FairWASP: Fast and Optimal Fair Wasserstein Pre-processing

Zikai Xiong[1], Niccolò Dalmasso[2,*], Alan Mishler[2], Vamsi K. Potluru[2], Tucker Balch[2], Manuela Veloso[2]

[1]Massachusetts Institute of Technology
[2]J.P. Morgan AI Research, New York
zikai@mit.edu, {niccolo.dalmasso, first.last}@jpmchase.com

Abstract

Recent years have seen a surge of machine learning approaches aimed at reducing disparities in model outputs across different subgroups. In many settings, training data may be used in multiple downstream applications by different users, which means it may be most effective to intervene on the training data itself. In this work, we present FairWASP, a novel pre-processing approach designed to reduce disparities in classification datasets without modifying the original data. FairWASP returns sample-level weights such that the reweighted dataset minimizes the Wasserstein distance to the original dataset while satisfying (an empirical version of) demographic parity, a popular fairness criterion. We show theoretically that integer weights are optimal, which means our method can be equivalently understood as duplicating or eliminating samples. FairWASP can therefore be used to construct datasets which can be fed into any classification method, not just methods which accept sample weights. Our work is based on reformulating the pre-processing task as a large-scale mixed-integer program (MIP), for which we propose a highly efficient algorithm based on the cutting plane method. Experiments demonstrate that our proposed optimization algorithm significantly outperforms state-of-the-art commercial solvers in solving both the MIP and its linear program relaxation. Further experiments highlight the competitive performance of FairWASP in reducing disparities while preserving accuracy in downstream classification settings.

Introduction

Machine learning is increasingly involved in decision making that impacts people's lives (Sloane, Moss, and Chowdhury 2022; Zhang et al. 2022). There is concern that models may inherit from the data bias against subgroups defined by race, gender, or other protected characteristics. Accordingly, there is a vast literature on methods to make machine learning models "fair." While there is no consensus about what it means for a model to be fair or unfair in a given setting, these methods commonly aim to minimize disparities in model outputs or model performance across different subgroups.

Fair machine learning methods are traditionally divided into three categories: (i) *pre-processing* methods intervene on the training data, (ii) *in-processing* methods apply constraints or regularizers during the model training process itself, and (iii) *post-processing* methods alter the outputs of previously trained models. See Hort et al. (2022) for a recent review of methods across all three categories.

Among these three, no one category of methods clearly dominates the others in terms of performance. Pre-processing methods are useful when the person who generates or maintains a dataset is not the same as the person who will be using it to train a model (Feldman et al. 2015), or when a dataset may be used to train multiple models. These methods typically require no knowledge of downstream models, so they are in principle compatible with any subsequent machine learning procedure.

Many pre-processing methods operate by changing the feature values or labels of the training data (Calders, Kamiran, and Pechenizkiy 2009; Žliobaite, Kamiran, and Calders 2011), subsampling or oversampling the data (Kamiran and Calders 2010; Yan, Kao, and Ferrara 2020; Chakraborty, Majumder, and Menzies 2021; Salazar et al. 2021), and/or generating synthetic data (Xu et al. 2018; Salimi et al. 2019). In high-stakes settings such as finance and healthcare, however, it may be unethical or even illegal to alter customer or patient attributes or labels, e.g. with current data regulations in the European Union (GDPR) or health information (HIPAA) in the United States. Maintaining separate, modified versions of datasets is possible but may be costly for large datasets. An alternative is to learn a set of sample weights that can be passed to a learning method at training time (Calders, Kamiran, and Pechenizkiy 2009; Chai and Wang 2022; Jiang and Nachum 2020; Li and Liu 2022). While there are many methods that do this, they focus on satisfying fairness constraints without providing guarantees about how much they alter the distribution of the data.

Contributions We present FairWASP, a novel pre-processing method that learns a set of sample weights for classification datasets without modifying the training data. FairWASP minimizes the Wasserstein distance between the original and reweighted datasets while ensuring that the reweighted dataset satisfies (an empirical version of) demographic parity, a popular fairness criterion, which we detail in Section . Our contributions are as follows:

1. Since directly solving the target optimization problem is

*Corresponding Author

computationally infeasible, we provide a three-step reformulation that leads to a tractable linear program in Section . We prove that the solution to this linear program is a solution to the original problem under a mild assumption and show theoretically that, over the set of real-valued weights, integer-valued weights are in fact optimal. This means that FairWASP can be understood equivalently as indicating which samples (rows) of the dataset should be duplicated or deleted at training time, so it is compatible with any downstream classification algorithm, not just algorithms that accept sample weights.

2. We contribute a highly efficient algorithm to solve the reformulated linear program (Section), that vastly outperforms state-of-the-art commercial solvers.
3. We extend FairWASP to satisfy a separate but equivalent definition of demographic parity (Section) by leveraging the linear program reformulation above.
4. We empirically show that FairWASP achieves competitive performance in reducing disparities while preserving accuracy in downstream classification settings when compared to existing pre-processing methods (Section).

See the Supplementary Materials for complete proofs of theoretical claims, more discussion, and details on our algorithm and experiments results.

Background

Setup Consider a dataset of n i.i.d. samples $\{Z_i = (D_i, X_i, Y_i)\}_{i=1}^n$ drawn from a joint distribution $p_Z = p_{D,X,Y}$ with domain $\mathcal{Z} = \mathcal{D} \times \mathcal{X} \times \mathcal{Y}$. In this context, D represents one or more protected variables such as gender or race, X indicates additional features used for decision-making, and Y is the decision outcome. For example, Y_i could represent a loan approval decision for individual i, based on demographic data D_i and credit score X_i. Learning tasks typically aim at learning the conditional distribution $P(Y|X)$ or $P(Y|X, D)$ from the samples $\{Z_i\}_{i=1}^n$. In this paper, we assume that the number of demographic classes $|\mathcal{D}|$ and the number of outcome levels $|\mathcal{Y}|$ are significantly smaller than n.

Demographic Parity (DP) Demographic parity (DP), also known as statistical parity, requires an outcome variable to be statistically independent of a sensitive feature (Dwork et al. 2012). This could mean, for example, that an algorithm used to screen resumes for interviews is required to recommend equal proportions of female and male applicants. DP is arguably the most widely studied fairness criterion to date (Hort et al. 2022). Violations of DP may be measured in different ways. For FairWASP, we adopt a measure similar to Dwork et al. (2012) and Calmon et al. (2017), namely the distances between the marginal distribution of an outcome variable and the distributions of that outcome variable conditional on levels of a sensitive feature. Additionally, we show in Section that measuring DP as the distance between outcome distributions for each level of the sensitive feature (Calmon et al. 2017) can also be reformulated in a similar way as the FairWASP optimization problem.

