Model-Based Debiasing for Groupwise Item Fairness

Yikai Zhang[1], Yeshaya Adler[1], Mina Dalirrooyfard[1], Teresa Datta[2], Volodymyr Volchenko[1], John P. Dickerson[2], Yuriy Nevmyvaka[1]

[1]Morgan Stanley [2]Arthur

Abstract

Recommendation systems are an essential tool for presenting items to users, and hence they are subject to many fairness considerations for users and items alike. Many post processing algorithms exist to handle unfairness in recommender systems; however, they can be very inefficient and not suitable to be used in real time as they need the whole data set to be able to calibrate the recommender system's output. We develop the first model-based group-wise item fairness post-processing algorithm for recommendation systems using a neural network architecture which learns in a data-dependent fashion. Our model adapts and refines itself based on the underlying data without significantly compromising the original utility during the training phase of the recommender system. These performance guarantees are ensured by VC theory and stochastic approximate analysis and we showcase our method's capabilities through experiments on synthetic data.

Introduction

The application of recommendation systems is undeniably one of the most critical domains in modern machine learning. These systems have wide-ranging applications across various sectors, including social networks, e-commerce, and finance (Li et al. 2023). Recommendation systems play a crucial role in delivering information to users, serving as the primary gateway for content discovery. Ensuring fairness is essential for these systems. To illustrate this, let's consider two scenarios: one highlighting user unfairness, and another emphasizing item unfairness.

In the first scenario, imagine a recommendation system that consistently suggests high-quality, popular content to a particular demographic, while neglecting the interests and preferences of another demographic. In this case, users from the underrepresented demographic may experience unfairness as they miss out on content tailored to their tastes and needs.[1] For the second scenario, consider a recommendation system that promotes a specific type of content, such as certain political news, disproportionately more than other content categories. This could lead to an unfair advantage for the promoted content, potentially influencing users' opinions and perspectives in a biased manner. In high-stakes domains, the consequences of item unfairness can be actively harmful. In the U.S. financial space, a backdrop of lending discrimination has meant that Black and Latino borrowers have historically been disproportionately recommended higher-cost, higher-risk mortgages (Steil et al. 2017). In hiring domains, candidate recommendation algorithms have been found to discriminate against historically marginalized individuals–for example, biasing against female candidates for technical roles (Dastin 2018).

To address these fairness concerns, numerous research efforts have emerged which tackle various definitions of fairness (Chouldechova 2017; Steil et al. 2017; Awasthi et al. 2020; Wan et al. 2021; Castelnovo et al. 2022; Caton and Haas 2020; d'Alessandro, O'Neil, and LaGatta 2017; Zafar et al. 2019; Caton and Haas 2020; Mehrabi et al. 2021; Li et al. 2023). These approaches employ several types of methods, including pre-processing (Goh et al. 2016), in-processing (Quadrianto and Sharmanska 2017), and post-processing methods (Hardt, Price, and Srebro 2016).

In particular, post-processing methods mitigate unfairness via transformations to the model output using an optimization procedure. These methods have the advantage of being model-agnostic and do not require retraining the recommendation model (Li et al. 2023). However, it's important to note that post-processing methods also have limitations. Firstly, the post-processing step usually involves sensitive information and requires large-scale datasets to ensure proper debiasing, which can be a significant constraint since sharing data with third party may raise privacy concern. Secondly, for streaming data, the post-processing procedure must solve optimization problems each time new data arrives, which can be computationally expensive and unsuitable for time-sensitive inference scenarios.

In order to overcome the limitations of post-processing methods, we propose a "model-based" post-processing method which works *independently* of the model that is used to infer the initial outputs of the recommender system. Hence our approach is still "model-agnostic." Our methodology involves training a debiaser to align the biased recommender system with specific fairness constraints. Leveraging the capabilities of modern deep neural networks (DNNs), our method seeks a balance between fairness constraints and the original utility function. The "model-based"

[1] For additional context, see Leonhardt, Anand, and Khosla (2018), where user fairness was first introduced.

method is also capable of addressing limitations of standard post-processing methods. Instead of directly sharing sensitive data for post-processing, it suffices to share the debiaser with a third party, thereby alleviating privacy concerns. Moreover, our post-processing debiaser is well-suited for time-sensitive inference, as it avoids the need for optimization with each new data arrival. Finally, our main theoretical result is a sample complexity guarantees for our method.

Fairness Methods and Related Works

Fairness in recommender systems has previously been explored for a variety of fairness criterion (Wang et al. 2023; Li et al. 2022). One such classification is on the target level, i.e. *group* vs *individual* fairness. Whereas another classification is based on the subject of the measurement, i.e. *who* is the recommender system fair for? Is it the *users* of the recommender system platform, the *items* (and the providers posting those items) being recommended, or both? Both (Wang et al. 2023) and (Li et al. 2022) survey recommender system fairness and provide an overview of works in each of these categories. In this paper we focus on group-wise item fairness in recommendation systems.

Historically the methods used in recommender systems to address unfairness concerns can be classified into three categories: data-oriented methods, ranking methods, and re-ranking methods. Each of these methods addresses one step of the recommender system pipeline. Data-oriented methods seek to improve fairness by adjusting the training data. Ranking methods modify the optimization target of the model, while re-ranking methods post-process the output of models before they are shown to the user. In this categorization, our approach falls under re-ranking methods.

