A Pre-processing Method for Fairness in Ranking

Ryosuke Sonoda
sonoda.ryosuke@fujitsu.com
Fujitsu Ltd.
Kanagawa, Kawasaki

ABSTRACT

Fair ranking problems arise in many decision-making processes that often necessitate a trade-off between accuracy and fairness. Many existing studies have proposed correction methods such as adding fairness constraints to a ranking model's loss. However, the challenge of correcting the data bias for fair ranking remains, and the trade-off of the ranking models leaves room for improvement. In this paper, we propose a fair ranking framework that evaluates the order of training data in a pairwise manner as well as various fairness measurements in ranking. This study is the first proposal of a pre-processing method that solves fair ranking problems using the pairwise ordering method with our best knowledge. The fair pairwise ordering method is prominent in training the fair ranking models because it ensures that the resulting ranking likely becomes parity across groups. As far as the fairness measurements in ranking are represented as a linear constraint of the ranking models, we proved that the minimization of loss function subject to the constraints is reduced to the closed solution of the minimization problem augmented by weights to training data. This closed solution inspires us to present a practical and stable algorithm that iterates the optimization of weights and model parameters. The empirical results over real-world datasets demonstrated that our method outperforms the existing methods in the trade-off between accuracy and fairness over real-world datasets and various fairness measurements.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; • **Applied computing** → *Law, social and behavioral sciences.*

KEYWORDS

Fairness, Ranking, Machine Learning

ACM Reference Format:
Ryosuke Sonoda. 2021. A Pre-processing Method for Fairness in Ranking. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

1 INTRODUCTION

Fairness in ranking is a fundamental problem in information retrieval as a data-driven ranking model is often used in various online contexts to search for content, products, and people [6]. The issues of fairness often arise when the ranking model is employed in decision-making processes, such as predicting recidivism for release and income for loan screening. Because data usually contains biases for certain groups (e.g., gender, race, or origin) [14], without proper treatment, the ranking model trained on such data can yield unfair results for those groups.

In this work, we study the fair machine learning method for the trade-off between accuracy and fairness in ranking. This means that we address the following two challenges. First, we must align with fairness measurements in ranking [2, 18] in the fair method to ensure unbiased results of item ordering across groups. Second, we need to make the fair method (i.e., loss minimization problems subject to the fairness measurements) tractable for the better trade-off as the fairness measurements are usually non-differentiable.

Although many fair machine learning methods have been proposed [2, 12, 15, 16, 18, 21, 25, 26], less studies have been made on fair ranking with our best knowledge. Jiang and Nachu have proposed the data re-weighting method to correct the biases in training data for fair classification [16]. Their method weights the observed labels in a way that the loss function of the model becomes unbiased, and was shown to be superior to existing methods in fair classification [12, 15]. Narasimhan et al. have proposed the method that adds the fairness constraints to the ranking model loss function for fair ranking [21]. This method considers the order of data in both loss function and constraints, and was shown to be more promising than existing methods for fair ranking [2, 18, 25]. Despite their solid theoretical foundations, these methods remain for better trade-off because they do not propose a pre-processing method that considers the fairness measurements in ranking.

In this paper, we propose a fair ranking framework that evaluates the order of training data in a pairwise manner as well as various fairness measurements in ranking. This study is the first proposal of a pre-processing method that solves fair ranking problems using the pairwise ordering method with our best knowledge. The fair pairwise ordering method is prominent in training the fair ranking models because it ensures that the resulting ranking likely becomes parity across groups. As far as the fairness measurements in ranking are represented as a linear constraint of the ranking models, we proved that the minimization of loss function subject to the constraints is reduced to the closed solution of the minimization problem augmented by weights to training data. This closed solution inspires us to present a practical and stable algorithm that iterates the optimization of weights and model parameters. The empirical results over real-world datasets demonstrated that our method outperforms the existing methods in the trade-off between

accuracy and fairness over real-world datasets and various fairness measurements.

The remainder of this paper is organized as follows: we describe works related to our study in Section 2. We describe the problem formulation, definition of measurements in ranking, and introduce the re-weighing method in Section 3. We present our proposed method that optimizes the trade-off between accuracy and fairness in ranking in Section 4. We show our experimental settings and results in Section 5. We conclude in Section 6.

2 RELATED WORK

In this section, we introduce related work on ranking algorithms, fairness in machine learning, and fairness in ranking algorithms.

2.1 Ranking algorithms

Ranking algorithms are widely recognized for their potential for societal impact [1], as they form the core of many online systems, including search engines, recommendation systems, news feeds, and online voting. The ranking algorithms typically use machine learning techniques to construct a ranking model, given a set of query-item pairs. The goal of the ranking model is to predict the score of new items and then sort the score for each query. The main challenge of this task is that data for training a ranking model is usually long-tailed, which leads to the rich-get-richer problem [20]. Thus, many studies have proposed ranking algorithms to deal with this problem [4, 5, 7, 19, 24].

The most straightforward way to solve the ranking problem is the algorithm based on the pointwise ordering method that aims to predict the label of each item. For example, classification-based algorithms and regression-based algorithms predict the score of the item based on its ground truth label [7, 19]. Therefore, the loss function of the pointwise ordering method only considers the relevance between the queries and the items, not the ranking of the items. In contrast, many works have proposed the pairwise ordering method that usually works better than the pointwise one [4, 5, 24]. This is because the key issue of ranking is to determine the orders of items but not to judge the label of items, which is exactly the goal of the pairwise ordering method. Moreover, in the long-tailed data, the pointwise ordering method falls into sub-optimal because its loss function is dominated by some queries with lots of items [20]. On the other hand, the pairwise ordering method, which considers the order of item pairs in the same query in its loss, can avoid this problem.

Given the above reasons, the pairwise ordering method is widely applied for ranking problems than the pointwise one. In this paper, we focus on the pairwise ordering method.

2.2 Fairness in Machine Learning

Fairness has become a focus of attention in machine learning techniques that have been integrated into the decision-making process, such as loan screening, release decisions, and online search. In machine learning, fairness is the absence of any prejudice toward certain groups (e.g., gender, race, or origin) [12]. The main challenges for this fairness are to determine how to assess fairness and how to improve fairness. There is no unique definition of fairness measurements to assess fairness because the fairness is context dependent [12, 25]. The fair methods to improve the trade-off between accuracy and fairness are detailed below.

