# Neutralizing Popularity Bias in Recommendation Models

Guipeng Xv*
xvguipeng.xgp@alibaba-inc.com
School of Informatics, Xiamen University
Xiamen, China

Chen Lin
chenlin@xmu.edu.cn
School of Informatics, Xiamen University
Xiamen, China

Hui Li
hui@xmu.edu.cn
School of Informatics, Xiamen University
Xiamen, China

Jinsong Su
jssu@xmu.edu.cn
School of Informatics, Xiamen University
Xiamen, China

Weiyao Ye
weiyao_ye@stu.hqu.edu.cn
College of Computer Science and Technology, Huaqiao University
Xiamen, China

Yewang Chen
ywchen@hqu.edu.cn
College of Computer Science and Technology, Huaqiao University
Xiamen, China

## ABSTRACT

Most existing recommendation models learn vectorized representations for items, i.e., item embeddings to make predictions. Item embeddings inherit popularity bias from the data, which leads to biased recommendations. We use this observation to design two simple and effective strategies, which can be flexibly plugged into different backbone recommendation models, to learn popularity neutral item representations. One strategy isolates popularity bias in one embedding direction and neutralizes the popularity direction post-training. The other strategy encourages all embedding directions to be disentangled and popularity neutral. We demonstrate that the proposed strategies outperform state-of-the-art debiasing methods on various real-world datasets, and improve recommendation quality of shallow and deep backbone models.

## CCS CONCEPTS

• **Information systems → Recommender systems**.

## KEYWORDS

recommender systems, popularity bias, disentangled representation

**ACM Reference Format:**
Guipeng Xv, Chen Lin, Hui Li, Jinsong Su, Weiyao Ye, and Yewang Chen. 2022. Neutralizing Popularity Bias in Recommendation Models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531907

## 1 INTRODUCTION

Recommender Systems (RS) have been widely applied in our daily lives [2]. Despite their huge success in E-commerce and many other domains, most RS suffer from popularity bias. They recommend popular items much more frequently than long-tail items, which have negative impacts on both users and businesses. The user experience is harmed because of non-personalized recommendations. Niche items are unfairly treated and revenue decline is expected for item providers. Moreover, there exists a vicious cycle of popularity bias: since user selection will be affected by how RS expose items, the popular items will become more and more popular.

**Table 1: Percentages $r$ of embedding directions in which popular and long-tail items are significantly different (t-test with confidence $\geq 0.95$). Percentages $p$ of embedding directions that are positively correlated with popularity (Spearman's Rank Correlation $\rho > 0$).**

| RS model | BPR | LightGCN | WMF | ItemAE | eALS |
|---|---|---|---|---|---|
| r | 95.31% | 100% | 73.44% | 70.31% | 43.75% |
| p | 59.38% | 53.12% | 56.25% | 50.00% | 62.50% |

Most recommendation models, including shallow models such as MF [19] and state-of-the-art deep learning based models [13, 14], represent items as numerical vectors called embeddings. A natural assumption is that biased predictions are made because the item embeddings inherit unintended popularity bias from user feedback. To investigate this assumption, we train five well-known recommendation models (i.e., BPR [21], LightGCN [12], WMF [20], ItemAE [23], eALS [15]) on an RS benchmark dataset MovieLens100K [11]. We first extract item embeddings for popular (i.e., top 10% movies with most ratings) and long-tail items (i.e., bottom 10% movies). As illustrated in Figure 1, popular and long-tail items cluster in distant regions of the embedding space. Moreover, we conduct statistical analysis to uncover the association between each embedding direction and popularity. We conduct paired samples t-test to popular and long-tail items on each embedding direction, and as shown in Table 1, a majority of directions are significantly different for popular and long-tail items. Next, for all items, on each embedding direction, we compute Spearman Rank Correlation between the direction and each item's corresponding popularity. As shown in Table 1, more than half of the directions are positively correlated with the popularity of items.

The above observation motivates us to seek a de-biasing strategy by learning popularity-neutral item embeddingss. This is not a

*Work done during internship at Alibaba Group.


