paper connects equal opportunity to popularity bias in implicit recommenders to introduce the problem of popularity-opportunity bias. That is, conditioned on user preferences that a user likes both items, the more popular item is more likely to be recommended (or ranked higher) to the user than the less popular one. This type of bias is harmful, exerting negative effects on the engagement of both users and item providers. Thus, we conduct a three-part study: (i) By a comprehensive empirical study, we identify the existence of the popularity-opportunity bias in fundamental matrix factorization models on four datasets; (ii) coupled with this empirical study, our theoretical study shows that matrix factorization models inherently produce the bias; and (iii) we demonstrate the potential of alleviating this bias by both in-processing and post-processing algorithms. Extensive experiments on four datasets show the effective debiasing performance of these proposed methods compared with baselines designed for conventional popularity bias.

CCS CONCEPTS

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

KEYWORDS

recommender systems; statistical parity; equal opportunity; recommendation bias

ACM Reference Format:
Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441820

1 INTRODUCTION

Statistical parity and equal opportunity are two important concepts for studying fairness and bias in classification and recommendation tasks [7, 8, 13, 39, 42]. Statistical parity requires the same *positive rate* over individuals or groups [18, 41]. On the other hand, equal opportunity requires the same *true positive rate* [7, 42]. Because statistical parity investigates algorithmic bias without conditioning on the ground truth, the bias identified and removed based on statistical parity is not necessarily an undesired harmful bias [7, 42].

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441820

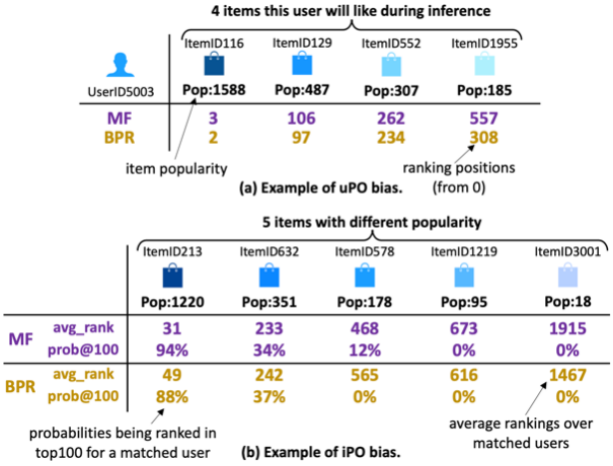

Figure 1: Examples of (a) uPO bias and (b) iPO bias in ML1M.

In this paper, we re-examine popularity bias from the perspective of equal opportunity. We observe that previous studies of popularity bias [3, 4, 6, 9, 10, 27] are mainly governed by statistical parity, and so inherit its limitations. We then connect the concept of equal opportunity to this conventional popularity bias to introduce the new problem of *popularity-opportunity bias* in implicit recommenders.

Suppose we consider the popularity of items as the number of feedback actions toward each item (clicks or views). Conventional popularity bias [3, 4, 6, 9, 10, 27] refers to the phenomenon that high rankings are tend to assigned for popular items at the expense of lower rankings for less popular items. These studies of conventional popularity bias examine the impact of item popularity on recommendation results alone, without taking user preferences into account. That is, the positive rate difference over items of different popularity is calculated for measuring the conventional popularity bias, which is essentially aligned with the concept of statistical parity [8, 13, 39]. However, such a bias definition is problematic because without conditioning on user preferences, the recommendation result (or positive rate) alone is not necessarily evidence of bias. For example, for a user u, one popular item i and one less popular item j, better ranking for the popular item i than the less popular item j is a biased recommendation defined by conventional popularity bias. Yet, if we know that u likes i but dislikes j, then this ranking result is in fact reasonable and not a harmful bias. Moreover, forcing similar rankings for i and j as in previous works [32, 33] to remove conventional popularity bias could actually hurt user satisfaction and engagement of the popular item i.

Thus, inspired by equal opportunity, we propose to investigate the **popularity-opportunity bias**: *conditioned on user preferences that a user likes both items*, is the more popular item more likely to be recommended (or ranked higher) to the user than the less popular one? That is, we calculate the true positive rate difference over items of different popularity for measuring the bias during

testing, and require the true positive rate to be the same for items of different popularity to achieve equal opportunity. To our best knowledge, this is the first work which studies popularity bias from the view of equal opportunity for recommender systems.

To identify popularity-opportunity bias during testing, one critical question is how do we know user preferences to measure the bias? That is, how do we know whether u likes i or j? In practice, the utility of a recommender system is typically evaluated through a train-test split, where a learned model (based on the training data) is evaluated over the testing data, where the testing data contains held-out evidence of user preferences (e.g., by likes, views, or clicks). In a similar way, we can leverage the same testing data as indicators of user preferences to identify popularity-opportunity bias.

User-side popularity-opportunity bias. More specifically, in this paper, we investigate the proposed popularity-opportunity bias from the views of users and items separately. To illustrate, let's first consider the example in Figure 1a. Here we show four items from the MovieLens 1M dataset [14] that user ID5003 likes during testing. That is, these items are not seen during training but are in the test set of this user, and the user will interact with these items once recommended (i.e., they are true positives). Item ID116 is the most popular one with 1588 feedback actions, while item ID1955 is the least popular with only 185 feedback records. Then, we show the ranking positions of these four items for user ID5003 according to two fundamental collaborative filtering models – matrix factorization with Root Mean Square Error loss (denoted as MF) [23] and Bayesian Personalized Ranking loss (denoted as BPR) [30]. We observe that popular items are ranked higher than less popular items by both models, *even though we know the user likes all of them*. We refer to this as *user-side popularity-opportunity bias* or uPO bias for short.

Item-side popularity-opportunity bias. Complementary to this user-side perspective, we show an example of five items in Figure 1b. Item ID213 is the most popular, while item ID3001 is the least popular. If we consider only the matched users who like each item in testing data (i.e., for item i, only the ranking positions for matched users who have i in their test set are considered), we observe that more popular items will have better rankings and higher probabilities of being ranked in the top-100. For example, item ID213 is ranked by MF in the top-100 for 94% of all matched users, whereas item ID3001 is never ranked in the top-100 for its matched users. This reveals a systematic low recommendation opportunity for low-popularity items. We refer to this as *item-side popularity-opportunity bias* or iPO bias for short.

Both this user-side and item-side bias raise critical issues. User-side (uPO) bias is harmful because a user's need corresponding to these low-popularity items is not acknowledged and not satisfied by the recommender. Moreover, low-popularity items sometimes are more important than popular items because they can be serendipitous and novel for users, crucial for extending the area of users' interests and promoting user engagement [6, 31]. Item-side (iPO) bias brings damaging outcomes that long-tail items may not have any chance to become popular or even known, and providers of these items will receive less engagement in the system. In the long-term, iPO bias could accumulate, leading to a recommender dominated by well-known popular items.

