Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems

Ziwei Zhu
Texas A&M University
zhuziwei@tamu.edu

Jianling Wang
Texas A&M University
jlwang@tamu.edu

James Caverlee
Texas A&M University
caverlee@tamu.edu

ABSTRACT

Recommendation algorithms typically build models based on user-item interactions (e.g., clicks, likes, or ratings) to provide a personalized ranked list of items. These interactions are often distributed unevenly over different groups of items due to varying user preferences. However, we show that recommendation algorithms can inherit or even amplify this imbalanced distribution, leading to item under-recommendation bias. Concretely, we formalize the concepts of ranking-based statistical parity and equal opportunity as two measures of item under-recommendation bias. Then, we empirically show that one of the most widely adopted algorithms – Bayesian Personalized Ranking – produces biased recommendations, which motivates our effort to propose the novel debiased personalized ranking model. The debiased model is able to improve the two proposed bias metrics while preserving recommendation performance. Experiments on three public datasets show strong bias reduction of the proposed model versus state-of-the-art alternatives.

CCS CONCEPTS

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

KEYWORDS

recommender systems; statistical parity; equal opportunity; recommendation bias

ACM Reference Format:
Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401177

1 INTRODUCTION

The social and ethical concerns raised by recommenders are increasingly attracting attention, including issues like filter bubbles [26], transparency [27], and accountability [30]. In particular, *item under-recommendation bias* – wherein one or more groups of items are systematically under-recommended – is one of the most common but harmful issues in a personalized ranking recommender. Item under-recommendation bias is common in scenarios where the training data used to learn a recommender has an imbalanced distribution of feedback for different item groups due to the inherent uneven preference distribution in the real world [31]. For example, ads for non-profit jobs may be clicked at a lower rate than high-paying jobs, so that a recommendation model trained over this skewed data will inherit or even amplify this imbalanced distribution. This can result in ads for non-profit jobs being under-recommended.

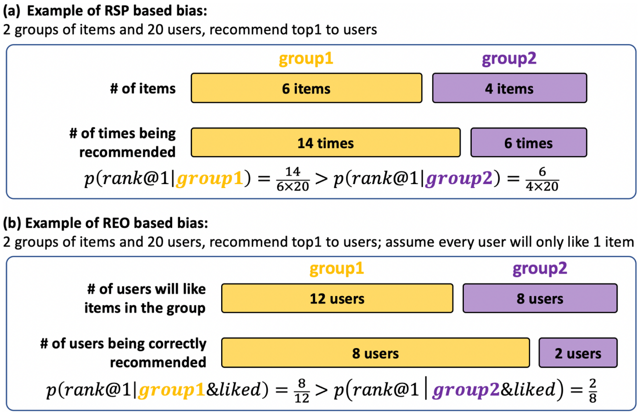

(a) Example of RSP based bias:
2 groups of items and 20 users, recommend top1 to users
group1
group2
# of items
6 items
4 items
# of times being recommended
14 times
6 times
$p(rank@1|group1) = \frac{14}{6\times20} > p(rank@1|group2) = \frac{6}{4\times20}$
(b) Example of REO based bias:
2 groups of items and 20 users, recommend top1 to users; assume every user will only like 1 item
group1
group2
# of users will like items in the group
12 users
8 users
# of users being correctly recommended
8 users
2 users
$p(rank@1|group1\&liked) = \frac{8}{12} > p(rank@1\,|\,group2\&liked) = \frac{2}{8}$

Figure 1: (a) is an example of the RSP-based bias measure. (b) is an example of the REO-based bias measure.

Previous works on item under-recommendation bias [16–19, 31, 33] mainly focus on investigating how to produce similar predicted score distributions for different groups of items (in other words, by removing the influence of group information when predicting preference scores). The main drawback of these works is that they mainly focus on the perspective of *predicted preference scores* [16–19, 31, 33]. In practice, however, predicted scores are an intermediate step towards a ranked list of items that serves as the final recommendation result, and having unbiased predicted scores does not necessarily lead to an unbiased ranking result. Thus, in this paper, we directly study item under-recommendation bias in the ranking results themselves. More specifically, we investigate two concrete scenarios of item under-recommendation bias and propose two metrics to measure the corresponding bias of each scenario: *ranking-based statistical parity* (**RSP**) and *ranking-based equal-opportunity* (**REO**).

Ranking-based statistical parity. RSP measures whether the probabilities for items in different groups to be recommended (that is, to be ranked in top k) are the same. Poor RSP means one or more groups have lower recommendation probabilities than others. That is, these groups are systematically under-recommended. Figure 1a provides an example of this RSP-based bias measure, where

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07...$15.00
https://doi.org/10.1145/3397271.3401177

there are two groups of items (group1 has 6 items and group2 has 4 items) being recommended to 20 users (where we recommend the top-1 item for each user). Within the 20 recommended items, 14 are from group1 and 6 are from group2, thus we can calculate that group1 items have higher recommendation probability (11.67%) than group2 items (only 7.5%). RSP is especially important when the item groups are determined by sensitive attributes (for example, gender or race when people are recommended, or political ideologies when political news are recommended) because systematic low recommendation probability for specific sensitive groups will result in social unfairness issues. For instance, if Figure 1a is to recommend male students (group1) and female students (group2) for college recruiting, then females are under-recommended and receive unfair treatment during the college recruiting process. Besides, from the view of decision making, RSP-based bias also means that the decision whether to recommend an item is partially determined by the sensitive attributes, i.e., whether recommending a student depends on the gender of the student: with everything else the same, a male student is more likely to be recommended, while a female student is less likely to be recommended.

Ranking-based equal opportunity. Another drawback of prior works [16–19, 31, 33] on item under-recommendation bias is that they view the bias as only depending on recommendation results without taking user preferences into account. For recommenders without sensitive attributes for items (like books or movies), we are less concerned with equal recommendations as in our previous RSP example, but demand that recommendations be driven by user preferences. However, general recommendation algorithms tend to overestimate popular groups and underestimate unpopular groups (a common issue of machine learning that also occurs in classification tasks [5, 32]). Thus these algorithms are not fully aligned with user preferences but rather assign lower recommendation probabilities for items in minority groups even if they are actually liked by users. Hence, we propose the second metric – REO – to measure the bias that items in one or more groups have lower recommendation probabilities given the items are liked by users. One example of REO-based bias is shown in Figure 1b, where we recommend two groups of items to 20 users (where we recommend the top-1 item for each user), and assume every user only has one liked item (12 users like group1 items, 8 users like group2 items). Ideally, the probability of being correctly recommended should be the same across groups, but from this example, we see that 8 users (out of 12) who like group1 items are correctly recommended, but only 2 users (out of 8) who like group2 items are correctly recommended. Compared to RSP, REO-based bias does not depend on sensitive attributes and hence is intrinsic to all recommender systems, potentially exerting damaging influence to both users and item providers. On the one hand, user needs corresponding to minority groups are not fully acknowledged, leading to lower user satisfaction. On the other hand, item providers of minority groups may receive less exposure than they should receive. For instance, if Figure 1b is to recommend crime movies (group1) and children's movies (group2), then children's movies are less likely to be correctly recommended, leading to an undesired imbalance. In this case, users who like children's movies cannot be satisfied by the recommendation; and children's movies receive less feedback, further exaggerating the imbalance and forming a vicious circle.