*SIGIR '22, July 11–15, 2022, Madrid, Spain.*

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531907

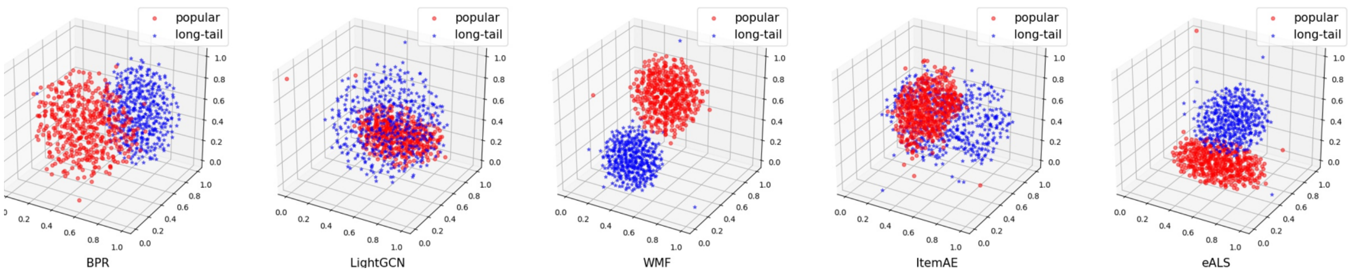


**Figure 1: T-sne visualizations of representations for popular and long-tail items learned by different recommendation models**

trivial problem as two challenges must be addressed. (1) Trade-off between unbiased recommendation (i.e., fair positioning of popular and long-tail items in the recommendation list) and overall recommendation accuracy (e.g., HitRatio at topK recommendations) [3] is often observed for previous de-biasing strategies [29, 32], thus the proposed debiasing strategy can not hurt the overall recommendation accuracy. (2) Since many recommendation models exhibit popularity bias in item embeddings, the debiasing strategy should be able to be applied to different recommendation models and achieve robust performance enhancement.

Although popularity bias has been extensively studied in the literature [1, 8–10, 16, 25–27, 29, 31–34], they mainly fall into three categories [7]. (1) At data level: inverse propensity scoring (IPS) methods [9, 16, 27] down-weigh popular items in the training process. (2) At loss level: unbiased objective methods augment the loss function to balance popular and long-tail items in the recommendation results [1, 34]. (3) At model level: causal inference methods [5, 25, 26, 29, 31, 32] use counterfactual reasoning to predict user interaction. De-biasing at the representation level and its effect in recommendation quality have remained relatively unexplored.

There is also a recent surge of research [4, 24, 30] in de-biasing word embeddings in Natural Language Processing. However, they mainly focus on bias related to categorical attributes such as gender bias, and they detect and reduce bias in pre-trained word embeddings. They can not be applied to learn item embeddings that are neutral with respect to a continuous attribute such as popularity.

In this paper, we present two strategies to derive popularity neutral item embeddings. The two strategies differ in the phase when neutrality is obtained. The first strategy isolats popularity bias in one embedding direction during training, and neutralizes the popularity direction during prediction. The second strategy encourages all directions to be disentangled and popularity neutral in the learning. Both strategies are implemented by adding regularization terms in the loss function, and thus can be flexibly plugged into different backbone models and reduce popularity bias while preserving the overall recommendation accuracy. Our experiments on various real RS datasets demonstrate that the proposed strategies can robustly improve the recommendation accuracy and reduce popularity bias on both shallow and deep models. The proposed strategies outperform state-of-the-art debiasing methods in terms of recommendation accuracy and recomendation fairness.

In summary, our contributions are two-fold. (1) We present two simple and effective strategies to neutralize popularity bias in item embeddings, in a pre-training and an in-training manner, respectively. (2) We show that directly neutralizing popularity bias in item embeddings can greatly improve recommendation quality of different backbone models on a variety of recommendation benchmark datasets.

