Providing Item-side Individual Fairness for Deep Recommender Systems

Xiuling Wang, Wendy Hui Wang
Stevens Institute of Technology
Hoboken, NJ, USA
xwang193@stevens.edu,hui.wang@stevens.edu

ABSTRACT

Recent advent of deep learning techniques have reinforced the development of new recommender systems. Although these systems have been demonstrated as efficient and effective, the issue of item popularity bias in these recommender systems has raised serious concerns. While most of the existing works focus on *group fairness* at item side, individual fairness at item side is left largely unexplored. To address this issue, in this paper, first, we define a new notion of individual fairness from the perspective of items, namely (α, β)-*fairness*, to deal with item popularity bias in recommendations. In particular, (α, β)-fairness requires that similar items should receive similar coverage in the recommendations, where α and β control item similarity and coverage similarity respectively, and both item and coverage similarity metrics are defined as task specific for deep recommender systems. Next, we design two bias mitigation methods, namely *embedding-based re-ranking* (ER) and *greedy substitution* (GS), for deep recommender systems. ER is an *in-processing* mitigation method that equips (α, β)-fairness as a constraint to the objective function of the recommendation algorithm, while GS is a *post-processing* approach that accepts the biased recommendations as the input, and substitutes high-coverage items with low-coverage ones in the recommendations to satisfy (α, β)-fairness. We evaluate the performance of both mitigation algorithms on two real-world datasets and a set of state-of-the-art deep recommender systems. Our results demonstrate that both ER and GS outperform the existing minimum-coverage (MC) mitigation solutions [Koutsopoulos and Halkidi 2018; Patro et al. 2020] in terms of both fairness and accuracy of recommendations. Furthermore, ER delivers the best trade-off between fairness and recommendation accuracy among a set of alternative mitigation methods, including GS, the hybrid of ER and GS, and the existing MC solutions.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Information systems** → **Data analytics**.

'22, June 21 – 24, 2022, Seoul, South Korea

ACM ISBN 978-1-4503-9352-2/22/06…$15.00
https://doi.org/10.1145/3531146.3533079

KEYWORDS

Individual fairness, deep recommender systems, algorithmic fairness in machine learning

ACM Reference Format:
Xiuling Wang, Wendy Hui Wang. 2022. Providing Item-side Individual Fairness for Deep Recommender Systems. In *FAccT '22: ACM International Conference on Fairness, Accountability, and Transparency, June 21 – 24, Seoul, South Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3531146.3533079

1 INTRODUCTION

Given the explosive growth of information available on the Web, recommender systems have been designed to search for and recommend a small set of items to meet users' personalized interests. They have been widely applied in various fields to recommend items such as movies, products, jobs, and courses. Recently, there is an increasing interest in extending deep learning approaches to recommender systems. The core component of the deep recommender systems is the graph embedding module, which encodes the user-item interaction and any possible side information of users and items (e.g., knowledge graphs) into low-rank embedding. Then the recommendations are calculated according to the embeddings of users and items. Numerous deep learning based recommendation techniques have been developed.[1] In this paper, we consider the deep learning based recommender systems due to their effectiveness and popularity.

An important concern of the recommender systems is the *fairness* of recommendations, which apply to both customers (users) and suppliers (items). At user side, the users are concerned if they are treated equally by the recommender systems, regardless of their demographic property [Xiao et al. 2017; Yao and Huang 2017] nor their activeness on the recommender system platform [Fu et al. 2020; Li et al. 2021]. At item side, on the other hand, the main fairness concern is the item popularity bias (i.e., popular items can get more exposure than those unpopular ones in recommendations) [Abdollahpouri and Burke 2019; Basu et al. 2020; Koutsopoulos and Halkidi 2018; Patro et al. 2020]. In this paper, we mainly focus on fairness at item side.

Most of the existing work on item-side fairness considers *group fairness*, which groups items based on their popularity, and require items across different groups should be treated statistically similarly by the recommender system. Although these work can achieve fairness at group level, it has the potential to be unfair to specific individual items. For example, consider a particular group fairness requirement that the ratio of items from different groups

[1] We refer the readers to a number of surveys [Gao et al. 2020; Guo et al. 2020; Liu et al. 2019] on deep learning based recommender systems.

in the recommendations should equal to the ratio in the historical user-item interaction [Fu et al. 2020; Patro et al. 2020; Steck 2018]. While the recommender system recommends certain unpopular items to achieve such group fairness requirement, it is unfair to other unpopular items, for example, those new items that have few user-item interaction, that are not recommended even though they are very similar to the recommended items. In addition, forcing statistically similar coverage rate across popular and unpopular item groups can lead to dramatic loss in recommendation accuracy as the distribution across these groups in historical user-item interactions are significantly different. However, while most of the existing works mainly focus on item-side group fairness in recommendations [Abdollahpouri et al. 2017, 2019; Adomavicius and Kwon 2011; Kamishima et al. 2012], the item-side individual fairness in recommendations has been rarely studied.

A straightforward approach to provide item-side individual fairness is to require that the coverage of each item in the recommendations must reach a given threshold, where the item coverage is measured as the percentage of users who are recommended with the given item [Koutsopoulos and Halkidi 2018]. However, requiring the minimum coverage for *all items* is too rigid and may lead to significant accuracy loss of recommendations [Patro et al. 2019]. Indeed, our experiments show that the minimum-coverage (MC) approach can incur up to 40% accuracy reduction (Section 6).

To address the weakness of the existing MC solution, in this paper, we consider a new item-side individual fairness notion that is adapted from the state-of-the-art notion of individual fairness [Dwork et al. 2012]. In particular, instead of requiring all items must have a minimum coverage in recommendations, our fairness notion requires that *any pair of items that are similar should receive similar coverage in the recommendations*. Therefore, by enforcing fairness on similar items only, our individual fairness notion better addresses the trade-off between fairness and recommendation accuracy.

However, it is challenging to equip individual fairness with deep learning based recommender systems due to the following reasons. First, defining a proper similarity metric is not straightforward, as it has to be task-specific [Chouldechova and Roth 2018]. In our problem setting, the similarity of items should be defined in the context of the deep recommender systems. Second, as realizing fairness may incur large significant loss of recommendation accuracy, it is important to minimize the accuracy loss while equipping the individual fairness constraint with the deep recommender systems.

In this paper, we address these two challenges and make the following contributions. First, as the deep recommender systems make predictions based on user and item embeddings, we define the item similarity metric as evaluating the distance between item embeddings. Based on this similarity metric, we formally define a new notion of item-side individual fairness, namely (α, β)-*fairness*, to quantify the disparity of item coverage at individual level. Intuitively, (α, β)-fairness requires that any two similar items (i.e., the distance between their embedding is within α) must receive similar coverage (i.e., the difference in their coverage in recommendations is up to β), where α and β are user-specified thresholds. Second, we design two bias mitigation algorithms, namely *embedding-based re-ranking* (ER) and *greedy substitution* (GS) algorithms, to achieve (α, β)-fairness while maximizing the accuracy of the recommendations. ER is an *in-processing* mitigation method that takes the user/item embeddings as the input. From the embeddings, ER learns a ranking policy for recommendation, with (α, β)-fairness equipped as a constraint of the objective function of the ranking learning problem. On the other hand, GS is a model-agnostic, *post-processing* method that takes the original (biased) recommendations as the input. It achieves (α, β)-fairness by replacing the items of high coverage in the recommendations with similar ones but of low coverage. Last but not least, we conduct an extensive set of experiments on two real-world datasets and three state-of-the-art deep recommender systems. Our results demonstrate that both GS and ER algorithms can effectively mitigate individual item coverage disparity in the recommendation, while minimizing the accuracy loss of recommender systems by the mitigation. Furthermore, ER outperforms GS as well as the existing minimum-coverage approaches [Koutsopoulos and Halkidi 2018; Patro et al. 2020] in terms of the trade-off between fairness and accuracy of recommendations.

