
Fair Wrapping for Black-box Predictions

Alexander Soen
Australian National University
alexander.soen@anu.edu.au

Ibrahim Alabdulmohsin
Google Research
ibomohsin@google.com

Sanmi Koyejo
Google Research
Stanford University
sanmik@google.com

Yishay Mansour
Google Research
Tel Aviv University
mansour@google.com

Nyalleng Moorosi
Google Research
nyalleng@google.com

Richard Nock
Google Research
Australian National University
richardnock@google.com

Ke Sun
Australian National University
CSIRO's Data61
sunk@ieee.org

Lexing Xie
Australian National University
lexing.xie@anu.edu.au

Abstract

We introduce a new family of techniques to post-process (“wrap”) a black-box classifier in order to reduce its bias. Our technique builds on the recent analysis of improper loss functions whose optimization can correct any *twist* in prediction, unfairness being treated as a twist. In the post-processing, we learn a wrapper function which we define as an α -tree, which modifies the prediction. We provide two generic boosting algorithms to learn α -trees. We show that our modification has appealing properties in terms of composition of α -trees, generalization, interpretability, and KL divergence between modified and original predictions. We exemplify the use of our technique in three fairness notions: conditional value-at-risk, equality of opportunity, and statistical parity; and provide experiments on several readily available datasets.

1 Introduction

The social impact of Machine Learning (ML) has seen a dramatic increase over the past decade – enough so that the bias of model outputs must be accounted for alongside accuracy [1, 14, 35]. Considering the various number of fairness targets [20] and the energy and CO2 footprint of ML [19, 28], the combinatorics of training accurate *and* fair models is non-trivial. This is especially so given the inherent incompatibilities of fairness constraints [17] and the underlying tension of satisfying fairness whilst maintaining accuracy. One trend in the field “decouples” the two constraints by post-processing *pretrained* (accurate) models to achieve fairer outputs [35]. Post-processing may be the only option if we have no access to the model’s training data / algorithm / hyperparameters (etc.).

Within the post-processing approach, three trends have emerged: learning a new fair model close to the black-box, tweaking the output subject to fairness constraints, and exploiting sets of classifiers. If the task is class probability estimation [24], the estimated black-box is an accurate but potentially unfair posterior $\eta_v : \mathcal{X} \rightarrow [0, 1]$ which neither can be opened nor trained further. The goal is then to learn a fair posterior η_f from it. In addition to the black-box constraint, a number of *desiderata* can be considered for post-processing. Ideally in correcting a black-box, we would want the approach to have (**flexibility**) in satisfying substantially different fairness criteria, (**proximity**) of the learnt

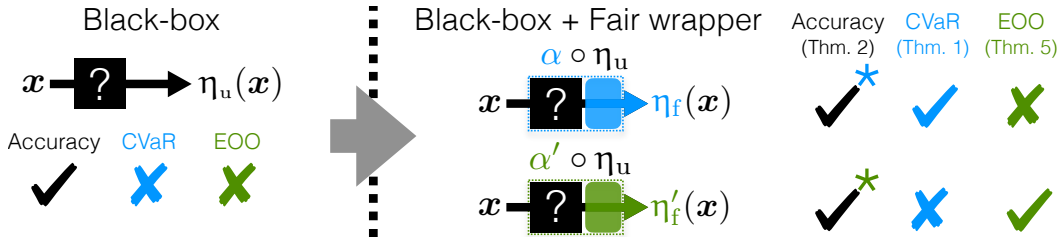


Figure 1: Summary of using different α -correction wrappers to obtain different fairness criteria guarantees. See Sections 5 and 6 for full details on guarantees.

η_f to the original η_u , and meaningful (**composability**) properties if, *e.g.*, η_f was later treated as a new black-box to be post-processed. To facilitate a specific style of correction, we may also want the representation of the correction to facilitate (**explainability**) for auditing the post-processing procedure and bounds on the increased model (**complexity**) of the final classifier η_f . In training the correction, algorithmically we would also want guarantees for (**convergence**).

Our contribution satisfies the aforementioned desiderata in its correction, representation, and algorithmic guarantees. By leveraging recent theory in *improper* loss functions, we utilize a *universal* correction of black-box posteriors defined by the α -loss. This allows for a flexible correction which yields convenient divergence bounds between η_f and η_u , a convenient form for the Rademacher complexity of the class of η_f , and a simple composability property. Representation-wise, the corrections we learn are easy-to-interpret tree-shaped functions that we define as α -trees. Algorithmically speaking, we provide two formal boosting algorithms to learn α -trees building upon seminal results [15]. We demonstrate our algorithm for conditional value-at-risk (CVAR) [32], equality of opportunity (EOO), and statistical parity; as depicted in Fig. 1. Experiments are provided against five baselines on readily available datasets. All proofs and more experiments are in an Appendix denoted as SI.

2 Related Work

Post-processing models to achieve fairness is one of three different categories in tackling the ML + fairness challenge [35, Section 6.2]. Although other notions exist, *e.g.* individual fairness [12], we limit our analysis to group fairness, which concerns itself with ensuring that statistics of sub-populations are similar. We further segment this cluster into three subsets: (I) approaches learning a new model with two constraints: being close to the pretrained model and being fair [16, 22, 31, 34]; (II) approaches biasing the output of the pretrained model at classification time, modifying observations for fairer outcomes [1, 14, 18, 21, 33, 34]; and (III) techniques consisting of exploiting sets of models to achieve fairness [11]. None of these approaches formulates substantial guarantees on all of the desiderata in the introduction. Some bring contributions with the (**flexibility**) of being applicable to more than two fairness notions [7, 31, 11, 34]. Two of which provide the convenience of analytic conditions on new fairness notions to fit in the approach [31, 11]. However, for all of them, the algorithmic price-tag is unclear [7, 11] or heavily depends on convex optimization routines [31]. [1, 34] provide strong guarantees regarding (**proximity**), w.r.t. *consistency and generalization*. To our knowledge, our approach of correcting prediction unfairness through improper losses [29] is new.

3 Setting and Motivating Example

Let \mathcal{X} be a domain of observations, $\mathcal{Y} \doteq \{-1, 1\}$ be labels and S is a sensitive attribute in \mathcal{X} . We assume that the modalities of S induce a partition of \mathcal{X} . We further let D denote the joint measure over $\mathcal{X} \times \mathcal{Y}$, M denote the marginal measure over \mathcal{X} , and $\pi \doteq \mathbb{P}[Y = 1]$ being the prior. We denote conditioning of M through a subscript, *e.g.*, M_s for $s \in S$ denotes M conditioned on a sensitive attribute subgroup $S = s$. We leave the σ -algebra to be implicit (which is assumed to be the same everywhere). As is often assumed in ML, sampling is i.i.d.; we make no notational distinction between empirical and true measures to simplify exposition – most of our results apply for both.