## 2 RELATED WORK

Recent years have witnessed a rapid growth of research papers on mitigating popularity bias in RS [1, 8–10, 16, 25–27, 29, 31–34]. One line of existing work addresses popularity bias from training data level by Inverse Propensity Scoring (IPS) [9, 16, 27], which re-weights each instance by the inverse popularity value, thus popular items are imposed lower weights, while the long-tail items are boosted. IPS-based methods can achieve state-of-the-art performance, however, they are highly sensitive to the weighting strategy. The second line incorporates regularization terms in the loss function, which usually reflects the degree of bias in the recommendation results. For example, the long-tail coverage in recommendation lists [1] or the popularity-rank correlation for users (PRU) and items (PRI) [34]. The third line alleviates popularity bias at model level, including exposure dependent models from missing-not-at-random implicit feedback [22] and recent causal learning [5, 28, 32] methods that estimate the causal effects of the treatment variables (e.g., exposure) on the feedback outcome.

In summary, literature that considers debiasing directly at embedding level is very limited. To the best of our knowledge, CausE [5] and DICE [32] are most similar to our work. However, they both learn two sets of embeddings instead of operating on embedding directions. Furthermore, CausE [5] trains unbiased embeddings on a small unbiased dataset. The unbiased embedding is more noisy because of insufficient training on the small dataset. DICE [32] needs cause-specific data under the framework of multi-task learning.

## 3 METHODOLOGY

**Preliminaries**. Our goal is to design methods that are generally applicable to any recommendation model $M$ which learns to encode item features in a matrix $\mathbf{V} \in \mathbb{R}^{N \times D}$, where $N$ is the number of items, and $D$ is the dimension size. The item representation for item $i, i = 1, \cdots, N$ is a row of $\mathbf{V}$, denoted by $\mathbf{V}_{i,:} \in \mathbb{R}^{1 \times D}$, which is a $D-$dimensional numerical row vector. Similarly, the $d-$th column of $\mathbf{V}$ is denoted by $\mathbf{V}_{:,d} \in \mathbb{R}^{N}$, which represents the $d-$th direction of the item space. The subscripts can be ranges, for example

$\mathbf{V}_{i,d}, d = 1, \cdots, D$ denotes the $d-$th component of item $i$'s representation, while $\mathbf{V}_{1:n,d}$ denotes representations of items $i = 1, \cdots, n$ in direction $d$. The item representations are usually a separate part of the model parameters, or are obtained by some feature transformation modules using a set of trainable weights $\phi$. Without loss of generality, the model predictions are made by $M_\phi(\mathbf{U}, \mathbf{V})$, where $M()$ is a function operated on user representations $\mathbf{U}$ and item representatons $\mathbf{V}$. The model parameters, including $\phi$ or/and $\mathbf{U}, \mathbf{V}$ are learned via minimizing a recommendation loss function $\mathbb{L}^{RS}$. As depicted in Section 1, such a training paradigm does not guarantee unbiased item representations. We next describe two strategies to de-bias item representations $\mathbf{V}$.

## 3.1 PID: Post-training De-biasing

A straightforward approach is to remove components in item representations that are biased towards popular items. However, as shown in Table 1, many directions are associated with popularity bias, removing them will seriously harm the RS performance.

Our intuition is to isolate one direction to be popularity biased during training, and neutralize this popularity direction post-training. Since only the popularity direction is corrected, information captured by other directions will be preserved and the recommendation performance is optimized.

**Algorithm 1:** Framework of PID

**Input:** loss coefficient $\alpha^{PID} \in (0, 1)$, popularity vector $\mathbf{p}$
**Output:** Predictions $M_\phi(\mathbf{U}, \mathbf{V})$
1 Randomly initialize $\mathbf{\Theta} = (\mathbf{U}, \mathbf{V}, \phi)$, fix $\mathbf{V}_{:,D} = \mathbf{p}$;
2 **for** *number of training epochs* **do**
3 Maximize $S(\mathbf{V}_{:,1:D-1}\mathbf{w}, \mathbf{p})$ with respect to $\mathbf{w}$;
4 Minimize $\alpha^{PID} S(\mathbf{V}_{:,1:D-1}\mathbf{w}, \mathbf{p}) + (1 - \alpha^{PID})\mathbb{L}^{RS}$ by SGD with $\mathbf{w}, \mathbf{p}$ fixed;
5 Neutralize $\mathbf{p} = \mathbf{0}$ in $\mathbf{V}$;
6 Compute $M_\phi(\mathbf{U}, \mathbf{V})$;