2 RELATED WORK

User-side fairness in recommendations. We categorize the existing works on user-side fairness in recommmendations into two classes: *group user-fairness* and *individual user-fairness*. Most of the existing group user-fairness works [Edizel et al. 2019; Yao and Huang 2017] group users by their values on the sensitive features. When the sensitive attributes are not available for defining group membership, the users can be grouped based on their activity [Fu et al. 2020; Li et al. 2021]. The fairness requirement is to balance the statistical measurement (e.g., error rate [Edizel et al. 2019] and average recommendation performance [Fu et al. 2020; Li et al. 2021]) of recommendations across different groups. User-side group fairness can be achieved by adding the fairness constraint to the objective function of recommendation problem [Fu et al. 2020; Li et al. 2021]. Regarding individual user-fairness, it can be measured as the disparity in recommendation quality as well as in explanation diversity disparity among different individual users [Fu et al. 2020].

Item-side fairness in recommendations. The existing efforts on item-side fairness in recommendations also can be classified into two types: *item-side group fairness* and *item-side individual fairness*. Regarding item-side group fairness, Celis et al. [Celis et al. 2019] consider the setting that items are grouped by the sensitive attributes associated with the items. They address the polarization phenomenon in recommendations (e.g., the news feed covers a small subset of topic groups), and design personalized algorithms that avoid polarization yet still optimize individual recommendation accuracy. Regarding item-side individual fairness, Koutsopoulos et al. [Koutsopoulos and Halkidi 2018] define the item-fairness requirement as that the coverage of all items must exceed a given threshold. The notion of individual item fairness by Chakraborty et al. [Chakraborty et al. 2017] is somewhat similar to [Koutsopoulos and Halkidi 2018]: all items should receive the amount of exposure proportional to their relevance. Unlike these two individual item-fairness notions that require minimum coverage for all items, our notion requires similar items must have similar coverage. We show in the experiments that our fairness notion better addresses the trade-off between fairness and recommendation accuracy than these existing works.

Fairness in ranking. Finding fair recommendations is relevant to the problem of finding a fair ranking of all items based on their predicted user-item probabilities. Zehlike et al. [Zehlike et al. 2017] extends the definition of group fairness to top-k ranking, and define the fair top-k ranking problem as ensuring that the proportion of protected group members in the top-k ranking is above a given threshold. Singh et al. [Singh and Joachims 2018] formulate fairness constraints on rankings in terms of exposure allocation. They design a fair ranking framework that maximizes the utility of ranking while satisfying the fairness of exposure constraints. Their following work [Singh and Joachims 2019] designs a learning-to-rank framework that optimizes a wide range of utility metrics (e.g. NDCG) with the constraint of fair exposure. Morik et al. [Morik et al. 2020] present a dynamic learning-to-rank approach that enforces merit based fairness [Biega et al. 2018; Singh and Joachims 2018] on groups of items, where the merit of an item is defined as its expected average relevance. The fairness notion requires that each item group should receive the exposure that is proportional to its relevance to users. Beutel et al. [Beutel et al. [n.d.]] treat recommendations as a point-wise prediction problem. Their notion of group item-fairness requires the likelihood of a clicked item being ranked above another relevant unclicked item is the same across both protected and unprotected groups. None of these efforts consider the same fairness notion as ours, thus their algorithms cannot be directly applied to our setting.

3 PRELIMINARIES

Deep learning based recommender systems. Typically, a recommender system considers a collection of user-item interactions (e.g., purchases and clicks), which can be represented as a *user-item graph* $\mathbf{G} = \{(u, i)|u \in \mathbf{U}, i \in \mathbf{I})\}$, where $\mathbf{U}$ and $\mathbf{I}$ denote the user and item sets respectively. The recommendation task is described as the following: Given a user-item graph $\mathbf{G}$ as the historical data, the goal is to predict the probability $p(u, i)$ that the item i will be adopted by the user u. Based on the predictions, each user will receive a *recommendation list* in which the items are ranked according to the prediction probability in descending order. In this paper, we only consider top-k recommendations, i.e. the items of the top-k probability.

Recently, deep learning techniques have been applied to recommender systems to achieve the recommendations of higher quality. Most of these deep learning based recommender systems follow the same idea: first, the deep learning models learn the graph embedding, which is the primarily representation of the given user-item graph in a lower dimension. An important property of the graph embedding is that they preserve the graph information that is obtained by aggregating information from multiple sources such as the knowledge graph (KG), the user-item interaction matrix, item's content, and item's attributes. Then the recommendations are made based on the learned graph embedding instead of the original graph. In particular, the probability $p(u, i)$ that the user u adopts the item i is calculated as $p(u, i) = f(\vec{u}, \vec{i})$, where $\vec{u}$ and $\vec{i}$ refer to the embedding of user u and item i respectively, and $f()$ refers to a function that maps the embedding of the user and item to the probability $p(u, i)$. The final recommendation results are generated from ranking of the predicted probability.

Algorithmic fairness. Over the past few years, a multitude of formal fairness definitions have been proposed. These fairness definitions can be categorized into two broad classes, namely *group fairness* and *individual fairness*. Group fairness [Calders et al. 2009; Calders and Verwer 2010; Feldman et al. 2015; Kamishima et al. 2011] defines specific groups and require that particular statistics, computed based on model decisions, should be equal for all groups. In particular, the *protected groups*, which is defined by the *sensitive attributes* (e.g., race and gender), should not be treated in a discriminatory way compared with the advantaged groups. Individual fairness, on the other hand, requires that similar individuals in the population are treated similarly [Dwork et al. 2012]. While most of the existing works focus on either user-fairness [Fu et al. 2020; Yao and Huang 2017] or group item-fairness [Celis et al. 2019] in recommendations, in this paper, we mainly focus on individual fairness at item side in recommender systems.

4 ITEM-SIDE INDIVIDUAL FAIRNESS

The conventional definition of individual fairness [Dwork et al. 2012] requires that an ML model should treat similar individuals with similar predictions. A simple adaption of this definition to our setting requires that *any two similar items must receive similar coverage in recommendations*. The main challenge is to define appropriate metrics to evaluate item similarity and item coverage similarity. These similarity metrics must be task-specific and faithfully represent the context [Ilvento 2020]. To address this challenge, first, we formally define *item similarity* and *item coverage disparity*.

Embedding-based item similarity. Given a universe of items $\mathbf{I}$, we say two items $i_1, i_2 \in \mathbf{I}$ are *α-similar*, denoted as $i_1 \overset{\alpha}{\approx} i_2$, if $d(i_1, i_2) \leq \alpha$, where $d : \mathbf{I} \times \mathbf{I} \rightarrow \mathbb{R}$ is a distance metric of items. There are quite a few candidates of distance metrics. Since we focus on deep learning based recommendation algorithms, we follow the literature [Badilla et al. 2020] and consider the embedding-based distance metric as the item similarity metric. Specifically, given two items $i_1, i_2 \in \mathbf{I}$, their distance $d(i_1, i_2)$ is measured based on the cosine similarity between their embeddings:

$$d(i_1, i_2) = 1 - \frac{\vec{i_1} \cdot \vec{i_2}}{|\vec{i_1}||\vec{i_2}|}, \tag{1}$$

where $\vec{i_1}$ and $\vec{i_2}$ are the embeddings of item nodes i_1 and i_2, and $\cdot$ is the inner product operator. We note that the cosine similarity based α-similarity is not transitive (i.e., $i_1 \overset{\alpha}{\approx} i_2$, $i_2 \overset{\alpha}{\approx} i_3$, but $i_1 \overset{\alpha}{\not\approx} i3$).

Item coverage disparity. Given a graph $\mathbf{G}$ that contains a set of users $\mathbf{U}$ and a set of items $\mathbf{I}$, let $\vec{R}_u = [R_{u,1}, \ldots, R_{u,k}]$ be the (top-k) recommendation list of user $u \in \mathbf{U}$. Let $\mathbf{R} = \{\vec{R}_u | u \in \mathbf{U}\}$ be the collections of recommendation lists for all users. We use $Cov(i)$ to denote the *coverage* of an item i in $\mathbf{R}$, which is measured as the number of occurrences of i in $\mathbf{R}$. We follow the literature [Chakraborty et al. 2017; Koutsopoulos and Halkidi 2018; Patro et al. 2020] and define the *coverage disparity* between items i and j (denoted as $CD(i_1, i_2)$) as the following:

$$CD(i_1, i_2) = \left| \frac{Cov(i_1)}{MCov(i_1, i_2)} - \frac{Cov(i_2)}{MCov(i_1, i_2)} \right| \tag{2}$$

Apparently, $CD(i_1, i_2) \in [0, 1]$. Intuitively, higher (smaller, resp.) $CD(i_1, i_2)$ indicates larger (smaller, resp.) coverage disparity between items i_1 and i_2. We use $\frac{Cov(i_1)}{MCov(i_1,i_2)}$, where $MCov(i_1, i_2) = max(Cov(i_1), Cov(i_2))$, to normalize the item coverage, as the item coverage may vary significantly. We note that the normalization is *local* - the same item in different item pairs can have different normalized item coverage. We do not define a global normalized item coverage value for each item, as individual fairness is measured for each item pair, not for each item.

