Identifying and Correcting Label Bias in Machine Learning

Heinrich Jiang
Google Research

Ofir Nachum
Google Research

Abstract

Datasets often contain biases which unfairly disadvantage certain groups, and classifiers trained on such datasets can inherit these biases. In this paper, we provide a mathematical formulation of how this bias can arise. We do so by assuming the existence of underlying, unknown, and unbiased labels which are overwritten by an agent who intends to provide accurate labels but may have biases against certain groups. Despite the fact that we only observe the biased labels, we are able to show that the bias may nevertheless be corrected by re-weighting the data points without changing the labels. We show, with theoretical guarantees, that training on the re-weighted dataset corresponds to training on the *unobserved* but *unbiased* labels, thus leading to an unbiased machine learning classifier. Our procedure is fast and robust and can be used with virtually any learning algorithm. We evaluate on a number of standard machine learning fairness datasets and a variety of fairness notions, finding that our method outperforms standard approaches in achieving fair classification.[1]

1 INTRODUCTION

Machine learning has become widely adopted in a variety of real-world applications that significantly affect people's lives (Guimaraes and Tofighi, 2018; Guegan and Hassani, 2018). Fairness in these algorithmic decision-making systems has thus become an increasingly important concern: It has been shown that without appropriate intervention during training or evaluation, models can be biased against certain groups (Angwin et al., 2016; Hardt et al., 2016). This is due to the fact that the data used to train these models often contains biases that become reinforced into the model (Bolukbasi et al., 2016). Moreover, it has been shown that simple remedies, such as ignoring the features corresponding to the protected groups, are largely ineffective due to redundant encodings in the data (Pedreshi et al., 2008). In other words, the data can be inherently biased in possibly complex ways, thus making it difficult to achieve fairness.

Research on training fair classifiers has therefore received a great deal of attention. One such approach has focused on developing *post-processing* steps to enforce fairness on a learned model (Doherty et al., 2012; Feldman, 2015; Hardt et al., 2016). That is, one first trains a machine learning model, resulting in an unfair classifier. The outputs of the classifier are then calibrated to enforce fairness. Although this approach is likely to decrease the bias of the classifier, by decoupling the training from the fairness enforcement, this procedure may not lead to the best trade-off between fairness and accuracy (Woodworth et al., 2017). Accordingly, recent work has proposed to incorporate fairness into the training algorithm itself, framing the problem as a constrained optimization problem and subsequently applying the method of *Lagrange multipliers* to transform the constraints to penalties (Zafar et al., 2015; Goh et al., 2016; Cotter et al., 2018b; Agarwal et al., 2018); however such approaches may introduce undesired complexity and lead to more difficult or unstable training (Cotter et al., 2018b,c). Both of these existing methods address the problem of bias by adjusting the machine learning model. However, oftentimes it is the training data itself – i.e., the observed features and corresponding labels – which are biased, and thus it is potentially more fruitful to tackle this problem directly.

In this paper, we provide an approach to machine learning fairness that addresses the underlying data bias problem directly. We introduce a new mathematical framework for fairness in which we assume that there exists an *unknown* but *unbiased* ground truth

[1] See experimental code at `https://github.com/google-research/google-research/tree/master/label_bias`.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

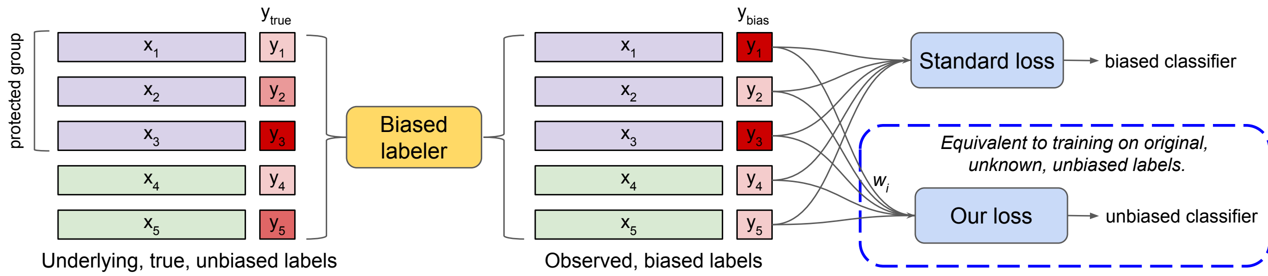

Figure 1: In our approach to training an unbiased, fair classifier, we assume the existence of a true but unknown label function which has been adjusted by a biased process to produce the labels observed in the training data. Our main contribution is providing a procedure that appropriately weights examples in the dataset, and then showing that training on the resulting loss corresponds to training on the original, true, unbiased labels.

label function and that the labels observed in the data are assigned by an agent who is possibly biased, but otherwise has the intention of being accurate. This assumption is natural in practice, where the observed data may be the result of manual labelling done by actors (e.g. human decision-makers) who strive to provide an accurate label while being affected by (potentially unconscious) biases; or in cases where the observed labels correspond to a process (e.g. results of a written exam) devised to be accurate and fair, but which is nevertheless affected by inherent biases.

Based on this mathematical formulation, we show how one may identify the bias of the observed label function as a closed form expression. Furthermore, our derived form for the bias suggests that its correction may be performed by assigning appropriate weights to each example in the training data. We show, with theoretical guarantees, that training the classifier under the resulting weighted objective leads to an unbiased classifier on the original un-weighted dataset. Notably, many pre-processing approaches and even constrained optimization approaches (e.g. (Agarwal et al., 2018)) optimize a loss which possibly modifies the observed labels or features, and doing so may be interpreted as training on *falsified* data. In contrast, our method *does not* modify any of the observed labels or features. Rather, we correct for the bias by changing the distribution of the sample points via re-weighting the dataset.

Our resulting method is general and can be applied to various notions of fairness, including demographic parity, equal opportunity, equalized odds, and disparate impact. Moreover, the method is practical and simple to tune: With the appropriate example weights, any off-the-shelf classification procedure can be used on the weighted dataset to learn a fair classifier. Experimentally, we show that on standard fairness benchmark datasets and under a variety of fairness notions our method can outperform previous approaches to fair classification.

2 BACKGROUND

In this section, we introduce our framework for machine learning fairness, which explicitly assumes an unknown and unbiased ground truth label function. We additionally introduce notation and definitions used in the subsequent presentation of our method.

2.1 Biased and Unbiased Labels

Consider a data domain $\mathcal{X}$ and an associated data distribution $\mathcal{P}$. An element $x \in \mathcal{X}$ may be interpreted as a feature vector associated with a specific example. We let $\mathcal{Y} := \{0, 1\}$ be the labels, considering the binary classification setting, although our method may be readily generalized.

We assume the existence of an *unbiased*, ground truth label function $y_{\text{true}} : \mathcal{X} \to [0, 1]$. Although y_{true} is the assumed ground truth, in general we do not have access to it. Rather, our dataset has labels generated based on a *biased* label function $y_{\text{bias}} : \mathcal{X} \to [0, 1]$. Accordingly, we assume that our data is drawn according to $(x, y) \sim \mathcal{D} \equiv x \sim \mathcal{P}, y \sim \text{Bernoulli}(y_{\text{bias}}(x))$, and we assume access to a finite sample $\mathcal{D}_{[n]} := \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ drawn from $\mathcal{D}$.

In a machine learning context, our directive is to use the dataset $\mathcal{D}$ to recover the unbiased, true label function y_{true}. In general, the relationship between the desired y_{true} and the observed y_{bias} is unknown. Without additional assumptions and with access only to labels from y_{bias}, it is impossible to learn a machine learning model to fit y_{true}. We will attack this problem in the following sections by proposing a minimal and natural assumption on the relationship between y_{true} and y_{bias}. The assumption will allow us to derive an expression for y_{true} in terms of y_{bias}, and the form of this expression will immediately imply that correction of the label bias may be done by appropriately re-weighting the data.

2.2 Notions of Bias

We now discuss precise ways in which y_{bias} can be biased. We describe a number of proposed notions of fairness; i.e., what it means for an arbitrary label function or machine learning model $h : \mathcal{X} \to [0, 1]$ to be biased (unfair) or unbiased (fair).