Algorithm 1 describes the framework of post-training item representation de-biasing (PID). To define a popularity direction, we simply compute the popularity of each item and assign a popularity vector $\mathbf{p} \in \mathbb{R}^N$, where $\mathbf{p}_i$ is the number of interactions (e.g., clicks, ratings, etc.) item $i$ receives. To initialize the training process, we fix the last column of the item space to be equal to the popularity vector (line 1). To isolate the popularity direction, we attempt to reconstruct popularity direction from the other $D - 1$ directions, i.e., $\mathbf{V}_{:,1:D-1}\mathbf{w}$, where $\mathbf{w} \in \mathbb{R}^{D-1}$ is a learnable reconstruction coefficient vector. The reconstruction is evaluated by a similarity metric $S()$ on $\mathbf{V}_{:,1:D-1}\mathbf{w}$ and $\mathbf{p}$. We alternatively maximize the similarity (line 3) and minimize the recommendation loss $\mathbb{L}^{RS}$, regularized by the similarity (line 4). Note that the minimization procedure is a set of stochastic gradient descent steps, depending on the actual implementation of the backbone model. The coefficient $\alpha^{PID}$ balances between recommendation performance $\mathbb{L}^{RS}$ and popularity independence of the subspace $\mathbf{V}_{:,1:D-1}$. In testing, we neutralize the popularity direction, e.g., by setting all components to zeros (line 5) and use the item representations to make predictions (line 6).

## 3.2 IID: In-training De-biasing

Instead of compressing popularity bias in one direction and removing it post-training, an alternative strategy is to encourage all directions to be popularity neutral during training. Intuitively, we can again define the popularity vector $\mathbf{p}$, evaluate the similarity for every direction, and minimize the aggregated similarity over all directions. However, since each direction of the item representation is essentially an arbitrary combination of distinct feature aspects, the risk of over-computing popularity bias is high. For example, suppose "reputation" of an item is one aspect that influences user feedback in RS, and it is correlated with item popularity. Thus, the similarity between "reputation" and popularity will be high, and will contribute for multiple times by all directions that encode "reputation".

To eliminate the effect of over-computing popularity bias, our solution is to disentangle the directions by imposing orthogonal regularization. Thus, the optimization objective is:

$$\mathbb{L}^{IID} = \alpha_1^{IID} \|\mathbf{V}^T\mathbf{V} - \mathbf{I}\|_2^2 + \alpha_2^{IID} \sum_d S(\mathbf{V}_d, \mathbf{p}) + (1 - \sum_{i=1,2} \alpha_i^{IID})\mathbb{L}^{RS}, \tag{1}$$

where $\|\mathbf{V}^T\mathbf{V} - \mathbf{I}\|_2^2$ is the orthogonal regularization to learn independent directions, $\mathbf{I} \in \mathbb{R}^{D\times D}$ is the identity matrix, $\sum_d S(\mathbf{V}_{:,d}, \mathbf{p})$ is the aggregated similarity over all directions, $S()$ is the similarity metric, $\alpha_i^{IID} \in (0, 1), i = 1, 2$ are coefficients to control the degree of disentanglement and popularity neutrality.

# 4 EXPERIMENT

In this section, we conduct experiments in order to answer the following research questions: **RQ1**: Do PID and IID improve the recommendation quality of different recommendation models, and outperform other debiasing methods? **RQ2**: How does the hyper-parameter, i.e., $\alpha$, affect the recommendation performance?

In the following, we first demonstrate our experiment setup in Sec. 4.1. Then, the performance of PID and IID is verified by both shallow and deep learning based recommendation models on three well-known recommendation benchmarks in Sec. 4.2 (RQ1). Finally, we investigate the parameter influence in Section 4.3 (RQ2). Source codes are available[1].