Many methods have been proposed to improve the trade-off using fairness measurements as constraints, and these methods can be categorized as post-, in-, and pre-processing methods, based on the processes they perform. The goal of the post-processing methods is to satisfy the constraint by appropriately changing the predictions from a machine learning model [15, 25]. These methods address the problem of fairness when modifying the ranking model or training dataset is challenging.

Many works have proposed the in-processing methods that tackle fairness in the learning process by adding constraints to model loss [8, 9, 28]. The advantage of in-processing methods is that they can apply the method of Lagrange multipliers to transform the constraints to penalties. Thus, the in-processing methods can optimize the ranking model parameters and the Lagrangian multipliers simultaneously. For this reason, in general, the in-processing methods have a better trade-off than the post-processing methods [8, 9].

The pre-processing methods address the problem of biases in training data, which was open in the post- and in-processing methods [12]. The main idea of the pre-processing methods is to create the "fair" dataset by modifying the original training dataset. In contrast to the post- and in-processing methods, the pre-processing methods have the advantage of stability and simplicity since the procedure is model agnostic. While the in-processing methods approximate non-differential constraints, leading to high complexity and unstable convergence of the ranking model [9], the pre-processing methods do not require such approximations and are therefore more practical to fairness.

In this paper, we address the problem of automatically estimating how much to correct biases in training data, which has not been addressed in existing pre-processing methods. In other words, we aim to improve the trade-off of the pre-processing methods over the post- and in-processing methods.

2.3 Fairness in Ranking Algorithms

Algorithms for fairness in ranking are a new direction and arising progress in machine learning. The goal of these algorithms is to develop machine learning techniques to achieve the trade-off in ranking.

Many works have proposed the techniques for fair ranking algorithms [2, 16, 18, 21, 25, 26]. Especially, Singh and Joachims [25] considered the fairness of rankings through the lens of exposure allocation between groups. Instead of defining a single measurement of fairness, they developed a general framework that employs the post-processing method and linear programming to optimize the accuracy-maximizing ranking under a class of constraints. Beutel et al. [2] have also provided multiple definitions of fairness measurements in rankings through pairwise data, called pairwise fairness. They have presented the in-processing method using a fixed constraint that correlated to pairwise fairness. Rather than using fixed constraints, Narasimhan et al. [21] have proposed the in-processing method that can handle a class of pairwise fairness as a constraint. Furthermore, they have shown that their in-processing

method outperforms the existing methods [2, 25] in a variety of experimental settings. Recently, Jiang and Nachum [16] presented a new framework to model label bias, assuming that there exists an unbiased ground truth. Their pre-processing method for correcting for this bias is based on re-weighting the training data points. The contribution of this work is that the algorithm can estimate the appropriate weights for each data point.

Existing studies have not addressed data bias and ranking problems simultaneously; thus, the trade-off is still a problem. Therefore, in this paper, we propose a pre-processing based on pairwise fairness by making the re-weighting method [16] pairwise.

3 PROBLEM FORMULATION

In this section, we introduce the settings handled by the pointwise ordering method and the pairwise ordering method. The setting for the pointwise ordering method is an extension of the setting introduced by [16] for ranking, while we introduce the new setting for the pairwise ordering method.

3.1 Settings for The Pointwise Ordering Method

We are given queries $q \in Q$ drawn from an underlying distribution $\mathcal{D}$, where each query q has a set of items $\mathcal{R}_q$ to be ranked. Consider a feature space $\mathcal{X} \subseteq \mathbb{R}^d$ where d is a dimension of the feature space and an associated feature distribution $\mathcal{P}$. Then each item $i \in \mathcal{R}_q$ is represented by an associated vector $x_i \in \mathcal{X}$ and a label $y_i \in \mathcal{Y}$ (e.g., for $\mathcal{Y} := \{0, 1\}$: $y_i = 1$ if query q and item i are relevant, $y_i = 0$ otherwise). In this paper, we use a binary label setting. However, our method may be readily generalized to other settings (e.g., $\mathcal{Y} \in \mathbb{R}$).

We assume that the existence of an unbiased, ground-truth label function $y_{\text{true}} \colon Q \times \mathcal{X} \to [0, 1]$. We usually do not have access to the "true" values of the function y_{true}. Thus, y_{true} is the assumed ground truth. The labels of our dataset are generated based on a biased label function $y_{\text{bias}} \colon Q \times \mathcal{X} \to [0, 1]$. Accordingly, we assume that our data for pointwise setting is drawn as follows:

$$(q, x, y) \sim \mathcal{D} \equiv q \sim Q, x \sim \mathcal{P}, y \sim \text{Bernoulli}(y_{\text{bias}}(q, x)). \tag{1}$$

In the following, we introduce accuracy and fairness measurements that the pointwise ordering method can handle. The conventional goal in pointwise ordering method is to find a ranking model $h \colon Q \times \mathcal{X} \to \mathbb{R}$ that maximizes the expected accuracy:

$$\arg\max_h \mathbb{E}_{q \sim Q} \left[P(h_q(x_i) = y_{\text{true}}(q, x_i)) \right], \tag{2}$$

where $h_q(x_i) = h(q, x_i)$. Also, we define the expected bias of h based on a class of constraint for pointwise constraints c^{point} introduced [9, 16]:

$$\mathbb{E}_{q \sim Q}[\mathbb{E}_{x_i \sim \mathcal{P}}[\langle h_q(x_i), c^{\text{point}}(q, x_i) \rangle]], \tag{3}$$

where $\langle h_q(x_i) c^{\text{point}}(q, x_i) \rangle := \sum_{y_i \in \mathcal{Y}} h_q(y_i \mid x_i) c^{\text{point}}(q, x_i, y_i)$. We use the shorthand $h(y_i \mid x_i)$ to denote the probability of sampling y_i from a Bernoulli random variable with $p = h_q(x_i)$; i.e., $h_q(y \mid x_i) := h_q(x_i)$ and $h(0 \mid x_i) := 1 - h_q(x_i)$. For c^{point}, one can employ the statistical parity fairness [12], the equal opportunity fairness [15], etc. If the the fairness is ideal, Eq.(3) is 0. When h is biased, some amount is given by Eq.(3).