Re-ranking methods are further divided into three categories: slot-wise, user-wise and global-wise (see (Wang et al. 2023) for a full comparison of the methods). Note that in each of these methods, the fairness type can be both user or item[2]. Global-wise methods re-rank multiple recommendation lists at once, considering multiple users during the allocation process. They usually allow for more control over accuracy and fairness parameters but this comes at the cost of higher computational complexity. A popular approach in global-wise re-ranking algorithms is linear programming relaxations of the problem.

Our approach is designed to benefit from the advantages of global-wise approaches while being applicable to "real time inference", for example without having to retrain the post-processing model every time a new user emerges. To the best of our knowledge there are no global-wise model-based recommendation system re-ranking methods prior to this work. However there are several works on more specific fairness problems in recommender systems (Zhu et al. 2021; Fu et al. 2020; Wu et al. 2021).

Method

Preliminaries

We first define a few notations. Let $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ be the context input and $y \in [m]$ denote an item where $[m]$ is a set of m distinct items. We assume data is generated from some distribution $\boldsymbol{z} \equiv (\boldsymbol{x}, y) \sim \mathcal{D}$, where $(\boldsymbol{x}, y)$ is a user context-item pair. A recommender system $f : \mathcal{X} \to [m]^k$ maps user context $\boldsymbol{x}$ into a set of k item candidates. Let Δ^m be an m-dimensional probability simplex. We denote by $\eta(\boldsymbol{x}) : \mathcal{X} \to \Delta^m$ a conditional probability distribution of item clicks for users characterized via context input $\boldsymbol{x}$, where $\eta(\boldsymbol{x})^j$ is the probability of item j being chosen by user $\boldsymbol{x}$. In the context of fairness metrics, we denote S as an enumeration of a discrete and finite set of sensitive variables. Clearly, S is a set of groups that partitions the input space $\mathcal{X}$. We denote $\boldsymbol{x} \in \mathcal{X}_s$ when values of the set of sensitive variables match the value of $s \in S$. Throughout this section, $\lesssim$ and $\gtrsim$ represent as shorthand for the $\leq$ and $\geq$ that ignores universal constants.

Groupwise User-Item Fairness

Our group-wise fairness definition comes from natural exposure-based fairness definitions (Li et al. 2023, 2021). In particular, (Li et al. 2021) defines group-fairness in a recommender system as follows: Let $M(W_i)$ correspond to the quality of the recommendation to user i. (Li et al. 2021) seeks to minimize $\left|\frac{1}{|Z_1|}\sum_{i\in Z_1} M(W_i) - \frac{1}{|Z_2|}\sum_{i\in Z_2} M(W_i)\right|$ for two user groups Z_1 and Z_2. Our fairness measure is similar to that of (Li et al. 2021), however our methodology is very different. [3]

Now we formally define our group fairness metric. Suppose we partition $\mathcal{X}$ by enumeration of sensitive variables $[S]$. A natural definition of group-wise item fairness similar to that of (Li et al. 2021) is to measure the difference between the probability that an item is recommended to different groups. In particular, given two groups $s_1, s_2 \in S$ and a recommender $\eta(\boldsymbol{x})$, we define the following *group-wise item* fairness constraint for item j :

$$\left|\frac{\int_{\boldsymbol{x}\in s_1} \eta(\boldsymbol{x})^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x}\in s_1} p(\boldsymbol{x}) d\boldsymbol{x}} - \frac{\int_{\boldsymbol{x}\in s_2} \eta(\boldsymbol{x})^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x}\in s_2} p(\boldsymbol{x}) d\boldsymbol{x}}\right| \leq C_j \quad (1)$$

where C_j represents tolerance quantity. Given $\Phi(\cdot, \cdot)$ be a uniformly bounded risk function, any "non-debiased" recommender $\eta^U(\boldsymbol{x})$, we aim to "debias" $\eta^U(\boldsymbol{x})$ to satisfy the fairness constraints by solving a stochastic program:

$$\begin{aligned} &\min_{\eta \in \mathcal{F}} \mathbb{E}_{\boldsymbol{x}} \Phi(\eta^U(\boldsymbol{x}), \eta(\boldsymbol{x})) \\ &\text{s.t. } \forall s, s' \in [S], j \in [m], \\ &\qquad \frac{\int_{\boldsymbol{x}\in s} \eta(\boldsymbol{x})^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x}\in s} p(\boldsymbol{x}) d\boldsymbol{x}} - \frac{\int_{\boldsymbol{x}\in s'} \eta(\boldsymbol{x})^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x}\in s'} p(\boldsymbol{x}) d\boldsymbol{x}} \leq C_j \quad (2) \end{aligned}$$

[2] the term "user-wise" must not be confused with "user-based" recommendation systems.

[3] (Li et al. 2021) takes a global-wise approach and writes an integer program using parameters $S_{i,j}$ that correspond to the preference of user i to item j, and solves this integer program using heuristics, outputting a list of recommendations for each user. Note that this approach suffers from the typical shortcomings of global-wise methods which use integer programming, described above.

Our goal is to obtain $\widehat{\eta}$ that approximates the solution of (2). Informally, we call a $\widehat{\eta} \in \mathcal{F}$ a (ϵ, δ)-optimal solution for (2) if it approximates the objective up to additive error ϵ and it obeys the constraints up to error ϵ. We provide formal definition in the Appendix.