Now, we are ready to define (α, β)-*fairness* based on item similarity and item coverage disparity.

Definition 4.1 ((α, β)-fairness). Given a graph $\mathbf{G}$ that consists of a set of items $\mathbf{I}$, we say a recommendation algorithm $\mathcal{A}$ provides (α, β)-*individual item-fairness* if and only if for all items $i_1, i_2 \in \mathbf{I}$ such that $i_1 \overset{\alpha}{\approx} i_2$, it must satisfy that $CD(i_1, i_2) \leq \beta$ (i.e., i_1 and i_2 are of β-*coverage disparity*), where α, β are user-specified thresholds.

Intuitively, Definition 4.1 requires that two similar items must receive similar coverage, where the similarity of items and their coverage are controlled by α and β respectively. Larger α as well as smaller β values indicate stronger fairness requirement. How α and β are valued affects the performance of recommendation accuracy and item fairness. We will discuss the impact of α and β values on fairness and recommendation accuracy in the experiments (Section 6).

5 BIAS MITIGATION ALGORITHMS

Given a graph $\mathbf{G}$ and an recommendation algorithm $\mathcal{A}$ that is biased, our goal is to generate new recommendations from $\mathbf{G}$ that satisfy (α, β)-fairness requirement. We design two mitigation methods which take different types of inputs:

- *Embedding-based Re-ranking* (**ER**) algorithm is an *in-processing* method that takes the node embeddings generated from the original graph as the input, and learns a stochastic ranking policy of items with (α, β)-fairness and maximized accuracy of the recommendations.
- *Greedy substitution* (**GS**) algorithm is a *post-processing* method that takes the original (biased) top-k recommendations as the input. It replaces the items of high coverage with those of low coverage to satisfy (α, β)-fairness, while minimizing the accuracy loss due to the replacement.

Next, we explain the details of both ER and GS algorithms respectively.

5.1 Embedding-based Re-ranking (ER)

At high level, the *embedding-based re-ranking* (ER) algorithm takes the user/item embedding of the original graph as the input, and generates the top-k recommendations $\mathbf{R}$ that satisfy (α, β)-fairness. Specifically, given a graph $\mathbf{G}$ that consists of n users $\mathbf{U}$ and m items $\mathbf{I}$, and a recommendation algorithm $\mathcal{A}$ that outputs the embedding of $\mathbf{U}$ and $\mathbf{I}$ (denoted as $E(\mathbf{U})$ and $E(\mathbf{I})$), the ER algorithm outputs a ranking policy $\pi = \{H_\theta(u) | \forall u \in \mathbf{U}\}$, where $H_\theta(u) = [H_\theta(u, i_1), \ldots, H_\theta(u, i_k)]$ is the top-k items for user u of the highest probabilities learned from $E(\mathbf{U})$ and $E(\mathbf{I})$ with the model parameters θ. We realize (α, β)-fairness as a constraint of learning the ranking policy π.

In particular, given a set of all possible rankings Π for all users, we consider stochastic ranking functions $\pi \in \Pi$, where $\pi(r|u)$ is a distribution over the rankings r for user u. The goal is to find a ranking policy π^* that maximizes the expected accuracy of π over all users $\mathbf{U}$:

$$\pi^* = arg\,max_{\pi \in \Pi} \mathbb{E}[Acc(\pi|\mathbf{U})] \tag{3}$$

Since we consider all users in $\mathbf{U}$, we can estimate the expectations with their empirical counterparts. Then the empirical analog of the optimization problem becomes

$$\pi^* = arg\,max_{\pi \in \Pi} \frac{1}{|\mathbf{U}|} \sum_{\forall u \in \mathbf{U}} Acc(\pi|u), \tag{4}$$

where the accuracy of a stochastic ranking policy π for a user $u \in \mathbf{U}$, denoted as $Acc(\pi|u)$, is defined as the expectation of an accuracy metric Δ over π:

$$Acc(\pi|u) = \mathbb{E}_{r \sim \pi(r|u)} \Delta(r). \tag{5}$$

There are several choices for the accuracy metric Δ such as DCG, NDCG, and average rank, to name a few.

Ranking with fairness constraint. We equip the requirement of individual item-fairness as a constraint with the learning problem. We use $B(\pi|\mathbf{U})$ to denote the violation of fairness (i.e., coverage disparity) of the ranking π for all users in $\mathbf{U}$. In particular, $B(\pi|\mathbf{U})$ is measured as follows:

$$B(\pi|\mathbf{U}) = \frac{1}{|D|} \sum_{\forall (i,i') \in D} max\{\big|CD(i, i')\big| - \beta, 0\}, \tag{6}$$

where D denotes the set of all α-similar item pairs, and $CD(i, i')$ denotes the coverage disparity between items i and i' (Eqn. 2) under the ranking policy π, and β is the parameter for (α, β)-fairness. Intuitively, $B(\pi|\mathbf{U}) = 0$ indicates that the β-coverage disparity requirement is met.

We then formulate the objective of fair ranking as the following:

$$\pi^* = arg\,max_{\pi \in \Pi} \frac{1}{|\mathbf{U}|} \sum_{\forall u \in \mathbf{U}} Acc(\pi|u)\ s.t.\ B(\pi|\mathbf{U}) = 0 \tag{7}$$

We resort to Lagrange multiplier and prepare the following formula to approximate Equation (7):

$$\pi^* = arg\,max_{\pi \in \Pi} \frac{1}{|\mathbf{U}|} \sum_{\forall u \in \mathbf{U}} Acc(\pi|u) - \lambda(B(\pi|\mathbf{U})), \tag{8}$$

where λ is the parameter that steers the accuracy/fairness trade-off. We follow the Plackett-Luce model [Cao et al. 2007; J. I. (Ed.). 1995] to model the distribution $\pi(r|u)$ over the top-k rankings r for user u.

$$\pi_\theta(r|u) = \prod_{j=1}^{k} \frac{exp(H_\theta^{r(j)}(u, i)))}{\sum_{\ell=j}^{k} exp(H_\theta^{r(\ell)}(u, i)))}, \tag{9}$$

where $\frac{exp(H_\theta^{r(j)}(u,i)))}{\sum_{\ell=j}^{k} exp(H_\theta^{r(\ell)}(u,i)))}$ returns the probability output by H_θ that item i ranked at the ℓ-th position in the top-k recommendations to user u.

Based on the stochastic ranking distributions, the optimization problem (Eqn. (4)) can be processed by using stochastic gradient descent (SGD). We define two gradients: *accuracy gradient* and *fairness gradient*, which are discussed below.

Accuracy gradient. The accuracy of a policy π_θ is defined as $\frac{1}{|\mathbf{U}|} \sum_{\forall u \in \mathbf{U}} Acc(\pi_\theta|u)$. As the space of ranking is exponential in the

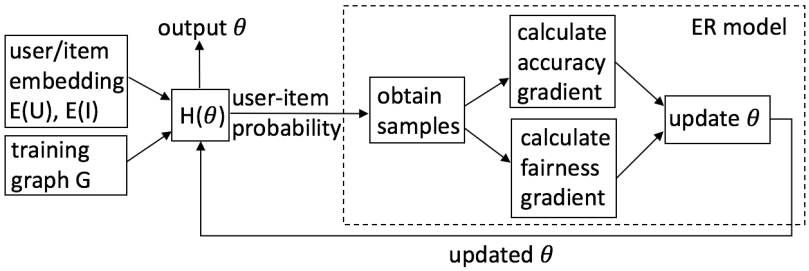

Figure 1: The diagram of ER algorithm.

