Ensuring Demographic Parity and Equalized Odds in Batch Classification

Manjish Pal,[1] **Subham Pokhriyal,** [2] **Niloy Ganguly** [1, 3]

[1] IIT-Kharagpur, India
[2] IIT-Ropar, India
[3] L3S Research Center, Hanover, Germany
manjishpal@iitkgp.ac.in, subham0100@gmail.com, ganguly.niloy@gmail.com

Abstract

In this paper, we develop a framework to achieve fairness notions in situations where selection of a set rather than an element is done. We observe such a phenomenon in recruitment, admission-like circumstances, we term this set selection as a *batch classification* problem. Leveraging this setting, we define a configuration model whereby the acceptance rate of each group can be regulated. We use this configuration based model to ensure two popular notions of fairness namely *demographic parity* and *equalized odds* based on which we make a more fair and comprehensive comparison of our algorithms with several state of the art competing baselines that ensure these criteria of fairness. We observe that configuration based batch-wise post-processing using the *confidence-scores* of a classifier allows simplicity, speed-up, flexibility and improvement in performance over existing baselines. In particular, the configuration model, allows us to efficiently handle the important case of multiple overlapping sensitive attributes. Since our approach performs classification in batches we also compare its performance with state of the art *fair learning-to-rank* and *fair subset selection* algorithms by performing meaningful adaption of these algorithms. We also show that the proposed algorithm can perform fair gerrymandering and provide better accuracy than a competing baseline.

Introduction

In the past years, researchers in fair machine learning have undertaken fair classification, fair ranking and fair subset-selection as important sub-tasks. In fair classification, notions of fairness like disparate impact, demographic parity, equalized odds (Barocas, Hardt, and Narayanan 2019) are implemented in a classification setting where individual records are classified independent of each other. In fair ranking algorithms, one needs to (re)-rank a set of records according to certain judgement score while ensuring fairness criteria. The fair subset-selection algorithms select a set of records that maximize a utility measure along with ensuring fairness. An extensive amount of research has been done in all these directions in the past decade in which the classification algorithms have considered both demographic parity and equalized odds (Kamiran and Calders 2009; Feldman et al. 2015; Chouldechova 2017; Menon and Williamson 2017; Agarwal et al. 2018; Zafar et al. 2019; Padala and Gujar 2020; Ruoss et al. 2020; Lohaus, Perrot, and Von Luxburg 2020; Cho, Hwang, and Suh 2020; Mary, Calauzenes, and El Karoui 2019; Hardt, Price, and Srebro 2016; Romano, Bates, and Candès 2020; Yang, Cisse, and Koyejo 2020; Celis et al. 2019) whereas fair ranking and fair subset-selection have focused on demographic parity based constraints (Zehlike and Castillo 2020; Zehlike et al. 2017; Singh and Joachims 2018; Kuhlman, VanValkenburg, and Rundensteiner 2019; Celis, Straszak, and Vishnoi 2017; Chierichetti et al. 2019; Celis, Huang, and Vishnoi 2017; Mehrotra and Celis 2021). Demographic parity looks into the difference in the acceptance rates among sensitive classes while equalized odds ensure equal TPR (true positive rate) and FPR (false positive rate) across sensitive subpopulations (Barocas, Hardt, and Narayanan 2019).

For a large class of problem settings like recruitment, college admission etc., a set of agents (candidates) are selected simultaneously; we refer this problem as *batch-classification*. In this paper, we tackle the problem of **attaining demographic parity** and **equalized odds** - two important measures of fairness (Barocas, Hardt, and Narayanan 2019) for such a *batch classification*[1,2] framework in the case of multiple and polyvalent sensitive attributes. The batch-classification algorithm shares inherent similarities with the underlying framework of fair classification, ranking and subset selection algorithms.

Batch-classification can be considered as a post-processing step where the entire test dataset is provided as input during classification. Thus, although similar, it differs from the standard *fair classification* setting, which is a point-wise task and elements are expected to be labelled one at a time and independent of other test elements. In a typical *fair ranking* problem, one is given a dataset with n items, each item having certain *judgement score* according to some utility function,

[1]The term *batch-wise classification* has been discussed in papers like (Vural et al. 2009) in which it is defined to mean samples in the same batch are learned and classified collectively. We perform the task of classifying and ensuring the fairness of an entire batch together during *test* time.

[2]There are DNN based algorithms like (Padala and Gujar 2020; Cho, Hwang, and Suh 2020) that rely on batch-wise training, however, here we are talking about batchwise inference during *test* time.

and the problem is to reorder the items and select a few $k << n$ items such that the overall utility of these items is maximised while forcing a constraint like demographic parity. A batch-classification is a relaxed version where the ranks within the selected set are not considered while the underlying problem setting requires much strict fairness constraints. In *fair subset selection* or *fair submodular optimization* a set V of n items (e.g., people), and $S_1, S_2, \ldots, S_m$ be subpopulations corresponding to sensitive attributes are given. A selection of items $S \subseteq V$ is considered to be fair if it satisfies $l_i \leq |S \cap S_i| \leq u_i$ for a given choice of lower and upper bounds $l_i, u_i \in \mathbb{Z}^{\geq 0}$ for $i \in [m]$. The goal is to select a set S such that $|S| = k$ (called *global cardinality constraints*) and maximize $f(S)$ where f is *submodular* w.r.t. S. However, the concept of classification and hence learning is generally absent. A special case of this problem is when $f(S) = \sum_{a \in S} w(a)$ where $w(a) \geq 0$. This case of fair subset-selection naturally leads to an LP-relaxation (Mehrotra and Celis 2021). Our proposed LP framework can be thought as a generalized version of this relaxation that incorporates the concept of classification in the framework and tackles equalized odds constraints along with the demographic parity.