Consider the task of learning a function $\eta \in [0, 1]^{\mathcal{X}}$ to estimate the true posterior $\eta^*(x) = \mathbb{P}[Y = 1 | X = x]$ in binary classification. For instance, we may want to predict the probability of hiring an applicant for a company. Given the pointwise loss $L(\eta(X), \eta^*(X))$, which determines the loss of a

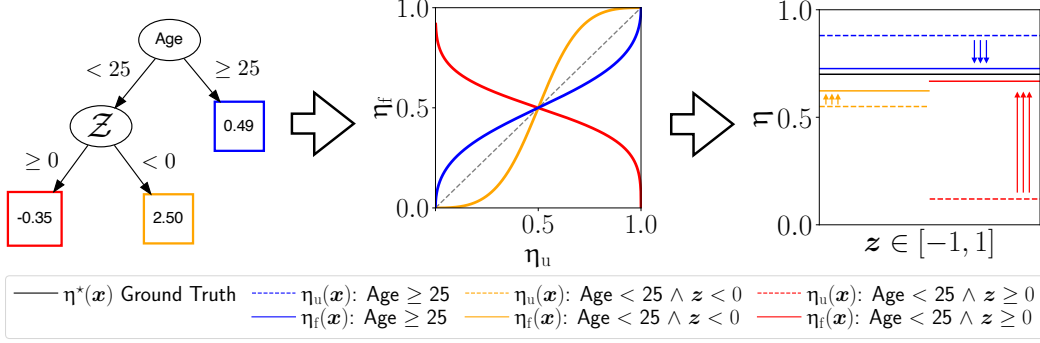


Figure 2: Improving CVAR for a toy hiring task with α -trees. An α -tree (left) transforms the posterior via (3) (middle); resulting in an input-dependent fairness correction of the posterior η_u (right).

single example, the (total) risk is defined as

$$L(\eta; M, \eta^*) \doteq \mathbb{E}_{X \sim M} [L(\eta(X), \eta^*(X))] \quad (1)$$

(with slight abuse of notation). A low risk corresponds to good classification performance. In this paper, we consider the risk determined by the *log-loss*:

$$L(\eta; M, \eta^*) \doteq \mathbb{E}_{X \sim M} [\eta^*(X) \cdot -\log \eta(X) + (1 - \eta^*(X)) \cdot -\log(1 - \eta(X))]. \quad (2)$$

We now consider a simple fairness problem, centered around the example of Fig. 2. Suppose that we are given a black-box η_u which predicts hiring probabilities without considering fairness. Although the minimized total risk of (1) might be small, there can be discrepancies in the performance between different subgroups. Instead of considering the total risk, the predictive performance of specific subgroups can be examined through the *subgroup risk* $L(\eta_u; M_s, \eta^*)$, for a subgroup $s \in S$. For instance, we might want to examine the discrepancy of subgroup risks among the age of applications. A natural fairness task would be to improve the worst performing subgroup, say $s_w \in S$.

To post-process unfairness, we want to learn a function $\alpha : \mathcal{X} \rightarrow \mathbb{R}$ which “wraps” η_u and *lowers* the worst subgroup risk. We propose the following “wrapping”, inspired by improper loss functions [29]

$$\eta_f(\mathbf{x}) \doteq \frac{\eta_u(\mathbf{x})^{\alpha(\mathbf{x})}}{\eta_u(\mathbf{x})^{\alpha(\mathbf{x})} + (1 - \eta_u(\mathbf{x}))^{\alpha(\mathbf{x})}} \in [0, 1]. \quad (3)$$

Notice when $\alpha(\mathbf{x}) = 1$ the resulting posterior is the original $\eta_u(\mathbf{x})$. Importantly, (3) is flexible enough to transform any input black-box η_u to any needed η_f . Looking at Fig. 2 (left and middle), intuitively by setting different $\alpha(\mathbf{x})$ values, (3) “sharpen” (yellow, $\alpha > 1$), “dampens” (blue, $0 < \alpha < 1$), or “polarity reverses” (red, $\alpha < 0$) the original posterior η_u . To improve fairness, we need a combination of these corrections to accommodate different subsets of the input domain (thus learning $\alpha(\cdot)$ as a function). We specifically learn $\alpha(\cdot)$ to be a tree structure, which allows an interpretable correction alongside other formal properties (Section 5).

Definition 1. An α -tree is a rooted, directed binary tree, with internal nodes labeled with observation variables. Outgoing arcs are labeled with tests over the nodes’ variable. Leaves are real valued. $\Lambda(\Upsilon)$ is the leafset of α -tree Υ . An α -tree induces a correction over posteriors as per (3) with $\alpha = \Upsilon$.

Our fairness problem is now learning an α -tree η which provides a corresponding correction that improves the worst subgroup risk, *i.e.*, $L(\eta_u; M_{s_w}, \eta^*) > L(\eta_f; M_{s_w}, \eta^*)$. The entirety of Fig. 2 presents such a process. In this hiring task example, the ground truth hiring rate is constant *w.r.t.* the inputs, $\eta^*(\mathbf{x}) = 0.7$. Despite this, the black-box $\eta_u(\cdot)$ unfairly depends on the age of applicants and incorrectly depends on a noise feature Z . By choosing $\alpha(\mathbf{x})$ as per Fig. 2 (left), the correction changes η_u to be closer to η^* , improving the risk of the worst subgroup (alongside the other subgroup in this example): the worse-case loss improves from 1.09 to 0.62.

Although the fairness criteria discussed might be considered simple, the procedure of iteratively minimizing the worse performing subgroup can be used to improve the *conditional value-at-risk* (lower is better) fairness criteria [32]:

$$\text{CVAR}_\beta(\eta_f) \doteq \mathbb{E}_S [L(\eta_f; M_S, \eta^*) \mid L(\eta_f; M_S, \eta^*) \geq L_\beta], \quad (4)$$

Algorithm 1 TOPDOWN ($M_t, \eta_t, \Upsilon_0, B$)

Input mixture M_t , posterior η_t , α -tree Υ_0 , $B \in \mathbb{R}_{++}$;

Step 1: $\Upsilon \leftarrow \Upsilon_0$;

Step 2 : **while** stopping condition not met **do**

Step 2.1 : pick leaf $\lambda^* \in \Lambda(\Upsilon)$; // i.e. heaviest leaf

Step 2.2 : $h^* \leftarrow \arg \min_{h \in \mathcal{H}} H(\Upsilon(\lambda^*, h); M_t, \eta_t)$;

Step 2.3 : $\Upsilon \leftarrow \Upsilon(\lambda^*, h^*)$; // split using h^* at λ^*

Step 3 : label leaves $\forall \lambda \in \Lambda(\Upsilon)$:

$$\Upsilon(\lambda) \doteq \tilde{\tau} \left(\frac{1 + e(M_\lambda, \eta_t)}{2} \right), \text{ // } \alpha\text{-value (5)}$$