## 4.1 Experimental Setup

**Dataset**. We use three benchmark data sets for RS in our experiments: ML-100K[2], Epinions[3] and Amazon Digital Music[4]. We apply 10-core pre-processing on ML-100k and Epinions dataset and 5-core pre-processing on Amazon Digital Music dataset to make sure each user/item has sufficient feedback. Tab. 3 lists the statistics of the three datasets.

**Backbone recommendation models**. We apply PID and IID to two recommendation baselines: (1) BPR [21] learns latent factors for users and items by optimizing a triplet loss based on the inner product of the user and item factors. (2) LightGCN [12]learns user and item embeddings by linearly propagating them on the user-item

[1]https://github.com/XMUDM/NeutralizingBias
[2]https://grouplens.org/datasets/movielens/100k/
[3]http://trustlet.org/downloaded_epinions.html
[4]http://jmcauley.ucsd.edu/data/amazon/

**Table 2: Recommendation performance of different methods. Best performance is shown in bold font. Second best performance is underlined. Improvements (Imp.) of PID and IID with respect to the backbone: ↑: better performance, ↓: worse performance, or −: comparable performance.**

| Dataset | ML-100K | | | | Amazon DM | | | | Epinions | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | R@20 | HR@20 | NDCG@20 | PRU | R@20 | HR@20 | NDCG@20 | PRU | R@20 | HR@20 | NDCG@20 | PRU |
| BPR | 0.1011 | 0.5101 | 0.0790 | 0.6598 | 0.0736 | 0.1645 | 0.0382 | 0.5465 | 0.0146 | 0.0929 | 0.0100 | 0.6638 |
| IPS | 0.0964 | 0.4963 | 0.0740 | 0.5481 | 0.0578 | 0.1283 | 0.0297 | 0.3968 | 0.0086 | 0.0656 | 0.0066 | 0.4606 |
| IPSC | 0.1149 | 0.5534 | 0.0893 | 0.5521 | 0.0583 | 0.1333 | 0.0306 | 0.3875 | 0.0115 | 0.0813 | 0.0084 | 0.4893 |
| IPSNC | 0.1213 | 0.5708 | 0.0881 | 0.5879 | 0.0823 | 0.1885 | 0.0454 | 0.4717 | 0.0198 | 0.1300 | 0.0147 | 0.5823 |
| CausE | 0.0950 | 0.4974 | 0.0670 | 0.7527 | 0.0151 | 0.0399 | 0.0080 | 0.5764 | 0.0066 | 0.0462 | 0.0042 | 0.6850 |
| DICE | 0.1114 | 0.5259 | 0.0738 | 0.7017 | 0.0576 | 0.1232 | 0.0300 | 0.6534 | 0.0179 | 0.1061 | 0.0117 | 0.7414 |
| PID | 0.1231 | 0.5767 | 0.0912 | 0.5735 | 0.0864 | 0.1968 | 0.0478 | 0.4647 | 0.0203 | 0.1312 | 0.0147 | 0.5657 |
| Imp. | ↑ 22% | ↑ 13% | ↑ 15% | ↑ 13% | ↑ 17% | ↑ 19% | ↑ 25% | ↑ 15% | ↑ 39% | ↑ 41% | ↑ 47% | ↑ 15% |
| IID | **0.1343** | **0.6021** | **0.1002** | **0.4994** | **0.0918** | **0.1968** | **0.0524** | 0.4172 | **0.0227** | **0.1377** | **0.0172** | **0.4293** |
| Imp. | ↑ 33% | ↑ 18% | ↑ 27% | ↑ 24% | ↑ 25% | ↑ 20% | ↑ 37% | ↑ 24% | ↑ 55% | ↑ 48% | ↑ 72% | ↑ 35% |
| LightGCN | 0.0957 | 0.4825 | 0.0691 | 0.8932 | 0.0090 | 0.0216 | 0.0042 | 0.6227 | 0.0034 | 0.0194 | 0.0021 | 0.7736 |
| IPS | 0.0235 | 0.2074 | 0.0221 | 0.4419 | 0.0082 | 0.0238 | 0.0039 | 0.3456 | 0.0032 | 0.0198 | 0.0021 | 0.9284 |
| IPSC | 0.1010 | 0.4921 | 0.0744 | 0.6605 | 0.0105 | 0.0254 | 0.0047 | 0.5402 | 0.0031 | 0.0194 | 0.0020 | 0.9365 |
| IPSNC | 0.1000 | 0.5016 | 0.0777 | 0.6531 | 0.0089 | 0.0240 | 0.0040 | 0.4963 | 0.0027 | 0.0194 | 0.0018 | 0.6011 |
| CausE | 0.0387 | 0.2804 | 0.0311 | 0.8874 | 0.0056 | 0.0173 | 0.0026 | 0.4050 | 0.0032 | 0.0196 | 0.0020 | 0.5019 |
| DICE | 0.1128 | 0.5407 | 0.0835 | 0.7629 | 0.0099 | 0.0262 | 0.0048 | 0.5878 | 0.0030 | 0.0226 | 0.0021 | 0.7531 |
| PID | 0.0818 | 0.4561 | 0.0610 | 0.8144 | 0.0077 | 0.0243 | 0.0040 | **0.2233** | **0.0034** | 0.0193 | 0.0020 | 0.6811 |
| Imp. | ↓ | ↓ | ↓ | ↑ 9% | ↓ | ↑ 13% | - | ↑ 64% | - | - | - | ↑ 12% |
| IID | 0.1016 | 0.4953 | 0.0691 | 0.7372 | 0.0167 | 0.0377 | 0.0071 | 0.3296 | 0.0034 | 0.0194 | 0.0024 | 0.4688 |
| Imp. | ↑ 6% | ↑ 3% | - | ↑ 17% | ↑ 86% | ↑ 75% | ↑ 69% | ↑ 47% | - | - | ↑ 14% | ↑ 39% |