3.2 Settings for The Pairwise Ordering Method

In this section, we introduce the settings for our pairwise ordering method. In the pairwise setting, we consider item pairs $i, j \in \mathcal{R}_q$ from dataset $\mathcal{D}$. Then we define the binary pair label l_{ij} that represents the relative order of the pair (y_i, y_j) (i.e., $l_{ij} = 1$ if $y_i > y_j$, and $l_{ij} = 0$ if $y_i < y_j$). Here, we introduce a new concept, a pair label function, to consider biases of pair labels. We assume the existence of an unbiased, ground-truth pair label function $l_{\text{true}} \colon Q \times \mathcal{X} \times \mathcal{X} \to [0, 1]$ We do not have the actual values for the ground-truth pair label function l_{true}. Instead, we observe the biased pair label function $l_{\text{bias}} \colon Q \times \mathcal{X} \times \mathcal{X} \to [0, 1]$. Accordingly, we assume that our data for pairwise settings $\mathcal{D}_{\text{pair}}$ is drawn as follows:

$$\begin{aligned} &(q, x_i, x_j, l_{ij}) \sim \mathcal{D}_{\text{pair}} \\ &\equiv q \sim Q, (x_i, x_j) \sim \mathcal{P}^2, l_{ij} \sim \text{Bernoulli}(l_{\text{bias}}(q, x_i, x_j)). \end{aligned} \tag{4}$$

In the following, we denote x_{ij} as (x_i, x_j) for simplicity.

Now, we introduce accuracy and fairness measurements that the pairwise ordering method can handle. For accuracy, we focus on Area Under the Curve (AUC) in this paper. The conventional goal in pairwise ordering method is to find a ranking model $h \colon Q \times \mathcal{X} \to \mathbb{R}$ that maximizes the expected AUC:

$$\arg\max_h \mathbb{E}_{q \sim Q} \left[P(h_q(x_i) > h_q(x_j) \mid y_{\text{true}}(q, x_i) > y_{\text{true}}(q, x_j)) \right], \tag{5}$$

We will hide the term $\mathbb{E}_{q \sim Q}$, but we only consider comparisons among relevant items for all following definitions.

Instead of single-mindedly maximizing this accuracy measurement, we include fairness measurements into the evaluation of the ranking model h. In this paper, we focus on constraints for pairwise constraints [2, 21]. However, our method can apply to other constraints (e.g., listwise constraints [25]).

We first introduce the predicted pair label function $\hat{l}$ to present the general class of pairwise constraints. So, let us denote $\hat{l}_q(x_{ij}) \in [0, 1]$ as a probability of the difference between the ranking model output for $h_q(x_i)$ and $h_q(x_j)$. For concretness, $\hat{l}_q(x_{ij}) = \sigma(h_q(x_i) - h_q(x_j))$ where $\sigma(x) = \frac{1}{1+e^{(-x)}}$ is the sigmoid function. This definition of $\hat{l}$ allows us to define the class of pairwise constraints c^{pair} as pointwise constraints Eq. (3). The pairwise constraints c^{pair} may be expressed or approximated as linear constraints on $\hat{l}$. That is, we define the expected bias of h based on a class of constraint for pairwise constraints c^{pair}:

$$\Delta = \mathbb{E}_{x_{ij} \sim \mathcal{P}^2}[\langle \hat{l}_q(x_{ij}), c^{\text{pair}}(q, x_{ij}) \rangle], \tag{6}$$

where $\langle \hat{l}_q(x_{ij}) c^{\text{pair}}(q, x_{ij}) \rangle := \sum_{l_{ij} \in \mathcal{Y}} \hat{l}_q(l_{ij} \mid x_{ij}) c^{\text{pair}}(q, x_{ij}, l_{ij})$ and we use the shorthand $\hat{l}_q(l_{ij} \mid x_{ij})$ to denote the probability of sampling l_{ij} from a Bernoulli random variable with $p = \hat{l}_q(x_{ij})$; i.e., $\hat{l}(1 \mid x_{ij}) := \hat{l}_q(x_{ij})$ and $\hat{l}(0 \mid x_{ij}) := 1 - \hat{l}_q(x_{ij})$. Therefore, a pair label function $\hat{l}$ is unbiased with respect to the constraint function c^{pair} if $\Delta = 0$. If $\hat{l}$ is biased, the degree of bias (positive or negative) is given by Δ.

We define the constraints c^{pair} with respect to a pair of protected group (G_k, G_l), and thus assume access to an indicator function $g_k(x) = 1[x \in G_k]$ where $k \in [K] \subseteq |X|$. We use $Z_{G_{kl}} := \mathbb{E}_{x_{ij} \sim \mathcal{P}^2}[g_k(x_i) \cdot g_l(x_j)]$ to denote the probability of a sample drawn

from $\mathcal{P}^2$ to be in (G_k, G_l). We use $P_{X_{ij}} = \mathbb{E}_{x_{ij}\sim\mathcal{P}^2}[l_{\text{true}}(q, x_{ij})]$ to denote the proportion of $\mathcal{X}^2$ which is positively labelled and $P_{G_{kl}} = \mathbb{E}_{x_{ij}\sim\mathcal{P}^2}[g_k(x_i)\cdot g_l(x_j)\cdot l_{\text{true}}(q, x_{ij})]$ to denote the proportion of $\mathcal{X}^2$ which is positively labelled and in (G_k, G_l). We now give some concrete examples of accepted notions of constraint functions, however, for all constraint functions, $c^{\text{pair}}_{kl}(q, x_{ij}, 0) = 0$:

Definition 1. For any $k \neq l$, we define a pairwise statistical parity constraint that requires that if two items are compared from different groups, then on average each group has an equal chance of being top-ranked. $c^{\text{pair}}_{kl}(q, x_{ij}, 1) = \frac{g_k(x_i)g_l(x_j)}{Z_{G_{kl}}} - 1$.

Definition 2. For any $k \neq l$, we define a pairwise inter-group constraint that requires that if two items are compared from different groups, then on average each group has an equal chance of being top-ranked. We define a pairwise marginal constraint that requires pairs of two items from different groups to be equally-likely to be ranked correctly. $c^{\text{pair}}_{kl}(q, x_{ij}, 1) = l_{\text{true}}(q, x_{ij})\left(\frac{g_k(x_i)g_l(x_j)}{P_{G_{kl}}} - \frac{1}{P_{X_{ij}}}\right)$.

Definition 3. For any $k = l$, we define a pairwise intra-group constraint that requires that if two items are compared from different groups, then on average each group has an equal chance of being top-ranked. We define a pairwise marginal constraint that requires pairs of two items from same groups to be equally-likely to be ranked correctly. $c^{\text{pair}}_{kl}(q, x_{ij}, 1) = l_{\text{true}}(q, x_{ij})\left(\frac{g_k(x_i)g_k(x_j)}{P_{G_{kk}}} - \frac{1}{P_{X_{ij}}}\right)$.