Output Υ ;

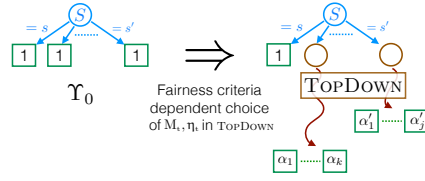


Figure 3: Picking Υ_0 a stump on the fairness attribute allows to finely tune growths of sub- α -trees to the fairness criterion at hand.

where L_β is the risk value for the β quantile among subgroups, which is user-defined. The difference is that CVAR_β not only considers the worse case subgroup, but all subgroups above the L_β risk value (let these subgroups be \mathcal{S}_β). One can simply iteratively improve all $s \in \mathcal{S}_\beta$, as done in the example of Fig. 2. Indeed, in the example, CVAR_β with $\beta = 0.9$ improves from 1.09 to 0.62 (which is equivalent to worse-case loss in this case).

4 Growing Alpha-Trees

We now introduce the procedure to grow an α -tree via a boosting algorithm TOPDOWN, Algorithm 1. TOPDOWN can be thought of as a generalization of the standard decision tree induction used for classification [15]. We first introduce relevant concepts from decision tree induction to explain TOPDOWN. We contextualize TOPDOWN through its application in improving the CVAR criteria.

We first introduce a technical assumption for the black-box η_u to be post-processed:

Assumption 1. *The black-box prediction is bounded away from the extremes: $\exists B > 0$ such that*

$$\text{Im}(\eta_u) \subseteq \mathbb{I} \doteq [(1 + \exp(B))^{-1}, (1 + \exp(-B))^{-1}] \quad (a.s.). \quad (6)$$

Compliance with Assumption 1 can be done by clipping the black-box’s output with a user-fixed B or making sure it is calibrated and then finding B .

Entropy-based updates for fairness: An important component in standard decision tree induction is the *edge* function, which measures the *label purity* (proportion of positive examples) of a decision tree node. We introduce a generalization which considers the *alignment purity* of a black-box.

Definition 2. *Let $\iota(u) \doteq \log(u/(1-u))$ the logit of $u \in (0, 1)$ and $\tilde{\tau}(u) \doteq \iota(u)/B$ a normalization which satisfies $\tilde{\tau}(\mathbb{I}) = [-1, 1]$. The **alignment edge** of M_t and η_t is defined as,*

$$e(M_t, \eta_t) \doteq \mathbb{E}_{(X, Y) \sim D_t} [Y \tilde{\tau}(\eta_u(X))], \quad (7)$$

where D_t is the joint measure induced by M_t and η_t . With Assumption 1 $e(M_t, \eta_t) \in [-1, 1]$.

By replacing the normalized logit $\tilde{\tau}$ with a constant 1, (7) reduces to a measure of label purity used in regular classification. In our case, (7) measures how well the black-box η_u “aligns” with the true labels Y through the logit. This also takes into account the “confidence” of the black-box’s predictions: for a high alignment purity, predictions not only need to be correct but also to be highly confident (η_u close to the endpoints of \mathbb{I}). Similar to how the splits of a decision tree classifier are determined by the entropy of a tree’s label purity, an α -tree splits based on its alignment entropy.

Definition 3. *Given an α -tree Υ with leafset Λ , when Assumption 1 is satisfied, the **entropy of Υ** is:*

$$H(\Upsilon; M_t, \eta_t) \doteq \mathbb{E}_{\lambda \sim M_{\Lambda(\Upsilon)}} [H_1(\lambda; M_t, \eta_t)], \quad (8)$$

where $H(q) \doteq -q \log(q) - (1-q) \log(1-q)$, $H_1(\lambda; M_t, \eta_t) \doteq H((1 + e(M_\lambda, \eta_t))/2)$, M_λ is M_t conditioned to leaf $\lambda \in \Lambda(\Upsilon)$, and $M_{\Lambda(\Upsilon)}$ is a measure induced on $\Lambda(\Upsilon)$ by leaf weights on M_t .

Theorem 1. *For any M_t, η_t , let Υ ’s leaves follow (5). Then $L(\eta_t; M_t, \eta_t) \leq H(\Upsilon; M_t, \eta_t)$.*

Algorithm 1 can now be explained by repeatedly leveraging Theorem 1. Suppose that we have a hypothesis set of possible splits \mathcal{H} to grow our α -tree. Denote $\Upsilon(\lambda, h)$ as the α -tree Υ split at leaf

$\lambda \in \Lambda(\Upsilon)$ using test h . The inner loop within Step 2 is the process of finding the best possible leaf splits which helps to minimize the α -tree’s entropy and to reduce the risk as per Theorem 1. The α -values of (5) calculated in step 3 are those used to ensure Theorem 1 holds. By setting $M_t \leftarrow M_{s_w}$ and $\eta_t \leftarrow \eta^*$, Algorithm 1 improves CVAR by iteratively improving the α -tree’s worst subgroup entropy $H(\Upsilon; M_{s_w}, \eta^*)$, which as a surrogate improves the worst subgroup risk $L(\eta_u; M_{s_w}, \eta^*)$. To accommodate for different β quantiles values for CVAR, TOPDOWN can be run repeatedly (replacing the initial input tree Υ_0) to progressively improve all $s \in \mathcal{S}_\beta$. Hence, to reduce CVAR, we basically

use TOPDOWN with $M_t \leftarrow M_s$ ($s \in \mathcal{S}_\beta$) and $\eta_t \leftarrow \eta^*$. (CVAR)

As alluded to by the notation used to instantiate TOPDOWN, the inputs of the algorithm can be instantiated to optimize for fairness criteria beyond CVAR. This is discussed in Section 6. In the usual ML setting, M_t, η_t can be *estimated* from a training sample (see Section 7).

Initialization: In the procedure of improving CVAR, the worst subgroups can be iteratively improved. However, we also need to make sure that improvement of a subgroup does not adversely affect another subgroup (which could potentially harm CVAR instead). As such, we introduce an additional structure to the α -tree Υ by tweaking the initial tree structure Υ_0 used in TOPDOWN. Since the fairness attribute S partitions the dataset, a convenient choice of initializing the α -tree is to split by the subgroup modalities, as depicted in Fig. 3. As such, we grow separate sub- α -trees for each of the sensitive modalities. For CVAR, this allows the subgroup risk of individual subgroups to be tweaked without adversely affecting other subgroups.

5 Formal Properties

We move to the formal properties of our approach. We first detail the background of improper loss functions which motivates our correction given in Section 3. We then present the formal properties of this correction. The useful properties of having $\alpha(\cdot)$ represented by a tree structure is then discussed. Finally, we present a convergence analysis of Algorithm 1 and an alternative boosting scheme.