Our contributions. Hence, this paper proposes a three-part study of both user-side and item-side popularity-opportunity bias.

i) Figure 1 shows cases of the bias, but is it prevalent beyond these examples? To answer this, we conduct a comprehensive data-driven study over four datasets to investigate the presence of popularity-opportunity bias. We focus on two fundamental collaborative filtering approaches (MF and BPR) that serve as foundations of many recommenders including recent neural ones [16]. We empirically demonstrate both models produce user-side and item-side bias.

ii) While this data-driven study showcases the prevalence of the bias, is it truly inherent to these models or an artifact of these datasets? To answer this, we theoretically analyze the impact of item popularity on ranking by MF and BPR to confirm the existence of the bias in both methods.

iii) Last, we investigate the potential of two approaches to reduce this bias: a post-processing approach to compensate for popularity in recommendation; and an in-processing approach that regularizes predicted scores and item popularity. Through experiments on four datasets, we explore the trade-offs between debiasing effectiveness and recommendation utility, showing the more effective debiasing performance of the two proposed methods over existing debiasing baselines designed for conventional popularity bias.

2 RELATED CONCEPTS

In this section, we discuss two topics that are highly related to the studied popularity-opportunity bias: conventional popularity bias, and item-side recommendation fairness.

Conventional Popularity Bias refers to the phenomenon that recommenders tend to assign high rankings for popular items at the expense of lower recommendation opportunities for less popular items [3, 4, 6, 9, 10, 27]. This concept and its influence on recommendations has been studied in [6, 9, 27], and later, Jannach et al. [17] empirically showed that different recommendation algorithms have different vulnerabilities to popularity bias. Long-tail items are considered valuable because they often represent novelty and serendipity [6, 11, 31], thus, they are important in terms of promoting user satisfaction and preventing the monopoly by big brands [3]. To mitigate the harmful effects of popularity bias, many debiasing approaches have been proposed [2–4, 20, 32, 33].

However, existing works [3, 4, 10, 20] mainly study the effects of item popularity on the ranking results themselves – e.g., are popular items recommended more often or ranked higher than less popular ones? – without considering what are the user preferences toward them (aligned with the concept of statistical parity). This is problematic because without conditioning on user preferences, recommendation difference is not necessarily evidence of bias. Thus, we propose popularity-opportunity bias in this work, which studies the impact of item popularity conditioned on user preferences (which is aligned with the concept of equal opportunity). Furthermore, most prior works study the group-level impact of popularity on recommendations by grouping items based on their popularity [2–5, 10, 17, 20]. These studies often consider two groups – popular items vs. long-tail items – which ignores the subtle distinction between individual items at different ranking positions. In contrast, this paper directly investigates rankings and popularity of individual items.

Item-side Recommendation Fairness is another related concept to the popularity-opportunity bias, which studies whether the recommender system treats different groups of items differently. These groups are often determined by sensitive attributes (e.g., gender, race). For example, some works study statistical parity based fairness [18, 19, 21, 25, 38, 41], to see whether different groups of items receive equal exposure in the recommender. Some recent works take user preferences into account to study equal opportunity based item group fairness [7, 12, 28], which is similar to the philosophy of this paper, but we consider the equal opportunity for individual items based on their item popularity. Many researchers have explored methods to enhance recommendation fairness for items [7, 12, 18, 19, 21, 25, 28, 38, 41]. Our work complements these prior efforts as popularity is one key reason driving unfairness for different groups of items.

3 PRELIMINARIES

In this section, we first describe the implicit recommendation problem, then introduce matrix factorization based collaborative filtering models with two different objective functions.

Implicit Recommendation. Suppose we have a user set $\mathcal{U} = \{1, 2, \ldots, N\}$ and an item set $\mathcal{I} = \{1, 2, \ldots, M\}$. We need to recommend a list of k items to every user u based on her implicit feedback record $O_u^+ = \{i, j, \ldots\}$, where $i, j, \ldots$ are the items u has provided positive feedback to before, which are used as training data for model learning. Besides, we have another item set $\widetilde{O}_u^+$ to represent the items that user will like during testing, which are the test data for evaluating recommendation utility and recommendation bias.

Matrix Factorization. Matrix factorization based collaborative filtering [23, 30] is the foundation of many state-of-the-art recommendation models [15, 24], as well as recent neural-network based models [16, 35, 36] that use matrix factorization as the final layer for predicting preference scores. The main idea is to learn low-dimensional latent representations for users and items based on existing user-item interactions, and then to predict preference scores for unobserved user-item pairs by the dot-product of latent representations: $\widehat{\mathbf{R}}_{u,i} = \mathbf{P}_u^\top \mathbf{Q}_i$, where $\mathbf{P}_u \in \mathbb{R}^{H\times 1}$ is the latent representation of user u, $\mathbf{Q}_i \in \mathbb{R}^{H\times 1}$ is the latent representation of item i, and H is the latent dimension.

There are two main categories of objective functions for matrix factorization models: point-wise objective functions (include Root Mean Square Error (RMSE) [23], Cross-Entropy [16], among others) and pair-wise objective functions (include Bayesian Personalized Ranking loss (BPR) [30], Hinge loss [40], and others). Since RMSE and BPR are two of the most widely applied objective functions, we focus on these two in the rest of the paper. We denote the matrix factorization model with RMSE as **MF**, and the one with BPR loss as **BPR**. The formulations are shown below:

$$\min_{\mathbf{\Theta}} \mathcal{L}_{MF} = \sum_{u\in\mathcal{U}} \sum_{i\in O_u^+ \cup O_u^-} \sqrt{(\widehat{\mathbf{R}}_{u,i} - \mathbf{R}_{u,i})^2}, \tag{1}$$

$$\min_{\mathbf{\Theta}} \mathcal{L}_{BPR} = -\sum_{u\in\mathcal{U}} \sum_{\substack{i\in O_u^+ \\ j\in O_u^-}} ln\, \sigma(\widehat{\mathbf{R}}_{u,i} - \widehat{\mathbf{R}}_{u,j}), \tag{2}$$

Table 1: Characteristics of the four public datasets.

	#users	#items	density	pop_avg	pop_std
ML1M	6,040	3,260	3.55%	214.41	276.85
Ciao	5,047	8,102	0.21%	10.82	19.13
Epinions	12,168	11,283	0.18%	21.88	33.07
App	16,201	4,869	0.23%	37.96	66.34

where O_u^- is the randomly sampled negative item set for u; $\sigma(\cdot)$ is the Sigmoid function; and $\mathbf{\Theta}$ represents the model parameters, i.e., the latent representations for users and items $\mathbf{P}$ and $\mathbf{Q}$.

4 DATA-DRIVEN STUDY

In this section, we conduct a data-driven study of popularity-opportunity bias over four datasets, and show how MF and BPR are vulnerable to this bias on both user (uPO bias) and item (iPO bias) sides. While many previous studies have identified conventional popularity bias, this is the first to identify popularity-opportunity bias.

We adopt four widely used datasets from different domains: ML1M [14], Ciao [34], Epinions [34], Amazon-App [26]. For all datasets, we consider the rating or reviewing behaviors as positive feedback from users to items, and regard the number of feedback actions an item receives as its popularity. We first filter out users and items with interactions fewer than 10, and then randomly split them into 60%, 20%, and 20% for training, validation, and testing. The details of these datasets are presented in Table 1, where pop_avg shows the average popularity of the items and pop_std shows the standard deviation of item popularity.