Contributions. With these two different scenarios in mind and corresponding bias metrics, we empirically demonstrate that a fundamental recommendation model – Bayesian Personalized Ranking (BPR) [28] – is vulnerable to this item under-recommendation bias, which motivates our efforts to address it. Then, we show how to overcome this bias based on the two introduced metrics through a debiased personalized ranking model (DPR) that has two key features: a multi-layer perceptron adversary that seeks to enhance the score distribution similarity among item groups and a KL Divergence based regularization term that aims to normalize the score distribution for each user. Incorporating these two components together, RSP (or REO depending on how we implement the adversary learning) based bias can be significantly reduced while preserving recommendation quality at the same time. Extensive experiments on three public datasets show the effectiveness of the proposed model over state-of-the-art alternatives. In general, DPR is able to reduce the two bias metrics for BPR by 67.3% on average (with an improvement of 48.7% over the best baseline), while only decreasing $F1@15$ versus BPR by 4.1% on average (with an improvement of 16.4% over the best baseline).

2 RELATED WORK

Recommendation Bias. Many efforts have focused on mitigating bias for recommendation tasks in the context of explicit rating prediction. More specifically, from the perspective of eliminating bias based on statistical parity, Kamishima et al. proposed regularization-based models [17, 18] to penalize bias in the predicted ratings; and Yao et al. [31] proposed three bias metrics on the user side in rating prediction tasks and generalized the regularization-based model to balance recommendation quality and the proposed metrics. Recent works have started to shift attention toward recommendation bias in personalized ranking tasks. However unlike this paper, most do not align bias with the ranking results. For example, although under the ranking recommendation setting, Kamishima et al. [16] adopted a regularization-based approach to eliminate bias for predicted preference scores; and Zhu et al. [33] proposed a parity-based model over predicted scores by first isolating sensitive features and then extracting the sensitive information. Research in [7], [8] and [4] proposed metrics that consider the ranking results. However, there are three main differences between them and the present work: i) neither of [7] nor [8] takes ground truth of user preferences into consideration; ii) [7] and [4] only consider two-group scenarios; and iii) [4] and [8] consider bias among item groups for individual users rather than consider system-level bias for different item groups. In sum, the main differences between this work and previous works are that the bias we investigate is over ranking results among multiple groups; is based on both statistical parity and equal opportunity; and is calculated from the system-level.

Popularity bias is another type of recommendation bias in which recommenders tend to recommend popular items more frequently than unpopular items. Existing works [2, 3] typically study popularity bias by grouping items based on their popularity (usually two groups: popular vs. unpopular), which has a similar problem setup as in this work. However, popularity bias does not consider specific meanings for item groups (such as gender, race or category).

Other Related Topics. There are some recent efforts investigating topics related to recommendation bias. For example, Beutel et al. [6] and Krishnan et al. [22] explored approaches to decrease the bias w.r.t. recommendation accuracy for niche items. Recommendation diversity [15], which requires as many groups as possible appearing in the recommendation list for each user, is related to the metric RSP in this work, but fundamentally different. Another similar concept to RSP called calibrated recommendations is proposed by Steck [29], which encourages the same group proportions as the historical record for each user and can be regarded as a special bias for individual users. The main differences of our work and these previous works are that research of recommendation diversity and recommendation calibration investigate the distribution skews for each individual user rather than for the whole system, and they only consider the recommendation distributions without taking into account the ground truth of user preference and item quality as in this work.

3 BIAS IN PERSONALIZED RANKING

In this section, we first describe the personalized ranking problem and ground our discussion through a treatment of Bayesian Personalized Ranking (BPR) [28]. Next, we introduce two proposed bias metrics for personalized ranking. Last, we empirically demonstrate that BPR is vulnerable to the item under-recommendation bias. By measuring the inherent bias in BPR, we aim to show that item under-recommendation bias is a common and critical issue, which motivates our efforts to address it.

3.1 Bayesian Personalized Ranking

Given N users $\mathcal{U} = \{1, 2, \ldots, N\}$ and M items $\mathcal{I} = \{1, 2, \ldots, M\}$, the personalized ranking problem is to recommend a list of k items to each user u based on the user's historical behaviors $\mathcal{I}_u^+ = \{i, j, \ldots\}$, where $i, j, \ldots$ are the items u interacts with before (and so can be regarded as implicit positive feedback). Bayesian Personalized Ranking (BPR) [28] is one of the most influential methods to solve this problem, which is the foundation of many cutting edge personalized ranking algorithms (e.g. [12, 13]). BPR adopts matrix factorization [21] as the base and minimizes a pairwise ranking loss, formalized as:

$$\min_{\Theta} \mathcal{L}_{BPR} = -\sum_{u \in \mathcal{U}} \sum_{\substack{i \in \mathcal{I}_u^+ \\ j \in \mathcal{I} \setminus \mathcal{I}_u^+}} ln\, \sigma(\widehat{y}_{u,i} - \widehat{y}_{u,j}) + \frac{\lambda_\Theta}{2} \|\Theta\|_F^2, \tag{1}$$

where $\widehat{y}_{u,i}$ and $\widehat{y}_{u,j}$ are the predicted preference scores calculated by the matrix factorization model for user u to positive item i and sampled negative item j; $\sigma(\cdot)$ is the Sigmoid function; $\|\cdot\|_F$ is the Frobenius norm; Θ represents the model parameters, i.e., $\Theta = \{\mathbf{P}, \mathbf{Q}\}$, where $\mathbf{P}$ and $\mathbf{Q}$ are the latent factor matrices for users and items; and λ_Θ is the trade-off weight for the l2 regularization.

With the trained BPR, we can predict the preference scores toward all un-interacted items and rank them in descending order for user u. A list of items with the top k largest scores $\{R_{u,1}, R_{u,2}, \ldots, R_{u,k}\}$ will be recommended to user u, where $R_{u,k}$ is the item id at the ranked k position.

3.2 Bias Metrics

However, there is no notion of debiasing in such a personalized ranking model. Here, we assume a set of A groups $\mathcal{G} = \{g_1, g_2, \ldots, g_A\}$, and every item in $\mathcal{I}$ belongs to one or more groups. A group here could correspond to gender, ethnicity, or other item attributes. We define a function $G_{g_a}(i)$ to identify whether item i belongs to group g_a. If it does, the function returns 1, otherwise 0. Next, we introduce two metrics for item under-recommendation bias for the personalized ranking problem.

Ranking-based Statistical Parity (RSP). Statistical parity requires the probability distributions of model outputs for different input groups to be the same. In a similar way, for the personalized ranking task, statistical parity can be defined as forcing the ranking probability distributions of different item groups to be the same. Because conventionally only the top-k items will be recommended to users, we focus on the probabilities of being ranked in top-k, which is also aligned with basic recommendation quality evaluation metrics such as *precision@k* and *recall@k*. As a result, we propose the ranking-based statistical parity metric – RSP, which encourages $P(R@k|g = g_1) = P(R@k|g = g_2) = \ldots = P(R@k|g = g_A)$, where $R@k$ represents 'being ranked in top-k', and $P(R@k|g = g_a)$ is the probability of items in group g_a being ranked in top-k. Formally, we calculate the probability as follows:

$$P(R@k|g = g_a) = \frac{\sum_{u=1}^{N} \sum_{i=1}^{k} G_{g_a}(R_{u,i})}{\sum_{u=1}^{N} \sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} G_{g_a}(i)},$$

where $\sum_{i=1}^{k} G_{g_a}(R_{u,i})$ calculates how many un-interacted items from group g_a are ranked in top-k for user u, and $\sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} G_{g_a}(i)$ calculates how many un-interacted items belong to group g_a for u. Last, we compute the *relative standard deviation* (to keep the same scale for different k) over the probabilities to determine *RSP@k*:

$$RSP@k = \frac{std(P(R@k|g = g_1), \ldots, P(R@k|g = g_A))}{mean(P(R@k|g = g_1), \ldots, P(R@k|g = g_A))},$$

where $std(\cdot)$ calculates the standard deviation, and $mean(\cdot)$ calculates the mean value.