To the best of our knowledge, the general notion of fairness inherently assumes similar acceptance rates across groups. We posit that fairness is a concept bounded by social conditions, e.g. in societies where female representation is minimal, a law ensuring (say) 30% participation of women in public positions can be considered fair and progressive. Hence, an algorithm needs to be independent of any underlying 'social' assumption and should be able to take the social constraints as input. We hence propose the **configuration model** whereby the desirable acceptance rates of different groups can be provided as an input to the algorithm. In order to deal with the configuration model, we consider each group as a set and develop a **linear programming based solution** which minimizes disparity in demographic parity and equalized odds while maintaining accuracy.

We perform **extensive experiments** on various datasets and compare with several state-of-the-art fair classification as well as fair ranking and subset-selection (by performing meaningful adaptations) algorithms for both demographic parity and equalized odds and notice improvement in efficiency at a given configuration. We show, beyond decent performance gains, the flexibility and generalizability of the algorithms to adapt to any given configuration and the capability to ensure fairness in multiple overlapping subpopulations.

Prior and Related Works

The classification algorithms which have been proposed to remove unfairness with respect to demographic parity, equalized odds and other notions from the data can be broadly classified in three groups: pre-processing, in-processing and post-processing based algorithms. **Pre-processing**: The goal is to pre-process the training data such that any classification algorithm trained on this data would generate unfairness-free outcomes. This is usually done by generating fair representations as is done by Feldman et al. (Feldman et al. 2015), Dwork et al. (Dwork et al. 2012), Kamiran and Calders (Kamiran and Calders 2009), Edwards and Storkey (Edwards and Storkey 2015), Madras et al. (Madras et al. 2019), Beutel et al. (Beutel et al. 2017), Ruoss et al. (Ruoss et al. 2020) Rodriguez et al. (Rodriguez-Galvez, Thobaben, and Skoglund 2020) and Zhao et al. (Zhao and Gordon 2022). **In-processing**: Here the idea is to add constraints for fairness to the classification optimization model like SVM; examples - Calders and Verwer (Calders and Verwer 2010), Kamishima et al. (Kamishima, Akaho, and Sakuma 2011), Bechavod and Ligett (Bechavod and Ligett 2017) Bilal Zafar et al. (Zafar et al. 2019), Agarwal et al. (Agarwal et al. 2018), Wu et al. (Wu, Zhang, and Wu 2019), Padala and Gujar (Manisha and Gujar 2018), Zhang et al. (Zhang, Lemoine, and Mitchell 2018), Yurochkin et al. (Yurochkin, Bower, and Sun 2020), Celis et al. (Celis et al. 2019), Roh et al. (Roh et al. 2020), Yang et al. (Yang, Cisse, and Koyejo 2020), Cho et al. (Cho, Hwang, and Suh 2020), Romano et al. (Romano, Bates, and Candès 2020), Mary et al. (Mary, Calauzenes, and El Karoui 2019). **Post-processing** : The third and final strategy consists of first running a standard classifier like SVM or Logistic-regression on the training data and then using the model to mitigate unfairness in the test data. This approach has been used by Hardt et al. (Hardt, Price, and Srebro 2016); Corbett-Davis et al. (Corbett-Davies and Goel 2018); Agarwal et al. (Agarwal et al. 2018) Narasimhan (Narasimhan 2018) and Wei et al. (Wei, Ramamurthy, and Calmon 2021). From the perspective of fair ranking algorithms, some important algorithms are Celis et al. (Celis, Straszak, and Vishnoi 2017), Singh et al. (Singh and Joachims 2018), Zehlike et al. (Zehlike et al. 2017) and Zehlike et al. (Zehlike and Castillo 2020). Out of these, the last one is an in-processing learning to rank algorithm whereas others are re-ranking algorithms without a train-test data. Fair subset selection, although has been predominantly studied in the streaming setting, there are some algorithms which have focussed on the static setting like Mehrotra et al. (Mehrotra and Celis 2021). The Greedy-fair algorithm mentioned in the paper is based on an old paper (Nemhauser, Wolsey, and Fisher 1978), and described more concisely, in the context of demographic parity, in Halabi et al. (Halabi et al. 2020). Since fair batch classification has not been studied per se in the literature, we compare our algorithms with best performing fair classification algorithms, fair ranking and subset selection algorithms. From the very definition of fair batch classification, it is clear that any fair classification algorithm is also a fair batch classification algorithm. Hence we can consider any fair classification algorithm as a baseline for comparison. In contrast, fair ranking and subset selection algorithms need to be adapted to make meaningful comparison with our batch classification algorithms. It is noteworthy that our post-processing approach is significantly different from those of (Agarwal et al. 2018; Hardt, Price, and Srebro 2016; Wei, Ramamurthy, and Calmon 2021). In the former two, the authors use a low dimensional linear constraint space to find a *threshold* τ that can be used on the class probabilities of the base classifier to predict classes whereas the third uses convex (non-linear) optimization to

transform the probabilities so that the transformed scores satisfy fairness criteria.

Demographic Parity and Equalized Odds

Several notions of fairness have been considered in fair machine learning research which include disparate impact, demographic parity, equalized odds, equal opportunity, calibration etc.. In this paper we focus on two most important and widely studied notions of fairness belonging to independence and separation notions of fairness (Barocas, Hardt, and Narayanan 2019). Demographic parity satisfies independence and Equalized Odds (Disparate Mistreatment) satisfies the separation criteria. Informally, demographic parity ensures equal selection rates across different populations in sensitive attributes whereas equalized odds ensures equal TPR and FPR across subpopulations. In the following we consider these notions in more detail.

Demographic Parity

Demographic parity is originally defined considering that the positive group selection rate (which we will refer to as *acceptance rate*) needs to be the same across the two groups of a binary sensitive attribute. This measure is also commonly referred to as statistical parity. Let the population size be n, then in case of single binary sensitive attribute, the set of subpopulations each corresponding to a sensitive value can represented as S = $\{S_1, S_2\}$ where $S_1 \cap S_2 = \phi$ and $S_1 \cup S_2 = n$. The term Demographic Disparity (DDP) [Single] can be accordingly defined as

$$DDP_S \quad = \quad \Big|\mathbb{P}[\hat{Y} = 1|S_2] - \mathbb{P}[\hat{Y} = 1|S_1]\Big|$$

and the system is deemed to have demographic parity when the value is close to zero. The $\mathbb{P}[\hat{Y} = 1]$ refers to the probability of being selected in the positive group.