We train MF and BPR models by the training sets of these datasets; tune hyper-parameters by grid search on validation sets; and report the results on test sets. Further details of the experimental setup can be found in Section 7.1.

4.1 Measuring uPO and iPO Bias

First, we introduce two metrics to measure uPO and iPO bias. Similar to recommendation utility metrics, such as *NDCG*, the two introduced bias metrics are calculated based on the test item set $\widetilde{O}_u^+$ for each user u.

Measuring uPO bias. For uPO bias, we want to know for each user u, among all items u will like during testing (items in $\widetilde{O}_u^+$), whether less popular items are ranked lower than more popular ones, i.e., whether the rankings are correlated with popularity given items are liked by the user. Thus, for each user u, we calculate the *Spearman's rank correlation coefficient* between the popularity of items in $\widetilde{O}_u^+$ and their ranking positions, then average all users to have the *popularity-rank correlation for users* (denoted as ***PRU***):

$$PRU = -\frac{1}{N}\sum_{u\in\mathcal{U}} SRC(pop(\widetilde{O}_u^+), rank_u(\widetilde{O}_u^+)), \tag{3}$$

where $SRC(\cdot,\cdot)$ calculates Spearman's rank correlation; $pop(\cdot)$ returns item popularity (it counts the number of feedback actions for each item) for given items; and $rank_u(\widetilde{O}_u^+)$ returns the rankings (from 0 to $M-1$, 0 represents the top-most ranking) of given items for user u by a specific model. Spearman's rank correlation coefficient assesses the monotonic relationship between two variables and has values in the range $[-1, 1]$. Hence, a large positive value (note that we add a negative sign before $SRC(\cdot,\cdot)$ to flip the

Table 2: Measuring uPO bias (*PRU*) and iPO bias (*PRI*) for MF and BPR on four datasets. * indicates that the Spearman's rank correlation coefficients are statistically significant for $p < 0.01$ judged by t-test.

	ML1M		Ciao		Epinions		App	
	MF	BPR	MF	BPR	MF	BPR	MF	BPR
PRU	0.835	0.779	0.542	0.591	0.684	0.708	0.567	0.636
PRI	0.980*	0.969*	0.363*	0.433*	0.535*	0.573*	0.609*	0.692*

sign) of *PRU* means that low popularity leads to low rankings for items a user likes during testing, which violates the requirement of equal opportunity for items of different popularity as discussed in Section 1, i.e., high uPO bias.

Measuring iPO bias. For iPO bias, we want to know whether the expected rankings of low-popularity items for matched users are lower than the expected rankings of high-popularity items, i.e., whether the expected ranking position of an item for a matched user is correlated with its popularity. Hence, we calculate the Spearman's rank correlation coefficient between the popularity of all items and their average ranking positions over matched users (for each item i, fetch all the users who have i in test set $\widetilde{O}_u^+$, and then average the ranking positions in the ranking lists of these users) to have the *popularity-rank correlation for items* (denoted as ***PRI***):

$$PRI = -SRC(pop(\mathcal{I}), avg\_rank(\mathcal{I})),$$

where $avg\_rank(i) = \frac{1}{|\widetilde{\mathcal{U}}_i|} \sum_{u \in \widetilde{\mathcal{U}}_i} rank_u(i)$ returns the average ranking for item i over the set of matched user $\widetilde{\mathcal{U}}_i$ (i.e., for each $u \in \widetilde{\mathcal{U}}_i$, i is in $\widetilde{O}_u^+$)). A large positive value of *PRI* means that lower popularity leads to worse rankings, violating the requirement of equal opportunity, i.e., high iPO bias. In our experiments, we also evaluate the iPO bias by calculating the probability of being ranked in top-k for a matched user (as examples in the Figure 1b), which shows similar pattern as the introduced metric *PRI*. Thus, in this paper, we will only report results based on *PRI*.

Compare *PRU* and *PRI*. Both *PRU* and *PRI* measure popularity-opportunity bias. The main difference is how they calculate the popularity-ranking correlation and aggregate across users. Due to this calculation difference, *PRU* and *PRI* measure different aspects of popularity-opportunity bias. *PRU* represents the expectation of popularity-ranking correlation of matched items a random user will get from a model, which is to say, it quantifies the bias from the view of users. On the other hand, *PRI* measures the correlation between item popularity and the expectation of ranking position from matched users for items, which is to say, it quantifies the bias from the view of items. Although in practice, these two metrics usually show similar patterns, they are essentially not the same. It is possible that a model generates high uPO bias measured by *PRU* while low iPO bias measured by *PRI*, or vice versa. Hence, it is necessary to study the proposed popularity-opportunity bias from both *PRU* and *PRI* perspectives.

4.2 Observations

In the following, we report our observations of uPO and iPO bias for MF and BPR over the four datasets.

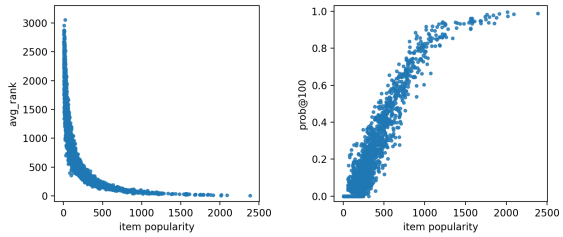

Figure 2: Scatter plots of ranking results by MF on ML1M.

Observations of uPO bias. First, we show *PRU* for both MF and BPR across all four datasets in Table 2. We can see that for both MF and BPR on all datasets, *PRU* values are large positive numbers, indicating both MF and BPR produce uPO bias. More precisely, for a user, even if we know that two items are equally liked by the user, the more popular one will have better ranking position than the less popular one. Note that we do not show the significance test results for *PRU* because the size of $\widetilde{O}_u^+$ in Equation 3 is small for most of the users which makes the significance test uninformative (because the p-value is always large when only few instances are included). An example of such uPO bias in ML1M dataset is shown in Figure 1a, which is consistent with our observations from Table 2.

Observations of iPO bias. Next, we focus on the metric *PRI* to evaluate the iPO bias in Table 2. For all four datasets and both models, *PRI* are large positive values, which means in the recommendations by MF and BPR, items with high popularity have better expected rankings for their matched users, while the opposite holds for low-popularity items. Thus, we can confirm that MF and BPR produce the iPO bias.

To better show the effects of iPO bias, we present two scatter plots in Figure 2 for ranking results of MF on ML1M data (BPR and other datasets have similar patterns). Each dot represents one item. In the left figure, we plot the average rankings of items over matched users (y-axis) vs. popularity (x-axis), from which we can observe a monotonic decreasing trend for the average rankings as the popularity increases. In the right figure, for each item, we plot the probability of being ranked in the top-100 for matched users (y-axis) vs. popularity (x-axis), where we see a monotonic increasing trend for the recommendation probabilities when the popularity increases. These observations are consistent with the conclusions drawn from the bias metric shown in Table 2 that more popular items have better rankings for matched users than less popular items do. Real examples of such iPO bias in ML1M dataset are presented in Figure 1b.

5 THEORETICAL STUDY

After empirically confirming the existence of bias in MF and BPR, we turn in this section to theoretically analyze the relationship between item popularity and ranking results generated by MF and BPR under two simplifying assumptions, to confirm the existence of uPO and iPO bias in MF and BPR.

