Addressing Fairness in Classification with a Model-Agnostic Multi-Objective Algorithm

Kirtan Padh[1]   **Diego Antognini**[2]   **Emma Lejal-Glaude**[3]   **Boi Faltings**[2]   **Claudiu Musat**[3]

[1]Helmholtz AI, Germany
[2]Ecole Polytechnique Fédérale de Lausanne, Switzerland
[3]Swisscom, Switzerland

Abstract

The goal of fairness in classification is to learn a classifier that does not discriminate against groups of individuals based on *sensitive attributes*, such as race and gender. One approach to designing fair algorithms is to use relaxations of fairness notions as regularization terms or in a constrained optimization problem. We observe that the hyperbolic tangent function can approximate the indicator function. We leverage this property to define a differentiable relaxation that approximates fairness notions provably better than existing relaxations. In addition, we propose a model-agnostic multi-objective architecture that can simultaneously optimize for multiple fairness notions and multiple sensitive attributes and supports all statistical parity-based notions of fairness. We use our relaxation with the multi-objective architecture to learn fair classifiers. Experiments on public datasets show that our method suffers a significantly lower loss of accuracy than current debiasing algorithms relative to the unconstrained model.

1 INTRODUCTION

Machine learning is omnipresent. Machine learning systems have become ubiquitous in our daily lives and society. They are being adopted into an increasing variety of applications at an accelerating pace, including high-impact domains such as healthcare, job hiring, education, and criminal justice, among others (Barocas et al., 2019). Despite this, questions remain on the ethical soundness of many such algorithms, as AI/ML systems have often been demonstrated to have unintentional and undesirable biases against *sensitive attributes* such as age, gender, and race.

Automated predictions can be biased. We consider an algorithm as biased or discriminatory when it does not sat-

isfy a preconceived notion of equality with respect to one or more sensitive attributes. The COMPAS score (Angwin et al., 2016), used in courts in the U.S. to predict the probability of recidivism, is one of the most well-known examples of discrimination by algorithms (Angwin et al., 2016). Among the defendants who do not re-offend, the algorithm predicts black defendants to be higher risk at a much higher rate than white defendants. This can, in turn, lead to a further exacerbation of systemic bias through a negative feedback loop where the results of the algorithm bias the data even further, reflecting the bias even more in the next round of predictions.

The bias can increase over time. A similar bias, which consists of reinforcing existing beliefs, is also present on social media: the filter bubble (Pariser, 2011). The system recommends content that we tend to agree with, further reinforcing our views and putting us in an "echo chamber" with other users with similar views, leading to polarization with users with opposing views. This is believed to have heavily influenced the 2016 U.S. presidential elections (Baer, 2016), and it is the kind of bias that can, over time, change the structure of society. Just as ever-present machine learning algorithms are in society, so is the unintentional algorithmic bias arising from such applications, thus making it critical to study fairness in machine learning.

Debiasing approaches can be divided into three main categories. Firstly, we have *pre-processing algorithms*, where the data is processed before training to rid it of bias with the expectation that the classifier learned on the modified data would be fair (Kamiran and Calders, 2012; Sattigeri et al., 2019; Calmon et al., 2017). Secondly, we have *in-processing algorithms* that propose changes at training time, often in the form of minor changes to existing architectures, or entirely different algorithms (Celis et al., 2019; Lohaus et al., 2020; Zafar et al., 2017b). One approach to in-processing is to define relaxations of fairness notions and solve a constrained optimization problem or use the relaxations as regularization terms. Lastly, there are the *post-processing algorithms* that filter the output of the classifier to ensure fairness (Hardt

et al., 2016; Chierichetti et al., 2017; Chzhen et al., 2019).

Debiasing is a naturally multi-objective problem. Most real-world applications have multiple sensitive attributes. We might want to satisfy different fairness notions for each attribute or several notions for a single attribute, making debiasing a naturally multi-objective problem. However, the research on multi-objective approaches to fairness is very sparse: most methods are specialized towards a specific fairness notion and only apply to a single attribute. Moreover, many fairness relaxations do not approximate the true fairness value well (Lohaus et al., 2020).

In this work, we first define a novel fairness relaxation and show that it approximates the true fairness value better than existing relaxations. Second, we propose a model-agnostic gradient-based multi-objective algorithm that supports multiple sensitive attributes and all notions of fairness that require a form of statistical parity across groups. Experiments on four real-world datasets show that our novel relaxation integrated with the proposed multi-objective algorithm finds fair algorithms while suffering a lower loss of accuracy than state-of-the-art algorithms. Moreover, it performs effectively in simultaneously debiasing for multiple sensitive attributes and measures of fairness with a very low loss of utility.

2 RELATED WORK

We consider the following notions of fairness for our analyses: *demographic parity* (DP) and *equality of opportunity* (EOP). Let the positive prediction be the favorable one in a binary classification problem. For example, for loan default prediction, predicting non-default is favorable. If the sensitive attribute is age with groups 'young' and 'adult,' DP requires the proportion of individuals labeled as positive to be the same for both 'young' and 'adult' groups. In contrast, EOP requires the true positive rate to be the same for both 'young' and 'adult' groups. These definitions are formalized in Section 3.