**Table 3: Statistics of datasets**

| Data | #Users | #Items | #Ratings | Sparsity |
|---|---|---|---|---|
| ML100K | 943 | 1,152 | 97,952 | 0.0902 |
| Amazon Digital Music | 5,531 | 3,568 | 64,706 | 0.0033 |
| Epinion | 10,706 | 8,945 | 300,304 | 0.0032 |

interaction graph. They are both commonly adopted in the literature as backbone models [32]. We use the public implementation[5] .

**Competitors**. We compare PID and IID with several classic inverse propensity scoring methods and recent causal inference methods, including (1) IPS [17] re-weights each instance by the inverse popularity value. (2) IPSC [6] adds max-capping on IPS weighing. (3) IPSNC [9] adds max-capping and normalization on IPS weighing. (4) CausE [5] trains two set of embeddings on a biased and an unbiased dataset respectively and force them to be similar with each other. (5) DICE [32] learns user preference and popularity bias into two sets of embeddings. The competitors can be applied to the backbone models.

**Implementation**. For the backbone recommendation models, the embedding dimensionality of users and items is 64. We set 0.001 as initial learning rate and the weight decay rate is 5e-6. We use Adam [18] for optimization. In PID and IID, we use Pearson Correlation Coefficient (PCC) as the similarity measurement $S()$. For a fair comparison, we split the datasets following the standard protocol [5, 32] to ensure all items have the same prevalence in the testing set.

**Evaluation metrics.** To analyze whether the recommendations are accurate, we use three commonly adopted ranking metrics: Recall, HitRatio, and NDCG at top K results. For each user, we compare the top-K recommendations with the ground-truth (i.e., which items receive user feedback in the testing set), and the evaluation metrics are computed over all users:

$$\begin{aligned} R@K &= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{\sum_{j\leq K} IK_{u,j}}{\sum_j IK_{u,j}}, \\ HR@K &= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{\sum_{j\leq K} IK_{u,j}}{K}, \\ NDCG@K &= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{Z_u}\sum_{j=1}^{K}\frac{2^{IK_{u,j}}-1}{\log_2(1+j)}, \end{aligned} \tag{2}$$

where $IK_j$ returns 1 if the recommendation at position $j$ receives feedback in the ground-truth, and returns 0 otherwise, and $Z_u$ is a normalization term which ensures that perfect ranking for user $u$ has a value of 1. Thus, R@K measures how many user preferred items are recommended, HR@K measures if the recommendation at least captures one preferred item, and NDCG@K measures the ranking accuracy from preferred to non-preferred items. The higher R@K, HR@K and NDCG@K are, the more accurate recommendations are made.