Ranking-based Equal Opportunity (REO). Our second metric is based on the concept of equal opportunity [5, 9, 32], which encourages the true positive rates (TPR) of different groups to be the same. Take a binary classification task with two groups as an example, equal opportunity requires:

$$P(\widehat{c} = 1|g = 0, c = 1) = P(\widehat{c} = 1|g = 1, c = 1),$$

where c is the ground-truth label, $\widehat{c}$ is the predicted label; $P(\widehat{c} = 1|g = 0, c = 1)$ represents the TPR for group 0, $P(\widehat{c} = 1|g = 1, c = 1)$ is the TPR for group 1. Similarly, in the personalized ranking system, equal opportunity demands the ranking based TPR for different groups to be the same. We can define the TPR as the probability of being ranked in top-k given the ground-truth that the user likes the item, noted as $P(R@k|g = g_a, y = 1)$, where $y = 1$ represents items are liked by users. The probability can be calculated by:

$$P(R@k|g = g_a, y = 1) = \frac{\sum_{u=1}^{N} \sum_{i=1}^{k} G_{g_a}(R_{u,i}) Y(u, R_{u,i})}{\sum_{u=1}^{N} \sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} G_{g_a}(i) Y(u, i)},$$

where $Y(u, R_{u,i})$ identifies the ground-truth label of a user-item pair $(u, R_{u,i})$, if item $R_{u,i}$ is liked by user u, returns 1, otherwise 0 (in

	Group	#Item	#Feedback	$\frac{\#feedback}{\#item}$
ML1M	Sci-Fi	271	157,290	580.41
	Adventure	276	133,946	485.31
	Crime	193	79,528	412.06
	Romance	447	147,501	329.98
	Children's	248	72,184	291.06
	Horror	330	76,370	231.42
	Relative std	-	-	**0.33**
Yelp	American(New)	1610	91,519	56.84
	Japanese	946	45,508	48.11
	Italian	1055	46,434	44.01
	Chinese	984	36,729	37.33
	Relative std	-	-	**0.17**
Amazon	Grocery	749	49,646	66.28
	Office	892	37,776	42.35
	Pet	518	16,260	31.39
	Tool	606	14,771	24.37
	Relative std	-	-	**0.44**

Table 1: Group information in the three datasets.

practice, $Y(u, i)$ identifies whether a user-item pair (u, i) is in the test set); $\sum_{i=1}^{k} G_{g_a}(R_{u,i})Y(u, R_{u,i})$ counts how many items in test set from group g_a are ranked in top-k for user u, and $\sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} G_{g_a}(i)Y(u, i)$ counts the total number of items from group g_a in test set for user u. Similar to RSP, we calculate the relative standard deviation to determine *REO@k*:

$$REO@k = \frac{std(P(R@k|g = g_1, y = 1) \dots P(R@k|g = g_A, y = 1))}{mean(P(R@k|g = g_1, y = 1) \dots P(R@k|g = g_A, y = 1))}.$$

For classification tasks, TPR is the recall of classification, and for personalized ranking, the probability $P(R@k|g = g_a, y = 1)$ is *recall@k* of group g_a. In other words, mitigating REO-based bias requires *recall@k* for different groups to be similar.

Note that for both *RSP@k* and *REO@k*, **lower values indicate the recommendations are less biased**. In practice, RSP is particularly important in scenarios where people or items with sensitive information are recommended (such as political news). Because RSP-based bias in these scenarios leads to social issues like gender discrimination during recruiting or political ideology unfairness during election campaigns. Conversely, REO is supposed to be enhanced in general item recommendation systems so that no user need is ignored, and all items have the chance to be exposed to users who like them.

3.3 BPR is Vulnerable to Data Bias

In this section, we empirically show that BPR is vulnerable to imbalanced data and tends to produce biased recommendation based on metrics RSP and REO. Since there is no standard public dataset related to recommendation bias with sensitive attributes, we adopt three public real-world datasets that have been extensively used in previous works [11, 13, 31]. However, conclusions we draw should still hold if we analyze the bias on datasets with sensitive features because the fundamental problem definition and the mechanism leading to bias are exactly the same as the experiments in this paper.

MovieLens 1M (ML1M) [10] is a movie rating dataset, where we treat all ratings as positive feedback indicating users are interested in rated movies. We consider the recommendation bias for movie genres of 'Sci-Fi', 'Adventure', 'Crime', 'Romance', 'Childrens', and 'Horror', and remove other films, resulting in 6, 036 users, 1, 481 items, and 526, 490 interactions.

		$P(R@k\|g)$			$P(R@k\|g, y = 1)$		
	Genres	@5	@10	@15	@5	@10	@15
ML1M	Sci-Fi	.00654	.01306	.01949	.09497	.16819	.22922
	Adventure	.00516	.01022	.01521	.08884	.15808	.21657
	Crime	.00456	.00888	.01318	.07469	.13017	.17941
	Romance	.00327	.00665	.01002	.06448	.12003	.16366
	Children's	.00251	.00494	.00742	.05852	.10470	.14464
	Horror	.00176	.00354	.00533	.05399	.10132	.13985
	RSP or REO	**.41054**	**.40878**	**.40579**	**.20885**	**.19316**	**.18933**
Yelp	American(New)	.00154	.00302	.00449	.06345	.10904	.14497
	Japanese	.00111	.00219	.00328	.04770	.08207	.11106
	Italian	.00093	.00194	.00297	.03890	.07087	.09658
	Chinese	.00072	.00146	.00222	.03376	.05626	.07961
	RSP or REO	**.28005**	**.26376**	**.25224**	**.24515**	**.24290**	**.22253**
Amazon	Grocery	.00283	.00572	.00869	.03931	.07051	.09297
	Office	.00165	.00336	.00506	.01196	.02039	.03180
	Pet	.00185	.00348	.00501	.04815	.07807	.10215
	Tool	.00082	.00165	.00250	.00552	.01105	.01519
	RSP or REO	**.40008**	**.40672**	**.41549**	**.68285**	**.65756**	**.62175**

Table 2: Ranking probability distributions and RSP and REO metrics on three datasets by BPR.

Yelp (https://www.yelp.com/dataset/challenge) is a review dataset for businesses. We regard the reviews as the positive feedback showing user interests and only consider restaurant businesses. We investigate the recommendation bias among food genres of 'American(New)', 'Japanese', 'Italian', and 'Chinese', resulting in 8, 263 users, 4, 420 items, and 211, 721 interactions.