Relaxation-based Approaches. The approach used by Donini et al. (2018); Zafar et al. (2017a,b) is to write DP or EOP in an equivalent but easier to handle form, and replace the indicator function by a relaxation. Zafar et al. (2017a,b) used a covariance measure between the sensitive attribute and the model parameters as a proxy for the fairness constraint. This leads to a convex constraint for DP (Zafar et al., 2017b) but a non-convex one for EOP (Zafar et al., 2017a). Zafar et al. (2017a) proposed a convex-concave optimization process to deal with the non-convex constraint. For linear models, the covariance constraints reduce to a linear relaxation of the fairness measure. Lohaus et al. (2020) designed an elegant approach where they used an existing convex relaxation of the fairness measures as a regularization term in the loss function, with regularization parameter λ. They proved that the relaxed fairness constraint is a continuous

function of λ, enabling a binary search of λ to find a provably fair classifier. Celis et al. (2019) proposed a method to solve multiple fairness measures simultaneously by reducing a constrained optimization of the loss function to an unconstrained problem by the lagrangian principle.

While these methods are all attractive approaches and work well in practice for a single sensitive attribute, they suffer from two drawbacks: 1. they cannot be integrated into any machine learning model, and 2. require distinct and separate algorithms to solve. Besides, Lohaus et al. (2020) require strong conditions on the classifier, and Zafar et al. (2017a,b) cannot handle multiple fairness measures simultaneously. In comparison, our method handles multiple parity-based measures and is model-agnostic.

Moreover, several existing relaxations inadequately approximate the true fairness value: the relaxations might be satisfied, but the model may still be unfair (Lohaus et al., 2020). Using the evaluation methods proposed in Lohaus et al. (2020) to gauge the effectiveness of different relaxations, we note that our novel relaxation is empirically better.

Multi-Objective Approaches. The line of research that involves multiple objectives in fairness is very recent. Valdivia et al. (2020) proposed an evolutionary approach to optimize for several objectives, using the multi-objective algorithm to search the space of hyperparameters of the model to find one that will work well on multiple objectives. However, it is possible that for some algorithms, there is no set of hyperparameters that perform well for all the objectives. This method is also infeasible to apply to large models since it involves training and evaluating hundreds of hyperparameter tuples. Finally, Celis et al. (2019) proposed an algorithm for a class of statistical-based fairness measures. Their proposal is a meta-algorithm that operates by estimating conditional probabilities. In contrast, our approach can take any existing loss-based model as part of the multi-objective architecture. This would make it much easier for example to reuse production models which have already been optimized and would save the need to implement new architectures from scratch to account for fairness.

3 BACKGROUND

Let $x \in \mathbb{R}^d$ be the features, where d is the total number of features, and $x = (z, a_1, a_2 \ldots a_i \ldots a_t)$. Each a_i refers to a *sensitive attribute*, and z the rest of the attributes. The feature space for z, a_i, and x is denoted by $\mathcal{Z}$, $\mathcal{A}_i$, and $\mathcal{X}$, respectively. Therefore, the domain of x is:

$$\mathcal{X} = \mathcal{Z} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \mathcal{A}_i \times \ldots \mathcal{A}_T \qquad (1)$$

For the sake of simplicity of notation, we assume that we have only one sensitive attribute, that is $T = 1$, and it is denoted simply by a, with feature space $\mathcal{A} = \{-1, 1\}$. Each

individual is assigned an outcome y from the feature space $\mathcal{Y} = \{-1, 1\}$, which is the label we want to predict for x. Assume that there is a distribution $\mathcal{P}_{\mathcal{D}}$ over the domain $\mathcal{D} = \mathcal{X} \times \mathcal{A} \times \mathcal{Y}$. Each (x, a, y) is sampled $i.i.d.$ from $\mathcal{P}_{\mathcal{D}}$. We denote the predictor by $h : \mathcal{X} \to \mathcal{Y}$, where the predicted outcome of x is $h(x) \in \{-1, 1\}$. We define $h(x)$ as $sign(f(x))$, where $f : \mathcal{X} \to \mathbb{R}$ maps each x to a real-valued number, and is fair with respect to the sensitive attributes.

Demographic Parity (DP): A classifier f satisfies demographic parity if the probability of the outcome is independent of the value of the sensitive attribute:

$$\mathbb{P}[f(x) > 0 \,|\, a = -1] = \mathbb{P}[f(x) > 0 \,|\, a = 1] \quad (2)$$

Difference of Demographic Parity (DDP): The first step, in writing Equation 2 as an expression that can be used in a gradient-based optimization, is to relax the definition to be a difference between the expected values of quantities on either side of the equality. This is called the *Difference of Demographic Parity* (DDP), defined as:

$$\text{DDP}(f) = \underset{\mathcal{P}_{\mathcal{D}}}{\mathbb{E}} \left[\mathbb{1}_{f(x)>0} | a = -1 \right] - \underset{\mathcal{P}_{\mathcal{D}}}{\mathbb{E}} \left[\mathbb{1}_{f(x)>0} | a = 1 \right]$$
$$(3)$$

where $\mathbb{1}_c$ is the indicator function on the condition c, which is to say that $\mathbb{1}_c$ is 1 if c is true, and 0 otherwise.