Can η_f as per (3) correct (any) potential unfairness? Yes. In short, this comes from recent theory in *improper loss functions* for class probability estimation (CPE) [29]. We are interested in the pointwise minimizer (eventually set-valued) of:

$$t_\ell(\eta) \doteq \arg \inf_{u \in [0,1]} L(u, \eta). \quad (9)$$

Dubbed as the *Bayes tilted estimate* of a loss ℓ [29], $t_\ell(\eta)$ is the set of optimal “responses” given a ground truth (pointwise) posterior η . Common loss functions are *proper*: the ground truth value $\eta \in t_\ell(\eta)$ is an optimal response. However, in the case where η cannot be trusted (for instance when it is *unfair*), we may not want to default to imitating η . In addition we also want to make sure that the Bayes tilted estimate can fit to any desired (in our context, *fair*) target. The so-called α -loss ℓ^α , which generalizes the (proper) log-loss, is a good candidate parameterized by a variable α . Its Bayes tilted estimate is the pointwise version of (3), for $\alpha \notin \{0, \infty\}$ and $\eta \neq 1/2$:

$$t_{\ell^\alpha}(\eta) = \{\eta^\alpha / (\eta^\alpha + (1 - \eta)^\alpha)\}. \quad (10)$$

Importantly for α -losses, for any $\eta \notin \{0, 1/2, 1\}$ and any $\eta' \in (0, 1)$, there exists $\alpha \in \mathbb{R}$ in (10) such that $t_{\ell^\alpha}(\eta) = \{\eta'\}$. This property, called *twist-properness* [29], allows for any pointwise correction. By extending α to a function (of $\mathbf{x} \in \mathcal{X}$, as per (3)), twist-properness ensures that given an initial unfair posterior an appropriately learned $\alpha(\cdot)$ can correct any unfairness. This allows for (**flexible**) fairness post-processing – different α functions can be learned for different criteria (*i.e.*, Fig. 1).

Why use the Log-Loss? As per Section 3, we minimize the log-loss. We choose the log-loss for two reasons: ① it is strictly proper and so minimizing $L(\eta_f; M_t, \eta_t)$ (*i.e.*, via TOPDOWN) “pushes” η_f towards target η_t ; and ② it is the α -loss for $\alpha = 1$, so we are guaranteed that for the minimizer $\eta_f \rightarrow \eta_u \iff \alpha \rightarrow 1$. With alternative (*i.e.* non-strictly proper) loss, we might have only “ \Rightarrow ”.

Do we have guarantees on some proximity of η_f with respect to η_u ? Yes, with light assumptions. We examine the (**proximity**) of black-box and post-processed posteriors with the KL divergence [2]:

$$\text{KL}(\eta_u, \eta_f; M) \doteq \mathbb{E}_{(X,Y) \sim D_u} [\log (dD_u(X, Y) / dD_f(X, Y))], \quad (11)$$

where D_u, D_f are the product measures defined from M and their respective posteriors. To bound the proximity (11), we present setting (S1).

(S1) Assumption [1](#) holds for some $0 < B \leq 3$ and function α satisfies $|\alpha(\mathbf{x}) - 1| \leq 1/B$ (a.s.).

This setting lead to the following *data independent* proximity bound.

Theorem 2. For any M , (S1) implies $\text{KL}(\eta_u, \eta_f; M) \leq \pi^2 / (6 \cdot (2 + \exp(B) + \exp(-B)))$.

As an example, fix $B = 3$ for (S1). In this case, we want $\alpha(\cdot) \in [2/3, 4/3]$ (a.s.) which is a reasonable sized interval centered at 1. The clamped black-box posterior's interval is approximately $[0.04, 0.96]$, which is quite flexible and the distortion is upperbounded as $\text{KL}(\eta_u, \eta_f; M) \leq 7.5E - 2$.

Is the composition of transformations meaningful? Yes. The analytical form in [3](#) brings the following easy-to-check (**composability**) property.

Lemma 1. The composition of any two wrapping transformations $\eta_u \xrightarrow{\alpha} \eta_f \xrightarrow{\alpha'} \eta'_f$ following [3](#) is equivalent to the single transformation $\eta_u \xrightarrow{\alpha \cdot \alpha'} \eta'_f$.

This gives an *invertibility* condition – wrapping η_f with $\alpha' = 1/\alpha$ recovers the original black-box η_u .

Given some capacity parameter for η_u , can we easily compute that of η_f ? Yes, e.g., for decision trees. Such a question is particularly relevant for generalization. As we are using the log-loss [2](#), a relevant capacity notion to assess the uniform convergence of risk minimization for the whole wrapped model is the Rademacher (**complexity**) [3](#). We examine the following set of functions:

$$\mathcal{H}_f \doteq \{ \eta_f : \eta_f(\mathbf{x}) \text{ given by } \a href="#">3 \text{ with } \alpha, \eta_u; \forall (\alpha, \eta_u) \}, \quad (12)$$

where we assume known the set of functions from which η_u was trained. We now assume we have a m -training sample $\mathcal{S} \doteq \{(\mathbf{x}_i, y_i) \sim D\}_{i=1}^m$. The empirical Rademacher complexity of a set of functions \mathcal{H} from \mathcal{X} to \mathbb{R} , $\mathfrak{R}_{\mathcal{S}}(\mathcal{H}) \doteq \mathbb{E}_{\sigma} \sup_{h \in \mathcal{H}} \mathbb{E}_i[\sigma_i h(\mathbf{x}_i)]$ (sampling uniform with $\sigma_i \in \{-1, 1\}$), is a capacity parameter that yields efficient control of uniform convergence when the loss used is Lipschitz [3](#) [Theorem 7], which is the case of the log-loss. To see how the α -tree affects the Rademacher complexity of classification using η_f instead of η_u , suppose real-valued prediction based on η_u is achieved via logit mapping, $\iota \circ \eta_u$ [12](#). Such mappings are common for decision trees [26](#).

Lemma 2. Suppose $\{\eta_u\}$ is the set of decision trees of depth $\leq d$ and denote $\mathfrak{R}_{\mathcal{S}}(\text{DT}(d))$ the empirical Rademacher complexity of decision trees of depth $\leq d$ [3](#) and d' the maximum depth allowed for α -trees. Then we have for \mathcal{H}_f in [12](#): $\mathfrak{R}_{\mathcal{S}}(\mathcal{H}_f) \leq \mathfrak{R}_{\mathcal{S}}(\text{DT}(d + d'))$.

The proof is straightforward once we remark that elements in \mathcal{H}_f can be represented as decision trees, where we plug at each leaf of η_u a copy of the α -tree Υ .