We will define the notions of fairness in terms of a *constraint function* $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Many of the common notions of fairness may be expressed or approximated as linear constraints on h (introduced previously by (Cotter et al., 2018c; Goh et al., 2016)). That is, they are of the form

$$\mathbb{E}_{x\sim\mathcal{P}}\left[\langle h(x), c(x)\rangle\right] = 0,$$

where $\langle h(x), c(x)\rangle := \sum_{y\in\mathcal{Y}} h(y|x)c(x, y)$ and we use the shorthand $h(y|x)$ to denote the probability of sampling y from a Bernoulli random variable with $p = h(x)$; i.e., $h(1|x) := h(x)$ and $h(0|x) := 1 - h(x)$. Therefore, a label function h is *unbiased* with respect to the constraint function c if $\mathbb{E}_{x\sim\mathcal{P}}\left[\langle h(x), c(x)\rangle\right] = 0$. If h is biased, the degree of bias (positive or negative) is given by $\mathbb{E}_{x\sim\mathcal{P}}\left[\langle h(x), c(x)\rangle\right]$.

We define the notions of fairness with respect to a protected group $\mathcal{G}$, and thus assume access to an indicator function $g(x) = 1[x \in \mathcal{G}]$. We use $Z_{\mathcal{G}} := \mathbb{E}_{x\sim\mathcal{P}}[g(x)]$ to denote the probability of a sample drawn from $\mathcal{P}$ to be in $\mathcal{G}$. We use $P_{\mathcal{X}} = \mathbb{E}_{x\sim\mathcal{P}}[y_{\text{true}}(x)]$ to denote the proportion of $\mathcal{X}$ which is positively labelled and $P_{\mathcal{G}} = \mathbb{E}_{x\sim\mathcal{P}}[g(x) \cdot y_{\text{true}}(x)]$ to denote the proportion of $\mathcal{X}$ which is positively labelled and in $\mathcal{G}$. We now give some concrete examples of accepted notions of constraint functions:

Demographic parity (Dwork et al., 2012): A fair classifier h should make positive predictions on $\mathcal{G}$ at the same rate as on all of $\mathcal{X}$. The constraint function may be expressed as $c(x, 0) = 0$, $c(x, 1) = g(x)/Z_{\mathcal{G}} - 1$.

Disparate impact (Feldman et al., 2015): This is identical to demographic parity, only that, in addition, during inference the classifier does not have access to the indicator function $g(x)$.

Equal opportunity (Hardt et al., 2016): A fair classifier h should have equal true positive rates on $\mathcal{G}$ as on all of $\mathcal{X}$. The constraint may be expressed as $c(x, 0) = 0$, $c(x, 1) = g(x)y_{\text{true}}(x)/P_{\mathcal{G}} - y_{\text{true}}(x)/P_{\mathcal{X}}$.

Equalized odds (Hardt et al., 2016): A fair classifier h should have equal true positive and false positive rates on $\mathcal{G}$ as on all of $\mathcal{X}$. In addition to the constraint associated with equal opportunity, this notion applies an additional constraint with $c(x, 0) = 0$, $c(x, 1) = g(x)(1 - y_{\text{true}}(x))/(Z_{\mathcal{G}} - P_{\mathcal{G}}) - (1 - y_{\text{true}}(x))/(1 - P_{\mathcal{X}})$.

In practice, there are often multiple fairness constraints $\{c_k\}_{k=1}^K$ associated with multiple protected groups $\{\mathcal{G}_k\}_{k=1}^K$. It is clear that our subsequent results will assume multiple fairness constraints and protected groups, and that the protected groups may have overlapping samples.

3 MODELING HOW BIAS ARISES

We now introduce our underlying mathematical framework to understand bias in the data, by providing the relationship between y_{bias} and y_{true} (Assumption 1 and Proposition 1). This will allow us to derive a closed form expression for y_{true} in terms of y_{bias} (Corollary 1). In Section 4 we will show how this expression leads to a simple algorithm that performs appropriate weighting on data with biased labels in order to train a classifier to produce the true, unbiased labels.

We begin with an assumption on the relationship between the observed y_{bias} and the underlying y_{true}.

Assumption 1. *Let the fairness constraints be $c_1, .., .c_K$, with respect to which y_{true} is unbiased (i.e. $\mathbb{E}_{x\sim\mathcal{P}}\left[\langle y_{\text{true}}(x), c_k(x)\rangle\right] = 0$ for $k \in [K]$). We assume that there exist $\epsilon_1, \ldots, \epsilon_K \in \mathbb{R}$ such that the observed, biased label function y_{bias} is the solution of the following constrained optimization problem:*

$$\arg\min_{\hat{y}:\mathcal{X}\to[0,1]} \mathbb{E}_{x\sim\mathcal{P}}\left[D_{\text{KL}}(\hat{y}(x)||y_{\text{true}}(x))\right]$$
$$\text{s.t.}\quad \mathbb{E}_{x\sim\mathcal{P}}\left[\langle \hat{y}(x), c_k(x)\rangle\right] = \epsilon_k \quad \text{for } k = 1, \ldots, K,$$

where we use D_{KL} to denote the KL-divergence.[2]

In other words, we assume that y_{bias} is the label function closest to y_{true} while achieving some amount of bias, where proximity to y_{true} is given by the KL-divergence. We note that this assumption is agnostic to the features (other than requiring that examples with equal features should have equal label functions). Thus it is also applicable to settings where the features themselves are biased and that the observed labels were generated by a process depending on the features (i.e. situations where there is bias in both the features and labels).

The KL-divergence in Assumption 1 presents a convex optimization problem, whose closed form is well-known (Friedlander and Gupta, 2006; Botev and Kroese, 2011). In our setting, the closed form provides us with the following simple expression for the observed y_{bias}. See the Appendix for a proof.

Proposition 1. *Suppose that Assumption 1 holds. Then for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, y_{bias} satisfies,*

$$y_{\text{bias}}(y|x) \propto y_{\text{true}}(y|x) \cdot \exp\Big\{-\sum_{k=1}^{K} \lambda_k \cdot c_k(x, y)\Big\},$$

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

[2] We use equality constraints to simplify the analysis. Inequality constraints may be used as well, with the same results.

Given this form of y_{bias} in terms of y_{true}, we can immediately deduce the form of y_{true} in terms of y_{bias}:

Corollary 1. *Suppose that Assumption 1 holds. The unbiased label function y_{true} is of the form,*

$$y_{\text{true}}(y|x) \propto y_{\text{bias}}(y|x) \cdot \exp\Big\{\sum_{k=1}^{K} \lambda_k \cdot c_k(x,y)\Big\},$$

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

We note that previous approaches to learning fair classifiers often formulate a constrained optimization problem similar to that appearing in Assumption 1 (i.e., maximize the accuracy or log-likelihood of a classifier subject to linear constraints) and then solve it, usually via the method of Lagrange multipliers which translates the constraints to penalties on the training loss. In our approach, rather than using the constrained optimization problem to formulate a training objective, we use it to express the generative process which modifies the true, underlying labels to the biased labels we observe. Furthermore, rather than training with respect to these biased labels, our approach aims to recover the true ones. As we will show in the following sections, this may be done by simply optimizing the training loss on a *re-weighting* of the dataset. In contrast, the penalties associated with Lagrangian approaches can often be cumbersome: The original, non-differentiable, fairness constraints must be relaxed or approximated before conversion to penalties. Even then, the derivatives of these approximations may be near-zero for large regions of the domain, causing difficulties during training.

4 LEARNING UNBIASED LABELS

We have derived a closed form expression for the true, unbiased label function y_{true} in terms of the observed label function y_{bias}, coefficients $\lambda_1, \ldots, \lambda_K$, and constraint functions $c_1, \ldots, c_K$. In this section, we elaborate on how one may learn a machine learning model h to fit y_{true}, given access to a dataset $\mathcal{D}$ with labels sampled according to y_{bias}. We begin by restricting ourselves to constraints $c_1, \ldots, c_K$ associated with demographic parity, allowing us to have full knowledge of these constraint functions. The same method may be extended to general notions of fairness, and we provide the details for this in the Appendix.

With knowledge of the functions $c_1, \ldots, c_K$, it remains to determine the coefficients $\lambda_1, \ldots, \lambda_K$ (which give us a closed form expression for the dataset weights) as well as the classifier h. We present our method by first showing how a classifier h may be learned assuming knowledge of the coefficients $\lambda_1, \ldots, \lambda_K$ (Section 4.1). We subsequently show how the coefficients themselves may be learned, thus allowing our algorithm to be used in general settings (Section 4.2).