Algorithm 1: Embedding-based Re-ranking (ER)

Input: Embedding of users and items $E(\mathbf{U})$ and $E(\mathbf{I})$, learning rate η.
Output: Model parameters θ

Initialize H_θ with $E(\mathbf{U})$ and $E(\mathbf{I})$ by following [He et al. 2017];
while *not yet converged* **do**
  Calculate the probability for each user-item pair by using $H(\theta)$;
  Calculate probability distribution (Eqn. (9));
  Compute gradient $\nabla\theta$ (Eqn (10), (11), (12));
  Update θ: $\theta = \theta + \eta \cdot \nabla\theta$;
end
Return θ

number of items and users when calculating gradients of θ in Eqn. 9, we use sampling in the log-derivative strategy [Williams 1992] and calculate the accuracy gradient $\nabla\theta_A$ as follows:

$$\nabla\theta_A = \frac{\partial}{\partial\theta}\frac{1}{U}\sum_{u=1}^{U} Acc(\pi_\theta|u) = \frac{1}{U}\sum_{u=1}^{U}\mathbb{E}_{r\sim\pi_\theta(r|u)}\frac{\partial}{\partial\theta}log(\pi_\theta(r|u))\Delta(r), \tag{10}$$

Fairness gradient. We compute the fairness gradient $\nabla\theta_F$ as following:

$$\begin{aligned}\nabla\theta_F &= \frac{\partial}{\partial\theta}\frac{1}{|D|}\sum_{\forall(i,i')\in D} 1[D(\pi_{\theta,i,i'}) > 0]\cdot D(\pi_{\theta,i,i'}) \\ &= \frac{1}{|D|}\sum_{\forall(i,i')\in D} 1[D(\pi_{\theta,i,i'}) > 0]\mathbb{E}_{r\sim\pi_\theta(r|u)}D(\pi_{\theta,i,i'})\frac{\partial}{\partial\theta}log(\pi_\theta(r|u)),\end{aligned} \tag{11}$$

where D denotes the set of all α-similar item pairs, and $D(\pi_{\theta,i,i'}) = \left|NCov_{\pi,i} - NCov_{\pi,i'}\right| - \beta$. Intuitively, $D(\pi_{\theta,i,i'}) > 0$ indicates the violation of β-coverage disparity. $1[D(\pi_{\theta,i,i'}) > 0]$ returns 1 if $D(\pi_{\theta,i,i'}) > 0$; otherwise it returns 0.

Update on both gradients. With both gradients $\nabla\theta_A$ and $\nabla\theta_F$, the gradient of θ is computed as:

$$\nabla\theta = \nabla\theta_A - \lambda\nabla\theta_F. \tag{12}$$

Algorithm. Based on all the discussions above, we design the ER algorithm to solve the optimization problem in Equation (4). The diagram of the ER method is shown in Figure 1. First, we initialize H_θ with $E(\mathbf{U})$ and $E(\mathbf{I})$ by using the multi-layer neural architecture [He et al. 2017]. During model training, we first calculate user-item probability generated by H_θ. Based on the user-item probability, we apply the Monte-Carlo sampling of possible permutations for each user, and calculate the probability distribution for each permutation by using the Plackett-Luce model. Next, we compute the gradient $\nabla\theta$ (Eqn. (12)). Finally, we update θ with $\nabla\theta$. We repeat these steps until we reach the convergence. In this paper, we set the termination condition as that the accuracy (Eqn. (8)) improves less than 1% compared with the last iteration for five continuous iterations. The the output of the ER method may not satisfy (α, β)-fairness.

5.2 Greedy Substitution (GS)

An alternative mitigation solution is to *post-process* the original biased recommendations to make them satisfy (α, β)-fairness. In this section, we present our greedy substation (GS) algorithm as one such post-processing solution.

Given a graph $\mathbf{G}$ that consists of n users $\mathbf{U}$ and m items $\mathbf{I}$, and a recommendation algorithm $\mathcal{A}$, the goal of $\mathcal{A}$ is to predict the preferences of users to specific items. We use $p(u, i)$ to indicate the probability predicted by $\mathcal{A}$ that user u will be interested in item i. Each user is recommended with the items of the top-k highest probabilities. Let $\vec{R}_u = [R_{u,i_1}, \ldots, R_{u,i_k}]$ be the top-k recommendation list returned by $\mathcal{A}$ for the user $u \in \mathbf{U}$, where $R_{u,i_\ell} \in \vec{R}_u (1 \le \ell \le k)$ consists of the recommended item i_ℓ and the probability $p(u, i_\ell)$ predicted by $\mathcal{A}$ that user u will be interested in the item i_ℓ. We measure the *accuracy* of $\vec{R}_u$, denoted as $UT(\vec{R}_u)$, as follows:

$$UT(\vec{R}_u) = \sum_{\ell=1}^{k} p(u, i_\ell)$$

Intuitively, the accuracy of $\vec{R}_u$ is measured as the sum of the probability of all items in the top-k list. Higher accuracy indicates that user u are more likely to be interested in the items in $\vec{R}_u$.

Let $\mathbf{R} = \{\vec{R}_u | u \in \mathbf{U}\}$ be the collections of original recommendation lists for all users. Then the problem of enforcing (α, β)-fairness on $\mathbf{R}$ can be formulated as an optimization problem:

$$\max_{\vec{R}_u \in \mathbf{R}} \sum_{\forall u \in \mathbf{U}} UT(\vec{R}_u), \quad \text{s.t. } \forall i, j \in \mathbf{R}\ i \overset{\alpha}{\approx} j,\ CD(i, j) \le \beta. \tag{13}$$

In other words, the goal is to modify $\mathbf{R}$ so that it can satisfy the fairness requirement while maximizing the accuracy of the recommendations.

The optimization problem in (13) is NP-hard [Koutsopoulos and Halkidi 2018]. A naive solution is to first identify all item pairs that violate the fairness requirement, then enumerate all possible recommendations of these items to find the one of the maximum accuracy. Apparently, this naive solution is computational expensive. Therefore, we design a *greedy substitution* (GS) algorithm to pick the items in the recommendations to be mitigated in an efficient

way. Consider the recommendations $\mathbf{R} = \{\vec{R}_u | \forall u \in \mathbf{U}\}$, where $\vec{R}_u = [R_{u,i_1}, \ldots, R_{u,i_k}]$ is the top-k recommendation list for the user u. For any two items $i_1, i_2 \in \mathbf{R}$ that are α-similar but violate the β-coverage disparity requirement, assume i_1 is the item with higher coverage. Let $\mathbf{R}^{i_1} \subseteq \mathbf{R}$ be the recommendations that contain item i_1. The GS algorithm picks a subset of recommendations $\mathbf{R}' \subseteq \mathbf{R}^{i_1}$, and replaces i_1 in $\mathbf{R}'$ with other items. Intuitively, all items that do not appear in the recommendation list of i_1 are the candidates for replacement with i_1. However, considering all these items can be computationally expensive. Thus GS algorithm picks the candidates in a greedy fashion: it picks those items that are α-similar to i_1 and of low coverage. Picking similar items is to minimize the accuracy loss of recommendations due to item substitution, while picking the items of low coverage for replacement is because picking items of high coverage will only continue increasing the coverage of those items, which does not help to mitigate popularity bias in these items.

In addition, the GS algorithm takes the accuracy of the recommendations into consideration when picking the items for replacement to meet the objective defined in (13). Specifically, we define the *cost* of replacing an item in a recommendation list as following. Let $\vec{R}_u$ be the original recommendation list for user u, and $\vec{R}_u{}'$ be the list after replacing item i with i' in $\vec{R}_u$. Then the cost $c(\vec{R}_u, \vec{R}_u{}')$ of changing $\vec{R}_u$ to $\vec{R}_u{}'$ is measured as $c(\vec{R}_u, \vec{R}_u{}') = |UT(\vec{R}_u{}') - UT(\vec{R}_u)|$. Since $\vec{R}_u{}'$ and $\vec{R}_u$ only differ at one item, we can easily infer that

$$c(\vec{R}_u, \vec{R}_u{}') = |p(u,i) - p(u,i')|.$$

In other words, the cost of replacing i with i' in $\vec{R}_u$ is equal to the difference in the probability that user u will be interested in i and i'. Therefore, the objective function in Formula 13 is equivalent to

$$\min_{\vec{R}_u{}'} \sum_{\forall u \in \mathbf{U}} c(\vec{R}_u, \vec{R}_u{}'), \tag{14}$$

Based on the above reasoning, we design the GS algorithm as a three-step process: *Step 1*: generate α-similar item clusters; *Step 2*: pick replacement items from their clusters to meet β-coverage disparity; and *Step 3*: item replacement and coverage updates. Next, we explain the details of these steps. Algorithm 2 shows the pseudo code of GS algorithm.