5.1 Existence of Bias in MF

We first formulate the input and output of the MF model. Given a training user-item interaction matrix $\mathbf{R} \in \{0,1\}^{N \times M}$ with N users, M items, 1 represents a known user-item interaction, and 0

represents an unknown user-item relationship. If we train an MF model on $\mathbf{R}$, we can get a user latent representation matrix $\mathbf{P} \in \mathbb{R}^{H\times N}$ and an item latent representation matrix $\mathbf{Q} \in \mathbb{R}^{H\times M}$. Now we have **Assumption 1**: we assume the model is trained in an ideal condition where the loss function in Equation 1 is minimized close to 0. Then the dot product of the latent matrices will reconstruct $\mathbf{R}$ with very minor error: $\mathbf{P}^\top\mathbf{Q} = \widehat{\mathbf{R}}$ and $\|\widehat{\mathbf{R}} - \mathbf{R}\|_F^2 < \epsilon$. This is to say that $\widehat{\mathbf{R}}_{u,i} \approx 1$ if $\mathbf{R}_{u,i} = 1$, and $\widehat{\mathbf{R}}_{u,i} \approx 0$ if $\mathbf{R}_{u,i} = 0$. We represent the reconstructed interaction matrix as $\widehat{\mathbf{R}} \in \{\sim 0, \sim 1\}^{N\times M}$, where ~ 0 and ~ 1 are numbers very close to 0 and 1. Without loss of generality, we assume values in $\widehat{\mathbf{R}}$ are non-negative because we can always add a positive constant to $\widehat{\mathbf{R}}$ to make all elements positive without changing the ranking results.

Because the number of ~ 1 values in columns of $\widehat{\mathbf{R}}$ can indicate the item popularity, we introduce the item popularity information to the formulations of $\mathbf{P}$ and $\mathbf{Q}$ by $\widehat{\mathbf{R}}$. Given a user u, the predicted preference scores for her toward all items can be calculated by $\mathbf{P}_u^\top\mathbf{Q} = \widehat{\mathbf{R}}_{u,:}$, where $\widehat{\mathbf{R}}_{u,:} \in \{\sim 0, \sim 1\}^{1\times M}$ is the u-th row in $\widehat{\mathbf{R}}$. Moving $\mathbf{Q}$ to the right-hand side by pseudo-inverse, we can have $\mathbf{P}_u^\top = \widehat{\mathbf{R}}_{u,:}\mathbf{Q}^\top(\mathbf{Q}\mathbf{Q}^\top)^{-1}$. Similarly, we have $\mathbf{Q}_i = (\mathbf{P}\mathbf{P}^\top)^{-1}\mathbf{P}\widehat{\mathbf{R}}_{:,i}$, where $\widehat{\mathbf{R}}_{:,i} \in \{\sim 0, \sim 1\}^{N\times 1}$ is the i-th column in $\widehat{\mathbf{R}}$.

Based on the new formulations of $\mathbf{P}_u$ and $\mathbf{Q}_i$, we define several new matrices for the analysis. First, we define the *normalized user latent representation*: $\mathbf{A} = (\mathbf{P}\mathbf{P}^\top)^{-1}\mathbf{P}$, which normalizes $\mathbf{P}$ by the variances of its principal components over the principal component directions. The explanation for $\mathbf{A}$ is that $\mathbf{P}\mathbf{P}^\top$ can be factorized as $\mathbf{P}\mathbf{P}^\top = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ by Eigen-Decomposition, where $\mathbf{U}$ is an orthogonal matrix ($\mathbf{U}^\top = \mathbf{U}^{-1}$) with eigenvectors of $\mathbf{P}\mathbf{P}^\top$ as columns, and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of $\mathbf{P}\mathbf{P}^\top$ as diagonal elements. Then based on the definition of Principal Component Analysis [37], $\mathbf{U}^\top\mathbf{P}$ are the principal components of $\mathbf{P}$, $\mathbf{\Lambda}$ are the variances of these principal components. As a result, $\mathbf{A} = (\mathbf{P}\mathbf{P}^\top)^{-1}\mathbf{P} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^\top\mathbf{P}$, i.e., $\mathbf{P}$ is first transformed to the principal component space by $\mathbf{U}^\top$, then normalized by the variances of principal components by $\mathbf{\Lambda}^{-1}$, and last, transformed back to the original space by $\mathbf{U}$. In the same way, we can have the *normalized item latent representation*: $\mathbf{B} = (\mathbf{Q}\mathbf{Q}^\top)^{-1}\mathbf{Q}$, and the *normalized preference matrix*: $\mathbf{Z} = \mathbf{A}^\top\mathbf{B} \in \mathbb{R}^{N\times M}$ (values in $\mathbf{Z}$ are non-negative because all calculations do not change sign).

Now we can derive the predicted score for a user-item pair. Given user u will like item i during testing (i is in $\widetilde{O}_u^+$):

$$\widehat{\mathbf{R}}_{u,i}^+ = \mathbf{P}_u^\top\mathbf{Q}_i = \widehat{\mathbf{R}}_{u,:}\mathbf{B}^\top\mathbf{A}\widehat{\mathbf{R}}_{:,i} = \widehat{\mathbf{R}}_{u,:}\mathbf{Z}^\top\widehat{\mathbf{R}}_{:,i} = \sum \widehat{\mathbf{R}}_{u,:}^\top\widehat{\mathbf{R}}_{:,i} \odot \mathbf{Z}^\top, \quad (4)$$

where $\widehat{\mathbf{R}}_{u,i}^+$ represents the predicted preference score from u to i given the ground truth for this user-item pair is positive; $\odot$ is the Hadamard product; and $\sum\mathbf{D}$ ($\mathbf{D}$ is a matrix) is to sum up all elements of $\mathbf{D}$. The intuitive way to interpret Equation 4 needs two steps: i) First, $\widehat{\mathbf{R}}_{u,:}^\top\widehat{\mathbf{R}}_{:,i} \in \{\sim 0, \sim 1\}^{M\times N}$ is the process to select *key user-item pairs* from a user candidate set $\mathcal{U}_i$ and an item candidate set $\mathcal{I}_u$ that help to indicate preference from u to i, where $\mathcal{U}_i$ are the users who like i in the training set, and $\mathcal{I}_u$ are the items u likes in the training set. Because $\mathcal{U}_i$ reveals characteristics of i and $\mathcal{I}_u$ reveals preferences of u, we can infer $\widehat{\mathbf{R}}_{u,i}$ based on the preferences of $\mathcal{U}_i$ toward $\mathcal{I}_u$, and elements with value ~ 1 in $\widehat{\mathbf{R}}_{u,:}^\top\widehat{\mathbf{R}}_{:,i}$ indicates these key user-item pairs. ii) Then, $\sum\widehat{\mathbf{R}}_{u,:}^\top\widehat{\mathbf{R}}_{:,i} \odot \mathbf{Z}^\top$ retrieves the preference scores of the selected key user-item pairs in $\mathbf{Z}$ and sums them up as $\widehat{\mathbf{R}}_{u,i}^+$.