It is clear that when $\text{DDP}(f) = 0$, we achieve perfect demographic parity, although that is usually not a realistic goal. We can relax this requirement by using a threshold: given a threshold $\tau \geq 0$, we say that f is τ-DDP fair if $|\text{DDP}(f)| \leq \tau$. However, this is still not enough to define a differentiable relaxation; we only have an empirical estimate $\widehat{\mathcal{P}}_{\mathcal{D}}$ of $\mathcal{P}_{\mathcal{D}}$ consisting of n points. In that manner, the empirical estimate of DDP can be written as:

$$\widehat{\text{DDP}}(f) = \frac{1}{n_{-1}} \sum_{\substack{\widehat{\mathcal{P}}_{\mathcal{D}} \\ a=-1}} \mathbb{1}_{f(x)>0} - \frac{1}{n_1} \sum_{\substack{\widehat{\mathcal{P}}_{\mathcal{D}} \\ a=1}} \mathbb{1}_{f(x)>0} \quad (4)$$

Here n_{-1} is the number of points with $a = -1$ and n_1 is the number of points with $a = 1$. The total number of points then is $n = n_{-1} + n_1$. This expression is very close to what we can use as a constraint. However, the main problem with using this expression directly in a gradient-based optimization is the non-differentiability because of the indicator function. The differences between different relaxations then come from how the indicator function is relaxed in the expression above.

Equality of Opportunity (EOP): A classifier f satisfies equality of opportunity if the probability of getting a true positive is independent of the value of the sensitive attribute:

$$\mathbb{P}[f(x) > 0 \,|\, a = -1, y = 1] = \mathbb{P}[f(x) > 0 \,|\, a = 1, y = 1]$$
$$(5)$$

Difference of Equality of Opportunity (DEO): We relax Equation 5, similarly than for the demographic parity in Equation 4, to get the *Difference of Equality of opportunity* (DEO). Then the empirical version of DEO is expressed as follows:

$$\widehat{\text{DEO}}(f) = \frac{1}{n_{-1}} \sum_{\substack{\widehat{\mathcal{P}}_{\mathcal{D}} \\ a=-1 \\ y=1}} \mathbb{1}_{f(x)>0} - \frac{1}{n_1} \sum_{\substack{\widehat{\mathcal{P}}_{\mathcal{D}} \\ a=1 \\ y=1}} \mathbb{1}_{f(x)>0} \quad (6)$$

3.1 FAIRNESS RELAXATIONS

The differences between relaxations come from how the indicator function is relaxed in the expressions $\widehat{\text{DDP}}(f)$ and $\widehat{\text{DEO}}(f)$. We conduct all analyses for demographic parity; the extension to EOP is straightforward by conditioning on the positive label.

Linear Relaxations: Donini et al. (2018); Zafar et al. (2017b) proposed a linear relaxation, where the indicator function is simply replaced by a linear classifier $f(x)$. $\widehat{\text{DDP}}(f)$ can then be written in the following equivalent form after substituting $\mathbb{1}_{f(x)>0}$ by $f(x)$ (Lohaus et al., 2020):

$$\left| \text{LR}_{\widehat{\text{DDP}}}(f) \right| = \left| \frac{1}{n} \sum_{\widehat{\mathcal{P}}_{\mathcal{D}}} C\left(a, \widehat{\mathcal{P}}_{\mathcal{D}}\right) f(x) \right| \leq \tau \quad (7)$$

where $C(a, \widehat{\mathcal{P}}_{\mathcal{D}})$ is simply a constant multiplicative factor.

Convex-Concave Relaxations: Zafar et al. (2017a) proposed the convex-concave relaxation, where $\mathbb{1}_{f(x)>0}$ is relaxed to $\min(0, f(x))$. Let $\hat{p}_1$ be the empirical estimate of the proportion of individuals with $a = 1$. For the case of such a relaxation for DP, $\widehat{\text{DDP}}(f)$ can be written in the following equivalent form after substituting $\mathbb{1}_{f(x)>0}$ by $\min(0, f(x))$ (Lohaus et al., 2020):

$$\left| \text{CCR}_{\widehat{\text{DDP}}}(f) \right| = \left| \frac{1}{n} \sum_{\widehat{\mathcal{P}}_{\mathcal{D}}} C'\left(a, \widehat{\mathcal{P}}_{\mathcal{D}}\right) \min(0, f(x)) \right| \leq \tau$$
$$(8)$$

4 A NOVEL FAIRNESS RELAXATION

The existing relaxations described do not approximate the true DDP value accurately. To illustrate this, we use a two-dimensional toy dataset for binary classification, similarly to Lohaus et al. (2020). Various Gaussian distributions are used to generate the points for each label. Each point is also assigned one of two groups to simulate the sensitive attribute. As we can see from Figure 1, existing relaxations do not faithfully capture the true DDP value.