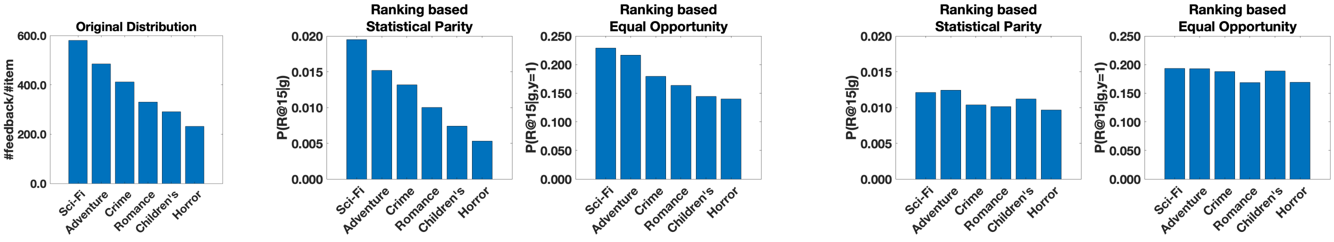
Amazon [25] contains product reviews on the Amazon e-commerce platform. We regard user purchase behaviors as the positive feedback, and consider recommendation bias among product categories of 'Grocery', 'Office', 'Pet', and 'Tool', resulting in 4, 011 users, 2, 765 items, and 118, 667 interactions.

Moreover, Table 1 lists the details of each group in the datasets, including the number of items, the number of feedback, and the ratio between them $\frac{\#feedback}{\#item}$. We use this ratio to identify the intrinsic data imbalance. The higher the ratio is, the more this group is favoured by users, and the relative standard deviation of ratios for all groups can indicate overall bias in the dataset. Hence, the Amazon and ML1M datasets contain relatively high bias; and Yelp has lower bias, but American(New) restaurants still have $\frac{\#feedback}{\#item}$ around 1.5 times higher than that of Chinese restaurants.

We run BPR on these datasets and analyze the ranking probability distributions. The detailed model hyper-parameter settings and data splitting are described in Section 5.2. Table 2 presents $P(R@k|g)$ and $P(R@k|g, y = 1)$ for different groups on three datasets by BPR, where we consider $k = 5, 10$, and 15. We also list the metrics *RSP@k* and *REO@k*. From the table, we have three major observations:

(i) For all datasets, the ranking probabilities are very different among groups, e.g., in ML1M, $P(R@5|g = \text{Sci-Fi})$ is four times higher than $P(R@5|g = \text{Horror})$, and $P(R@5|g = \text{Sci-Fi}, y = 1)$ is two times higher than $P(R@5|g = \text{Horror}, y = 1)$. And the high values of *RSP@k* and *REO@k* for all k and datasets demonstrate the biased recommendations by BPR.

(ii) The distributions of $P(R@k|g)$ and $P(R@k|g, y = 1)$ for all datasets basically follow the distributions of $\frac{\#feedback}{\#item}$ shown in



(a) Original. (b) BPR results. (c) DPR results.

Figure 2: The original distribution of #feedback/#item over different groups of ML1M data, and the ranking top15 probability distributions (both statistical parity and equal opportunity based) produced by BPR and proposed DPR.

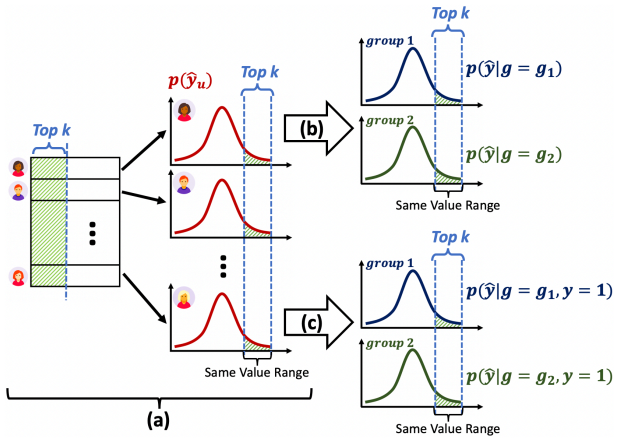


Figure 3: Illustration of the intuition of the proposed DPR.

Table 1, and sometimes the deviations of the ranking probability distributions are even larger than $\frac{\#feedback}{\#item}$ distributions, for example, the relative standard deviation of $P(R@15|g)$ in ML1M is 0.4058 while that of $\frac{\#feedback}{\#item}$ is 0.3344, which indicates that BPR preserves or even amplifies the inherent data bias.

(iii) As k decreases, the values of $RSP@k$ and $REO@k$ increase. In other words, the results are more biased for items ranked at top positions. This phenomenon is harmful for recommenders since attention received by items increases rapidly with rankings getting higher [23], and top-ranked items get most of attention from users.

Moreover, we also plot the original $\frac{\#feedback}{\#item}$ distribution of ML1M in Figure 2a and the ranking probability distributions by BPR in Figure 2b, which visually confirms our conclusion that BPR inherits data bias and produces biased recommendations. This conclusion motivates the design of a debiased personalized ranking framework as the models proposed in this paper. Figure 2c shows the ranking probability distributions generated by the proposed Debiased Personalized Ranking models, illustrating more evenly distributed and unbiased recommendations compared to BPR.

4 DEBIASED PERSONALIZED RANKING

Previous works on debiased recommendation [16, 19, 33] mainly focus on forcing different groups to have similar score distributions, which cannot necessarily give rise to unbiased rankings. One key reason is that users have different predicted score distributions, which means a high score from one user to an item does not necessarily result in a high ranking, and a low score does not lead to a low ranking. Conversely, if every user has an identical score distribution, the value ranges of the scores in top-k for all users will be the same, as demonstrated in Figure 3a. Then, the top-k scores in different item-group score distributions (noted as $p(\widehat{y}|g)$) are also in the same value range. Last, as illustrated in Figure 3b, if we enforce identical score distribution for different item groups, the proportions of top-k scores in the whole distribution for different groups will be the same, i.e., we have the same probability $p(R@k|g)$ for different groups (the definition of RSP). Similarly, if the positive user-item pairs in different groups have the same score distribution (noted as $p(\widehat{y}|g, y = 1)$), we will have the same probability $p(R@k|g, y = 1)$ for different groups (the definition of REO), as presented in Figure 3c. Based on this intuition, the proposed DPR first enhances the score distribution similarity between different groups by adversarial learning, then normalizes user score distributions to the standard normal distribution by a Kullback-Leibler Divergence (KL) loss. We introduce the two components of DPR and the model training process in the following subsections.

4.1 Enhancing Score Distribution Similarity

Adversarial learning has been widely applied in supervised learning [5, 24, 32] to mitigate model bias, with theoretical guarantees and state-of-the-art empirical performance. Inspired by these works, we propose to leverage adversarial learning to enhance the score distribution similarity between different groups. We first take the metric RSP as the example to elaborate the proposed method, and then generalize it to REO. Last, we show the advantages of the proposed adversarial learning over previous methods.