Does Algorithm [1](#) have any convergence properties? Yes, it is a boosting algorithm. Following a similar blueprint to classical decision tree induction, it comes with no surprise that TOPDOWN can achieve boosting compliant convergence. To show that TOPDOWN is a boosting algorithm, we need a *Weak Hypothesis Assumption (WHA)*, which postulates informally that each chosen split brings a small edge over random splits for a tailored distribution.

Definition 4. Let $\lambda \in \Lambda(\Upsilon)$ and $D_{\iota\lambda}$ be the product measure on $\mathcal{X} \times \mathcal{Y}$ conditioned on λ . The **balanced** product measure $D'_{\iota\lambda}$ at leaf λ is defined as ($z \doteq (\mathbf{x}, y)$ for short):

$$D'_{\iota\lambda}(z) \doteq \frac{1 - e(M_{\lambda}, \eta_{\iota}) \cdot y \tilde{\iota}(\eta_u(\mathbf{x}))}{1 - e(M_{\lambda}, \eta_{\iota})^2} \cdot D_{\iota\lambda}(z). \quad (13)$$

We check that $\int_{\lambda} dD'_{\iota\lambda} = 1$ because of Def. [2](#). Our balanced distribution is named after [15](#)'s: ours indeed generalizes theirs. The key difference comes from the change in setting, where we consider the alignment purity of a leaf and not its label purity. The “*fairness-free case*” where $\tilde{\iota}(\cdot)$ is replaced by constant 1 yields the original balanced distribution [15](#). We now state our WHA.

Assumption 2. Let $h : \mathcal{X} \rightarrow \mathcal{Y}$ be the function splitting leaf λ . For $\gamma > 0$, then h γ -**witnesses** the *Weak Hypothesis Assumption (WHA)* at λ iff

$$(i) \left| \mathbb{E}_{(\mathbf{x}, Y) \sim D'_{\iota\lambda}} [\Upsilon \tilde{\iota}(\eta_u(\mathbf{X})) \cdot h(\mathbf{X})] \right| \geq \gamma; \quad (ii) e(M_{\lambda}, \eta_{\iota}) \cdot \mathbb{E}_{(\mathbf{x}, Y) \sim D_{\iota\lambda}} [(1 - \tilde{\iota}^2(\eta_u(\mathbf{X}))) \cdot h(\mathbf{X})] \leq 0.$$

Intuitively, (i) gives a condition on the split h 's correlation with unfair posterior η_u agreement with labels and (ii) does the same for the confidence of η_u predictions. Similarly to the balanced

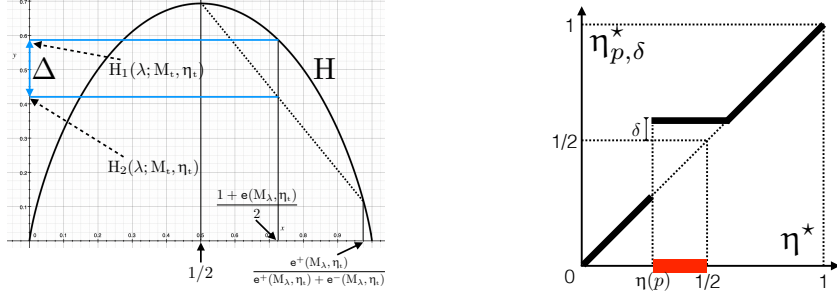


Figure 4: *Left*: Difference between the per-leaf bounds on risk $L(\eta_u; M_t, \eta_t)$ using (8) and Theorem 1 (conservative scoring) and (15) (audacious scoring). Details in the proof of Lemma 3. *Right*: A representation of the (p, δ) -pushup of η^* , where $\eta(p) \doteq \inf \eta^*(\mathcal{X}_p) < 1/2$ (Def. A). All posteriors in $[\eta(p), 1/2 + \delta]$ are mapped to $1/2 + \delta$; others do not change. New posterior $\eta_{p, \delta}^*$ eventually reduces the accuracy of classification for observations whose posterior lands in the thick red interval (x -axis).

distribution, in the fairness-free case where we only care about the label purity of splits, the WHA simplified to that of [15]’s – *i.e.*, only (i) matters. A further discussion on our balanced distribution and WHA is in the SI, Section III. We now state TOPDOWN’s boosting compliant (**convergence**).

Theorem 3. *Suppose (a) Assumption 1 holds, (b) we pick the heaviest leaf to split at each iteration in Step 2.1 of TOPDOWN and (c) $\exists \gamma > 0$ such that each split h^* (Step 2.2) in Υ γ -witnesses the WHA. Then there exists a constant $c > 0$ such that $\forall \varepsilon > 0$, if the number of leaves of Υ satisfies $|\Lambda(\Upsilon)| \geq (1/\varepsilon)^{c \log(\frac{1}{\varepsilon})/\gamma^2}$, then η_f crafted from (3) using TOPDOWN’s Υ achieves $L(\eta_f; M_t, \eta_t) \leq \varepsilon$.*

Are there alternative ways of growing α -trees? Yes. Let us call *conservative* the scoring scheme in (5). There is an alternative scoring scheme, which can lead to substantially larger corrections in absolute values, hence the naming, and yields better entropic bounds for the α -tree.

Definition 5. *For any mixture M_t and posteriors η_u, η_l , let $e^+(M_t, \eta_t)$ and $e^-(M_t, \eta_t)$ be defined by*

$$e^\pm(M_t, \eta_t) \doteq \mathbb{E}_{(X, Y) \sim D_t} [\max\{0, \pm Y \tilde{\tau}(\eta_u(X))\}]. \quad (14)$$

The *audacious* scoring schemes at the leaves of the α -tree replaces (5) in Step 3 by:

$$\Upsilon(\lambda) \doteq \tilde{\tau} \left(\frac{e^+(M_\lambda, \eta_t)}{e^+(M_\lambda, \eta_t) + e^-(M_\lambda, \eta_t)} \right), \quad \forall \lambda \in \Lambda(\Upsilon).$$

Theorem 4. *Suppose Assumption 1 holds and let $H_2(q) \doteq H(q)/\log 2 \in [0, 1]$, H being defined in Definition 3. For any leaf $\lambda \in \Lambda(\Upsilon)$, denote for short:*

$$H_2(\lambda; M_t, \eta_t) \doteq \log(2) \cdot \left(1 + (e_\lambda^+ + e_\lambda^-) \cdot \left(H_2 \left(\frac{e_\lambda^+}{e_\lambda^+ + e_\lambda^-} \right) - 1 \right) \right),$$

where let $e_\lambda^b \doteq e^b(M_\lambda, \eta_t), \forall b \in \{+, -\}$. Using audacious scoring, we get instead of Theorem 1:

$$L(\eta_f; M_t, \eta_t) \leq \mathbb{E}_{\lambda \sim M_{\Lambda(\Upsilon)}} [H_2(\lambda; M_t, \eta_t)]. \quad (15)$$

While upperbounds in Theorem 1 and Theorem 4 may look incomparable, it takes a simple argument to show that (15) is never worse and can be much tighter.