Pre-processing via Reweighting Calders, Kamiran, and Pechenizkiy (2009) proposed utilizing a set of sample weights based on the sensitive feature and the outcome variable to target DP. Since then, a variety of papers have utilized similar reweighting approaches (Kamiran and Calders 2010; Jiang and Nachum 2020; Chai and Wang 2022; Li and Liu 2022). However, previous papers provide no guarantees about how the sample weights will change the overall distribution of the data. If the weights alter the distribution of the data significantly, the downstream model might not learn the correct conditional distribution between target variables and features, i.e., $P(Y|X)$ or $P(Y|X, D)$. While minimizing data perturbation has been considered in pre-processing papers which seek to learn transformations of the data itself (Zemel et al. 2013; Calmon et al. 2017), to our knowledge, FairWASP is the first reweighting approach that seeks to minimize the overall distributional distance from the original data.

Wasserstein Distance The general Wasserstein distance (or optimal transport metric) between two probability distributions (μ, ν) supported on a metric space $\mathcal{X}$ is defined as the optimal objective of the (possibly infinite-dimensional) linear program (LP):

$$\mathcal{W}_c(\mu, \nu) \overset{\text{def.}}{=} \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y) \mathrm{d}\pi(x, y), \qquad (1)$$

where $\Pi(\mu, \nu)$ is the set of couplings composed of joint probability distributions over the product space $\mathcal{X} \times \mathcal{X}$ with marginals (μ, ν). Equation (1) is also called the Kantorovitch formulation of optimal transport (Kantorovitch 1958). Here, $c(x, y)$ represents the "cost" to move a unit of mass from x to y. A typical choice in space $\mathcal{X}$ with metric $d_{\mathcal{X}}$ is $c(x, y) = d_{\mathcal{X}}(x, y)^p$ for $p \geq 1$, and then $\mathcal{W}_c^{1/p}$ is referred to as the p-Wasserstein distance between probability measures. Using the Wasserstein distance between distributions is particularly useful as it provides a bound for functions applied to samples from those distributions. In other words, define the following deviation:

$$d(\mu, \nu) \overset{\text{def.}}{=} \sup_{f \in \mathcal{F}} |\mathbb{E}_{z\sim\mu} f(z) - \mathbb{E}_{z\sim\nu} f(z)| \ ,$$

where $\mathcal{F}$ is a family of functions f. If $\mathcal{F} = \mathit{Lip}_1$, the class of Lipschitz-continuous functions with Lipschitz constant of 1, then the deviation $d(\mu, \nu)$ is equal to the 1-Wasserstein distance (Santambrogio 2015; Villani et al. 2009). Analogous bounds can be derived for the 2-Wasserstein distance when $\mathcal{F} = \left\{f \mid \|f\|_{\mathcal{S}^1(\mu)} \leq 1\right\}$, the class of functions with unitary norm over the Sobolev space $\mathcal{S} = \left\{f \in L^2 \mid \partial_{x_i} f \in L^2\right\}$ (Claici, Genevay, and Solomon 2018). This fact provides a theoretical intuition for downstream utility preservation, i.e., the closer two distributions are in Wasserstein distance, the more similar the downstream performance of learning models trained on such distributions is expected to be. Finally, the Wasserstein distance has been used to express fairness constraints in several in-processing methods (Chzhen et al. 2020; Chzhen and Schreuder 2022). To our knowledge, however, it has not previously been used in a pre-processing setting.

FairWASP Optimization Problem

In this section we propose FairWASP, which casts dataset pre-processing as an optimization problem that aims at minimizing the distance to the original data distribution while satisfying fairness constraints.

Given a dataset $Z = \{(D_i, X_i, Y_i)\}_{i=1}^n$, we can write the reweighted distribution of the dataset as:

$$p_{Z;\theta} \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} \theta_i \delta_{Z_i},$$

with $\{\theta_i\}_{i\in[n]}$ such that $\sum_i \theta_i = n$, and Dirac measures δ_{Z_i} centered on Z_i. Here $[n] \overset{\text{def.}}{=} \{1, 2, \ldots, n\}$. Note that the empirical distribution of the original dataset can be written in the form above by setting $\theta_i = e_i = 1$ for any i, i.e., $p_{Z;e} = \frac{1}{n}\sum_{i=1}^n e_i \delta_{Z_i}$. We will use e to represent the n-vector with all entries being 1. We use the Wasserstein distance between $p_{Z;\theta}$ and $p_{Z;e}$ to measure the discrepancy between the original and reweighted datasets. To control for discrimination, we adopt the fairness constraints proposed by Calmon et al. (2017), which are equivalent to imposing demographic parity over the original dataset. In our formulation, this translates to requiring the conditional distribution for all possible values of D under the weights $\{\theta_i\}_{i\in[n]}$ to closely align with the marginal distribution over Y in the original dataset, which we denote p_Y,

$$J\left(p_{Z;\theta}(Y = y|D = d), p_Y(y)\right) \le \epsilon,\ \forall\, d \in \mathcal{D}, y \in \mathcal{Y} \quad (2)$$

where $J(\cdot,\cdot)$ denotes a distance function between scalars. We will use the shorthand $p_{Z;\theta}(y|d)$ for $p_{Z;\theta}(Y = y|D = d)$. This definition corresponds to the enforcing demographic parity by constraining the selection rates across groups $D = d$ to be equal to the overall selection rate. However, unlike Calmon et al. (2017) who defined $J(p, q)$ as $|\frac{p}{q} - 1|$, we define J as the subsequent symmetric probability ratio measure:

$$J(p, q) = \max\left\{\frac{p}{q} - 1, \frac{q}{p} - 1\right\}. \quad (3)$$

We believe our definition is more practical and theoretically sound because it is symmetric with respect to p and q. We note that the two definitions are equivalent when $p > q$ and similar when p is not much smaller than q.