4.1 Learning h Given $\lambda_1, \ldots, \lambda_K$

Although we have the closed form expression $y_{\text{true}}(y|x) \propto y_{\text{bias}}(y|x) \exp\{\sum_{k=1}^{K} \lambda_k c_k\}$ for the true label function, in practice we do not have access to the values $y_{\text{bias}}(y|x)$ but rather only access to data points with labels sampled from $y_{\text{bias}}(y|x)$. We propose the *weighting* technique to train h on labels based on y_{true}.[3] The weighting technique weights an example (x, y) by the weight $w(x,y) = \widetilde{w}(x,y)/\sum_{y' \in \mathcal{Y}} \widetilde{w}(x,y')$, where $\widetilde{w}(x,y') = \exp\{\sum_{k=1}^{K} \lambda_k c_k(x,y')\}$.

We have the following theorem, which states that weighting a loss according to $w(x,y)$ is equivalent to training with respect to the true, unbiased labels while biasing the feature distribution.

Theorem 1. *For any loss function ℓ, training a classifier h on the weighted objective $\mathbb{E}_{x\sim\mathcal{P}, y\sim y_{\text{bias}}(x)}[w(x,y) \cdot \ell(h(x),y)]$ is equivalent to training the classifier on the objective $\mathbb{E}_{x\sim\tilde{\mathcal{P}}, y\sim y_{\text{true}}(x)}[\ell(h(x),y)]$ with respect to the underlying, true labels, for some distribution $\tilde{\mathcal{P}}$ over $\mathcal{X}$.*

Theorem 1 states that the bias in observed labels may be corrected in a simple and straightforward way by re-weighting the training examples. Theorem 1 also explicitly describes the trade-off introduced: While we gain the ability to train on unbiased labels, we implicitly modify the feature distribution $\tilde{\mathcal{P}}$ over features x. In Section 5, we will resolve this issue and show that given some mild conditions (see Theorem 2), the change in feature distribution does not affect the bias of the final learned classifier. Therefore, in these cases, training with respect to weighted examples with biased labels is equivalent to training with respect to the same examples and the true labels.

4.2 Determining the Coefficients $\lambda_1, \ldots, \lambda_K$

We now continue to describe how to learn the coefficients $\lambda_1, \ldots, \lambda_K$. One advantage of our approach is that, in practice, K is often small. Thus, we propose to iteratively learn the coefficients so that the final classifier satisfies the desired fairness constraints either on the training data or on a validation set. We first discuss how to do this for demographic parity; we discuss extensions to other notions of fairness in the Appendix. See the full pseudocode for learning h and $\lambda_1, \ldots, \lambda_K$ in Algorithm 1.

Intuitively, the idea is that if the positive prediction rate for a protected class $\mathcal{G}$ is lower than the overall positive prediction rate, then the corresponding coefficient should be increased; i.e., if we increase the

[3]See the Appendix for an alternative to the weighting technique – the *sampling* technique, based on a coin-flip.

Algorithm 1 Training a fair classifier for Demographic Parity, Disparate Impact, or Equal Opportunity (see Appendix for Equalized Odds).

Inputs: Learning rate η, number of loops T, training data $\mathcal{D}_{[n]} = \{(x_i, y_i)\}_{i=1}^N$, classification procedure H. constraints $c_1, ..., c_K$ corresponding to protected groups $\mathcal{G}_1, ..., \mathcal{G}_K$.
Initialize $\lambda_1, ..., \lambda_K$ to 0 and $w_1 = w_2 = \cdots = w_n = 1$.
Let $h := H(\mathcal{D}_{[n]}, \{w_i\}_{i=1}^n)$
for $t = 1, ..., T$ **do**
  Let $\Delta_k := \mathbb{E}_{x\sim\mathcal{D}_{[n]}}[\langle h(x), c_k(x)\rangle]$ for $k \in [K]$.
  Update $\lambda_k = \lambda_k - \eta \cdot \Delta_k$ for $k \in [K]$.
  Let $\widetilde{w_i} := \exp\left(\sum_{k=1}^K \lambda_k \cdot 1[x \in \mathcal{G}_k]\right)$ for $i \in [n]$
  Let $w_i = \widetilde{w_i}/(1 + \widetilde{w_i})$ if $y_i = 1$, otherwise $w_i = 1/(1 + \widetilde{w_i})$ for $i \in [n]$
  Update $h = H(\mathcal{D}_{[n]}, \{w_i\}_{i=1}^n)$
end for
Return h

weights of the positively labeled examples of $\mathcal{G}$ and decrease the weights of the negatively labeled examples of $\mathcal{G}$, then this will encourage the classifier to increase its accuracy on the positively labeled examples in $\mathcal{G}$, while the accuracy on the negatively labeled examples of $\mathcal{G}$ may fall. Either of these two events will cause the positive prediction rate on $\mathcal{G}$ to increase, and thus bring h closer to the true, unbiased label function.

Accordingly, Algorithm 1 works by iteratively performing the following steps: (1) evaluate the demographic parity constraints; (2) update the coefficients by subtracting the respective constraint violation multiplied by a fixed step-size; (3) compute the weights for each sample based on these multipliers using the closed-form provided by Proposition 1; and (4) retrain the classifier given these weights. Algorithm 1 takes in a classification procedure H, which given a dataset $D_{[n]} := \{(x_i, y_i)\}_{i=1}^n$ and weights $\{w_i\}_{i=1}^n$, outputs a classifier. In practice, H can be any training procedure which minimizes a weighted loss function over some parametric function class (e.g. logistic regression).

4.3 Extensions to other fairness notions

The initial restriction to demographic parity was made so that the values of the constraint functions $c_1, \dots, c_K$ on any $x \in \mathcal{X}, y \in \mathcal{Y}$ would be known. We note that Algorithm 1 works for disparate impact as well: The only change would be that the classifier does not have access to the protected attributes. However, in other notions of fairness, such as equal opportunity or equalized odds, the constraint functions depend on y_{true}, which is unknown.

For these cases, we propose to apply the same technique of iteratively re-weighting the loss to achieve the desired fairness notion, with the weights $w(x, y)$ on each example determined only by the protected attribute $g(x)$ and the observed label $y \in \mathcal{Y}$. This is equivalent to using Theorem 1 to derive the same procedure presented in Algorithm 1, but approximating the unknown constraint function $c(x, y)$ as a piece-wise constant function $d(g(x), y)$, where $d : \{0, 1\} \times \mathcal{Y} \to \mathbb{R}$ is unknown. Although we do not have access to d, we may treat $d(g(x), y)$ as an additional set of parameters – one for each protected group attribute $g(x) \in \{0, 1\}$ and each label $y \in \mathcal{Y}$. These additional parameters may be learned in the same way the λ coefficients are learned. In some cases, their values may be wrapped into the unknown coefficients. For example, for equal opportunity, there is in fact no need for any additional parameters. On the other hand, for equalized odds, the unknown values for $\lambda_1, \dots, \lambda_K$ and $d_1, \dots, d_K$, are instead treated as unknown values for $\lambda_1^{TP}, \dots, \lambda_K^{TP}, \lambda_1^{FP}, \dots, \lambda_K^{FP}$; i.e., separate coefficients for positively and negatively labelled points. Due to space constraints, see the Appendix for further details on these and more general constraints. We explain how to use the procedure for Equal Opportunity below.

Equal Opportunity: Algorithm 1 can be directly used by replacing the demographic parity constraints with equal opportunity constraints. Recall that in equal opportunity, the goal is for the positive prediction rates on the positive examples of the protected group $\mathcal{G}$ to match that of the overall. If the positive prediction rate for positive examples $\mathcal{G}$ is less than that of the overall, then Algorithm 1 will up-weight the examples of $\mathcal{G}$ which are positively labeled. This encourages the classifier to be more accurate on the positively labeled examples of $\mathcal{G}$, which in other words means that it will encourage the classifier to increase its positive prediction rate on these examples, thus leading to a classifier satisfying equal opportunity. In this way, the same intuitions supporting the application of Algorithm 1 to demographic parity or disparate impact also support its application to equal opportunity. We note that in practice, we do not have access to the true labels function, so we approximate the constraint violation $\mathbb{E}_{x\sim\mathcal{D}_{[n]}}[\langle h(x), c_k(x)\rangle]$ using the observed labels as $\mathbb{E}_{(x,y)\sim\mathcal{D}_{[n]}}[h(x) \cdot c_k(x, y)]$.