The sensitive attributes need not be binary but may be multi-valued. For a multi-attribute, multi-valued case, let there be k attributes, with the i^{th} attribute assuming m_i values, then m (= $\sum_{i=1}^{k} m_i$) is the total number of sensitive values the population can assume. Correspondingly, $S = \{S_1, S_2 \dots S_m\}$ is the set of subpopulations each representing a sensitive value. Unlike in the previous case, here the subpopulations are not mutually exclusive. Here Demographic Disparity (DDP) [Multiple] can be defined as (Zafar et al. 2019; Padala and Gujar 2020).

$$DDP_M \quad = \quad \Big|\max_j \mathbb{P}[\hat{Y} = 1|S_j] - \min_j \mathbb{P}[\hat{Y} = 1|S_j]\Big|$$

Equalized Odds

In (Hardt, Price, and Srebro 2016), the notion of equalized odds for single binary sensitive attribute S was proposed based on which we define the Difference of Equalized Odds (DEO) [Single] as follows:

$$\begin{aligned} DEO_S = (&\mathbb{P}[\hat{Y} = 1|Y = 1, S = 1] - \mathbb{P}[\hat{Y} = 1|Y = 1, S = 0]) \\ &+(\mathbb{P}[\hat{Y} = 1|Y = 0, S = 1] - \mathbb{P}[\hat{Y} = 1|Y = 0, S = 0]) \end{aligned}$$

This definition essentially demands that the True Positive Rate (TPR = $\mathbb{P}[\hat{Y} = 1|Y = 1, S = a]$) and the False Positive Rate (FPR = $\mathbb{P}[\hat{Y} = 1|Y = 0, S = a]$) be the same across all the subpopulations ($a \in \{0, 1\}$) of a given sensitive attribute S. To generalize the definition of difference of equalized odds to multiple sensitive attributes we define Difference of Equalized Odds (DEO) [Multiple] for multiple sensitive attributes as follows

$$\begin{aligned} DEO_M \quad = \quad &(\max_j TPR_j - \min_j TPR_j) + \\ &(\max_j FPR_j - \min_j FPR_j) \end{aligned}$$

Let $DTPR_M = (\max_j TPR_j - \min_j TPR_j)$ be the Difference of TPR and $DFPR_M = (\max_j FPR_j - \min_j FPR_j)$ be the Difference of FPR. Thus $DEO_M = DTPR_M + DFPR_M$. We can easily observe that there is a trivial solution to achieve zero DEO_M. Simply assign label 1 to all the records. But this trivial solution may lead to very low accuracy. Similarly a uniformly random 0/1 labelling also leads to zero DEO_M on expectation. Instead, we are looking for labellings that lead to low DEO_M and high accuracy.

Configuration Model

A configuration ('config' in short) refers to the acceptance rates $[\boldsymbol{\beta}] = \{\beta_1, \beta_2 \cdots \beta_m\}$ corresponding to each subpopulation. Normally, it is assumed the acceptance rate is same for all sub-population. However, in real life this may not be the case; in many cases policymakers may decide to have different acceptance rates for different subpopulation.

Relationship of Demographic Parity and Equalized Odds with Configuration Model.

We can assume that there is a desirable $[\boldsymbol{\beta}]$ which is taken as input by an algorithm and the algorithm outputs a $[\boldsymbol{\beta'}]$ which is achieved. The Demographic DisParity [Configuration] in such a setting can be defined as

$$DDP_C([\boldsymbol{\beta}], [\boldsymbol{\beta'}]) = \big\|[\boldsymbol{\beta}] - [\boldsymbol{\beta'}]\big\|_\infty = \max_j |\beta_j - \beta'_j| \quad (1)$$

The equation measures how much deviation a particular configuration has gone through w.r.t. the desirable condition. Under certain constraints, one can establish a relation between DDP_M and DDP_C

If one dissect the definition of Equalized Odds, it has the notion of configuration inbuilt in the definition. According to configuration model, we had defined $\beta_j = \mathbb{P}[\hat{Y} = 1|S = S_j]$. Using Bayes rule of conditional probability we can show the dependence of TPR_j and FPR_j on β_j as

$$\begin{aligned} \mathbb{P}[\hat{Y} = 1|Y = 1, S = S_j] \quad &= \quad \mathbb{P}[\hat{Y} = 1|S = S_j] \cdot \Delta_j \\ \mathbb{P}[\hat{Y} = 1|Y = 0, S = S_j] \quad &= \quad \mathbb{P}[\hat{Y} = 1|S = S_j] \cdot \Gamma_j \end{aligned}$$

where $\Delta_j = \frac{\mathbb{P}[Y=1|\hat{Y}=1,S=S_j]}{\mathbb{P}[Y=1|S=S_j]}$. and $\Gamma_j = \frac{\mathbb{P}[Y=0|\hat{Y}=1,S=S_j]}{\mathbb{P}[Y=0|S=S_j]}$ Similarly we can write the FPR_j in terms of β_j as well and hence can write DEO_M as

$$\begin{aligned} DEO_M \quad = \quad &(\max_j \beta_j \cdot \Delta_j - \min_j \beta_j \cdot \Delta_j) \\ &+(\max_j \beta_j \cdot \Gamma_j - \min_j \beta_j \cdot \Gamma_j) \end{aligned}$$

The linking of a configuration with DEO_M allows us to compare two competing algorithms whereby we measure the DEO_M and accuracy achieved by two algorithms at the same configuration [$\boldsymbol{\beta}$]. Note that fixing a configuration [$\boldsymbol{\beta}$] and minimizing DEO_M doesn't necessarily mean that we are trying to satisfy both DP and EO criteria together (which are otherwise known to be incompatible (Kleinberg, Mullainathan, and Raghavan 2016)).

Minimizing DDP_M and DEO_M - Linear Programming Based Approaches

In this section, we consider the data to be tagged (with ground truth label) and write an LP that can be used to reduce the DDP and DEO while ensuring high accuracy.