Our proposed approach FairWASP finds *integer* weights $\{\theta_i\}_{i\in[n]}$ via solving the following optimization problem:

$$\begin{aligned} \min_{\theta\in\mathcal{I}^n\cap\Delta_n}\ & \mathcal{W}_c(p_{Z;\theta}, p_{Z;e}) \\ \text{s.t.}\ & J\left(p_{Z;\theta}(y|d), p_Y(y)\right) \le \epsilon,\ \forall\, d \in \mathcal{D}, y \in \mathcal{Y}, \end{aligned} \quad (4)$$

where $\mathcal{I}^n$ is the set of integer vectors in $\mathbb{R}^n$, and Δ_n is the set of valid weights $\{\theta \in \mathbb{R}_+^n : \sum_{i=1}^n \theta_i = n\}$. The use of integer weights can be understood simply as duplicating or eliminating samples in the original datasets. This is in contrast with other approaches such as Kamiran and Calders (2012) and Bachem, Lucic, and Krause (2017), in which the sample-level weights are *real-valued*. The problem of solving the optimal real-valued weights is instead as follows:

$$\begin{aligned} \min_{\theta\in\Delta_n}\ & \mathcal{W}_c(p_{Z;\theta}, p_{Z;e}) \\ \text{s.t.}\ & J\left(p_{Z;\theta}(y|d), p_Y(y)\right) \le \epsilon,\ \forall\, d \in \mathcal{D}, y \in \mathcal{Y}. \end{aligned} \quad (5)$$

Note that (5) is in fact an LP relaxation of (4). In practice, using real-valued weights requires either (i) resampling each sample proportionally to its weight, which introduces statistical noise in the reweighted distribution, or (ii) including sample weights in the loss function during the learning process. Using integer weights, however, ensures the constructed dataset has exactly the optimal reweighted distribution (in the sense of (4) and (5)), and the reweighted dataset can be fed into any classification method, not just methods which accept sample weights. In addition, Theorem 3 and Lemma 4 show that using integer weights achieves the optimal value of the objective in the optimization problem for real-valued weights, i.e., the optimal solution of (4) is also an optimal solution for (5).

Reformulations of the Optimization Problem

In this section, we provide a computationally tractable equivalent formulation of (4). In Step 1, we reformulate (4) as a mixed-integer program (MIP). However, directly solving this problem is infeasible due to its scale. In Step 2, we demonstrate that, through specific reformulations, the dual of the LP relaxation becomes more computationally manageable. In Step 3, we prove that the solution of the dual problem can lead to an optimal solution of (4).

Step 1: Reformulating (4) as a MIP

First, we show that the constraint (2) can be reformulated as linear constraints on θ of the form $A\theta \ge \mathbf{0}$. The conditional probability in constraint (2) can be rewritten as

$$p_{Z;\theta}(y|d) = \frac{\sum_{i\in[n]:d_i=d,y_i=y}\theta_i}{\sum_{i\in[n]:d_i=d}\theta_i}.$$

By substituting the definition of the distance $J(\cdot,\cdot)$ from (3), the fairness constraints equivalently become linear constraints on $\{\theta_i\}_{i=1}^n$ (via inverting a fractional linear transformation), taking the following form for all $d \in \mathcal{D}, y \in \mathcal{Y}$:

$$\begin{aligned} \textstyle\sum_{i\in[n]:d_i=d,y_i=y}\theta_i &\le (1+\epsilon)\cdot p_Y(y)\cdot\sum_{i\in[n]:d_i=d}\theta_i\,, \\ \textstyle\sum_{i\in[n]:d_i=d,y_i=y}\theta_i &\ge \tfrac{1}{1+\epsilon}\cdot p_Y(y)\cdot\sum_{i\in[n]:d_i=d}\theta_i\,. \end{aligned} \quad (6)$$

In total, (6) defines $2|\mathcal{Y}||\mathcal{D}|$ linear constraints on θ in the format of $A\theta \ge \mathbf{0}$, where A is a $2|\mathcal{Y}||\mathcal{D}|$-row matrix[1].

Regarding the objective, the Wasserstein distance can be equivalently formulated as a linear program with n^2 variables (Peyré, Cuturi et al. 2019). Let $C \in \mathbb{R}^{n\times n}$ represent the matrix formed by the transportation costs, i.e., $C_{ij} = c(z_i, z_j)$. Then, according to definition (1), the objective function $\mathcal{W}_c(p_{Z;\theta}, p_{Z;e})$ is given by the optimal objective of the following problem:

$$\min_{P\in\mathbb{R}^{n\times n}} \langle C, P\rangle \ \text{ s.t. } Pe = e,\ P^\top e = \theta, P \ge \mathbf{0}_{n\times n} \quad (7)$$

where $\langle\cdot,\cdot\rangle$ is the Frobenius inner product and recall that $e = 1$ is the vector of ones. Hence, the integer-weight opti-

[1]Note that when Y is binary, e.g., $\mathcal{Y} = \{0, 1\}$, half of the linear constraints induced by (2) are redundant and can be removed.

mization problem in (4) is equivalent to the following MIP:

$$\begin{aligned} \min_{\theta\in\mathbb{R}^n, P\in\mathbb{R}^{n\times n}} \quad & \langle C, P\rangle \\ \text{s.t.} \quad & Pe = e,\ P^\top e = \theta, P \geq \mathbf{0}_{n\times n} \\ & \theta \in \mathcal{I}^n \cap \Delta_n,\ A\theta \geq \mathbf{0} \end{aligned} \tag{8}$$

Similarly, the real-valued weights problem in (5) is equivalent to the following LP:

$$\begin{aligned} \min_{\theta\in\mathbb{R}^n, P\in\mathbb{R}^{n\times n}} \quad & \langle C, P\rangle \\ \text{s.t.} \quad & Pe = e,\ P^\top e = \theta, P \geq \mathbf{0}_{n\times n} \\ & \theta \in \Delta_n,\ A\theta \geq \mathbf{0} \end{aligned} \tag{9}$$

Note that (9) is actually also the LP relaxation of (8). However, this reformulation is not yet practically useful as problem (9) involves $O(n^2)$ variables, which poses a challenge for both conventional LP algorithms and state-of-the-art MIP methods, such as the LP based branch-and-bound methods (Gurobi Optimization, LLC 2023).

Step 2: Dual Problem of the LP Relaxation

In this step, we propose a solution of the LP relaxation (9) by considering its dual problem.