Lemma 3. $\forall \alpha$ -tree $\Upsilon, \mathbb{E}_{\lambda \sim M_{\Lambda(\Upsilon)}} [H_2(\lambda; M_t, \eta_t)] \leq H(\Upsilon; M_t, \eta_t)$.

It thus comes at no surprise that using the audacious scoring also results in a boosting result for TOPDOWN guaranteeing the same rates as in Theorem 3. It also takes a simple picture to show that the per-leaf slack in Lemma 3 can be substantial, a slack which can be represented using a simple picture, see Figure 4 (left), following from the use of Jensen’s inequality in the Lemma’s proof.

As audacious scoring is better boosting-wise, is conservative scoring useful? Yes. If we only cared about accuracy, we would barely have any reason to use the conservative correction. Even thinking about generalization, the Rademacher complexity of decision trees is a function of their depth so faster the convergence, the better [3, Section 4.1]. Adding fairness substantially changes the picture: some constraints, like equality of opportunity (Section 6) can antagonize accuracy to some extent. In such a case, using the conservative correction can keep posteriors η_u and η_l close enough (Theorem 2) so that fairness can be achieved without substantial sacrifice on accuracy.

6 Fairness and Societal Considerations

In this section, we present the fairness guarantees TOPDOWN can achieve. In particular, we provide a discussion about how Theorem 3 can guarantee minimization of the CVAR criteria. Furthermore, we provide alternative inputs to TOPDOWN which allows for EOO to be targeted as a fairness criteria. In the SI Section III, we further present a treatment of statistical parity. Lastly, we discuss how using α -trees provides explainable corrections and how utilization of the sensitive attribute (as per Fig. 3) can be circumvented.

Guarantees on CVaR: As discussed in previous sections, one way to improve the CVAR fairness criteria (as per (4)) is to focus optimization on the worst treated subgroups. Given a specified quantile group β and the set of worse subgroups S_β , we can repeat (CVAR) until $\text{CVAR}_\beta(\eta_t)$ gets below a threshold or (more specifically) its worst tread group gets a risk below a threshold (*i.e.*, a stopping criterion). Importantly, Theorem 3 provides a guarantee: to ensure CVAR_β is below ε , we simply need to boost for $|S|$ times the tree size bound $|\Lambda(Y)|$ given in Theorem 3.

Guarantees on EOO: EOO requires to smooth discrimination within an “advantaged” group, modeled by the label $Y = 1$ [14]. We say that η_t achieves ε -equality of opportunity iff a mapping h_t of η_t to \mathcal{Y} (*e.g.* using the sign of its logit ι) satisfies

$$\max_{s \in S} \mathbb{P}_{X \sim P_s} [h_t(X) = 1] - \min_{s \in S} \mathbb{P}_{X \sim P_s} [h_t(X) = 1] \leq \varepsilon, \quad (16)$$

where P_s is the positive observations’ measure conditioned to value $S = s$ for the sensitive attribute. It is clear that EOO can be antagonistic to accuracy: the rate of advantage in the data D may not be equal among the subgroups. As such, unlike CVAR, we do not want to target the Bayes posterior $\eta_t \neq \eta^*$ for EOO. Instead, we target a skewed posterior which aims to improve the least advantaged subgroup, *i.e.*, increasing $s^\circ \in \arg \min_{s \in S} \mathbb{P}_{X \sim P_s} [h_t(X) = 1]$. Our strategy consists of picking a target posterior which skews part of the original η^* to be more advantaged, thus reducing the LHS of (16) until (16) is satisfied¹. For this, we create a (p, δ) -pushup of η^* , defined in SI Appendix I.

Fig. 4 (right) presents an example of a pushup map. Notice that the pushup only changes the predicted probability of example which do not have a “confident prediction” (the interval $[\eta(p), 1/2 + \delta]$). Intuitively, p controls how many examples are corrected and δ controls how much the correction “pushes up” advantage, further discussion in SI Appendix I. We then run TOPDOWN using as mixture the *positive* measure conditioned to $S = s^\circ$ and $p \doteq \mathbb{P}_{X \sim P_{s^*}} [h_t(X) = 1] + \varepsilon/(K - 1)$, $\delta \doteq K\varepsilon/(K - 1)$, with $K > 1$ user-fixed. Thus, we do

Use TOPDOWN with $M_t \leftarrow P_{s^\circ}$ and $\eta_t \leftarrow \eta_{p,\delta}^*$. (EOO)

Theorem 5. *If TOPDOWN is run until $L(\eta_t; M_t, \eta_t) \leq (\varepsilon^4/2) + \mathbb{E}_{X \sim M_t} [H(\eta_t(X))]$, then after the run we observe $\mathbb{P}_{X \sim P_{s^*}} [h_t(X) = 1] - \mathbb{P}_{X \sim P_{s^\circ}} [h_t(X) = 1] \leq \varepsilon$.*

In the full context of EOO, in the optimization we should not wait to get the bound on $L(\eta_t; M_t, \eta_t)$. Rather, we should make sure (a) we update $\arg \min_{s \in S} \mathbb{P}_{X \sim P_s} [h_t(X) = 1]$ (and thus s°) after each split in the α -tree and (b) we keep $\arg \max_{s \in S} \mathbb{P}_{X \sim P_s} [h_t(X) = 1]$ as is, to prevent switching targets and eventually composing pushup transformations for the same $S = s^\circ$, which would not necessarily comply with our theory. Notably, the guarantee presented in Theorem 5 depends on the mapping h_t and not the direct posterior η_t , as typically considered [14]. When taking a threshold (sign of the logit), h_t can be interpreted as forcing the original posterior to be extreme values of 0 or 1.

Unlike the CVAR case, EOO (as per (17), SI) requires an explicit approximation of η^* . In practice, we find that taking a simple approximation of η^* still can yield fairness gains. However, if one does not want to make such an approximation, one can adapt the statistical parity approach (detailed in SI, Section III). Similarly, if wants to consider the typical EOO definitions depending on posterior values, the target measure can be replaced (*i.e.*, swapping measure M_t with the positive examples P).

Explainability: α -trees using the initialization proposed in Section 4 (and Fig. 3) allows for (explainability) properties similar to that of decision tree classifiers. Fixing a sensitive attribute

¹If we instead *reduce* $\arg \max_{s \in S} \mathbb{P}_{X \sim P_s} [h_t(X) = 1]$ we get a symmetric strategy. The application informs which to use: if positive class implies money spending (*e.g.* loan prediction), then our strategy implies spending more money; while the latter aims to reduce money lent to achieve fairness.