Batch Classification: Elements with tag

Given a dataset X with $|X| = n$ elements, s.t. $\forall x_i \in X$ there are tags (unknown ground truth) $y(x_i) \in \{0, 1\}$ and $r(x_i) \in \{0, 1\}$ (known classifier prediction). Let $S = \{S_1, S_2, \ldots, S_m\}$ be the set of subpopulations belonging to all the sensitive attributes, the objective is to find a $0/1$ labelling χ of the records that ensures (a) a given configuration [$\boldsymbol{\beta}$] is attained (DDP constraint), (b) the TPR_j (and FPR_j) w.r.t. tags $y(.)$ are almost same for all the subpopulations S_j (DEO constraint) and (c) maximize $\frac{1}{n} \cdot \sum_i \chi(x_i) y(x_i)$ (accuracy constraint). In the following, we describe a LP based solution to this problem and describe in detail how the above three constraints are ensured.

We propose **LPCA** (**L**inear **P**rogramming framework with **C**onfiguration and **A**ccuracy) and **LPCEO** (**L**inear **P**rogramming framework with **C**onfiguration and **E**qualized **O**dds) which try to satisfy the two constraints (in case of DDP) and equalized odds constraints (in case of DEO). The basic structure of **LPCA** and **LPCEO** is given below. **LPCA** is the linear program that doesn't contain the constraints corresponding to tpr, fpr variables. If we add the constraints that involve the variables tpr and fpr, we obtain **LPCEO** that can be used to ensure the equalized odds fairness constraint. The proposed LP can be considered to be a generalization of the LP relaxation studied in (Mehrotra and Celis 2021).

Maximizing Accuracy. Each element in the set is assigned a binary $\{0, 1\}$ label. Hence to attain higher accuracy while choosing members of one group (say female), primarily those members who have acceptance tags ($y(x_i) = 1$) need to be chosen. However, note the ground truth y is **not** known while the selection is made. So the tags are estimated (predicted) using a classifier, better the classifier, better the estimation. The classifier besides inferring the class (tag) of each point also returns a **confidence value**, we leverage the understanding that the classification error would minimize if one chooses the elements which have been classified with higher confidence values (Kamiran and Calders 2009; Niculescu-Mizil and Caruana 2005) to maximize the accuracy. More specifically, we derive a weight $w_j(a)$ from the confidence value (rank) for every element a in the test data that depends on the subpopulation j (like male, black etc.) and minimize $\sum_{j=1}^{m} \sum_{a \in S_j} \chi(a) w_j(a)$. We put $w_j(a) = r_j$ i.e. the rank (in descending order of confidence values predicted by the classifier, in the subpopulation S_j) of the element a. To bring in uniformity $w_j(a)$ is normalized with the size of the group, hence $w_j(a) = \alpha_j \cdot \frac{r_j}{|S_j|}$, where α_j is a hyperparameter and $|S_j|$ is the number of elements present in that group.

Achieving the desired configuration. We provide [$\boldsymbol{\beta}$] as a hard constraint along with a small value ϵ as a relaxation. (Again experimental results show that in most of the cases, the output deviation from [$\boldsymbol{\beta}$] is smaller than ϵ.) In **LPCA** and **LPCEO**, we put the single sided error.

LPCEO

LPCA

$$\min \sum_{a \in X} \chi(a) w(a)$$

$$\left(\sum_{a \in S_i} \chi(a) \right) \geq \beta_i |S_i| \; \forall S_i \in S$$

$$\left(\sum_{a \in S_i} \chi(a) \right) \leq (\beta_i + \epsilon) |S_i| \; \forall S_i \in S_j$$

$$0 \leq \chi(a) \leq 1 \; \forall a \in X$$

$$\beta_i = (\alpha) \beta_i^{\text{initial}} + (1 - \alpha) \hat{\beta}$$

$$\sum_{a \in S_i; r(a)=1} \chi(a) r(a) \geq tpr_i \cdot \hat{S}_i \; \forall i \in [m]$$

$$\sum_{a \in S_i; r(a)=1} \chi(a) r(a) \leq (tpr_i + \delta_1) \cdot \hat{S}_i \; \forall i \in [m]$$

$$\sum_{a \in S_i; r(a)=0} \chi(a)(1 - r(a)) \geq fpr_i \cdot \hat{S}'_i \; \forall i \in [m]$$

$$\sum_{a \in S_i; r(a)=0} \chi(a)(1 - r(a)) \leq (fpr_i + \delta_2) \cdot \hat{S}'_i \; \forall i \in [m]$$

$$0 \leq tpr_i, fpr_i \leq 1 \; \forall i \in [m]$$

Achieving Low DEO_M. In **LPCEO** the variables tpr_i, fpr_i correspond to the TPR and FPR of the subpopulation S_i and $\hat{S}_i(\hat{S}'_i)$ is an approximate number of 1s (0s) in the test data in the subpopulation S_i (based on classifier's prediction). Since we do not have access to $y(.)$, we use $r(.)$ that is the tag returned by a chosen classifier as a proxy to $y(.)$. Here δ_1, δ_2 are tunable parameters which control the DEO_M of the output configuration $[\boldsymbol{\beta}_\chi]$. More specifically, $DEO_M([\boldsymbol{\beta}_\chi]) \leq \delta_1 + \delta_2$.

Classifier: The accuracy of **LPCA** and **LPCEO** would

depend on the efficiency of the underlying classifier. In this paper we assume that the base classifier is a Random Forest (RF) and provide our results according to its *confidence scores*. We get similar results if we use Logistic Regression, SVM or a DNN based classifier. All results are obtained using fixed 70%/30% train/test splits of all the datasets.

Experimental Setup

In this section, we describe the experimental setup that includes a description of various datasets, baselines performance metrics and configuration generation to compare with baselines. **Reproducibility**: All the codes and datasets are available at https://github.com/alphaaccount/fair-batch-classification.