5 THEORETICAL ANALYSIS

In this section, we provide theoretical guarantees on a learned classifier h using the weighting technique. We show that with the coefficients $\lambda_1, ..., \lambda_K$ that satisfy Proposition 1, training on the re-weighted dataset leads to a finite-sample non-parametric rates of consistency on the estimation error provided the classifier has sufficient flexibility.

We need to make the following regularity assumption on the data distribution, which assumes that the data is supported on a compact set in $\mathbb{R}^D$ and y_{bias} is smooth (i.e. Lipschitz).

Assumption 2. *$\mathcal{X}$ is a compact set over $\mathbb{R}^D$ and both $y_{\text{bias}}(x)$ and $y_{\text{true}}(x)$ are L-Lipschitz (i.e. $|y_{\text{bias}}(x) - y_{\text{bias}}(x')| \leq L \cdot |x - x'|$).*

We now give the result. The proof is technically involved and is deferred to the Appendix.

Theorem 2 (Rates of Consistency)**.** *Let $0 < \delta < 1$. Let $\mathcal{D}_{[n]} = \{(x_i, y_i)\}_{i=1}^n$ be a sample drawn from $\mathcal{D}$. Suppose that Assumptions 1 and 2 hold. Let $\mathcal{H}$ be the set of all $2L$-Lipschitz functions mapping $\mathcal{X}$ to $[0, 1]$. Suppose that the constraints are $c_1, ..., c_K$ and the corresponding coefficients $\lambda_1, ..., \lambda_K$ satisfy Proposition 1 where $-\Lambda \leq \lambda_k \leq \Lambda$ for $k = 1, ..., K$ and some $\Lambda > 0$. Let h^* be the optimal function in $\mathcal{H}$ under the weighted mean square error objective, where the weights satisfy Proposition 1. Then there exists C_0 depending on $\mathcal{D}$ such that for n sufficiently large depending on $\mathcal{D}$, we have with probability at least $1 - \delta$:*

$$||h^* - y_{\text{true}}||_2 \leq C_0 \cdot \log(2/\delta)^{1/(2+D)} \cdot n^{-1/(4+2D)}.$$

where $||h - h'||_2 := \mathbb{E}_{x\sim\mathcal{P}}[(h(x) - h'(x))^2]$.

Thus, with the appropriate values of λ_1,...,λ_K given by Proposition 1, we see that training with the weighted dataset based on these values will guarantee that the final classifier will be close to y_{true}. However, the above rate has a dependence on the dimension D, which may be unattractive in high-dimensional settings. See the Appendix for an intrinsic dimension dependent result.

6 RELATED WORK

Post-processing: One approach to fairness is to perform a post-processing of the classifier outputs. Examples of previous work in this direction include (Doherty et al., 2012; Feldman, 2015; Hardt et al., 2016). However, this approach of post-processing the outputs to encourage fairness has limited flexibility. (Pleiss et al., 2017) showed that a deterministic solution is only compatible with a single error constraint and thus cannot be applied to fairness notions such as equalized odds. Moreover, decoupling the training and fairness constraint can lead to models with poor accuracy trade-off. In fact (Woodworth et al., 2017) showed that in certain cases, post-processing can be provably suboptimal. Other works discussing the incompatibility of fairness notions include (Chouldechova, 2017; Kleinberg et al., 2016).

Lagrangian Approach: There has been much recent work done on enforcing fairness by transforming the constrained optimization problem via the method of Lagrange multipliers. Some works (Zafar et al., 2015; Goh et al., 2016) apply this to the convex setting. In the nonconvex case, there is work which frames the constrained optimization problem as a two-player game (Kearns et al., 2017; Agarwal et al., 2018; Cotter et al., 2018b) . Related approaches include (Edwards and Storkey, 2015; Corbett-Davies et al., 2017; Narasimhan, 2018). There is also recent work similar in spirit which encourages fairness by adding penalties to the objective; e.g, (Donini et al., 2018) studies this for kernel methods and (Komiyama et al., 2018) for linear models. However, the fairness constraints are often irregular and have to be relaxed in order to optimize. Notably, our method does not use the constraints directly in the model loss, and thus does not require them to be relaxed. Moreover, these approaches typically are not readily applicable to equality constraints as feasibility challenges can arise; thus, there is the added challenge of determining appropriate slack during training. Finally, the training can be difficult as (Cotter et al., 2018c) has shown that the Lagrangian may not even have a solution to converge to.

When the classification loss and the relaxed constraints have the same form (e.g. a hinge loss as in (Eban et al., 2017)), the resulting Lagrangian may be rewritten as a cost-sensitive classification, explicitly pointed out in (Agarwal et al., 2018), who show that the Lagrangian method reduces to solving an objective of the form $\sum_{i=1}^n w_i 1[h(x_i) \neq y_i']$ for some non-negative weights w_i. In this setting, y_i' may not necessarily be the true label, which may occur for example in demographic parity when the goal is to predict more positively within a protected group and thus may be penalized for predicting correctly on negative examples. In contrast, our approach is a non-negative re-weighting of the original loss (i.e., does not modify the observed labels) and is thus simpler and more aligned with intuition.

Pre-processing: This approach has primarily involved massaging the data to remove bias. Examples include (Calders et al., 2009; Kamiran and Calders, 2009; Žliobaite et al., 2011; Zemel et al., 2013; Fish et al., 2015; Feldman et al., 2015; Beutel et al., 2017). Many of these approaches involve changing the labels and features of the training set, which may have legal implications since it is a form of training on falsified data (Barocas and Selbst, 2016). In contrast, our approach does not modify the training data and only reweights the importance of certain sensitive groups. The re-weighting approach is simple and intuitive, thus it is natural that strategies similar to ours have been proposed (Kamiran and Calders, 2012; Krasanakis et al., 2018). These approaches have thus far come with few theoretical guarantees. Our approach is unique for pro-

Figure 2: **Results as λ changes**: We take the optimal $\lambda = \lambda^*$ found by Algorithm 1, and for each value x on the x-axis, we train with data weights based on the setting $\lambda = x \cdot \lambda^*$ and plot the error and violations. When $x = 1$, we train based on the λ found by Algorithm 1 and get the lowest fairness violation; $x = 0$ corresponds to training on the unweighted dataset and gives us the lowest prediction error. More charts are in the Appendix.

viding a mathematically grounded motivation for the reasons for and specific form of re-weighting. The motivation we employ is similar to (Calmon et al., 2017), where the authors propose to modify the data distribution to satisfy some constraints while staying close in a statistical sense to the original data. Our approach is more practical since (1) we focus on a KL-divergence on the labels, which leads to a simple re-weighting strategy, and (2) we provide an automatic method for tuning the modified data distribution (through λ).

7 EXPERIMENTS

7.1 Baselines

We compare results with the following baselines:

Unconstrained (Unc.): This method applies logistic regression on the dataset without any constraints.

Post-calibration (Cal.): This method Hardt et al. (2016) first trains without consideration for fairness, and then determines appropriate thresholds for the protected groups such that fairness is satisfied in training. In the case of overlapping groups, we treat each intersection as their own group.

Lagrangian approach (Lagr.) This general method used in several works Eban et al. (2017); Goh et al. (2016); Zafar et al. (2015); Cotter et al. (2018b) proceeds by jointly training the Lagrangian in both model parameters and Lagrange multipliers and uses a hinge approximation of the constraints to make the Lagrangian differentiable in its input. We fix the model learning rate and use ADAM optimizer with learning rate 0.01 and train for 100 epochs. We select a slack for the constraints as without slack, we often converged to degenerate solutions. We chose the smallest slack in increments of 5% until the procedure returned a non-degenerate solution.

Our method: We use Algorithm 1 for demographic parity, disparate impact, and equal opportunity and Algorithm 2 for equalized odds (see the appendix). We fix the learning rate $\eta = 1.$ and number of loops $T = 100$ across all of our experiments.

For all of the methods except for the Lagrangian, we train using Scikit-Learn's Logistic Regression Pedregosa et al. (2011) with default hyperparameter settings. We test our method against the unconstrained baseline, post-processing calibration, and the Lagrangian approach with hinge relaxation of the constraints. For all of the methods, we fix the hyperparameters across all experiments.

7.2 Benchmark Fairness Experiments

Our main results are presented in Table 1. We compare our method for learning fair classifiers on a number of benchmark datasets: Bank Marketing Lichman et al. (2013), Communities and Crime Lichman et al. (2013), ProPublica's COMPAS ProPublica (2018), German Statlog Credit Data Lichman et al. (2013), and Adult Lichman et al. (2013).