Adversary for RSP. The intuition of adversarial learning in this case is to play a minimax game between the BPR model and a discriminator. The discriminator is to classify the groups of the items based on the predicted user-item scores by BPR. As a result, BPR does not only need to minimize the recommendation error, but also needs to prevent the discriminator from correctly classifying the groups. If the discriminator cannot accurately recognize the groups given the outputs (predicted scores) from BPR, then the predicted score distributions will be identical for different groups. More specifically, in the adversarial learning framework, each training user-item pair (u, i) is first input to a conventional BPR model; then the output of BPR, $\widehat{y}_{u,i}$ is fed into a multi-layer perceptron

(MLP) to classify the groups $\widehat{\mathbf{g}}_i$ of the given item i. $\widehat{\mathbf{g}}_i \in [0,1]^A$ is the output of the last layer of MLP activated by the *sigmoid* function, representing the probability of i belonging to each group, e.g., $\widehat{\mathbf{g}}_{i,a}$ means the predicted probability of i belonging to group g_a. The MLP is the adversary, which is trained by maximizing the likelihood $\mathcal{L}_{Adv}(\mathbf{g}_i, \widehat{\mathbf{g}}_i)$, and BPR is trained by minimizing the ranking loss shown in Equation 1 as well as minimizing the adversary objective $\mathcal{L}_{Adv}(\mathbf{g}_i, \widehat{\mathbf{g}}_i)$. $\mathbf{g}_i \in \{0,1\}^A$ is the ground-truth groups of item i, if i is in group g_a, $\mathbf{g}_{i,a} = 1$, otherwise 0. We adopt the log-likelihood as the objective function for the adversary:

$$\max_{\Psi} \mathcal{L}_{Adv}(i) = \sum_{a=1}^{A} (\mathbf{g}_{i,a} log\, \widehat{\mathbf{g}}_{i,a} + (1 - \mathbf{g}_{i,a}) log\,(1 - \widehat{\mathbf{g}}_{i,a})),$$

where we denote $\mathcal{L}_{Adv}(\mathbf{g}_i, \widehat{\mathbf{g}}_i)$ as $\mathcal{L}_{Adv}(i)$ for short, and Ψ is the parameters of the MLP adversary. Combined with the BPR model, the objective function can be formulated as:

$$\min_{\Theta} \max_{\Psi} \sum_{u \in \mathcal{U}} \sum_{\substack{i \in \mathcal{I}_u^+ \\ j \in \mathcal{I} \setminus \mathcal{I}_u^+}} \mathcal{L}_{BPR}(u,i,j) + \alpha(\mathcal{L}_{Adv}(i) + \mathcal{L}_{Adv}(j)), \tag{2}$$

$$\text{where } \mathcal{L}_{BPR}(u,i,j) = -ln\, \sigma(\widehat{y}_{u,i} - \widehat{y}_{u,j}) + \frac{\lambda_\Theta}{2} \|\Theta\|_{\mathrm{F}}^2,$$

and α is the trade-off parameter to control the strength of the adversarial component.

Adversary for REO. As for REO, we demand the score distributions of positive user-item pairs rather than all the user-item pairs to be identical for different groups. Therefore, instead of feeding both scores for positive and sampled negative user-item pairs $\widehat{y}_{u,i}$ and $\widehat{y}_{u,j}$, we only need to feed $\widehat{y}_{u,i}$ into the adversary as:

$$\min_{\Theta} \max_{\Psi} \sum_{u \in \mathcal{U}} \sum_{\substack{i \in \mathcal{I}_u^+ \\ j \in \mathcal{I} \setminus \mathcal{I}_u^+}} \mathcal{L}_{BPR}(u,i,j) + \alpha \mathcal{L}_{Adv}(i). \tag{3}$$

Advantages of adversarial learning. There are two existing approaches to achieve a similar effect: a regularization-based method [16, 17, 31]; and a latent factor manipulation method [33]. The advantages of the proposed adversarial learning over previous works can be summarized as: (i) it can provide more effective empirical performance than other methods, which will be further demonstrated in Section 5.4; (ii) it is flexible to swap in different bias metrics (beyond just RSP and REO); (iii) it can handle multi-group circumstances; and (iv) it is not coupled with any specific recommendation models and can be easily adapted to methods other than BPR (such as more advanced neural networks).

4.2 Individual User Score Normalization

After the enforcement of group distribution similarity, the next step towards debiasing personalized ranking is to normalize the score distribution for each user. We can assume the score distribution of every user follows the normal distribution because based on the original BPR paper [28], every factor in the user or item latent factor vector follows a normal distribution. Then $\mathbf{P}_u^\top \mathbf{Q}_i$ (for a given user u, $\mathbf{P}_u$ is a constant and $\mathbf{Q}_i$ is a vector of normal random variables) follows a normal distribution as well. Thus we can normalize the score distribution of each user to the standard normal distribution by minimizing the KL Divergence between the score distribution of each user and a standard normal distribution as the KL-loss:

$$\mathcal{L}_{KL} = \sum_{u \in \mathcal{U}} D_{\mathrm{KL}}(q_\Theta(u) || \mathcal{N}(0,1)),$$

where $q_\Theta(u)$ is the empirical distribution of predicted scores for user u, and $D_{\mathrm{KL}}(\cdot||\cdot)$ computes KL Divergence between two distributions.

	#Users	#Items	#Ratings	Density
ML1M-2	5,562	543	215,549	7.14%
Yelp-2	6,310	2,834	117,978	0.66%
Amazon-2	3,845	2,487	84,656	0.89%

Table 3: Characteristics of the three 2-group datasets.

4.3 Model Training

Combining the KL-loss with Equation 2 leads to the complete DPR model to optimize RSP, noted as DPR-RSP:

$$\min_{\Theta} \max_{\Psi} \sum_{u \in \mathcal{U}} \sum_{\substack{i \in \mathcal{I}_u^+ \\ j \in \mathcal{I} \setminus \mathcal{I}_u^+}} (\mathcal{L}_{BPR}(u,i,j) + \alpha(\mathcal{L}_{Adv}(i) + \mathcal{L}_{Adv}(j))) + \beta \mathcal{L}_{KL},$$

where β is the trade-off parameter to control the strength of KL-loss. Similarly, we can optimize REO by combining KL-loss with Equation 3 to arrive at a DPR-REO model as well. Note that although the proposed DPR is built with BPR as the model foundation, it is in fact flexible enough to be adapted to other recommendation algorithms, such as more advanced neural networks [14].

Then, we train the model in a mini-batch manner. Generally, during model training, there are two phases in each epoch: first we update weights in the MLP adversary to maximize the classification objective, then update BPR to minimize the pairwise ranking loss, classification objective and KL-loss all together. Concretely, following the adversarial training process proposed in [24], in each epoch, we first update the MLP adversary by the whole dataset (in a stochastic way), then update BPR by one mini-batch, which empirically leads to fast convergence. And in practice, we usually first pre-train the BPR model for several epochs and then add in the adversarial training part.

5 EXPERIMENTS

In this section, we empirically evaluate the proposed model w.r.t. the two proposed bias metrics as well as the recommendation quality. We aim to answer three key research questions: **RQ1** What are the effects of the proposed KL-loss, adversary, and the complete model DPR on recommendations? **RQ2** How does the proposed DPR perform compared with other state-of-the-art debiased models from the perspectives of mitigating item under-recommendation bias and recommendation quality preserving? and **RQ3** How do hyper-parameters affect the DPR framework?

5.1 Datasets

The three datasets used in the experiments have been introduced in Section 3.3. Since the state-of-the-art baselines can only work for binary group cases, to answer **RQ2**, we create subsets keeping the most popular and least popular groups in the original datasets:

ML1M-2 ('Sci-Fi' vs 'Horror'), **Yelp-2** ('American(New)' vs. 'Chinese'), and **Amazon-2** ('Grocery' vs. 'Tool'). The specifics of the 2-group datasets are presented in Table 3. All datasets are randomly split into 60%, 20%, 20% for training, validation, and test sets. Note that there is no standard public dataset with sensitive features, thus we use public datasets for general recommendation scenarios to evaluate the performance of mitigating RSP-based bias. However, conclusions we draw should still hold if we analyze the debiasing performance on datasets with sensitive features because the fundamental problem definition and the mechanism leading to bias are exactly the same as the experiments in this paper.

5.2 Experimental Setup

Metrics. In the experiments, we need to consider both recommendation quality and recommendation bias. For the recommendation bias, we report *RSP@k* and *REO@k* as described in Section 3.2. As for the recommendation quality we adopt *F1@k*. We report the results with $k = 5, 10$, and 15. Note that we also measure NDCG in the experiments, which shows the same pattern as F1, hence we only report *F1@k* for conciseness.