Definition 4. We define a pairwise marginal constraint that requires pairs to be equally-likely to be ranked correctly regardless of the protected group membership of both members of the pair. $c^{\text{pair}}_{kl}(q, x_{ij}, 1) = l_{\text{true}}(q, x_{ij})\left(\frac{g_k(x_i)}{\sum_{l\in[K]} P_{G_{kl}}} - \frac{1}{P_{X_{ij}}}\right)$.

3.3 Pointwise Modeling Biases in Dataset

In this section, we introduce a pre-processing method that is most related to our method, proposed by Jiang and Nachum [16]. [16] assumes that a given biased label function y_{bias} is closest to an ideal unbiased label function y_{true} in terms of KL-divergence D_{KL}.

Definition 5. A pointwise loss function subjected to a pointwise constraint for some $\epsilon_k \in \mathbb{R}$.

$$\begin{aligned} &\arg\min_h \mathbb{E}_{x_i\sim\mathcal{P}}\left[D_{KL}(h_q(x_i)\|y_{\text{true}}(q, x_i))\right] \\ &\text{s.t. } \mathbb{E}_{x_i\sim\mathcal{P}}\left[c^{\text{point}}(q, x_i)\right] = \epsilon_k \text{ for all } k. \end{aligned} \tag{7}$$

According to this KL-divergence loss function with the pointwise constraint, [16] has derived the relationship between y_{bias} and y_{true} based on the standard theorem [3] (see Appendix in [16] for proof).

Proposition 1. *Based on Definition 5, y_{bias} satisfies the following for all $x_i \in \mathcal{X}$.*

$$y_{\text{true}}(y \mid q, x) \propto y_{\text{bias}}(y \mid q, x)\cdot \exp\left(\sum_{k=1}^{K}\lambda_k c^{\text{point}}_k(q, x, y)\right), \tag{8}$$

for some $\lambda_1, \ldots, \lambda_K \in \mathbb{R}$.

Based on Proposition 1, [16] has proposed a pre-processing method to recover y_{true} by re-weighting y_{bias} by the inverse of the second term on the RHS of Eq. (8). In other words, this pre-processing method minimizes the weighted KL-divergence loss function using the re-weighted biased labels.

This pre-processing method is based on the pointwise ordering method. In a ranking problem, the pairwise ordering method that considers the label orders of a given pair of items was found to be more accurate for the task and empirically outperforms the pointwise ordering method [4, 24]. In addition, the pointwise ordering method can not fully optimize the fairness measurements for ranking, such as pairwise fairness [2]. Thus, a pairwise re-weighting method for unbiased pairwise learning should be developed to improve the fair ranking quality from biased labels.

4 METHOD

In this section, we explain our pre-processing method based on the pairwise ordering method for solving fair ranking problems. In Section 4.1, we first define our loss function as a KL-divergence loss function with constraints using the pair label function. According to this loss function, we can derive a closed-form expression for the true pair label function l_{true} in terms of the biased pair label function l_{bias}, the coefficients $\lambda_{11}, \ldots, \lambda_{KK}$, and the constraint functions $c^{\text{pair}}_{11}, \ldots, c^{\text{pair}}_{KK}$. In Section 4.2, we present how to use this closed-form expression to weight the observed data. Subsequently, we present a weighted loss function by reformulating the constrained KL-divergence loss function using these weights. In Section 4.3, we present our algorithm that minimizes this weighted loss function. We show how this minimization is to optimize the trade-off between accuracy and fairness in rankings. Finally, we describe how to extend our method to more general measurements of fairness in rankings (Section 4.4).

4.1 Pairwise Modeling Biases in Dataset

We now introduce our mathematical framework to evaluate bias in pairs of items by derivation of the relationship between l_{true} and l_{bias}. One key point of this derivation is to formulate the constrained KL-divergence loss function using paired dataset $\mathcal{D}_{\text{pair}}$. We formulate the following constrained optimization problem.

Definition 6. Now, we assume that there exist $\epsilon_{kl} \in \mathbb{R}$ such that the observed, biased pair label function l_{bias} is the solution of the following constrained optimization problem:

$$\begin{aligned} &\arg\min_h \mathbb{E}_{x_{ij}\sim\mathcal{P}^2}\left[D_{KL}(\hat{l}_q(x_{ij})\|l_{\text{true}}(q, x_{ij}))\right] \\ &\text{s.t. } \mathbb{E}_{x_{ij}\sim\mathcal{P}^2}[\langle\hat{l}_q(x_{ij}), c^{\text{pair}}(q, x_{ij})\rangle] = \epsilon_{kl} \text{ for all } k, l. \end{aligned} \tag{9}$$

This pairwise ordering method differs from the pointwise one Eq. (7) in terms of sampling items. Also, our approach can directly consider the pairwise constraint, a fairness measurement of ranking, in its loss function. Therefore, Eq. (9) is more suitable to address the fair ranking problem than Eq. (7).

According to the above KL-divergence loss function with constraints, we can derive a closed-form expression for the observed l_{bias} in terms of l_{true} using the same procedure appeared in previous studies [3, 13, 16]. We derive the following relationship.

Proposition 2. *Based on Definition 6, l_{bias} satisfies the following for all $x_{ij} \in \mathcal{X}^2$.*

$$l_{\text{true}}(l_{ij} \mid q, x_{ij}) \propto l_{\text{bias}}(l_{ij} \mid q, x_{ij}) \cdot \exp\left(\sum_{k,l=1}^{K} \lambda_{kl} c_{kl}^{\text{pair}}(q, x_{ij}, l_{ij})\right), \quad \text{for some } \lambda_{11}, \ldots, \lambda_{KK} \in \mathbb{R}. \tag{10}$$

Based on this relationship Eq. (10), we propose a pre-processing method to recover l_{true} by weighting l_{bias} by the inverse of the second term on the RHS of Eq. (10). In other words, this weighting method minimizes the weighted KL-divergence loss function using the weighted observed pair labels. We show this weighted loss function is unbiased for the ground truth in the next section.