Given sufficient amount of data, one can obtain (ε, δ)-optimal solution, for some $\varepsilon > 0, \delta > 0$, by solving the following sample averaged (empirical) version of Equation (2):

$$\min_{\eta \in \mathcal{F}} \frac{1}{n} \sum_{i \in [n]} \Phi(\eta^U(\boldsymbol{x}_i), \eta(\boldsymbol{x}_i)) \tag{3}$$

s.t. $\forall s, s' \in [S], j \in [m]$,

$$\frac{1}{|\boldsymbol{x}_{1:n} \bigcap s|} \sum_{\boldsymbol{x}_i \in \mathcal{X}_s} \eta(\boldsymbol{x}_i)^j - \frac{1}{|\boldsymbol{x}_{1:n} \bigcap s'|} \sum_{\boldsymbol{x}_i \in \mathcal{X}_{s'}} \eta(\boldsymbol{x}_i)^j \leq C_j$$

Where $\boldsymbol{x}_{1:n} \bigcap s$ is the set of $\boldsymbol{x}_i$ that belongs to the group $s \in S$. Note that in this case the probability in Definition 1 is over the sample set.

Sample complexity analysis

We state our main result here. The definitions of covering number, VC-dimension, and pseudo-dimension can be found in (Pollard 2012; Wellner et al. 2013; Mohri, Rostamizadeh, and Talwalkar 2018) and we include them in the appendix.

Assume our hypothesis class is $\mathcal{F}^m : \{\mathcal{X} \to \Delta^m\}$, which could be a family of neural networks with softmax output of size m, mapping a user context $\boldsymbol{x}$ into m-dimensional probability simplex. In particular $\mathcal{F}^1$ denotes the family of neural network with sigmoid output of size 1. Next we present our main theorem.

Theorem 1 *Let $\eta^*(\boldsymbol{x})$ be optimal solution of problem in* (2)*. Let $\widehat{\eta}(\boldsymbol{x})$ be the solution of Equation* (3)*. If the hypothesis class $\mathcal{F}$ has finite Pseudo dimension, is $\beta-$Lipschitz and has B bounded risk, then the following holds: Given $\delta > 0$, with probability at least $1 - \delta$ over the sample set we have:*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}} \Phi(\widehat{\eta}(\boldsymbol{x}), \eta^U(\boldsymbol{x})) &\lesssim \mathbb{E}_{\boldsymbol{x}} \Phi(\eta^*(\boldsymbol{x}), \eta^U(\boldsymbol{x})) \\ &+ \big(log(\frac{m|S|}{\delta}) + \log\big(\frac{n}{md_P(\mathcal{F}^1)}\big)\big)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n}}\end{aligned} \tag{4}$$

And $\forall s, s' \in [S], j \in [m]$,

$$\begin{aligned}&\frac{\int_{\boldsymbol{x} \in s} \widehat{\eta}^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x} \in s} p(\boldsymbol{x}) d\boldsymbol{x}} - \frac{\int_{\boldsymbol{x} \in t} \widehat{\eta}^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x} \in t} p(\boldsymbol{x}) d\boldsymbol{x}} \lesssim C_j + \big(log(\frac{m|S|n}{md_P(\mathcal{F}^1)\delta})\big) \cdot \\ &\max\big\{\frac{1}{\sqrt{|\boldsymbol{x}_{1:n} \bigcap \mathcal{X}_s|}}, \frac{1}{\sqrt{|\boldsymbol{x}_{1:n} \bigcap \mathcal{X}_{s'}|}}\big\}\end{aligned} \tag{5}$$

First note the assumptions made in Theorem 1 are standard in VC-theory and statistical learning. Furthermore, note that if we set ϵ to be equal to the maximum of $\big(log(\frac{m|S|}{\delta}) + \log\big(\frac{n}{md_P(\mathcal{F}^1)}\big)\big)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n}}$ and $\big(log(\frac{m|S|}{\delta}) + \log\big(\frac{n}{md_P(\mathcal{F}^1)}\big)\big) \cdot \max\big\{\frac{1}{\sqrt{|\boldsymbol{x}_{1:n} \bigcap \mathcal{X}_s|}}, \frac{1}{\sqrt{|\boldsymbol{x}_{1:n} \bigcap \mathcal{X}_{s'}|}}\big\}$, Theorem 1 proves that solving 3 indeed obtains a (ϵ, δ)-optimal solution over the the sample set. It worth noting that the sub-optimal gap ε characterized by quantity $\frac{1}{\sqrt{|\boldsymbol{x}_{1:n} \bigcap \mathcal{X}_s|}}$, decreases with $n \to \infty$, at a $1/\sqrt{n}$ rate.

Practical Algorithm: Progressive constrained optimization

The sample average approximate program in Equation 3 can be effectively solved as a convex program when $\Phi(\cdot, \cdot)$ exhibits convexity with respect to the parameter being optimized. However, when $\mathcal{H}$ represents a family of neural networks, the convexity of Equation 3 may not hold.

To adapt Equation 3 to be compatible with deep learning techniques, we introduce an alternating minimization approach. While previous research has addressed constrained problems using methods such as the Augmented Lagrangian method (Sangalli et al. 2021) and constraint completion (Donti, Rolnick, and Kolter 2021), none of these approaches tackle problems involving constraints that span multiple samples. We propose the following natural algorithm for addressing the constrained training problem for neural networks, sharing a similar spirit with the work presented in (Donti, Rolnick, and Kolter 2021).