First, note that some constraints are currently redundant. For any feasible (θ, P), θ already lies in Δ_n; given a feasible P, we have (i) $\theta = P^\top e$, (ii) $e^\top e = n$ and (iii) $Pe = e$, so it follows that $\theta^\top e = e^\top P e = e^\top e = n$. Consequently, we can replace θ with Pe and reformulate (9) equivalently as:

$$\min_{P\in\mathbb{R}^{n\times n}} \langle C, P\rangle \ \text{ s.t. } Pe = e,\ P \geq \mathbf{0}_{n\times n},\ AP^\top e \geq \mathbf{0}\ . \tag{P}$$

Therefore, the optimal $\theta^\star$ of (9) can be reconstructed from the optimal $P^\star$ of (P) using $\theta^\star = (P^\star)^\top e$.

Second, we use a property of LP problems to reformulate (P). When the feasible set of the LP problem (P) is nonempty and the optimal solution $P^\star$ exists, $P^\star$ is part of a saddle point of the saddle-point problem on the Lagrangian,

$$\min_{P\in S_n} \max_{\lambda\in\mathbb{R}^m_+} L(P,\lambda) \overset{\text{def.}}{=} \langle C, P\rangle - \lambda^\top A P^\top e \tag{PD}$$

where $S_n \overset{\text{def.}}{=} \{P \in \mathbb{R}^{n\times n} : Pe = e,\ P \geq \mathbf{0}_{n\times n}\}$. Since $L(\cdot,\cdot)$ is bilinear, the minimax theorem (Du and Pardalos 1995) guarantees that (PD) is equivalent to $\max_{\lambda\in\mathbb{R}^m_+} \min_{P\in S_n} L(P,\lambda)$. This is then equal to the dual:

$$\max_{\lambda\geq 0} -F(\lambda), \text{ where } F(\lambda) \overset{\text{def.}}{=} \max_{P\in S_n} \left\langle \bar{C}, P \right\rangle \tag{D}$$

where $\bar{C} = \sum_{j=1}^m \lambda_j e a_j^\top - C$ and $a_j^\top$ is the j-th row of A.

Unlike the problem in (9), the dual problem (D) can be directly solved, as we show in Lemma 1 below.

Lemma 1. *For function $G(\bar{C}) \overset{\text{def.}}{=} \max_{P\in S_n}\langle \bar{C}, P\rangle$, it is a convex function of $\bar{C}$ in $\mathbb{R}^{n\times n}$. It has the following function value and subgradient. For each $i \in [n]$, let $\bar{c}_{ij_i^\star}$ denote a largest component on the i-th row of $\bar{C}$, then $G(\bar{C}) = \sum_{i=1}^n c_{ij_i^\star}$. Define the components of of $P^\star \in \mathbb{R}^{n\times n}$ as*

$$p_{ij} = \begin{cases} 0\,, & \text{if } j \neq j_i^\star \\ 1\,, & \text{if } j = j_i^\star \end{cases} \tag{10}$$

and then $P^\star \in \arg\max_{P\in S_n}\langle \bar{C}, P\rangle$ and $P^\star \in \partial G(\bar{C})$.

Proof Sketch. The proof directly uses the convexity of the maximium LP's optimal objective on the cost function. The problem can be divided into independent separate smaller LP on simplexes, each having a closed-form maximizer. □

Due to the chain rule, Lemma 1 shows that $F(\lambda)$ is convex and the function values and subgradients of $F(\lambda) = G(\sum_{j=1}^m \lambda_j e a_j^\top - C)$ can be computed as well. This implies (D) is equivalent to

$$\min_{\lambda\in\mathbb{R}^m_+} F(\lambda)\ , \tag{D-2}$$

whose objective function $F(\cdot)$ is a convex function of λ (see Lemma 1). Here m is the number of rows in matrix A, which as shown before is at most $2|\mathcal{Y}||\mathcal{D}| \ll n$. Reformulation (D-2) is important as it makes it possible to use methods that need only subgradients of the dual problem (D), such as the subgradient descent method and the cutting plane method (Nesterov 2018), as we show below in Section .

Finally, we consider the implications for the uniqueness of the primal optimal solution $P^\star$.

Assumption 1. *The problem $\min_{P\in S_n}\langle C - \sum_{j=1}^m \lambda_j^\star e a_j^\top, P\rangle$ has a unique minimizer for the optimal solution $\lambda^\star$ for* (D).

Corollary 2. *Under Assumption 1, the primal optimal solution $P^\star$ given by Lemma 1 is the unique maximizer of $\max_{P\in S_n}\langle \bar{C}, P\rangle$.*

Proof. Once the optimal $\lambda^\star$ of (D) is computed, using Assumption 1 the optimal solution $P^\star$ of (P) then lies in $\arg\min_{P\in S_n} L(P,\lambda^\star)$, or equivalently $\arg\max_{P\in S_n}\langle \sum_{j=1}^m \lambda_j^\star e a_j^\top - C, P\rangle$. □

Assumption 1 ensures there are no ties when calculating the row-wise max in the $\bar{C}$ matrix. Ties occur only for $\bar{C}$ in a set of measure zero, as the set of $\bar{C}$ such that $\max_{P\in S_n}\langle \bar{C}, P\rangle$ has multiple maximizers is the $\bar{C}$ with a row containing two or more largest components, which is of a strictly smaller dimension than the full space and thus zero measure. In practice, Assumption 1 holds almost always due to rounding errors and the termination tolerance when computing the optimal solution $\lambda^\star$.

Step 3: Using the Dual Solution to Solve the Original MIP

In this section, we show how to recover the optimal $P^\star$ and $\theta^\star$ of (9) given the optimal solution $\lambda^\star$ of (D). The following theorem demonstrates that the optimal solution $(\theta^\star, P^\star)$ of the LP (9) recovered in this manner is also optimal for the MIP (8).

Theorem 3. *Let $\lambda^\star$ be an optimal dual solution of* (D) *and let Assumption 1 hold. $P^\star$ is an optimal primal solution obtained through Lemma 1 using the form of* (10)*. Then it holds that $\theta^\star = (P^\star)^\top e$ and $P^\star$ are optimal solutions for both the LP* (9) *and the MIP* (8)*.*

Proof Sketch. The proof uses the fact that problems (9) and (8) have the same objective function while the feasible set of (8) is smaller than that of (9), so if an optimal solution of (9) is also feasible for (8), then it is optimal for (8) as well. □

Algorithm 1: General Cutting Plane Method for (D-2)

1: Choose a bounded set E_0 containing an optimal solution
2: **for** k from 0 to n **do**
3: Choose λ^k from E_k
4: Compute $g \in \mathbb{R}^m$ such that

$$g^\top \lambda^k \geq g^\top \lambda^\star \text{ for any } \lambda^\star \in \Lambda^\star$$

5: Choose $E_{k+1} \supseteq \{\lambda \in E_k : g^\top \lambda \leq g^\top \lambda^k\}$
6: **end for**

Theorem 3 shows that once (9) is solved by the dual problem (D), then (8) can be solved immediately. Finally, we can then also conclude that the solutions found by FairWASP are optimal even among real-valued weights.