4.2 Proposed Unbiased Loss Function

In this section, we present an unbiased loss function, that is, the weighted loss function using the biased pair label function. For simplicity, we first present how a ranking model h may be learned assuming knowledge of the coefficients $\lambda_{11}, \ldots, \lambda_{KK}$. We now have the closed-form expression Eq. (10) for the true pair label function. Based on this expression, we propose a weighting technique to train h on pair labels based on l_{true}. This weighting technique weights an pair of items x_{ij} by the weight $w(x_{ij}, l_{ij})$:

$$w(x_{ij}, l_{ij}) = \frac{\tilde{w}(x_{ij}, l_{ij})}{\sum_{l'_{ij} \in \mathcal{Y}} \tilde{w}(x_{ij}, l'_{ij})}, \tag{11}$$

where

$$\tilde{w}(x_{ij}, l'_{ij}) = \exp\left(\sum_{k,l \in [K]} \lambda_{kl} c_{kl}^{\text{pair}}(q, x_{ij}, l'_{ij})\right). \tag{12}$$

We have the following theorem, which states that training a ranking model based on pairs of items with biased pair labels weighted by $w(x_{ij}, l_{ij})$ is equivalent to training a ranking model on pairs of items labeled according to the true, unbiased pair labels.

Theorem 7. *For any loss function L from paired dataset $\mathcal{D}_{\text{pair}}$, training a ranking model h on the weighted objective $\mathbb{E}_{x_{ij} \sim \mathcal{P}^2, l_{ij} \sim l_{\text{bias}}(q, x_{ij})}\left[w(x_{ij}, l_{ij}) \cdot L(\hat{l}_q(x_{ij}), l_{ij})\right]$ is equivalent to training the ranking model on the objective $C \cdot \mathbb{E}_{x_{ij} \sim \tilde{\mathcal{P}}, l_{ij} \sim l_{\text{true}}(q, x_{ij})}\left[L(\hat{l}_q(x_{ij}), l_{ij})\right]$ with respect to the underlying, true pair labels, for some distribution $\tilde{\mathcal{P}}$ over $\mathcal{X}^2$*

Proof. For a given x_{ij} and for any l_{ij}, based on Proposition 2:

$$w(x_{ij}, l_{ij}) l_{\text{bias}}(l_{ij} \mid q, x_{ij}) = \phi(x_{ij}) l_{\text{true}}(l_{ij} \mid q, x_{ij}), \tag{13}$$

where $\phi(x_{ij}) = \sum_{l'_{ij} \in \mathcal{Y}} w(x_{ij}, l_{ij}) l_{\text{bias}}(l'_{ij} \mid q, x_{ij})$ depends on x_{ij}. Therefore, $\tilde{\mathcal{P}}$ denotes the feature distribution $\tilde{\mathcal{P}}(x_{ij}) \propto \phi(x_{ij}) P(x_{ij})$, we have,

$$\begin{aligned} &\mathbb{E}_{x_{ij} \sim \mathcal{P}^2, l_{ij} \sim l_{\text{bias}}(q, x_{ij})}\left[w(x_{ij}, l_{ij}) \cdot L(\hat{l}_q(x_{ij}), l_{ij})\right] \\ &= C \cdot \mathbb{E}_{x_{ij} \sim \tilde{\mathcal{P}}, l_{ij} \sim l_{\text{true}}(q, x_{ij})}\left[L(\hat{l}_q(x_{ij}), l_{ij})\right], \end{aligned} \tag{14}$$

where $C = E_{x_{ij} \sim \mathcal{P}^2}\left[\phi(x_{ij})\right]$, and this completes the proof. □

Theorem 7 is the modest significant contribution of our study. It states that the bias in observed pair labels can be simply and straightforwardly corrected by weighting the training pairs. Theorem 7 suggests that when we weight the training pairs, we trade off the ability to train on unbiased pair labels to train on a slightly different distribution $\tilde{\mathcal{P}}$ over $\mathcal{X}^2$. However, the change in the feature distribution does not affect the bias of the final learned model when given some mild conditions. In these cases, training using the weighted pairs of items with the biased pair labels is equivalent to training using the same pairs of items but with the true pair labels.

4.3 Determining the Coefficients

We now describe how to learn the coefficients λ. In practice, K^2 is often small in our approach, which is advantageous. Thus, we propose to iteratively learn the coefficients so that the final model satisfies the desired pairwise constraints either on the training data or on a validation set. For simplicity, we first restrict the pairwise constraint as a pairwise statistical constraint. Then we explain how to apply our algorithm to other pairwise constraints.

Intuitively, if the average exposure of G_k is lower than the average exposure of G_l, then the corresponding coefficients should be increased, (i.e., if we increase the weights of the pairs (G_k, G_l) with positive pair labels and decrease the weights of the pairs (G_k, G_l) with negative pair labels, then the ranking model will be encouraged to rank items of G_k higher than items of G_l while items of G_l are placed lower than items of G_k. Both of these two events will cause the difference in average exposure between G_k and G_l to reduce. Thus, $\hat{l}$ will move closer to l_{true}, namely, h will move closer to the true, unbiased ranking model.

Accordingly, Algorithm 1 works by iteratively performing the following steps: (1) Evaluate the pairwise statistical constraint. (2) Update the coefficients by subtracting the respective constraint violation multiplied by a step size. (3) Compute the weights for each pair based on these multipliers using the closed-form provided by Proposition 2. (4) Retrain the ranking model given these weights.

Algorithm 1 employs the pairwise ordering method procedure H, which given a set of pairs from dataset $\mathcal{D}$ and weights w_{ij} to output a ranking model. In practice, H can be any pairwise training procedure that minimizes a weighted loss function over some parametric function class (e.g., [4, 5, 24]).

Our resulting algorithm simultaneously minimizes the weighted loss and maximizes the fairness via learning the coefficients. These processes may be interpreted as competing goals with different objective functions. Thus, it is a form of a non-zero-sum two-player game. The use of non-zero-sum two-player games in fairness was proposed by [10, 16] for classification and [21] for ranking and regression.

4.4 Extension to Other Constraints

In the previous section, we assumed that l_{true} is known, and now we will explain how to approximate it in practice. The initial restriction to pairwise constraints was made so that the values of the constraint functions c_{kl}^{pair} on any $x_{ij} \in \mathcal{X}^2, l_{ij} \in \mathcal{Y}$ would be known. In general, the constraint functions depend on l_{true}, which is unknown. For these cases, we propose applying the same technique of iteratively weighting the loss to satisfy the desired fairness