Algorithm 1: Progressive Constrained Optimization

```
Input: Dataset $A_n \subset R^d \times \{1, 2, \cdots, m\}$, loss function $\Phi$ (e.g., cross-entropy), learning rate $\eta$, number of iterations max_iter, Non-debiased model $\eta^U(\cdot)$
$\widehat{\eta}(\cdot) \leftarrow \eta^U(\cdot)$
Obtain Pseudo Label $\widehat{y}_{1:n}$ by solving convex program:
    $\min_{y_{1:n} \in \{\Delta^m\}^n} \frac{1}{n} \sum_{i \in [n]} \Phi(y_i, \widehat{\eta}(\boldsymbol{x}_i))$    (6)
    s.t. $\forall s, s' \in S, j \in [m]$,
    $\frac{1}{|\boldsymbol{x}_{1:n} \bigcap s|} \sum_{\boldsymbol{x}_i \in \mathcal{X}_s} y_i^j - \frac{1}{|\boldsymbol{x}_{1:n} \bigcap s'|} \sum_{\boldsymbol{x}_i \in \mathcal{X}_{s'}} y_i^j \leq C_j$
Train $\widehat{\eta}(\cdot)$ based using $\{(\boldsymbol{x}_i, \widehat{y}_i)\}_{1:n}$ using unconstrained loss: $\widehat{\eta} := \arg\min_{\eta \in \mathcal{F}} \frac{1}{n} \sum_i^n \Phi(\eta(\boldsymbol{x}_i), \widehat{y}_i)$.
Output: $\widehat{\eta}(\cdot)$
```

Experiments

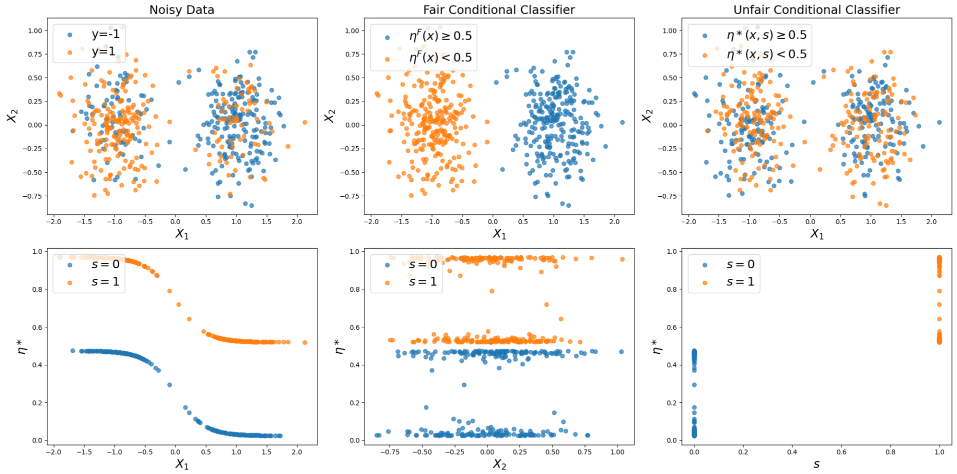
In this section, we study a toy example. The data consists of non-sensitive variables $\boldsymbol{x}$ and sensitive variable s where $\boldsymbol{x}$ and s are assumed to be independent of each other. We first define 'fair' conditional probability $\eta^F(\boldsymbol{x}) \triangleq \mathbb{P}[y|\boldsymbol{x}]$. We use a mixture of Spherical Gaussians $\pi = 0.5\pi_1 + 0.5\pi_2$ in $\mathbb{R}^2$ where $\pi_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \sigma I)$ represents the distribution for $\boldsymbol{x}$ under class $y = 1$ and $\pi_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \sigma I)$ for $y = -1$. Their means are $\boldsymbol{\mu}_1 = [1, 0]^T$ and $\boldsymbol{\mu}_2 = [-1, 0]^T$. The prior probability is set equal for both classes. Let $\phi(\boldsymbol{x}) = \frac{1}{2\pi}\exp(-\frac{\boldsymbol{x}^T\boldsymbol{x}}{2})$ be the density for 2-d standard Gaussian. We define $\Gamma(\cdot)$as follows:

$$\Gamma(\boldsymbol{x}) = \frac{\phi(\boldsymbol{x} - \boldsymbol{\mu}_1)}{\phi(\boldsymbol{x} - \boldsymbol{\mu}_1) + \phi(\boldsymbol{x} - \boldsymbol{\mu}_2)}$$

Suppose $\rho \in (0, 1]$, we have $\eta^F(\boldsymbol{x})$ as follows:

$$\eta^F(\boldsymbol{x}) = 1 - \Gamma(\boldsymbol{x}) + \rho\big(2\mathbb{1}\{\Gamma(\boldsymbol{x}) \geq \frac{1}{2}\} - 1\big)\big(\Gamma(\boldsymbol{x}) - \frac{1}{2}\big)^2$$

Figure 1: In the top row, we show the non-sensitive variables with respect to the noisy labels, the ‘fair’ conditional probability, and the ‘unfair’ conditional probability, respectively. In the bottom row, we show the ‘unfair’ conditional risk for values of each of the non-sensitive variables with respect to the sensitive variables.