Baselines. We compare the proposed DPR with biased method BPR shown in Section 3.1 and two state-of-the-art debiased recommendation methods:

FATR [33]. This is a tensor-based method, which enhances the score distribution similarity for different groups by manipulating the latent factor matrices. We adopt the 2D matrix version of this approach. Note that FATR is designed for statistical parity based metric, hence we do not have high expectation for the performance w.r.t. equal opportunity.

Reg [16, 17, 31]. The most commonly used debiasing method for two-group scenarios, which penalizes recommendation difference by minimizing a regularization term. Following [16], we adopt the squared difference between the average scores of two groups for all items as the regularization to improve RSP, denoted as **Reg-RSP**. For REO, we adopt the squared difference between the average scores of positive user-item pairs as the regularization, denoted as **Reg-REO** (it is similar to DPR-REO but enhances the distribution similarity by static regularization rather than adversary).

To have a fair comparison, we modify the loss functions of all baselines to the BPR loss in Equation 1. Moreover, to align the baselines with the bias metrics for ranking, we further add the proposed KL-loss introduced in Section 4.2 to both baselines.

Reproducibility. Code and data for this work can be found at https://github.com/Zziwei/Item-Underrecommendation-Bias. We implement the proposed model using Tensorflow [1] and adopt Adam [20] optimization algorithm. We tune the hyper-parameters of the models involved by the validation set, the basic rules are: (i) we search the hidden dimension over $\{10, 20, 30, 40, 50, 60, 70, 80\}$; (ii) search the L_2 regularizer λ_Θ over $\{0.01, 0.05, 0.1, 0.5, 1.0\}$; (iii) search the adversary regularizer α over range $[500, 10000]$ with step 500; (iv) search the KL-loss regularizer β over range $[10, 70]$ with step 10; and (v) search the model specific weight in FATR over $\{0.01, 0.05, 0.1, 0.5, 1.0\}$, and model specific weight for Reg-RSP and Reg-REO over the range $[1000, 10000]$ with step 2000. Note that selections of α and β should consider the balance between recommendation quality and recommendation bias.

	ML1M	Yelp	Amazon
BPR	0.1540	0.0808	0.0836
BPR w/ KL-loss	0.0571	0.0254	0.0313
Δ	**-62.92%**	**-68.56%**	**-62.56%**

Table 4: Comparison between BPR w/o KL-loss for JS Divergences among user score distributions over three datasets.

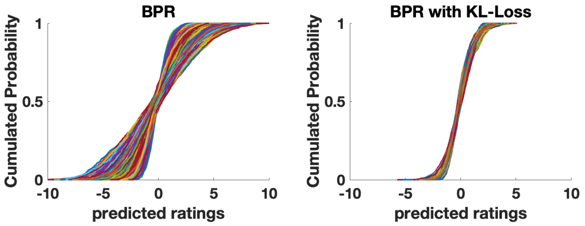


Figure 4: CDFs of user score distributions predicted by BPR and BPR with KL-loss over ML1M dataset.

There are two sets of experiments: experiments over multi-group datasets (ML1M, Yelp, and Amazon) to answer **RQ1** and **RQ3**; and experiments over binary-group datasets (ML1M-2, Yelp-2, and Amazon-2) to answer **RQ2**.

In the first set of experiments, for all three datasets: we set 20 as the hidden dimensions for BPR, DPR-RSP, and DPR-REO; we set the learning rate 0.01 for BPR, and η_{BPR} 0.01 as well for DPR-RSP and DPR-REO. For all methods, we set $\lambda_\Theta = 0.1$ for ML1M and Amazon; set $\lambda_\Theta = 0.05$ for Yelp. As for adversary learning rate η_{Adv}, we set 0.005 for ML1M and Yelp, 0.001 for Amazon. For all three datasets, we set $\alpha = 5000$ for DPR-RSP. As for DPR-REO, we set $\alpha = 1000$ for ML1M, 5000 for Yelp, and 10000 for Amazon.

In the second set of experiments, we set different hidden dimensions for different datasets, but for the same dataset all methods have the same dimension: we set 10 for ML1M-2, 40 for Yelp-2, and 60 for Amazon-2. We set the learning rate 0.01 for baselines, and 0.01 as η_{BPR} for DPR-RSP and DPR-REO. As for adversary learning rate η_{Adv}, we set 0.005 for all three datasets.

For all methods in all experiments, we have negative sampling rate 5 and mini-batch size 1024. For all debiased methods, we set $\beta = 30$. And we adopt a 4-layer MLP with 50 neurons with ReLU activation function in each layer as the adversary for DPR.

5.3 RQ1: Effects of Model Components

In this subsection, we aim to answer three questions: whether the KL-loss can effectively normalize user score distribution? whether the adversary can effectively enhance score distribution similarity among groups? and whether DPR-RSP and DPR-REO can effectively improve the bias metrics RSP and REO?

Effects of KL-loss. The KL-loss is to normalize the user score distribution. Hence, we adopt the Jensen-Shannon Divergence (JS Divergence) to measure the deviation between user score distributions, where lower JS Divergence indicates that the user score distributions are normalized better. We compare BPR and BPR with KL-loss over all three datasets, the results are shown in Table 4, and the improvement rates (noted as Δ) are also calculated. We can observe that with the KL-loss, the divergence among user score distributions is largely reduced, demonstrating the effectiveness of KL-loss. To better show the effects of KL-loss, we visualize the score distribution for every user produced by BPR with and without

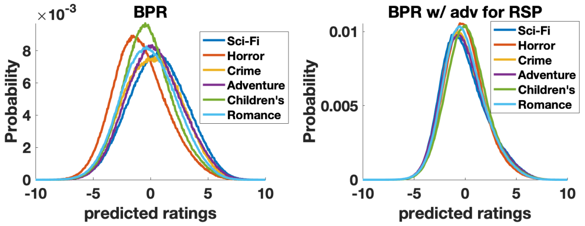


Figure 5: PDFs of $p(\widehat{y}|g)$ for different groups by BPR and BPR w/ adv for RSP over ML1M dataset.

		ML1M	Yelp	Amazon
RSP setting	BPR	0.0222	0.0011	0.0215
	BPR w/ adv	0.0090	0.0004	0.0046
	Δ	**-59.46%**	**-63.64%**	**-78.60%**
REO setting	BPR	0.0128	0.0045	0.0378
	BPR w/ adv	0.0047	0.0041	0.0087
	Δ	**-63.28%**	**-8.89%**	**-76.98%**

Table 5: Comparison between BPR and BPR w/ adv for JS Divergences of score distribution among different groups.

		ML1M	Yelp	Amazon
F1@15	BPR	0.1520	0.0371	0.0230
	DRP-RSP	0.1439	0.0354	0.0221
	Δ	**-5.31%**	**-4.32%**	**-3.90%**
RSP@15	BPR	0.4058	0.2522	0.4155
	DPR-RSP	0.0936	0.0856	0.0607
	Δ	**-76.92%**	**-66.07%**	**-85.40%**

Table 6: Comparison between BPR and DPR-RSP w.r.t. $F1@15$ and $RSP@15$ over three datasets.

KL-loss for ML1M in Figure 4, where each curve represents the Cumulative Distribution Function (CDF) of a single user's scores. The closely centralized CDFs in the right figure verify the effectiveness of the proposed KL-loss.