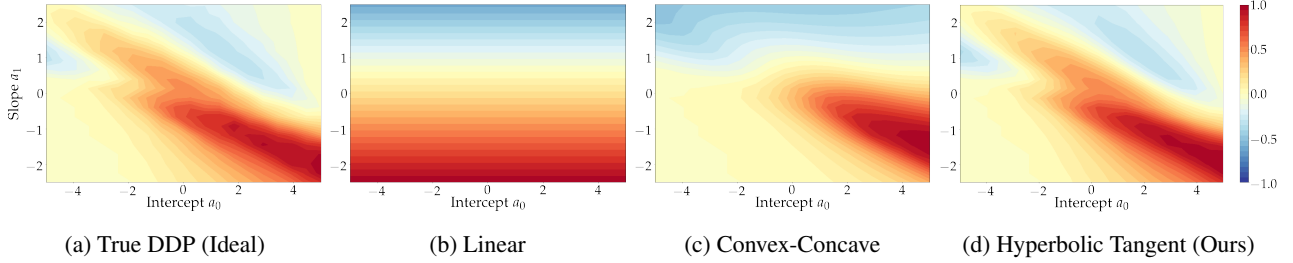| (a) True DDP (Ideal) | (b) Linear | (c) Convex-Concave | (d) Hyperbolic Tangent (Ours) |

Figure 1: Each plot describes the family of linear classifiers in two dimensions which can be used to separate the classes in a two dimensional synthetic dataset. The decision boundary is defined as $x_2 = a_1 x_1 + a_0$, meaning that $f(x) = -x_2 + a_1 x_1 + a_0$. The point at (a_0, a_1) on each plot gives normalized value of each of the following quantities for a classifier $f(x)$ with parameters (a_0, a_1): (1a) True Difference of demographic parity (DDP), (1b) Linear relaxation of the DDP, (1c) Convex-concave relaxation of the DDP, and (1d) Hyperbolic tangent relaxation (HTR). Yellow is fair. Ideally, we want the plot of the relaxations to be like that of the true DDP (1a).

To solve this problem, we introduce a new relaxation, called the hyperbolic tangent relaxation (HTR). Let $\mathrm{sign}(x)$ denote the signum of x, i.e. $sign(x)$ is 1 if $x > 0$, -1 if $x < 0$ and 0 if $x = 0$. Figure 1 further illustrates that our relaxation is the best at capturing the true DDP.

Theorem 1. *The hyperbolic tangent of $n * x$ converges to the sign of x for every fixed $x \in \mathbb{R}$ as n goes to infinity. Formally,*

$$\lim_{n \to \infty} \tanh(n * x) = \mathrm{sign}(x) \,\forall x \in \mathbb{R} \qquad (9)$$

Proof. The idea is that replacing x by $n * x$ in $\tanh(x)$ compresses the horizontal scale. A more detailed proof is provided in the appendix . $\square$

We can leverage Theorem 1 to find an expression that converges to the indicator function of $x > 0$.

Lemma 1. $\tanh(n * \max(0, x))$ *converges to the indicator function of $x > 0$ as n goes to infinity. Formally,*

$$\lim_{n \to \infty} \tanh(n * \max(0, x)) = \mathbb{1}_{x>0} \,\forall x \in \mathbb{R} \qquad (10)$$

Proof. The proof is by simply replacing x in Theorem 1 by $\max(0, x)$. The details are worked out in the appendix. $\square$

Hyperbolic Tangent Relaxation (HTR): Instead of relaxing $\mathbb{1}_{f(x)>0}$ by $f(x)$ or $min(0, f(x))$ as proposed in the linear and convex-concave relaxations, respectively, we propose $\tanh(c * \max(0, f(x)))$, for small constants c. The larger the value of c, the better we can approximate the indicator function, but at the cost of degradation in the gradient's behavior.

We denote $\tanh(c * \max(0, x))$ as $t(c, x)$. Formally, the hyperbolic tangent relaxation for the DDP, denoted by HTR

can be written as follows, for a chosen constant c:

$$\mathrm{HTR}_{\widehat{DDP}}(f) = \frac{1}{n_{-1}} \sum_{\substack{\widehat{\mathcal{P}}_{\mathcal{D}} \\ a=-1}} t(c, f(x)) - \frac{1}{n_1} \sum_{\substack{\widehat{\mathcal{P}}_{\mathcal{D}} \\ a=1}} t(c, f(x))$$

$$(11)$$

Finally, Figure 1 demonstrates how the HTR is a better approximation of DDP than existing relaxations.

5 THE MAMO-FAIR ALGORITHM

As our multi-objective optimization method, we use the algorithm of Poirion et al. (2017) with modifications suggested by Milojkovic et al. (2019). We assume without loss of generality that all objectives are to be minimized. A multi-objective optimization problem can then be formulated as follows:

$$\min_{w \in \mathbb{R}^d} L(w) = \min_{w \in \mathbb{R}^d} (\ell_1(w), \ell_2(w), \dots \ell_k(w)) \qquad (12)$$

where $\ell_i : \mathbb{R}^d \to \mathbb{R} \quad \forall i = 1, ..., k$ are the k objectives, with $k \geq 2$. We interpret $L(w)$ as a multi-objective loss function and each $\ell_i(w)$ as one of the loss functions to be optimized by a machine learning model, with w being the model parameters. Unlike in single-objective optimization problems, solutions of a multi-objective optimization problem are not ordered linearly. They are instead compared by dominance of solutions.

Definition 1 (**Dominance of a Solution**). A solution w_1 of Equation 12 dominates another solution $w_2 \neq w_1$ if $\ell_i(w_1) \leq \ell_i(w_2) \,\forall\, i = 1, ..., k$ and there exists $i_0 \in [1, k]$ such that $\ell_i(w_1) < \ell_i(w_2)$.