The ‘fair’ optimal classifier is thus $f^*(\boldsymbol{x}) = 2\,\mathbb{1}\{\eta^F(\boldsymbol{x}) \geq \frac{1}{2}\} - 1 = 2\,\mathbb{1}\{x_1 \geq 0\} - 1$. Given the sensitive variable $s \in \{0, 1\}$, we define $\eta^S(s) \triangleq \mathbb{P}[y|s] = s$. The observation y is generated by Bernoulli with mixing parameter λ with

$$\mathbb{P}[y|\boldsymbol{x}, s] \triangleq \eta^*(\boldsymbol{x}, s) \triangleq \lambda \cdot \eta^F(\boldsymbol{x}) + (1 - \lambda) \cdot \eta^S(s) \quad (7)$$

In our experiments, we set $\lambda = 0.5$, $\rho = 0.2$, $\sigma = 0.1$, and $n = 500$. We choose C_j to be half of the maximum empirical group difference $0.5\big(\max_j[\frac{1}{|\boldsymbol{x}_{1:n} \bigcap s|}\sum_{\boldsymbol{x}_i \in \mathcal{X}_s} \eta(\boldsymbol{x}_i)^j - \frac{1}{|\boldsymbol{x}_{1:n} \bigcap s'|}\sum_{\boldsymbol{x}_i \in \mathcal{X}_{s'}} \eta(\boldsymbol{x}_i)^j]\big)$. In Step 1 of 1, we learn the noisy labels with an MLP, using Binary Cross Entropy Loss. We achieve good performance, but without high accuracy with respect to the ground truth conditional probability. In the second step, we perform convex optimization to learn pseudo labels which correct for item-wise group differences between the sensitive groups. We use *cvxpy* (Diamond and Boyd 2016) to solve this convex constrained optimization. Lastly, the same three layer MLP trains the noisy data on the pseudo labels. Results and discussions are shown in Table1.

	Non-debiased	Optimization	Debiased
Train. MGD	0.416	NaN	0.217
Valid. MGD	0.388	0.148	0.217
Train. Acc	0.750	NaN	0.716
Valid. Acc	0.740	0.548	0.738

Table 1: Let $\boldsymbol{z} \triangleq (\boldsymbol{x}, s, y)$, in the column Non-debiased, we present the mean group difference (MGD) as $\|\sum_{\boldsymbol{z},s=1} \eta^U(\boldsymbol{x}, s) - \sum_{\boldsymbol{z},s=0} \eta^U(\boldsymbol{x}, s)\|_1$. In the column Optimization, we report the mean group difference as $\|\sum_{\boldsymbol{z},s=1} \widehat{y} - \sum_{\boldsymbol{z},s=0} \widehat{y}\|_1$. In the columns Debiasing, we display mean group difference as $\|\sum_{\boldsymbol{z},s=1} \widehat{\eta}(\boldsymbol{x}, s) - \sum_{\boldsymbol{z},s=1} \widehat{\eta}(\boldsymbol{x}, s)\|_1$. The training/testing accuracy follows standard manner. It could be observed that: (*1*) The optimization step effectively mitigates group differences. (*2*) During the debiasing step, the neural network leverages information from the ‘debiased label,’ resulting in fair outputs that generalize well on testing data. (*3*) The decrease in testing accuracy for the debiased model is at a benign level.

Conclusion

In this study, we introduce a post-processing framework for addressing groupwise user-item fairness in recommender systems. When provided with enough samples of the recommender outputs, our method is designed to learn a model directly by incorporating all fairness constraints into the loss function of the recommender. We provide an analysis of sample complexity to ensure the generalization performance of our model learned from finite data samples.

For a family of neural network-based recommenders, we also present a heuristic algorithm to effectively solve the optimization problem while considering fairness constraints. In our experimental evaluation, we demonstrate the effectiveness of our proposed approach using synthetic data. There are several future directions remain to explore including

- **Convergence guarantee of Algorithm 1** It would be interesting to investigate the convergence behavior of Algorithm 1 and its sub-optimality w.r.t to population risk and constraints in Equation 2.
- **Realistic synthetic data** In our current synthetic data experiments, the relationship between the sensitive variable and non-sensitive variables is not entirely representative of real-world scenarios, which often involve more sophisticated problem structures. We are eager to explore the use of more realistic synthetic data in our studies, such as the synthetic data examined in (Chaudhari et al. 2022). This approach will enable us to better capture the complexities of real-world situations in our research.
- **Real world data.** We look forward to more empirical study of Equation 3 and Algorithm 1 on real world datasets. This will allow us to validate and extend our findings to practical, real-world scenarios, providing valuable insights into the applicability and performance of our methods in real-world settings.

References

Awasthi, P.; Cortes, C.; Mansour, Y.; and Mohri, M. 2020. Beyond individual and group fairness. *arXiv preprint arXiv:2008.09490*.

Castelnovo, A.; Crupi, R.; Greco, G.; Regoli, D.; Penco, I. G.; and Cosentini, A. C. 2022. A clarification of the nuances in the fairness metrics landscape. *Scientific Reports*, 12(1): 4209.

Caton, S.; and Haas, C. 2020. Fairness in machine learning: A survey. *ACM Computing Surveys*.