To simplify Equation 4, we have **Assumption 2**: we assume the preference scores in $\mathbf{Z}$ for key user-item pairs follow the same distribution. The intuitive interpretation of this assumption is that similar users (and similar items) share similar feedback patterns. Or from another aspect, any positive user-item interaction can be inferred by other user-item relationships. Based on this assumption, we denote the expectation of the preference score in $\mathbf{Z}$ for a key user-item pair as $\mathbf{E}[\mathbf{Z}_+]$ ($\mathbf{E}[\mathbf{Z}_+]$ is non-negative). We can further derive Equation 4 as:

$$\widehat{\mathbf{R}}_{u,i}^+ = \sum\widehat{\mathbf{R}}_{u,:}^\top\widehat{\mathbf{R}}_{:,i} \odot \mathbf{Z} = (\sum\widehat{\mathbf{R}}_{u,:})(\sum\widehat{\mathbf{R}}_{:,i})\mathbf{E}[\mathbf{Z}_+].$$

Theorem 5.1. *Given Assumption 1 and 2, MF produces uPO bias.*

Proof. Suppose user u will like items i and j during testing, and i is more popular than j, i.e., $(\sum\mathbf{R}_{:,i}) > (\sum\mathbf{R}_{:,j})$, which is also equivalent to $(\sum\widehat{\mathbf{R}}_{:,i}) > (\sum\widehat{\mathbf{R}}_{:,j})$, the difference between predicted preference scores of the two is:

$$\widehat{\mathbf{R}}_{u,i}^+ - \widehat{\mathbf{R}}_{u,j}^+ = (\sum\widehat{\mathbf{R}}_{u,:})((\sum\widehat{\mathbf{R}}_{:,i}) - (\sum\widehat{\mathbf{R}}_{:,j}))\mathbf{E}[\mathbf{Z}_+] > 0,$$

which is to say for user u, even though both items are liked by u, the lower popularity of j makes it have a worse ranking than i in the recommendation list for u, i.e., MF produces uPO bias. □

Theorem 5.2. *Given Assumption 1 and 2, MF produces iPO bias.*

Proof. First, we formulate the expectation of the preference score of item i from matched users as:

$$\mathbf{E}[\widehat{\mathbf{R}}_{:,i}^+] = \mathbf{E}[(\sum\widehat{\mathbf{R}}_{u,:})](\sum\widehat{\mathbf{R}}_{:,i})\mathbf{E}[\mathbf{Z}_+],$$

where $\mathbf{E}[(\sum\widehat{\mathbf{R}}_{u,:})]$ is the expectation of the sum of predicted scores for a user, which is independent with items. Hence, given two items i, j, where i is more popular than j, we calculate the difference between expected scores of i and j:

$$\mathbf{E}[\widehat{\mathbf{R}}_{:,i}^+] - \mathbf{E}[\widehat{\mathbf{R}}_{:,j}^+] = \mathbf{E}[(\sum\widehat{\mathbf{R}}_{u,:})]((\sum\widehat{\mathbf{R}}_{:,i}) - (\sum\widehat{\mathbf{R}}_{:,j}))\mathbf{E}[\mathbf{Z}_+] > 0,$$

which is to say that the lower popularity of j brings worse expected ranking for users who like j than i, i.e., MF produces iPO bias. □

5.2 Existence of Bias in BPR

In a similar fashion, we analyze the bias in BPR. Due to the pair-wise BPR loss, we cannot directly apply the same process in Section 5.1 to BPR. Thus, we need to first transform a BPR model to an MF one.

Because the pair-wise objective function in BPR is calculated by fixing a user and then computing the difference of predicted scores between one pair of positive and negative items, the output matrix $\widehat{\mathbf{R}}$ is not an approximated version of $\mathbf{R}$ as in MF. Instead, a well trained BPR model will have $\widehat{\mathbf{R}}$ where $\sigma(\widehat{\mathbf{R}}_{u,i} - \widehat{\mathbf{R}}_{u,j}) \approx 1$ given $\mathbf{R}_{u,i} = 1$ and $\mathbf{R}_{u,j} = 0$. Without loss of generality, we can remove the Sigmoid function, and assume that $\widehat{\mathbf{R}}_{u,i} - \widehat{\mathbf{R}}_{u,j} \approx a$ (a is a large positive number) for $\mathbf{R}_{u,i} = 1$ and $\mathbf{R}_{u,j} = 0$. Besides, we define a vector $\mathbf{x} \in \mathbb{R}^{N\times 1}$ to record the expectations of predicted scores for items not in the training set (i.e., $\mathcal{I} \setminus O_u^+$) for each user as $\mathbf{x}_u = \mathbf{E}[\widehat{\mathbf{R}}_{u,\mathcal{I}\setminus O_u^+}]$. Now, for user u, $\widehat{\mathbf{R}}_{u,:}$ is a vector consisting of values close to $\mathbf{x}_u$ and $\mathbf{x}_u + a$, denoted as $\sim \mathbf{x}_u$ values and $\sim (\mathbf{x}_u + a)$

values, where $\sim \mathbf{x}_u$ are for items in $\mathcal{I} \setminus O_u^+$ and $\sim (\mathbf{x}_u + a)$ are for items in O_u^+.

Next, we define a *centralized preference matrix* $\widetilde{\mathbf{R}} \in \{\sim 0, \sim 1\}^{N \times M}$ by subtracting $\mathbf{x}_u$ and dividing a for each user: $\widetilde{\mathbf{R}} = \frac{1}{a}(\widehat{\mathbf{R}} - \mathbf{J} \circ \mathbf{x})$, where $\mathbf{J} = \{1\}^{N \times M}$; and $\circ$ times elements of $\mathbf{x}$ to corresponding rows of $\mathbf{J}$. $\widetilde{\mathbf{R}}$ contains ~ 0 and ~ 1 values, which is exactly the same as the $\widehat{\mathbf{R}}$ in for MF. Meanwhile, $\widetilde{\mathbf{R}}$ maintains the item ranking orders for all users compared with $\widehat{\mathbf{R}}$ generated by BPR because the ranking is executed for each row of $\widehat{\mathbf{R}}$, thus, subtracting and dividing constants will not change the order of the elements in one row. Then, we have a new user latent representation matrix:

$$\widetilde{\mathbf{P}} = \frac{1}{a}(\mathbf{P} - \mathbf{J} \circ \mathbf{x}\mathbf{Q}^\top(\mathbf{Q}\mathbf{Q}^\top)^{-1}),$$

so that $\widetilde{\mathbf{P}}^\top \mathbf{Q} = \widetilde{\mathbf{R}}$. Now, we transform the original BPR model with latent matrices $\mathbf{P}$ and $\mathbf{Q}$ to a new model with $\widetilde{\mathbf{P}}$ and $\mathbf{Q}$, where the two models have the same recommendation results. Last, we can easily apply the same analysis process for MF to the new model to prove the existence of uPO and iPO bias in BPR.