Step 1. Generate α-similar item clusters. To speed up the search for similar items during replacement, first, the GS algorithm partitions all items in $\mathbf{I}$ into a number of clusters, so that the items in each cluster are pairwise α-similar. We implement this clustering step by initializing each similar pair as a cluster. Then we merge any two clusters into one if they have at least one item in common and all items are pairwise α-similar. We repeat the merge step until there is no further merge can be performed. After clustering, the GS algorithm simply can pick those items in the same cluster for substitution. Therefore, we mainly focus on a cluster of items in the following discussions.

Step 2. Pick items for replacement to meet β-coverage disparity. Since all items in the same cluster are α-similar, the next step of the GS algorithm is to make all items in the same cluster to achieve β-coverage disparity. The basic idea is to replace items of high coverage with the items in the same cluster of low coverage. One challenge of realizing this strategy is to determine whether an item is of "high/low" coverage. To address this challenge, we calculate the lower- and upper-bound of item coverage of all items in a cluster that is guaranteed to make the whole cluster achieve (α, β)-individual item-fairness. In particular, consider a cluster of items C, let Cov_{avg}, Cov_{max}, Cov_{min} be the average, maximum, and minimum of the coverage of all items in C respectively. We have the following theorem.

Algorithm 2: Greedy substitution (GS) algorithm

Input: User-item graph $\mathbf{G}$, users $\mathbf{U}$, items $\mathbf{I}$, the user-item probability matrix P predicted by a biased recommendation algorithm $\mathcal{A}$, the top-k recommendations $\mathbf{R} = \{\vec{R}_u | u \in \mathbf{U}\}$ generated by $\mathcal{A}$.
Output: The modified top-k recommendations $\hat{\mathbf{R}}$ that satisfy (α, β)-fairness.

```
Partition I into clusters such at the items in each cluster are pairwise α-similar; // Step 1
for each cluster C do
    Calculate Cov_avg, Cov_max and Cov_min (i.e., average, maximum and minimum of item coverage) of items in C; // Step 2
    Cov^sup ← min{⌊Cov_avg + β · Cov_max⌋, Cov_max};
    Cov^inf ← Cov^sup − β · Cov_max;
    Generate C^+ ⊆ C that contains all items to be replaced;
    Generate C^− ⊆ C that contains all items as candidates for replacement;
    while C^− ≠ ∅ or C^+ ≠ ∅ do
        Find one item i ∈ C^+ and one item i' ∈ C^− such that argmin_{i,i',u} (|P(u,i) − P(u,i')|);
        Replace i in the recommendation list of user u with item i'; // Step 3
        Cov(i) = Cov(i) − 1, Cov(i') = Cov(i') + 1;
        Update Cov_max, Cov_min, Cov^sup, Cov^inf, C^+, C^−;
    end
end
Return R̂;
```

Theorem 5.1. *Given a set of items C which are pairwise α-similar, all items in C satisfy (α, β)-individual item fairness if the largest item coverage Cov_{max} in C satisfies the following condition:*

$$Cov_{max} \leq \lfloor Cov_{avg} + \beta \cdot max(Cov(i_1), Cov(i_2)) \rfloor, \forall i_1, i_2 \in C \tag{15}$$

We omit the proof of Theorem 5.1 due to the limited space and only include the proof sketch here. Consider the extreme case that there is only one item with coverage of Cov_{max}, while all other items are of the minimum coverage ($Cov_{max} - \beta \cdot M(i_1, i_2)$). To satisfy (α, β)- fairness, Equation (15) must hold. Following Theorem 5.1, since $max(Cov(i_1), Cov(i_2)) \leq Cov_{max}, \forall i_1, i_2 \in C$, the *upper-bound of item coverage* Cov^{sup} to satisfy β-coverage disparity is then set as: $Cov^{sup} = min\{\lfloor Cov_{avg} + \beta \cdot Cov_{max} \rfloor, Cov_{max}\}$. Following the requirement of β-coverage disparity, we can infer the *lower-bound of item coverage* Cov^{inf} to satisfy β-coverage disparity as: $Cov^{inf} = \lfloor Cov^{sup} - \beta \cdot Cov^{sup} \rfloor$. Based on the reasoning of the lower-bound and upper-bound of item coverage to satisfy β-coverage disparity, given an α-similar cluster C, all the items in C are assigned to one of the three sets in terms of its coverage:

- $C^+ = \{i|Cov(i) > Cov^{sup}, i \in C\}$;
- $C^- = \{i|Cov(i) < Cov^{inf}, i \in C\}$;
- $C_0 = C - C^+ - C^-$.

Intuitively, each item in C^+ will be substituted by an item in C^-. Before we start the substitution, we handle the special case that $C^+ = \emptyset$ or $C^- = \emptyset$. In particular, when $C^+ = \emptyset, C^+ \leftarrow C_0 \backslash \{i|Cov_i = Cov^{inf}\}$. Also when $C^- = \emptyset, C^- \leftarrow C_0 \backslash \{i|Cov_i = Cov^{inf}\}$.

Step 3: Item replacement and coverage updates. We substitute each item $i \in C^+$ with an item $i' \in C^-$. To maximize the accuracy of the recommendations, we find the item $i' \in C^-$ that is of the closest probability as i. Since each item is associated with multiple probabilities, with one for each user, we look for a user u and an item $i' \in C^-$ such that $|p(u,i) - p(u,i')|$ is the minimum, among all users that i is recommended to. After one item substitution, we update the coverage of items i and i', as well as $Cov_{max}, Cov_{min}, Cov^{sup}, Cov^{inf}, C^+, C^-$. The GS algorithm repeats Step 2 & 3 until there is no element in both C^+ and C^-, i.e., all items in the cluster C satisfy (α, β)-fairness.

6 EXPERIMENTS

In this section, we present the experimental results of our methods, and discuss these results. The original recommendation algorithms and GS algorithm are executed on NVIDIA DGX-2 (V100/32GB) GPU, and the ER algorithm is executed on Intel(R) Xeon(R) 12-core CPUs with 128 GB memory. All algorithms are implemented in Python [2].

6.1 Setup

Datasets. We use the Amazon e-commerce datasets collection [Ni et al. 2019]. In particular, we consider two categories in the datasets: *Cell Phone* and *Beauty*. *Beauty* dataset consists of 22,323 user nodes, 12,101 item nodes, and 37,205,822 edges, and *Cell phone* dataset consists 27,879 user nodes, 10,429 item nodes, and 37,403,062 edges. We omit the detailed description and graph statistics due to the limited space. For each category, we randomly select 70% for training, and treat the remaining 30% as for testing.

Recommendation algorithms. We use three state-of-the-art deep learning based recommendation algorithms: (1) **CKE** [Zhang et al. 2016] with its implementation[3]; (2) **PGPR** [Xian et al. 2019] with the implementation[4]; (3) and **KGAT** [Wang et al. 2019] with the implementation.[5] All the three algorithms output the node embeddings of the original graph; the embeddings can be used to generate the recommendations.

Fairness setting. The setting of α and β for (α, β)-fairness is shown in Table 1. The three α values are determined by controlling the percentage of item pairs evaluated as similar on each dataset, they lead to 0.1%, 0.5%, and 1% of such similar item pairs respectively. The β values are determined by empirical analysis of coverage disparity of the three recommendation algorithms (CKE, PGPR, KGAT) on the datasets, where the average coverage disparity of these three algorithms is within 0.1906 ± 0.0263. For each (α, β) setting, we tried a number of λ values (Eqn. 12) and pick the one that delivered the best performance.