Effects of Adversary. The adversary in DPR is to enhance the score distribution similarity among different groups. To evaluate the effectiveness of the adversarial learning, we compare the performances of BPR and BPR with adversary for both metrics (noted as *BPR w/ adv for RSP* and *BPR w/ adv for REO*). More specifically, we compare BPR with BPR w/ adv for RSP w.r.t. JS Divergence among $p(\widehat{y}|g)$ for different groups, and compare BPR with BPR w/ adv for REO w.r.t. JS Divergence among $p(\widehat{y}|g, y = 1)$ for different groups. Results are shown in Table 5, where the top three rows are calculated on all user-item pairs not in the training set (fit the RSP setting), the bottom three rows are calculated on user-item pairs only in the test set (fit the REO setting). The table demonstrates the extraordinary effectiveness of the proposed adversarial learning for enhancing distribution similarity under both settings. To further validate this conclusion, we visualize the distributions of $p(\widehat{y}|g)$ for different groups from ML1M in Figure 5 (distributions of $p(\widehat{y}|g, y = 1$ have the same pattern), where the Probability Distribution Function (PDF) of every group's score distribution is plot as a single curve. We can find that PDFs by BPR w/ adv are close to each other, while PDFs by the ordinary BPR differ considerably.

Effects of DPR. The effects of the complete DPR should be evaluated from the perspectives of both recommendation quality and recommendation bias. We first investigate the performance of DPR-RSP. $F1@15$ and $RSP@15$ results of both BPR and DPR-RSP over three datasets are listed in Table 6, where the change rates for them are calculated. From the table we have three observations: (i) DPR-RSP improves the bias metric RSP over BPR greatly (decreases $RSP@15$ by 76% on average); (ii) DPR-RSP effectively preserves the recommendation quality (only drops $F1@15$ by 4% on average); and (iii) for different datasets with different degrees of bias, DPR-RSP can reduce the bias to a similar level ($RSP@15$ for three datasets by DPR-RSP are all smaller than 0.1).

		ML1M	Yelp	Amazon
F1@15	BPR	0.1520	0.0371	0.0230
	DRP-REO	0.1527	0.0363	0.0208
	Δ	**+0.49%**	**-1.94%**	**-9.81%**
REO@15	BPR	0.1893	0.2225	0.6217
	DPR-REO	0.0523	0.0874	0.3577
	Δ	**-72.38%**	**-60.73%**	**-42.47%**

Table 7: Comparison between BPR and DPR-REO w.r.t. $F1@15$ and $REO@15$ over three datasets.

		ML1M-2	Yelp-2	Amazon-2
RSP setting	BPR	0.0564	0.0034	0.0514
	FATR	**0.0218**	0.0027	**0.0332**
	Reg-RSP	0.0276	**0.0026**	0.0378
	DPR-RSP	**0.0155**	**0.0020**	**0.0079**
	Δ	-28.90%	-23.08%	-76.20%
REO setting	BPR	0.0422	0.0216	0.1531
	FATR	**0.0044**	0.0078	0.1844
	Reg-REO	0.0179	**0.0062**	**0.0219**
	DPR-REO	**0.0011**	**0.0018**	**0.0038**
	Δ	-75.00%	-70.97%	-82.65%

Table 8: Comparison between DPR and baselines for JS Divergences of score distribution among groups.

Similar conclusions can be drawn for DPR-REO based on Table 7, where comparison between BPR and DPR-REO w.r.t. $F1@15$ and $REO@15$ are listed. We can observe that DPR-REO is able to decrease metric $REO@15$ to a great extent while preserving high $F1@15$ as well. Generally speaking, DPR-REO demands less recommendation quality sacrifice because the definition of REO is less stringent and debiasing is easier to achieve than RSP. However, there is one exception that in Amazon dataset, DPR-REO drops $F1@15$ by 9.8%. It may be because for the Amazon dataset, every item group has its own collection of users, and there are few users giving feedback to more than one group, which exerts difficulty for DPR-REO training.

5.4 RQ2: Comparison with Baselines

We next compare the proposed DPR with state-of-the-art alternatives to answer two questions: (i) how does the proposed adversarial learning perform in comparison with baselines for predicted score distribution similarity enhancement? and (ii) how does the proposed DPR perform for both bias metrics compared with baselines? Because baselines Reg-RSP and Reg-REO can only work for binary-group cases, we conduct the experiment over ML1M-2, Yelp-2, and Amazon-2 datasets in this subsection.

To answer the first question, we report JS Divergences of score distributions for different groups in Table 8, where the top five rows are calculated on all user-item pairs not in the training set

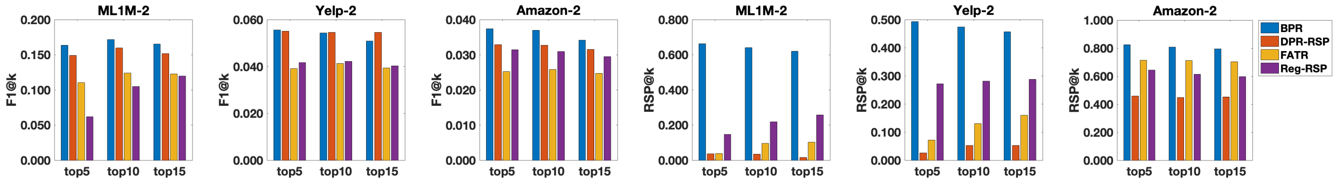


Figure 6: F1@k and RSP@k of four different models over three datasets.

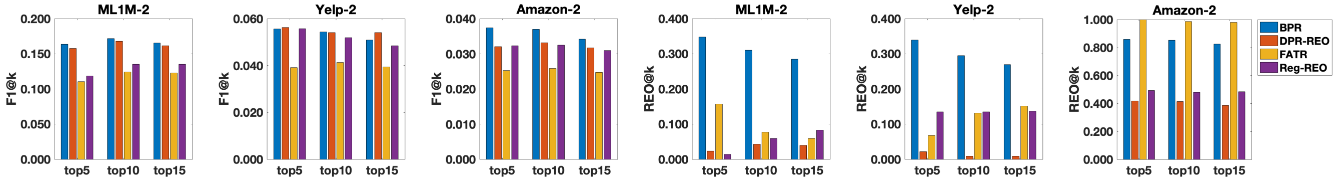


Figure 7: F1@k and REO@k of four different models over three datasets.

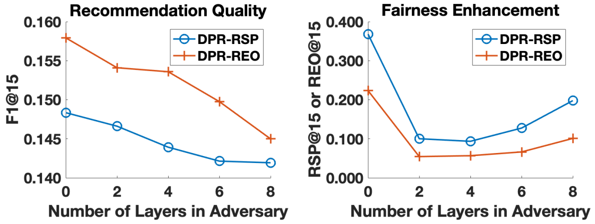


Figure 8: *F*1@15, *RSP*@15**, and** *REO*@15 **of DPR-RSP and DPR-REO w.r.t. different numbers of layers over ML1M.**

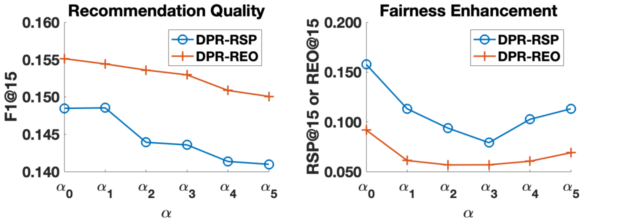


Figure 9: *F*1@15, *RSP*@15 **and** *REO*@15 **of DPR-RSP and DPR-REO w.r.t. different α over ML1M.**

(fitting the RSP setting), and the bottom five rows are calculated on user-item pairs only in the test set (fitting the REO setting). The improvement rates of DPR over the best baselines also are calculated. From the table we can conclude that the proposed adversarial learning can more effectively enhance score distribution similarity than baselines. Although less competitive, both FATR and Reg models can improve the distribution similarity to some degree compared with BPR.