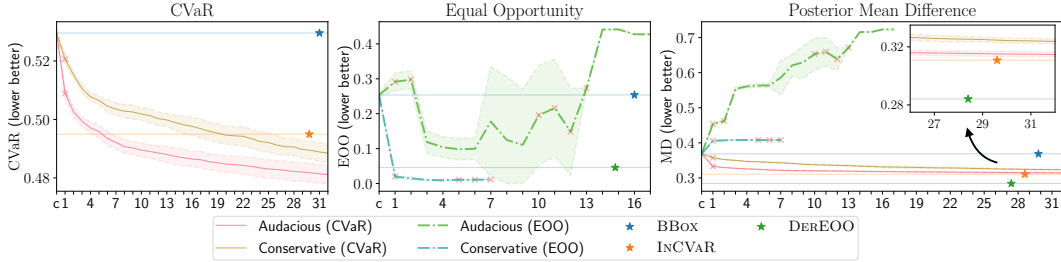


Figure 5: ACS 2015 with Binary Sensitive Attribute and Random Forest Black-box: Evaluation of TOPDOWN over boosting iterations (x-axis) for different fairness criteria. ‘c’ on the x-axis denotes the clipped black-box. ‘x’ denote when a subgroup’s α -tree is initiated (over any fold). The shade denotes \pm a standard deviation from the mean, this disappears when folds have early stopping.

$S = s$, the corresponding sub- α -tree Υ_s can be examined to scrutinize the correction done for the corresponding subgroup. If the splits of the α -tree are simple, similarly to standard decision tree classifiers, corresponding partitions of the input domain can be examined. Furthermore, the type of corrections can also be examined, as discussed in Section 3; where corrections can be classified as “sharpening”, “dampening”, or “polarity flipping” depending on the leaves’ α -values.

Usage of sensitive attribute: Post-processing methods have been flagged in the context of fair classification for the fact that they require explicit access to the sensitive feature at classification time [35, § 6.2.3]. Our basic approach to the induction of α -trees falls in this category (Fig. 3), but there is a simple way to *mask* the use of the sensitive attribute and the polarity of disparate treatment it induces: it consists in first inducing a decision tree to *predict* the sensitive feature based on the other features and use this decision tree as an alternative initialization to naively splitting on subgroups. We thus also *redefine* sensitive groups based on this decision tree – thus alleviating the need to use the sensitive attribute in the α -tree. The use of *proxy sensitive attributes* in a similar manner has seen ample use in a various domain such as health care [6, 5] and finance [13]. We however note that its application in post-process and α -trees may not be appropriate across all domains [8].

7 Experiments

To evaluate TOPDOWN², we consider three datasets presenting a range of different size / feature types, Bank and German Credit (preprocessed by AIF360 [4]) and the American Community Survey (ACS) dataset preprocessed by Folktables³ [9]. The SI (pg 32) presents all results at length (including considerations on proxy sensitive attributes, distribution shift, and interpretability), along with the different black-boxes considered (random forests and neural nets). We concentrate in the Section on the ACS dataset for income prediction in the state of CA and evaluate TOPDOWN’s application to various fairness criteria (as per Section 6 and SI pg 15) with Random Forests (RF). For these experiments, we consider *age* as a binary sensitive attribute with a bin split at 25 (a trinary modality is deferred to the SI). For the black-box, we consider a clipped (Assumption 1 with $B = 1$) random forest (RF) from `scikit-learn` calibrated using Platt’s method [23]. The RF consists of an ensemble of 50 decision trees with a maximum depth of 4 and a random selection of 10% of the training samples per decision tree. Data is split into 3 subsets for black-box training, post-processing training, and testing; consisting of 40:40:20 splits in 5 fold cross validation. For EOO, we utilize an out-of-the-box Gaussian Naive Bayes classifier from `scikit-learn` to approximate η^* .

Multiple fairness criteria We evaluate TOPDOWN for CVAR, equality of opportunity EOO, and statistical parity SP. The complete treatment of SP is pushed to SI (Sections II, XII). SP aims to make subgroup’s expected posteriors similar and is popular in a various post-processing methods [31, 1]. The definition can be found in SI (pg 15) along with the strategy used in TOPDOWN. Conservative and audacious updates rules are also tested. For each of these TOPDOWN configurations, we boost for 32 iterations. The initial α -tree is initialized as in Fig. 3.

We compare against 5 baseline approaches. For CVAR we consider the in-processing approach (INCVAR) presented in [32]. For EOO, we consider a derived predictor (DEREOO) [14]. Our

²Implementation public at: <https://github.com/alexandersoen/alpha-tree-fair-wrappers>

³Public at: <https://github.com/zykls/folktables>

SP baselines include an optimized score transformation approach (OST) [31]; a derived predictor modified for SP (DERSP) [14]; and a randomized threshold optimizer approach (RTO) [1]. We denote the clipped black-box as BBOX. The experiments for CVAR and EOO are summarized in Fig. 5; the full plot with SP is presented at SI Fig. 10. For clarity we only plot the baselines and wrappers which are directly associated to each fairness criteria. We also plot the posterior mean difference between the data and debiased posteriors MD (0/1 loss) to examine the effects on accuracy.

For CVAR, both conservative and audacious approaches decreases CVAR, which results in better CVAR values than both the original BBOX and in-processing baseline INCVAR – which is good news since INCVAR directly optimizes CVAR. We note that there are cases in which the in-processing approach is better than ours (trinary sensitive attributes in SI), but this is expected given INCVAR’s optimization goal. Interestingly, the audacious update is superior in both CVAR and MD than the conservative update. This is also consistent for trinary sensitive attributes. Thus, the audacious update is desirable when optimizing CVAR. Another observation is that only one sensitive attribute subgroup’s α -tree is initialized (only one ‘×’). This indicates that after 32 iterations the worse case subgroup does not change in the binary case.

For EOO, there is a huge difference between conservative and audacious updates as the former gets to the most fair outcomes of all baselines. Even if we used early stopping or pruning of the α -tree (taking an earlier iterations) the audacious update would fail at producing outcomes as fair as its conservative counterpart. Furthermore, the audacious update comes with a significant degradation of accuracy MD. Furthermore, by looking at the iterations in which subgroup α -trees are initialized, the audacious update causes large (primarily bad) jumps in performance. This rejoins our remark on the interest of having a conservative update in Section 5. When compared to DEREOO, we find that the conservative TOPDOWN approach produces lower EOO. However, DEREOO tend to have better accuracy scores in MD. These observations are consistent with the trinary sensitive attribute (SI).

For SP, we can observe fairness results that can be on par with contenders for the conservative update, but observe a substantial degradation of MD. This, we believe, follows from a simple plug-in instantiation of M, η_t for the fairness notion in SI Section II resulting in potentially harsh updates. In SI (pg 15), we discuss an alternative approach using ties with optimal transport.