Lemma 4. *When Assumption 1 holds, the optimal integer-weight solution of* (4) *is as good as the optimal real-valued-weight solution of* (5)*.*

Cutting Plane Method for the Reformulated Problem

The cutting plane method (Khachiyan 1980) is a class of methods for convex problems in settings where the *separation oracle* is accessible. For any $\lambda \in \mathbb{R}^m$ in problem (D-2), a separation oracle is an operator that returns a vector g such that $g^\top \lambda \geq g^\top \lambda^\star$ for any $\lambda^\star \in \Lambda^\star$, where $\Lambda^\star$ denotes the set of optimal solutions. The cutting plane method iteratively makes use of the separation oracle to restrict the feasible sets until convergence[2]. Algorithm 1 shows a pseudo-code breakdown of the cutting plane algorithm; variants of the cutting plane methods differ in the implementation of lines 3 and 5 (see Nesterov 2018 for more details).

For the problem (D-2), a separation oracle (line 4 in Algorithm 1) can be obtained from the subgradients, which are efficiently accessible according to Lemma 1. Corollary 5 below provides an analysis of both time and space complexity; see Supplementary Material A for more details on the separation oracle, implementation, and the proof.

Corollary 5. *With efficient computation and space management, the cutting plane method is able to solve the problem* (D-2) *within* $\tilde{O}\left(n^2 + |\mathcal{D}|^2|\mathcal{Y}|^2 n \cdot \log(R/\varepsilon)\right)$ *flops and* $O(n|\mathcal{D}||\mathcal{Y}|)$ *space.*[3]

Comparison with Other LP Algorithms Table 1 compares theoretical complexities and convergence rates of our cutting plane method implementation with the traditional simplex method and interior point method, as well as a recently proposed practical first-order method (Applegate et al. 2022, 2021) based on the primal-dual hybrid gradient (PDHG). Other LP algorithms, such as (Wang et al. 2022), are only for problems with special structures. Note that the original LP problem (9) has $O(n)$ constraints and $O(n^2)$ nonnegative variables, which scale badly with large values of n. Table 1 has already considered the benefit of sparse matrix multiplication; see Section for an empirical comparison of the computational efficiency of our cutting plane algorithm against existing commercial solvers, and Supplementary Material A for more details on the comparison.

Method	Conv.	Time		Space
		Init.	Per Iter.	
Ours	Fast	$O(n^2)$	$O(n\|\mathcal{D}\|\|\mathcal{Y}\|)$	$O(n\|\mathcal{D}\|\|\mathcal{Y}\|)$
Simplex	Slow	$O(n^2)$	$O(n^3)$	$O(n^2)$
IPM	Fast	$O(n^2)$	$O(n^3)$	$O(n^2)$
PDHG	Slow	$O(n^2)$	$O(n^2)$	$O(n^2)$

Table 1: Convergence speeds and complexity of different LP algorithms.

FairWASP-PW: Extension to Pairwise Demographic Parity Constraints

As pointed out in Calmon et al. (2017), demographic parity can be expressed in multiple equivalent forms. In particular, we can rewrite (2) to constrain the selection rates to be (approximately) equal across groups $D = d$, rather than constraining them to be equal to the marginal distribution of Y in the original dataset:

$$J\left(p_{Z;\theta}(y|d_1), p_{Z;\theta}(y|d_2)\right) \leq \epsilon,\ \forall d_1, d_2 \in \mathcal{D}, y \in \mathcal{Y}\ . \tag{11}$$

This turns the optimization problem in (4) into:

$$\begin{aligned} &\min_{\theta \in \mathcal{I}^n \cap \Delta_n} \mathcal{W}_c(p_{Z;\theta}, p_{Z;e}) \\ &\text{s.t. } J\left(p_{Z;\theta}(y|d_1), p_{Z;\theta}(y|d_2)\right) \leq \epsilon,\ \forall d_1, d_2 \in \mathcal{D}, y \in \mathcal{Y}\ . \end{aligned} \tag{12}$$

This section introduces FairWASP-PW, which extends FairWASP to constraints (11). We show how to solve (12) by (i) pointing out a connection between constraints (2) and (11), (ii) reformulating problem (12) and connecting it to problem (4) ,and (iii) solving (12) via zero-th order optimization.

(1) Connection between constraints (2) **and** (11)**.** For any $|\mathcal{Y}|$-vector $t \in [0,1]^{|\mathcal{Y}|}$ denoting the marginal distribution $p_Y(y)$ in (2), let $\Theta_{\epsilon;t}$ denote the θ that satisfies the fairness constraint (2):

$$\Theta_{\epsilon;t} \overset{\text{def.}}{=} \{\theta \in \Delta_n : J\left(p_{Z;\theta}(y|d), t\right) \leq \epsilon,\ \forall\, d \in \mathcal{D}, y \in \mathcal{Y}\}. \tag{13}$$

Hence, the feasible sets of (4) under constraint (2) is $\mathcal{I}^n \cap \Theta_{\epsilon;\bar{t}}$, where $\bar{t}_y = p_Y(y)$. As for the feasible set of problem (12) under constraint (11), define

$$\Theta_\epsilon \overset{\text{def.}}{=} \left\{\theta \in \Delta_n : \begin{array}{c} J\left(p_{Z;\theta}(y|d_1), p_{Z;\theta}(y|d_2)\right) \leq \epsilon, \\ \forall\ d_1, d_2 \in \mathcal{D}, y \in \mathcal{Y} \end{array}\right\}, \tag{14}$$

obtaining $\mathcal{I}^n \cap \Theta_\epsilon$ as the corresponding feasible set.

The following lemma shows how the feasible set for problem (4) is a subset of problem (12)'s feasible set. More specifically, Θ_ϵ is equal to the union of $\Theta_{\bar{\epsilon};\bar{t}}$ for all $\bar{t} \in [0,1]^{\mathcal{Y}}$ and a certain $\bar{\epsilon}$.