Parameter	Amazon beauty dataset	Amazon cellphone dataset
α	0.2, 0.26, 03	0.25, 0.3, 0.35
β	0.01, 0.05, 0.1	0.01, 0.05, 0.1

Table 1: Settings of α,β values.

Evaluation metrics of recommendation accuracy. We use three metrics: *normalized discounted cumulative gain* (NDCG), *precision*, and *recall* of the top-10 recommendations. All the results are computed as the average over all users.

Mitigation methods for comparison. We consider the following two baseline methods:

- Baseline method (**MC**): We implement the mitigation method in the literature [Koutsopoulos and Halkidi 2018; Patro et al. 2020] that enforces "minimum item coverage" requirement on the recommendations. In particular, given the original (biased) recommendations and an item coverage threshold, we randomly select an item i whose coverage is below the threshold, and replace it with the item that is of the highest coverage that has not been replaced yet. We repeat the substitution until the coverage of all items is above the threshold. We name the baseline mitigation method as **MC**. We use 10 and 13 as item coverage threshold for Beauty and Cell phone datasets respectively. We pick these values as they are approximately half of the average item coverage of the recommendations.
- Hybrid method (**HB**): We first apply ER to generate the recommendation list and user-item probability matrix, which are then used as the input of the GS method. The output of the GS method is considered as the final recommendations.

Training setting. We tested the batches of size 128, 256, 512, and 1024, and pick the batch size of 256 in the reported experiments as it delivers the best performance. The learning rate η is set as 0.001. The size of the three hidden layers is set as 128, 64, 32 respectively. We set the dimension in the embedding of PGPR as 100, and 64 for KGAT and CKE. We set the Monte-Carlo sampling size as 30.

6.2 Effectiveness of Bias Mitigation

We measure the effects of bias mitigation on the performance of three recommendation algorithms (PGPR, KGAT, CKE). In particular, we measure the bias B as the percentage of α-similar pairs of items that violate the β-coverage parity requirement. Furthermore, we measure *bias reduction* as follows:

$$\text{Bias reduction} = \frac{B_{before} - B_{after}}{B_{before}}, \tag{16}$$

where B_{before} and B_{after} indicate the bias of the recommendations before and after mitigation. Intuitively, larger Bias reduction value indicates that the bias mitigation method is more effective.

Figure 2 shows the bias reduction by the four mitigation methods under various α and β values on Amazon Beauty dataset. The results on Amazon Cellphone dataset are similar and are omitted due to the limited space. The results show non-negligible bias reduction by the four mitigation methods, in the range [12.5%, 93.6%]. Second, we observe that the bias reduction by ER is significantly higher than MC when CKE is used as the recommendation algorithm, and when $\alpha \geq 0.26$ for PGPR recommendation algorithm. On the other hand, when KGAT is used as the recommendation algorithm,

[2]Our code is available at https://github.com/xiulingwang33/Individual-item-fairness
[3]CKE: https://github.com/xiangwang1223/knowledge_graph_attention_network implemented by the authors of KGAT [Wang et al. 2019].
[4]PRPG: https://github.com/orcax/PGPR.
[5]KGAT: https://github.com/xiangwang1223/knowledge_graph_attention_network.

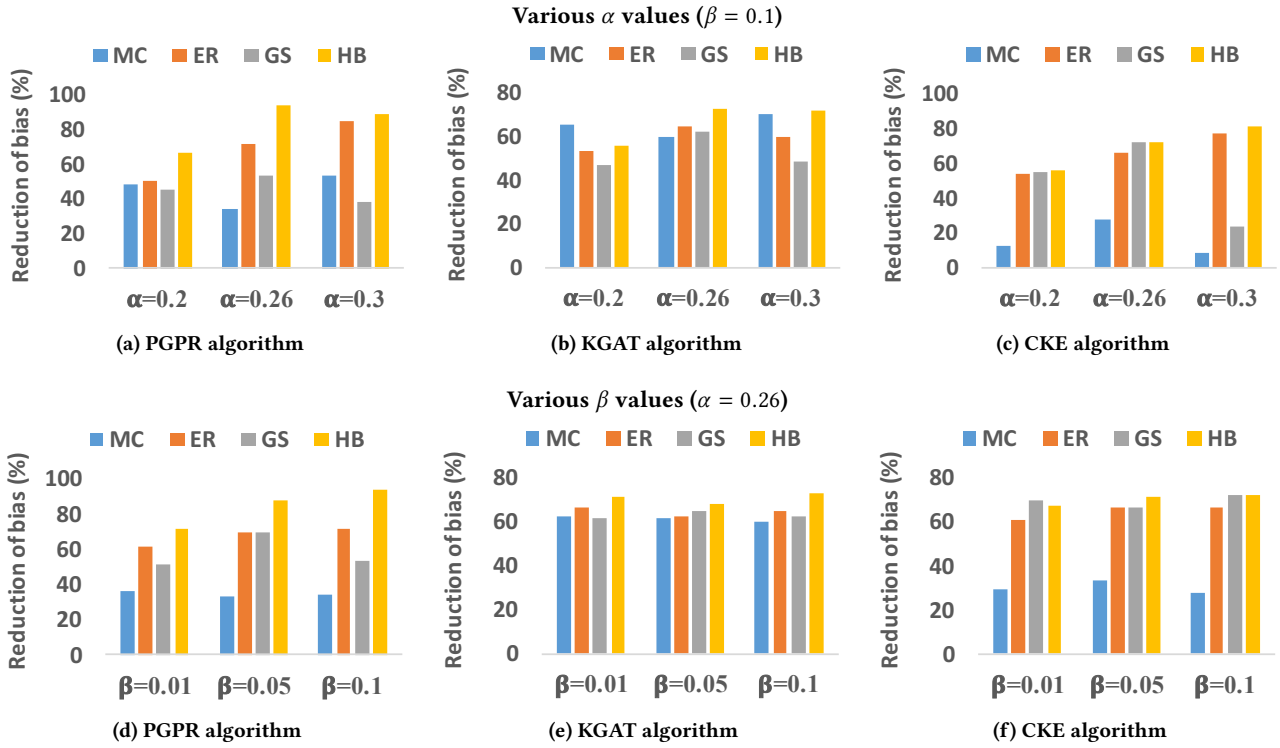

Figure 2: Bias reduction by MC, GS, ER and HB mitigation methods (Amazon Beauty dataset)

the three mitigation methods (MC, ER, GS) have comparable bias reduction. Third, regarding ER versus GS, ER either outperforms GS (for most of the settings of PGPR and KGAT recommendation algorithms and $\alpha = 0.3, \beta = 0.1$ for CKE algorithm) or has similar performance as GS. In another word, ER is more effective than GS in bias reduction, possibly due to the fact that as an in-processing method, ER interacts more closely with the recommendation models than GS. Meanwhile, as expected, the HB approach always has the largest bias reduction among all four mitigation methods. We also observe that the bias reduction of ER and GS does not change consistently with the growth of α and β values for both algorithms. This is indeed expected as the amounts of bias reduction rely on neither α nor β values. Smaller α values will only lead to more dissimilar pairs to be treated, while smaller β values will incur more item replacement.

6.3 Accuracy Loss by Bias Mitigation

To evaluate the impact of bias mitigation on recommendation accuracy, we measure the accuracy loss as

$$\text{Accuracy loss} = \frac{Acc - Acc'}{Acc},$$

where Acc and Acc' indicate the accuracy (NDCG@10, Precision@10, Recall@10) of the recommendations before and after mitigation.

Table 2 shows the results of accuracy loss by the four mitigation algorithms. The results on Amazon cellphone dataset are similar and thus are omitted due to the limited space. All the mitigation methods witness the accuracy loss to some extent as the price to be paid for fairness. First, among these four methods, MC method always witnesses the largest accuracy loss. Indeed, its accuracy loss is significant (in the range of [31.53%, 40.89%]). This demonstrates the weakness of the minimum-coverage bias mitigation method in the literature [Koutsopoulos and Halkidi 2018; Patro et al. 2020]. Second, when comparing GS with ER in terms of accuracy loss, there is no winner between these two methods. For example, GS outperforms ER when KGAT is used as the recommendation algorithm, while ER outperforms GS in some settings (e.g., $\alpha = 0.2$ and $\beta = 0.01$) when PGPR makes the recommendations. Third, HB always brings higher accuracy loss that of GS and ER, as it has to pay the price of accuracy loss for both GS and ER methods.