Chaudhari, B.; Choudhary, H.; Agarwal, A.; Meena, K.; and Bhowmik, T. 2022. FairGen: Fair Synthetic Data Generation. *arXiv preprint arXiv:2210.13023*.

Chouldechova, A. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2): 153–163.

d'Alessandro, B.; O'Neil, C.; and LaGatta, T. 2017. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data*, 5(2): 120–134.

Dastin, J. 2018. Amazon scraps secret AI recruiting tool that showed bias against women.

Diamond, S.; and Boyd, S. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1): 2909–2913.

Donti, P. L.; Rolnick, D.; and Kolter, J. Z. 2021. DC3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*.

Fu, Z.; Xian, Y.; Gao, R.; Zhao, J.; Huang, Q.; Ge, Y.; Xu, S.; Geng, S.; Shah, C.; Zhang, Y.; et al. 2020. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 69–78.

Goh, G.; Cotter, A.; Gupta, M.; and Friedlander, M. P. 2016. Satisfying real-world goals with dataset constraints. *Advances in neural information processing systems*, 29.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Leonhardt, J.; Anand, A.; and Khosla, M. 2018. User fairness in recommender systems. In *Companion Proceedings of the The Web Conference 2018*, 101–102.

Li, Y.; Chen, H.; Fu, Z.; Ge, Y.; and Zhang, Y. 2021. User-oriented fairness in recommendation. In *Proceedings of the Web Conference 2021*, 624–632.

Li, Y.; Chen, H.; Xu, S.; Ge, Y.; Tan, J.; Liu, S.; and Zhang, Y. 2022. Fairness in Recommendation: A Survey. arXiv:2205.13619.

Li, Y.; Chen, H.; Xu, S.; Ge, Y.; Tan, J.; Liu, S.; and Zhang, Y. 2023. Fairness in Recommendation: Foundations, Methods and Applications. *ACM Transactions on Intelligent Systems and Technology*.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6): 1–35.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of machine learning*. MIT press.

Pollard, D. 2012. *Convergence of stochastic processes*. Springer Science & Business Media.

Quadrianto, N.; and Sharmanska, V. 2017. Recycling privileged learning and distribution matching for fairness. *Advances in neural information processing systems*, 30.

Sangalli, S.; Erdil, E.; Hötker, A.; Donati, O.; and Konukoglu, E. 2021. Constrained optimization to train neural networks on critical and under-represented classes. *Advances in Neural Information Processing Systems*, 34: 25400–25411.

Steil, J. P.; Albright, L.; Rugh, J. S.; and Massey, D. S. 2017. The social structure of mortgage discrimination. *Housing Studies*, 33(5): 759–776.

Wan, M.; Zha, D.; Liu, N.; and Zou, N. 2021. Modeling techniques for machine learning fairness: A survey. *arXiv preprint arXiv:2111.03015*.

Wang, Y.; Ma, W.; Zhang, M.; Liu, Y.; and Ma, S. 2023. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3): 1–43.

Wellner, J.; et al. 2013. *Weak convergence and empirical processes: with applications to statistics*. Springer Science & Business Media.

Wu, L.; Chen, L.; Shao, P.; Hong, R.; Wang, X.; and Wang, M. 2021. Learning fair representations for recommendation: A graph-based perspective. In *Proceedings of the Web Conference 2021*, 2198–2208.

Zafar, M. B.; Valera, I.; Gomez-Rodriguez, M.; and Gummadi, K. P. 2019. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1): 2737–2778.

Zhu, Z.; Kim, J.; Nguyen, T.; Fenton, A.; and Caverlee, J. 2021. Fairness among New Items in Cold Start Recommender Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, 767–776. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380379.

Appendix

We begin by some definitions and then provide the proof of Theorem 1.

Definition 1 (**(ε, δ)-optimal solution**) *Let $\eta^*(\boldsymbol{x})$ be any solution of equation (2). We say an algorithm gives a (ε, δ)-optimal solution for the problem in Equation* (2) *if with probability at least $1-\delta$, it outputs a $\widehat{\eta} \in \mathcal{F}$ such that the following probabilities hold:*

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}}\Phi(\eta^U(\boldsymbol{x}), \widehat{\eta}(\boldsymbol{x})) \lesssim \mathbb{E}_{\boldsymbol{x}}\Phi(\eta^U(\boldsymbol{x}), \eta^*(\boldsymbol{x})) + \varepsilon \\ &\textit{s.t. } \forall s, s' \in [S], j \in [m], \\ &\frac{\int_{\boldsymbol{x}\in s} \widehat{\eta}(\boldsymbol{x})^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x}\in s} p(\boldsymbol{x}) d\boldsymbol{x}} - \frac{\int_{\boldsymbol{x}\in s'} \widehat{\eta}(\boldsymbol{x})^j p(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{x}\in s'} p(\boldsymbol{x}) d\boldsymbol{x}} - C_j \lesssim \varepsilon \end{aligned} \tag{8}$$

Assumption 1 *Throughout the analysis, we make the following assumptions:*

- ***Hypothesis class complexity**: We assume that the family of recommenders is from hypothesis class $\mathcal{F}^1 : \{\mathcal{X} \to \Delta^1\}$ with finite Pseudo dimension $d_P(\mathcal{F}^1) < \infty$.*
- ***β-Lipschitz and B bounded Risk:** We assume risk function $\Phi(\cdot, \cdot) : \Delta^m \times \Delta^m \to [0, B]$ is β–Lipschitz.*