Algorithm 1 Training a fair ranking model for all of pairwise constraints

Input: Learning rate η, number of loops T, training data $\mathcal{D}$, pairwise learning procedure H, constraints $c_{11}^{\text{pair}}, \ldots, c_{KK}^{\text{pair}}$ corresponding to the pair of protected groups $(G_1, G_1), \ldots, (G_K, G_K)$
Initialize $\lambda_{kl} = 0$ for all k, l and $w_{ij} = 1$ for all i, j
Let $h = H(\mathcal{D}, w_{ij})$
for $t = 1, \ldots, T$ **do**
  Compute fairness violation Δ_{kl} using Eq.(6)
  Update $\lambda_{kl} = \lambda_{kl} - \eta \cdot \Delta_{kl}$ for all k, l
  Let $\tilde{w}_{ij} := \exp\left(\sum_{k,l \in [K]} \lambda_{kl} \mathbb{1}\left[x_{ij} \in (G_k, G_l)\right]\right)$ for all i, j
  Compute weights w_{ij} using Eq. (11)
  Update model $h = H(\mathcal{D}, w_{ij})$
end for
return h

constraint. The weights $w(x_{ij}, l_{ij})$ on each pair are determined only by the pair of protected attributes $g(x_i), g(x_j)$ and the observed pair label $l_{ij} \in Y$. This is equivalent to using Theorem 7 to derive the same procedure presented in Algorithm 1. However, the unknown constraint function $c_{kl}^{\text{pair}}(q, x_{ij}, l_{ij})$ is approximated by a piece-wise constant function $d((g(x_i), y_i), (g(x_j), y_j))$, where $d\colon g(x_i) \times g(x_i) \times \mathcal{Y} \to \mathbb{R}$ is unknown. Although we do not have access to d, we may treat $d((g(x_i), y_i), (g(x_j), y_j))$ as an additional set of parameters (i.e., one for each pair of protected groups attribute $(g(x_i), g(x_j))$ and each pair of labels (y_i, y_j)). These additional parameters may be learned in the same way the λ coefficients are learned. All the fairness metrics introduced in this study do not need any additional parameters. Thus, we can apply other pairwise constraints in Algorithm 1.

5 EXPERIMENT

5.1 Experimental Setup

Datasets. We use three real-world datasets for benchmark fair ranking tasks in our experiments. The datasets are commonly used in fair ranking tasks and are publicly available. We also analyze the datasets in a single query task and a multi-query task.

The first dataset is the COMPAS dataset [22] which consists of a query. The query consists of 6172 individuals. Each individual is described by a feature vector x consisting of 31 attributes with both numerical and categorical features, along with a label risk score classified as recidivism ($y_i = 1$) or not ($y_i = 0$). For this dataset, we consider two groups based on their race attributes (Caucasian, Non-Caucasian).

The second dataset is the Adult dataset [11] which consists of a query. The query consists of 48842 individuals. We preprocess the Adult dataset following [28]. Each individual is described by a feature vector x consisting of 122 attributes with both numerical and categorical features, along with a label of the individual's income with $> 50k$ per year classified as high ($y_i = 1$) and $\leq 50k$ classified as low ($y_i = 0$). We consider the four intersectional groups based on their gender and race attributes (Male-Caucasian, Male-Non-Caucasian, Female-Caucasian, Female-Non-Caucasian).

The third dataset is the Microsoft Learning to Rank (MSLR) Dataset [23] which consists of 2805 queries. Each query has an average of 132 documents. We preprocess the MSLR dataset following [27]. Each document is described by a feature vector x consisting of 135 attributes with numerical and categorical features. and integral relevance scores$[0, 4]$. Each web page is described by a feature vector x consisting of 135 attributes with numerical and categorical features. We use 2 or more relevance scores as relevant ($y_i = 1$) or not ($y_i = 0$). We consider five groups based on their quality scores with the same numbers of items for each group.

We randomly split the queries into training, validation, and test sets with a ratio of $1/2 : 1/4 : 1/4$. The validation set is used to tune the hyperparameters. We evaluate all metrics for individual queries and report the average across queries in the test set.

Baselines. We compare our method against the following four methods: (1) The unconstrained method that minimizes unweighted pairwise loss function to optimize only for accuracy and not for fairness. (2) The re-weighting method [1] that minimizes pointwise loss function Eq.(7) by weighting each label to satisfy the equal of opportunity constraint [16]. (3) The post-processing method [2] on estimated labels from a fairness-oblivious linear regression to satisfy the disparate impact constraint [25]. This method solves a Linear Problem per query with some slack λ in the constraint. (4) The in-processing method [3] that employs the Lagrangian approach and jointly learns the model parameters and Lagrange multipliers to satisfy the pairwise fairness [21]. This method uses a hinge relaxation to make the constraints differentiable and adjusts a slack λ for the constraints.

We employ a linear ranking model h and use Adam [17] for gradient updates in all of the methods in the experiments.

Hyperparameters. We tune the hyperparameters across all of the experiments using a validation set. Using the validation set, we tune the learning rate of Adam within the range of $[10^{-1}, 10^{-4}]$ for all of the methods. For the unconstrained methods, we tune the number of training epochs within the range of $[10, 50]$, and the minibatch size within the range of $[2^8, 2^{11}]$. For the re-weighting method, we fix the learning rate for the coefficients to $\eta = 1$ and tune the number of loops T within the range of $[50, 100]$.For the post-processing method, we chose the smallest slack for its constraint in increments of 10% until the LPs are feasible.For the in-processing method, we fixed the number of training epochs at $2,500$ and we chose the smallest slack for its constraint in increments of 5% until the method returned a non-degenerate solution. Also, we tune the minibatch size within the range of $[2^4, 2^7]$ for the in-processing method. For our method, we fix the learning rate for the coefficients to $\eta = 1$ and tune the number of loops T within the range of $[10, 50]$.

Evaluation Metrics. We use AUC (Eq. (5)) as the accuracy in all our experiments. For all of our experiments, we evaluate our procedures with respect to the pairwise statistical, the pairwise inter, the pairwise intra, and the pairwise marginal constraint. These constraints are computed according to definitions 1 to 4 respectively, but we

[1] https://github.com/google-research/google-research/tree/master/label_bias
[2] https://github.com/ashudeep/Fair-PGRank
[3] https://github.com/google-research/google-research/tree/master/pairwise_fairness