[2] In our case, convergence is achieved when the gap between the primal and dual problems is lower than a given tolerance.

[3] We use the notation $\tilde{O}(\cdot)$ to hide m, n, $|\mathcal{D}|$, and $|\mathcal{Y}|$ in the logarithm function. Here R denotes the maximum norm of the optimal solutions of (D-2)

Lemma 6. *Let* $\Theta_{\epsilon;t}$ *and* Θ_ϵ *be defined as* (13) *and* (14)*, then it holds that for any* $\epsilon \in [0, 1)$, $\Theta_\epsilon = \bigcup_{t\in[0,1]^{\mathcal{Y}}} \Theta_{\bar{\epsilon};t}$*, in which* $\bar{\epsilon} = \sqrt{1+\epsilon} - 1$.

Proof Sketch. Inclusion from both sides can be shown via constructing an element of each set respectively. □

Note that Θ_ϵ is not convex, as the union of convex sets is not necessarily convex, making problem (12) not convex.

(2) Reformulation of Problem (12)**.** Using Lemma 6, we can rewrite problem (12) as:

$$\min_{\theta\in\mathbb{R}^n} \mathcal{W}_c(p_{Z;\theta}, p_{Z;e}) \text{ s.t. } \theta \in \mathcal{I}^n \cap (\cup_{t\in[0,1]^{\mathcal{Y}}} \Theta_{\bar{\epsilon};t}) , \quad (15)$$

which is in turn equivalent to the following problem that simultaneously optimizes over t:

$$\min_{\theta\in\mathbb{R}^n, t\in[0,1]^{\mathcal{Y}}} \mathcal{W}_c(p_{Z;\theta}, p_{Z;e}) \text{ s.t. } \theta \in \mathcal{I}^n \cap \Theta_{\bar{\epsilon};t} . \quad (16)$$

Compared with problem (4), problem (16) has t as part of the decision variables with $p_Y(y) = t$ and $\epsilon = \bar{\epsilon}$. In other words, if we denote $H_I(t; \bar{\epsilon})$ the optimal objective values for the MIP in (4), then (16) is equal to:

$$\min_{t\in[0,1]^{\mathcal{Y}}} H_I(t; \bar{\epsilon}). \quad (17)$$

Once the optimal $t^\star$ of (17) is obtained, fixing $t = t^\star$ in (16) and optimizing over θ yields the optimal weights $\theta^\star$.

(3) Zero-th Order Optimization Methods for (17)**.** We propose to employ zero-th order optimization methods for the minimization problem in (17). In our setting, this is a particularly efficient choice, as:

- the value of $H_I(t; \bar{\epsilon})$ can be computed via the dual problem (D), as discussed above. Since the cost matrix remains unchanged, after solving (D) for the first time, the complexity of solving the problem again with any different t is only $\tilde{O}(n|\mathcal{Y}|^2|\mathcal{D}|^2 \log(R/\varepsilon))$;
- the problem in (17) is of dimension $|\mathcal{Y}|$, so low-dimensional, with only unit box constraints.

Many methods have shown fast convergence to stationary points for very-low-dimensional problems in practice, such as the multi-dimension golden search method (Chang 2009) and the Nelder-Mead method (Gao and Han 2012). We opt for the latter in our implementation.

Optimality of Integer Weights. Note that once the optimal $t^\star$ of (17) is obtained, the problem (16) with t fixed as $t^\star$ is an instance of problem (4) with $p_Y(y) = t^\star$ and $\epsilon = \bar{\epsilon}$. Hence, according to Theorem 3 and Lemma 4, the optimality of integer weights also carries over to problem (12).

Experiments

In this section, we use a synthetic dataset to provide an efficiency analysis of FairWASP against established state-of-the-art commercial solvers. In addition, we show on various real datasets how FairWASP achieves competitive performance compared to existing methods in reducing disparities while preserving accuracy in downstream classification settings.

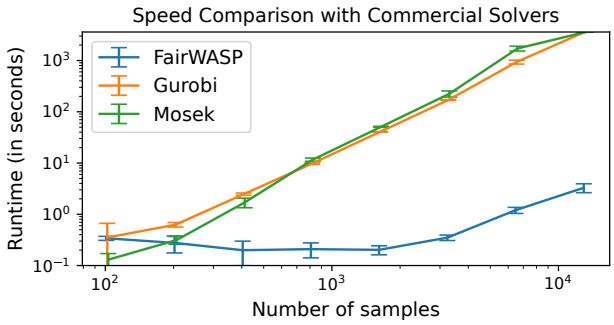

Figure 1: Speed comparison with commercial solvers. FairWASP has significantly better runtime and scalability.

Synthetic dataset We generate a synthetic dataset in which one of the features is highly correlated with the protected variable D, in order to induce a dependency of the outcome on D. We generate a binary protected variable $D = \{0, 1\}$ and features $\mathbf{X} = [X_1, X_2] \in \mathbb{R}^2$, such that X_1 is dependent on the value of D and X_2 is not. More specifically, $X_1 \sim \mathcal{U}[0, 10] \cdot \mathbb{I}(D = 1)$, where $\mathcal{U}$ indicates the uniform distribution and $\mathbb{I}$ the indicator function, so that $X_1 = 0$ if $D = 0$, and $X_2 \sim \mathcal{N}(0, 25)$. The outcome Y is binary and defined as $Y = \mathbb{I}(X_1 + X_2 + \varepsilon > m_X)$, where $m_X = \mathbb{E}(X_1 + X_2)$ and $\varepsilon \sim \mathcal{N}(0, 1)$ is random noise.

Figure 1 compares the runtime of the FairWASP and commercial solvers, Gurobi and Mosek, in solving problem (9) for the synthetic data with different number of samples n (mean and standard deviation over 5 independent trails, with n doubling from $n = 100$ up to $n = 12,800$). The runtime limit for all methods is set to 1 hour, which both commercial solvers exceed when $n > 10,000$. In contrast, FairWASP has a significantly faster runtime than commercial solvers, solving all optimization problems within 5 seconds. As the commercial solvers are run with default settings, we show that the solutions found by FairWASP are comparable to the commercial solver solutions in Supplementary Material C.