Datasets

In our study we have used four real datasets namely Adult (UCI 1996), Bank (UCI 2012), COMPAS (ProPublica Recidivism) (ProPublica.Org 2019), German (UCI 1994) and a synthetic dataset for evaluating the performance of the algorithms. The number of instances and classes in each attribute is written within the bracket below. First we describe the real world datasets as follows:

Adult (48,842 examples) Here the task is to predict whether someone makes more than $50k per year, with gender(2) and race(5) as the protected attribute.

Bank (41,188 examples) Each example has 20 features and the target variable is whether the client has subscribed to the term deposit service or not. We have taken age group(2) and marital status(4) (MS) as the sensitive attributes.

ProPublica recidivism (7,918 examples) In ProPublica's COMPAS recidivism data, the task is to predict recidivism from someone's criminal history, jail and prison time, demographics, and COMPAS risk scores, with race(2) and gender(2) as the protected attributes.

German (1,000 examples) The German credit dataset contains attributes such as personal status and sex, credit score, credit amount, housing status etc. It can be used in studies about gender inequalities on credit-related issues. The sensitive attribute being gender(2) and age(2).

Baselines

In fair classification, there has been certain works that have focused only on the equalized odds criteria (Hardt, Price, and Srebro 2016; Romano, Bates, and Candès 2020; Awasthi, Kleindessner, and Morgenstern 2020). In fair ranking, most of the works have focused on demographic parity (Zehlike et al. 2017; Zehlike and Castillo 2020; Singh and Joachims 2018, 2017) [3]. In fair subset selection, research has only been done towards demographic parity constraint in both streaming (Wang, Fabbri, and Mathioudakis 2021; Halabi et al. 2020) and static setting (Celis, Straszak, and Vishnoi 2017; Mehrotra and Celis 2021).

Demographic Parity. We have chosen four recent and most competitive classification baselines which can tackle multiple attributes and multiple values together. These algorithms are (a). **Agarwal et al.** (Agarwal et al. 2018), (b). **Zafar et al.** (Zafar et al. 2019), (c). **Padala et al.** (Manisha and Gujar 2018), (d). **Yang et al,** (Yang, Cisse, and Koyejo 2020) and (e) **Madras et al.** (Madras et al. 2018) [4]. For comparison with ranking we have chosen **DELTR** (Disparate Exposure in Learning to Rank) algorithm of **Zehlike et al.** (Zehlike and Castillo 2020). From fair subset-selection algorithm, LP-relaxation studied in **Mehrotra et al.** (Mehrotra and Celis 2021) and the Greedy-Fair algorithm in fair submodular maximization described in **Halabi et al.** (Halabi et al. 2020) are considered.

The **DELTR** algorithm takes as input a training data with binary sensitive attribute and a set of judgement scores $s(a)$ associated with each record a and learns a ranking function that can be used to assign scores to a test data such that demographic parity is minimized and certain utility measure (measured in terms of cross-entropy loss) is maximized. To adapt DELTR to our setting we provide the input $s(a) = 1$ if $RF(a) = 1$ and 0 otherwise where RF is the chosen classifier (Random Forest in this paper). We adapt the LP-relaxation of (Mehrotra and Celis 2021) by taking $w(a) = 1$ if $RF(a) = 1$ and 0 otherwise where all positively selected elements are given equal weightage in **Mehrotra**. In the **Greedy-Fair** algorithm, we adapt the greedy algorithm for fair submodular optimization by considering the special case when $f(S) = \sum_{a \in S} w(a)$ (which is the objective function of **LPCA** assuming $\chi(a) \in \{0, 1\}$). The greedy algorithm returns a set S^* such that $f(S^*)$ is an 1/2-approximate answer while ensuring the demographic parity constraints.

Equalized Odds We have chosen seven baselines (all classification as we didn't find any baselines related to ranking or assignment) which can tackle multiple values together for the case of comparative analysis for equalized odds. These algorithms are (a). **Agarwal et al.** (Agarwal et al. 2018), (b). **Zafar et al.** (Zafar et al. 2019), (c). **Padala et al.** (Manisha and Gujar 2018), (d). **Romano et al.** (Romano, Bates, and Candès 2020), (e) **Cho et al.** (Cho, Hwang, and Suh 2020), (f) **Mary et al.** (Mary, Calauzenes, and El Karoui 2019) and (g) **Hardt et al.** (Hardt, Price, and Srebro 2016).

Deriving a configuration

As mentioned, one of the strengths of **LPCA** and **LPCEO** is the ability to generate configurations based on the choice of the users. We generate configurations using a tunable parameter α. Let's assume that there is an initial configuration $[\boldsymbol{\beta}^{\text{initial}}]$ which can be either dictated by the output of the underlying classifier or can be any arbitrary input provided by the user. Also, there is a target uniform configuration where each class has $\hat{\beta}$ acceptance rate. In many situations, one may not want to equalize the acceptance rate as it may heavily affect the accuracy of the system. Hence the tunable parameter α, using which an interpolation is done, the classes whose acceptance rate are higher than $\hat{\beta}$ is downward

[3] However, there are certain papers which implement EO constraints in the case of pairwise error metrics (Kuhlman, VanValkenburg, and Rundensteiner 2019). These topics are excluded from our study.

[4] The algorithms of Madras et al. only work for single binary sensitive attribute

interpolated and vice versa. In **LPCA** the DDP at certain α can be written as $DDP_M([\boldsymbol{\beta}_\alpha]) = \alpha DDP_M([\boldsymbol{\beta}^{\text{initial}}])$ and $DDP_C([\boldsymbol{\beta}_\alpha], [\hat{\boldsymbol{\beta}}]) = \alpha DDP_C([\boldsymbol{\beta}^{\text{initial}}], [\hat{\boldsymbol{\beta}}])$. In case of **LPCEO**, we can write $DEO_M([\boldsymbol{\beta}_\alpha]) = \alpha DEO_M([\boldsymbol{\beta}^{\text{initial}}]) + (1-\alpha) \cdot \hat{\beta} \cdot \Delta$ where Δ depends on $[\hat{\boldsymbol{\beta}}]$.