6 DEBIASING APPROACHES

After empirically and theoretically studying popularity-opportunity bias in matrix factorization models, we next explore several approaches to alleviate this bias. Many methods [2–4, 20, 32, 33] have been studied for alleviating conventional popularity bias, which aim to promote the rankings of low-popularity items in the recommendations. These methods can also help promote the rankings of low-popularity items for matched users, which may mitigate the popularity-opportunity bias investigated in this paper. However, this could also promote the rankings of low-popularity items for unmatched user, which could significantly degrade the overall recommendation utility. Hence, we explore debiasing methods that are designed explicitly for the popularity-opportunity bias.

Typically, there are three categories of methods: *pre-processing* [29], *post-processing* [4, 25], and *in-processing* [3, 7, 38] methods. Pre-processing approaches modify the training data so that models trained on the purified data are free of undesired issues (like bias). However, these kinds of algorithms are usually hard to design and may be ineffective since they cannot remove the algorithmic bias inherent in model architectures.

Hence, we focus here on the potential of post-processing and in-processing approaches to alleviate the bias. Concretely, we propose a simple but effective post-processing algorithm – Popularity Compensation (**PC** for short) and a regularization-based in-processing debiasing model (**Reg** for short).

6.1 Post-processing: Popularity Compensation

We begin by investigating a post-processing approach that modifies the predicted user-item preference matrix $\widehat{\mathbf{R}}$ by adding compensation to items with small popularity so that they have higher preference scores and thus higher ranking positions. We propose such a *popularity compensation* that follows three key guidelines:

Guideline 1: Compensation should follow item popularity: items with lower popularity should be compensated more.

Guideline 2: Compensation should follow user preferences: items with higher probabilities of being liked by a user should be compensated more.

Guideline 3: Compensation should follow the value scale of each user: for a user who has a larger value scale for $\widehat{\mathbf{R}}_u$, item candidates for her should be compensated more.

Guideline 1 promotes low-popularity items to mitigate the bias. Guideline 2 ensures that items a user does not like but with low popularity will not be mistakenly promoted by the algorithm. Guideline 3 makes sure that users with large value scales of predicted preference scores will have large compensation to items so that the algorithm is effective to all users.

Based on these guidelines, we propose the Popularity Compensation (PC) debiasing algorithm. Given a user u, we have the user-item interaction records in the training data $\mathbf{R}_{u,:} \in \{0, 1\}^{1 \times M}$, the interacted item set in the training data O_u^+, and the predicted preference scores from u to items generated by MF or BPR $\widehat{\mathbf{R}}_{u,:} \in \mathbb{R}^{1 \times M}$. The PC algorithm has three steps. First, we calculate the norm of predicted scores for user u by:

$$\mathbf{n}_u = \|(\widehat{\mathbf{R}}_{u,:} \odot (1 - \mathbf{R}_{u,:}))/(M - |O_u^+|)\|_{\mathrm{F}},$$

where we only consider the predicted preference scores to items that are not in the training data (by $\widehat{\mathbf{R}}_{u,:} \odot (1 - \mathbf{R}_{u,:})$) because the ranking is executed only on these un-interacted items and we should exclude the influence of items in the training set. Second, we calculate the popularity compensation score for one item i given u:

$$\mathbf{C}_{u,i} = \frac{1}{pop(i)} \cdot (\widehat{\mathbf{R}}_{u,i} \cdot \beta + 1 - \beta),$$

where there are two parts: $1/pop(i)$ is to achieve Guideline 1, and $(\widehat{\mathbf{R}}_{u,i} \cdot \beta + 1 - \beta)$ is to achieve Guideline 2 by using the predicted score as the indicator of user preference to i. $\beta \in [0, 1]$ is a trade-off weight to control the ratio of predicted preference score in the compensation: larger β means higher ratio for predicted scores. Last, following Guideline 3, we need to scale the compensation to match the user preference score scale and add it to $\widehat{\mathbf{R}}_{u,i}$:

$$\widehat{\mathbf{R}}^*_{u,i} = \widehat{\mathbf{R}}_{u,i} + \alpha \cdot \mathbf{C}_{u,i} \cdot \mathbf{n}_u / \mathbf{m}_u,$$

where $\widehat{\mathbf{R}}^*_{u,i}$ is the new preference score from u to i; $\mathbf{m}_u = \|(\mathbf{C}_u \odot (1 - \mathbf{R}_u))/(M - |O_u^+|)\|_{\mathrm{F}}$ is the norm of compensation scores of u excluding those for items in O_u^+; $\mathbf{n}_u/\mathbf{m}_u$ is to normalize the compensation scores based on Guideline 3; and α is the trade-off weight for the whole PC algorithm. With new preference scores for all candidate items, we can provide a debiased ranking list for u.

6.2 In-processing: Regularization

In this section, we introduce a regularization-based in-processing way to debias. The proposed method is inspired by previous work enhancing equal opportunity based recommendation fairness for different item groups [7], which try to decrease the correlation between item group variable and model output scores to achieve fairness. We adapt this idea to the context of alleviating the popularity-opportunity bias by decreasing the correlation between item popularity and model output scores.

We adopt the square of the *Pearson correlation coefficient* between predicted preference scores for positive user-item pairs and corresponding item popularity as a regularization term, and mitigate the bias by minimizing this regularization term together with the recommendation error:

$$\min_{\Theta} \mathcal{L}_{Rec} + \gamma PCC(\widehat{\mathbf{R}}_{+}, pop(\mathcal{I}))^2,$$

where $\mathcal{L}_{Rec}$ is the loss of recommendation models as shown in Section 3; $PCC(\widehat{\mathbf{R}}_{+}, pop(\mathcal{I}))$ computes Pearson correlation coefficient between predicted scores for positive user-item pairs and the popularity of corresponding items; and γ is the trade-off weight.

The proposed Reg is designed to decouple the item popularity with the model preference predictions to alleviate the popularity-opportunity bias. However, minimizing the correlation between item popularity and the predicted score is a challenging task because item popularity is continuous and unevenly distributed. Thus, a decrease in recommendation utility is expected when we aim to reduce the bias significantly by Reg, which will be further examined in Section 7. We leave the improvement for future work.

7 EXPERIMENTS

In this section, we investigate the impact of the proposed debiasing methods w.r.t. recommendation utility and debiasing performance, compared with biased base models and baselines of removing conventional popularity bias. Then, we illustrate these impacts over the same examples from Figure 1 to better understand their effects. Last, we study the impact of hyper-parameters on the two proposed debiasing algorithms.

7.1 Experiment Setup

Data and Baselines. We use the same four datasets introduced in Section 4. We compare the biased models MF and BPR with their debiased versions: **MF-PC** and **BPR-PC** denote the debiased versions based on the Popularity Compensation algorithm, while **MF-Reg** and **BPR-Reg** denote the debiased versions based on the regularization-based model. Besides, we also include two baselines which are designed to remove the conventional popularity bias for comparison, in other words, models forcing items of different popularity to receive similar rankings for all users.

The first baseline removes the conventional popularity bias by weighted matrix factorization [32], which assigns weights to training samples in the recommendation loss in Equation 1 and Equation 2 based on the popularity of involved items – items of low popularity will be assigned with high weights to promote the predicted scores for them. The weight for item i is chosen as $w_i \propto 1/pop(i)^e$, where e is an exponent to control the strength of the debiasing effect. We denote the corresponding versions with MF and BPR as base models as **MF-weight** and **BPR-weight**.