Real Datasets We consider the following four real datasets widely used in the fairness literature (Fabris et al. 2022): (i) the Adult dataset (Becker and Kohavi 1996), (ii) the Drug dataset (Fehrman et al. 2017), (iii) the Communities and Crime dataset (Redmond 2009) and (iv) the German Credit dataset (Hofmann 1994). We compare the performance of FairWASP and FairWASP-PW with the following existing pre-processing approaches:

- *DisparateImpactRemover* (DIR, Feldman et al. 2015), which transforms feature values in a rank-preserving fashion,
- *Learning fair representations* (LFR, Zemel et al. 2013), which identifies a latent representation uncorrelated with the protected attributes,
- *Reweighing* (Kamiran and Calders 2012), which weights each sample according to the respective (D, Y) values,
- *Optimized pre-processing* (Calmon et al. 2017), which learns a probabilistic transformation to be applied to the

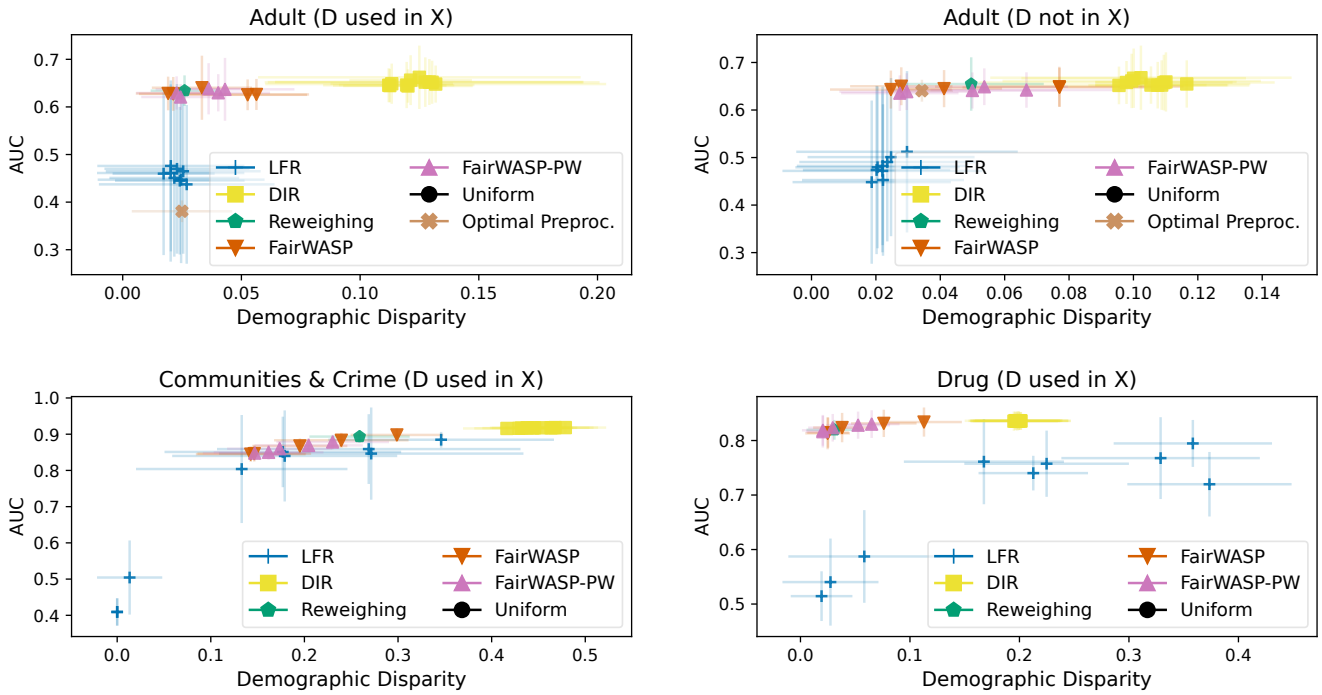

Figure 2: Downstream fairness-utility tradeoff, indicated by the demographic disparity and downstream classifier area under the curve (AUC). The x-axis refers to the absolute difference in the mean classifier outcome for the two groups, with a value of 0 corresponding to perfect demographic parity. Points and error bars correspond to averages plus/minus one standard deviation, computed over 10 different train/test split. FairWASP and FairWASP-PW consistently provide one of the best tradeoffs, significantly improving over using the original dataset as-is. See text and Supplementary Material C for more details.

dataset so that it satisfies group fairness, individual distortion and fidelity constraints.

We also include the *Uniform* approach, which corresponds to the baseline of training on the dataset as-is. In all methods, the pre-processed dataset (or the dataset with no pre-processing, for the *Uniform* approach) is used to train a multi-layer perceptron (MLP) classifier with one hidden layer with 20 nodes and ReLu activation function. Figure 2 shows the fairness-utility tradeoff, indicated by the demographic disparity (defined in the caption) and the classifier AUC, for the Adult dataset (top row), Communities & Crime dataset (bottom left) and Drug dataset (bottom right). We include the settings in which the protected variable D is included among the features X or not; the latter corresponds to the realistic scenario in, e.g., loan credit approvals, in which the US Equal Credit Opportunity Act of 1974[4] prohibits the use of such protected features. In all settings, FairWASP and FairWASP-PW are consistently part of the so-called "Pareto frontier" of the fairness utility tradeoff (Ge et al. 2022), meaning they usually achieve either the best or among the best fairness-utility tradeoffs (closest to the $(0, 1)$ in the top left corner), significantly improving over the empirical distribution (the *Uniform* approach). See Supplementary Material C for more details on datasets, hyper-parameter settings and downstream fairness-accuracy tradeoffs for all datasets.

[4]https://www.law.cornell.edu/uscode/text/15/1691

Conclusions

We propose FairWASP, a novel pre-processing algorithm that returns sample-level weights for a classification dataset without modifying the training data. FairWASP solves an optimization problem that minimizes the Wasserstein distance between the original and the reweighted dataset while satisfying demographic parity constraints. We solve the optimization problem by reformulating it as a mixed-integer program, for which we propose a highly efficient algorithm that we show to be significantly faster than existing commercial solvers. FairWASP returns integer weights, which we show to be optimal, and hence which can be understood as eliminating or duplicating existing samples, making it compatible with any downstream classification algorithm. We empirically show how FairWASP achieves competitive performance with existing pre-processing methods in reducing discrimination while maintaining accuracy in downstream classification tasks.

For future work, we would like to (i) characterize the finite sample properties of FairWASP for the downstream fairness-utility tradeoff, (ii) explore the downstream effect of using different distances in the calculation of the cost matrix C, such as the Wasserstein transform (Memoli, Smith, and Wan 2019), and (iii) extend the proposed optimization framework to non-linear fairness constraints as well as to general LPs and MIPs with similar structures.