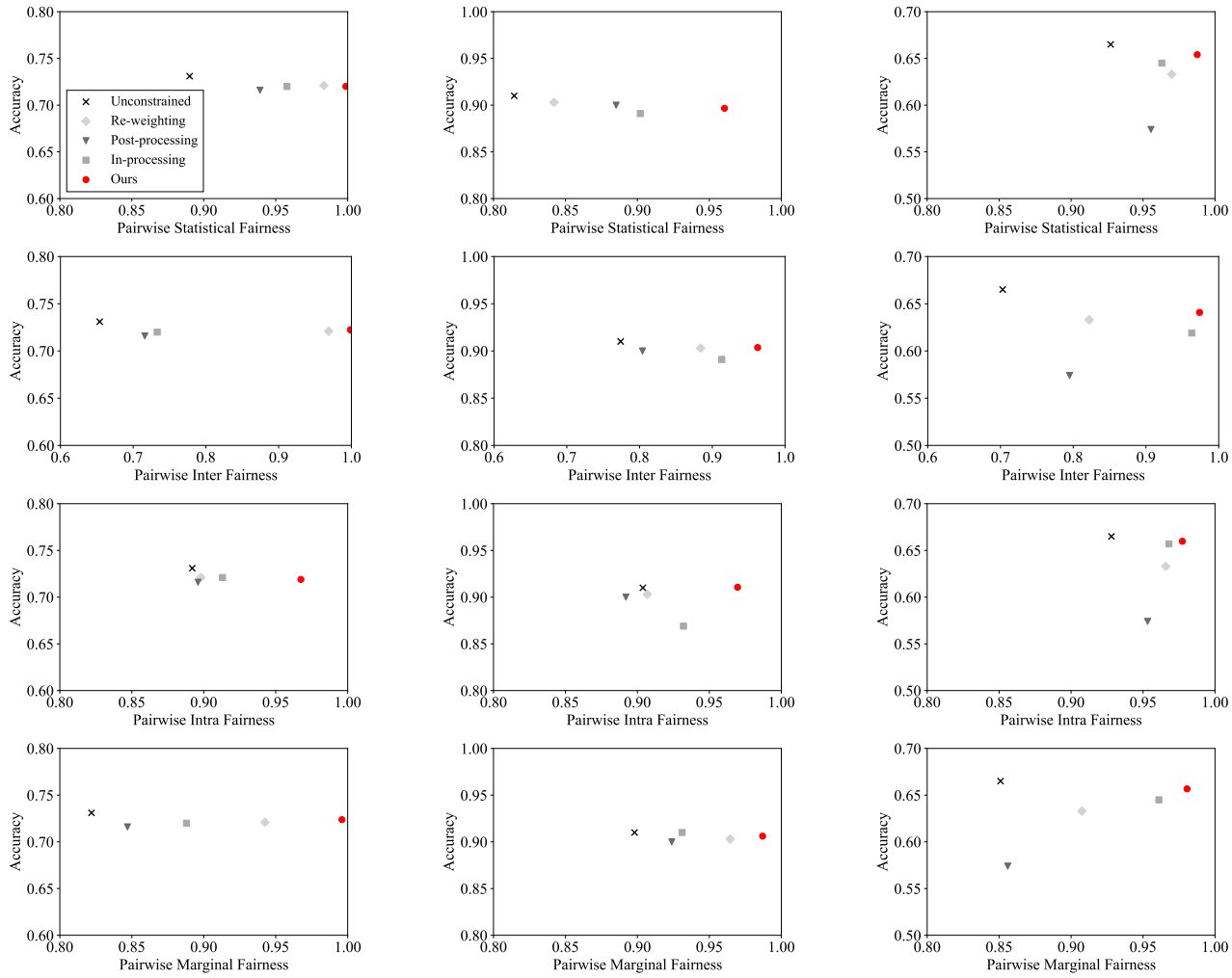


Figure 1: Experiment results for fair ranking tasks: Each row corresponds to a fairness measurement. Each column corresponds to a dataset: Left COMPAS. Middle Adult. Right MSLR. In each graph, we show the best result for five methods: the unconstrained method, the re-weighting method [16], the post-processing method [25], the in-processing method [21], and our method. The best trade-off point is located in the upper right corner of each graph. All reported results are evaluated on the test set.

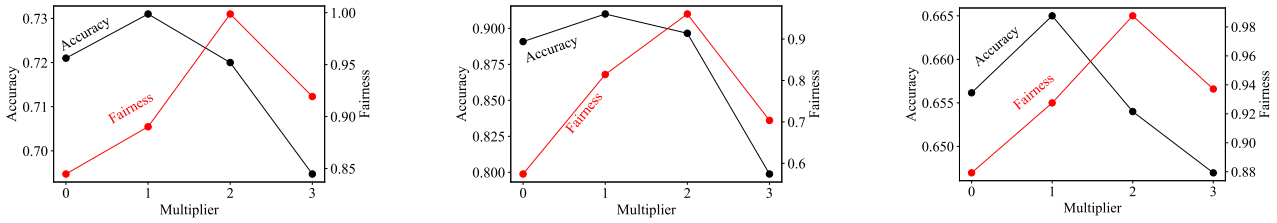
do not have access to the true labels function. Instead, we approximate the true pair label function l_{true} using the observed pair label function l_{bias} in the constraints. Also, we report the fairness of each experiment in the range $[0, 1]$ using the following equation that returns 1 if the model is completely fair.

$$1 - \underbrace{\max\left\{\Delta_{kl} - \Delta_{lk} \mid k, l \in [K]\right\}}_{\text{fairness violation}}. \tag{15}$$

5.2 Results

We present the results in Figure 1. Our method often yields a ranking model with the highest test fairness of the examined methods (Figure 1). The results suggest that the fairness in ranking can be significantly improved by weighting the observed data in the pairwise manner. We also include test accuracy in the results. Our method can effectively trade-off between accuracy and fairness though the major goal of fair methods is to produce fair outputs.

Also, the results highlight the disadvantages of existing methods for producing fair ranking. The re-weighing method often yields a poor trade-off between fairness and accuracy in the MSLR dataset. As the re-weighing method uses the pointwise ordering method that ignores the query-level dataset structure, its loss funcion is dominated by the queries with a large number of items [20]. Moreover, the re-weighing method does not consider the order of items in its loss function and its constraint, resulting in sub-optimal solutions [2, 20]. Although the in-processing method uses the pairwise ordering method, its results do not consistently provide a fair ranking. As the in-processing method requires approximating



(a) COMPAS (b) Adult (c) MSLR

Figure 2: Results of the trade-off curve for our method as the coefficient λ changes. We show the test accuracy and fairness for pairwise statistical constraint as a function of changing in weights.

non-differentiable constraints with a certain amount of slack to make gradient-based training possible, it suffers from overfitting [8]. The post-processing method often failed to improve the fairness well even with tuning its slack for the constraint. In addition, this method performs with poor accuracy because it often requires a large slack in the constraints to solve the LP per query feasibly.