Definition 2 (**Pareto Optimality**). A solution w^* of Equation 12 is *pareto optimal* if no other solution w dominates it.

Definition 3 (**Pareto Front**). The *pareto front* of a set of solutions of Equation 12 is the set of all non-dominated solutions.

We denote the gradient of objective $\ell_i(w)$ by $\nabla_w \ell_i(w)$. The key idea of the algorithm in optimizing simultaneously several objectives is to find a single vector, that gives the descent direction for every objective. This is called the common descent vector (CDV). The Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951) provide necessary optimality conditions for the solution of a deterministic gradient-based optimization. A solution which satisfies the KKT conditions for a multi-objective optimization problem is called a pareto stationary point.

Definition 4 (Pareto Stationary). A solution w is pareto stationary if:

$$\exists (\alpha_1, , \alpha_2, \ldots, \alpha_k) \left| \sum_{i=1}^{k} \alpha_i = 1, \sum_{i=1}^{k} \alpha_i \nabla_w \ell_i(w) = 0 \right. \quad (13)$$

Note that pareto stationarity is a necessary but not sufficient condition for optimality. The pareto stationary point admits a solution in the convex hull of the set $\{\nabla_w l_i(w) \mid i \in [k]\}$ (Désidéri, 2012), which is the same as saying that the zero vector needs to be in the convex hull. The key idea is that the pareto stationary point can be found by iteratively solving the following optimization problem.

Definition 5 (Quadratic Constrained Optimization Problem (QCOP)). The QCOP for our purpose is defined as follows:

$$\min_{\alpha_1, \ldots, \alpha_n} \left\{ \left\| \sum_{i=1}^{n} \alpha_i \nabla_w \ell_i(w) \right\|^2 \left| \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0 \right. \right\} \quad (14)$$

Let p^* be the vector of a solution of the Equation 14, meaning that it is a convex combination of gradients specified by alpha. Then we have either:

1. $\|p^*\| = 0$, implies that the solution w is pareto stationary;

2. $\|p^*\| > 0$, the solution w is not pareto stationary and $\nabla_w \mathbf{L}(\mathbf{w}) = p^*$, where $\nabla_w \mathbf{L}(\mathbf{w})$ denotes the common descent vector.

The only key ingredient missing to describe the algorithm is the gradient normalization, proposed by Milojkovic et al. (2019). This allows us to overcome the issue of having losses with different scales.

Definition 6 (Gradient normalization). Let $l_i(w), \ldots, l_k(w)$ be the k objectives and $\nabla_w(l_i(w))$ the gradient of $l_i(w)$ for all $i = 1, \ldots, k$. We define w_{init} as the initial weight of the model. Then, we normalize the gradient as follows:

$$\nabla_w \overline{l_i(w)} = \frac{\nabla_w l_i(w)}{l_i(w_{init})} \quad (15)$$

We now have all the components to describe the final algorithm. The general idea is:

1. Calculate and normalize each gradient;

2. Find the common descent direction through QCOP;

3. Update gradients by performing the descent step;

4. Repeat for an appropriate number of batches and epochs.

The pseudocode is provided in the appendix. The procedure is model-agnostic, so long as the model supports gradient-based optimization. In particular, unlike other methods which require convexity or are based on specific optimization algorithms, this method works well with neural networks as well. This is note-worthy because increasingly many real-world applications use complex non-convex models.

The key to using the algorithm is implementing fairness notions as loss functions, which is where our hyperbolic tangent relaxation comes into play.

6 EXPERIMENTS

In this section, we assess the performance of our method based on experiments on four publicly available datasets.

6.1 DATASETS

We use the following datasets:

- **Adult** (Dua and Graff, 2017): the task is to predict if income is above or below 50k\$. Among the 14 features are attributes gender and race. We use *sex* and a binarized version of *race* as sensitive attributes. $y = 1$ corresponds to the favorable prediction (income$\geq$50k\$). There are a total of 48,842 instances;

- **Compas** (Angwin et al., 2016): the task is to predict if a defendant will racedeviate. There are 53 attribute, among them *race* and *sex*, which we use as sensitive attributes. There are 6,167 samples in total;

- **Dutch census** (Žliobaite et al., 2011): Census data of the Netherlands in 2001. Occupation is used as a proxy for low and high income, and *sex* is used as a sensitive attribute. The data contains 60,420 instances with 12 features;

- **Celeb attributes** (Liu et al., 2015): it is a dataset containing 202,599 face images of celebrities. This is accompanied by a list of 40 binary attributes for each image. We use this attribute dataset for classification, with the attribute *smiling* used as a label, and *sex* as a sensitive attribute.

For the Compas dataset we use 3,000 samples for training, 2,000 for validation and the rest for testing. For the others we use 10,000 samples for training, 5,000 for validation, and the rest for testing.

6.2 BASELINES

Two Objectives. We consider three baselines: a constrained optimization method with the linear relaxation of Zafar et al. (2017b); the recent method of Cotter et al. (2019) for solving the lagrangian, and the searchFair algorithm of Lohaus et al. (2020). We directly report the results of our baselines from Lohaus et al. (2020). As the authors provide all experimental details necessary, we ensured to use precisely the same setting to be able to compare the relaxation-based approaches. In particular, we use the same sizes for training and test sets and the number of runs, as well as the same sets of features and pre-processing.