As for the second question, we show *F*1@*k*, *RSP*@*k*, and *REO*@*k* comparison between all methods over all datasets in Figure 6 and Figure 7. On the one hand, from the leftmost three figures in both Figure 6 and Figure 7, we can observe that DPR-RSP and DPR-REO preserve relatively high *F*1@*k* from BPR and outperform other baselines significantly. On the other hand, from the rightmost three figures, we are able to see that DPR-RSP and DPR-REO enhance RSP and REO to a great extent respectively, which also outperform other debiased methods considerably. Besides, one potential reason for better recommendation quality for DPR on Yelp is that the intrinsic bias in Yelp is small, thus DPR can promote unpopular groups and keep the original high rankings for popular groups simultaneously, leading to better recommendation performance.

5.5 RQ3: Impact of Hyper-Parameters

Finally, we investigate the impact of three hyper-parameters: (i) the number of layers in the MLP adversary; (ii) the adversary trade-off regularizer α; and (iii) the KL-loss trade-off regularizer β. For conciseness, we only report experimental results on ML1M dataset, but note that the results on other datasets show similar patterns.

Impact of Layers in Adversary. First, we experiment with the number of layers in MLP adversary varying in $\{0, 2, 4, 6, 8\}$, and the other parameters are the same as introduced in Section 5.2 including that the number of neurons in each MLP layer is still 50. Generally speaking, with more layers, the adversary is more complex and expressive, which intuitively results in better bias reduction performance. The *F*1@15 results of DPR-RSP and DPR-REO w.r.t. different numbers of layers are shown at the left in Figure 8, and *RSP*@15 and *REO*@15 results are presented at the right in Figure 8. From these figures, we can infer that with a more powerful adversary, the recommendation quality drops more; however, the bias reduction effect first gets promoted but then weakened due to difficulty of model training. The best value is around 2 to 4. Besides, we can also find that it is easier to augment the metric REO than RSP with less recommendation quality sacrificed, which is consistent with the observation in Section 5.3.

Impact of α. Then, we vary the adversary trade-off regularizer α and plot the results in Figure 9, where the x-axis coordinates $\{\alpha_0, \alpha_1, \ldots, \alpha_5\}$ are $\{1000, 3000, 5000, 7000, 9000, 11000\}$ for DPR-RSP and $\{200, 600, 1000, 1400, 1800, 2200\}$ for DPR-REO. The left figure demonstrates the *F*1@15 results with different α, which shows that with larger weight for the adversary, the recommendation quality decreases more. For the bias reduction performance, as presented at the right in Figure 9, with larger α, both DPR-RSP and DPR-REO first decrease the bias, but then increase it again, which is most likely due to the dominating of adversary over KL-loss in the objective function. To balance the recommendation quality and recommendation bias, setting $\alpha = 5000$ for DPR-RSP and $\alpha = 1000$ for DPR-REO are reasonable choices.

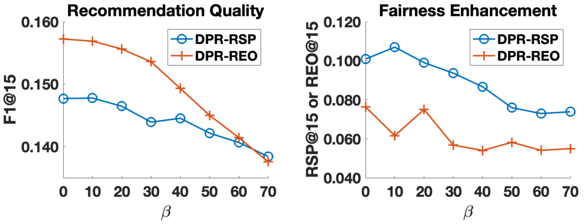


Figure 10: $F1@15$, $RSP@15$ and $REO@15$ of DPR-RSP and DPR-REO w.r.t. different β over ML1M.

Impact of β. Last, we study the impact of the KL-loss trade-off regularizer β and vary the value in the set $\{0, 10, 20, 30, 40, 50, 60, 70\}$. The left figure in Figure 10 shows the change tendency of $F1@15$, which implies that larger β leads to lower recommendation quality. The bias mitigation performance of DPR-RSP and DPR-REO with different β are shown at the right in Figure 10, from which we can observe that with higher β, the bias is mitigated better, and converges to a certain degree. However, the impact of β is not as strong as that of α (the value changes of $RSP@15$, and $REO@15$ in Figure 10 are smaller than those in Figure 9).

6 CONCLUSION AND FUTURE WORK

In this paper, we study the issue of item under-recommendation bias in the personalized ranking task. We first propose two bias metrics designed specifically for personalized ranking recommendation tasks based on well known concepts of statistical parity and equal opportunity. Then we empirically show that the influential Bayesian Personalized Ranking model is vulnerable to the inherent data imbalance and tends to generate biased recommendations w.r.t. the proposed bias metrics. Next we propose a novel debiased personalized ranking model incorporating adversarial learning to augment the proposed bias metrics. At last, extensive experiments show the effectiveness of the proposed model over other state-of-the-art alternatives.

In our future work, we are interested in investigating position-aware bias metrics, which take the ranking order into account when evaluating the recommendation bias. We are also interested in exploring concepts well studied in the supervised learning community – such as equalized odds, disparate treatment, and disparate impact – in the context of personalized ranking systems.

ACKNOWLEDGMENTS

This work is, in part, supported by NSF (#IIS-1939716 and #IIS-1841138).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 42–46.

[3] 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*.

[4] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. *arXiv preprint arXiv:1903.00780* (2019).

[5] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075* (2017).

[6] 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*.

[7] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Conference on Fairness, Accountability and Transparency*. 202–214.

[8] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. [n.d.]. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*.

[9] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*.

[10] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* (2016).

[11] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 161–169.

[12] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[13] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference*.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[15] Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* (2011).

[16] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on Recommendation Independence for a Find-Good-Items Task. (2017).

[17] T Kamishima, S Akaho, H Asoh, and J Sakuma. 2013. Efficiency Improvement of Neutrality-Enhanced Recommendation.. In *Decisions@RecSys*.

[18] Toshihiro Kamishima, S Akaho, H Asoh, and J Sakuma. 2018. Recommendation Independence. In *Conference on Fairness, Accountability and Transparency*.

[19] T Kamishima, S Akaho, H Asoh, and I Sato. [n.d.]. Model-based approaches for independence-enhanced recommendation. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[22] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM CIKM Conference*.

[23] Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Geri Gay, Laura Granka, Fabio Pellacini, and Bing Pan. 2008. Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the American Society for Information Science and Technology* 59, 7 (2008), 1041–1052.

[24] Gilles Louppe, Michael Kagan, and Kyle Cranmer. 2017. Learning to pivot with adversarial networks. In *Advances in Neural Information Processing Systems*.

[25] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference*.

[26] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. ACM, 677–686.

[27] M Nilashi, D Jannach, O bin Ibrahim, Mohammad D Esfahani, and H Ahmadi. 2016. Recommendation quality, transparency, and website quality for trust-building in recommendation agents. *Electronic Commerce Research and Applications* (2016).

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press.

[29] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 154–162.

[30] Sujith Xavier. 2016. Learning from below: Theorising Global Governance through Ethnographies and Critical Reflections from the Global South. *Windsor YB Access Just.* (2016).

[31] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*. 2921–2930.

[32] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. *arXiv preprint arXiv:1801.07593* (2018).

[33] 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.