Configurations for Baselines: Similarly, we observe that all the baseline algorithms have some tunable parameters changing which we can generate different configurations with varying DDPs. These parameters are: *multiplicative covariance factor* $f \in [0, 1]$ (Zafar et al.), *difference-bound* $\in [0, 1]$ (Agarwal et al.), Lagrangian multiplier $\lambda \in [0, B_1]$ (Padala et al.) and primal-dual parameter $\delta \in [0, B_2]$ (Yang et al.). For each of these parameters, we divide its range into three equal parts and take the end points to generate four different configurations. For example, we take $f = 0, 0.33, 0.66, 1$ to generate configurations of Zafar et al. The least DDP_M configuration is attained at one of the end-points of the range of these parameters. For instance, it is attained at $f = 1$ for Zafar et al., *difference-bound* = 0 for Agarwal et al. etc.. However, we must reiterate that we cannot input any arbitrary configuration in these cases.

We also generate configurations from the equalized-odds implementation of the various baselines by tuning parameters namely : *DCCP parameters* $(\tau, \mu > 0)$ (Zafar et al.), *difference-bound* (Agarwal et al.), Lagrangian multiplier (Padala et al.), varying random seeds (Cho et al., Romano et al., Hardt et al.), varying epochs (Mary et al.). We compare configuration-wise DEO_M and accuracy of **LPCEO** with these baselines discussed in next section.

Experimental Evaluation

In this section, we first check the efficacy of the linear programming framework by seeing the extent to which **LPCA** produces DDP when given a configuration as input. Then we compare the performances of **LPCA** and **LPCEO** with various baselines.

Comparison with Baselines (DDP_M)

We present the results by considering two different types of configurations. First we consider the configuration where the DDP is the minimum achievable by an algorithm and then we consider four configurations for each baseline and present **LPCA**'s performance with respect to those configurations.

Minimum DDP_M. We here present a comparative analysis of the lowest DDP_M values and the corresponding accuracies achieved by the baselines and **LPCA** in Table 1. The lowest value is achieved by tuning the individual parameters as mentioned in the previous section (page 5, Deriving a configuration). We find that **LPCA** outperforms almost all the algorithms (performing on multiple overlapping subpopulations) by one or two order of magnitude in terms of attaining minimum DDP_M. Among the baselines, Yang performs the best and produces the best result in the Bank Dataset. Since DELTR and Greedy-fair are implemented on single binary sensitive attribute their least DDP values is better than that of **LPCA** in Table 1. Also for the case of DDP_M, the framework of Mehrotra et al. is same as that of **LPCA** except the choice of $w(a)$, hence we observe results close to that of **LPCA** . The accuracies are comparable across baselines, **LPCA** performing a bit better. For **LPCA** the least DDP_M obtained corresponds to $\alpha = 0, \hat{\beta} = 0.2, 0.05, 0.45$ and 0.7 for Adult, Bank, COMPAS and German datasets respectively.

Different configurations. For each baseline algorithm, we generated four configurations by regulating a tunable parameter (page 5, Deriving a configuration). Subsequently, taking those four configurations as input, we run **LPCA** and collect the results. We have to resort to this for a fair comparison.

Comparison with Baselines (DEO_M)

We perform configuration based comparison of DEO_M with various baselines. For each of the baselines that can cater to equalized odds criteria of fairness, we find the configurations $[\boldsymbol{\beta}]$ corresponding to their final output and use that $[\boldsymbol{\beta}]$ in **LPCEO** to return output configurations $[\boldsymbol{\beta'}]$ and compare DEO_M of the baseline and that of **LPCEO**. Note **LPCEO** ensures that $DDP_C([\boldsymbol{\beta}], [\boldsymbol{\beta'}]) \leq \epsilon$ for all the cases. We have considered 4 different configurations for each baseline. We observe that none of the baselines except Agarwal et al. and Yang et al. can cater to multiple overlapping sensitive attributes. Among the chosen baselines, the algorithms of Zafar et al., Romano et al. and Padala et al. can only deal with single binary sensitive attribute while the rest can cater to single sensitive attribute with multiple subpopulations. Thus, in our comparisons we use binary and non-binary sensitive attributes according to the ability of the baselines. In Table 2, we show the average difference (over 4 configurations) of DEO_M and accuracy between **LPCEO** and other baselines. In the multiple overlapping subpopulations case, we observe that Agarwal et al. can achieve slightly better DEO_M (at cost of accuracy) in the datasets except Adult whereas **LPCEO** performs better w.r.t DEO_M when compared with Yang et al. in datasets other than Adult. In case of single sensitive attribute with multiple subpopulations, Mary et al. performs better on Adult whereas **LPCEO** performs better than Cho et al. and Hardt et al. in terms of DEO_M. In the case of single binary sensitive attribute, **LPCEO** performs better than Zafar et al,, Padala et. al. and Romano et al. w.r.t. DEO_M. In almost all the cases **LPCEO** performs better in terms of test accuracy, although the improvement is modest.

Fair Gerrymandering.

The problem of fair gerrymandering was stated by Kearns et al. (Kearns et al. 2018) to show that there could be large hidden subgroups in a particular data for which fairness may not naturally flow even if the overall system is fair. This problem although cursorily mentioned by Zafar et al. (Zafar et al. 2019), has been really emphasized by Yang et al. (Yang, Cisse, and Koyejo 2020) who construct these gerrymandering subgroups and realise low DDP_M. For the case of two sensitive attributes (which is the case with most of our datasets) S_1 and S_2 containing k_1 and k_2 subpopulations respectively, the number of *gerrymandering* groups according