5.3 Results for Changes in Coefficients

We now validate our algorithm for estimating the coefficients. Our method does not require a hyperparameter to adjust the fairness constraint. Instead, we train a model using appropriate weights of the training items generated by the Algorithm 1. Therefore, we investigate whether the weights obtained by this algorithm can improve fairness. To investigate, we take the optimal coefficients $\lambda = \lambda^*$ found by Algorithm 1. Then we train a model on weighted training pairs with changing λ.

In Figure 2, we illustrate the test accuracy and fairness for pairwise statistical constraint as a function of changing in weights. We only show the results of the pairwise statistical constraint as we have similar observations on other constraints. For each constant value, x on the x-axis, and we train a model with the data weights based on the setting $\lambda = x \cdot \lambda^*$. Then we plot the accuracy and fairness.

For $x = 1$, corresponding to our method, the best fairness is achieved (Figure 2). The results suggest that our algorithm can correct the biases appropriately. On the other hand, $x = 0$ that corresponds to training on the unweighted pairs gives us the highest accuracy. Interestingly, for $x = 2$, the results show the lowest accuracy and poor fairness. We believe this is because too many irrelevant items are placed at the top simply as they belong to a socially salient group.

6 CONCLUSIONS

In this paper, we presented a pre-processing method based on the pairwise ordering method for fair ranking. Our pairwise ordering method can identify biases in a pair of groups by assuming that there exists a pair of ground-truth labels.Our method for correcting these biases is based on weighting the training pairs of items. We showed that a ranking model trained on the weighted dataset is unbiased to the ground truth. Experimentally, we showed that the proposed method leads to a fair ranking model in various fairness contexts. These results demonstrate the advantage of our pre-processing method in producing a fair ranking. In future work, we will consider biases of data in a listwise manner rather than in a pairwise manner.

ACKNOWLEDGMENTS

We would like to thank Hiroya Inakoshi for his in-depth feedback on this paper.

REFERENCES

[1] Ricardo Baeza-Yates. 2018. Bias on the web. *Commun. ACM* 61, 6 (2018), 54–61. https://doi.org/10.1145/3209581

[2] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2212–2220. https://doi.org/10.1145/3292500.3330745

[3] Zdravko I Botev and Dirk P Kroese. 2011. The Generalized Cross Entropy Method, with Applications to Probability Density Estimation. *Methodology and Computing in Applied Probability* 13, 1 (2011), 1–27. https://doi.org/10.1007/s11009-009-9133-7

[4] Christopher J. C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report. Microsoft Research. http://research.microsoft.com/en-us/um/people/cburges/tech_reports/MSR-TR-2010-82.pdf

[5] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting Ranking SVM to Document Retrieval. In *The 29th Annual International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 186–193. https://doi.org/10.1145/1148170.1148205

[6] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. In *45th International Colloquium on Automata, Languages, and Programming (LIPIcs, Vol. 107)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 28:1–28:15. https://doi.org/10.4230/LIPIcs.ICALP.2018.28

[7] Wei Chu and S. Sathiya Keerthi. 2005. New Approaches to Support Vector Ordinal Regression. In *Proceedings of the 22nd International Conference on Machine Learning (ACM International Conference Proceeding Series, Vol. 119)*. ACM, 145–152. https://doi.org/10.1145/1102351.1102370

[8] Andrew Cotter, Maya R. Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake E. Woodworth, and Seungil You. 2019. Training Well-Generalizing Classifiers for Fairness Metrics and Other Data-Dependent Constraints. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 1397–1405. http://proceedings.mlr.press/v97/cotter19b.html

[9] Andrew Cotter, Heinrich Jiang, Maya R. Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. 2019. Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals. *J. Mach. Learn. Res.* 20 (2019), 172:1–172:59. http://jmlr.org/papers/v20/18-616.html

[10] Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. 2019. Two-Player Games for Efficient Non-Convex Constrained Optimization. In *Algorithmic Learning Theory (Proceedings of Machine Learning Research, Vol. 98)*. PMLR, 300–332. http://proceedings.mlr.press/v98/cotter19a.html

[11] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness Through Awareness. In *Innovations in Theoretical Computer Science 2012*. ACM, 214–226. https://doi.org/10.1145/2090236.2090255

[13] Michael P. Friedlander and M. R. Gupta. 2006. On minimizing distortion and relative entropy. *IEEE Trans. Inf. Theory* 52, 1 (2006), 238–245. https://doi.org/10.1109/TIT.2005.860448

[14] Anthony G. Greenwald and Linda Hamilton Krieger. 2006. Implicit Bias: Scientific Foundations. *California Law Review* 94, 4 (2006), 945–967. https://doi.org/doi:10.2307/20439056

[15] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29*. 3315–3323. https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

[16] Heinrich Jiang and Ofir Nachum. 2020. Identifying and Correcting Label Bias in Machine Learning. In *The 23rd International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 108)*. PMLR, 702–712. http://proceedings.mlr.press/v108/jiang20a.html

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*. http://arxiv.org/abs/1412.6980

[18] Caitlin Kuhlman, MaryAnn Van Valkenburg, and Elke A. Rundensteiner. 2019. FARE: Diagnostics for Fair Ranking using Pairwise Error Metrics. In *The World Wide Web Conference*. ACM, 2936–2942. https://doi.org/10.1145/3308558.3313443

[19] Ping Li, Christopher J. C. Burges, and Qiang Wu. 2007. McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc., 897–904. https://proceedings.neurips.cc/paper/2007/hash/b86e8d03fe992d1b0e19656875ee557c-Abstract.html

[20] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (2009), 225–331. https://doi.org/10.1561/1500000016

[21] Harikrishna Narasimhan, Andrew Cotter, Maya R. Gupta, and Serena Wang. 2020. Pairwise Fairness for Ranking and Regression. In *The 34th AAAI Conference on Artificial Intelligence*. AAAI Press, 5248–5255. https://aaai.org/ojs/index.php/AAAI/article/view/5970

[22] ProPublica. 2018. Compas Recidivism Risk Score Data and Analysis. https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis

[23] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* abs/1306.2597 (2013). http://arxiv.org/abs/1306.2597

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461. https://dl.acm.org/doi/10.5555/1795114.1795167

[25] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2219–2228. https://doi.org/10.1145/3219819.3220088

[26] Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. In *Advances in Neural Information Processing Systems 32*. 5427–5437. https://proceedings.neurips.cc/paper/2019/hash/9e82757e9a1c12cb710ad680db11f6f1-Abstract.html

[27] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. 2019. Fair Learning-to-Rank from Implicit Feedback. arXiv:1911.08054 [cs.LG]

[28] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*. PMLR, 962–970. http://proceedings.mlr.press/v54/zafar17a.html