Acknowledgments

Some of this work was performed while the first author was at JPMorgan Chase & Co. This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("J.P. Morgan"), and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

Applegate, D.; Díaz, M.; Hinder, O.; Lu, H.; Lubin, M.; O'Donoghue, B.; and Schudy, W. 2021. Practical large-scale linear programming using primal-dual hybrid gradient. *Advances in Neural Information Processing Systems*, 34: 20243–20257.

Applegate, D.; Hinder, O.; Lu, H.; and Lubin, M. 2022. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Mathematical Programming*, 1–52.

Bachem, O.; Lucic, M.; and Krause, A. 2017. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*.

Becker, B.; and Kohavi, R. 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Calders, T.; Kamiran, F.; and Pechenizkiy, M. 2009. Building classifiers with independency constraints. In *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, 13–18. IEEE.

Calmon, F.; Wei, D.; Vinzamuri, B.; Natesan Ramamurthy, K.; and Varshney, K. R. 2017. Optimized pre-processing for discrimination prevention. *Proceedings of the 30th Advances in Neural Information Processing Systems*.

Chai, J.; and Wang, X. 2022. Fairness with adaptive weights. In *Proceedings of the 39th International Conference on Machine Learning*, 2853–2866. PMLR.

Chakraborty, J.; Majumder, S.; and Menzies, T. 2021. Bias in machine learning software: Why? How? What to do? In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 429–440.

Chang, Y.-C. 2009. N-dimension golden section search: Its variants and limitations. In *Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics*, 1–6. IEEE.

Chzhen, E.; Denis, C.; Hebiri, M.; Oneto, L.; and Pontil, M. 2020. Fair regression with wasserstein barycenters. *Proceedings of the 33rd Advances in Neural Information Processing Systems*, 33: 7321–7331.

Chzhen, E.; and Schreuder, N. 2022. A minimax framework for quantifying risk-fairness trade-off in regression. *The Annals of Statistics*, 50(4): 2416–2442.

Claici, S.; Genevay, A.; and Solomon, J. 2018. Wasserstein measure coresets. *arXiv preprint arXiv:1805.07412*.

Du, D.-Z.; and Pardalos, P. M. 1995. *Minimax and applications*, volume 4. Springer Science & Business Media.

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.

Fabris, A.; Messina, S.; Silvello, G.; and Susto, G. A. 2022. Algorithmic fairness datasets: The story so far. *Data Mining and Knowledge Discovery*, 36(6): 2074–2152.

Fehrman, E.; Muhammad, A. K.; Mirkes, E. M.; Egan, V.; and Gorban, A. N. 2017. The five factor model of personality and evaluation of drug consumption risk. In *Data Science: Innovative Developments in Data Analysis and Clustering*, 231–242. Springer.

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.

Gao, F.; and Han, L. 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1): 259–277.

Ge, Y.; Zhao, X.; Yu, L.; Paul, S.; Hu, D.; Hsieh, C.-C.; and Zhang, Y. 2022. Toward Pareto efficient fairness-utility trade-off in recommendation through reinforcement learning. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, 316–324.

Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual. Available on https://www.gurobi.com Accessed: 2023-08-03.

Hofmann, H. 1994. Statlog (German Credit Data). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77.

Hort, M.; Chen, Z.; Zhang, J. M.; Sarro, F.; and Harman, M. 2022. Bias mitigation for machine learning classifiers: A comprehensive survey. *arXiv preprint arXiv:2207.07068*.

Jiang, H.; and Nachum, O. 2020. Identifying and correcting label bias in machine learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 702–712. PMLR.

Kamiran, F.; and Calders, T. 2010. Classification with no discrimination by preferential sampling. In *Proceedings of the 19th Machine Learning Conference of Belgium and The Netherlands*, volume 1. Citeseer.

Kamiran, F.; and Calders, T. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1): 1–33.

Kantorovitch, L. 1958. On the translocation of masses. *Management Science*, 5(1): 1–4.

Khachiyan, L. G. 1980. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1): 53–72.

Li, P.; and Liu, H. 2022. Achieving fairness at no utility cost via data reweighing with influence. In *Proceedings of the 39th International Conference on Machine Learning*, 12917–12930. PMLR.

Memoli, F.; Smith, Z.; and Wan, Z. 2019. The Wasserstein Transform. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 4496–4504. PMLR.

Nesterov, Y. 2018. *Lectures on convex optimization*, volume 137. Springer.

Peyré, G.; Cuturi, M.; et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Redmond, M. 2009. Communities and Crime. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C53W3X.

Salazar, T.; Santos, M. S.; Araújo, H.; and Abreu, P. H. 2021. FAWOS: Fairness-aware oversampling algorithm based on distributions of sensitive attributes. *IEEE Access*, 9: 81370–81379.

Salimi, B.; Rodriguez, L.; Howe, B.; and Suciu, D. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, 793–810.

Santambrogio, F. 2015. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63): 94.

Sloane, M.; Moss, E.; and Chowdhury, R. 2022. A Silicon Valley love triangle: Hiring algorithms, pseudo-science, and the quest for auditability. *Patterns*, 3(2).

Villani, C.; et al. 2009. *Optimal transport: Old and new*, volume 338. Springer.

Wang, H.; Cheng, M.; Basu, K.; Gupta, A.; Selvaraj, K.; and Mazumder, R. 2022. A Light-speed Linear Program Solver for Personalized Recommendation with Diversity Constraints. *arXiv preprint arXiv:2211.12409*.

Xu, D.; Yuan, S.; Zhang, L.; and Wu, X. 2018. FairGAN: Fairness-aware generative adversarial networks. In *Proceedings of the 2018 IEEE International Conference on Big Data*, 570–575. IEEE.

Yan, S.; Kao, H.-t.; and Ferrara, E. 2020. Fair class balancing: Enhancing model fairness without observing sensitive attributes. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1715–1724.

Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning*, 325–333. PMLR.

Zhang, A.; Xing, L.; Zou, J.; and Wu, J. C. 2022. Shifting Machine Learning for Healthcare from Development to Deployment and from Models to Data. *Nature Biomedical Engineering*, 6(12): 1330–1345.

Žliobaite, I.; Kamiran, F.; and Calders, T. 2011. Handling conditional discrimination. In *Proceedings of the 11th IEEE International Conference on Data Mining*, 992–1001. IEEE.