6.4 Trade-off between Fairness and Recommendation Accuracy

The previous results have shown that the four bias mitigation methods can mitigate bias effectively, but suffer from accuracy loss in recommendations. To perform a quantitative comparison of these methods in terms of how they address the the *trade-off* between fairness and accuracy. We measure the *trade-off* between fairness and accuracy as:

$$\text{Tradeoff T} = \frac{\text{Bias reduction}}{\text{Accuracy loss}}$$

Intuitively, larger T value indicates higher reduction of bias and smaller accuracy loss and thus the better trade-off.

Figure 3 shows the results of trade-off between fairness and accuracy (NDCG@10) by the four mitigation methods under various α and β values on Amazon Beauty dataset. We normalize the trade-off values across all four mitigation methods to make them in the range [0, 1]. The results on Amazon Cellphone dataset are similar and thus are omitted. We have the following observations. First, unsurprisingly, the trade-off exists for all the four bias mitigation methods. Second, all our methods (ER, GS, HB) have better trade-off

Table 2: Accuracy loss (%) of recommendations by GS , ER, HB and MC methods on Amazon Beauty dataset. The smallest accuracy reduction per accuracy metric per setting of the four mitigation methods is highlighted in bold.

α			**0.2**			**0.26**			**0.3**		
β			0.01	0.05	0.1	0.01	0.05	0.1	0.01	0.05	0.1
PGPR	NDCG@10	GS	28.23	**0.48**	**0.02**	21.85	**8.60**	**5.82**	**17.53**	**14.52**	**11.94**
		ER	**21.56**	11.54	5.69	20.25	11.49	10.63	20.73	16.96	15.88
		HB	35.87	29.42	22.29	36.07	32.13	21.74	35.39	28.90	22.14
		MC	32.54								
	Precision@10	GS	25.28	**0.01**	**0.56**	20.72	**7.87**	**5.06**	24.27	**13.48**	**11.24**
		ER	**19.49**	9.23	11.79	**18.97**	8.72	7.18	18.46	14.87	15.38
		HB	26.15	25.64	25.64	**18.97**	22.56	18.97	**17.44**	17.95	17.44
		MC	32.08								
	Recall@10	GS	26.13	**3.47**	**0.69**	24.85	**8.55**	**6.24**	21.45	14.80	**12.37**
		ER	**21.56**	11.54	13.69	21.19	9.16	7.28	**18.10**	**14.57**	14.79
		HB	35.87	29.42	22.29	**20.65**	23.95	19.76	22.96	22.08	21.63
		MC	31.53								
KGAT	NDCG@10	GS	**5.78**	**5.78**	**5.78**	**9.35**	**8.30**	**6.85**	**15.46**	14.21	12.45
		ER	11.35	13.51	11.35	20.12	22.47	23.08	20.09	**12.42**	**10.71**
		HB	26.14	25.98	31.72	25.95	29.07	31.36	26.67	29.75	29.30
		MC	39.35								
	Precision@10	GS	**0.58**	**0.58**	**0.58**	**8.67**	**7.51**	**5.78**	**14.45**	13.29	11.56
		ER	11.35	13,51	11.35	14.05	15.13	15.13	16.76	**13.51**	**11.35**
		HB	17.84	17.30	17.84	17.30	20.00	21.08	22.70	21.62	20.00
		MC	35.14								
	Recall@10	GS	**2.29**	**1.93**	**1.69**	**9.88**	**8.80**	**8.07**	**16.14**	14.58	12.77
		ER	11.08	13.56	10.38	12.97	15.68	15.45	16.74	**12.15**	**9.31**
		HB	15.57	15.33	15.57	15.33	17.92	18.87	20.52	19.81	18.16
		MC	37.72								
CKE	NDCG@10	GS	**8.48**	6.96	**6.87**	**6.98**	**6.15**	**5.89**	**11.11**	**9.74**	**8.57**
		ER	8.53	**5.90**	9.67	14.69	8.31	11.33	14.26	11.92	12.01
		HB	20.67	23.10	24.83	20.68	22.79	23.98	20.82	20.78	23.51
		MC	40.32								
	Precision@10	GS	**1.27**	**1.27**	**1.26**	**7.59**	**6.96**	**6.33**	**11.03**	**10.76**	**3.16**
		ER	7.65	5.61	7.14	11.73	7.65	9.18	11.73	12.75	11.22
		HB	19.90	19.90	20.41	22.45	21.94	22.96	23.47	22.96	22.96
		MC	40.89								
	Recall@10	GS	**0.98**	**0.84**	**0.70**	**7.74**	**7.03**	**6.89**	**12.38**	**10.98**	**10.13**
		ER	8.40	5.56	8.18	9.71	8.07	9.49	12.98	12.98	12.00
		HB	22.79	22.90	23.02	25.19	24.97	23.57	26.83	26.06	25.41
		MC	38.82								

than the baseline MC. This shows the effectiveness of our methods in addressing the trade-off between recommendation accuracy and fairness. Third, in most of the settings, ER has the best trade-off among the four methods. The only exception is when $\alpha = 0.2$ and $\beta = 0.1$ for PGPR recommendation algorithm, where GS method has the best trade-off. We studied the reason behind this observation and found that the bias reduction and accuracy loss for GS under this setting is 44.86% and 0.018% respectively, while they are 50.34% and 5.69% for ER. Thus the trade-off of GS is much higher than that of ER. Thus, ER is more suitable than the other three alternative mitigation solutions.

7 DISCUSSIONS

Alternative fairness definition. The (α, β)-fairness sets the same α and β values for all item pairs, which may be too rigid as different item pairs may have different degree of bias. An alternative fairness definition is to follow the Lipschitz requirement [Dwork et al. 2012]: the recommendations of two items i_1 and i_2 is *L-Lipschitz* fair if $CD(i_1, i_2) \leq L \times d(i_1, i_2))$ for some given L, where $dis()$ and $CD()$ are measured by Equations 1 and 2 respectively. We will investigate how to equip the recommender systems with Lipstchitz fairness in the future.

In-processing vs. post-processing bias mitigation. As a post-processing method, the GS method is model-agnostic and can be adapted to any recommender system. On the other hand, as an in-processing method, the ER method relies on the models that the recommender systems use. Since ER interacts more closely with the recommendation models, ER outperforms GS in terms of the effect of bias mitigation. However, ER is less generally applicable than GS.

Personalized item similarity. In recommender systems, different users have different preferences and needs. Therefore, how the users evaluate the accuracy of the recommended items should be personalized for each user [Kim et al. 2011]. This raises the concern that the notion of embedding based item similarity may need to be personalized too. This issue can be addressed by using personalized