Beyond Two Objectives. For more than two objectives, we cannot compare against traditional debiasing algorithms. In this case, we employ the following baselines:

- **Sum of losses:** Multiple models with a single objective optimization. We represent the final objective as the sum of all objectives;

- **Unconstrained model:** A model without any constraint regarding fairness.

6.3 OBJECTIVES

We recall that our algorithm solves the optimization problem described in Equation 12. When optimizing for a single sensitive attribute for a single measure of fairness, we have two objectives: ℓ_1 and ℓ_2. ℓ_1 is the **performance objective**, for which we use the binary cross-entropy (BCE), and ℓ_2 is the **fairness objective**. ℓ_2 corresponds to the hyperbolic tangent relaxation of the fairness notion along with BCE added as a regularizer. For the DDP, the fairness objective is:

$$\ell_2 = \mathrm{HTR}_{\widehat{DDP}}(f) + \lambda * BCE(f) \qquad (16)$$

where λ is the binary cross-entropy regularizer. The regularizer is needed to avoid trivial constant solutions that attain perfect fairness, hence taking the fairness loss to zero.

The choice of the constant c in the HTR relaxation. Recall that the HTR relaxation defined in Equation 11 requires the specification of a constant c which allows us to decide how closely we want to approximate the true fairness. There is a trade-off between the behaviour of the gradient and the value of c. A higher value of c gives a better approximation of the fairness value but a worse-behaving gradient. We choose $c = 3$ based on the empirical results. Exploring the

impact of this constant on the optimization process for various relaxations could be an interesting direction of future work.

For each additional sensitive attribute or fairness notion we want to optimize for, we add an analogous fairness objective. In other terms, the hyperbolic tangent relaxation of the fairness notion in question, with the BCE as a regularization term.

6.4 METRICS

Single Fairness. The goal is to learn classifiers that give the best improvement in fairness for the least decrease in accuracy, compared to the unconstrained model. We report the fairness difference metric (DDP or DEO) and the accuracy. We emphasize that DDP and DEO are representative choices, and the algorithm supports an array statistical parity based metrics. See Appendix C for details.

Multi-Fairness. When having more than one sensitive attribute and/or fairness notion, a single point solution is not representative of the overall performance. Therefore, we compare the pareto fronts instead, that we denote by S. The pareto front consists of a set of points in $\mathbb{R}^k$, where k is the number of objectives.

Constructing the pareto front. The pareto front is constructed through a single training run of the algorithm. After each epoch, the trained model is added to the pareto front if it is pareto-optimal with respect to every point in the existing pareto front. The same method is used for the multi-objective algorithm. While doing several runs for our algorithm to construct the pareto front would make our results look stronger, we have avoided this to not give our approach an undue advantage over methods that do not have trade-off parameters.

As metrics, we employ the hypervolume and the spread of the pareto front:

- **Hypervolume** (Zitzler et al., 2007): the dominance volume enclosed by the pareto set in $\mathbb{R}^k$ with respect to the reference point. The larger the hypervolume, the better the solution. For our purpose, the reference point is always the origin;

- **Spacing** (Okabe et al., 2003): the spacing of S is a measure of how spread out the pareto front is. Spacing is low when the solutions are all in a single cluster, and high when they form a spread out pareto front. Formally, the spacing is defined as:

$$SP(S) = \sqrt{\frac{1}{|S-1|} \sum_{i=1}^{|S|} \left(d_i - \bar{d}\right)^2} \qquad (17)$$

where d_i is the shortest l_1-norm from s_i to any other

point in S:

$$d_i = \min_{s_r \in S, s_r \neq s_i} \sum_{m=1}^{k} |l_m(s_i) - l_m(s_r)| \qquad (18)$$

6.5 SOLUTION SELECTION

Selecting the best solution from the pareto front of a single run is nontrivial. Wang and Rangaiah (2017) list several strategies of selecting a point from the pareto front. Here we use the Linear Programming Technique for Multidimensional Analysis of Preference (LINMAP) method proposed by Srinivasan and Shocker (1973). LINMAP selects the point in the pareto front closest to ideal point. We choose this strategy as we can expect it to not favour a particular objective and give a model that finds a good trade-off between different objectives.

We use a training, validation, and test set for each run of the multi-objective algorithm. For each run, the model trained on the training set is evaluated on the validation set first, and the LINMAP strategy is used on the results of validation set to select the final point. The model corresponding to this point is the chosen model for each run and used for evaluation of the test samples. In this manner, we ensure that we are not fitting to the test samples for the results.

6.6 OPTIMIZATION FRAMEWORK

We implemented the MAMO-fair algorithm as a publicly-available modular framework which implements most statistical parity based group fairness metrics. All implementation is in pytorch. The full list of implemented objectives is provided in the appendix. The framework is easy to extend by implementing other fairness notions and datasets, with instructions and documentation on how to do so provided with the implementation. This is in addition to the pre-processing and optimization code already available within the framework for the four datasets used in our experiments.