In addition, we also measure whether the recommendations are biased towards popular items by PRU [34]. For each user $u$, among the items that $u$ interacts with in the testing set $\mathcal{I}_u^+$, we compare their ranking positions in the recommendation list $rl(\mathcal{I}_u^+)$ and their

[5]https://github.com/tsinghua-fib-lab/DICE

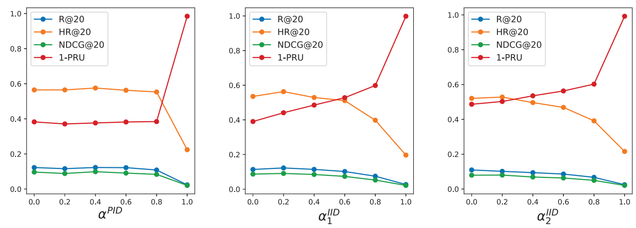


**Figure 2: Recommendation performance with respect to different loss coefficients by PID of BPR on ML-100K**

popularity positions $\mathbf{p}(\mathcal{I}_u^+)$, and compute PRU as:

$$PRU = \frac{1}{|\mathcal{U}|} SRC\big(rl(\mathcal{I}_u^+), \mathbf{p}(\mathcal{I}_u^+)\big), \tag{3}$$

where $SRC(\mathbf{x}, \mathbf{y}) = \frac{cov(\mathbf{x},\mathbf{y})}{\sigma(\mathbf{x})\sigma(\mathbf{y})}$ is the Spearman’s Rank Coefficients, $cov()$ is covariance of the rank variables, $\sigma$ is the the standard deviations of the rank variables. The SRC is averaged over all users. Higher R@K, HR@K, NDCG@K and lower PRU values imply more accurate and less biased recommendations.

### 4.2 Comparative Performance

To answer **RQ1**, we conduct the backbone models with different debiasing methods. We have the following observations from Table 2. (1) IID with BPR outperforms all competitors in almost all evaluation metrics. PID with BPR obtains the second best performance. (2) The proposed methods can robustly reduce popularity bias of backbone models BPR and LightGCN on different datasets, while preserving accurate recommendations. We can see that PID and IID greatly improve the R@20, HR@20, NDCG@20, and PRU results for BPR on all datasets. Applied to LightGCN, PID and IID can obtain higher or comparable R@20, HR@20 and NDCG@20 in most cases. (3) On the contrary, IPS style methods tend to achieve good PRU at the cost of decreased recommendation accuracy. Causal inference methods are not as effective as IPS style methods in debiasing and their PRU results tend to be much higher.

### 4.3 Impact of Parameters

To answer **RQ2**, we analyze the change of recommendation performance with respect to different values of loss coefficients when applying PID and IID to BPR on the ML-100K dataset. For better illustration purpose, we report $1-PRU$ instead of $PRU$, so that higher values imply higher recommendation quality for all evaluation metrics. We set the coefficient to 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, respectively. For $\alpha_1^{IID}, \alpha_2^{IID}$, we change them separately, i.e., we change one coefficient at a time and fix the other as 0.

**Analysis**. We have the following observations from Figure 2. (1) For both PID and IID, larger coefficient value leads higher PRU. When the coefficient value equals to one, $PRU$ approaches to zero, indicating that the recommendations have no relationship with popularity. Thus, the regularization terms proposed in PID and IID can directly control the degree of bias in RS. (2) For IID, although the recommendation accuracy decreases according to larger coefficient values, we can obtain stable performance with $\alpha^{PID} < 0.6$, and still increase fairness (i.e., higher $1-PRU$). (3) $\alpha_1^{IID}$ has a stronger impact on PRU than $\alpha_2^{IID}$: increasing $\alpha_1^{IID}$ reduces $PRU$ more than $\alpha_2^{IID}$. It verifies our assumption that popularity bias can be correctly computed only if the representations are disentangled in latent spaces.