The second baseline removes the conventional popularity bias by rescaling the training data [33], which multiplies rescaling values to the binary training samples based on the popularity of involved items to uniformly promote the scores of low-popularity items. Then, it trains the vanilla MF or BPR models on the rescaled training data. The rescaling values are determined by the same way as the weights in the weighted model: $w_i \propto 1/pop(i)^e$ with the exponent e to control the debiasing strength. We denote the corresponding baselines as **MF-rescale** and **BPR-rescale**.

Table 3: Evaluation of recommendation utility (*NDCG@k*), uPO bias (*PRU*), and iPO bias (*PRI*) for MF based models on four datasets. * indicates the correlation coefficients are statistically significant for $p < 0.01$.

		NDCG@k		*PRU*	*PRI*
		@20	@50		
ML1M	MF	0.2726	0.2930	0.8350	0.9799*
	MF-weight	0.1484	0.1793	0.4845	0.6407*
	MF-rescale	0.1361	0.1658	0.4365	0.6936*
	MF-Reg	0.1492	0.1720	0.1910	0.5916*
	MF-PC	0.1435	0.1980	0.4552	0.5594*
Ciao	MF	0.0717	0.0934	0.5420	0.3625*
	MF-weight	0.0447	0.0675	0.3174	0.3293*
	MF-rescale	0.0425	0.0608	0.3219	0.2526*
	MF-Reg	0.0497	0.0639	0.2881	0.1905*
	MF-PC	0.0647	0.0845	0.3073	−0.0150
Epinions	MF	0.0693	0.0938	0.6840	0.5351*
	MF-weight	0.0349	0.0526	0.3453	0.2341*
	MF-rescale	0.0343	0.0509	0.3678	0.2182*
	MF-Reg	0.0386	0.0516	0.2175	0.2251*
	MF-PC	0.0605	0.0848	0.3549	−0.0415
App	MF	0.1026	0.1359	0.5667	0.6089*
	MF-weight	0.0388	0.0596	0.3552	0.2334*
	MF-rescale	0.0384	0.0583	0.3350	0.2147*
	MF-Reg	0.0439	0.0599	-0.0571	0.2207*
	MF-PC	0.0965	0.1280	0.3527	−0.0487

Because the two conventional popularity bias based baselines uniformly promote low-popularity items in recommendations, the popularity-opportunity bias is expected to be reduced as well. However, these baselines modify the recommendations without considering the potential user preferences as the two proposed debiasing models do. Hence, it is also expected that the two baselines will decrease the recommendation utility significantly.

Metrics. We evaluate user-side and item-side bias for all the models using the metrics introduced in Section 4.1, and compare the recommendation utility based on *NDCG@k* with $k = 20$ and 50.

Reproducibility. All models are implemented in Tensorflow [1] and optimized by Adam [22] algorithm. For all models and all datasets, we fix the latent dimension as 64, set the learning rate as 0.001, the negative sampling rate as 2, and set the mini-batch size as 1024. Then we tune hyper-parameters for all models by grid search over validation sets. More specifically, for post-processing methods MF-PC and BPR-PC, we directly apply the PC algorithm on the outputs from MF and BPR, and tune α in $[0.1, 1.5]$ with step 0.1, tune β in $[0.0, 1.0]$ with step 0.1. For in-processing models, we tune γ in $\{1e2, 1e3, 1e4, 1e5, 1e6, 1e7\}$. Note that for all the debiasing models, there is a trade-off between recommendation utility and debiasing performance. Hence, we explore hyper-parameters that minimize the bias metrics while preserving an acceptable utility.

7.2 Comparing Debiasing Performance

We begin in Table 3 with a comprehensive study on four datasets for all MF based models (including original biased model: MF; debiased baselines designed for conventional popularity bias: MF-weight and MF-rescale; and the proposed debiased ones designed for the popularity-opportunity bias: MF-Reg and MF-PC). Here, we walk through the key findings:

Table 4: Evaluation of recommendation utility, uPO bias (*PRU*), and iPO bias (*PRI*) for BPR based models on ML1M datasets. * indicates the correlation coefficients are statistically significant for $p < 0.01$.

		NDCG@k		*PRU*	*PRI*
		@20	@50		
ML1M	BPR	0.2983	0.3220	0.7793	0.9688*
	BPR-weight	0.1458	0.1757	0.5121	0.6249*
	BPR-rescale	0.1446	0.1784	0.4349	0.6064*
	BPR-Reg	0.1660	0.1769	0.2862	0.5633*
	BPR-PC	0.2308	0.2711	0.5712	0.5080*

First, we investigate the recommendation utility of the two proposed debiasing models and the two baselines compared with the original MF. Typically there is a trade-off between recommendation utility and debiasing effectiveness, and we observe such a trade-off here as well. Focusing on the *NDCG* columns for different values of k, we see that in all cases there is a drop in recommendation utility between original MF and its debiased versions (proposed MF-PC and MF-reg, and baselines for conventional popularity bias MF-weight and MF-rescale). Then, by comparing the four debiasing models, we observe that the MF-PC can preserve recommendation utility more effectively than the others, and MF-Reg performs similarly to the two baselines. Given these utility results, if we can observe lower bias by the proposed models, we can conclude that proposed models are able to achieve more effective debiasing performance with recommendation utility preserved.

Hence, we next study the impact different approaches have on reducing user-side (uPO) bias. Let's focus on the *PRU* column (which measures the popularity-rank correlation for users: high values correspond with high bias). We observe that all debiasing algorithms can significantly reduce *PRU* compared with the original MF. And these findings hold across all four datasets. Comparing the four debiasing models, in general, MF-Reg is able to improve *PRU* more significantly, and MF-PC performs similarly to MF-weight and MF-rescale. It may be because MF-Reg reduces the correlation between popularity and model predictions, which can effectively shuffle the rankings of matched items for each user. While the other three debiasing models are to re-rank items based on heuristics, which are expected to keep the original rankings to some degree. Another reason of less effective performance of MF-PC compared with MF-Reg is that MF-PC provide much better recommendation utility than MF-Reg, and a lower *PRU* is expected if we strengthen the debiasing effect for MF-PC.

Third, we investigate the impact different approaches have on reducing item-side (iPO) bias. Here, we focus on the *PRI* column (which measures the popularity-rank correlation for items: high values correspond with high bias). All four debiasing methods can improve *PRI* against original MF. Comparing the four debiasing methods, the PC algorithm is much more effective, which can reduce the *PRI* to a great extent. Although with a smaller improvement, the proposed Reg algorithm is more effective than the two baselines for removing conventional popularity bias.

Similar results can be observed from experiments on BPR and its debiasing variations (the results on ML1M dataset is shown in table 4). Based on these results, we can draw the conclusion that the proposed two debiasing algorithms can indeed mitigate both uPO and iPO bias, with the post-processing PC algorithm preserving recommendation utility better than the in-processing Reg approach. Comparing the two proposed methods with the two baselines, we can conclude that both proposed debiasing methods can alleviate the popularity-opportunity bias and preserve the recommendation utility more effectively than baseline methods designed for removing the conventional popularity bias.