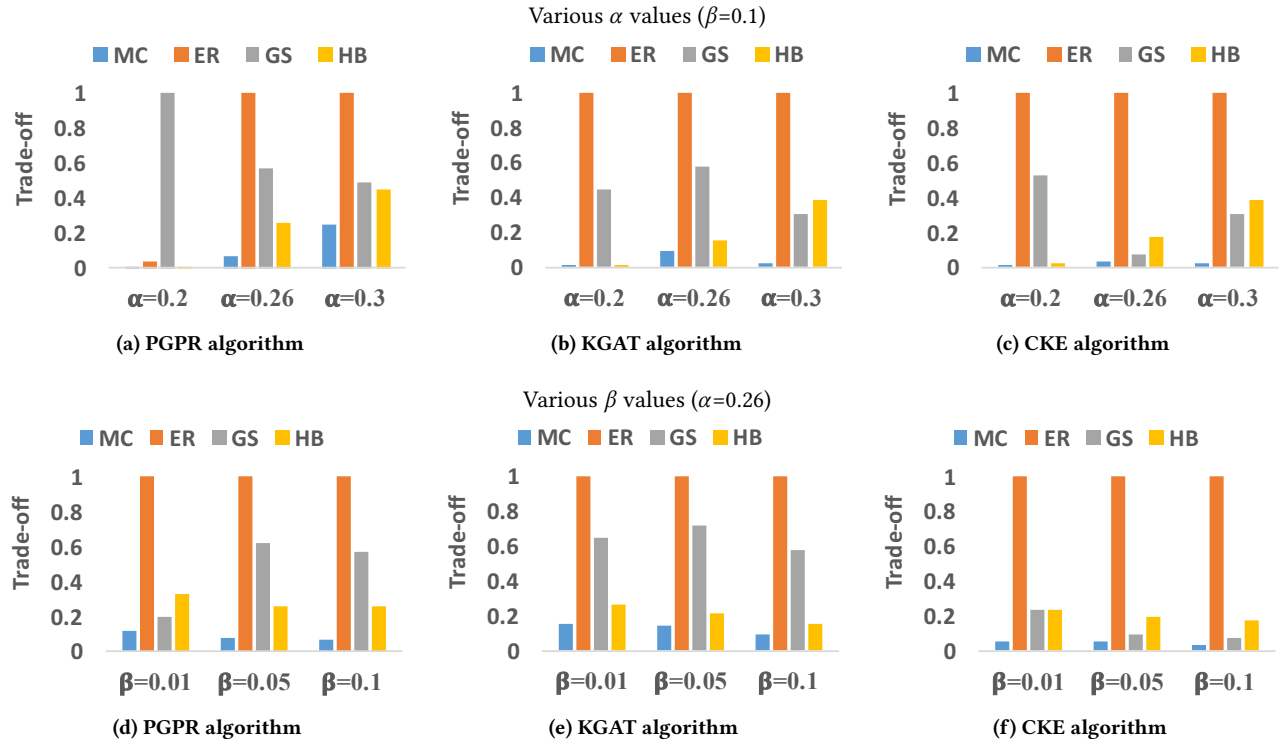

Figure 3: Trade-off of fairness and accuracy of MC, GS, ER and HB mitigation methods (Amazon Beauty dataset)

user and item embeddings [Nguyen and Takasu 2018; Ruan et al. 2021; Wang et al. 2021] that incorporate users implicit shopping behaviors. Then the item similarity can be measured as the distance between the personalized embeddings.

Impact of two-sided fairness. We only consider item fairness so far. When user fairness is also required, ensuring fairness on the item side may affect the fairness on the user side, and vice versa. The trade-off indeed exists between the fairness objectives at both sides. Intuitively, realizing fairness objectives simultaneously at both sides can be considered as a multi-objective optimization problem. The challenge is to study the dominance among all fairness solutions, and identify the Pareto optimal one.

8 CONCLUSION

In this paper, we focus on item-side individual fairness in recommender systems. First, we formalize a new notion of individual fairness named (α, β)-*fairness*, which requires similar items (where item similarity is controlled by α) must receive similar coverage in recommendations (where coverage similarity is controlled by β). Next, we design two bias mitigation methods named *embedding-based re-ranking* (ER) and *greedy substitution* (GS) that take different types of inputs for bias mitigation over recommendations. Our experiments demonstrate that both ER and GS methods can address the trade-off between fairness and accuracy of recommendations. Furthermore, ER outperforms both GS and the existing solution [Koutsopoulos and Halkidi 2018; Patro et al. 2020] in terms of the trade-off both fairness and recommendation accuracy.

For the future work, first, we will explore how to realize multiple fairness objectives at both user and item sides simultaneously while minimizing accuracy loss of recommendations. Second, we will optimize the ER algorithm by partially updating model parameters for particular target item pairs (e.g., similar item pairs only).

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This project was supported by the National Science Foundation (#CNS-2135988). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

Himan Abdollahpouri and Robin Burke. 2019. Multi-stakeholder recommendation and its connection to multi-sided fairness. *arXiv preprint arXiv:1907.13158* (2019).

Himan Abdollahpouri, R. Burke, and B. Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. *Proceedings of the ACM Conference on Recommender Systems* (2017).

Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-ranking. (2019). arXiv:1901.07555 [cs.IR]

Gediminas Adomavicius and YoungOk Kwon. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2011), 896–911.

Pablo Badilla, Felipe Bravo-Marquez, and Jorge Pérez. 2020. WEFE: The Word Embeddings Fairness Evaluation Framework. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 430–436.

Kinjal Basu, Cyrus DiCiccio, Heloise Logan, and Noureddine El Karoui. 2020. A Framework for Fairness in Two-Sided Marketplaces. *arXiv preprint arXiv:2006.12756* (2020).

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. [n.d.]. Fairness in Recommendation Ranking through Pairwise Comparisons. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* ([n. d.]).

Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *International acm sigir conference on research & development in information retrieval*. 405–414.

Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building classifiers with independency constraints. In *International Conference on Data Mining Workshops*. 13–18.

Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the International Conference on Machine Learning*. 129–136.

L. Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. 2019. Controlling Polarization in Personalization: An Algorithmic Framework. In *Proceedings of the conference on fairness, accountability, and transparency (FaccT)*. 160–169.

Abhijnan Chakraborty, Aniko Hannak, Asia J Biega, and Krishna Gummadi. 2017. Fair sharing for sharing economy platforms. In *Fairness, Accountability and Transparency in Recommender Systems-Workshop on Responsible Recommendation*.

Alexandra Chouldechova and Aaron Roth. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810* (2018).

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of innovations in theoretical computer science conference*. 214–226.

Bora Edizel, F. Bonchi, S. Hajian, A. Panisson, and Tamir Tassa. 2019. FaiRecSys: mitigating algorithmic bias in recommender systems. *International Journal of Data Science and Analytics* (2019), 1–17.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 259–268.

Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. 2020. Fairness-Aware Explainable Recommendation over Knowledge Graphs. *arXiv preprint arXiv:2006.02046* (2020).

Yang Gao, Yi-Fan Li, Yu Lin, Hang Gao, and Latifur Khan. 2020. Deep learning on knowledge graph for recommender system: A survey. *arXiv preprint arXiv:2004.00387* (2020).

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. http://arxiv.org/abs/1708.05031

Christina Ilvento. 2020. Metric Learning for Individual Fairness. In *1st Symposium on Foundations of Responsible Computing (FORC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Marden J. I. (Ed.). 1995. *Analyzing and modeling rank data.* London: Chapman and Hall.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 35–50.

Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *International Conference on Data Mining Workshops*. 643–650.

Heung-Nam Kim, Inay Ha, Kee-Sung Lee, Geun-Sik Jo, and Abdulmotaleb El-Saddik. 2011. Collaborative user modeling for enhanced content filtering in recommender systems. *Decision Support Systems* 51, 4 (2011), 772–781.

I. Koutsopoulos and M. Halkidi. 2018. Efficient and Fair Item Coverage in Recommender Systems. *International conference on Dependable, Autonomic and Secure Computing* (2018), 912–918.

Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021*. 624–632.

Chan Liu, Lun Li, Xiaolu Yao, and Lin Tang. 2019. A survey of recommendation algorithms based on knowledge graph embedding. In *IEEE International Conference on Computer Science and Educational Informatization*. 168–171.

Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020). https://doi.org/10.1145/3397271.3401100

ThaiBinh Nguyen and Atsuhiro Takasu. 2018. NPE: Neural Personalized Embedding for Collaborative Filtering. arXiv:1805.06563 [cs.IR]

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In *Empirical Methods in Natural Language Processing*.

Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2020. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of The Web Conference*. 1194–1204.

Gourab K Patro, Abhijnan Chakraborty, Niloy Ganguly, and Krishna P. Gummadi. 2019. Incremental Fairness in Two-Sided Market Platforms: On Smoothly Updating Recommendations. arXiv:1909.10005 [cs.SI]

Qunsheng Ruan, Yiru Zhang, Yuhui Zheng, Yingdong Wang, Qingfeng Wu, Tianqi Ma, and Xiling Liu. 2021. Recommendation Model Based on a Heterogeneous Personalized Spacey Embedding Method. *Symmetry* 13, 2 (2021). https://www.mdpi.com/2073-8994/13/2/290

Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.

Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 32. 5426–5436.

Harald Steck. 2018. Calibrated recommendations. In *Proceedings of ACM conference on recommender systems*. 154–162.

Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. 2021. Personalized Embedding-based e-Commerce Recommendations at eBay. arXiv:2102.06156 [cs.IR]

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph A ention Network for Recommendation. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of ACM conference on recommender systems*. 107–115.

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

Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *Proceedings of the Conference on Information and Knowledge Management CIKM*. 1569–1578.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.