6.7 OUR MODELS

We compare the baselines against two variants of our MAMO-fair model:

- S-MAMO-fair (Single fairness): the algorithm optimizes for only one notion of fairness at a time;
- M-MAMO-fair (Multi-fairness): the multi-fair MAMO-fair algorithm, where we have a single algorithm optimized simultaneously for DDP and DEO.[1]

---

[1]Code is available at https://github.com/kirtanp/MAMO-fair/

One of the strengths of our approach is that it is model-agnostic, so it also works with neural networks unlike other debiasing algorithms (Zafar et al., 2017a,b; Celis et al., 2019; Lohaus et al., 2020). We demonstrate it by using a feedforward neural network with 2 hidden layers of sizes 60 and 25 respectively, a ReLu activation function (Xu et al., 2015), dropout (Srivastava et al., 2014) with $p = 0.2$ between each layer, and a sigmoid at the output layer.

6.8 HYPERPARAMETER SELECTION

The most important hyperparameter choice is that of λ in the fairness objectives (Equation 16). We found that a value of $\lambda = 0.1$ works well for all datasets and both metrics. We use a batch size of 512 for Adult and Compas, and 200 for Dutch and CelebA. We use a learning rate of 0.01 for all experiments. We did not need to perform automated hyperparameter tuning of our method to achieve results comparable to the baselines.

7 RESULTS

We present the results of our experiments for single and multiple fairness objectives.

7.1 SINGLE FAIRNESS

Figure 2 shows the results for the case of single-fairness. We see that our algorithm significantly improves on fairness with a very low loss of accuracy on both fairness metrics. While traditional models are optimized for a single fairness notion, we show that when trained on both fairness notions DDP and DEO simultaneously, our model (M-MAMO-fair) achieves higher performance on two out of four cases.

Least Loss of Accuracy. First, we see that our algorithm (S-MAMO-fair) always has the least loss of accuracy among all the methods. Second, we observe that whenever another algorithm matches the accuracy achieved by S-MAMO-fair, our model achieves a better performance on fairness. The only exception to this among the eight experiments is in Figure 2a (DEO), where Zafar performs marginally better than the S-MAMO-fair algorithm with the same loss of accuracy. In Figure 2c Zafar has a slightly better accuracy than our methods, but with a much worse fairness value.

Good Trade-off between Error and Fairness. Methods that have a better performance than S-MAMO-fair on fairness often lose out significantly in the accuracy and end up being close to the trivial constant model. This is most clearly seen in results for the Adult and Compas datasets. For the Dutch and CelebA datasets, all methods perform well on fairness, but S-MAMO-fair still achieves the best
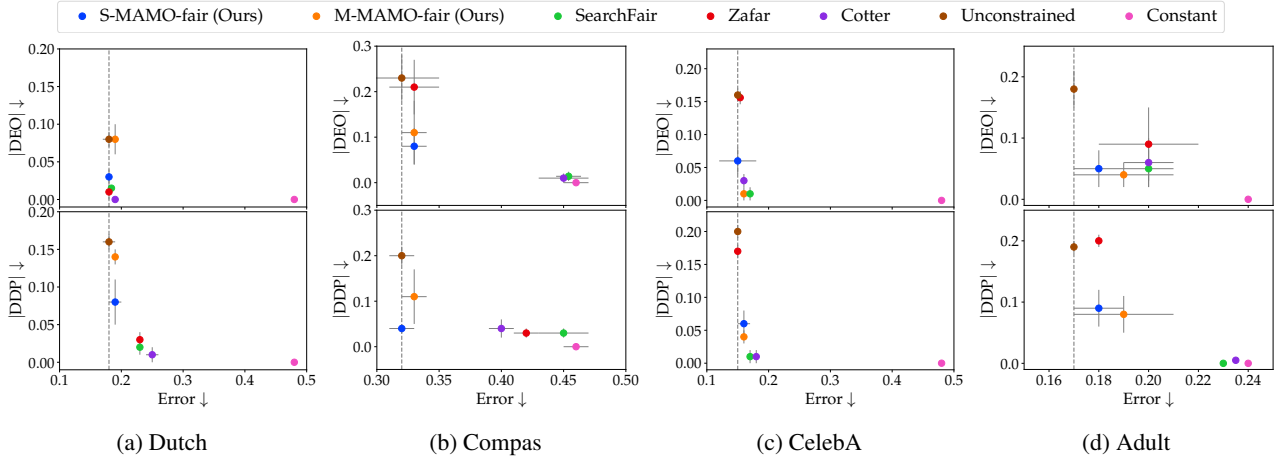
Figure 2: Single fairness results. The first and second rows contain the results for the DEO and DPP fairness metric, respectively. For both axes, a lower value is better. So the closer a point to the origin, the better the method. We emphasize that for each dataset, **M-MAMO-fair is optimized simultaneously for both DDP and DEO**. The dotted line shows the error of the unconstrained model. The closer a point is to the dotted line, the lower is the loss of accuracy suffered by the corresponding method. We see that our method achieves the best error among all methods while significantly improving fairness compared to the unconstrained model. The full tables are available in the appendix.

Table 1: Compas: simultaneously for race and gender.

	M-MAMO-fair	Sum of Losses	Unconstrained
HV	**0.61 ± 0.01**	0.55 ± 0.06	0.34 ± 0.03
SP	**0.21 ± 0.09**	0.05 ± 0.04	0.02 ± 0.01

Table 2: Adult: simultaneously for race and gender.