	ItemID116	ItemID129	ItemID552	ItemID1955
UserID5003	Pop:1588	Pop:487	Pop:307	Pop:185
MF	3	106	262	557
MF-PC	19	75	141	314
MF-Reg	2195	48	64	297

Figure 3: Case study: ranking results for items that user 5003 in ML1M will like by different models.

		ItemID213	ItemID632	ItemID578	ItemID1219	ItemID3001
		Pop:1220	Pop:351	Pop:178	Pop:95	Pop:18
MF	avg_rank	31	233	468	673	1915
	prob@100	94%	34%	12%	0%	0%
MF-PC	avg_rank	127	358	464	289	693
	prob@100	78%	51%	33%	60%	20%
MF-Reg	avg_rank	1907	348	269	353	1322
	prob@100	3%	4%	42%	35%	0%

Figure 4: Case study: average ranking results of items for matched users in ML1M by different models

7.3 Case Study

To further understand the effects of the proposed models, we compare the recommendation results of the debiasing algorithms and the base model MF for the same examples shown in Figure 1 (results for BPR based models show similar pattern). First, Figure 3 shows the ranking results for matched items of user 5003 by different models (recall this is based on the ML1M dataset). By comparing the debiasing models with their original base model, we can see that both debiasing algorithms are able to promote the rankings for unpopular items. The PC algorithm promotes unpopular items and maintains relatively high rankings for popular ones, meaning that it is fairly effective at overcoming popularity-opportunity bias. But the Reg model cannot preserve high rankings for these popular items, giving insight into the challenges Reg faced in Table 3.

Next, we show the results from the perspective of iPO bias in Figure 4, where we compare the recommendation results for five items by different models. We can see that compared with MF, the debiasing models promote the less popular items to have better ranking results. For example, MF-PC decreases the recommendation probability (assuming 100 items are recommended for each user) for item213 from 94% to 78%, but increases the probabilities for items with lower popularity, especially for item1219 and item3001, which do not have any chance to be exposed to users who like them by MF, but have 60% and 20% probabilities by MF-PC. Similar in spirit to our previous observation, the Reg also increases rankings for unpopular items but cannot preserve rankings for popular items.

7.4 Impact of Hyper-parameters

Finally, we study the impact of the hyper-parameters. Due to the space limitation, we only show the conclusions based on the experiments here but do not show the detailed results.

For the PC algorithm, we have two hyper-parameters: α controls the ratio of popularity compensation, with larger values meaning more weight to the compensation; β controls the strength of predicted preference scores on the popularity compensation, with larger values meaning more weight for predicted preference scores. Based on our experimental results, we observe that as α increases, recommendation utility decreases and the debiasing performance is being improved. This result is because a larger α means a higher ratio of the popularity compensation in the final output, leading to worse recommendation utility but less bias. For β, we observe that the recommendation utility keeps increasing, and the debiasing effect is first improved and then degraded as β increases. The reason behind this is that reasonable β can indicate user preferences and help calculate accurate compensation scores, but higher β makes preference scores dominate the compensation lead to a decrease in the debiasing performance. For the Reg algorithm, as γ increases, the recommendation utility is reduced, and the debiasing performance first improves then decreases due to overfitting.

8 CONCLUSION AND FUTURE WORK

In this paper, we conduct a three-part study to investigate popularity-opportunity bias in matrix factorization based models: i) we empirically show the vulnerability of two matrix factorization models to the bias by a data-driven study on four datasets; ii) we theoretically show how these two models inherently produce the popularity-opportunity bias on both user and item sides; and iii) we explore the potential of in-processing and post-processing approaches to alleviate the bias. Experiments on four datasets validate the debiasing effectiveness of both proposed methods over debiasing baselines designed for conventional popularity bias. In the future, we are interested in exploring more effective debiasing algorithms and studying popularity-opportunity bias in other collaborative filtering algorithms like KNN, AutoEncoder, and graph neural networks.

ACKNOWLEDGMENTS

This work is in part supported by NSF grant IIS-1939716.

REFERENCES

[1] M Abadi, P Barham, J Chen, Z Chen, A Davis, J Dean, M Devin, S Ghemawat, G Irving, M Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*.
[2] Himan Abdollahpouri and Robin Burke. 2019. Reducing Popularity Bias in Recommendation Over Time. (2019).
[3] H Abdollahpouri, R Burke, and B Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the 11th RecSys*.
[4] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking. In *The 32nd International Flairs Conference*.
[5] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *RMSE Workshop at RecSys* (2019).
[6] Chris Anderson. 2006. *The long tail: Why the future of business is selling less of more.* Hachette Books.
[7] A Beutel, J Chen, T Doshi, H Qian, L Wei, Y Wu, L Heldt, Z Zhao, L Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th SIGKDD*.
[8] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *Workshop on Fairness, Accountability, and Transparency in Machine Learning* (2017).
[9] Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. 2006. From niches to riches: Anatomy of the long tail. *Sloan Management Review* (2006).
[10] 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*.
[11] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. 2019. How Serendipity Improves User Satisfaction with Recommendations? A Large-Scale User Evaluation. In *The World Wide Web Conference*.
[12] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th SIGKDD*.
[13] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*.
[14] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *ACM transactions on interactive intelligent systems (tiis)* (2016).
[15] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*.
[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th WWW*.
[17] 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* (2015).
[18] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on recommendation independence for a find-good-items task. (2017).
[19] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2013. Efficiency Improvement of Neutrality-Enhanced Recommendation. In *Decisions@RecSys workshop in conjunction with the 7th RecSys*.
[20] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2014. Correcting Popularity Bias by Enhancing Recommendation Neutrality. In *RecSys*.
[21] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2018. Recommendation independence. In *FAT**.
[22] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR* (2015).
[23] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* (2009).
[24] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th RecSys*.
[25] W Liu and R Burke. 2018. Personalizing fairness-aware re-ranking. (2018).
[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.
[27] Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of RecSys*.
[28] Flavien Prost, Hai Qian, Qiuwen Chen, Ed H Chi, Jilin Chen, and Alex Beutel. 2019. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *"ML with Guarantees" workshop at 33rd Conference on Neural Information Processing Systems* (2019).
[29] 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 12th WSDM*.
[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th conference on uncertainty in artificial intelligence*.
[31] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*.
[32] Harald Steck. 2011. Item popularity and recommendation accuracy. In *Proceedings of the 5th RecSys*.
[33] Harald Steck. 2019. Collaborative filtering via high-dimensional regression. (2019).
[34] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: discerning multi-faceted trust in a connected world. In *Proceedings of the 5th WSDM*.
[35] Jianling Wang and James Caverlee. 2020. Recommending Music Curators: A Neural Style-Aware Approach. In *European Conference on Information Retrieval*.
[36] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd SIGIR*.
[37] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* (1987).
[38] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *NeurIPS*.
[39] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*.
[40] Feipeng Zhao and Yuhong Guo. 2016. Improving Top-N Recommendation with Heterogeneous Loss. In *IJCAI*.
[41] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-aware tensor-based recommendation. In *Proceedings of the 27th CIKM*.
[42] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems. In *Proceedings of the 43rd SIGIR*.