We see that our method consistently leads to more fair classifiers, often yielding a classifier with the lowest test violation out of all methods. We also include test error rates in the results. Although the primary objective of these algorithms is to yield a fair classifier, we find that our method is able to find reasonable trade-offs between fairness and accuracy. Our method often provides either better or comparative predictive error than the other fair classification methods (see Figure 2 for more insight into the trade-offs found by our algorithm).

The results in Table 1 also highlight the disadvantages of existing methods for training fair classifiers. Although the post-processing method is an improvement over an unconstrained model, it is often unable to find a classifier with lowest bias.

We also find the results of the Lagrangian approach to not consistently provide fair classifiers. As noted in previous work Cotter et al. (2018c); Goh et al. (2016), constrained optimization can be inherently unstable or requires a certain amount of slack in the objective as the constraints are typically relaxed to make gradient-based training possible and for feasibility purposes. Moreover, due to the added complexity of this method, it can overfit and have poor fairness generalization as

Table 1: **Experiment Results: Benchmark Fairness Tasks**: Each row corresponds to a dataset and fairness notion. We show the accuracy and fairness violation of training with no constraints (Unc.), with post-processing (PP), the Lagrangian approach (Lagr.) and our method. **Bolded** is the method achieving lowest fairness violation for each row. All reported numbers are evaluated on the test set.

	Metric	Unc. Err.	Unc. Vio.	PP Err.	PP Vio.	Lagr. Err.	Lagr. Vio.	Our Err.	Our Vio.
Bank	Dem. Par.	9.41%	.0349	9.70%	.0068	10.46%	.0126	9.63%	**.0056**
	Eq. Opp.	9.41%	.1452	9.55%	.0506	9.86%	.1237	9.48%	**.0431**
	Eq. Odds	9.41%	.1452	N/A	N/A	9.61%	.0879	9.50%	**.0376**
	Disp. Imp.	9.41%	.0304	N/A	N/A	10.44%	.0135	9.89%	**.0063**
COMPAS	Dem. Par.	31.49%	.2045	32.53%	.0201	40.16%	.0495	35.44%	**.0155**
	Eq. Opp.	31.49%	.2373	31.63%	**.0256**	36.92%	.1141	33.63%	.0774
	Eq. Odds	31.49%	.2373	N/A	N/A	42.69%	**.0566**	35.06%	.0663
	Disp. Imp.	31.21%	.1362	N/A	N/A	40.35%	.0499	42.64%	**.0256**
Comm.	Dem. Par.	11.62%	.4211	32.06%	.0653	28.46%	.0519	30.06%	**.0107**
	Eq. Opp.	11.62%	.5513	17.64%	**.0584**	28.45%	.0897	26.85%	.0833
	Eq. Odds	11.62%	.5513	N/A	N/A	28.46%	.0962	26.65%	**.0769**
	Disp. Imp.	14.83%	.3960	N/A	N/A	28.26%	.0557	30.26%	**.0073**
German	Dem. Par.	24.85%	.0766	24.85%	.0346	25.45%	.0410	25.15%	**.0137**
	Eq. Opp.	24.85%	.1120	24.54%	.0922	27.27%	.0757	25.45%	**.0662**
	Eq. Odds	24.85%	.1120	N/A	N/A	34.24%	.1318	25.45%	**.1099**
	Disp. Imp.	24.85%	.0608	N/A	N/A	27.57%	.0468	25.15%	**.0156**
Adult	Dem. Par.	14.15%	.1173	16.60%	.0129	20.47%	.0198	16.51%	**.0037**
	Eq. Opp.	14.15%	.1195	14.43%	.0170	19.67%	.0374	14.46%	**.0092**
	Eq. Odds	14.15%	.1195	N/A	N/A	19.04%	**.0160**	14.58%	.0221
	Disp. Imp.	14.19%	.1108	N/A	N/A	20.48%	**.0199**	17.37%	.0334

Table 2: **MNIST with Label Bias**

Method	Test Acc.
Trained on True Labels	97.85%
Unconstrained	88.18%
Post-processing	89.79%
Lagrangian	94.05%
Our Method	**97.00%**

noted in Cotter et al. (2018a). Accordingly, we find that the Lagrangian method often yields poor trade-offs in fairness and accuracy, at times yielding classifiers with both worse accuracy and more bias.

7.3 MNIST with Label Bias

We investigate the behavior of our method on a variant of the well-known MNIST task. We take the MNIST dataset under the standard train/test split and then randomly select 20% of the training data points and change their label to 2, yielding a biased set of labels. On such a dataset, our method should be able to find appropriate weights so that training on the weighted dataset roughly corresponds to training on the true labels. To this end, we train a classifier with a demographic-parity-like constraint on the predictions of digit 2; i.e., we encourage a classifier to predict the digit 2 at a rate of 10%, the rate appearing in the true labels. We compare to the same baseline methods as before.

We present the results in Table 2. We report test accuracy computed with respect to the true labels. We find that our method is the only one that is able to approach the accuracy of a classifier trained with respect to the true labels. Both unconstrained training or post-processing provide little improvement, with error rates greater than 10%. Compared to the next best method (the Lagrangian), our proposed technique improves error rate by roughly half. These results give further evidence of our method's ability to effectively train on the true labels despite only observing biased labels.

8 CONCLUSION

We presented a new framework to model how bias can arise in a dataset, assuming that there exists an unbiased ground truth. Our method for correcting for this bias is based on re-weighting the training examples. Given the appropriate weights, we showed with finite-sample guarantees that the learned classifier will be approximately unbiased. We gave practical procedures which approximate these weights and showed that the resulting algorithm leads to fair classifiers in a variety of settings.

Acknowledgements

We thank Maya Gupta, Andrew Cotter and Harikrishna Narasimhan for many insightful discussions and suggestions as well as Corinna Cortes for helpful comments.

References

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. `https://www.propublica.org/article/machine-bias-risk-assessments-in-/criminal-sentencing`, May 2016. (Accessed on 07/18/2018).

Sivaraman Balakrishnan, Srivatsan Narayanan, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Cluster trees on manifolds. In *Advances in Neural Information Processing Systems*, pages 2679–2687, 2013.

Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Cal. L. Rev.*, 104:671, 2016.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357, 2016.

Zdravko I Botev and Dirk P Kroese. The generalized cross entropy method, with applications to probability density estimation. *Methodology and Computing in Applied Probability*, 13(1):1–27, 2011.

Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *Data mining workshops, 2009. ICDMW'09. IEEE international conference on*, pages 13–18. IEEE, 2009.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3992–4001, 2017.

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

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. ACM, 2017.

Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. *arXiv preprint arXiv:1807.00028*, 2018a.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. *arXiv preprint arXiv:1804.06500*, 2018b.

Andrew Cotter, Heinrich Jiang, Serena Wang, Taman Narayan, Maya Gupta, Seungil You, and Karthik Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *arXiv preprint arXiv:1809.04198*, 2018c.

Neil A Doherty, Anastasia V Kartasheva, and Richard D Phillips. Information effect of entry into credit ratings market: The case of insurers' ratings. *Journal of Financial Economics*, 106(2): 308–330, 2012.

Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *arXiv preprint arXiv:1802.08626*, 2018.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. Scalable learning of non-decomposable objectives. In *Artificial Intelligence and Statistics*, pages 832–840, 2017.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

Michael Feldman. Computational fairness: Preventing machine-learned discrimination. 2015.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

Benjamin Fish, Jeremy Kun, and Ádám D Lelkes. Fair boosting: a case study. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning.* Citeseer, 2015.

Michael P Friedlander and Maya R Gupta. On minimizing distortion and relative entropy. *IEEE Transactions on Information Theory*, 52(1):238–245, 2006.

Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems*, pages 2415–2423, 2016.

Dominique Guegan and Bertrand Hassani. Regulatory learning: how to supervise machine learning models? an application to credit scoring. *The Journal of Finance and Data Science*, 2018.

Abel Ag Rb Guimaraes and Ghassem Tofighi. Detecting zones and threat on 3d body in security airports using deep learning machine. *arXiv preprint arXiv:1802.00565*, 2018.

Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

Heinrich Jiang. Density level set estimation on manifolds with dbscan. In *International Conference on Machine Learning*, pages 1684–1693, 2017.

Faisal Kamiran and Toon Calders. Classifying without discriminating. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1–6. IEEE, 2009.