	M-MAMO-fair	Sum of Losses	Unconstrained
HV	**0.60 ± 0.09**	0.30 ± 0.02	0.31 ± 0.05
SP	**0.04 ± 0.02**	0.02 ± 0.01	0.02 ± 0.01

accuracy, suggesting that these datasets are easier to debias than Adult and Compas.

Multi-Fairness Works Well. Interestingly, we note that our multi-fairness algorithm outperforms single-fairness baselines in half of the cases. In particular, for the Adult and CelebA datasets, the M-MAMO-fair algorithm performs very close to the S-MAMO-fair algorithm and gives a better accuracy and better fairness than the baseline methods.

Inherent Limitations of Multi-Fairness. For the Compas dataset, M-MAMO-fair performs well for DEO but not for DDP, which is in line with the *impossibility results for fairness*: it is not possible to satisfy DP and error rate based metrics simultaneously if the base rate of classification is different for different groups (Corbett-Davies et al., 2017; Goel et al., 2018). This explains the poor performance of the M-MAMO-fair algorithm on the Dutch dataset as well as the fact that it performs well only on DEO and not on DDP for the Compas dataset. However, this makes multi-objective algorithms for fairness even more essential, so as to find the best possible trade-offs between different fairness metrics, which our algorithm is shown to do well. The parameter λ in the fairness objective (Equation 16) can be used to control the trade-off.

7.2 MULTI-FAIRNESS

Here we further illustrate the power of the algorithm to debias simultaneously for multiple sensitive attributes. Two of the datasets, Compas and Adult, contain both race and gender as sensitive attributes. For each dataset, we debias with respect to demographic parity simultaneously for race and gender. The metrics and baselines are as defined in Section 6.4 and Section 6.2 respectively. Table 1 and Table 2 show that our method outperforms the baselines on both metrics.

8 CONCLUSION

In this paper, we addressed the important problem of social discrimination in machine learning classifiers. We considered a specific class of debiasing algorithms which looks at relaxations of fairness notions. We have empirically shown that existing relaxations do not approximate the true fairness value well enough.

Motivated by this, we proposed new relaxations which provably approximate fairness notions better than existing ones. In addition, we observed that debiasing is a naturally multi-objective problem, but there is a dearth of research in the

field of multi-objective debiasing algorithms. We have taken a first step towards alleviating this scarcity by proposing a model-agnostic multi-objective method for finding fair and accurate classifiers. We demonstrated through experiments on four real-world publicly available datasets that our algorithm performs better than current state-of-the-art models at finding trade-offs between accuracy and fairness. Moreover, it can be used to simultaneously debias for multiple definitions of fairness and multiple sensitive attributes.

Acknowledgements

References

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23:2016, 2016.

Drake Baer. The 'filter bubble' explains why trump won and you didn't see it coming. *New York Magazine. Retrieved from http://nymag.com/scienceofus/2016/11/how-facebook-and-the-filter-bubble-pushed-trump-to-victory.html*, 2016.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org.

Flavio P Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3995–4004, 2017.

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 319–328, 2019.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Leveraging labeled and unlabeled data for consistent fair binary classification. *arXiv preprint arXiv:1906.05082*, 2019.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 797–806, 2017.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pages 300–332. PMLR, 2019.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, pages 2791–2801, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 116–116, 2018.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*, 2016.

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

William Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press. URL https://projecteuclid.org/euclid.bsmsp/1200500249.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Michael Lohaus, Michaël Perrot, and Ulrike von Luxburg. Too relaxed to be fair. In *International Conference on Machine Learning*, 2020.

Nikola Milojkovic, Diego Antognini, Giancarlo Bergamin, Boi Faltings, and Claudiu Musat. Multi-gradient descent for multi-objective recommender systems. *arXiv preprint arXiv:2001.00846*, 2019.

Tatsuya Okabe, Yaochu Jin, and Bernhard Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *The 2003 Congress on Evolutionary*

Computation, 2003. CEC'03., volume 2, pages 878–885. IEEE, 2003.

Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

Fabrice Poirion, Quentin Mercier, and Jean-Antoine Désidéri. Descent algorithm for nonsmooth stochastic multiobjective optimization. *Computational Optimization and Applications*, 68(2):317–331, 2017.

Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3–1, 2019.

Venkataraman Srinivasan and Allan D Shocker. Linear programming techniques for multidimensional analysis of preferences. *Psychometrika*, 38(3):337–369, 1973.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Ana Valdivia, Javier Sánchez-Monedero, and Jorge Casillas. How fair can we go in machine learning? assessing the boundaries of fairness in decision trees. *arXiv preprint arXiv:2006.12399*, 2020.

Zhiyuan Wang and Gade Pandu Rangaiah. Application and analysis of methods for selecting an optimal solution from the pareto-optimal front obtained by multiobjective optimization. *Industrial & Engineering Chemistry Research*, 56(2):560–574, 2017.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017a.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017b.

Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer, 2007.

Indre Žliobaite, Faisal Kamiran, and Toon Calders. Handling conditional discrimination. In *2011 IEEE 11th International Conference on Data Mining*, pages 992–1001. IEEE, 2011.