Definition 2 (**L_2-Covering Number**) *Let $\boldsymbol{x}_{1:n}$ be set of points and let $\mathcal{F} : \mathcal{X} \to \Delta^m$ be a hypothesis class. A set $U \subseteq \mathbb{R}^n$ is an ε-cover w.r.t L_2-norm of $\mathcal{F}$ on $x_{1:n}$, if $\forall f \in \mathcal{F}$, $\exists u \in U$, s.t. $\sqrt{\frac{1}{n}\sum_{i=1}^n \|[u]_i - f(x_i)\|_2^2} \leq \varepsilon$, where $[u]_i$ is the i-th coordinate of u. The covering number $\mathcal{N}_2(\varepsilon, \mathcal{F}, n)$ with 2-norm of size n on $\mathcal{F}$ is :*

$$\sup_{\boldsymbol{x}_{1:n}\in\mathcal{X}^n} \min\{|U| \textit{: } U \textit{ is an } \varepsilon\textit{-cover of } \mathcal{F} \textit{ on } x_{1:n}\} \tag{9}$$

Definition 3 (**VC-dimension**) *The VC-dimension $d_{\mathrm{VC}}(\mathcal{F})$ of a hypothesis class $\mathcal{F} = \{f : \mathcal{X} \to \{1, -1\}\}$ is the largest cardinality of the set $A \subseteq \mathcal{X}$ such that $\forall \bar{A} \subseteq A$, $\exists f \in \mathcal{F}$:*

$$f(x) = \begin{cases} 1 & \textit{if } x \in \bar{A} \\ -1 & \textit{if } x \in A \setminus \bar{A} \end{cases} \tag{10}$$

Definition 4 (**Pseudo-dimension**) *The Pseudo-dimension $d_P(\mathcal{F}^1)$ of a real-valued hypothesis class $\mathcal{F} = \{f : \mathcal{X} \to [a, b]\}$ is the VC-dimension of the hypothesis class $\mathcal{H} = \{h : \mathcal{X} \times \mathbb{R} \to \{-1, 1\} | h(\boldsymbol{x}, t) = \mathrm{sign}(f(\boldsymbol{x}) - t), f \in \mathcal{F}\}$.*

Proof of Theorem 1. First we define some notation. Let A_n be the set of samples. Define $\|\eta_1 - \eta_2\|_{A_n} := \sqrt{\frac{1}{n}\sum_{i=1}^n \|\eta_1(\boldsymbol{x}_i) - \eta_2(\boldsymbol{x}_i)\|^2}$ and $\|\eta_1 - \eta_2\|_{\mu(\boldsymbol{x})} := \sqrt{\mathbb{E}_{\boldsymbol{x}}[\|\eta_1(\boldsymbol{x}) - \eta_2(\boldsymbol{x})\|^2]}$. Let $\mathcal{C}$ be a γ/β-cover for hypothesis class $\mathcal{F}^m$ projected on A_n for some γ that we define later. For any hypothesis f let $c(\eta)$ be an element in $\mathcal{C}$ that covers η. In particular we have for any η there exists $c(\eta) \in \mathcal{C}$ such that $\|c(\eta) - \eta\|_{A_n} \leq \gamma/\beta$. Clearly, for any $\eta \in \mathcal{F}$ that satisfies the constraints in (3), we have $\sum_{i=1}^n \Phi(\widehat{\eta}(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i)) \leq \sum_{i=1}^n \Phi(\eta(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i))$. Moreover we can observe that γ/β-cover of $\mathcal{F}$ on A_n is also a γ-cover of $\Phi(\eta, \eta^U)$ for $\eta \in \mathcal{F}$, projected on A_n. To see this:

$$\begin{aligned} &\|\Phi(\eta_1, \eta^U) - \Phi(c(\eta_1), \eta^U)\|_{A_n}^2 \\ =&\frac{1}{n}\sum_{i=1}^n (\Phi(\eta_1(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i)) - \Phi(c(\eta_1)(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i)))^2 \\ \leq&\frac{\beta^2}{n}\sum_{i=1}^n \|\eta_1(\boldsymbol{x}_i) - c(\eta_1)(\boldsymbol{x}_i)\|^2 \\ \leq&\gamma^2 \end{aligned} \tag{11}$$

Where in 11 we use the β-Lipschitz property of Φ. Since

$$\sum_{i=1}^n \Phi(\widehat{\eta}(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i)) \leq \sum_{i=1}^n \Phi(\eta^*(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i)), \tag{12}$$

we have

$$\sum_{i=1}^n \Phi(c(\widehat{\eta}(\boldsymbol{x}_i)), \eta^U(\boldsymbol{x}_i)) \leq \sum_{i=1}^n \Phi(\eta^*(\boldsymbol{x}_i), \eta^U(\boldsymbol{x}_i)) + \gamma. \tag{13}$$