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

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv preprint arXiv:1711.05144*, 2017.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In *International Conference on Machine Learning*, pages 2742–2751, 2018.

Emmanouil Krasanakis, Eleftherios Spyromitros-Xioufis, Symeon Papadopoulos, and Yiannis Kompatsiaris. Adaptive sensitive reweighting to mitigate bias in fairness-aware classification. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 853–862. International World Wide Web Conferences Steering Committee, 2018.

Moshe Lichman et al. Uci machine learning repository, 2013.

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

Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. ACM, 2008.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017.

ProPublica. Compas recidivism risk score data and analysis, Mar 2018. URL `https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis`.

Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953, 2017.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*, 2015.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

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

Appendix

A Sampling Technique

We present an alternative to the weighting technique. For the *sampling* technique, we note that the distribution $P(Y = y) \propto y_{\text{bias}}(y|x) \cdot \exp\left\{\sum_{k=1}^{K} \lambda_k c_k(x, y)\right\}$ corresponds to the conditional distribution,

$$P(A = y \text{ and } B = y | A = B),$$

where A is a random variable sampled from $y_{\text{bias}}(y|x)$ and B is a random variable sampled from the distribution $P(B = y) \propto \exp\left\{\sum_{k=1}^{K} \lambda_k c_k(x, y)\right\}$. Therefore, in our training procedure for h, given a data point $(x, y) \sim \mathcal{D}$, where y is sampled according to y_{bias} (i.e., A), we sample a value y' from the random variable B, and train h on (x, y) if and only if $y = y'$. This procedure corresponds to training h on data points (x, y) with y sampled according to the true, unbiased label function $y_{\text{true}}(x)$.

The sampling technique ignores or skips data points when $A \neq B$ (i.e., when the sample from $P(B = y)$ does not match the observed label). In cases where the cardinality of the labels is large, this technique may ignore a large number of examples, hampering training. For this reason, the *weighting* technique may be more practical.

B Algorithms for Other Notions of Fairness

Algorithm 2 Training a fair classifier for Equalized Odds.

Inputs: Learning rate η, number of loops T, training data $\mathcal{D}_{[n]} = \{(x_i, y_i)\}_{i=1}^{N}$, classification procedure H. True positive rate constraints $c_1^{TP}, ..., c_K^{TP}$ and false positive rate constraints $c_1^{FP}, ..., c_K^{FP}$ respectfully corresponding to protected groups $\mathcal{G}_1, ..., \mathcal{G}_K$.
Initialize $\lambda_1^{TP}, ..., \lambda_K^{TP}, \lambda_1^{FP}, ..., \lambda_K^{FP}$ to 0 and $w_1 = w_2 = \cdots = w_n = 1$. Let $h := H(\mathcal{D}_{[n]}, \{w_i\}_{i=1}^{n})$
for $t = 1, ..., T$ **do**
  Let $\Delta_k^A := \mathbb{E}_{x \sim \mathcal{D}_{[n]}}[\langle h(x), c_k^A(x) \rangle]$ for $k \in [K]$ and $A \in \{TP, FP\}$.
  Update $\lambda_k^A = \lambda_k^A - \eta \cdot \Delta_k^A$ for $k \in [K]$ and $A \in \{TP, FP\}$.
  $\widetilde{w_i}^T := \exp\left(\sum_{k=1}^{K} \lambda_k^{TP} \cdot 1[x \in \mathcal{G}_k]\right)$ for $i \in [n]$.
  $\widetilde{w_i}^F := \exp\left(-\sum_{k=1}^{K} \lambda_k^{FP} \cdot 1[x \in \mathcal{G}_k]\right)$ for $i \in [n]$.
  Let $w_i = \widetilde{w_i}^T / (1 + \widetilde{w_i}^T)$ if $y_i = 1$, otherwise $w_i = \widetilde{w_i}^F / (1 + \widetilde{w_i}^F)$ for $i \in [n]$
  Update $h = H(\mathcal{D}_{[n]}, \{w_i\}_{i=1}^{n})$
end for
Return h

Equalized Odds: Recall that equalized odds requires that the conditions for equal opportunity (regarding the true positive rate) to be satisfied and in addition, the false positive rates for each protected group match the false positive rate of the overall. Thus, as before, for each true positive rate constraint, we see that if the examples of $\mathcal{G}$ have a lower true positive rate than the overall, then up-weighting positively labeled examples in $\mathcal{G}$ will encourage the classifier to increase its accuracy on the positively labeled examples of $\mathcal{G}$, thus increasing the true positive rate on $\mathcal{G}$. Likewise, if the examples of $\mathcal{G}$ have a higher false positive rate than the overall, then up-weighting the negatively labeled examples of $\mathcal{G}$ will encourage the classifier to be more accurate on the negatively labeled examples of $\mathcal{G}$, thus decreasing the false positive rate on $\mathcal{G}$. This forms the intuition behind Algorithm 2. We again approximate the constraint violation $\mathbb{E}_{x \sim \mathcal{D}_{[n]}}[\langle h(x), c_k^A(x) \rangle]$ using the observed labels as $\mathbb{E}_{(x,y) \sim \mathcal{D}_{[n]}}[h(x) \cdot c_k^A(x, y)]$ for $A \in \{TP, FP\}$.

More general constraints: It is clear that our strategy can be further extended to any constraint that can be expressed as a function of the true positive rate and false positive rate over any subsets (i.e. protected groups) of the data. Examples that arise in practice include equal accuracy constraints, where the accuracy of certain subsets of the data must be approximately the same in order to not disadvantage certain groups, and high confidence samples, where there are a number of samples which the classifier ought to predict correctly and thus appropriate weighting can enforce that the classifier achieves high accuracy on these examples.

C Proof of Proposition 1

Proof of Proposition 1. The constrained optimization problem stated in Assumption 1 is a convex optimization with linear constraints. We may use the Lagrangian method to transform it into the following min-max problem:

$$\min_{\hat{y}:\mathcal{X}\to[0,1]} \max_{\lambda_1,\dots,\lambda_K\in\mathbb{R}} J(\hat{y}, \lambda_1, \dots, \lambda_K),$$

where $J(\hat{y}, \lambda_1, \dots, \lambda_K)$ is define as

$$\mathbb{E}_{x\sim\mathcal{P}}\left[D_{\mathrm{KL}}(\hat{y}(x)||y_{\mathrm{true}}(x)) + \sum_{k=1}^{K}\lambda_k(\langle\hat{y}(x), c_k(x)\rangle - \epsilon_k)\right].$$

Note that the KL-divergence may be written as an inner product:

$$D_{\mathrm{KL}}(\hat{y}(x)||y_{\mathrm{true}}(x)) = \langle\hat{y}(x), \log\hat{y}(x) - \log y_{\mathrm{true}}(x)\rangle.$$

Therefore, we have

$$J(\hat{y}, \lambda_1, \dots, \lambda_K) = \mathbb{E}_{x\sim\mathcal{P}}\left[\left\langle\hat{y}(x), \log\hat{y}(x) - \log y_{\mathrm{true}}(x) + \sum_{k=1}^{K}\lambda_k c_k(x)\right\rangle - \sum_{k=1}^{K}\lambda_k\epsilon_k\right].$$

In terms of $\hat{y}$, this is a classic convex optimization problem Botev and Kroese (2011). Its optimum is a Boltzmann distribution of the following form:

$$\hat{y}^*(y|x) \propto \exp\left\{\log y_{\mathrm{true}}(y|x) - \sum_{k=1}^{K}\lambda_k c_k(x,y)\right\}.$$

The desired claim immediately follows. □

D Proof of Theorem 1

Proof of Theorem 1. For a given x and for any $y \in \mathcal{Y}$, due to Corollary 1 we have,

$$y_{\mathrm{true}}(y|x) = \frac{\widetilde{w}(x,y)y_{\mathrm{bias}}(y|x)}{\sum_{y'\in\mathcal{Y}}\widetilde{w}(x,y')y_{\mathrm{bias}}(y'|x)} = \frac{w(x,y)y_{\mathrm{bias}}(y|x)}{\sum_{y'\in\mathcal{Y}}w(x,y')y_{\mathrm{bias}}(y'|x)}. \tag{1}$$