	Adult		Bank		COMPAS		German	
Baseline	DDP_M	Accuracy	DDP_M	Accuracy	DDP_M	Accuracy	DDP_M	Accuracy
Zafar	0.1074	0.8357	0.0656	0.9027	0.0668	0.6559	0.0841	0.74
Agarwal	0.0711	0.8035	0.0335	0.9049	0.0251	0.6344	0.0895	0.75
Yang	0.018	0.7812	0.0069	0.8935	0.0356	0.5580	0.0655	0.7333
Padala	0.0658	0.8031	0.02	0.8735	0.0154	0.6256	0.0396	0.69
Mehrotra	0.0049	0.8061	0.0198	0.9044	0.0028	0.63	0.0025	0.7066
Madras	0.0332	0.8358	0.0403	0.9075	0.0403	0.6616	0.0534	0.7200
DELTR	0.0034	0.6529	0.0048	0.8291	0.0049	0.4981	0.0312	0.6066
Greedy-Fair	0.0004	0.56	0.0688	0.68	0.0014	0.5033	0.0073	0.4733
LPCA	**0.0049**	**0.8444**	0.008	**0.908**	**0.0009**	**0.6723**	0.0075	**0.7533**

Table 1: The *minimum* DDP_M achieved by various baselines along with the corresponding **test accuracies** for various datasets. In almost all the datasets, the least DDP_M and the highest accuracy in that configuration is achieved by **LPCA**. The results of DELTR (ranking) and Greedy-fair (subset-selection) are obtained for single (binary) sensitive attribute their values and hence are not compared with other algorithms performing on multi-attribute case.

	Adult		Bank		COMPAS		German	
Baseline	D_{DEO_M}	$\boldsymbol{D_{Acc}}$	D_{DEO_M}	$\boldsymbol{D_{Acc}}$	D_{DEO_M}	$\boldsymbol{D_{Acc}}$	D_{DEO_M}	$\boldsymbol{D_{Acc}}$
Zafar	0.1239	0.0354	0.3199	0.2866	0.083	0.0197	0.0695	-0.0022
Agarwal	0.0678	0.0332	-0.0992	0.0068	-0.0284	-0.0181	0.0360	0.0183
Padala	0.06281	0.0282	NA	NA	0.0089	0.0085	0.0165	0.0078
Yang	-0.081	0.0448	0.2817	0.0031	0.0893	-0.0032	0.0649	0.0167
Romano	0.1175	0.0360	0.1430	0.015	-0.0109	0.0052	0.0726	0.0092
Mary	0.0387	-0.0048	0.0602	0.0652	0.0224	0.0227	0.041	0.0025
Cho	0.0883	0.005	NA	NA	0.0311	0.0167	NA	NA
Hardt	0.0389	0.0071	NA	NA	-0.04	0.014	0.0003	0.0092

Table 2: Average Difference (over 4 configurations) of DEO_M and Accuracy between various baselines and **LPCEO**. Positive values imply that **LPCEO** is performing better. Zafar, Padala and Romano can handle only single binary sensitive attributes. NA entries refer to a scenario in which the baseline is giving trivial classification as output (all 1's or all 0's) that results in $DEO_M = 0$ and hence **LPCEO** also attains the same accuracy and DEO_M at that configuration.

to the definition of Yang is $k_1 \cdot k_2 + k_1 + k_2$. Thus, in all there are 17, 14, 8 and 8 gerrymandering groups for Adult, Bank, COMPAS and German datasets respectively of which only 13 and 10 have been considered for the Adult and Bank datasets respectively due to the small size of other groups. By regulating the tunable parameters of the algorithm, we could generate several configurations with varying DDP_M, We took those configurations as input and run **LPCEO**. **LPCEO** is able to realize all the configurations, and we compare the performance of **LPCEO** in terms of accuracy and DEO_M with Yang et al. and present the result in Table 3, We find for the same DDP_M, **LPCEO** (**LPCA**) has a much better accuracy. The DEO_M results are evenly distributed with **LPCEO** performing particularly good in COMPAS.

		Yang		**LPCEO**	
Baseline	DDP_M	Accuracy	DEO_M	Accuracy	DEO_M
Adult	0.0683	0.7899	0.2716	0.828	0.3261
Bank	0.0632	0.9016	0.1323	0.9073	0.1272
COMPAS	0.1135	0.6564	0.113	0.6581	0.1366
German	0.1343	0.7498	0.2071	0.7683	0.2207

Table 3: Comparison of test accuracies, and DEO_M between Yang et al. (Yang, Cisse, and Koyejo 2020) and **LPCEO** averaged over four configurations generated by Yang on various datasets. The number of gerrymandering groups for Adult, Bank, COMPAS and German datasets are 13, 10, 8 and 8 respectively.

Conclusion

The primary contribution is, however, identifying the presence of a special but widespread batch-admission-like situation where batch classification is a natural operation. This decoding of an apparently obvious real-life setting helps us to design a simple LP-based algorithm that is being able to compete and perform better than sophisticated classification algorithms. The main contribution lies in uncovering important advantages which can be derived when we consider the batch classification setting. We carefully generalize the definition of demographic parity and equalized odds for multiple sensitive attributes. These definitions help us to develop the LP framework which enables the generation of the desired configuration, be it expressed in terms of demographic disparity, average acceptance rate or a simple externally defined distribution of acceptance rates. The configuration based LP framework provides us a lot of flexibility to handle the case of multiple overlapping subpopulations efficiently and deal with the interesting problem of fair gerrymandering. Most importantly, it helps us to compare more accurately diverse fair classification, ranking, and subset selection algorithms that implement the demographic parity and equalized odds constraints on various datasets. As part of future work, it would be interesting to see how our framework can be applied to notions of fairness like counterfactual fairness, calibration, etc. which are quite different from independence and separation based notions of fairness.

References

Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; and Wallach, H. 2018. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*.

Awasthi, P.; Kleindessner, M.; and Morgenstern, J. 2020. Equalized odds postprocessing under imperfect group information. In *International Conference on Artificial Intelligence and Statistics*, 1770–1780. PMLR.

Barocas, S.; Hardt, M.; and Narayanan, A. 2019. *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

Bechavod, Y.; and Ligett, K. 2017. Learning fair classifiers: A regularization-inspired approach. *arXiv preprint arXiv:1707.00044*, 1733–1782.

Beutel, A.; Chen, J.; Zhao, Z.; and Chi, E. H. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*.

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

Celis, L. E.; Huang, L.; Keswani, V.; and Vishnoi, N. K. 2019. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, 319–328.