We apply an empirical process argument, using a Hoeffding type inequality with symmetricity (Pollard 2012) and taking the union bound on the covering set $\mathcal{C}$. We have

$$\mathbb{P}_{A_n}\left[\sup_{\eta\in\mathcal{F}}\left\{\left|\mathbb{E}_{\boldsymbol{x}}\big[\Phi(\eta(\boldsymbol{x}),\eta^U(\boldsymbol{x}))\big]-\frac{1}{n}\sum_{i=1}^{n}\Phi(\eta(\boldsymbol{x}_i),\eta^U(\boldsymbol{x}_i))\right|\geq\gamma\right\}\right]\leq 2\mathbb{E}_{A_n}[|\mathcal{C}|]e^{(-\frac{n\gamma^2}{2})} \tag{14}$$

The size of the γ/β-covering number of $\mathcal{F}^m$ could be bounded using Cartesian product of m cover of $\mathcal{F}^1$ with radius , $\gamma/(\beta\sqrt{m})$. The covering number of $\mathcal{F}^1$ could be bounded using the Pseudo-dimension using Theorem 2.6.4 in (Wellner et al. 2013). Formally: $|\mathcal{C}|\leq\left\{c_2 d_P(\mathcal{F}^1)c_3^{d_P(\mathcal{F}^1)}\left(\frac{1}{\gamma m}\right)^{2d_P(\mathcal{F}^1)}\right\}^m$ where $c_2,c_3<\infty$ are some universal constants. By setting

$$\gamma\lesssim\left(\log\big(\frac{n}{md_P(\mathcal{F}^1)}\big)+\log\big(\frac{m|S|}{\delta}\big)\right)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n}},$$

we have $\mathbb{E}_{A_n}[|\mathcal{C}|]e^{(-\frac{n\gamma^2}{2})}\lesssim\frac{\delta}{m|S|^2}$. Using inequality (14), we have that with probability at least $1-\frac{\delta}{m|S|^2}$:

$$\mathbb{E}_{\boldsymbol{x}}\big[\Phi(\widehat{\eta}(\boldsymbol{x}),\eta^U(\boldsymbol{x}))\big]-\frac{1}{n}\sum_{i=1}^{n}\Phi(\widehat{\eta}(\boldsymbol{x}_i),\eta^U(\boldsymbol{x}_i))\leq\left(log(\frac{1}{\delta})+\log\big(\frac{n}{md_P(\mathcal{F}^1)}\big)\right)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n}}$$

$$\frac{1}{n}\sum_{i=1}^{n}\Phi(\eta^*(\boldsymbol{x}_i),\eta^U(\boldsymbol{x}_i))-\mathbb{E}_{\boldsymbol{x}}\big[\Phi(\eta^*(\boldsymbol{x}),\eta^U(\boldsymbol{x}))\big]\leq\left(log(\frac{1}{\delta})+\log\big(\frac{n}{md_P(\mathcal{F}^1)}\big)\right)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n}} \tag{15}$$

which gives (4). Next we show (5). Since S is a partition of $\mathcal{X}$, one can view $\boldsymbol{x}_{1:n}\in\mathcal{X}_s$ as i.i.d samples, conditional on $\mathcal{X}_s$. Let $n_s\equiv|\boldsymbol{x}_{1:n}\bigcap\mathcal{X}_s|$, by similar arguments as that of (14) we have for any $\gamma>0$:

$$\mathbb{P}\left[\sup_{\eta\in\mathcal{F}}\left\{\left|\mathbb{E}_{\boldsymbol{x}}\big[\eta^j(\boldsymbol{x})|\boldsymbol{x}\in\mathcal{X}_s\big]-\frac{1}{n_s}\sum_{\boldsymbol{x}_i\in\mathcal{X}_s}\Phi(\eta(\boldsymbol{x}_i),\eta^U(\boldsymbol{x}_i))\right|\geq\gamma\right\}\right]\leq 2\mathbb{E}_{A_n}[|\mathcal{C}|]e^{(-\frac{n_s\gamma^2}{2})} \tag{16}$$

It suffices to pick $\gamma\lesssim\left(\log\big(\frac{n_s}{md_P(\mathcal{F}^1)}\big)+\log\big(\frac{m|S|}{\delta}\big)\right)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n_s}}$, so that the R.H.S of (16) becomes $\frac{\delta}{m|S|^2}$. Taking a union bound on m items and $|S|^2-|S|$ groups for the constraints in (3), we have with probability at least $1-\delta$, for all $j\in[m]$, $s,s'\in[S]$,

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}}\big[\eta^j(\boldsymbol{x})|\boldsymbol{x}\in\mathcal{X}_s\big]-\frac{1}{|n_s|}\sum_{\boldsymbol{x}_i\in\mathcal{X}_s}\eta^j(\boldsymbol{x}_i)-\mathbb{E}_{\boldsymbol{x}}\big[\eta^j(\boldsymbol{x})|\boldsymbol{x}\in\mathcal{X}_{s'}\big]+\frac{1}{|n_{s'}|}\sum_{\boldsymbol{x}_i\in\mathcal{X}_{s'}}\eta^j(\boldsymbol{x}_i)\\ \lesssim&\left(\log\big(\frac{n_s}{md_P(\mathcal{F}^1)}\big)+\log\big(\frac{m|S|}{\delta}\big)\right)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n_s}}+\left(\log\big(\frac{n_{s'}}{md_P(\mathcal{F}^1)}\big)+\log\big(\frac{m|S|}{\delta}\big)\right)\frac{Bmd_P(\mathcal{F}^1)}{\sqrt{n_{s'}}}\end{aligned} \tag{17}$$

This finishes the proof.