Denote the denominator as $\phi(x) = \sum_{y'\in\mathcal{Y}}w(x,y')y_{\mathrm{bias}}(y'|x)$, and note that this normalization only depends on x. We may write equation 1 as, $\phi(x)y_{\mathrm{true}}(y|x) = w(x,y)y_{\mathrm{bias}}(y|x)$. Now, letting $\tilde{\mathcal{P}}$ denote the feature distribution $\tilde{\mathcal{P}}(x) \propto \phi(x)\mathcal{P}(x)$, we have,

$$\mathbb{E}_{x\sim\mathcal{P},y\sim y_{\mathrm{bias}}(x)}[w(x,y)\cdot\ell(h(x),y)] = C\cdot\mathbb{E}_{x\sim\tilde{\mathcal{P}},y\sim y_{\mathrm{true}}(x)}[\ell(h(x),y)], \tag{2}$$

where $C = \mathbb{E}_{x\sim\mathcal{P}}[\phi(x)]$, and this completes the proof. □

E Proof of Theorem 2

Proof of Theorem 2. The corresponding weight for each sample (x, y) is $w(x)$ if $y = 1$ and $1 - w(x)$ if $y = 0$, where

$$w(x) := \sigma\left(\sum_{i=1}^{K}\lambda_k\cdot c_k(x,1)\right),$$

$\sigma(t) := \exp(t)/(1 + \exp(t))$. Then, we have by Proposition 1 that

$$y_{\mathrm{true}}(x) = \frac{w(x)\cdot y_{\mathrm{bias}}(x)}{w(x)\cdot y_{\mathrm{bias}}(x) + (1-w(x))(1-y_{\mathrm{bias}}(x))}. \tag{3}$$

Let us denote w_i the weight for example (x_i, y_i). Suppose that h^* is the optimal learner on the re-weighted objective. That is,

$$h^* \in \arg\min_{h \in \mathcal{H}} J(h)$$

where $J(h) := \frac{1}{n}\sum_{i=1}^n w_i \cdot \ell(h(x_i), y_i)$ and $\ell(\hat{y}, y) = (\hat{y} - y)^2$. Let us partition $\mathcal{X}$ into a grid of D-dimensional hypercubes with diameter R, and let this collection be Ω_R. For each $B \in \Omega_R$, let us denote the center of B as B_c. Define

$$J_R(h) := \frac{1}{n}\sum_{B \in \Omega_B} \left((1 - y_{\text{bias}}(B_c))(1 - w(B_c))h(B_c)^2 + y_{\text{bias}}(B_c)w(B_c)(1 - h(B_c))^2\right) \cdot |B \cap X_{[n]}|,$$

where $X_{[n]} := \{x_1, .., x_n\}$.

We now show that $|J_R(h) - J(h)| \le E_{R,n} := 6LR + \frac{C \cdot \log(2/\delta)}{n}\sum_{B \in \Omega_R}\sqrt{|B \cap X_{[n]}|}$. We have

$$\begin{aligned}
J(h) &= \frac{1}{n}\sum_{i=1}^n w_i \cdot \ell(h(x_i), y_i) = \frac{1}{n}\sum_{B \in \Omega_R}\sum_{x_i \in B} w_i \cdot \ell(h(x_i), y_i) \\
&= \frac{1}{n}\sum_{B \in \Omega_R}\left(\sum_{\substack{x_i \in B \\ y_i = 0}} w_i \cdot h(x_i)^2 + \sum_{\substack{x_i \in B \\ y_i = 1}} w_i \cdot (1 - h(x_i))^2\right) \\
&\le \frac{1}{n}\sum_{B \in \Omega_R}\left(\sum_{\substack{x_i \in B \\ y_i = 0}} w_i \cdot h(B_c)^2 + \sum_{\substack{x_i \in B \\ y_i = 1}} w_i \cdot (1 - h(B_c))^2\right) + 4L^2R^2 + 4LR \\
&\le \frac{1}{n}\sum_{B \in \Omega_R}\left((1 - y_{\text{bias}}(B_c))(1 - w(B_c))h(B_c)^2 + y_{\text{bias}}(B_c)w(B_c)(1 - h(B_c))^2\right) \cdot |B \cap X_{[n]}| \\
&\quad + 5LR + 4L^2R^2 + \frac{C \cdot \log(2/\delta)}{n}\sum_{B \in \Omega_R}\sqrt{|B \cap X_{[n]}|} \\
&= J_R(h) + 5LR + 4L^2R^2 + \frac{C \cdot \log(2/\delta)}{n}\sum_{B \in \Omega_R}\sqrt{|B \cap X_{[n]}|} \\
&\le J_R(h) + E_{R,n}.
\end{aligned}$$

where the first inequality holds by smoothness of h; the second inequality holds because the value of $w(x)$ for each $x \in B$ will be the same assuming that R is chosen sufficiently small to not allow examples from different protected attributes to be in the same $B \in \Omega_R$ and then applying Bernstein's concentration inequality so that this holds with probability at least $1 - \delta$ for some constant $C > 0$; finally the last inequality holds for R sufficiently small. Similarly, we can show that $J(h) \ge J_R(h) - E_{R,n}$, as desired.

It is clear that $y_{\text{true}} \in \arg\min_{h \in \mathcal{H}} J_R(h)$.

We now bound the amount h^* can deviate from y_{true} at B_C on average. Let $h^*(B_c) = y_{\text{true}}(B_c) + \epsilon_B$. Then, we have

$$2E_{R,n} \ge J_R(h^*) - J_R(y_{\text{true}})$$

because otherwise,

$$J(h^*) \ge J_R(h^*) - E_{R,n} > J_R(y_{\text{true}}) + E_{R,n} \ge J(y_{\text{true}}),$$

contradicting the fact that h^* minimizes J.

We thus have

$$\begin{aligned}2E_{R,n} &\geq J_R(h^*) - J_R(y_{\text{true}}) \\ &= \frac{1}{n}\sum_{B\in\Omega_B}\Big((1-y_{\text{bias}}(B_c))(1-w(B_c))(h^*(B_c)^2 - y_{\text{true}}(B_c)^2) \\ &\qquad\qquad + y_{\text{bias}}(B_c)w(B_c)((1-h^*(B_c))^2 - (1-y_{\text{true}}(B_c))^2)\Big)\cdot|B\cap X_{[n]}| \\ &= \frac{1}{n}\sum_{B\in\Omega_B}\epsilon_B^2\Big((1-y_{\text{bias}}(B_c))(1-w(B_c)) + y_{\text{bias}}(B_c)w(B_c)\Big)\cdot|B\cap X_{[n]}| \\ &\geq \frac{\exp(-K\Lambda)}{1+\exp(-K\Lambda)}\cdot\frac{1}{n}\sum_{B\in\Omega_B}\epsilon_B^2\cdot|B\cap X_{[n]}|,\end{aligned}$$

where the last inequality follows by lower bounding $\min\{w(B_C), 1-w(B_C)\}$ in terms of Λ.

Thus,

$$\frac{1}{n}\sum_{B\in\Omega_B}\epsilon_B^2\cdot|B\cap X_{[n]}| \leq \frac{2(1+\exp(-K\Lambda))}{\exp(-K\Lambda)}\cdot E_{R,n}.$$

By the smoothness of h^* and y_{true}, it follows that

$$\frac{1}{n}\sum_{x\in X_{[n]}}(h^*(x) - y_{\text{true}}(x))^2 \leq \frac{4(1+\exp(-K\Lambda))}{\exp(-K\Lambda)}\cdot E_{R,n}.$$

All that remains is to bound this quantity. By Lemma 1, there exists constant C_1 such that $\frac{1}{n}\sum_{B\in\Omega_R}\sqrt{|B\cap X_{[n]}|} \leq C_1\sqrt{\frac{1}{nR^D}}$. We now see that for n sufficiently large depending on the data distribution (with the understanding that $R\to 0$ and thus it suffices to not consider dominated terms), there exists constant C'' such that the above is bounded by

$$C''\left(R + \log(2/\delta)\sqrt{\frac{1}{nR^D}}\right).$$

Now, choosing $R = n^{-1/(2+D)}\cdot\log(2/\delta)^{2/(2+D)}$, we obtain the desired result. □

Lemma 1. *There exists constant C_1 such that*

$$\frac{1}{n}\sum_{B\in\Omega_R}\sqrt{|B\cap X_{[n]}|} \leq C_1\sqrt{\frac{1}{nR^D}}.$$

Proof. We have by Root-Mean-Square Arithmetic-Mean Inequality that

$$\begin{aligned}\frac{1}{n}\sum_{B\in\Omega_R}\sqrt{|B\cap X_{[n]}|} &= \frac{|\Omega_R|}{n}\frac{1}{|\Omega_R|}\sum_{B\in\Omega_R}\sqrt{|B\cap X_{[n]}|} \\ &\leq \frac{|\Omega_R|}{n}\sqrt{\frac{1}{|\Omega_R|}\sum_{B\in\Omega_R}|B\cap X_{[n]}|} = \sqrt{\frac{|\Omega_R|}{n}} \leq C_1\sqrt{\frac{1}{nR^D}},\end{aligned}$$

for some $C_1 > 0$ where the inequality follows because the support $\mathcal{X}$ is compact and thus the size of the hypercube partition will grow at a rate of $1/R^D$. □

F Intrinsic dimension dependent and ambient dimension independent result

If the data lies on a d-dimensional submanifold, then Theorem 3 below says that without any changes to the procedure, we will enjoy a rate that depends on the manifold dimension and independent of the ambient dimension. Interestingly, these rates are attained without knowledge of the manifold or its dimension.