Celis, L. E.; Huang, L.; and Vishnoi, N. K. 2017. Multiwinner voting with fairness constraints. *arXiv preprint arXiv:1710.10057*.

Celis, L. E.; Straszak, D.; and Vishnoi, N. K. 2017. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvtiskii, S. 2019. Matroids, matchings, and fairness. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2212–2220. PMLR.

Cho, J.; Hwang, G.; and Suh, C. 2020. A fair classifier using kernel density estimation. *Advances in Neural Information Processing Systems*, 33: 15088–15099.

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

Corbett-Davies, S.; and Goel, S. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 214–226.

Edwards, H.; and Storkey, A. 2015. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268.

Halabi, M. E.; Mitrović, S.; Norouzi-Fard, A.; Tardos, J.; and Tarnawski, J. 2020. Fairness in Streaming Submodular Maximization: Algorithms and Hardness. *NeurIPS*.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29: 3315–3323.

Kamiran, F.; and Calders, T. 2009. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, 1–6. IEEE.

Kamishima, T.; Akaho, S.; and Sakuma, J. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, 643–650. IEEE.

Kearns, M.; Neel, S.; Roth, A.; and Wu, Z. S. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, 2564–2572.

Kleinberg, J.; Mullainathan, S.; and Raghavan, M. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

Kuhlman, C.; VanValkenburg, M.; and Rundensteiner, E. 2019. Fare: Diagnostics for fair ranking using pairwise error metrics. In *The World Wide Web Conference*, 2936–2942.

Lohaus, M.; Perrot, M.; and Von Luxburg, U. 2020. Too relaxed to be fair. In *International Conference on Machine Learning*, 6360–6369. PMLR.

Madras, D.; Creager, E.; Pitassi, T.; and Zemel, R. 2018. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, 3384–3393. PMLR.

Madras, D.; Creager, E.; Pitassi, T.; and Zemel, R. 2019. Fairness through causal awareness: Learning causal latent-variable models for biased data. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 349–358.

Manisha, P.; and Gujar, S. 2018. FNNC: Achieving Fairness through Neural Networks. *arXiv preprint arXiv:1811.00247*.

Mary, J.; Calauzenes, C.; and El Karoui, N. 2019. Fairness-aware learning for continuous attributes and treatments. In *International Conference on Machine Learning*, 4382–4391. PMLR.

Mehrotra, A.; and Celis, L. E. 2021. Mitigating Bias in Set Selection with Noisy Protected Attributes. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 237–248.

Menon, A. K.; and Williamson, R. C. 2017. The cost of fairness in classification. *arXiv preprint arXiv:1705.09055*.

Narasimhan, H. 2018. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, 1646–1654.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming*, 14(1): 265–294.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of*

the 22nd international conference on Machine learning, 625–632.

Padala, M.; and Gujar, S. 2020. FNNC: Achieving Fairness through Neural Networks. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2277–2283.

ProPublica.Org. 2019. Propublica Risk Assessment.

Rodriguez-Galvez, B.; Thobaben, R.; and Skoglund, M. 2020. A Variational Approach to Privacy and Fairness. *arXiv preprint arXiv:2006.06332*.

Roh, Y.; Lee, K.; Whang, S.; and Suh, C. 2020. FR-Train: A mutual information-based approach to fair and robust training. In *International Conference on Machine Learning*, 8147–8157. PMLR.

Romano, Y.; Bates, S.; and Candès, E. J. 2020. Achieving Equalized Odds by Resampling Sensitive Attributes. *arXiv preprint arXiv:2006.04292*.

Ruoss, A.; Balunovic, M.; Fischer, M.; and Vechev, M. 2020. Learning certified individually fair representations. *Proc. 34th Annual Conference on Advances in Neural Information Processing Systems (NeurIPS 2020).*

Singh, A.; and Joachims, T. 2017. Equality of opportunity in rankings. In *Workshop on Prioritizing Online Content (WPOC) at NIPS*, 31.

Singh, A.; and Joachims, T. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2219–2228.

UCI. 1994. German Dataset.

UCI. 1996. Adult Dataset.

UCI. 2012. Bank Dataset.

Vural, V.; Fung, G.; Krishnapuram, B.; Dy, J. G.; and Rao, B. 2009. Using local dependencies within batches to improve large margin classifiers. *The Journal of Machine Learning Research*, 10: 183–206.

Wang, Y.; Fabbri, F.; and Mathioudakis, M. 2021. Fair and Representative Subset Selection from Data Streams. In *Proceedings of the Web Conference 2021*, 1340–1350.

Wei, D.; Ramamurthy, K. N.; and Calmon, F. P. 2021. Optimized Score Transformation for Consistent Fair Classification. *Journal of Machine Learning Research*, 22(258): 1–78.

Wu, Y.; Zhang, L.; and Wu, X. 2019. On Convexity and Bounds of Fairness-aware Classification. In *The World Wide Web Conference*, 3356–3362.

Yang, F.; Cisse, M.; and Koyejo, S. 2020. Fairness with Overlapping Groups. *arXiv preprint arXiv:2006.13485*.

Yurochkin, M.; Bower, A.; and Sun, Y. 2020. Training individually fair ML models with sensitive subspace robustness. *8th International Conference on Learning Representations, ICLR*.

Zafar, M. B.; Valera, I.; Gomez-Rodriguez, M.; and Gummadi, K. P. 2019. Fairness Constraints: A Flexible Approach for Fair Classification. *Journal of Machine Learning Research*, 20(75): 1–42.

Zehlike, M.; Bonchi, F.; Castillo, C.; Hajian, S.; Megahed, M.; and Baeza-Yates, R. 2017. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1569–1578.

Zehlike, M.; and Castillo, C. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*, 2849–2855.

Zhang, B. H.; Lemoine, B.; and Mitchell, M. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 335–340.

Zhao, H.; and Gordon, G. J. 2022. Inherent Tradeoffs in Learning Fair Representations. *Journal of Machine Learning Research*, 23(1427): 1–26.