8 Limitations and Conclusion

Given the context of fairness, it is important to highlight possible limitations of our approach and the potential social harm from such misuse. We highlight two of these for our TOPDOWN approach. Firstly, our approach has shown to have failure cases in *small data cases*. This can be seen in the experiments on German Credit dataset (SI, Section XIII, XV). This, we believe, has a formal basis in our approach. For instances, EOO requires accurate data posterior estimation which may be difficult in small data regimes. Secondly, TOPDOWN is of course not unilaterally better than all other post-processing fairness approaches – there is *No Free Lunch*. As such, we do not claim that our instantiation of TOPDOWN is optimal in CVAR, EOO, or SP. However, considering that such different fairness models instantiated in the *same* algorithm can lead to competitive results with the respective state of the art, the avenue for improved instantiations or accurate extensions to new fairness constraints appears promising. We leave these for future work.

Acknowledgments and Disclosure of Funding

YM received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 882396), the Israel Science Foundation (grant number 993/17), Tel Aviv University Center for AI and Data Science (TAD), and the Yandex Initiative for Machine Learning at Tel Aviv University. AS and LX thank members of the ANU Humanising Machine Intelligence program for discussions on fairness and ethical concerns in AI, and the NeCTAR Research Cloud for providing computational resources, an Australian research platform supported by the National Collaborative Research Infrastructure Strategy.

References

- [1] I. Alabdulmohsin and M. Lucic. A near-optimal algorithm for debiasing trained machine learning models. In *NeurIPS*34*, 2021.
- [2] S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. Oxford University Press, 2000.
- [3] P.-L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.
- [4] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *IBM J. Res. Dev.*, 63:4:1–4:15, 2019.
- [5] David P Brown, Caprice Knapp, Kimberly Baker, and Meggen Kaufmann. Using bayesian imputation to assess racial and ethnic disparities in pediatric performance measures. *Health services research*, 51(3):1095–1108, 2016.
- [6] Consumer Financial Protection Bureau. Using publicly available information to proxy for unidentified race and ethnicity: A methodology and assessment. *Washington, DC: CFPB, Summer*, 2014.
- [7] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. *CoRR*, abs/1701.08230, 2017.
- [8] Anupam Datta, Matthew Fredrikson, Gihyuk Ko, Piotr Mardziel, and Shayak Sen. Use privacy in data-driven systems: Theory and experiments with machine learnt programs. In *24th ACM SIGSAC*, pages 1193–1210, 2017.
- [9] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *NeurIPS*34*, 2021.
- [10] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R.-S. Zemel. Fairness through awareness. In *ITCS'12*, pages 214–226, 2012.
- [11] C. Dwork, N. Immorlica, A.-T. Kalai, and M.-D.-M. Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, volume 81, pages 119–133. PMLR, 2018.
- [12] 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, 2012.
- [13] Allen M Fremont, Arlene Bierman, Steve L Wickstrom, Chloe E Bird, Mona Shah, José J Escarce, Thomas Horstman, and Thomas Rector. Use of geocoding in managed care settings to identify quality disparities. *Health Affairs*, 24(2):516–526, 2005.
- [14] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NeurIPS'16*, pages 3315–3323, 2016.
- [15] M.J. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. In *Proc. of the 28th ACM STOC*, pages 459–468, 1996.
- [16] M.-P. Kim, A. Ghorbani, and J.-Y. Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254. ACM, 2019.
- [17] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

- [18] P.-K. Lohia, K.-N. Ramamurthy, M. Bhide, D. Saha, K.-R. Varshney, and R. Puri. Bias mitigation post-processing for individual and group fairness. In *ICASSP'19*, pages 2847–2851. IEEE, 2019.
- [19] K. Martineau. Shrinking deep learning’s carbon footprint. <https://news.mit.edu/2020/shrinking-deep-learning-carbon-footprint-0807>, 2020.
- [20] N. Mehrabi, F. Morstatter, N. Saxena, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM CSUR*, 54:1–35, 2022.
- [21] A.-K. Menon and R.-C. Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, volume 81, pages 107–118. PMLR, 2018.
- [22] F. Petersen, D. Mukherjee, Y. Sun, and M. Yurochkin. Post-processing for individual fairness. *CoRR*, abs/2110.13796, 2021.
- [23] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.
- [24] M.-D. Reid and R.-C. Williamson. Information, divergence and risk for binary experiments. *JMLR*, 12:731–817, 2011.
- [25] R.-T. Rockafellar and S. Uryasev. Optimisation of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.
- [26] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *9th COLT*, pages 80–91, 1998.
- [27] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *MLJ*, 37:297–336, 1999.
- [28] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In *ACL'19*, pages 3645–3650, 2019.
- [29] T. Sypherd, R. Nock, and L. Sankar. Being properly improper. In *ICML'22*, 2022.
- [30] T. van Erven and P. Harremoës. Rényi divergence and kullback-leibler divergence. *IEEE Trans. IT*, 60:3797–3820, 2014.
- [31] D. Wei, K.-N. Ramamurthy, and F. du Pin Calmon. Optimized score transformation for fair classification. In *AISTATS'20*, volume 108, pages 1673–1683, 2020.
- [32] Robert C. Williamson and Aditya Krishna Menon. Fairness risk measures. In *International Conference on Machine Learning*, pages 6786–6797, 2019.
- [33] B.-E. Woodworth, S. Gunasekar, M.-I. Ohannessian, and N. Srebro. Learning non-discriminatory predictors. In *COLT'17*, volume 65, pages 1920–1953. PMLR, 2017.
- [34] Forest Yang, Mouhamadou Cisse, and Oluwasanmi O Koyejo. Fairness with overlapping groups; a probabilistic perspective. *Advances in Neural Information Processing Systems*, 33, 2020.
- [35] M.-B. Zafar, I. Valera, M. Gomez-Rodriguez, and K.-P. Gummadi. Fairness constraints: A flexible approach for fair classification. *JMLR*, 20:75:1–75:42, 2019.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
 - (b) Did you describe the limitations of your work? [Yes] At the end of the last page.
 - (c) Did you discuss any potential negative societal impacts of your work? [Yes] In limitation section and corresponding Appendix sections.
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [Yes] All assumptions are stated in the Theorems, Lemmas, and Corollaries.
 - (b) Did you include complete proofs of all theoretical results? [Yes] Full proofs are presented in the Appendix.
3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See code in supplementary material.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Summary presented in main-text with further details in Appendix.
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Error bars are present in all relevant plots.
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details present in Appendix [X1](#).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [Yes] Data sources and baselines cited.
 - (b) Did you mention the license of the assets? [Yes] Yes, when available.
 - (c) Did you include any new assets either in the supplemental material or as a URL? [No]
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We use public datasets. For the newer dataset, we defer to the cited paper's Section.
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The datasets are standard benchmarks used extensively in ML. For the newer dataset, we defer to the reference's Section.
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]