## 5 CONCLUSION

We explore how unbiased recommendations can be obtained in a model-independent manner, by removing popularity bias in the item embeddings, or training popularity-neutral item embeddings. We show that these simple strategies can effectively enhance recommendation quality, in terms of recommendation accuracy and fairness, of different backbone recommendation models.

## ACKNOWLEDGEMENTS

Chen Lin is the corresponding author. This work is supported by the Natural Science Foundation of China (No. 61972328,62002303), Alibaba Innovative Research Program, Natural Science Foundation of Fujian Province China (No. 2020J05001, 2020J06001), and Youth Innovation Fund of Xiamen (No. 3502Z20206059).

# REFERENCES


[1] 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, RecSys 2017, Como, Italy, August 27-31, 2017*. ACM, 42–46.

[2] Charu C. Aggarwal. 2016. *Recommender Systems - The Textbook*. Springer.

[3] Amin Bigdeli, Negar Arabzadeh, Shirin Seyedsalehi, Morteza Zihayat, and Ebrahim Bagheri. 2021. *On the Orthogonality of Bias and Utility in Ad Hoc Retrieval*. Association for Computing Machinery, New York, NY, USA, 1748–1752.

[4] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 4349–4357.

[5] Stephen Bonner and Flavian Vasile. 2018. Causal Embeddings for Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. ACM, New York, NY, USA, 104–112.

[6] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis Xavier Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: the example of computational advertising. *J. Mach. Learn. Res.* 14, 1 (2013), 3207–3260.

[7] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *CoRR* abs/2010.03240 (2020). arXiv:2010.03240

[8] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 579–588.

[9] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About PlaylistRecommendation Algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. ACM, New York, NY, USA, 420–428.

[10] Priyanka Gupta, Ankit Sharma, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2021. CauSeR: Causal Session-based Recommendations for Handling Popularity Bias. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 3048–3052.

[11] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2015), 19 pages.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 639–648.

[13] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Trans. Knowl. Data Eng.* 30, 12 (2018), 2354–2366.

[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, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 173–182.

[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 549–558.

[16] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*. ACM, 781–789.

[17] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2018. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. ijcai.org, 5284–5288.

[18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[19] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[20] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan M. Lukose, Martin Scholz, and Qiang Yang. 2008. One-Class Collaborative Filtering. In *ICDM*. IEEE Computer Society, 502–511.

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618 (2012). arXiv:1205.2618

[22] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. *Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback*. Association for Computing Machinery, New York, NY, USA, 501–509.

[23] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy) *(WWW '15 Companion)*. Association for Computing Machinery, New York, NY, USA, 111–112.

[24] Tianlu Wang, Xi Victoria Lin, Nazneen Fatema Rajani, Bryan McCann, Vicente Ordonez, and Caiming Xiong. 2020. Double-Hard Debias: Tailoring Word Embeddings for Gender Bias Mitigation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 5443–5453.

[25] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 1717–1725.

[26] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 1791–1800.

[27] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-Not-at-Random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. ACM, New York, NY, USA, 279–287.

[28] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A Survey on Causal Inference. *ACM Trans. Knowl. Discov. Data* 15, 5, Article 74 (may 2021), 46 pages.

[29] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. ACM, New York, NY, USA, 11–20.

[30] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning Gender-Neutral Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 4847–4853.

[31] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2021. Popularity Bias Is Not Always Evil: Disentangling Benign and Harmful Bias for Recommendation. *CoRR* abs/2109.07946 (2021). arXiv:2109.07946

[32] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. ACM, New York, NY, USA, 2980–2991.

[33] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity Bias in Dynamic Recommendation. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2439–2449.

[34] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*. ACM, 85–93.