Theorem 3 (Rates on Manifolds)**.** *Suppose that all of the conditions of Theorem 2 hold and that in addition, $\mathcal{X}$ is a d-dimensional Riemannian submanifold of $\mathbb{R}^D$ with finite volume and finite condition number. Then there exists C_0 depending on $\mathcal{D}$ such that for n sufficiently large depending on $\mathcal{D}$, we have with probability at least $1-\delta$:*

$$||h^* - y_{\text{true}}||_2 \leq C_0 \cdot \log(2/\delta)^{1/(2+d)} \cdot n^{-1/(4+2d)}.$$

We now proceed with the proof for Theorem 3.

The following gives a lower bound on the volume of a ball in $\mathcal{X}$. The result follows from Lemma 5.3 of Niyogi et al. (2008) and has been used before e.g. Balakrishnan et al. (2013); Jiang (2017) so we omit the proof.

Lemma 2. *Let $\mathcal{X}$ be a compact d-dimensional Riemannian submanifold of $\mathbb{R}^D$ with finite volume and finite condition number $1/\tau$. Suppose that $0 < r < 1/\tau$. Then,*

$$Vol_d(B(x,r) \cap \mathcal{X}) \geq v_d r^d (1 - \tau^2 r^2),$$

where v_d denotes the volume of a unit ball in $\mathbb{R}^d$ and Vol_d is the volume w.r.t. the uniform measure on $\mathcal{X}$.

We now give an analogue to Lemma 1 but for the manifold setting.

Lemma 3. *Suppose the conditions of Theorem 3. There exists constant C_1 such that*

$$\frac{1}{n} \sum_{B \in \Omega_R} \sqrt{|B \cap X_{[n]}|} \leq C_1 \sqrt{\frac{1}{nR^d}}.$$

Proof. We have by Lemma 2 that there exists $C_2 > 0$ such that for each $B \in \Omega_R$, and R sufficiently small, $\text{Vol}_d(B \cap \mathcal{X}) \geq C_2 \cdot R^d$. Thus,

$$|\Omega_R| \leq \frac{\text{Vol}_d(\mathcal{X})}{C_2 \cdot R^d}.$$

We then continue as in Lemma 1 proof and have by Root-Mean-Square Arithmetic-Mean Inequality that

$$\frac{1}{n} \sum_{B \in \Omega_R} \sqrt{|B \cap X_{[n]}|} = \frac{|\Omega_R|}{n} \frac{1}{|\Omega_R|} \sum_{B \in \Omega_R} \sqrt{|B \cap X_{[n]}|}$$
$$\leq \frac{|\Omega_R|}{n} \sqrt{\frac{1}{|\Omega_R|} \sum_{B \in \Omega_R} |B \cap X_{[n]}|} = \sqrt{\frac{|\Omega_R|}{n}} \leq C_1 \sqrt{\frac{1}{nR^d}},$$

for some $C_1 > 0$, as desired. □

Proof of Theorem 3. The proof is the same as that of Theorem 2 except we can now replace the usage of Lemma 1 with Lemma 3 to obtain rates in terms of d instead of D. □

G Additional Experimental Results

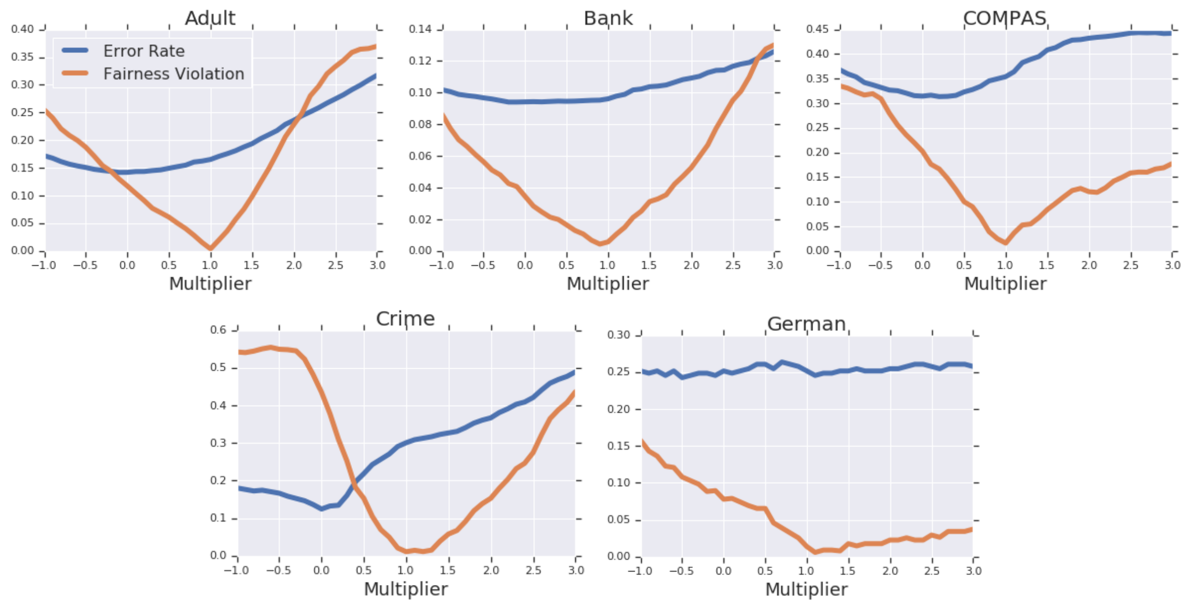

Figure 3: **Results as λ changes**: we show test error and fairness violations for demographic parity as the weightings change. We take the optimal $\lambda = \lambda^*$ found by Algorithm 1. Then for each value c on the x axis, and we train a classifier with data weights based on the setting $\lambda = c \cdot \lambda^*$ and plot the error and violations. We see that indeed, when $c = 1$, we train based on the λ found by Algorithm 1 and thus get the lowest fairness violation. For $c = 0$, this corresponds to training on the unweighted dataset and gives us the lowest error.

H Further Experimental Details

H.1 Datasets

Bank Marketing Lichman et al. (2013) (45211 examples). The data is based on a direct marketing campaign of a banking institution. The task is to predict whether someone will subscribe to a bank product. We use age as a protected attribute: 5 protected groups are determined based on uniform age quantiles.

Communities and Crime Lichman et al. (2013) (1994 examples). Each datapoint represents a community and the task is to predict whether a community has high (above the 70-th percentile) crime rate. We pre-process the data consistent with previous works, e.g. Cotter et al. (2018a) and form the protected group based on race in the same way as done in Cotter et al. (2018a). We use four race features as real-valued protected attributes corresponding to percentage of White, Black, Asian and Hispanic. We threshold each at the median to form 8 protected groups.

ProPublica's COMPAS ProPublica (2018) Recidivism data (7918 examples). The task is to predict recidivism based on criminal history, jail and prison time, demographics, and risk scores. The protected groups are two race-based (Black, White) and two gender-based (Male, Female).

German Statlog Credit Data Lichman et al. (2013) (1000 examples). The task is to predict whether an individual is a good or bad credit risk given attributes related to the individual's financial situation. We form two protected groups based on an age cutoff of 30 years.

Adult Lichman et al. (2013) (48842 examples). The task is to predict whether the person's income is more than 50k per year. We use 4 protected groups based on gender (Male and Female) and race (Black and White). We follow an identical procedure to Zafar et al. (2015); Goh et al. (2016) to pre-process the dataset.